文章编号:1001 - 9081(2010)01 - 0101 - 03

一种高速 crossbar 调度算法及其性能分析

姜小波,杜小伟
(华南理工大学 电子与信息学院,广州 510641)
(smallwe@ tom. com)

摘　要:分析了高速 crossbar 调度算法 iSLIP 在处理突发业务时性能严重恶化的原因。结合 LQF/iLQF 算法的思想,提出了又一种输入排队 crossbar 调度算法 iPGQM。仿真结果表明:该调度算法在均匀业务流量下和 iSLIP 算法的性能基本相同;在突发业务的条件下,iPGQM 算法具有更好的抗突发特性;特别在重负载的条件下,与 iSLIP 算法相比,不仅具有更高的吞吐量,而且平均延迟降低了 10% 左右。

关键词:crossbar;调度算法;输入排队;非均匀业务流;iSLIP
中图分类号: TP393　　**文献标志码**:A

Novel scheduling algorithm in high-speed crossbar and its performance analysis

JIANG Xiao-bo, DU Xiao-wei
(*School of Electronic and Information, South China University of Technology, Guangzhou Guangdong* 510641, *China*)

Abstract: This paper analyzed the reasons for which the high-speed crossbar scheduling algorithm iSLIP has a serious deterioration of performance under burst traffics. With reference to the ideas of LQF/iLQF, this paper proposed a novel input-queued crossbar scheduling algorithm called iPGQM (iterative Parallel Graded-Length Queue Matching). Simulation results show that iPGQM has the same performance as iSLIP under uniform traffics. Furthermore it has better performance than iSLIP under burst traffics. Especially in heavy load conditions, iPGQM not only achieves higher throughput, but also reduces the average delay about 10% compared with iSLIP algorithm.

Key words: crossbar; scheduling algorithm; input queueing; nonuniform traffic; iSLIP

0　引言

输入排队(Input Queuing, IQ)crossbar 作为一种简单、高效的高速交换结构,被广泛应用于高速路由器中[1]。在这种交换结构中,如果输入队列只采用简单的先进先出(First In First Out, FIFO)队列机制,其线头阻塞(Head Of Line blocking,HOL)将限制交换网络的最大吞吐量为 58.6%[2-4]。因而,一般采用虚拟输出排队技术(Virtual Output Queueing, VOQ),即一个输入端为每一个输出端维护一个 FIFO 队列。由于采用了 VOQ,还必须使用有效的 crossbar 控制算法解决输入/输出端的竞争,保证无冲突地传输信元[5]。基于输入排队调度算法中 iSLIP(iterative SLIP)算法以其高吞吐量和低复杂度得到了广泛的应用,但 iSLIP[6](iterative SLIP)算法在突发流量下性能严重下降。为此,又提出了以队列长度为权重的调度算法[7-8],如 LQF 算法,该算法对任何容许的流量都是稳定的。但硬件实现较为复杂,目前应用较少。

本文首先分析了高速 crossbar 调度算法 iSLIP 在处理突发业务时性能严重恶化的原因,并结合 LQF/iLQF 算法的思想,提出一种简单、硬件易实现的调度算法 iPGQM(iterative Parallel Graded-Length Queue Matching),该算法根据 VOQ 队列的长度来调节提请求的机会,从而增加负载高的输入队列的匹配机会。结果表明,该算法能够明显改善高速 crossbar 调度算法在非均匀业务流下的吞吐量和时延等性能。

1　问题描述

1.1　输入排队交换和匹配问题

crossbar 交换结构是输入排队的典型实现方式,这种结构实质上相当于一个多总线的结构,支持多个点到点的链路同时进行通信,一般用于高性能交换。

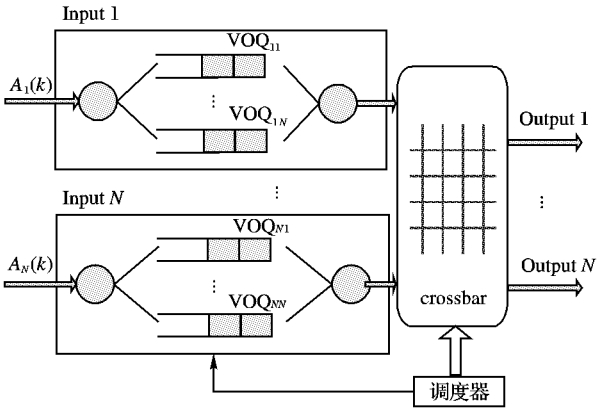

图 1　IQ crossbar 逻辑结构

图 1 所示是一个基于输入排队的 $N \times N$ 规模的 crossbar 交换结构:每个输入端口的缓存分为 N 个 VOQ 队列,每个 VOQ 队列存储从输入端口 i 到达,目的端口为 j 的信元。输入端总共有 N^2 个 FIFO,构成交换网络的队列系统。图中调度器用于管理队列信息,执行调度算法。调度算法首先根据输入队列的状态,做出匹配结果,然后控制交换结构中交叉点的开合,最

收稿日期:2009 - 07 - 13 ;**修回日期**:2009 - 08 - 18。

作者简介:姜小波(1972 -),男,副教授,博士,主要研究方向:通信 IC 设计、低功耗技术;　杜小伟(1984 -),男,湖北枣阳人,硕士研究生,主要研究方向:高速交换体系及其调度算法。

终建立输入/输出端信元传输通道。

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 $A_i(n)$ 表示，$1 \leqslant i \leqslant N$，$n$ 指时隙数，每个时隙在每个输入端有 0 或者 1 个信元到达。到达输入端 i 且目的端为 j 的信元被存储在 VOQ_{ij} 中，信元平均到达 VOQ_{ij} 的速率为 λ_{ij}，即每个时隙平均到达的信元数。

定义 1 如果 λ_{ij} 满足约束条件：

$$\begin{cases} \sum_{j=1}^{N} \lambda_{ij} \leqslant 1; \quad 1 \leqslant i \leqslant N \\ \sum_{i=1}^{N} \lambda_{ij} \leqslant 1; \quad 1 \leqslant j \leqslant N \end{cases} \tag{1}$$

则称 $A_i(k)$ 是容许的，否则就是非容许的。

由于 crossbar 无内部存储、无阻塞，为了避免发送和接收冲突，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。使用 $m_{ij}(n)$ 表示第 n 个时隙时输入端口 i 与输出端口 j 的连接关系。可以将 crossbar 结构的连接约束描述如下：

$$\begin{cases} \sum_{j=1}^{N} m_{ij} \leqslant 1; \quad 1 \leqslant i \leqslant N \\ \sum_{i=1}^{N} m_{ij} \leqslant 1; \quad 1 \leqslant j \leqslant N \end{cases} \tag{2}$$

1.2 iSLIP 和 LQF/iLQF 算法分析

调度算法按照一定的规则决定输入端和输出端的匹配关系，解决信元在输入端和输出端的竞争。路由器能达到的吞吐率要求和延时特性取决于其所采用的调度算法。具有代表性的如 PIM、iSLIP、LQF 和 OCF 等[9]。这四种算法的特性比较如表 1。

表 1 四种典型算法的特性比较

算法	稳定性	公平性	延迟控制	复杂度
PIM	差	较好	差	低
iSLIP	差	好	差	低
LQF	好	较好	好	高
OCF	好	较好	好	高

由于 PIM、iSLIP 等算法的低复杂度，易于硬件实现。iSLIP 被应用于 CISCO 的 GSR12000[10] 系列高速路由器中，但这类算法在突发流量情况下，性能会急剧下降[6]。缺乏 LQF、OCF 等算法的稳定性，但 LQF、OCF 算法的复杂度又较高。

iSLIP 算法只需要 $\log N$ 次迭代就能收敛[6]，并且 iSLIP 算法在均匀业务流下能够达到 100% 吞吐量，但对于突发业务，该算法的平均时延迅速增大，在高负载条件下出现大量丢包。原因在于 iSLIP 调度算法采用 round-robin 的方式调度输入队列中的信元，这样虽然对于每个输入端内的 VOQ 得到的服务是 max-min 公平的，但是无法对不同的队长，不同等待时间的数据包做出合理的区别调度。当到达信元分布不均匀时，round-robin 服务致使队列长度变化不均衡，负载较大的 VOQ 的队列长度容易持续增长，从而限制了吞吐量，进而使延迟和丢包率增大。

LQF 算法是一种最大权重匹配算法，它把 VOQ_{ij} 长度 L_{ij} 设为权重 ω_{ij}，优先服务有较高 VOQ 长度的输入。对于独立到达的流量 LQF 算法可获得 100% 的吞吐量，对于任何容许流量该算法都是稳定的。iLQF 算法[11] 作为 LQF 算法的迭代近似算法也包括请求、响应、确认三步。iLQF 在输入端向相应的输出端发送请求的同时也发送 VOQ 长度，这往往需要更多的参数编码比特数和更大的时间复杂度，由于现有电子器件很难在几十甚至几纳秒内为 G 比特级端口速率的 Scheduler 做出调度决策，从而限制了这些最大权重算法的使用。

2 iPGQM 算法

2.1 算法思想

传统 crossbar 调度算法输入/输出端竞争采用 round-robin 方式来解决，同等对待各个有请求的输入端口。在突发流量条件下，这种策略会使队列的长度变化不均衡，从而限制了吞吐量，最终会降低整个交换网络性能。

为了提高算法在非均匀流量下适应性，需要考虑到达各输入端口负载的不均匀性，增加负载较重端口的匹配机会。本文提出的调度算法 iPGQM，通过增加重负载端口提请求机会的策略，来保证长队列得到较多的服务机会。从而提高在突发业务流下的吞吐量和时延性能。

增加重负载 VOQ 匹配机会的策略为：根据迭代次数 n，为去往相同目的端口的 VOQ 设置 n 个门限值 $th_1, th_2, \cdots, th_n$（$th_1 \geqslant th_2 \geqslant \cdots \geqslant th_n$），第 k 次迭代允许长度超过 th_k 的 VOQ_{ij}（并且输入端 i 和输出端 j 都未匹配）向相应的输出端提请求。由于 $th_k \geqslant th_{k+1}$，如果该 VOQ_{ij} 在第 k 次迭代中没有得到匹配机会，可以在第 $k+1$ 次迭代中继续参与竞争。这就使那些负载较重的队列获得了更多的发送机会，从而实现根据输入流量的非均匀性区别对待各个端口。

2.2 iPGQM 算法描述

iPGQM 算法同样分为请求→响应→接受三个步骤。根据迭代次数 n，为所有去往输出端口 j 的 VOQ 设置 n 个门限值 $th_{j,1}, th_{j,2}, \cdots, th_{j,n}$（$th_{j,1} \geqslant th_{j,2} \geqslant \cdots \geqslant th_{j,n}$）。只有长度大于门限值的 VOQ 才会提请求，把请求作为输入，经过 n 次迭代则得到一个新的匹配。

第 k 次迭代的三个步骤为：

步骤 1 请求。如果一个未匹配的输入端 i 有信元等待发送，根据要求发送的输出端口号 j，如果相应的 VOQ_{ij} 大于 $th_{j,k}$，则向输出端 j 发送请求信号。

步骤 2 许可。如果一个未匹配的输出端接收到多个请求信号，从中随机选取一个进行许可。

步骤 3 接受。如果输入端接收到多个许可信号，从中随机选取一个进行接受，这样就建立了一个匹配边。

3 性能仿真

对于输入排队 crossbar 调度算法的性能分析，由于解析分析的困难性，计算机仿真是一种有效而广泛采用的研究手段。

仿真条件是针对 32 × 32 的 crossbar，使用 Bernoulli 业务源模拟均匀业务源和基于 Markov 过程调制的 ON-OFF 过程模拟突发业务源。所有输入端口的输入 buffer 最大缓冲 1 万个 cell，每个 VOQ 最多缓存 500 个 cell。因为 $N = 32$，所以为去往相同输出端口号的 VOQ 设置 5 个门限，分别为：th，$3/4th$，$1/2th$，$1/4th$ 和 0，其中 th 等于去往相同输出端口的 VOQ 长度和除以 N。仿真长度均为 100 000 个时隙。

3.1 Bernoulli 业务

首先针对所有输入端口使用 Bernoulli 业务源，每个信元的目的端口在各个输出端口中等概率选择。对算法 iPGQM

和 iSLIP 进行了仿真,结果如图 2 所示,横坐标为负载 λ,纵坐标为单位信元的平均时延 t。图 2 显示 iPGQM 算法的平均时延基本上与 iSLIP 算法相同。在这种业务下两种算法都表现出极好的性能,系统时延低并且没有出现丢包。这表明 iPGQM 算法在普通业务条件下性能与 iSLIP 算法基本相同。

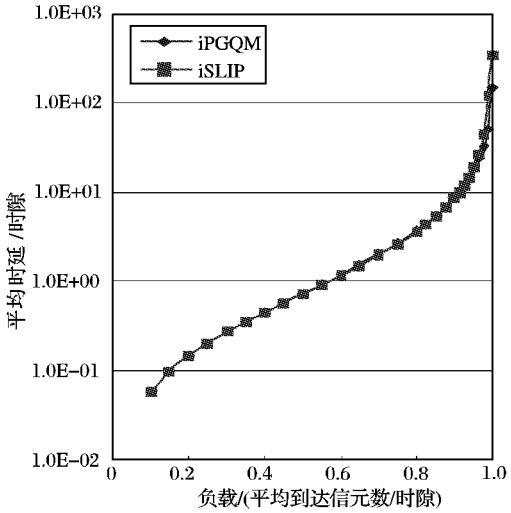

图 2　iPGQM 与 iSLIP 算法在 Bernoulli 业务下的平均时延

3.2　突发业务

以上我们都假设信元按照 Bernoulli 过程到达的,现实中分组往往突发到达,并且每个分组包含多个交换的信元,因此信元突发到达更加接近网络真实情况,通常用基于 Markov 过程调制的 ON-OFF 过程来模拟这种流量。在每段突发过程中,数据包的目的地址不变,每段突发数据包的目的端口在各个输出端口中等概率选择。下面我们考虑当所有端口业务源均采用突发业务的条件下,平均突发长度变化对 iPGQM 以及 iSLIP 算法的影响。这里取递增的 32、64、128 三种突发长度。

图 3 反映了时延随负载增加和突发长度增加的变化情况,可以看出随着流量负载的增加,iPGQM 算法的平均延迟曲线上升较缓慢。特别在流量负载大于 0.7 的情况下,平均延迟明显低于 iSLIP 算法。

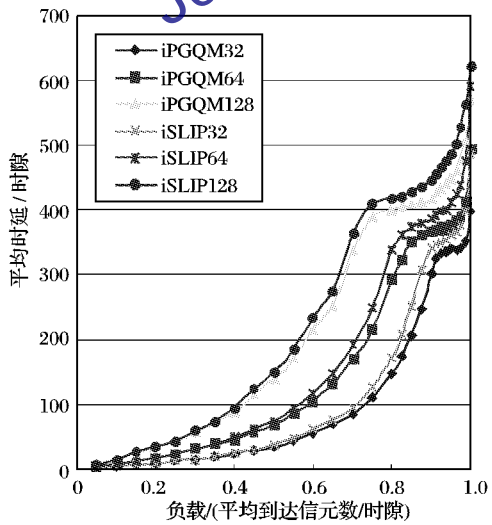

图 3　iPGQM 与 iSLIP 算法在突发业务下的平均时延

图 4 为这两种算法丢包率的对比情况。可以看出 iPGQM 算法的丢包率在三种突发长度的情况下均显著低于 iSLIP 算法的丢包率。这是因为 iPGQM 算法中充分考虑了信元到达信息从而增大了 crossbar 的通过率,从而使平均延迟和丢包率减小,因而具有更好的流量适应性。比 iSLIP 算法相比,不仅大大降低了丢包率,而且平均时延减少了 10% 左右。

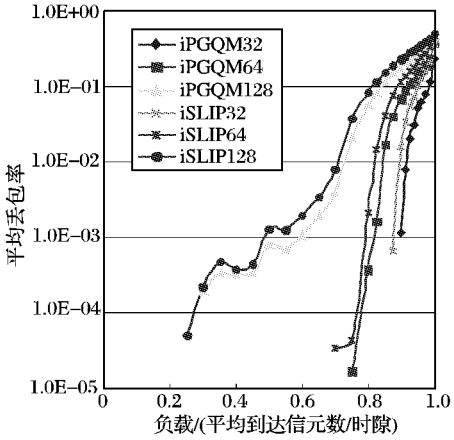

图 4　iPGQM 与 iSLIP 算法在突发业务下的平均丢包率

综上可以看出在 Bernoulli 业务下,iPGQM 算法和 iSLIP 算法具有同样的性能。在突发业务的条件下,iPGQM 算法比 iSLIP 算法具有更低的时延和丢包率。特别是在重负载的条件下,与 iSLIP 算法相比,性能有了显著提升。

4　结语

随着高速路由器的发展,传统的调度方案由于性能或实现复杂度的原因,逐渐成为系统性能的瓶颈。本文提出了一种高性能输入排队 crossbar 调度算法 iPGQM。仿真结果表明,该调度算法在均匀业务流量和突发流量的条件下均具有较好的性能,因而可以用于骨干网核心路由器中。

参考文献:

[1] PARTRIDGE C, CARVEY P, BURGESS E, *et al*. A 50Gb/s IP router[J]. IEEE/ACM Transactions on Networking, 1998, 6(3): 237 – 248.

[2] BONUCCELLI M A, URPI A. A multicast FCFS output queued switch without speedup[C]// Proceedings of 2nd IFIP Conference in Networking. London: Springer-Verlag, 2002: 1057 – 1068.

[3] PRAKASH A, SHARIF S, AZIZ S. An O($\log_2$ N) parallel algorithm for output queuing[EB/OL]. [2009 – 05 – 20]. http://users.ece.utexas.edu/ ~ adnan/publications/sched-infocom-02.pdf.

[4] KAROL M, HLUCHYJ M, MORGAN S. Input versus output queuing on a space-division packet switch[J]. IEEE Transactions on Communications, 1987, 35(12): 1347 – 1356.

[5] MHAMDI L. A partially buffered crossbar packet switching architecture and its scheduling[EB/OL]. [2009 – 05 – 20]. http://ce.et.tudelft.nl/publicationfiles/1504_550_Lotfi_ISCC_2008.pdf.

[6] McKEOWN N. Scheduling algorithms for input-queued cell switches [D]. Berkeley, CA, USA: University of California at Berkeley, 1995.

[7] MCKEOWN N, ADISAK M, VENKAT A, *et al*. Achieving 100% throughput in an input-queued switch[J]. IEEE Transactions on Communications, 1999, 47(8): 1260 – 1267.

[8] ADISAK M. Scheduling non-uniform traffic in high speed packet switches and routers[D]. Stanford, CA: Stanford University, 1998.

[9] MNEIMNEH S. Matching from the first iteration: An iterative switching algorithm for an input queued switch[J]. IEEE/ACM Transactions on Networking, 2008, 16(1): 206 – 217.

[10] Cisco Systems. Cisco 12000 Gigabit Switch Router[EB/OL]. [2009 – 04 – 20]. http://www.cisco.com/warp/public/cc/pd/rt/12000/prodlit/gsr_ov.pdf.

[11] McKEOWN N, ANDERSON T E. A quantitative comparison of scheduling algorithms for input-queued switches[J]. Computer Networks and ISDN Systems, 1998, 30(11): 319 – 352.