面向下一代互联网实验平台的新型报文处理模型——EasySwitch

李 韬 孙志刚 陈一骄 贾春波 苏 琪 郭腾飞

(国防科学技术大学计算机学院 长沙 410073)

摘 要 下一代互联网实验平台应能提供网络原型系统快速开发及部署能力,以有效支撑新型互联网体系结构关键技术的实现和验证.基于 FPGA(Field Programmable Gate Array)技术构建的网络实验平台可以提供较高的可编程性和性能,而它对硬件逻辑设计能力的要求则严重限制了平台的广泛应用.新型报文处理模型 EasySwitch 通过优化设计并预置通用报文交换及处理逻辑,有效实现用户定制报文处理逻辑与通用报文处理逻辑解耦;良定义的用户模块接口则使用户仅需关注业务特定逻辑实现,有效简化用户逻辑设计.理论分析表明,EasySwitch 可通过提供确定性资源约束模型,有效支持 FPGA 资源的优化利用.此外,EasySwitch 具有较低的报文调度处理延迟,对实验系统输入流量真实特性影响较小.EasySwitch 模型在 NetMagic 平台的有效实现及应用表明该模型可为下一代互联网新型报文处理机制及协议的快速设计、开发和验证提供有力支撑.

关键词 报文处理模型;实验平台;报文交换;NetMagic 平台;FPGA

中图法分类号 TP393 **DOI 号**: 10.3724/SP.J.1016.2011.02187

A Novel Packet Processing Model for Next-Generation Internet Experimental Platform——EasySwitch

LI Tao SUN Zhi-Gang CHEN Yi-Jiao JIA Chun-Bo SU Qi GUO Teng-Fei

(*School of Computer, National University of Defense Technology, Changsha* 410073)

Abstract To support the implementation and verification of the key technologies of the next-generation Internet architecture, the corresponding network experimental platforms should provide the ability of rapid development and deployment of network prototype systems. The FPGA (Field Programmable Gate Array) based experimental platforms provide both high programmability and high performance. However, the requirement of the hardware logic design ability greatly limits the widespread application of these platforms. A novel packet processing model, EasySwitch, is proposed to reduce the barriers of the network experimental system design and development. By pre-placing the common packet switching and processing functions, EasySwitch decouples the customized packet processing logic and the common packet processing logic. Its well-defined standard User Module socket interfaces allow the users to focus on implementing their own logic, simplifying the programming of hardware logic. Theoretical analysis shows that EasySwitch can optimize the hardware resource utilization of the FPGA with its deterministic resource constraint model. Moreover, with the low packet scheduling latency, EasySwitch model has less impact on the real characteristics of the input traffic from the experimental system. The imple-

收稿日期:2011-08-29;最终修改稿收到日期:2011-09-16.本课题得到国家"九七三"重点基础研究发展规划"大规模流媒体高效传输技术"(2009CB320503)资助.**李 韬**,男,1983 年生,博士,助理研究员,主要研究方向为网络处理器、路由与交换.E-mail: taoli.nudt@gmail.com. **孙志刚**,男,1973 年生,博士,研究员,主要研究领域为网络通信、路由与交换. **陈一骄**,男,1972 年生,博士,副研究员,主要研究方向为网络通信、可重构路由器. **贾春波**,男,1987 年生,硕士研究生,主要研究方向为网络测量、网络交换. **苏 琪**,男,1985 年生,硕士研究生,主要研究方向为网络协议、网络测量. **郭腾飞**,男,1984 年生,硕士研究生,主要研究方向为网络协议、网络测量.

mentation of EasySwitch model on the NetMagic platform demonstrates that EasySwitch can provide efficient support for the rapid design, development and verification of the novel packet processing mechanisms and network protocols.

Keywords packet processing model; experimental platform; packet switching; NetMagic platform; FPGA

1 引 言

在真实网络环境中进行实验并收集数据结果是下一代互联网创新技术研究最有效和最具说服力的手段.因此,下一代互联网体系结构研究[1]无论是采用渐进式演进路线(如中国下一代互联网示范工程CNGI)还是Clean Slate革命性路线(如美国国家科学基金会FIND项目)都强调网络实验平台在实现和验证新型网络协议和服务方面的关键地位.NSF GENI(Global Environments for Network Innovation)计划、EU FP7 FIRE计划,以及日本的JGNX计划都试图通过构建大规模可编程实验床支持下一代互联网技术研究.

以路由器为代表的网络设备作为支撑互联网运行的基础,是支撑下一代互联网实验平台构建的关键.为有效支持以软件定义网络、虚拟化、原语扩展等下一代互联网技术,支撑网络技术研究的实验平台必须提供可编程、可重构以及可重用特性,能有效支持网络技术快速开发与部署,降低验证测试成本,匹配新型互联网功能原语的时空演化特性.

面向上述需求,出现了基于通用微处理器(纯软件)、网络处理器、图形处理单元GPU(Graphics Processing Unit)、ASIC(Application-Specific Integrated Circut)以及FPGA(Field Programmable Gate Array)等的网络实验平台.其中,FPGA技术由于具有可重构能力,可取得在性能和灵活性方面的良好折衷,目前受到广泛关注.作为采用FPGA技术的典型代表,NetFPGA平台[2]基于模块化可重用设计思想,提供清晰的接口及丰富的参考设计,广泛应用于网络教学和科研.然而,NetFPGA平台缺乏支持用户开发的优化报文处理模型,难以为用户逻辑功能的规划设计提供有效支撑.SwitchBlade[3]基于流水化思想,设计了一种可定制报文处理模型,能够有效支持新型网络协议和虚拟化技术.然而,为减少硬件综合时间,该模型预置了丰富的可配置流水线功能模块,占用硬件资源较多,限制了用户开发灵活性.

基于FPGA技术的实验平台设计门槛较高,需要开发者具有一定硬件开发及设计经验,很多平台还需要用户提前规划实验方案的性能指标和硬件资源使用,从而阻碍了FPGA实验平台的广泛应用.面向下一代互联网技术的实验平台及相应模型应能够最大限度简化用户硬件逻辑开发,降低设计开发门槛.此外,针对FPGA硬件资源有限的特点还应提供确定性资源占用及性能评估模型,以指导和协助用户完成设计规划.

针对上述需求,本文提出了一种面向下一代互联网实验平台的新型报文处理模型——EasySwitch.EasySwitch模型主要特点包括:

(1)通过预置通用报文交换处理逻辑及良定义用户逻辑开发接口,实现用户业务逻辑与通用处理逻辑显式分离;

(2)支持并行数据平面和资源隔离,支持网络虚拟化技术;

(3)提供确定性模型指导硬件资源配置及性能优化,为用户逻辑规划和开发提供充分设计裕度.

2 相关工作

随着下一代互联网研究的开展和深入,支撑实验床构建的网络设备平台体系结构和处理模型也逐渐引起关注.

Click模块化路由器[4]支持以软件方式进行定制协议和报文转发操作的快速开发.而基于内核的报文转发很难实现线速处理性能.高性能软件路由器RouteBricks[5]基于商用多核处理器实现可获得较高报文转发性能,但可扩展性受限于连接网卡和CPU的PCI-E(Peripheral Component Interconnect Express)总线带宽.此外,采用RouteBricks开发的原型也很难移植到硬件上.PacketShader[6]利用GPU加速软件路由器报文处理过程,通过应用程序(库程序)控制硬件转发过程,可以达到每秒数百万报文的处理速度.然而,所采用的批处理方式会引入较高的报文处理延迟[3].

Supercharged PlanetLab（SPP）[7] 基于 Intel IXP 网络处理器实现报文数据平面处理，具有同时线速处理来自多个端口报文的能力. 由于部署代价昂贵，目前仅适合在大规模交换中心中部署. PNIC[8] 采用配备了 IXP2855 16 核网络处理器的 Netronome NFE-i8000 构建，支持数据中心网络相关技术，可以灵活实现虚拟网卡、Openflow 交换[9] 及时钟同步等功能. 基于网络处理器的方式可以获得较高的处理性能和编程灵活性. 然而，编程开发与特定厂商平台相关，缺乏可移植性，限制了其数据平面功能的优化实现与重用.

PLUG[10] 提供了一个编程模型框架用于实现高性能报文查找芯片，可以获得高速报文处理能力，然而全定制芯片设计代价较高，开发周期较长. ServerSwitch[11] 采用集成商用交换芯片方式实现报文定制处理，由服务器 CPU 负责报文控制平面和数据平面处理，可以有效实现数据中心网络多种组网方式所需的路由交换功能. 然而商用交换芯片的可编程能力仍然局限于特定报文字段的匹配处理，难以有效支持新型网络协议的实现与验证.

与基于 ASIC 的平台相比，基于 FPGA 的设计不仅可以提供功能可重构能力，还可以极大缩短开发和部署周期；相比基于软件的平台，在性能方面更具优势. 例如，NetFPGA 平台作为一款面向网络课程教学的评估卡，在教学科研中已得到十分广泛的应用. NetFPGA 通过提供丰富的参考设计（如 IPv4 路由器、OpenFlow 交换机等），以重用和修改移植等方式，为用户逻辑功能实现提供支持. 然而，由于缺乏对用户逻辑和平台预置逻辑进行清晰划分的报文处理模型，用户逻辑开发受限于特定参考设计的复杂度和相似度，在一定程度上制约了网络实验原型系统的快速构建.

基于 FPGA 实现的 OpenFlow 交换模型和 SwitchBlade 处理模型[3] 都试图通过硬件提供丰富报文处理功能集合以增强新型网络技术的快速开发和部署能力. 然而，互联网技术的演进性将导致预置报文处理功能集合不断膨胀，很难期望能够获得一个稳定的功能基底. 此外，上述模型预置的通用报文处理逻辑较为复杂，与实验无关的特定功能逻辑占用较多 FPGA 硬件资源，难以为用户开发提供充足的可用资源.

3　EasySwitch 处理模型

EasySwitch 处理模型面向可重构 FPGA 技术提出，试图通过预置优化的报文交换及处理功能，简化用户逻辑开发及调试工作，并通过定义简洁而清晰的接口，将用户业务逻辑与预置逻辑显式分离，使用户可以专注于业务逻辑开发，而不需了解其它外围模块的功能实现细节.

与已有模型不同，EasySwitch 处理模型充分考虑 FPGA 资源的有限性，仅试图通过预置网络设备（路由器、交换机等）基础、稳定的通用报文处理功能，即报文交换功能，将报文分类、查表匹配、深度报文检测等高级报文处理功能以可重用硬件模块构件方式提供给用户，供用户按需集成. 除报文交换功能外，EasySwitch 还基于聚类合并方法，预置了核心报文处理功能逻辑. 基于对通用报文处理操作的归纳总结，通过编码形成报文处理规则集合.

如图 1 所示，EasySwitch 报文处理模型主要由三部分组成，即输入控制 IC（Input Control）、输出控制 OC（Output Control）以及用户模块 UM（User Module）. 基本的报文交换功能由 IC/OC 中输入调度器 IS（Input Scheduler）和输出调度器 OS（Output Scheduler）协同完成，而报文处理规则指定的报文处理操作则由 OC 通过对来自 UM 的处理规则信号解码完成. IC/OC 是 EasySwitch 模型定义的预置处理逻辑，UM 承载用户业务处理逻辑实现. 用户模块接口 UMS（User Module Socket）则显式定义了预置处理逻辑与用户自定义逻辑间的接口，用户逻辑的开发只需考虑满足 UMS 规范要求，无须关心预置逻辑的具体实现.

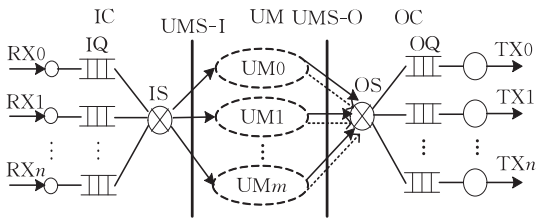

图 1　EasySwitch 处理模型

为有效支持网络虚拟化技术，EasySwitch 模型支持集成多个 UM 实现并行数据平面. 通过为多个 UM 提供独立的处理和存储资源（如转发表），可以实现并行数据平面间的资源隔离. EasySwitch 也允许多个 UM 间的资源共享，避免资源复制，以提高 FPGA 资源利用率.

3.1　输入/输出控制

EasySwitch 模型中的输入控制负责接收来自多个网络接口的报文，在完成校验后进行汇聚，并根据报文分类规则分派到一个或多个 UM 中处理，同一报文仅允许分派到一个 UM. 由于多时钟域设计

会加剧 FPGA 逻辑设计复杂度，EasySwitch 模型要求各 UM 以及 UMS 运行在统一的时钟频率上. 输出控制 OC 负责接收 UM 处理完成后的报文以及相应的处理规则，通过译码处理规则执行对该报文指定的处理操作(如截断、复制等)，之后将报文发送到目标输出缓冲队列.

EasySwitch 模型中输入调度器 IS 采用连续工作模式(即若某输入缓冲队列中包含报文尾，调度器将调度输出该完整报文，保证输出链路不空闲)，对输入缓冲队列 IQ(Input Queue)进行调度. 由于不能预先假定 UM 对 IQ 报文的处理策略，IS 采用公平调度算法. 可重用硬件模块库中也包含支持贪婪、加权轮转等其它调度算法的 IS，可供用户选择集成. 输出调度器 OS 主要负责响应并处理来自 UM 的报文处理请求，也可采用类似 IS 的连续工作模式和公平调度算法，以防止请求“饿死”. 各 UM 也可以选择集成内部输出缓冲队列，基于灵活性考虑，其管理和配置由用户根据应用需求确定，EasySwitch 模型不作规定.

3.2 用户模块 UM

EasySwitch 模型可灵活支持多种 UM 开发模式. 以单 UM 为例，包括旁路处理(bypass)模式、穿透处理(pass-through)模式以及混杂处理(promiscuous)模式.

(1)旁路处理模式. UM 不修改报文内容，只根据报文中提取的关键字，通过查表等方式，决定报文的处理行为. 通过在 IC 与 OC 间设置输入报文缓冲 PB(Packet Buffer)，到达的报文可在 PB 中缓存，OC 在接收到 UM 的处理控制信息后，从 PB 中读取报文，并对其执行相应的处理.

(2)穿透处理模式. IQ 中报文全部进入 UM，并在 UM 中缓存和处理. UM 可以直接修改报文字段，进行如地址替换、TTL(Time To Live)更新等操作. OC 接收 UM 处理完成后的报文，并按照对应的规则对报文进行相应的转发控制操作.

(3)混杂处理模式. UM 不修改报文内容，到达的报文在 PB 中缓存，并根据输入报文信息构造产生新的报文. UM 必须产生与原始输入报文和新产生报文一一对应的处理规则，并发送到输出控制，输出控制根据处理规则，选择 PB 或 UM 中对应报文完成指定转发控制操作. 该模式不支持多 UM 实现.

需要注意的是，为简化控制复杂度，EasySwitch 模型要求多 UM 必须采用同一模式设计实现.

3.3 UMS 接口

UMS 定义了用户模块 UM 与预置模块 IC/OC 间的接口，即 UMS-I 和 UMS-O，所有满足 UMS 接口约束的 UM 都可无缝集成到 EasySwitch 模型.

UMS-I 为报文输入接口，是 IC 与 UM 间的通信接口，采用简单的报文 FIFO(First In First Out)接口定义. UMS-I 接口包括带内数据和带外信息两部分. 带内数据为完成校验处理的有效报文数据，数据宽度通常设为多个字节以提高报文处理流量，带外信息则包括编码后的报文头尾标识、输入接口号等.

UMS-O 为报文输出接口，是 OC 与 UM 间的通信接口，由报文数据接口和输出控制接口组成. 报文数据接口定义与 UMS-I 接口类似，其中带内报文数据位宽与 UMS-I 相同，带外信息不包含报文输入接口号信息. 输出控制接口用于传递 UM 对报文的决策结果，以报文处理规则形式定义，OC 通过译码处理规则，确定报文的下一步处理动作. EasySwitch 模型定义并实现的报文处理规则由以下字段构成：

(1)接口 bitmap，用于选择接口集合中的一个或多个接口输出报文；

(2)操作字段，指定处理动作即丢弃、转发或截断；

(3)报文截断长度；

(4)保留字段.

上述处理规则各字段通过相互组合可以覆盖按端口转发、多端口复制转发、丢弃、截断后单端口或多端口报文转发等多种报文转发处理操作，并通过设置保留字段可支持功能扩展. 此外，输出处理规则还定义了报文规则绑定指示，用于选择从 UM 或 PB 接收报文.

4 资源及性能分析

对于基于 FPGA 的网络实验平台，由于用户硬件逻辑设计受 FPGA 片上资源、处理频率等多方面因素影响，在进行实验方案设计时，必须对方案的资源占用及性能指标进行评估. NetFPGA 采用基于参考设计的开发方式，用户业务逻辑与预置逻辑间缺乏明显划分，无法有效评估预置逻辑对业务逻辑在资源和性能等方面的影响；SwitchBlade 模型中预置处理逻辑则过于复杂，也难以提供精确的资源及性能分析评估模型.

EasySwitch 模型基于简洁的 UMS 接口，实现了用户逻辑与预置逻辑的清晰分离，可以提供确定性资源约束及性能评估模型. 下面介绍面向接口数目及缓冲队列长度需求建立的 EasySwitch 资源约

束模型，并针对网络测量应用需求，评估 EasySwitch 模型报文调度处理性能. 表 1 列出了 EasySwitch 模型资源及性能分析涉及的相关参数. 简化起见，网络接口设为千兆以太网，内部数据通路(FIFO 接口)位宽为 128 位，IS 采用连续工作模式.

表 1　EasySwitch 处理模型相关参数

N	输入接口的数目	B	输入缓冲队列容量
V_i	网络接口速率(1 Gbps)	f	IS(即数据路径)的处理频率(Hz)
V_o	IS 调度输出速率($V_o=f\times B_d$)	L	链路报文的最大长度(1500 Bytes)
$I_i(t)$	t 时刻第 i 个输入缓冲队列长度	B_d	数据通路位宽
C	FPGA 缓冲队列容量上限		

4.1　资源约束模型接口

EasySwitch 模型中的预置逻辑在 FPGA 上实现必须满足以下约束：一方面，FPGA 片上资源(尤其是存储资源)有限，EasySwitch 模型中缓冲队列(即 IQ/OQ)占用存储资源比例不应过大，影响用户逻辑开发自由度；另一方面，EasySwitch 模型必须保证在报文处理过程中预置逻辑不能造成输入报文丢失，即输入缓冲队列不应出现溢出，这要求输入缓冲队列必须有充足的缓冲存储资源①. 上述约束模型 RS 可以形式化为

$$RS:\begin{cases} N\times B\leq C & (1)\\ N<(f\times B_d\times B+B\times V_i-\\ \quad f\times B_d\times L)/(B\times V_i) & (2)\end{cases}$$

其中，式(1)反映了 FPGA 存储资源约束，式(2)则用于保证输入缓冲队列在公平调度情况下不会出现溢出，相关证明参见附录中定理 1.

以 EasySwitch 处理模型在 NetMagic-24② 和 NetFPGA 网络实验平台的实现为例，假设允许输入缓冲队列最多占用 40% 存储资源. NetMagic-24 平台 FPGA 存储容量约为 8×10^6 bits($C_{NetMagic}=3.2\times10^6$ bits)，数据通路位宽 B_d 为128 位；NetFPGA 平台 FPGA 存储容量约为 4.9×10^6 bits($C_{NetFPGA}=1.96\times10^6$ bits)，数据通路位宽 B_d 为 64 位. 在上述条件下，图 2 和图 3 分别给出了 NetMagic-24 及 NetFPGA 平台在不同 IS 处理频率下，接口数目 N 与输入缓冲队列容量 B 间的约束关系.

由图可知，数据通路(调度器)处理频率越高，可支持的接口数目越多. 如图 2 所示，NetMagic 平台下，处理频率为 75 MHz 时，最多可支持 10 个千兆接口；100 MHz 时，则最多可支持 14 个千兆接口. 经过综合仿真，EasySwitch 处理模型在 NetMagic-24 平台最高数据通路时钟频率 f_{max} 可达 169.43 MHz，考虑到用户业务逻辑的嵌入可能会导致综合后 f_{max} 的下降，为保留设计裕度，其数据通路时钟频率 f 设定为 125 MHz. 因此，由图 2 可知，在该频率下 NetMagic-24 最多可支持 16 个千兆接口. 由于 NetFPGA 平台数据通路时钟频率也设定在 125 MHz，由图 3 可知，若该平台采用 EasySwitch 处理模型，在该时钟频率下，最多可支持 8 个千兆接口；若时钟频率提高到 150 MHz，NetFPGA 则最多可支持 10 个千兆接口. 然而，NetFPGA 实际仅集成了 4 个千兆接口. NetFPGA 采取保守设计的主要原因在于采用参考路由器作为预置逻辑实现，功能复杂，资源占用较多，无法保证更多接口报文的线速处理.

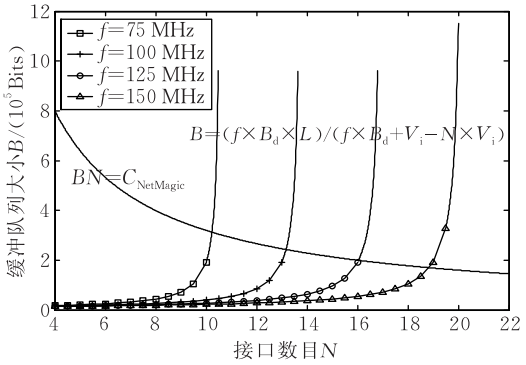

图 2　NetMagic-24 平台资源约束

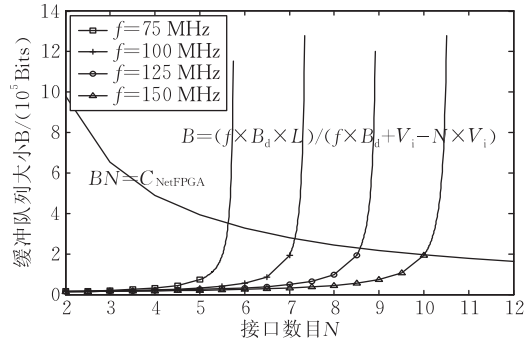

图 3　NetFPGA 平台资源约束

4.2　报文调度性能

基于 EasySwitch 处理模型的网络实验平台可以支持多种应用部署场景，不仅可作为报文流旁路处理设备，也具有部署在报文数据通路内处理的能力. 后者是支持网络测量技术实现和验证极为重要的手段. 对于测量精度要求较高的实验，EasySwitch 模型可保证调度处理对输入报文流量特性的影响可

① 由于无法预知和假设 UM 报文向 OQ 写入速率，OQ 长度难以通过理论分析获得，通常采用经验值设定.

② 参见 NetMagic 官方网站 www.netmagic.org

确定或可忽略.下面将具体分析 EasySwitch 处理模型报文调度性能.

EasySwitch 模型对报文 p 处理调度延迟 $E(p)$ 可以分为两部分,即报文传输延迟以及排队调度延迟.

$$E(p)=T(p)+Q(p) \tag{3}$$

其中,$T(p)$表示报文 p 进入 UM 处理前经过输入缓冲队列的传输延迟(即报文头进至尾进延迟),$Q(p)$表示报文 p 经过输入控制 IC 的排队调度延迟(即报文尾进至头出延迟).

由于缓冲队列分布、数据通路位宽及处理频率在实现时已知,且报文 p 的长度 $l(p)$可在接收时获得,因此 $T(p)$可由式(4)计算获得.对于网络测量等应用场景,$T(p)$可以通过在 UM 中利用时间戳修正机制消除.

$$T(p)=\lceil l(p)/(f\times B_d)\rceil \tag{4}$$

在 EasySwitch 处理模型中,不可修正的性能误差主要由 $Q(p)$构成.由附录 1 中定理 2 可知,报文 p 在 N 进 1 出的连续工作模式调度系统中,排队调度延迟 $Q(p)$具有上限,即

$$Q(p)<(N\times L-l(p))/(f\times B_d-(N-1)V_i) \tag{5}$$

根据式(5),图 4 给出了在 IS 不同处理频率下,最大报文调度延迟与网络接口数目的关系.

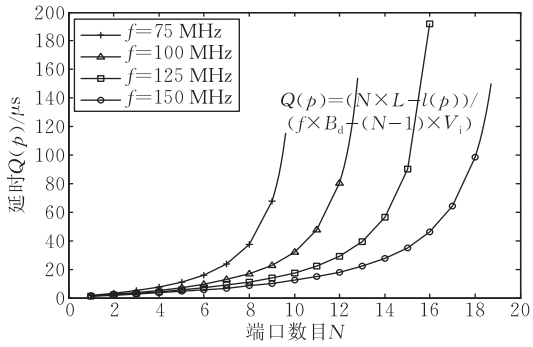

图 4 报文调度延迟分析

由图可知,相同接口数目下,调度器处理频率越高,报文调度处理延迟越低.对于特定处理频率,支持的输入接口数目不应超过某一特定值,否则报文调度延迟将急剧增长,以 $f=100$ MHz 为例,支持接口数目 N 应小于 12,以保证获得可接受的报文调度延迟.此外,对于延迟敏感的应用,可以通过提高处理频率及减少占用接口的方法降低 EasySwitch 模型调度延迟对系统延迟的影响,通常可以控制在 100 μs 内.

5 模型实现与应用

我们基于 NetMagic 平台对 EasySwitch 处理模型进行了实现和验证.如图 5 所示,NetMagic 目前提供 NetMagic-24 和 NetMagic-08 两个版本.NetMagic-24 采用部分可编程交换体系结构,将大容量 FPGA(MagicFPGA)与商用交换芯片相结合,提供可编程性与性能的良好折中.24 个千兆以太网接口中,16 个由 MagicFPGA(Altera Arria II GX EP2AGX125)控制,具有完全可编程能力.NetMagic-08 是 NetMagic-24 的 lite 版本,采用低成本和小型化设计方案,基于 Altera Arria II GX EP2AGX45 FPGA 提供共 8 个(4 个光口 4 个电口)完全可编程千兆以太网接口.

(a) NetMagic-24

(b) NetMagic-08

图 5 开放式可重构网络实验平台 NetMagic

5.1 基本结构

如图 6 所示,MagicFPGA 是 NetMagic 平台实现报文处理可重构的核心,EasySwitch 处理模型在 MagicFPGA 中以预置方式实现,支持 8 或 16 个千兆以太网接口报文接收,并提供转发、截断、复制等 EasySwitch 定义的基本报文处理规则.此外,EasySwitch 模块对平台集成的外部存储器(如

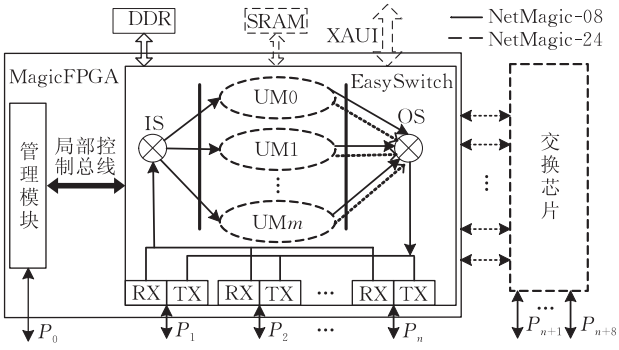

图 6 EasySwitch 处理模型实现

DDR、SRAM)以及高速 XAUI 接口等进行封装，为 UM 提供良定义访问接口以及仲裁机制，有效隔离各 UM 访问资源. 用户仅需专注于 UM 业务逻辑，而不必考虑网络接口、数据缓冲、平台管理等其它外围逻辑的具体实现. UM 与其它外围逻辑之间数据通路位宽为 128 bits，处理频率为 125 MHz.

EasySwtich 模型仅定义报文数据平面处理规范，NetMagic 平台提供基于以太网的通用管理控制接口及协议，为控制平面处理功能实现提供兼容性和可移植性支持. 用户可在任何带有以太网接口的主机或平台上，通过 Socket 编程，按照 NetMagic 访问控制(NetMagic Accesss and Control，NMAC)协议标准①，将管理控制命令封装成标准以太网报文，发送到 NetMagic 管理控制接口 P0 对平台进行管理和控制. MagicFPGA 内部的管理模块负责基于 NMAC 协议与外部控制器建立连接，并对管理报文中封装的命令进行解析，转换为局部控制总线信号，完成对平台中 EasySwitch 等模块及存储器的初始化配置及管理控制.

表 2 及表 3 给出了 NetMagic-24 平台中管理模块及全配置(接口数目 $N=16$)EasySwitch 模型的资源占用情况.

表 2　管理模块资源占用

资源名称	使用数量	占用比例/%
查找表 LUT	2 383	2
寄存器	2 262	2
存储器位	285 184	3

表 3　EasySwitch 模块资源占用

资源名称	使用数量	占用比例/%
查找表 LUT	5 781	6
寄存器	8 327	8
存储器位	3 709 632	46

表 4 给出了在支持 4 个千兆输入端口线速报文转发情况下，EasySwitch 模型与 NetFPGA 参考路由器及 SwitchBlade 基本模型占用 FPGA 逻辑/存储资源情况. 其中，NetFPGA 及 SwitchBlade 相关数据来源于文献[3]. 由表可得，与其它模型相比，EasySwitch 模型通过优化设计占用 FPGA 硬件资源(尤其是存储资源)较少，可为用户提供极大的设计裕度，有效减少用户逻辑开发的限制.

表 4　EasySwitch 模块资源占用($N=4$)

资源名称	NetFPGA 参考路由器	SwitchBlade 基本模型	EasySwitch 模型
查找表 LUT	23 552	37 888	5 672
存储器位	2 644 992	3 698 688	180 042

5.2　应用实例

在 EasySwitch 处理模型下，网络报文数据平面处理功能的定制工作基于硬件逻辑编程语言(如 Verilog 等)对 UM 实现. 用户可以选择在 UM 中的自主实现报文定制处理功能，也可以选择通过修改、移植参考设计等方式加速逻辑设计工作. 下面介绍基于 EasySwitch 处理模型在 NetMagic 平台上实现的多种新型报文处理机制和网络协议.

5.2.1　RLI 延迟测量与评估

RLI(参考延迟插值)延迟测量体系[12]是普渡大学在 SIGCOMM 2010 上提出的一种流级精确延迟测量技术. RLI 延迟测量体系可为互联网中微秒级低延迟应用提供有效监测手段，支持网络故障快速定位和排除. RLI 通过在报文的发送端插入携带时间戳的参考报文，并在接收端计算参考报文的真实延迟，采用线性插值方法评估被测报文流的传输延迟. 我们基于 EasySwitch 模型在 NetMagic 平台上实现了 RLI 延迟测量与评估功能.

如图 7 所示，RLI 延迟测量与评估系统采用 Pass-Through 模式实现，由参考报文产生 UM 和延迟评估 UM 构成. 参考报文产生 UM 负责接收来自流量产生端(S)的待测报文，根据 RLI 算法完成参考报文自适应插入，将提取的待测报文及参考报文摘要信息(包括发送时间戳)封装后，通过 UMS-O 接口发送到 OS 调度输出. 其中，待测报文及参考报文的目标端口为 TX3(即待测设备输入端)，而摘要报文的目标端口则为 TX5(即性能监测端). 延迟估计 UM 负责接收自待测设备返回的待测报文和参考报文，并根据当前时间戳计算参考报文真实延迟，最后封装待测及参考报文摘要信息(包括发送时间戳等)并发送. 返回的待测报文通过 OS 发送到流量接收端(R)，摘要报文发送到性能监测端(M)，参考报文则被废弃.

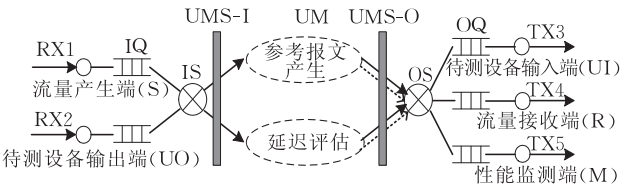

图 7　RLI 延迟测量与评估实现

RLI 延迟测量与评估系统仅使用 2 个输入端口，即 $N=2$. 根据式(2)，输入缓冲队列 IQ 容量设

① NMAC 协议实现 (NMAC Protocol Implementation)，NetMagic 研究组，http://www.netmagic.org/Data/Code/NMAC.rar.

为 12800 bits(队列长度为 100)即可保证报文队列不溢出;由式(5)可得,对于任意以太网报文,$Q(p)<1.568\ \mu s$,可满足 RLI 延迟测量体系精度($10\ \mu s\sim100\ \mu s$)要求.由于篇幅限制,详细内容参见文献[13].

5.2.2 其它应用

我们与香港理工大学合作基于 EasySwitch 模型在 NetMagic 平台上实现了 OneProbe 网络测量机制[14]的硬件加速,相比传统基于软件的实现方式具有更高的性能和精确性.此外,针对支持流媒体高效传输的新型传输层协议 Labelcast① 的实现,则进一步验证了 EasySwitch 模型对新型网络协议实现和验证的有效支持,关于 EasySwitch 模型更多应用实例可参见 NetMagic 论坛②.

5.2.3 特点与优势

EasySwitch 模型在 NetMagic 平台的成功应用表明,EasySwitch 模型具有结构简单、接口清晰以及资源占用少的实现特点,易于硬件实现.预置的 IS/OS 等调度器及接口逻辑,允许应用开发聚焦于用户逻辑 UM 实现,并可为多数据通路资源隔离提供支持.通过采用 UMS 接口将 UM 与预置逻辑解耦,可支持基于标准接口构建可重用硬件模块库.EasySwitch 模型资源、性能等特征参数可以采用精确的数学模型表述,从而为高精度网络实验的规划及实现提供理论依据,将实验系统误差控制在可接受范围内,适用于网络测量等对网络设备性能及精度要求较高的网络实验场景.

6 总 结

面向下一代互联网技术创新实验需求,针对现有 FPGA 实验平台报文处理模型缺失、设计开发门槛高、硬件资源有限等问题,本文提出了一种新型报文处理模型 EasySwitch.EasySwitch 的设计基于严谨的科学分析和丰富的设计经验,在考虑 FPGA 资源约束情况下,通过优化设计预置逻辑,为用户逻辑开发提供一个稳定、开放、通用的接口 UMS,允许用户专注于业务逻辑实现,减少设计开发工作量和周期.EasySwitch 模型支持通过集成多个 UM 实现并行数据平面及资源隔离,从而为虚拟化技术提供实现基础.通过提供精确的资源约束模型和性能分析模型,EasySwitch 可为网络实验系统设计规划提供科学指导和优化.目前,EasySwitch 模型已成功应用于多种新型网络处理机制及协议的开发和验证.

参 考 文 献

[1] Wu Jian-Ping, Wu Qian, Xu Ke. Research and exploration of next-generation internet architecture. Chinese Journal of Computers, 2008, 31(9): 1536-1548(in Chinese)
(吴建平, 吴茜, 徐恪. 下一代互联网体系结构基础研究及探索. 计算机学报, 2008, 31(9): 1536-1548)

[2] Naous J, Gibb G, Bolouki S, McKeown N. NetFPGA: Reusable router architecture for experimental research//Proceedings of the ACM Workshop on Programmable Routers for Extensible Services of Tomorrow (PRESTO'08). New York, USA, 2008: 1-7

[3] Anwer B, Tariq M, Motiwala M, Feamster N. SwitchBlade: A platform for rapid deployment of network protocols on programmable hardware//Proceedings of the ACM SIGCOMM 2010. New Delhi, India, 2010: 183-194

[4] Kohler E, Morris R, Chen B, Jannotti J, Kaashoek M F. The click modular router. ACM Transactions on Computer Systems, 2000, 18(3): 263-297

[5] Dobrescu M, Egi N, Argyraki K et al. RouteBricks: Exploiting parallelism to scale software routers//Proceedings of the 22nd ACM Symposium on Operating Systems Principles (SOSP'09). Big Sky, USA, 2009: 15-28

[6] Han S, Jang K, Park K, Moon S. PacketShader: A GPU-Accelerated software router//Proceedings of the ACM SIGCOMM 2010. New Delhi, India, 2010: 195-206

[7] Turner J, Crowley P, DeHart J et al. Supercharging PlanetLab: A high performance, multi-application, overlay network platform//Proceedings of the ACM SIGCOMM 2007. Kyoto, Japan, 2007: 85-96

[8] Luo Yan, Murray Eric, Ficarra Timothy L. Accelerated virtual switching with programmable NICs for scalable data center networking//Proceedings of the 2nd ACM SIGCOMM Workshop on Virtualized Infrastructure Systems and Architectures (VISA'10). New Delhi, India, 2010: 65-72

[9] McKeown N, Anderson T, Balakrishnan H, Parulkar G, Peterson L, Rexford J, Shenker S, Turner J. OpenFlow: Enabling innovation in campus networks. ACM SIGCOMM Computer Communication Review, 2008, 38(2): 69-74

[10] Carli L D, Pan Y, Kumar A, Estan C, Sankaralingam K. Flexible lookup modules for rapid deployment of new protocols in high-speed routers//Proceedings of the ACM SIGCOMM. Barcelona, Spain, 2009: 207-218

[11] Lu G et al. ServerSwitch: A programmable and high performance platform for data center networks//Proceedings of the 8th USENIX Symposium on Networked Systems Design and Implementation (NSDI'11). Boston, USA, 2011: 1-14

① Labelcast Protocol. Internet Engineering Task Force Draft. http://tools.ietf.org/html/draft-sunzhigang-sam-labelcast-02

② NetMagic 官方论坛. http://bbs.netmagic.org

[12] Lee Myungjin, Duffield Nick, Kompella Ramana Rao. Not all microseconds are equal: Fine-grained per-flow measurements with reference latency interpolation//Proceedings of the ACM SIGCOMM 2010. New Delhi, India, 2010: 27:38

[13] Li Tao, Sun Zhigang, Jia Chunbo, Su Qi, Lee Myungjin. Using NetMagic to observe fine-grained per-flow latency measurements//Proceedings of the ACM SIGCOMM 2011. Toronto, Canada, 2011: 466-467

[14] Luo Xiapu, Chan Edmond W W, Chang Rocky K C. Design and implementation of TCP data probes for reliable and metric-rich network path monitoring//Proceedings of the 2009 Conference on USENIX Annual Technical Conference. San Diego, USA, 2009: 1-14

附录

定理 1. 连续工作模式的输入调度器，当接口数目 $N<(f\times B_d\times B+B\times V_i-f\times B_d\times L)/(B\times V_i)$ 时，报文在输入缓冲区中不溢出.

证明. 采用反证法. 假设输入缓冲中的报文在 T_1 时刻有溢出，在此前调度器最后一次由空闲状态进入工作状态的时刻为 T_0，即在区间$[T_0,T_1]$中调度器持续调度报文输出. 又假设设在 T_1 时刻接收缓冲区 i 溢出，在区间$[T_0,T_1]$中调度器调度缓冲区 i 报文输出的时间为 t_1，调度其它$(N-1)$个缓冲区报文输出的时间为 t_2，则有 $T_0+T_1=t_1+t_2$.

由于在 T_0 之前调度器处于空闲状态，因此每个缓冲区中都没有完整的报文待调度，即每个缓冲区的数据容量均小于 L. 对于缓冲区 i，在 T_1 时刻溢出，可得

$$B<I_i(T_0)+(t_1+t_2)\times V_i-V_o\times t_1<L+(t_1+t_2)\times V_i-V_o\times t_1 \tag{6}$$

即 T_1 溢出，表明在$[T_0,T_1]$区间内进入缓冲区的数据加上 T_0 时缓冲区原有的数据去除调度离开的数据大于缓冲区容量 B.

对于其余 $N-1$ 个缓冲区，有

$$t_2\times f\times B_d<(t_1+t_2)\times V_i\times(N-1)+\sum_1^{N-1}I_i(T_0)<(N-1)(t_1+t_2)\times V_i+(N-1)\times L \tag{7}$$

即在$[T_0,T_1]$区间，$N-1$ 个缓冲区中原有的数据量加上到达的数据量大于调度输出的数据量.

由式(6)、(7)相加可得

$$t_1+t_2<(N\times L-B)/[V_o-N\times V_i] \tag{8}$$

由式(6)得

$$t_1+t_2>(B-L)/V_i \tag{9}$$

由式(8)、(9)得

$$(B-L)/V_i<(N\times L-B)/[V_o-N\times V_i] \tag{10}$$

即 $N>(f\times B_d\times B+B\times V_i-f\times B_d\times L)/(B\times V_i)$，与假设矛盾，定理得证. 证毕.

引理 1. 在任何时刻 t，N 个输入队列积累的数据和 $TD(t)$ 小于 $N\times L$.

证明. 在任何$[T_0,T_1]$连续调度周期中，输入调度的速率 $S\times N\times V_i$ 不小于 N 个端口输入速率之和，因此在时间区间$[T_0,T_1]$内，$TD(T_0)$最大. $TD(T_0)$不可能超过 $N\times L$. 否则在 T_0 之间的很短的某个时刻，必有某个输入队列长度大于 L，即至少积累一个完成报文，这与 T_0 时刻才开始调度是矛盾的. 证毕.

定理 2. 在连续工作模式公平调度算法下，长度为$l(p)$的报文在输入缓冲队列中最大排队调度延时为 $Q(p)<(N\times L-l(p))/((f\times B_d-(N-1)V_i)$.

证明. 任何调度周期/非调度周期内，$N-1$ 个缓冲队列中原有数据加上到达数据的量大于调度输出的数据量. 因此，在$[T_0,T_1]$连续调度中，报文 p 在 t_1 时刻完全进入缓冲区，其尾进头出延时 $Q(p)$满足

$$Q(p)\times f\times B_d<TD(t_1)-l(p)+(N-1)\times V_i\times Q(p)<N\times L-l(p)+(N-1)\times V_i\times Q(p),$$

因此可得 $Q(p)<(N\times L-l(p))/(f\times B_d-(N-1)\times V_i)$，定理得证. 证毕.

LI Tao, born in 1983, Ph. D., lecturer. His research interests include network processor and routing and switching.

SUN Zhi-Gang, born in 1973, Ph. D., professor. His research interests include network communication, routing and switching.

CHEN Yi-Jiao, born in 1972, Ph. D., associate professor. His research interests include network communication and reconfigurable routers.

JIA Chun-Bo, born in 1987, M. S. candidate. His research interests include network measurement and network switching.

SU Qi, born in 1985, M. S. candidate. His research interests include network protocol and network measurement.

GUO Teng-Fei, born in 1984, M. S. candidate. His research interests include network protocol and network switching.

Background

Network experimental platform is one of the key components of the next-generation Internet technology research, such as software-defined network (SDN), network virtualization, primitive extension, etc. The FPGA-based network experimental platform can achieve high flexibility and performance. Up to now, lots of researches have focus on the FPGA-based network experimental platform. However, there are two main problems of existing platforms. The first is the lack of accurate and optimized packet processing model to support and simplify the development for the users. Secondly, the scarce hardware resources restrict the users to develop and deploy their customized packet processing logic.

The main contributions of this work are as follows: (1) A novel packet processing model is proposed for the FPGA-based network experimental platform. (2) With the preplaced packet switching logic and well-defined UMS interface, the customized packet processing logic and the common packet processing logic can be efficiently separated. The users do not need to handle the common packet processing or interface logics, such as input/output packet interface, packet scheduler, etc. (3) Based on the model, theoretical analysis can be utilized for optimizing hardware resources configuration and packet processing performance, which provides efficient directions for the development and implementation of the network experimental systems.

The EasySwitch model has been implemented on the NetMagic platform, which is an open reconfigurable network experimental platform designed by the National University of Defense Technology. Several prototype systems have been successfully built utilizing the platform with the EasySwitch model. It is proved that the model can provide rapid development and deployment and the required high packet processing performance for the new network technologies. Moreover, compared with the existing models, the EasySwitch requires less hardware resources for implementing the preplaced processing logic, which brings more design space to the users. It is worth mentioning that all gatewares and documents of the hardware modules developed under the EasySwitch model on the NetMagic platform are open to the network community to encourage the hardware reuse.

The research is partially supported by the National Basic Research Program (973 Program) of China under grant No. 2009CB320503. Especially, the research is mainly focus on building an efficient experimental platform for implementing and verifying the new network protocols and packet processing mechanisms proposed in the program.