

一种基于最长队列预测的 CICQ 交换结构调度算法

彭来献 恽姿 赵文栋 田畅
(解放军理工大学通信工程学院 南京 210007)

摘要: CICQ(Combined Input Crosspoint Queued)是一种在 crossbar 交叉点加入少量缓存的交换结构, 具有无需内部加速比及分布并行调度的特性。为了自适应网络环境中各种业务流量, 提高在非均匀流量下的性能, 该文提出了一种基于最长队列预测的高效 CICQ 交换结构调度算法——RR-LQD (Round Robin with Longest Queue Detecting)。RR-LQD 算法复杂度为 $O(1)$, 具有良好的可扩展性; 通过预测局部“最长”队列并尽力为其服务, 保持调度中队列长度的均衡, 能够适应各种非均匀流量的网络环境。仿真结果表明: 在各种均匀和非均匀流量下, RR-LQD 算法均能达到 100%的吞吐量, 并且具有优良的时延性能。该文使用 FPGA 芯片实现了 RR-LQD 算法仲裁器, 能够满足高速、大容量交换结构的设计需要。

关键词: 调度算法; CICQ; 非均匀流量; 最长队列预测

中图分类号: TP393.05

文献标识码: A

文章编号: 1009-5896(2010)06-1457-06

DOI: 10.3724/SP.J.1146.2009.00908

A Novel Scheduling Algorithm Based on Longest Queue Detecting for CICQ Switching Fabrics

Peng Lai-xian Yun Zi Zhao Wen-dong Tian Chang

(Institute of Communications Engineering, PLA University of Science and Technology, Nanjing 210007, China)

Abstract: CICQ (Combined Input Crosspoint Queued) is a switch fabric with buffers in crosspoints, which brings its nice distributed parallel scheduling property with no internal speedup required. In order to adapt to various traffics in the network environment, and improve performance under nonuniform traffics, a novel efficient scheduling algorithm with longest queue detecting——RR-LQD (Round Robin based on Longest Queue Detecting) is proposed. RR-LQD provides fine scalability with the complexity of $O(1)$. The local longest queue being detected and serviced all along guarantees length balance among input queues, which makes RR-LQD adapt to the network environment of nonuniform traffic automatically. The simulation results show that the RR-LQD can achieve 100% throughput and low delay under various uniform and non-uniform traffics. In this paper, the arbiter of RR-LQD is implemented by a FPGA chip. That proves RR-LQD can meet the demands of high-speed and large capacity switching fabrics.

Key words: Scheduling algorithm; Combined Input Crosspoint Queued(CICQ); Nonuniform traffic; Longest Queue Detecting(LQD)

1 引言

随着近年来 VLSI 技术的发展, 在 crossbar 的交叉点能够植入少量缓存, 依据目前的技术条件, 交叉点缓存容量比较小, 仍然需要与输入排队结合使用, 这就形成了联合输入交叉点排队(CICQ)交换结构。与传统基于 crossbar 的高速交换结构相比, CICQ 由于在 crossbar 交叉点设置了缓存, 能够将输入端与输出端的数据传输竞争隔离开来, 调度算法可以在输入/输出端独立并行工作, 不需要集中控

制。因此, CICQ 交换结构具有分布式实现、易扩展的良好特性, 是构建高速、大容量、可扩展路由器的理想选择^[1]。同时, CICQ 交换结构及其调度算法也成为高速交换技术领域近期一个研究热点。国内外对此进行了大量的研究, 文献[2]首次提出了在输入端使用 VOQ(Virtual Output Queue)排队机制的 CICQ 结构, 在业务流量负载较大(大于 65%)情况下数据传输平均时延优于单纯的输入排队 crossbar 交换结构, 目前对调度算法的研究普遍基于这种 VOQ+CICQ 的结构。

根据研究方法的不同, 整体上看, 关于CICQ交换结构调度算法的研究分为两大类: 一类是纯理论

2009-06-23 收到, 2009-10-19 改回

江苏省自然科学基金(BK2007001)资助课题

通信作者: 彭来献 penglaixian@sina.com

分析,探讨CICQ调度算法的吞吐量、时延、服务质量保证等性能的理论基础。文献[3-9]分别研究了100%吞吐量、带宽保证、时延保证、公平性等问题,研究成果为CICQ交换结构调度算法的研究和发展奠定了坚实的理论基础,但是提出的算法和调度机制却存在复杂、硬件难以实现的问题,无法在高速的环境下应用。

关于CICQ交换结构调度算法另一类研究方法是从实用性出发,以交叉点小缓存、高效、低复杂度、硬件易实现作为性能指标,设计各种易于硬件实现的快速启发式调度算法。这类算法的共同点在于减少控制信息量和复杂的计算,一般采用两种手段:一种是依据交叉点缓存的状态信息进行调度;另一种是基于轮转(round-robin)机制。文献[10,11]提出的算法MCBF(Most Critical Buffer First)和LFF-LBF算法都是依据交叉点缓存的忙闲来进行输入/输出调度,使得交叉点缓存得到充分的利用,仿真结果表明算法在均匀流量下具有良好的时延性能。然而这种算法没有考虑输入端队列的状态,在非均匀流量下会出现不稳定现象,导致吞吐量、时延性能严重下降。

由于轮转机制具有低复杂度、硬件易实现等优良特性,成为实用性研究方面的重点,其中典型的调度算法有RR(Round-Robin)-RR(Round-Robin)^[12]和LQF(Longest Queue First)-RR^[3]。虽然RR-RR简单易实现、复杂度仅为 $O(1)$,但是在非均匀流量下不能提供良好的吞吐量和时延性能。LQF-RR算法在输入端总是优先服务最长的VOQ,尽量保持各个队列调度均衡,在各种流量下都具有优良的性能,然而该算法复杂度为 $O(\log N)$,限制了算法在高速处理环境下的应用。为了提高CICQ交换结构在非均匀流量下的性能,研究者通过改进轮转指针更新的规则,基于RR-RR提出了众多改进算法,例如RR-AF(Adaptable-size Frame-based)^[13], DRR(Differential Round Robin)^[14], FD(Full Draining)-RR^[15], QD(Quantum-based Draining)-RR^[15]以及FDR(Fair service and Dynamic Round robin)^[16]等算法。这些算法基本思想都是通过为每个VOQ分配固定的调度份额、或者根据相邻调度VOQ长度关系的“差分因子”来约束轮转指针的更新。上述算法复杂度都是 $O(1)$,保留了RR-RR低复杂度的特点,仿真结果表明,当每个交叉点容量是1个信元大小时,在非均匀流量下能够获得较好的吞吐量和时延性能,然而它们的“份额”或者“差分因子”的取值都是仿真得到的经验值,难以穷举各种复杂的业务流量进行可靠性验证。

因此,高效、实用的调度算法应当在简单的RR-RR算法的基础上,通过某些指针更新策略弥补其在非均匀流量下性能不佳的缺憾,并且新的更新策略不依赖其他人为设定的“经验”参数,能够保持低复杂度,并能适应各种非均匀流量网络环境,为此,本文也从实用性角度出发,提出一种基于最长队列预测的轮转型调度算法RR-LQD(Round Robin Based on Longest Queue Detecting),其主要思想是在输入端内局部预测队列最长的VOQ并尽力为该队列服务,保持调度中各个队列长度均衡,提高系统稳定性。RR-LQD算法依据队列长度信息进行调度,消除了对各种经验参数的依赖,从根本上能够适应各种业务流量,仿真结果证明,这十分有助于提高CICQ交换结构在非均匀流量下吞吐量、时延等性能。同时,RR-LQD算法以求解局部最佳取代LQF-RR算法的全局最佳,从而省略了排序比较的过程,大大降低了算法的复杂度,算法复杂度仅为 $O(1)$,却能够提供近似于LQF-RR算法的吞吐量、时延性能。

2 CICQ 交换结构及其调度描述

如图1所示,一个 $N \times N$ 的CICQ交换结构由 N 个输入/输出端和带交叉点缓存的Crossbar组成。每个交叉点的缓存记为CB(Crosspoint Buffer),在每个输入端内均采用VOQ的队列组织方式,用 VOQ_{ij} 表示第 i 个输入端缓存中的第 j 条虚拟输出队列,将与之对应的交叉点缓存记做 CB_{ij} ,其中 $1 \leq i, j \leq N$ 。

为解决输入/输出端的竞争,每一个输入/输出端都分别设有一个调度器。通常,为了实现高速交换和控制上的方便,交换结构处理的数据单元为固定长度的信元。各个输入/输出端支持相同的速率,传输一个信元所需的时间称为一个时隙。在每个时隙内,每个输入调度器从其输入端的 N 个VOQ队列中选出一个送往相应的交叉点;每个输出调度器从对应的 N 个交叉点缓存中选出一个输出。输入和输出端调度器之间不需信息的交互,可以分别独立执行。

3 RR-LQD 算法

3.1 算法描述

为了算法描述方便,引入如下符号: VOQ_{ij} 在 t 时刻的队列长度表示为 $L(VOQ_{ij}, t)$; CB_{ij} 在 t 时刻的队列长度表示为 $L(CB_{ij}, t)$; 一个交叉点缓存的最大容量用 C 表示; t 时刻时,若 $L(VOQ_{ij}, t) > 0$ 且 $L(CB_{ij}, t) < C$,称 VOQ_{ij} 在 t 时刻为EVOQ(Eligible VOQ); t 时刻时,若 $L(CB_{ij}, t) > 0$,称 CB_{ij} 为ECB(Eligible Crosspoint Buffer)。

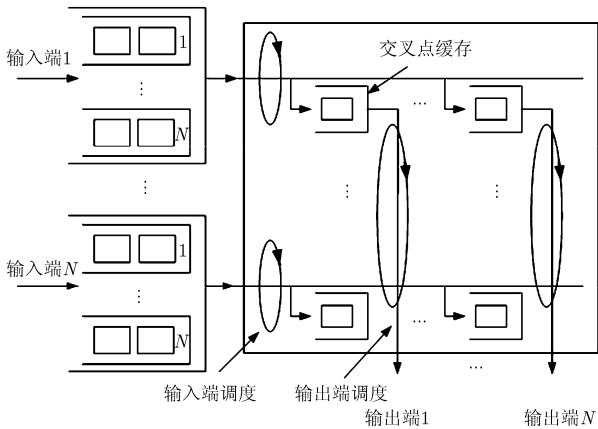


图1 $N \times N$ CICQ交换结构示意图

RR-LQD算法分为输入调度和输出调度两部分，每个输入调度器维护两个指针：最长队列预测指针 dp 和EVOQ轮转指针 rp ；每个输出端调度器维护一个ECB轮转指针 p 。这些指针分别指向当前优先服务的队列(交叉点)。算法具体描述如下：

输入端调度：

(1)最长队列预测 输入调度器从 dp 指向的位置开始，通过轮转规则寻找第一个队列长度大于当前 dp 所指向VOQ的队列。若找到，更新 dp 指针，指向该预测的队列；否则， dp 指针不更新。

(2)判断预测队列是否为EVOQ 此时，调度器认为 dp 指针指向的队列就是该输入端中“最长”的队列，要优先对它服务，判断预测队列是否为EVOQ：

(a)若是EVOQ，则调度器将它的队头信元调度至相应的交叉点缓存；

(b)若不是EVOQ，则调度器从 rp 指向的位置开始，通过轮转规则寻找下一个EVOQ，若找到则调度器将它的队头信元调度至相应的交叉点缓存，并将 rp 更新至该EVOQ的下一个位置；若找不到则 rp 保持不变。

输出端调度：

从 p 指向的位置开始，通过轮转规则寻找下一个 ECB，若找到，则将该交叉点的队头信元调度输出；若找不到则 p 保持不变。

RR-LQD 算法的特点在于：

(1)为降低复杂性，最长队列预测只是寻找局部最长，并向全局最长队列逼近， dp 指针更新不依赖于其它经验性参数。使得一旦预测某一队列为“最长”队列，无论本时段内是否被调度， dp 指针都指向它，在后继时段优先为其服务，从而保证尽快地调度滞留有较多分组的 VOQ，使各个 VOQ 的队长保持均衡，以适应不同网络流量的变化并提高排队

系统的稳定性。

(2) rp 指针用于在当前预测的最长队列不能调度的情况下，找到一个可以被调度的 VOQ，缓解“饥饿”现象的发生，提高了算法调度效率和公平性。

(3)算法不需要 LQF-RR 算法的排序操作，只需要对队长进行大小比较，虽然最坏情况下需要比较 $N-1$ 次才能找到预测的最长队列，但在 3.2 节中分析可知，硬件实现时这些比较可以并行执行，算法的复杂度仅为 $O(1)$ ，与 LQF-RR 算法相比大大降低。

(4)RR-LQD 算法输入端调度器比一般的 RR 调度器要复杂，但从第 4 节的性能分析可以看出，它能够获得更加优良的吞吐量、时延等性能。

3.2 算法的实现

RR-LQD 算法执行由输入/输出调度器共同完成，输出调度器主要由一个简单的 RR 仲裁器构成；输入调度器则相对复杂，图 2 显示了一个输入端调度器的实现框图。

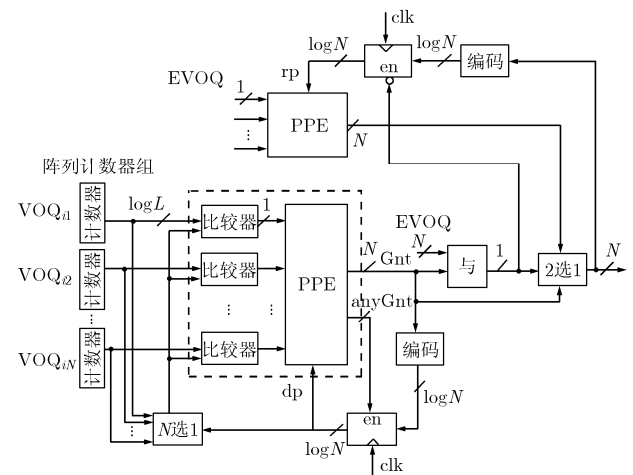


图2 输入调度器的实现框图

输入调度器由一组队列长度比较器和两个 RR 仲裁器组成，一个指针为 rp ，用于寻找下一个 EVOQ；另一个为 dp ，用于最长队列预测。RR 仲裁器主要由可编程优先级编码器(Programmable Priority Encoder, PPE)和一些用来存储指针的触发器组成，其中 PPE 是核心部件，它根据 rp 或 dp 在 N 个输入请求中按轮转规则选择当前最高优先级的请求。最长队列预测过程如下：首先将 dp 指针指向的 VOQ 长度分别与其它队列长度进行比较，如果小于其它队列，则输出‘1’，否则输出‘0’；然后 PPE 根据 dp 值寻找下一个较长的队列。从图 2 中可以看出，两个仲裁器并行工作，如果预测队列是 EVOQ，则输出调度结果，同时禁止 rp 的更新；如果预测队

列不是 EVOQ，则调度结果取决于 rp 的执行结果。由于比较器复杂度与 PPE 相当，而且在每个调度周期中， N 个比较器之间以及两个 PPE 之间都是并行执行的，所以算法的硬件复杂度为 $O(1)$ 。输出调度器只是简单的 RR 仲裁器，这里不再赘述。

4 仿真结果与性能分析

本节对 RR-LQD 算法和已有的几个重要算法：RR-RR, DRR, FD-RR, QD-RR, RR-AF, MCBF, LQF-RR 进行各种流量下吞吐量、时延等性能的分析比较。吞吐量是指一个输出端在一个时隙内平均发送的信元数($\times 100\%$)，也等于输出端口平均利用率。时延是指信元从进入输入队列到被发送到输出端的时间间隔，单位为时隙。仿真采用 32×32 的 CICQ 交换结构，每个交叉点容量是 1 个信元大小，流量模型采用均匀、均匀突发、Log-diagonal、diagonal 和 nonuniform 5 种流量模型，其中后面 3 种是典型的非均匀流量模型。仿真工具采用文献[17]提出的高速交换网络仿真系统，仿真长度为 500,000 个时隙。

(1)均匀流量：一种理想的业务流，每个输入端到达的信元均匀分布于各个输出端。信元到达服从参数为 λ 的独立同分布(i.i.d.)贝努利过程， λ 表示输入端流量负载。

(2)均匀突发流量：信元到达服从参数为 b 的 ON/OFF 突发模型， b 表示突发长度。

(3)Log-diagonal 流量：信元到达是 i.i.d. 贝努利过程，各个 VOQ 按 $\lambda_{ij} = 2\lambda_{|j+1|}$ 分布，其中 $|j+1| = (j+1) \bmod N, 1 \leq i, j \leq N$ 。

(4)diagonal 流量：信元到达过程也是 i.i.d. 贝努

利过程，每个输入端的负载只分布在两个 VOQ 上，按式(1)分布：

$$\lambda_{ij} = \begin{cases} \frac{2}{3}\lambda, & j = i \\ \frac{1}{3}\lambda, & j = |i + 1| \end{cases}, 1 \leq i, j \leq N \quad (1)$$

(5)nonuniform 流量：每个输入端到达的信元按式(2)分布于各个 VOQ：

$$\lambda_{ij} = \begin{cases} \lambda \cdot \left(w + \frac{1-w}{N} \right), & j = i \\ \lambda \cdot \frac{1-w}{N}, & j \neq i \end{cases}, 1 \leq i, j \leq N \quad (2)$$

其中 $0 \leq w \leq 1$ 被称为非均匀(nonuniform)因子，当 $w=0$ 时，该流量就是均匀业务流量； $w=0.5$ 时，非均匀程度最高。

4.1 吞吐量

表 1 列出了均匀 i.i.d. 贝努利流量下各个算法的最大吞吐量，可以看出每种算法都具有接近 100% 的吞吐量，说明在该流量下还无法比较出算法性能的差异。均匀突发流量下的吞吐量和表 1 的结果相似，在此省略。

如表 2 所示，在 Log-diagonal 流量下，RR-LQD 算法的最大吞吐量为 0.9764，略低于 LQF-RR 的 0.9985，相差仅 2%，而比 RR-RR 算法高出 12%，也略高于其它算法。在 diagonal 流量下，RR-LQD 算法拥有与 LQF-RR 算法相同的吞吐量，均高于其它算法。在这两种流量下，RR-RR 和 MCBF 算法的吞吐量性能大大下降，说明不能很好地适应非均匀的流量。

表 1 各算法在均匀 i.i.d. 贝努利流量下的最大吞吐量

RR-RR	DRR	FD-RR	QD-RR	RR-AF	MCBF	LQF-RR	RR-LQD
0.9987	0.9988	0.9987	0.9987	0.9988	0.9988	0.9987	0.9987

表 2 各算法在 Log-diagonal 和 diagonal 流量下的最大吞吐量

流量	RR-RR	DRR	FD-RR	QD-RR	RR-AF	MCBF	LQF-RR	RR-LQD
Log-diagonal	0.8716	0.9678	0.9638	0.9684	0.9441	0.9059	0.9985	0.9764
diagonal	0.8793	0.9983	0.9976	0.9956	0.9019	0.8794	0.9987	0.9987

图 3¹⁾显示的是 nonuniform 流量下， w 取不同值时各算法的最大吞吐量比较。可以看出明显的差距： w 趋于 0.5 时，RR-RR 的吞吐量降至 0.85 以下，性能较差。而 RR-LQD 算法和 LQF-RR 算法能够很好地适应非均匀流量的变化，始终保持接近 100%

的吞吐量，略高于其它算法。

4.2 平均时延

图 4 显示的是各算法在均匀流量模型下的平均时延。仿真结果说明，在均匀分布业务流到达情况下，各算法的平均时延性能几乎相同。图 5 比较了各算法在不同突发长度的突发流到达情况下信元的平均时延性能。可以发现，在同一突发长度的流量

¹⁾为了绘图清晰方便，只选择图中 5 种代表性的算法进行比较说明。

下,各算法时延性能基本相同。突发长度增加时,会导致各算法平均时延性能的下降。这是由于当信元突发到达时,队列中的信元得不到及时调度,造成等待时间的累积增加。流量突发导致时延性能下降的现象在现实网络中普遍存在,可以使用流量整形等缓解突发的措施,以提高系统的时延性能。

Log-diagonal 和 diagonal 两种流量下的时延性能仿真结果如图 6、图 7 所示。RR-LQD 算法的性能仅在 Log-diagonal 流量下略次于 LQF-RR 算法,而相对于其它算法,RR-LQD 算法都具有明显的优势,原因在于它的最长队列预测机制在保证较长队列拥有较多服务机会的同时,公平地服务其它队列,使输入端队列长度趋于平衡,从而有效确保了排队

系统的稳定。同时可以看出,RR-RR 算法在重负载下难以适应这两种非均匀流量,出现不同程度的不稳定现象,导致吞吐量和时延性能的恶化。

对于 nonuniform 流量模型,仿真了 $w=0.1\sim 0.9$ 的各种取值下各个算法的平均时延,由于篇幅所限,这里只给出 $w=0.5$ 时的仿真结果,如图 8 所示。RR-LQD 算法拥有与 LQF-RR 算法相似的时延性能,而远优于其它算法。RR-RR 算法在负载 0.85 以下就出现不稳定现象,QD-RR 算法的平均时延也在高负载下快速增长,而 RR-LQD 和 LQF-RR 算法即使在 0.99 的负载下,平均时延也只在 60 以下,显示了理想的时延性能。这也充分说明 RR-LQD 算法中最长队列预测机制能有效发挥作用,可以灵活地适应各种非均匀流量的变化。

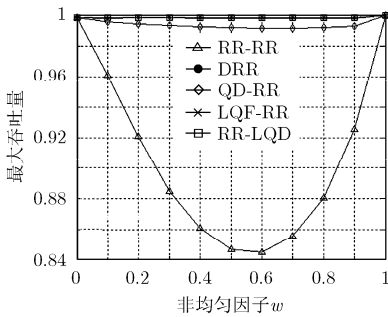


图 3 nonuniform 流量下的最大吞吐量

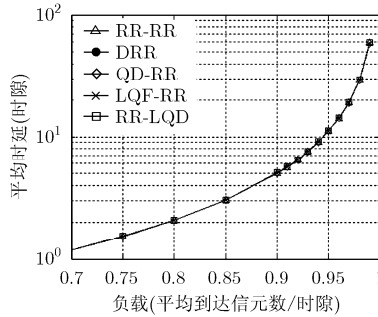


图 4 均匀 i.i.d. 贝努利流量下的平均时延

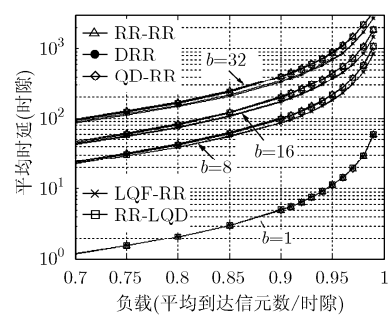


图 5 均匀突发流量下的平均时延

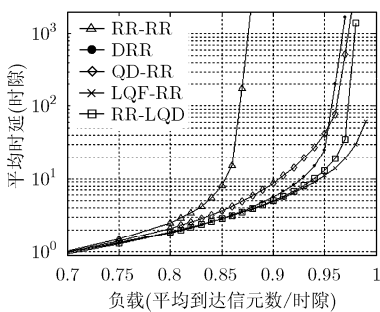


图 6 Log-diagonal 流量下的平均时延

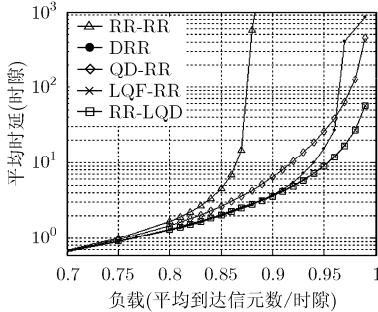


图 7 diagonal 流量下的平均时延

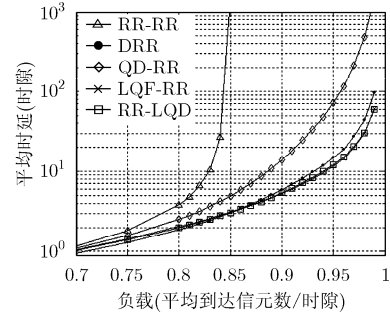


图 8 nonuniform($w=0.5$)流量下的平均时延

5 调度器的 FPGA 实现

从上文图 2 中可以看出,算法输入调度器相对复杂,硬件实现时面积和时延耗费主要集中在图中虚线方框内。本节主要使用 FPGA 设计并实现 PPE 和比较器。文献[18]中提到了几种 PPE 的实现方法: Ripple, Exh, PPE_cmb, Shift_Enc 和一种新的结构称作 Proposed,其中 Proposed 结构的 PPE 综合性能最优,面积耗费较大,但是时延耗费最小。该 PPE 也在 CISCO 高速路由器中得到实用,本文选用此结构的 PPE。

在 32×32 的交换规模下,队列长度计数器的长

度设定为 16 bit。选择目标 FPGA 器件为 Virtex II VC2V40 FG256,使用 Xilinx WebPACK 10.1 工具,采用上述不同 PPE 结构进行 FPGA 设计和仿真,队长比较器和 PPE 总共的时延和面积耗费仿真结果如表 3 所示。综合时延和面积耗费两个指标考虑^[18],Proposed 结构的 PPE 是最佳的选择。假设信元长度为网络中最短的 IP 包(40 byte),那么一个时隙长度为 32 ns,显然 RR-LQD 算法调度器能够满足 10 Gbps 链路的高速处理需求。

6 结束语

为了在低复杂度的条件下,提高 CICQ 交换结

表 3 时延(ns)与面积耗费(slices)

耗费 时延 面积	Ripple	PPE_cmb	Exh	Shift_Enc	Proposed ^[18]
	43.661	48.089	19.830	27.392	19.856
	47	70	311	204	91

构调度算法在复杂网络流量下的性能,从实用性角度出发,本文提出一种基于最长队列预测的算法 RR-LQD。RR-LQD 算法在 RR-RR 算法基础上发展而来,在输入端调度中预测局部“最长”队列并尽力为其服务,保持调度中各个队列长度的均衡。这种策略不仅具有复杂度低的优势,还能够自动适应网络流量的变化,特别是在各种非均匀流量下具有良好的性能。仿真结果表明,算法在各种均匀、非均匀流量下均能达到 100% 的吞吐量,并且具有近似于 LQF-RR 算法的时延性能,且复杂度仅为 $O(1)$,易于硬件实现。性能分析和 FPGA 实现结果表明,RR-LQD 算法能够满足高速、大容量 CICQ 交换结构的设计需要,可应用于高性能的路由交换设备中。

参 考 文 献

- [1] 张小平,刘振华,赵有健,关洪涛. 可扩展路由器[J]. 软件学报, 2008, 19(6): 1452-1464.
Zhang X P, Liu Z H, Zhao Y J, and Guan H T. Scalable router. *Journal of Software*, 2008, 19(6): 1452-1464.
- [2] Nabeshima M. Performance evaluation of a combined input- and crosspoint-queued Switch[J]. *IEICE Transactions on Communications*, 2000, E83-B(3): 737-741.
- [3] Javid T, Magill R, and Hrabik T. A high-throughput scheduling algorithm for a buffered crossbar switch fabric[C]. IEEE International Conference on Communications (ICC). Helsinki, Finland: 2001: 1586-1591.
- [4] Magill R, Rohrs C, and Stevenson R. Output-queued switch emulation by fabrics with limited memory[J]. *IEEE Journal on Selected Areas in Communications*, 2003, 21(4): 606-615.
- [5] Mhamdi L and Hamdi M. Output queued switch emulation by a one-cell-internally buffered crossbar switch[C]. IEEE Global Telecommunications Conference(GLOBECOM). San Francisco, USA: 2003: 3688-3693.
- [6] Chuang S T, Iyer S, and McKeown N. Practical algorithms for performance guarantees in buffered crossbars[C]. Proceedings of IEEE INFOCOM, Miami, Florida: 2005: 981-991.
- [7] He S M, Sun S T, and Guan H T, *et al.* On guaranteed smooth switching for buffered crossbar switches[J]. *IEEE/ACM Transactions on Networking*, 2008, 16(3): 718-731.
- [8] George F G. Buffered cross-bar switches, revisited: Design steps, proofs and simulations towards optimal rate and minimum buffer memory. *IEEE/ACM Transactions on Networking*, 2008, 16(6): 1340-1351.
- [9] Pan D and Yang Y Y. Localized independent packet scheduling for buffered crossbar switches[J]. *IEEE Transactions on Computers*, 2009, 58(2): 260-274.
- [10] Mhamdi L and Hamdi M. MCBF: A high-performance scheduling algorithm for buffered crossbar switches[J]. *IEEE Communications Letters*, 2003, 7(9): 451-453.
- [11] 郑敏, 郑竹林, 王斌. 一种基于缓存状态的 CICQ 交换机[J]. 电子与信息学报, 2007, 29(12): 2978-2980.
Zheng M, Zheng Z L, and Wang B. A new CICQ fabric based on state of cross-point-buffer. *Journal of Electronics & Information Technology*, 2007, 29(12): 2978-2980.
- [12] Yoshigoe K and Christensen K J. An evolution to crossbar switches with virtual output queuing[J]. *IEEE Network*, 2003, 17(5): 48-56.
- [13] Rojas-Cessa R and Oki E. Round-robin selection with adaptable-size frame in a combined input-crosspoint buffered switch[J]. *IEEE Communications Letters*, 2003, 7(11): 555-557.
- [14] Luo J Z, Le Y, and Wu J. A non-uniform traffic oriented scheduling algorithm in combined input-crosspoint-queued (CICQ) switch[J]. *Emergent and Distributed System*, 2006, 21(4): 279-292.
- [15] 郑燕峰, 孙书韬, 贺思敏等. 基于双轮转指针的输入与交叉点联合排队型(CICQ)交换结构调度算法[J]. 计算机研究与发展, 2006, 43(7): 1225-1232.
Zheng Y F, Sun S T, and He S M, *et al.* A dual round-robin algorithm for combined input-crosspoint-queued switches. *Journal of Computer Research and Development*, 2006, 43(7): 1225-1232.
- [16] 扈红超, 伊鹏, 郭云飞等. 一种公平服务的动态轮询调度算法[J]. 软件学报, 2008, 19(7): 1856-1864.
Hu H C, Yi P, and Guo Y F, *et al.* A fair service and dynamic round robin scheduling algorithm. *Journal of Software*, 2008, 19(7): 1856-1864.
- [17] Stanford University. SIM manual[R]. <http://klamath.stanford.edu/tools/SIM/>. 2008.10.
- [18] Gupta P and McKeown N. Designing and implementing a fast crossbar scheduler[J]. *IEEE Micro*, 1999, 19(1): 20-28.

彭来献: 男, 1978 年生, 副教授, 博士, 研究方向为宽带网络技术、Ad hoc 网络技术。

恽姿: 女, 1985 年生, 硕士生, 研究方向为宽带网络技术。

赵文栋: 男, 1972 年生, 讲师, 博士生, 研究方向为无线传感器网络、宽带网络技术。

田畅: 男, 1963 年生, 教授, 博士, 从事宽带网络技术、网络安全和无线分组网的研究。