

基于 NetFpga 的数据包发生器的实现

姚 铭 应福军

(厦门大学信息科学与技术学院 福建 厦门 361005)

摘 要 数据包发生器广泛地应用于各种网络环境中 , 对于交换机、路由器以及网络本身的测试与分析有着重要的作用。针对大多数基于软件的数据包发生器不能达到千兆线速度发包的问题 , 详细介绍一种基于 NetFpga 平台的硬件数据包发生器 , 实现 4 端口的千兆线速度发包 , 并可通过软件配置每个端口的数据包发送速度、数目及延时等。

关键词 NetFpga 包发生器 模块化

中图分类号 TP311 文献标识码 A DOI: 10. 3969/j. issn. 1000-386x. 2013. 01. 061

NETFPGA-BASED DATA PACKET GENERATOR IMPLEMENTATION

Yao Ming Ying Fujun

(School of Information Science and Engineering, Xiamen University, Xiamen 361005, Fujian, China)

Abstract Data packet generator is widely used in various network environments. It plays an important role in testing and analysing the performance of switches, routers and networks. To address the problem that most software-based data packet generators cannot send the packet at gigabit speed, in this article we describe in detail a hardware data packet generator based on NetFpga platform. It realises four-port package sending at gigabit speed, and can also configure the sending speed, number and delay of the package at each port.

Keywords NetFpga Packet generator Modularisation

0 引 言

包发生器^[1]是一种可以产生特定类型、特定长度、特定速率的数据分组的工具,主要功能就是对真实的网络环境的模拟,通过数据包的构造,生成数据流,达到测试网络设备或者网络应用的目的。包发生器在交换机、路由器等网络产品的测试和网络新协议的开发中,有着举足轻重的作用。测试者只需为被测对象搭建一个简单的测试环境就可以进行测试了,而不必将其放入实际运行的网络中去,同时降低测试的风险和成本。

包发生器必须是一个实时的系统,以往通过编写软件实现的包发生器无法达到实时高速线速发包的要求,因为 CPU 执行指令是串行执行的,执行完一条才能执行下一条,在处理突发事件时只能调用有限的中断资源。而 NetFpga 是现场可编程逻辑器件,其并行执行,硬件实现的方式可以应对设计中大量的高速电子线路设计的需求。

目前,市场上也可以购买到一些公司开发的产品,如 Ixia 公司等,但价格都非常昂贵。为此,本文介绍如何在 NetFpga 平台上构建一个速率可达 1Gbps 的包发生器。

1 NetFpga 硬件开发平台

NetFpga 平台^[2,3](结构如图 1 所示)是由美国斯坦福大学开发的低成本、可重用的开发平台,它主要作为教授网络硬件和路由器设计的工具。由于它能够很好地支持模块化设计,复杂

的硬件设计可以通过各个模块的组合来完成;而且,NetFpga 是一个基于 Linux 的开放性平台,所有对它感兴趣的人都可以利用平台上现有的资源,在前人开发的基础上搭建自己的系统,而自己开发的系统也可以被其他人所用,而不再需要重复地搭建外围模块、开发驱动和 GUI 等,只要添加自己的模块和修改现有的系统即可,因此 NetFpga 已经吸引了越来越多的开发者。

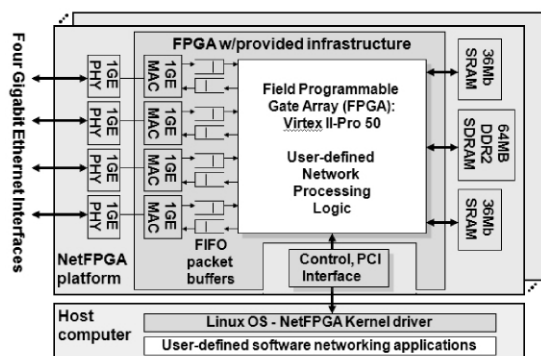


图 1 NetFpga 平台

平台包含 2 个 FPGA(1 个核心时钟频率为 125 MHz, 基于用户自定义的逻辑编程的 Virtex-II Pro50 FPGA; 1 个用于实现连接主机 CPU 的 PCI 接口控制逻辑的 Xilinx Spartan 的小型 FP-GA); 4 个作为 FPGA 软核的千兆以太网 MAC 控制器; 内存由 2 个与 FPGA 核心逻辑同步运行于 125 MHz 时钟频率、字长 36 比

收稿日期: 2012-04-23。姚铭, 高工, 主研领域: 嵌入式系统, 计算机网络安全。应福军, 硕士生。

特的 SRAM 和 2 片与 FPGA 异步运行运行于 200 MHz 时钟频率、字长 32 比特的 DRAM 组成。

2 包发生器的硬件实现

NetFpga 开发平台的优势在于它的开放性和模块化设计理念,在 NetFpga 基础开发包中已经包含了多个优秀的参考设计,而且其代码是完全开源的,开发人员通过自行设计或者调用修改已经存在的硬件模块,通过综合形成新的数据处理通路,可以很方便地实现千兆级别的高性能网络系统。本文实现的包发生器就是在参考路由设计^[4]基础上的二次开发。

2.1 数据包发生器的原理

包发生器采用的是数据包的回放机制,从功能上看,包发生器实现了千兆速率线速发包和网络流量的实时捕获。

在产生 Packet 时,按照标准 PCAP 文件(PCAP 文件格式是常用的数据报存储格式,包括 Wireshark 在内的主流抓包软件都可以生成这种格式的数据包^[5])产生相应的 Packet 并将其存储在 SRAM 中,应用软件可以精确配置 Packet 的发送速率、Packet 间的延时及数目等,再从相应的以太网口发送出去。由于 SRAM 容量有限,PCAP 文件的大小受到了限制,因此采用了循环迭代的方法满足高速发包要求。

在捕获 Packet 时,添加 Packet 的时间戳(数据包到达的时间),以 PCAP 格式存储捕获的 Packet,通过 DMA 方式经 PCI 总线将捕获到的 Packet 传送给主机。

2.2 流水线的设计方案

所谓流水线设计^[6,7],如同生产装配线一样,将操作执行工作量分成若干个时间上均衡的操作段,从流水线的起点连续的输入,流水线的各操作阶段以并行的方式执行,使得操作执行速度只与流水线输入的速度有关,而与处理所需的时间无关,因此在理想的流水线操作状态下其运行效率很高。

在包发生器中,其结构如图 2 所示,数据包的处理流程被分成 9 个模块,每个 Packet 处理模块都包含一个同步 FIFO,用来实现 Packet 的缓存增加 Packet 在每个模块的停留时间^[8],整个数据处理是单流向的,即没有反馈运算,前一个模块的输出作为后一个模块的输入,可以达到很高的数据吞吐量;同时也使每个模块只实现简单的功能,非常有利于编程的实现,使复杂的数据处理问题变得非常简单。

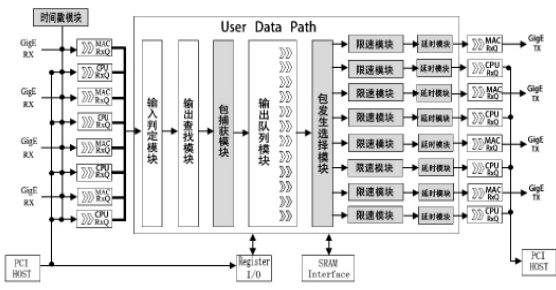


图 2 参考流水线设计

2.3 功能模块的实现

包发生器是以参考路由设计为基础的,对比发现,包发生器只是在参考路由上添加和修改了几个模块而已(图中的灰色部分),新的模块几乎全在 User_Data_Path 内,只有 Time Tamp 添加在 nf2_mac_grp 里面。下面我们对这些模块逐一介绍:

(1) 时间戳模块(timestamp)

时间戳模块是为了在捕获 Packet 时,当 Packet 被接收到 MAC 队列时,记录此时的时间值,并将该值提供给 rx_queue,并以 PCAP 格式存储捕获的 Packet,这种方式标记的时间非常精确。

(2) 输入判定模块(Input_Arbitrer)和输出查找模块(Output_Port_Lookup)

这两个模块是未经修改直接从基础开发包中调用的。Input_Arbitrer 依然采用参考路由设计中的电路,采用轮询的方式访问 8 个输入缓冲队列,当有一个队列非空,则从相应的 FiFo 中读取 Packet,直到从一个队列读完才转到下一个队列;Output_Port_Lookup 则没有采用参考路由设计中的复杂电路,而是采用了参考网卡设计中的简单实现,其功能包含两层意思:将来自 CPU 队列中的 Packet 转发到 MAC 端口;将来自 MAC 端口的 Packet 转发到相应的 CPU 队列。

(3) 包捕捉模块(Packet Capture)

包捕捉模块有选择地捕获来自 4 个 MAC 端口的 Packet,统计捕获的包数目和时间。一个单独的使能信号可以开启和关闭该模块,一旦该模块关闭时,时间戳模块也跟着被剥离,此时的 NetFpga 只能作为一个普通的四端口网卡使用。

(4) 输出队列模块(Output Queues)

输出队列模块在原有的基础上作了小的修改,回顾参考路由中 output_queues 的几个模块,单纯从接口信号看,remove_pkt_parser 和 store_pkt 保持不变,remove_pkt 在原来 8 个队列的基础上再添加 4 个队列,这 4 个队列用于从 SRAM 中读取 PCAP 格式数据,即要发送的数据包。12 个队列的大小由软件配置的寄存器来决定,这样一来队列的大小就可以与 PCAP 文件包含的数据多少相匹配,可以最大化地利用接受队列的资源。

(5) 包发生选择模块(Packet Generator Output Select)

Packet Generator Output Select 决定 4 个 MAC 输出端口从 output_queue 的 MAC 队列还是从 PCAP 数据队列读取 Packet,前者用于正常的网络数据传输,后者用于从 SRAM 中读取 PCAP 数据,即产生 Packet 的过程。

(6) 限速模块(Rate Limiter)

由于数据包发生器是作为标准的网络测试设备,因此对于自身发送速度的控制有着极高的要求,该模块根据 NetFpga 每个时钟节拍能传输 8bit 数据,通过设定连续传输的字节数目及相邻数据流的时钟节拍间隔(token_interval)达到控制速度的目的。例如,设定发送速度为 200Mbps,传输 1 个时钟节拍的数据,间隔 1 个时钟节拍;设定的速度为 600Mbps,连续传输 3 个时钟节拍的数据后,间隔 2 个时钟节拍。

(7) 延时模块(Delay Module)

为了能够对每个数据包的延时时间进行控制,我们给每个输出端口添加一个延时模块。若使能延时模块,则每个数据包按照设定的延时时间发送,最小为 0;反之,则根据每个数据包原有的延时时间发送。

3 软件功能的实现

在 NetFpga 的参考设计中,每个模块执行一定的功能,但为了提供一种方式对每个模块进行配置的功能,便出现了寄存器系统,其功能是提供一个模块与外界接口,以供外界对该模块执行相应的配置,以及获取该模块的基本信息的功能。用户软

件读/写 module 内的寄存器与 I/O 寄存器的读/写类似^[9]。

在包发生器中,软件部分使用非常简洁的 PERL 语言编写。主要完成以下功能:根据调用的 PCAP 文件产生标准数据包,再由用户设定的参数,计算相应的寄存器数值,通过调用 nf_read() 函数读寄存器和 nf_regwrite() 函数写寄存器,最终完成对硬件模块的使能和控制。包发生器的寄存器如表 1 所示。

表 1 寄存器列表

寄存器	功能描述
PKT_GEN_CTRL_ENABLE	该信号有效启动 pkt_captur 和发包操作
PKT_GEN_CTRL_PKT_CNT_x	记录捕获数据包数目
PKT_GEN_CTRL_BYTE_CNT_HI_x	记录捕获的字节数高位
PKT_GEN_CTRL_BYTE_CNT_LO_x	记录捕获的字节数低位
PKT_GEN_CTRL_TIME_FIRST_HI_x	记录捕获的第一个数据包到达的时间高位
PKT_GEN_CTRL_TIME_FIRST_LO_x	记录捕获的第一个数据包到达的时间低位
PKT_GEN_CTRL_TIME_LAST_HI_x	记录最后一个捕获数据包到达的时间高位
PKT_GEN_CTRL_TIME_LAST_LO_x	记录最后一个数据包到达的时间低位
RATE_LIMIT_ENABLE_BIT_NUM	使能限速模块
RATE_LIMIT_INCLUDE_OVERHEAD_BIT_NUM	使能过载控制
DELAY_RESET	延迟模块复位信号

4 包发生器的功能验证

为了对包发生器进行功能验证我们将包发生器通过一台千兆交换机的 g0/6 端口接入校园网,在网路的另一端通过一台主机对数据包进行捕获,如图 3 所示,其中 PC1 为一台装有 NetFpga 的高性能主机,PC2 为一台装有抓包软件的普通主机。

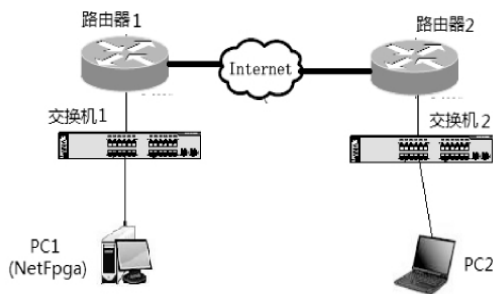


图 3 测试网络拓扑图

为了方便测试,我们没有编写自定义数据包,而是采用了从网络中捕获的数据包,修改数据包的源/目的 IP 地址、源/目的 MAC 地址、校验值等信息,再以 10Mbps 的速度发送 10 000 个数据包,在 PC2 查看是否捕获到了来自 PC1 的数据包。实验结果显示,PC2 捕获到了 10 000 个来自 PC1 的数据包,无丢失,这说明了包发生器的数据包功能正常。

由于普通 PC 机无法通过软件达到千兆包捕获的能力,为了测试包发生器的性能,在假设交换机速度准确的情况下,通过连接交换机 1 的 Console 端口,利用超级终端的 show interface g0/6 命令查看交换机的端口速度,通过对比得出包发生器的发包性能。表 2 是包发生器以不同速度发包在交换机上查看到的接收速度。

表 2 发包速度不同下的接收速度

包发生器的设定速度(Mbps)	交换机收包速度(bit/sec)
100	100456451
500	500365780
900	905896431
950	950620186
980	980210774
990	984602867
1000	988307966

从上表分析可知,在 980Mbps 速度以内,包发生器的发包速度和交换机测得的速度是非常接近的,我们可以认为包发生器具备 980Mbps 以内准确发包的能力;在趋近与 1000Mbps 发包的时候误差较大,但是这还不足以说明包发生器不具备千兆发包的能力,因为千兆速度已经达到交换机的极限速度,所以我们必须借助更高速的设备才能得出准确结论,却由于实验条件限制没有做进一步的测试。

5 结语

随着高速互联网的不断发展,在网络设备测试和网络安全领域,高速的数据包发生器已经越来越受到人们的亲睐。例如在上一节的网络拓扑中,通过查看交换机 2 的运行状态,可以得出在不同传输速度下的丢包率、传输延时等。又由于数据包的可重构性,通过发送各种攻击数据包,可用于测试网络防攻击的能力。

本文只是对 NetFpga 平台应用开发的一种尝试,重点在于理解和掌握基于参考路由设计的模块化设计方法,为今后的深入研究打下基础。实现的以太网包发生器已经达到了设计的要求,可完成四网口的千兆级别线速发包,可以根据不同的应用,通过软件调整包发生器的相关参数,就可以满足不同的应用要求,如配置各网口的发包速度、延时等,从而为下步的网络安全研究打下基础。

参 考 文 献

[1] 李必颖,胡越明,白英彩. 基于 INTEL 网络处理器的流量发生器[J]. 微型电脑应用, 2005, 21(3): 22-25.
 [2] <http://www.netfpga.org>.
 [3] Lockwood J W, McKeown P, Watson P. NetFPGA—An Open Platform for Gigabit-Rate Network Switching and Routing [C]//Proceedings of the 2007 IEEE International Conference on Microelectronic Systems Education.
 [4] 洪毅清,秦雅娟,周华春. 基于 NetFPGA 的模块化硬件路由器实现[J]. 计算机应用与软件, 2011, 28(8): 28-30.
 [5] <http://www.wireshark.org/>.

(下转第 280 页)

以将历史数据库分为 6 个子库,基于这 6 个字库就可以一一对应地训练出 6 个支路模型。

可根据图 3 分类树的叶子结点类型和不同属性下的所有属性值,构建一张权值表并初始化。具体参见表 2,初始化时让对应的支路模型权值系数相等。

表 2 权值表的初始化

模型类型	天气		工作日?		上下班高峰?	
	好	恶劣	是	否	是	否
结点 1	1/3		1/3		1/3	
结点 2	1/3		1/3			1/3
结点 3	1/2			1/2		
结点 4		1/3	1/3		1/3	
结点 5		1/3	1/3			1/3
结点 6		1/2		1/2		

由表 2 可知,动态权值表的更新就是实时地改变这些叶子节点对应的权值系数。

数据归一化处理,利用 2011-07-01 到 2011-07-10 这 10 天为训练样本,2011-07-11(周一、晴天),2011-07-12(周二、大雨)作为预测样本。AOSVR 算法核函数选取经典的 RBF 核函数,选取参数值: $\sigma = 0.5, C = 1, \epsilon = 0.05$ 。传统的 AOSVR、属性值为天气-好对应的支路预测模型 AOSVR(w_{good})和 WT-AOSVR 预测模型的预测性能比较结果如表 3 所示。

表 3 各模型性能分析表

预测时间	算法	评价指标	
		MAPE	RMSE
2011-07-11	AOSVR	7.61%	9.75%
	AOSVR(w_{good})	6.82%	9.07%
	WT-AOSVR	6.34%	8.29%
2008-07-08	AOSVR	7.73%	9.84%
	AOSVR(w_{good})	7.87%	9.91%
	WT-AOSVR	6.28%	8.21%

从表 3 中可以看出,三个模型的 MAPE 评价指标均小于 10%,能够满足交通预测的需求。在晴天的情况下,AOSVR(w_{good})和 WT-AOSVR 性能明显优于传统的 AOSVR;在雨天的情况下,AOSVR(w_{good})性能最差,AOSVR 次之;无论天气状况如何,WT-AOSVR 预测模型的平均绝对相对误差最低,而且均方百分比误差明显低于前两者。这说明 WT-AOSVR 预测模型的方法要比传统的 AOSVR 的方法具有更好的预测表现,而且有较好的稳定性。

利用 2011-07-01 到 2011-07-10 这 10 天为训练样本,来预测 2011-07-11 的流量信息,用 Matlab 7.0 为仿真工具,预测的结果如图 4 所示。相对传统的 AOSVR 模型,WT-AOSVR 预测模型的方法具有较好的拟合度,表现出较好的预测精度。

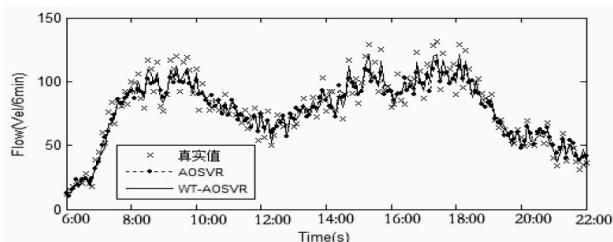


图 4 交通流量预测结果

3 结 语

针对现阶段有关核函数选择缺乏科学的理论依据,为了提高预测模型的学习和推广能力,本文从数据挖掘的角度提出一种 WT-AOSVR 交通流预测模型,针对交通流的特性以及影响交通流的环境因素进行数据挖掘,分类处理。为了提高样本的针对性和准确性以提高模型的学习能力,按照分类树不同属性下的所有属性值对数据进行分类,分别训练出一个对应支路模型;为了保证模型的泛化能力和推广能力,构建具有自适应能力的动态权值表,然后基于权值表,选取不同属性下的支路预测模型组合生成最终的模型。通过仿真实验及结果对比分析,表明该方法能够很好地反映交通流量的趋势和规律,实现模型自适应能力,有效地提高了交通的预测精度。

参 考 文 献

- [1] 杨兆升,王媛,管青. 基于支持向量机的短时交通流量预测方法[J]. 吉林大学学报:工学版, 2006, 36(6): 881-884.
- [2] Si Fengqi, Carlose Romerob, Yao Zheng. A new approach for function approximation in boiler combustion optimization based on modified structural AOSVR [J]. Expert System with Applications, 2009, 36(4): 8691-8704.
- [3] Vladimir Vapnik. The nature of statistical learning theory [M]. New York: Springer, 1995.
- [4] Ma junshui, James Theiler, Simon Perkins. Accurate on-line support vector regression [J]. Neural Computation, 2003, 15(11): 2683-2703.
- [5] Manoel Castro-Neto, Jeong Young-Seon, Jeong Myong-Kee, et al. Online - SVR for short term traffic flow prediction under typical and a typical traffic conditions [J]. Expert Systems with Applications, 2008, 36(3): 6164-6173.
- [6] 刘大同, 彭宇, 彭喜元. 基于残差预测修正的局部在线时间序列预测方法 [J]. 电子学报, 2008, 36(12A): 81-85.
- [7] 吴微, 张凌. 自适应参数的 AOSVR 算法及其在股票预测中的应用 [J]. 大连理工大学学报, 2009, 49(4): 605-610.
- [8] 王凡. 基于支持向量机的交通流预测方法研究 [D]. 大连: 大连理工大学, 2010.
- [9] 张元亮, 卢鹏. 基于粗糙集的城市交通拥堵预警算法分析 [J]. 交通科技与经济, 2009, 11(2): 74-76.

(上接第 244 页)

- [6] 夏宇闻. Verilog 数字系统设计教程 [M]. 2 版. 北京: 北京航空航天大学出版社, 2008.
- [7] Richard E Haskell, Darrin M Hanna. FPGA 数字逻辑设计教程: Verilog [M]. 郑利浩, 王荃, 陈华锋, 译. 北京: 电子工业出版社, 2010.
- [8] 陆佳华. 零存整取 NetFPGA 开发指南 [M]. 北京: 北京航空航天大学出版社, 2009.
- [9] Alex. NetFPGA 寄存器 [EB/OL]. [2011-05-26]. <http://www.docin.com/p-106939010.html>.
- [10] 李太君, 林元乖, 张晋, 等. 计算机网络 [M]. 北京: 清华大学出版社, 2009.
- [11] 林金, 杨波. 基于 NetFPGA 的网络流量采集器 [J]. 济南大学学报: 自然科学版, 2011, 25(1): 7-10.