

引用格式:贾正义,陈晓,王劲林,等. 基于多核网络处理器的时间感知整形技术研究[J]. 微电子学与计算机, 2019, 36(12):1-5. [JIA Z Y, CHEN X, WANG J L, et al. Research and implementation of time-aware shaping mechanism based on multi-core network processor[J]. Microelectronics & Computer, 2019, 36(12):1-5.]

基于多核网络处理器的时间感知整形技术研究

贾正义^{1,2}, 陈晓^{1,2}, 王劲林^{1,2}, 叶晓舟^{1,2}

(1 中国科学院 声学研究所, 北京 100190; 2 中国科学院大学 电子电气与通信工程学院, 北京 100049)

摘 要: 时间感知整形(Time Aware Shaper, TAS)是时间敏感网络(Time-Sensitive Network, TSN)的关键机制, 对于 TAS 的研究与实现具有重要意义. 基于多核网络处理器设计了 TAS 机制功能模块结构和 TAS 引擎处理流程, 针对由于内核时钟与 PTP(Precision Time Protocol)时钟存在误差而导致的时隙长度偏差问题, 提出了调整保护时隙算法来校正时隙. 最后, 基于 Cavium 多核网络处理器实现了 TAS 机制, 实验结果表明: TAS 机制可以按照预配置时隙发包; 增加保护时隙调整, 可以有有效的校正时隙误差, 增加 TAS 机制的鲁棒性.

关键词: 时间敏感网络; TSN; 时间感知整形; 多核网络处理器

中图分类号: TP393

文献标识码: A

文章编号: 1000-7180(2019)12-0001-05

Research and implementation of Time-Aware Shaping mechanism based on multi-core network processor

JIA Zheng-yi^{1,2}, CHEN Xiao^{1,2}, WANG Jin-lin^{1,2}, YE Xiao-zhou^{1,2}

(1 Institute of Acoustics, Chinese Academy of Sciences, Beijing 100190, China;

2 School of Electronic, Electrical and Communication Engineering, University of Chinese Academy of Sciences, Beijing 100049, China)

Abstract: The Time-Sensitive Networks (TSN) provides important network technology support for “Made in China 2025” plan. The design and implementation of the Time Aware Shaper (TAS), the fundamental mechanism of TSN, is of great significance. In this paper, based on the multi-core network processor, we propose a method for quantifying the bandwidth into time slots and design the function module structure and the processing flow of the TAS mechanism. Moreover, for the problem of time slot length variation caused by the error between the CPU clock and the PTP (Precision Time Protocol) clock, we propose a protection band adjustment algorithm to correct the time slot of the port. Finally, we develop the TAS mechanism based on the Cavium multi-core network processor. The experimental results show that the TAS mechanism can send packets according to the pre-allocated time slots. Besides, enabling the protection band adjustment can effectively correct the time slot error and increase the robustness of the TAS mechanism.

Key words: Time-Sensitive Networks; Time Aware Shaper; multi-core network processor

1 引言

IEEE 时间敏感网络^[1] (Time-Sensitive Net-

working, TSN)能够为工业控制网络、5G 任务关键型网络切片提供底层传输技术支撑^[2]. TSN 工作组开放了 IEEE820.1Qbv 标准, 然而该标准只是提出

了时间感知整形 (Time Aware Shaper, TAS) 机制的框架结构, 如何将带宽量化为时隙, 如何结合硬件平台实现 TAS 机制, 标准中并未提及. 此外, 目前关于 TAS 机制的研究主要集中于时隙调度算法研究^[3,4], 很少有文章讨论上述问题.

多核网络处理器具有高并行处理能力, 已经成功应用于高速流处理及协议分析与控制^[5], 流量识别^[6]和流量控制^[7], 是 TAS 机制实现的理想平台. 基于多核网络处理器, 本文研究了 TAS 机制设计与实现方法.

2 时间感知整形技术

TAS 机制在交换节点的每个输出端口, 都有优先级为 0~7 的 8 个输出队列. 端口按周期从队列取包发送, 每个周期分为若干个时间片, 在每个时间片内只能从一个或多个选定的队列取包发送, 具体队列选择策略由开关控制列表 (Gate Control List) 决定. 队列选择控制列表由二元组 {时间片, 队列状态} 组成, 其中“队列状态”由 8 位 {0, 1} 二进制数标识, 对应 8 个优先级队列. “1”代表队列被选中, 即当前时间片内, 从该队列取包并发送; “0”代表队列关闭, 即当前时间片内, 该队列不允许发包^[8].

由上可知, TAS 机制的关键是时隙的分配和按时隙发包机制. 所以, 实现时间感知整形机制需要完成两方面工作: (1) 将带宽量化为时隙, 建立时隙资源池. (2) 基于多核网络处理器实现 TAS 按时隙发包机制.

3 带宽量化为时隙

针对由于带宽资源竞争产生的延迟抖动问题, TAS 机制通过将带宽量化为时隙, 按时隙进行数据转发, 实现业务带宽资源预留. 多核网络处理器是以包为单位处理数据流的, 带宽计量单位为 pps (packet per second). 对于多核网络处理器, 设备的吞吐并非只取决于端口带宽, 还取决于设备 CPU 处理能力. 因此, 我们将时隙分为端口时隙和内核时隙.

端口时隙 (Port Slot, PS): 每个输出端口时分复用, 端口按周期 T 发包, 每个周期 T 内部, 划分成 N 个时隙, 这些时隙称为 PS. 所有 TSN 设备的 PS 周期和数量都是相同的, 全网时钟同步后, 按照 PS 来进行时隙的调度和分配.

内核时隙 (Core Slot, CS): 每个 CPU 按周期 T

处理包, 每个周期 T 内部, 划分成 M 个 CS, 每个 CS 处理 m 个数据包 ($m \geq 1$, 整数). 所有端口时分复用 CPU, 共享 CPU 的 CS. CS 是可调整的, 与设备相关的. 若 CS_{len} 为 CS 长度, 那么设备带宽粒度 $b\omega = m/CS_{len}$ (pps). 由于每个流只有在分配到的时隙内才可以被处理, 所以如果为优先业务流分配 n 个时隙, 则为优先业务流预留的带宽为 $BW = n * b\omega$ (pps).

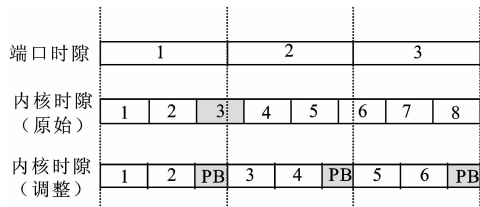


图 1 CS 与 PS 映射方法示意图

在交换机启动时, 交换机可以得到 CS 长度 CS_{len} 和 PS 长度 PS_{len} 信息, 并且满足条件 ($CS_{len} \leq PS_{len}$). 如图 1 所示, 一个 PS 长度可能等于多个 CS. 如果 CS_{len} 不能被 PS_{len} 整除, 那么在一个 PS 内, 经过若干个 CS 后, 剩余长度会不足一个 CS, 此时, 我们需要停止处理数据包, 否则会侵占下一个 PS. 这里我们将 PS 剩余的不足一个 CS 的部分作为保护时隙 (Protect Band, PB), 防止在时隙快要结束时开始处理一个新的数据包, 从而侵占了下一个 PS 的时间. PB 还有另外一个重要作用, 我们在 TAS 引擎设计里会详细介绍.

在实际工程实现中, 在单个核心上, 每隔 CS_{perps} 个 CS, 我们进入保护时隙 PB, 在 PB 内我们不做任何数据处理, 只是等待 PB 结束. 这里 $CS_{perps} = PS_{len}/CS_{len}$, $PB_{len} = PS_{len} \% CS_{len}$; 所有核的 CS 总数 $TM = CS_{perps} * N * C_{num}$ 构成时隙资源池, 其中 N 为一个周期内的 PS 数量, C_{num} 为内核数量.

在带宽量化为时隙后, 我们可以进行时隙的分配. 由于多个端口是可以并行发包的, 所以每个端口的时隙理论上是独立的, 但是内核在一个 PS 内能并行处理的包数量 (CS) 是有限的, 所以端口实际的可用时隙是受到 CS 的限制的. 所以时隙的分配方式是: 首先根据转发路径确定转发端口; 申请该端口的时隙; 端口再向时隙资源池申请 CS.

4 TAS 机制设计与实现

4.1 Cavium 多核网络处理器

Cavium OCTEON II 处理器支持两种软件执

行环境:(1)SE (Simple Executive)环境,是一种实时操作系统. SE 环境提供了一套硬件的 API. 应用可以直接管理底层的处理核心、内存、协处理器等硬件资源. 在实时操作系统下,没有系统调用、进程调度的开销,也没有内核空间与用户空间区分带来的内存复制开销. 利用 OCTEON 消息机制,应用可以轮询获取报文,进行实时处理. SE 环境适合对性能和延迟要求高的应用,适合用于 TAS 机制的实现.

(2)SMP (Symmetric Multi-Processing) Linux 环境,可以运行于一个或多个处理核心上. 借助 Linux 通用和可移植的运行环境,用户可以非常灵活地搭建和部署各类应用. 相对 SE 环境而言, Linux 环境的报文处理性能较低,适用于对灵活性和可移植性要求较高的应用,因此 Linux 环境可用于时隙的配置和管理.

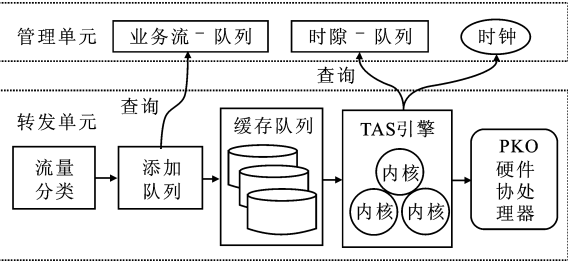


图2 基于多核网络处理器的 TAS 机制功能模块图

4.2 架构设计

一个典型的逻辑交换机主要包含管理单元与转发单元,如图 2 所示,我们在图中只显示了与 TAS 相关的部分.

管理单元主要负责更新和维护 TAS 配置信息,包括业务流与端口队列的映射,以及 TAS 引擎转发时隙与端口队列的映射关系.

时隙-队列映射. 我们在每个端口都为时间敏感流预留相应队列,即时间敏感业务流与队列实现一对一映射. 同时,我们在为业务分配 CS,将申请到的 CS 与该队列绑定.

转发单元主要负责 TSN 业务流的识别与添加队列. 在识别数据流类型后,根据预先配置的业务流和队列映射信息,将数据流缓存到相应队列.

数据流的转发是由 TAS 引擎完成. 根据控制器下发的时隙与队列绑定关系, TAS 引擎根据传输控制列表来按时隙从相应队列中取包发送. 如图 2 所示, TAS 引擎由时钟驱动,根据时间信息来查询相应的队列,然后从相应端口转发,具体 TAS 引擎实现我们在后边详细讨论.

时钟为全局同步时钟. 通过运行 PTP 精准时钟同步协议,实现与其他设备的时钟同步.

PKO 硬件协处理器即 Packet-Output Processing 单元,维护发送队列,提供硬件卸载 TCP/UDP 校验和功能. PKO 支持多个优先级队列,静态优先级队列中的报文总是会优先处理;若静态优先级队列为空,则采用加权公平轮询算法 (Weighted Fair Queuing, WFQ) 访问剩余队列.

4.3 TAS 引擎

TAS 引擎算法流程主要包含以下步骤:

读取 PTP 时钟. TAS 机制的前提条件为全网设备时钟同步. 多核网络处理器的 Precision Time Protocol 单元,提供 96 比特的 PTP 专用时钟. 以 PTP 时钟作为参考时间,进行按时隙调度.

计算当前所处 CS,根据 CS 查询队列. 控制器预先配置开关控制列表 (GCL),即 CS 与队列的映射关系,内核根据当前时隙可以查询到相应队列,取包发送. 如果队列为空,则从默认队列 (普通数据流) 中取包发送.

时隙更新. 每次按时隙取包发送后,如果都重头开始,读取 PTP 时钟并计算 CS,会引入较大的计算开销,占用 CPU 资源,从而产生带宽浪费. 另一方面, PTP 时钟同步调整间隔周期是 1 秒,所以在 1 秒内连续访问 PTP 时钟也并不能增加时间同步精确度. 所以,为了节省 CPU 资源,在一定周期内,我们考虑采用 CS 自计数法.

4.4 CS 自计数法

每个 CS 处理一个数据包,在取包发送后, CS 更新: $CS_{next} = (CS_{last} + 1) \% M$, M 为单核 CS 总数, CS_{last} 为更新前的 CS;判断 $CS_{next} \% CS_{perps}$ 是否为 0, 如果为 0 则进入保护时隙 PB,等待 PB 结束, CS 再次更新.

采用 CS 自计数法更新时隙时,要求 CPU 时钟与 PTP 时钟频率是同步的. 中央处理器 (CPU) 执行指令所必须有的时钟频率信号是由晶振产生的. CPU 一切指令的执行都是建立在这个基础上的. 而内核时钟与 PTP 时钟是存在误差的. 误差产生原因主要包括两方面:一方面,随着 PTP 时钟频率和相位的不断校准, CS 长度会开始与 PTP 时钟产生偏差;另一方面,每个晶振输出信号的频率不可避免会有一定的偏差,所以 CPU 时钟频率自身也存在误差. 这种误差会导致不同设备的 TAS 周期长度不同, PS 产生偏差,从而破坏 TAS 整形机制,同时全网时钟同步也失去意义.

为了与 PTP 时钟保持一致,确保 PS 长度能够稳定不出现偏差累积,我们通过两次校准来解决该问题.第一次校准为 PS 校准,每隔一定时间我们读取一次 PTP 时钟,重新计算当前所处 PS 和 CS;第二次校准是通过调整保护时隙来进行相应的 PS 长度校准,校准方法如图 3 所示.

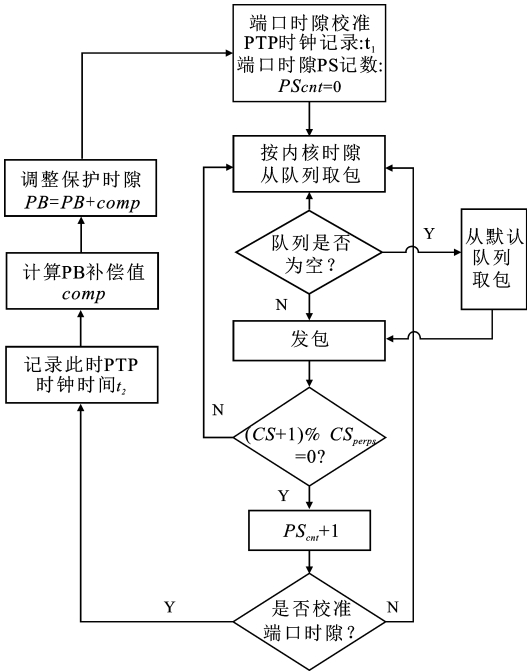


图 3 保护时隙调整算法流程图

- (1)PTP 时钟每隔一定时间校准一次当前所处得到 PS 并计算此时的 CS,同时记录此时 PTP 时钟时间 t_1 和 PS 计数器 PS_{cnt} .
- (2)根据校准后的 CS 查找队列发包,判断 $(CS_{last} + 1) \% CS_{perps}$ 是否为 0,如果为 0 则进入保护时隙 PB,等待 PB 结束,则一个 PS 结束,PS 计数器 PS_{cnt} 加 1.
- (3)判断是否需要校准 PS.在校准 PS 前,先计算 PB 的补偿值.通过调整 PB,可以补偿由于内核频率与 PTP 时钟频率偏差造成的 PS 长度误差.补偿值 $comp = PS_{len} - (t_2 - t_1) / PS_{cnt}$.调整后新的 $PB: PB_{new} = PB_{last} + comp$.

5 实验评估

5.1 TAS 整形测试

基于多核网络处理器的 TSN 交换机系统参数设置如表 1 所示,我们将端口传输周期划分为 100 个时隙,每个时隙长度为 $2\ \mu s$;我们的系统内核处理一个包的时间约为 $1.5\ \mu s$,所以 CS 为 $1.5\ \mu s$,保护时隙初始为 $500\ ns$.

表 1 多核网络处理器系统参数描述

系统参数	描述	取值
N	PS 数量	100 (个)
PS_{len}	PS 长度	2 (μs)
CS_{len}	CS 长度	1.5 (μs)
M	单核时隙数量	100 (个)
PB_{len}	PB 长度	0.5 (μs)

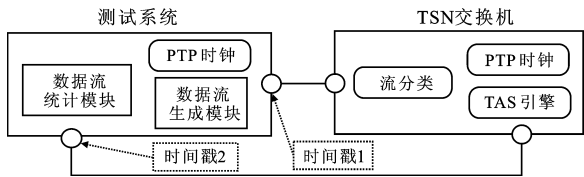


图 4 TAS 机制测试环境示意图

TAS 测试环境如图 4 所示,主要包括测试系统和 TSN 交换机.测试系统负责生成数据流并记录数据流发送时间(时间戳 1)和接收时间(时间戳 2),时间戳均是源于 PTP 时钟. TSN 交换机主要包括流分类模块,PTP 时钟同步模块和 TAS 引擎机制.数据流 ID,时隙和队列配置如表 2 所示,TSN 交换机的系统参数设置如表所示.测试系统按照数据流 ID 顺序,轮询发包,每轮每条流发一个包,发包速率为 10 Kpps. TAS 引擎按照表 2 所配置时隙发包.

统计结果如图 5 所示,我们选取一个 TAS 周期($200\ \mu s$)的数据流统计结果进行分析,横坐标为 PTP 时间戳记录的时间,纵坐标为数据流 ID,不同形状的点代表不同数据流的包.如图 5(a)所示,可以看出数据流之间的间隔是均匀的,且相邻的不同流之间间隔约为 $10\ \mu s$,与测试系统发包频率 10 Kpps 相符.但是,在经过 TSN 交换机时,TAS 引擎按照表 2 配置时隙对数据流整形,整形后如图 5(b)所示,在一个周期内,相邻数据流之间的间隔发生了变化,我们配置的相邻流之间的理论间隔为: $20\ \mu s$ (10 个时隙), $10\ \mu s$ (5 个时隙), $40\ \mu s$ (20 个时隙), $30\ \mu s$ (15 个时隙),从图 5(b)中也可以看出相邻流之间的间隔基本符合 TAS 配置.由此可以得出 TAS 引擎可以有效的按照预配置时隙发包.

表 2 TAS 时隙配置

数据流(ID)	时隙(CS)	队列
1	5	1
2	15	2
3	20	3
4	40	4
5	55	5

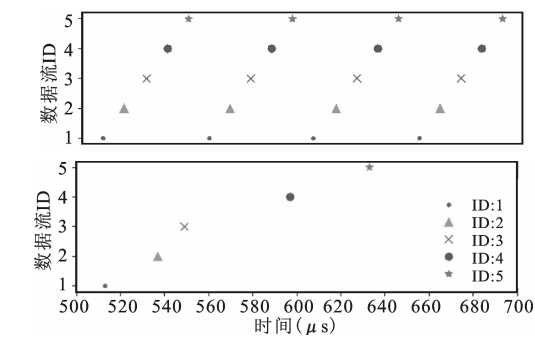


图 5 TAS 机制功能测试结果

5.2 保护时隙校准

如表 3 所示,我们采集了一个周期里,在使能保护时隙调整前和使能后的每条数据流的接收时间.从表 3 中第二列我们可以看出,在未使能保护时隙调整前,虽然 TAS 引擎可以按照预配置时隙发包,但是数据流之间的实际间隔与我们配置的时隙间隔还是存在一定误差的.而误差原因,如我们在第 4.4 节所分析,一方面,随着 PTP 时钟频率和相位的不断校准,CS 长度会开始与 PTP 时钟产生偏差;另一方面,每个晶振输出信号的频率不可避免会有一定的偏差,所以 CPU 时钟频率自身也存在误差.而这种误差会导致不同设备的 TAS 周期长度不同,PS 产生偏差,从而破坏 TAS 整形机制.

使能 PB 调整后,数据流在一个周期内的接收时间如表 3 第三列所示,需要强调的是,我们关心的不是接收时间本身的变化,而是不同流的接收时间之间的相对间隔是否符合预配置的时隙间隔.我们计算所有流相对于流(ID=1)的时间间隔,如图 6 所示,横坐标为数据流 ID,纵坐标为在一个周期内(200 μs)其他 4 条流相对于流(ID=1)的时间间隔.图中最上面的线为未应用 PB 调整时的时间间隔,中间的虚线为理想的时间间隔(预配置的时隙间隔),最下面的实线为应用了 PB 调整后的时间间隔.

表 3 使能 PB 调整前/后接收时间采集

ID	接收时间 (未使能 PB 调整)	接收时间 (使能 PB 调整)
1	513 465	532 277
2	537 466	550 888
3	549 472	561 090
4	597 472	600 078
5	633 466	628 880

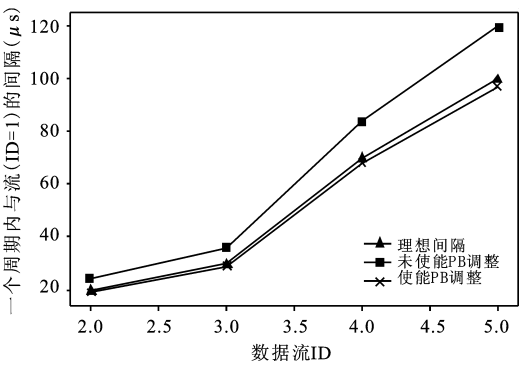


图 6 保护时隙调整算法效果评估

(ID=1)预配置间隔时隙的增加,实际的数据流时间间隔误差也在增加.这是因为误差是出现在每个采用 CS 自计数的 PS 内的,随着间隔时隙的增加,这种误差也在不断的累积.当我们引入 PB 调整后,从图 6 中最下面实线可以看出,实际的时间间隔开始逼近理想时间间隔.从本实验结果,我们可以得出,增加保护时隙调整,可以有效的减少由于 CPU 时钟频率误差及 PTP 时钟频率调整所引入的 PS 偏移,增强了 TAS 机制的鲁棒性.

6 结束语

时间感知整形(TAS)是时间敏感网络(TSN)的重要机制,TAS 的实现依赖于具体的平台.多核网络处理器具有高并行处理能力,可以作为 TAS 机制实现的理想平台.本研究详细论述了基于多核网络处理器的 TAS 机制实现方案,可为其他平台的 TAS 机制设计与实现提供理论参考.然而,本文只是从功能验证了 TAS 引擎的有效性,讨论了 TAS 机制的鲁棒性的提升,并没有进一步讨论 TAS 性能优化,我们后续研究将着力于基于多核网络处理器的 TAS 性能的测试和优化.

参考文献:

[1] MESSENGER J L. Time-Sensitive Networking: An Introduction [J]. IEEE Communications Standards Magazine, 2018, 2 (2): 29-33.

[2] 朱明. 5G 中网络切片研究的现状与挑战 [J]. 网络新媒体技术, 2018, 7 (1): 1-6.

[3] DÜRR F, NAYAK N G. No-wait packet scheduling for IEEE time-sensitive networks (TSN)[C]// Proceedings of the 24th International Conference on Real-Time Networks and Systems. Brest, France: ACM, 2016: 203-212.

quantum-dot cellular automata[J]. IEEE Transactions on Nanotechnology, 2004, 3(1): 26-31.

[7] TOUGAW P D, LENT C S. Logical devices implemented using quantum cellular automata[J]. Journal of Applied Physics, 1994, 75(3):1818-1825.

[8] TOUGAW P D, LENT C S. Dynamic behavior of quantum cellular automata [J]. Journal of Applied Physics, 1996, 80(8):4722-4736.

[9] TOTH G, LENT C S. Role of correlation in the operation of quantum-dot cellular automata[J]. Journal of Applied Physics, 2001,89(12): 7943-7953.

[10] TORRES F S, WILLE R, NIEMANN P, et al. An energy-aware model for the logic synthesis of quantum-dot cellular automata[J]. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 2018, 37(12):3031-3041.

[11] KARIM F, WALUS K. Efficient simulation of correlated dynamics in quantum-dot cellular automata (QCA) [J]. IEEE Transactions on Nanotechnology, 2014, 13(2):294-307.

[12] COPLIEN J. Curiously recurring template patterns [J]. C++ Report, 1995, 7(2):24-27.

[13] CAMPOS C A T, MARCIANO A L, VILELA NETO O P, et al. USE: A universal, scalable, and efficient clocking scheme for QCA[J]. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 2016, 35(3):513-517.

[14] TASKIN B, HONG B. Improving line-based QCA memory cell design through dual phase clocking[J]. IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 2008, 16(12):1648-1656.

作者简介:

肖贞杰 男,(1992-),硕士研究生. 研究方向为量子电路设计.

彭 斐 男,(1985-),博士研究生. 研究方向为电子设计自动化.

徐 栋 男,(1993-),硕士研究生. 研究方向为数字集成电路设计.

邝 锐 男,(1994-),硕士研究生. 研究方向为量子电路设计.

解光军(通讯作者) 男,(1970-),博士,教授,博士生导师. 研究方向为微纳电路与系统、集成电路设计.

E-mail: gjxie8005@hfut.edu.cn

(上接第 5 页)

[4] NAYAK N G, DüRR F, ROTHERMEL K. Time-sensitive Software-defined Network (TSSDN) for Real-time Applications[C]// Proceedings of the 24th International Conference on Real-Time Networks and Systems. Brest, France: ACM, 2016:193-202.

[5] 杨吉喆,王玲玲,陆建德. 基于 Octeon 多核网络处理器的 IPv6 联动 IPS 研究与设计 [J]. 微电子学与计算机, 2011, 28 (12): 79-83.

[6] 吴霖. 基于多核网络处理器的 P2P 高速流量识别系统的设计与实现 [D]. 苏州: 苏州大学, 2010.

[7] 基于多核网络处理器的流控技术研究 [D]. 北京: 北京邮电大学, 2014.

[8] IEEE. 802.1Qbv-2015. IEEE Standard for Local and metropolitan area networks -Bridges and Bridged Networks-Amendment 25: Enhancements for Scheduled Traffic [S]. New York, NY, USA: IEEE, 2016.

作者简介:

贾正义 男,(1991-),博士研究生. 研究方向为软件定义网络、时间敏感网络. E-mail: jiazy@dsp.ac.cn

陈 晓 男,(1964-),硕士,研究员. 研究方向可编程网络、软件定义网络.

王劲林 男,(1964-),硕士,研究员. 研究方向为 5G 网络与信息安全、时间敏感网络.

叶晓舟 男,(1980-),博士,研究员. 研究方向 5G 网络与信息安全.