

高速信元交换调度算法研究

戴礼森, 洪佩琳, 李津生

(中国科技大学电子工程与信息科学系, 合肥 230027)

摘要: 输入缓存交换结构的特点是缓存器和交换结构的运行速率与端口速率相等、实现容易, 但存在队头阻塞(HOL), 其吞吐率只有约58%。采用虚拟输出排队方法(VOQ)和适当的信元调度算法可消除HOL, 使吞吐率达到100%。本文通过仿真对几种调度算法: PIM、iSLIP和LPF进行了全面地研究、比较和评价。

关键词: 交换结构; 路由器; 排队系统

中图分类号: TP393 **文献标识码:** A **文章编号:** 0372-2112(2000)05-096-03

Study of Scheduling Algorithms for High Bandwidth Cell Switches

DAI Lirsen, HONG Peilin, LI Jirsheng

(University of Science and Technology of China, Hefei 230027, China)

Abstract: Input queued cell switch has the advantage of same bandwidth of buffer and switching architecture as that of inputs and outputs. But the head of line blocking in input queued switch limits its throughput to approximately 58%. If VOQ queueing policies are used with some cell scheduling algorithms, the 100% throughput can be achieved. This paper quantitatively evaluates and compares the performance of three scheduling algorithms: PIM, iSLIP and LPF.

Key words: switching fabric; router; queueing system

1 引言

传统路由器均采用共享总线和输出缓冲方式^[1], 但由于总线和存储器的速率必须为端口线速的N倍(N为端口数), 当端口速率和端口数量增加时实现困难。而输入缓冲无阻塞交换结构中存储器和交换结构速率与端口线速相等, 非常适用于高速和大容量交换。因此, 近年来许多高速交换机和路由器均倾向于采用输入缓冲无阻塞交换结构, 并将变长分组在输入端截成定长的、称作信元的短分组, 通过交换结构后恢复成原数据。采用信元交换可大大简化交换结构和调度算法的设计^[4]。然而, 采用单个FIFO(先进先出)的输入缓冲交换结构存在队头(HOL)阻塞, 使其吞吐率只有58.6%^[2]。如果在每个输入端口放置N个FIFO缓冲队列, 每个FIFO对应一个输出端口, 通过调度算法将无冲突信元交换到输出端口, 就可消除HOL。本文将对几种最新的调度算法: PIM(Parallel Iterative Matching)^[1]、iSLIP(Iterative Round Robin Matching with SLIP)^[3]和LPF(Longest Post First)^[4]进行比较研究。

2 算法模型

图1是VOQ调度算法模型。图中 $A_i(t)$ 表示信元到达输入端口 i 的离散过程。任一信元时隙最多有一个信元到达输入端口 i , 且此信元的目的端口确定为 $j(1 \leq j \leq N)$, 该信元被放入输入端口 i 的 j 缓冲队列 $Q(i, j)$ (其长度用 $L(i, j)$ 表示)。整个信元到达过程 $A(t) = \{A_i(t); 1 \leq i \leq N\}$ 必须是可接

受的, 也就是说每个端口输入和输出的总速率不超过其物理端口的带宽, 即 $\sum_i \lambda_{i,j} < 1, \sum_j \lambda_{i,j} < 1$ 。

调度算法所寻求的结果就是在一个信元时隙内通过多次迭代达到输入和输出端口的最优匹配, 从而使吞吐率接近100%。调度算法实质上是二分图的匹配问题(见图2), 图 $G = [V, E]$, V 为顶点集($I, J, |I| = |J| = N$), E 为边集。匹配的结果使得 M 是 E 的子集, 而且任意两个顶点之间最多只有一个边。加权值 $w_{i,j}$ 在不同的算法中有不同的含义, 在PIM和iSLIP中 $w_{i,j}$ 总是等于0或1, 表示队列 $Q(i, j)$ 是否有信元输出; 而在LPF算法中 $w_{i,j}$ 则表示输入端口缓冲队列长度以及信元竞争输出的拥塞程度。

2.1 PIM算法

PIM算法由文献[1]提出, 其基本思想是在信元时隙通过多步迭代在输入和输出端口之间匹配尽可能多的发送通道, 并采用随机数防止仲裁的不公正性。在信元时隙开始, 所有的输入和输出端口状态都标为空闲, 每一步迭代之后都有一部分输入和输出端口被匹配上。每一步迭代中都只考虑未匹配上的端口, 每次迭代包含以下三个步骤: (1)请求: 哪个输入端口FIFO非空, 有信元要输出, 该端口就向相应的输出端口提出发送请求。(2)授权: 如果尚未匹配的输出口收到发送请求, 它就在同时向它提出申请的多个输入端口中均匀地、随机地选择一个予以接受并对其授权。(3)接受: 如果一个输入端

口得到授权后(可能同时得到多个输出端口的授权),就从授权的端口中选择一个作为发送目标端口。

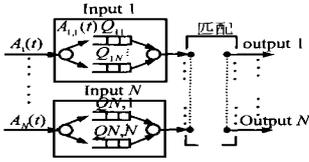


图1 VOQ 交换结构模型

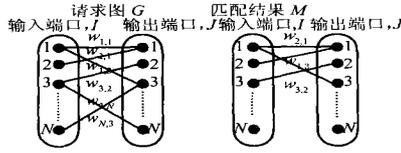


图2 VOQ 信元调度算法的二分图模型

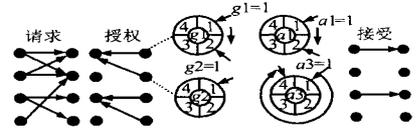


图3 iSLIP 算法的三个步骤

由于每次迭代只考虑上次迭代中未匹配的端口,所以迭代的结果不会和以前已经匹配的通道重复或冲突。算法的第二步:在多个请求的输入端口中随机地选择一个予以授权,可以确保所有的发送请求总能得到授权。

2.2 iSLIP 算法

和PIM 算法一样,iSLIP 也是一种迭代算法,它克服了PIM 算法的复杂和不公平性。每次迭代也有三个步骤:(1)请求:与PIM 算法步骤1 同。(2)授权:如果尚未匹配的输出口收到发送请求,它就按照循环优先级方式选择一个输入端口并通知该输入端口是否被许可。当且仅当输入端口接受该授权时,最高优先级指针 g_i 才加 1(以端口数 N 为模),指向获得授权端口的下一个端口。而刚被授权且接受授权的输入端口在下一时隙其优先级将为最低。(3)接受:当一个输入端口得到授权后(可能同时得到多个输出端口的授权),它就按照循环优先级方式选择一个,并使最高优先级指针 a_i 为 1(以端口数 N 为模)指向接受输出端口的下一个端口。

图 3 示出了一次迭代的三个步骤。图中输入端口 1 有信元分别指向输出端口 1 和 2,输入端口 2 也有信元指向输出端口 1 和 2, ... 在信元时隙最后 a_1 和 a_3 分别指向了 2 和 4,由于本次迭代输入端口 1 和输出端口 1、输入端口 3 和输出端口 3 达成了匹配,而输入端口 1 虽然得到了输出端口 2 的授权,但它选择了输出端口 1,所以指针 g_2 没有改变,待下次迭代进行匹配。可见,采用循环优先级指针不仅大大简化了实现的复杂度而且改善了公平性。此次建立起来的通道中的输入端口在下次匹配时将成为优先级最低的端口,而且当一个输入端口的请求未被满足时,它会在下次匹配时继续提出该请求。所以,在输入端口的 N 个缓冲队列中,任何一个队列在 N^2 个时隙内必然得到发送机会。

2.3 LPF 算法

LPF 算法是一种最大加权匹配算法,它考虑到了各个输入端口缓冲队列长度以及信元竞争输出的拥塞程度。其加权值定义为: $w_{i,j}(n) = \begin{cases} R_i(n) + C_j(n), & L_{i,j}(n) > 0 \\ 0, & L_{i,j}(n) = 0 \end{cases}$ 式中 $L_{i,j}(n)$ (n) 是第 n 时隙队列 $Q_{i,j}$ 的长度。 $R_i(n) = \sum_j L_{i,j}(n)$ 表示输入端口 i 在第 n 时隙所有缓冲队列总和; $C_j(n) = \sum_i L_{i,j}(n)$ 表示所有要发送到输出端口 j 的信元缓冲队列长度总和。令 $S_{i,j}(n)$ 表示缓冲队列 $Q_{i,j}$ 得到发送的机会(得到发送机会, $S_{i,j}(n) = 1$; 否则, $S_{i,j}(n) = 0$)。LPF 算法就是寻求这样的匹配:在条件 $\sum_{i=1}^N S_{i,j}(n) \leq 1$ 和 $\sum_{j=1}^N S_{i,j}(n) \leq 1$ 下,使得 $\sum_{i,j} S_{i,j}(n) w_{i,j}(n)$

最大。

3 性能分析

为了对上述三种算法的性能进行比较,我们通过计算机仿真对采用上述算法的 VOQ 交换结构(16×16)进行了仿真分析,业务量模型分为随机均匀业务和突发均匀业务两种情况。

3.1 均匀业务量

设各输入端信元到达过程为独立同分布 Bernoulli 过程,且为均匀业务分布。图 4 示出了采用 PIM、iSLIP 和 LPF 算法的平均排队时延与业务负荷的仿真结果。图中 PIM1 和 PIM4 分别表示采用 1 次和 4 次迭代的 PIM 算法;iSLIP1 和 iSLIP4 分别表示采用 1 次和 4 次迭代的 iSLIP 算法。从图中可以看出,采用一次迭代的 PIM 其吞吐率只有 60% 多一点,而采用一次迭代的 iSLIP 算法,虽排队时延仍比较大,但吞吐率已接近 100%。4 次迭代的 PIM 和 iSLIP 算法几乎一样,具有非常理想的排队时延性能。而 LPF 算法在输入负荷大于 0.8 以上时平均时延大于 iSLIP 和 PIM。

3.2 突发业务量

我们采用 ON/OFF 业务模型,平均突发长度为 20 个信元,一个突发期内信元指向同一输出端口。图 5 示出了三种算法的仿真结果,由图可见,在突发业务量下所有算法的性能均变差,PIM 和 iSLIP(均为 4 次迭代)性能基本相当,而当输入负荷大于 0.95 时 LPF 算法性能优于其余两种算法。

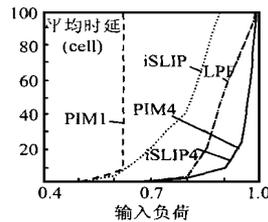


图4 均匀业务量下输入负荷与平均排队时延关系

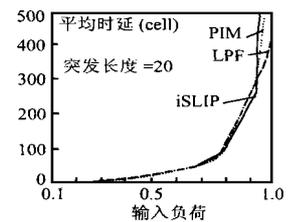


图5 突发业务量下输入负荷与平均排队时延关系

3.3 非均匀业务量

要对所有的非均匀业务进行仿真是困难的,但我们可以以一个例子来说明象 PIM 和 iSLIP 一类的最大匹配算法在非均匀业务量下具有不稳定性,也就是说其吞吐率将达不到 100%。如图 6 所示,各个信元到达率均为 $\frac{1}{2} - \delta$ ($\delta > 0$),设在时刻 n , $A_{2,1}(n)$ 和 $A_{3,2}(n)$ 均有信元到达(概率为 $(\frac{1}{2} - \delta)^2$),

且 $L_{1,1}(n) > 0, L_{1,2}(n) > 0$, 则输入端口 1 得到授权发送的概率为 $2/3$ (因 3 个输入端口的信元均指向输出端口 1 和 2, 所以最大匹配时 3 个输入端口中必有 2 个被授权发送), 因此, 端口

1 可以发送信元的总速率为: $\frac{2}{3}(\frac{1}{2} - \delta)^2 + [1 - (\frac{1}{2} - \delta)^2] = 1 - \frac{1}{3}(\frac{1}{2} - \delta)^2$, 而输入端口 1 的信元到达率为 $1 - 2\delta$, 如果 $1 - 2\delta > 1 - \frac{1}{3}(\frac{1}{2} - \delta)^2$, 即 $\delta < 0.0358$, 则输入端口 1 的信元到达率就

大于可以发送到输出端口的速率, 也就是说吞吐率没有达到 100%. 与 PIM 和 iSLIP 不同, LPF 算法由于它考虑到了各端口的拥塞程度, 并能优先选择长队列的输入端口, 所以可以证明即使在业务量为非均匀时其吞吐率也达到 100%^[4].

4 实现的复杂性

由于 PIM 算法采用随机数, 所以实现比较复杂, 而 iSLIP 算法仅需 $2N$ 个端口的调度器, 而且每个调度器就是一个带优先级的 N 输入端编码器, 所以实现比较简单. 前面描述的 LPF 算法并不便于硬件实现, 为了高速的硬件实现, 必须将 LPF 算法变为迭代形式, 但实现仍较复杂.

表 1

算法	主要特点	吞吐率	非均匀业务量性能	实现复杂性
PIM	随机优先级最大匹配, 多次迭代	单次迭代 63%, 多次迭代 100%	差	复杂
iSLIP	循环优先级最大匹配, 多次迭代	100%	差	简单
LPF	最大加权匹配, 可迭代实现	100%	优	复杂

5 总结

表 1 给出了三种算法的总体评价. 我们在“高速路由器的研制”项目中, 根据上述几种算法的总体性能, 选用了 4 次迭代 iSLIP 算法, 将加以扩展使其支持 4 个优先级.

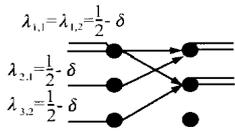


图 6 非均匀业务量时最大匹配算法不稳定的例子 (3x3 交换)

参考文献

- [1] Anderson, T., et al. High speed switch scheduling for local area network. ACM Trans on Computer Systems, Nov. 1993: 319~ 352
- [2] Karol M. J., et al. Input versus output queueing on a space division packet switch. IEEE Trans. Commun, 1987, 35: 1347~ 1356
- [3] McKeown N. Fast Switched Backplane for a Gigabit Switched Router. <http://tinytera.stanford.edu/~nickm/papers>
- [4] Adisak Mekittikul, et al. A practical scheduling Algorithm to achieve 100% throughput in input queued switches. IEEE INFOCOM 98, San Francisco, April 1998



戴礼森 1962 年生, 1988 年毕业于南京邮电学院, 1989 年至 1996 年在安徽省邮电研究所从事通信产品研究开发, 现为中国科技大学电子工程与信息科学系博士生. 主要研究方向: 宽带交换技术.



洪佩琳 1986 年毕业于中国科技大学电子工程与信息科学系, 工学硕士. 现为中国科技大学电子工程与信息科学系副研究员. 从事信号处理、语音通信和网络技术等方面的研究. 发表论文十余篇.

李津生 中国科技大学电子工程与信息科学系教授, 博士生导师, 承担过多项“863”攻关项目和国家自然科学基金研究项目. 著作有:《综合业务数字网》、《ISDN&ATM》、《IC 卡实用化技术》和《B-ISDN&ATM-LAN》等.