高速 crossbar 控制算法 iDRR 及其性能分析

彭来献,田 畅,郑少仁
(解放军理工大学通信工程学院交换技术与 ATM 研究中心,江苏南京 210007)

摘 要: DRR(Dual Round-Robin)算法[6]是一种公平、高效、可扩展性强、硬件实现简单的 crossbar 控制算法. 为了进一步改善算法的时延性能和公平性,文中提出了多重迭代 DRR 算法,即 iDRR 算法,它继承了 DRR 算法所有优点. 仿真结果表明 iDRR 算法可达到 100 %吞吐量,在时延性能和公平性方面都优于 DRR 算法. 使用可编程逻辑器件实现了基于 iDRR 算法的仲裁器,工作频率达 80MHz,可支持 10Gbps 速率的输入端口,可用于超高速、大容量的路由器中.

关键词: 输入排队; 匹配算法; VOQ; DRR; iDRR
中图分类号: TP393. 05 **文献标识码**: A **文章编号**: 0372-2112 (2003) 10-1465-04

iDRR Algorithm in High-speed Crossbar and Its Performance Analysis

PENG Lai-xian, TIAN Chang, ZHENG Shao-ren
(Institute of Communication Engineering, PLA University of Science and Technology, Nanjing, Jiangsu 210007, China)

Abstract: Dual Round-Robin (DRR) algorithm[6] is a crossbar control algorithm that is fair, highly efficient, scalable and easy to implement by hardware. To improve delay performance and fairness about DRR algorithm, we advance DRR to iDRR which can run in iterations and inherits all of advantages of DRR. The results of simulation indicate that iDRR can achieve 100 % throughput, furthermore, both delay performance and fairness about iDRR precede that about DRR. We implement arbiter based on iDRR by programmable logic device which can operate at clock frequency of 80MHz. It can support 10Gbps input ports, and can be used in super high speed and large capacity routers.

Key words: input queuing; matching algorithm; VOQ; DRR; iDRR

1 引言

随着多媒体业务和其它 Internet 业务的增加,高速交换机和 IP 路由器在 Internet 干线上得到广泛的应用. 为了提高效率,通常采用定长交换技术,不同长度的 IP 分组在交换前先划分成固定长度的"信元",在输出端重组后再发送到链路上去. crossbar 作为一种快速的定长交换网络,被广泛应用于高速路由器[1,3,4]、LAN 和 ATM 交换机中.

为了缓解拥塞,通常需要在输入端(或输出端)设置信元缓存. 传统的低端交换设备采用输出排队,其最大的优点是对任何分布的数据流都能获得最优的时延、吞吐量等性能,但是要求交换网络的速率是线路速率的 N(N 指输入端数目)倍,而在 Internet 干线中线路速率往往很高(如 OC-192、10GE),交换网络难以达到数十 Gbps 的速率,并且系统可扩展性差. 输入排队却只要求交换网络的速率与线路速率相同,与输出排队相比,输入排队更适合用于高速交换.

如果输入队列只采用简单的 FIFO(First In First Out)队列机制,其 HOL 阻塞(Head of Line Blocking)将限制交换网络的最大吞吐量为 58.6 %[2]. 因而,一般采用虚拟输出排队技术(VOQ: virtual output queueing)来消除 HOL 阻塞,即一个输入端为每一个输出端维护一个 FIFO 队列,这样可使交换网络获得 100 %的吞吐量[5]. 目前许多产品都采用基于 crossbar 交换网络的"输入排队 + VOQ"的队列系统,例如,Tiny-Tera 太比特路由器原型[1],Cisco GSR12000 系列 IP 路由器[3]和 BBN MGR 系列 IP 路由器[4].

另一方面,由于采用了 VOQ,还必须使用有效的 crossbar 控制算法解决输入端的信元输出竞争,保证交换网络的带宽得到充分利用,并保证时延等其它性能. crossbar 控制算法又称为匹配算法,它根据输入队列的状态,决定输入和输出端的匹配关系,并控制 crossbar 交叉点的开合,最终建立输入/输出端信元传输通路.

当前,对匹配算法的研究已取得众多理论成果,例如采用最大权重匹配算法可以获得 100 %的吞吐量,算法稳定[5]. 但是,这些算法复杂度高,或者需要维护的信息结构复杂,难以用硬件实现,缺乏实用性. 在具体应用中通常使用近似或启发式算法,又称为极大匹配算法,如 iSLIP[5],DRR[6]等等. 这类算法复杂度低,可扩展性强,简单易实现.

文献[6]中提出的 DRR 算法在输入和输出端同时使用轮

收稿日期:2002-04-27;修回日期:2002-08-30
基金项目:国家"863"支持课题(No. 2001AA121063)

转优先级(Round-Robin)仲裁器解决竞争,算法简单、高效、可扩展性强,适用于分布式处理,硬件易实现.为了进一步改善DRR算法的时延等性能,本文对DRR算法进行了改进,采用多重迭代的方法降低了信元等待时延,同时也改善了算法的公平性.

2 iDRR算法描述

2.1 输入排队交换机和匹配问题

图1(a)所示是一个基于crossbar交换网络的 $N \times N$ 输入排队交换机,它采用了VOQ技术.图中仲裁器用于管理队列信息,执行匹配算法,控制crossbar开合.输入端总共有 N^2 个FIFO,构成交换机的队列系统,信元存储在输入队列中.

crossbar交换一个信元的时间间隔称为一个时隙.信元到达过程可用一个离散时间随机过程 $A_i(k)$ 表示,$1 \le i \le N$, k 指时隙数.到达输入端 i 且目的端为 j 的信元,被存储在FIFO Q_{ij} 中,信元平均到达 Q_{ij} 的速率为 λ_{ij},即每个时隙平均到达的信元数.为了下文讨论方便,假设 Q_{ij} 长度无限.

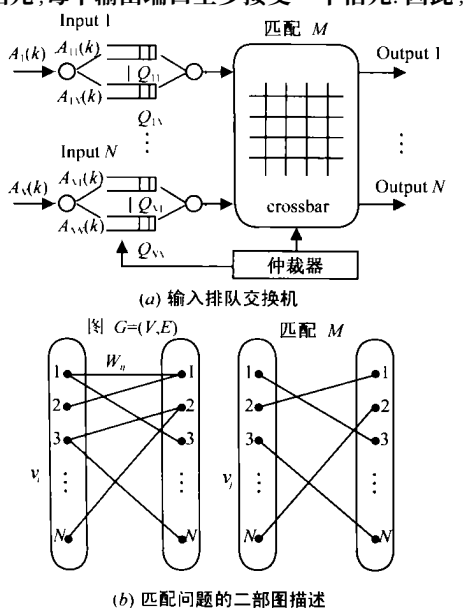
定义1 如果 λ_{ij} 满足约束条件:

$$\begin{cases} \sum_{j=1}^{N} \lambda_{ij} \le 1, & 1 \le i \le N \\ \sum_{i=1}^{N} \lambda_{ij} \le 1, & 1 \le j \le N \end{cases} \tag{1}$$

则称 $A_i(k)$ 可接受(admissible),否则不可接受(inadmissible).

在本文中,如果信元按均匀独立同分布到达每个输入端,并且平均到达速率为 λ,则有 $\lambda_{ij} = \lambda / N$, $\lambda_i = \sum_{j=1}^{N} \lambda_{ij} = \lambda (1 \le i, j \le N)$.在本文中数据流均是可接受的单播数据.

由于crossbar无阻塞、无内部存储,为了避免发送和接受冲突,匹配算法必须保证在一个时隙内每个输入端口至多发送一个信元,每个输出端口至多接受一个信元.因此,crossbar的匹配问题通常采用二部图来描述,如图1(b)所示.



(a) 输入排队交换机

(b) 匹配问题的二部图描述

图1 (a)输入排队交换机;(b)匹配问题的二部图描述

图1(b)中左边结点表示输入端,右边表示输出端;如果结点 v_i 与 v_j 之间有一条边,则表示 Q_{ij} 有信元等待发送,并且该边被赋予权重 w_{ij}.一个匹配 M 就是这些边的一个子集,并且保证任一结点不能同时有两条边连接.

2.2 iDRR算法描述

首先给出DRR算法的描述.在DRR算法中,每个输入和输出端都有一个轮转优先级仲裁器,根据轮转优先级规则在二部图中寻找合适的"边",构成一个匹配.算法一次执行过程的具体描述如下[6,7]:

Step1:请求 每个输入端仲裁器有一个最高优先级请求指针 r_i.如果输入端有发送请求,从 r_i 指向的位置开始,根据轮转优先级规则,输入端选择VOQ发送请求. r_i 当且仅当step2中该请求被许可才更新,等于被选择的VOQ端口号加1 (mod N),否则 r_i 不变.

Step2:许可 每个输出端仲裁器有一个最高优先级许可指针 g_i.如果一个输出端仲裁器接收到一个或多个请求,从 g_i 指向的位置开始,根据轮转优先级规则,许可某个输入端请求,其它的则不予许可.然后通知各个发送请求的输入端是否得到许可,最后更新 g_i,等于当前被许可的输入端口号加1 (mod N).

图2为在4×4的交换机中DRR算法的一次执行过程.

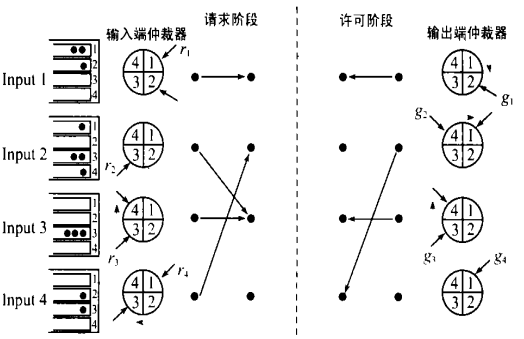


图2 4×4交换机中DRR算法一次执行过程

DRR算法虽然避免了"饿死"现象,但是一次匹配产生的次数较少,没有充分发挥交换网络的效率,如图2中虽然输出端4空闲, Q_{24} 中有信元,但输入端2与输出端4之间也无法建立一条"边", Q_{24} 中的信元只有等待下一次执行过程的仲裁.为了改善性能,我们使用多重迭代的启发式策略,尽可能增加匹配端口数,尽可能快地转发输入队列中的信元.多重迭代的原则是"在每一次迭代过程中,前面迭代中已经匹配的端口将不参与本次仲裁,只在尚未匹配的端口之间寻找新的'边'".从上面DRR算法描述中我们可以看出,在一次执行过程中,某个输入端请求一旦不被许可,其仲裁器指针保持不变,下次迭代也不会被许可.因此DRR算法不能直接多重迭代.下面我们对DRR算法的Step1稍作修改,允许多重迭代运行,本文称改进后的算法为iDRR算法(i表示迭代次数),也就是说,iDRR算法一次执行过程包括 i 次迭代.

为每一个输入端仲裁器增加一个辅助指针 e_i,每次迭代

后按 $e_i = (e_i + 1) \bmod N$ 更新,输入端请求根据 e_i 仲裁. 下面给出 iDRR 算法描述:

Step1 :请求　每个输入端仲裁器有一个请求指针 r_i 和一个辅助指针 e_i, r_i 指向当前最高优先级请求发送的 VOQ,当一次执行过程开始时 $e_i = r_i$. 从 e_i 指向位置开始,根据轮转优先级规则,输入端选择紧接着的 VOQ 发送请求. e_i 每次迭代后都更新[$e_i = (e_i + 1) \bmod N$], r_i 当且仅当该请求在第一次迭代的 Step2 中被许可才更新,否则 r_i 不变.

Step2 :许可　同 DRR 算法.

3　iDRR 性能分析和仿真结果

3.1　均匀分布业务流

本节主要分析 iDRR 算法在信元按均匀独立同分布的贝努里过程到达情况下的时延性能. 当 i 取不同值时,一个 16 × 16 iDRR 交换机的信元平均时延的比较如图 3 所示,2-DRR 与 4-DRR 的性能几乎相同,在负载较轻时 2-DRR 和 4-DRR 的性能明显优于 DRR. 另外,从图 3 中可以看出,当负载接近 1 时, iDRR 算法仍能正常工作,表明能获得 100 %吞吐量.

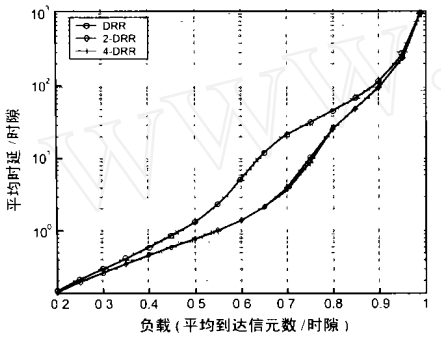


图 3　当 i = 1,2,4 时 16 ×16 iDRR 变换机平均时延

在重载下,DRR 算法中每个 VOQ 的请求得到许可只需要等待 N 个时隙[7],iDRR 算法情况类似,所以在图 3 中当 0.8 < λ < 1 时,iDRR 算法增加迭代次数也不能很好改善时延特性(因此下文中我们只讨论 2-DRR 算法的性能).

图 4 给出了 16 ×16 2-DRR 交换机在突发流到达情况下的性能. 突发流模型是 on-off 马尔可夫过程[5],注意到交换机平均时延性能下降,且平均时延大致与突发长度成正比.

3.2　非均匀分布业务流

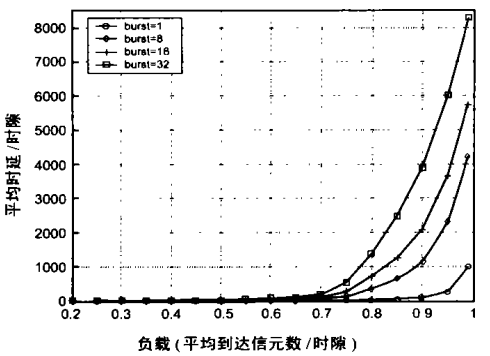


图 4　16 ×16 2-DRR 变换机在不同突发流长度下的平均时延比较

本节主要分析 iDRR 算法在一种特殊的非均匀业务流 —hot-spot[7]业务流 —到达下的时延性能. hot-spot 业务流:所有输入端负载相同,即 $\lambda_i = \lambda$,但各个 VOQ 的到达是非均匀的,某个 VOQ 信元到达平均速率是其它 VOQ 的两倍,而其它 VOQ 是均匀分布的,那个特殊的 VOQ 对应的输出端被称为 hot-spot,这种业务流被称为 hot-spot 业务流. 对于一个 $N \times N$ 交换机,假设 hot-spot 为输出端 1,则有:

$$\lambda_{ij} = \begin{cases} 2\lambda/(N+1), & j = 1 \\ \lambda/(N+1), & j \neq 1 \end{cases}, \quad 1 \leq i, j \leq N \qquad (2)$$

如果取 N = 16,只有当 $\lambda \leq 17/32 \approx 0.53$ 时,输出端 1 才不会超载,即业务流是可接受的.

图 5 比较了 16 ×16 2-DRR 交换机在独立同分布和 hot-spot 业务流下的时延性能. 在非均匀业务流到达下,时延性能明显差于均匀业务流到达下的性能.

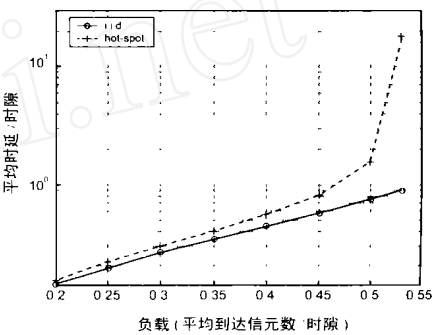


图 5　16 ×16 2-DRR 交换机在 hot-spot 业务流到达下的平均时延

3.3　公平性

crossbar 控制算法的公平性是指保证输入队列的发送请求必须在有限的时间内得到许可,即不能产生"饿死"现象,否则算法就是不公平的. 如果算法是公平的,并且所有发送请求等待许可的时间越平均,那么说明算法的公平性越好. 我们知道,DRR 算法是公平的[7].

定理 1　对于 $N \times N$ iDRR 交换机,信元在队头最长等待时延为 N^2 个时隙.

证明　设某个信元在第 k 个时隙到达某个 Q_{ij} 的队头,它的请求被输入端选择发出至多需要等待指针 r_i 更新 N 次,而该输入端的每一个请求被输出端许可至多需要等待 N 个时隙,因此,输入端仲裁器指针 r_i 更新一次至多需要 N 个时隙. 所以,该信元的请求被发出并被许可至多需要等待 N^2 个时隙.　　■证毕.

定理 1 说明 iDRR 算法是公平的,不会产生"饿死"现象. 在此前提下,信元平均时延的均方差可以衡量算法的公平性的好坏,均方差越小,说明信元通过交换机时延抖动越小,公平性越好[10]. 图 6 比较了 16 ×16 DRR 和 2-DRR 交换机在不同突发流情况下的时延均方差,结果表明 2-DRR 算法的公平性始终比 DRR 算法好,并且当突发长度越大,2-DRR 算法的公平性更加明显.

4　基于 iDRR 算法的仲裁器的实现

iDRR 交换机包括输入端和输出端仲裁器,两类仲裁器都

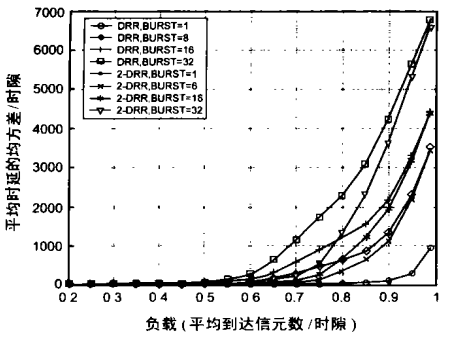


图 6 16 ×16 DRR 和 2-DRR 交换机在不同突发流长度下的时延均方差比较

是轮转优先级仲裁器,结构基本一致,如图 7 所示. 仲裁器主要由一个可编程优先级编码器(*PPE: Programmable Priority Encoder*)和一些状态寄存器组成,*PPE* 是仲裁器设计的关键,它的输入 R_i 表示请求矢量,e_i, r_i 或 g_i 表示 iDRR 算法中的当前最高优先级. 仲裁器工作时,PPE 根据当前最高优先级从 R_i 中选出请求,将结果输出到 G_i,另一个输出 f(1-bit)表示是否有请求被许可. j 表示是否为第一次迭代. 从硬件结构而言,基于 iDRR 算法的仲裁器只比基于 DRR 算法的仲裁器增加了输入 j 和简单的逻辑电路,增加的代价微乎其微. 在输入端仲裁器中,当 $j=1$ 时,$e_i = r_i$; f 和 j 同时为"1"时,r_i 才更新. 在输出仲裁器中,j 始终取 1,只要 $f=1$ 时,g_i 就更新.

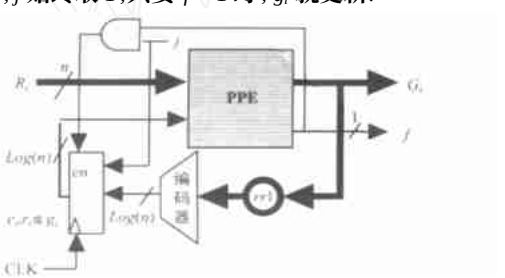


图 7 基于 iDRR 算法的仲裁器

我们使用 xilinx Spartan 系列 FPGA 实现了基于 iDRR 算法的仲裁器,选用了性能优良的温度计编码型 PPE[8,9]. 仲裁器一次迭代由请求、许可和配置 crossbar 三个步骤组成. 这三个步骤可以按"流水线"方式执行,即第 k 次迭代的请求、第 $k-1$ 次迭代的许可和第 $k-2$ 次迭代的配置 crossbar 可以在同一个时钟节拍内完成. 如果仲裁器一次执行过程包含 i 次迭代,那么 m 次执行过程需要 $m \times i + 2$ 个时钟节拍,即平均每次执行过程需要 i 个时钟节拍. 对于一个 16 ×16 iDRR 交换机,设仲裁器工作频率为 80MHz, $i=4$,信元长度为 64 字节,那么仲裁器一次执行过程的时间为 4/(80MHz) = 50ns,仲裁器支持的交换机吞吐量为 1/(50ns) ×64 ×8bit ×16 = 163.84Gbps,每个端口可以达到 10Gbps 的容量. 因此,基于 iDRR 算法的仲裁器可用于超高速路由器中.

5 结束语

本文介绍了 DRR 算法并提出了改进后的可多重迭代的 iDRR 算法. 通过仿真比较了两个算法的时延性能,以及 iDRR 算法在均匀和非均匀分布业务流到达情况下的时延性能,此外还证明了 iDRR 算法的公平性. 最后使用 xilinx spartan 系列 FPGA 实现了基于 iDRR 算法的仲裁器. 所有结果表明,iDRR 算法不但继承了 DRR 算法公平、有效、可扩展性强以及硬件实现简单等特点,而且在时延性能、公平性方面有较大改进,可用于超高速、大容量的路由器中.

参考文献:

[1] Nick Mckeown et al. The tiny tera: A packet switch core[J]. IEEE Micro Magazine, 1997, 17: 26 - 33.

[2] M J Karol, M Hluchyj and S Morgan. Input versus output queuing on a space-division packet switch[J]. IEEE Trans. On Communications, 1987, 35: 1347 - 1356.

[3] Cisco Inc. Cisco 12000 series---Internet Router. Product Overview [Z]. http://www.cisco.com, Oct 2001.

[4] Partridge C et al. A 50-Gb/s IP router[J]. IEEE Trans. on Networking, 1998, 6: 237 - 248.

[5] N Mckeown. Scheduling algorithm for input-queued cell switches[D]. Ph. D. Thesis, UC Berkeley, May 1995.

[6] H J Chao and J S Park. Centralized contention resolution schemes for a large-capacity optical ATM switch[A]. Proc. IEEE ATM Workshop, Fairfax, VA. USA[C]. 1998. 11 - 16.

[7] Yihan Li, Shivendra Panwar, H Jonathan Chao. On the performance of a dual round-robin switch[A]. IEEE INFOCOM'2001[C]. AK, USA, 2001. 1688 - 1697.

[8] 彭来献,田畅,郑少仁. 基于 iSLIP 算法的高速交叉矩阵调度器的 FPGA 设计与实现[A]. 第十五届南京地区研究生通信年会论文集[C]. 中国南京, 2000. 475 - 480.

[9] Pankaj Gupta, Nick Mckeown. Design and Implementation of a fast crossbar scheduler[J]. IEEE Micro Magazine, 1999, 19: 20 - 28.

[10] 孙志刚,苏金树,卢锡城. 高效的 crossbar 仲裁算法——ISP[J]. 计算机学报, 2000, 23: 1078 - 1082.

作者简介:

彭来献 男,1978 年 3 月出生于安徽省阜阳,1999 年于解放军通信工程学院获工学学士,同年在该院以硕博连读的方式继续深造,现在攻读通信与信息系统专业博士学位,主要研究领域为高速路由器体系结构和高速交换网络.

田 畅 男,1963 年 2 月出生于山东省青岛市,副教授,2001 年于解放军理工大学获通信与信息系统专业博士学位,中国电子学会高级会员,主要从事宽带交换技术、网络安全和无线分组网的研究,在国内外有关刊物、会议发表论文 30 余篇.

郑少仁 男,1939 年生于安徽省绩溪县,教授,博士生导师,现为解放军理工大学全军交换技术与 ATM 研究中心主任,中国通信学会会士,中国电子学会会士,长期从事程控交换与宽带通信网方面的教学与研究工作.