星上交换系统输入缓存调度算法

张 怡 周 诠 黎 军
(西安空间无线电技术研究所国家级重点实验室 西安 710100)

摘 要：为改善星上交换系统的性能，该文提出了一种新的输入缓存调度算法。该算法基于 Crossbar 交换结构，采用了串行调度思想，在兼顾每个端口公平性的基础上调整了输出端口的仲裁策略，增加了端口匹配的概率。该算法大大减小了调度时延和丢失率。分析与仿真结果表明，该算法在平均调度时延和信元丢失率等方面的性能指标均优于已有算法而且实现复杂度不增加。

关键词：卫星通信；调度算法；输入缓存；串行调度
中图分类号：TN927 **文献标识码**：A **文章编号**：1009-5896(2009)06-1429-04

An Input-Buffer Scheduling Algorithm in Satellite Switching System

Zhang Yi Zhou Quan Li Jun
(*National Key Laboratory, Xi'an Institute of Space Radio Technology, Xi'an* 710100, *China*)

Abstract: In order to improve satellite switching performance, a new scheduling algorithm is proposed. Based on Crossbar switch fabric, the algorithm adopts serial scheduling strategy and adjusts the output port arbitrating strategy for the better equity of every port. Consequently, it increases the matching probability. The algorithm can greatly reduced the scheduling delay and loss rate. The analysis and simulation results show that the proposed algorithm has the better performance than others in average delay and cell loss rate, and has the equivalent complexity.

Key words: Satellite communication; Scheduling algorithm; Input-buffer; Serial scheduling

1 引言

卫星通信是军用和民用一种重要的通信手段，星上交换是卫星通信的发展趋势之一。星上采用交换技术，可进行卫星多波束交换，方便多个地面站间的通信，构成天地一体化的无线通信网络，因此卫星交换技术将会大大促进卫星通信的发展。目前，在国外已有宽带卫星上使用星上交换处理单元[1]，国内也开展了广泛的研究[2]。

为实现快速分组转发，星上交换单元一般采用基于定长信元的体制，交换单元的缓存方式主要有输入缓存、输出缓存和共享缓存[3]。对于一个 $N\times N$ 交换单元，输出缓存中存储器的访问速度是线路速率的 $(N+1)$ 倍；采用共享缓存方式是线路速率的 $2\times N$ 倍；而对于输入缓存方式，存储器的访问速度是线路速率的两倍，与交换单元的规模无关[3]。为增大交换容量，支持更高的线路速率，在卫星交换系统中有必要研究输入缓存的方式。

对于输入缓存中存在的队头阻塞(Head Of Line blocking, HOL)问题[4]，文献[5,6]指出采用虚拟输出排队(VOQ)技术可改善该问题，使吞吐量达到 100%。VOQ 要获得高的性能，关键问题是要设计一个匹配输入和空闲输出端口的高效、公平的调度算法，也就是二分图的匹配问题。因此，研究基于 Crossbar 结构的卫星交换系统时，采用一种性能较好、实现简单的输入缓存调度算法显得犹为重要。

常见的输入缓存调度算法主要有 PIM(Parallel Iteration Matching)和 iSLIP 等。其中 iSLIP 算法已在 Cisco GSR12000 路由器中使用。这些算法并行匹配所有端口，采用多次迭代方式增加输入输出端口的匹配数目，采用 Round Robin 方式保证公平性。上述算法每次迭代需要请求、响应和接受三步，所有输入/输出端口自由竞争所有输出/输入端口，实现复杂，在重负载情况下，算法的性能较差。

串行调度思想[7]采用串行方式消解输入输出端口阻塞，克服了以上的不足。文献[7]对输入串行调度算法(ISP)进行了介绍，并和 iSLIP 算法进行了比较，结果表明串行调度算法重负载时性能优于 iSLIP 算法，实现起来也较为简单。

本文提出的星上交换输入缓存调度算法基于串行调度的思想，该算法在各个输出端口串行处理收到的请求，通过对原算法进一步的改进增加了每个时隙内匹配更多输入输出端口数目的概率，改善了调度算法的调度时延及丢失率等性能。

下面首先介绍了本文提出的改进算法，第 3 节对改进算法进行性能分析；在第 4 节中，通过 OPNET 构造了一个 16

2008-07-18 收到，2008-12-15 改回
国家重点实验室基金(9140C5302010802，9140C5302010702)和国家预研基金(9140A21050107HT5402)资助课题

×16 输入缓存的 Crossbar 交换结构的仿真模型，并对改进算法及其他算法进行仿真、分析和比较；结果表明，本文提出的改进算法在独立同分布贝努里和突发过程到达下都具有较好的平均调度时延及丢失率等性能；第 5 节对该算法的实现进行了分析。

2 一种改进的串行调度算法

星上交换机由于卫星信道时延长且星上芯片资源有限等原因，要求调度算法具有较好的调度时延及丢失率等性能，串行调度算法可避免多个输出端口响应同一输入端口的同步现象，但在一定程度上还无法达到最大匹配，每个时隙里，优先仲裁的端口具有较大的选择权，而可选范围较小的端口较晚仲裁时，该端口得到匹配的概率也就更小了。因此需要对算法进一步改进来增加每个时隙里的匹配数目，改善星上交换机的性能。

本文对原有输出串行调度算法进行了改进，改进算法在对端口仲裁时，优先仲裁收到请求数目最少的输出端口，较晚仲裁请求数目较多的输出端口，这样能够增加每个时隙里匹配更多输入/输出端口的可能性。每个端口仲裁时采用轮询指针的方法，为保证公平性，每次成功匹配后都要更新指针。

本文提出的改进算法主要由以下几个步骤组成：

第 1 步　每个输入端口向它的队列中信元可能到达的输出端口发送请求信号。

第 2 步　对所有收到请求的输出端口进行仲裁。

首先对收到请求数目较少的输出端口进行仲裁，请求数目较多的输出端口较晚仲裁。每个输出端口仲裁时，从轮询指针所指的输入端口开始，选择一个尚未匹配的输入端口，通知每一个输入端口其请求是否被准许，并更新该输出端口处的轮询指针。

完成一个输出端口的仲裁后，重新开始下一个输出端口的仲裁，直到轮询完所有输出端口。

3 性能分析

对于不同的调度算法来说，很难通过建立数学模型来比较其性能，因此本文对改进算法进行性能分析，通过仿真来验证分析结果。

衡量 Crossbar 调度算法性能的主要指标有带宽利用率、调度时延及公平性等。带宽利用率也就是平均每次调度匹配的输入输出端口数与开关总端口数之比。在总端口数一定的情况下，每次调度匹配的输入输出端口数越多，带宽利用率就越高，且信元在缓存队列中逗留的时间短，信元的平均延时也就越短。在缓存队列有限的情况下，每次匹配的端口数目越多，也会降低信元的丢失率。

3.1 仲裁顺序对带宽利用率的影响

首先分析并比较仲裁顺序不同对匹配端口数目的影响。

假设输出端口 i 和 j 分别收到 $k1$, $k2$ 个输入端口的请求，且 $k1>k2$，先匹配端口 j 时，端口 i 得到匹配的概率大于先匹配端口 i 时端口 j 得到匹配的的概率。

证明

(1)若两个输出端口收到的请求中没有重复的输入端口，则按照两种顺序得到的匹配数目相同。

(2)若两个输出端口收到的请求中有 k 个重复的输入端口($k<k1,k2$)

如果先匹配端口选择的输入端口不在这 k 个端口中，那么顺序不影响匹配结果，这里主要考虑先匹配端口在 k 个输入端口中选择的情况。

先匹配端口 i 时，以 $k/k1$ 的概率选择 k 中端口；再匹配端口 j 时，可选择输入端口的数目变为 $k2-1$，在这种情况下端口 j 得到匹配的概率为

$$p1=\frac{k}{k1}\times\frac{k2-1}{k2} \tag{1}$$

同理，先匹配端口 j，再匹配端口 i，端口 i 得到匹配的概率为

$$p2=\frac{k}{k2}\times\frac{k1-1}{k1} \tag{2}$$

由于 $k1>k2$，可以得出：$p2>p1$。　　证毕

通过该证明可知，优先仲裁收到请求数目较少的输出端口得到较多匹配数目的概率较大，也就增加了提高带宽利用率的可能性。

3.2 keepfull 过程下的最大等待时延比较

在 keepfull 到达过程(保证任何时刻、任何 VOQ 队列都非空)下，原串行调度算法中信元在 VOQ 队头等待交换的最大延时等于 N^2，而改进算法中信元在队头等待交换的最大延时等于 N。

证明　设某信元在时刻 t 成为队列 $Q(i,j)$ 的 HOL(Head Of Line)信元：

(1)根据原算法的定义，t 时刻后的 N^2 次仲裁中必有 N 次仲裁首先从端口 j 开始迭代，又由于 Round Robin(轮询)指针只在第 1 次迭代后修改，在这 N 次迭代中，端口 j 的 Round Robin 指针必有一次等于 i，因此该信元在 t 时刻后的 N^2 个时隙内必能交换到端口 j。

(2)在 keepfull 到达过程下，改进算法能够得到完全匹配，由于该算法每次成功匹配后都要更新该端口的 Round Robin 指针，各输出端口处的优先级指针必然不同。在该情况下，每个输出端口收到的请求数目都为 N，每次轮询的顺序不变，那么该信元在 t 时刻后的 N 个时隙内，端口 j 的 Round Robin 指针必有一次等于 i，因此该信元在 t 时刻后的 N 个时隙内必能交换到端口 j。　　证毕

通过前面的分析可知，本文提出的改进算法保证了选择范围较小的输出端口先仲裁，而选择范围较大的端口即使较晚仲裁，较之收到请求少的输出端口，得到匹配的概率也会较大。因此，在每个信元时隙里，改进算法比原算法获得更多匹配数目，改善了调度时延、丢失率等性能，进一步满足

了星上交换的需要。另外，该算法在每次成功匹配后都要更新输出端口处的轮询指针，也保证了公平性。

4　仿真结果及比较

本节通过OPNET仿真来比较改进算法与其他算法的性能。由于iSLIP算法目前比较常用，具有一定的代表性，且该算法通常少于$\log_2 N$次迭代后收敛[8,9]，因此，本节对4次迭代的SLIP算法、原有串行调度算法及改进算法进行仿真，根据仿真结果分析比较这3种算法的平均调度时延性能，并在缓存队列有限的情况下对改进算法与原有串行调度算法的丢失率进行比较。

4.1 仿真模型

根据星上多波束的特点，每个波束应对应交换机的一个端口，本节选择对16端口的交换机进行仿真和研究。

图1是通过OPNET构造的一个16×16输入缓存的Crossbar交换节点，其中包括输入队列(采用VOQ技术)、交换模块和调度模块。不同的调度算法通过改变调度模块及输入队列中的进程来实现。

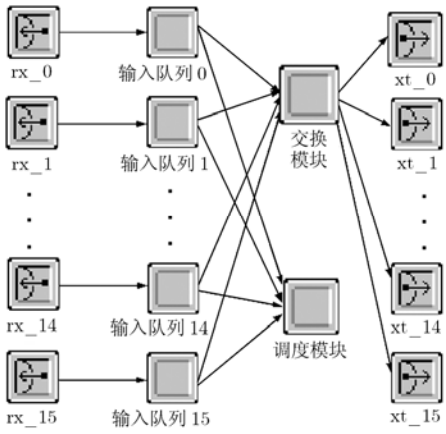

图1 OPNET构造的16×16交换节点

仿真中，信元的产生、缓存、交换和输出都将在一个时隙内同时进行，这里的一个时隙指的是交换网络交换一个信元的时间；假设各输入端的信元到达过程相互独立，且输入/输出队列容量足够大。仿真中采用两种信元到达过程[3]：(1)独立同分布贝努里过程；(2)突发过程。

4.2 仿真结果

本节仿真了200000个时隙，其中前100000个时隙使各VOQ有一定的信元积累，结果由后100000个时隙内收集的统计信息得到。各调度算法在不同负载下，平均时延变化较大，这里分别给出高、低负载的结果。突发过程下，负载较低时，由于各突发长度性能差别不大，只给出了突发长度为16的仿真结果。

从图2，图3可看出，两种信元到达过程下，低负载时，各算法的平均调度时延都不是很大，本文提出的改进算法略

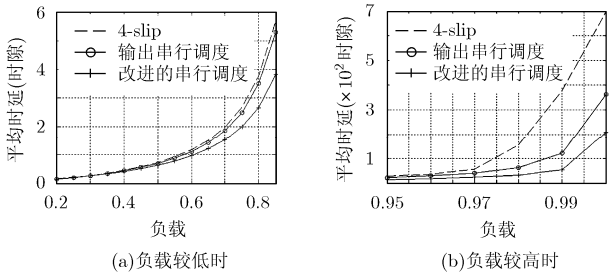

图2 16×16交换网络按独立同分布
贝努里过程到达情况下的平均时延比较

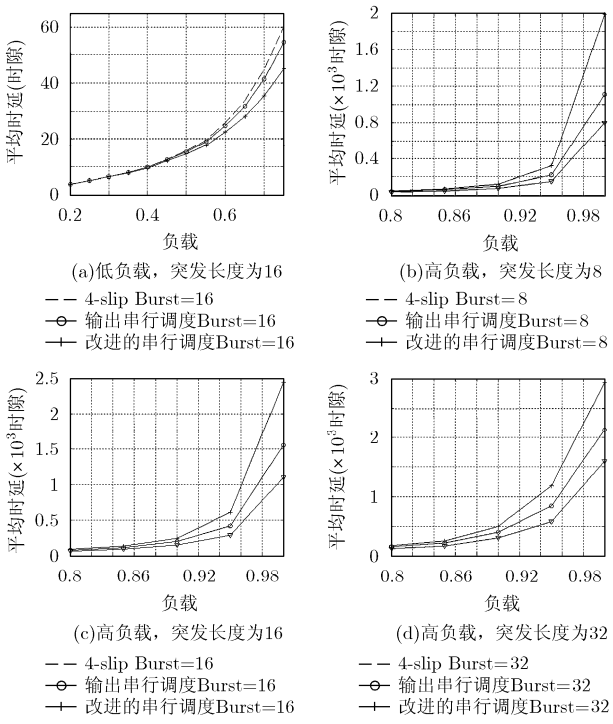

图3 16×16交换网络在不同突发长度下的平均时延比较

优于其他两种算法；当负载增大时，改进的串行调度算法平均调度时延明显小于iSLIP算法和原有输出串行调度算法。这是因为输出串行调度算法避免了并行迭代算法中的同步现象，增加了每次调度中匹配的数目，也降低了平均时延；因此该算法在两种模型下的时延都小于iSLIP算法。而本文提出的改进算法在原有串行调度的基础按照先仲裁请求数目较少输出端口的原则，增加了调度中匹配的数目，因此平均时延也就相应降低；改进算法在每个输出端口得到匹配后就更新轮询指针，保证了公平性。

图4比较了负载分别为0.65和0.95、突发长度为10时，原输出串行调度算法和改进算法在不同VOQ队列长度下的丢失率。从图中可看出，改进算法带来的信元丢失率小于输出串行调度算法；两种算法在负载较低(0.65)情况下的丢失率小于负载较高(0.95)时；缓存区的增加对于减少低负载(0.65)到达信元的丢失率效果明显，若要减小高负载到信元

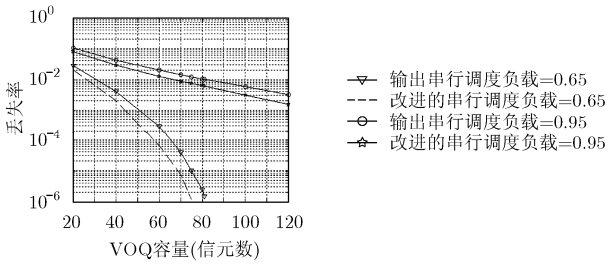

图 4 当信元到达服从 L=10 的 ON/OFF 过程时，VOQ 容量、调度算法和信元丢失率的关系，N=16

的丢失率，缓存区还需要很大。

5 硬件实现的分析比较

(1)逻辑复杂性 在实现方面，iSLIP 算法并行匹配所有端口，每次仲裁得到尽可能多的匹配，调度器内部不同端口处理逻辑之间通信复杂，且不同端口的竞争和选择是同时进行的，因此调度器需要复杂的仲裁逻辑，而串行调度每次仲裁时最多只产生一对匹配，设计起来要比并行调度简单，易于流水实现。

(2)仲裁器的数目 iSLIP 算法每次迭代需要请求、响应和接受 3 步，且所有端口同时响应和接受，因此该算法需要有 $2N$ 个仲裁器；而串行调度每次迭代只需 2 步，且串行仲裁，只需一个仲裁器就可完成端口的匹配。

(3)每次迭代所需时钟节拍 在每次调度中，iSLIP 算法迭代小于 $\log_2 N$ 次就能达到收敛，而串行调度算法则需要匹配 N 次，因此实现时需要的时钟节拍较多，但在 N 较小时，通过流水线实现，不会成为交换的瓶颈。

本文提出的改进算法也是串行调度的思想，实现起来与输出串行调度算法基本相同，相比原算法需要记录每个输出端口收到请求的数目并比较，但由于调度算法与交换并行完成，因此改进算法不会带来额外的时间开销。

6 结束语

本文针对星上交换的性能要求在串行调度算法的基础上进行了改进，并在均匀和突发两种模型下对几种调度算法进行了仿真和比较。结果表明，本文提出的改进算法在平均调度时延方面的性能明显优于 iSLIP 算法，也优于原有串行调度算法；在缓存大小有限的情况下，改进算法的丢失率小于原串行调度算法。通过硬件分析可知，该算法实现起来比并行调度算法更加简单。

作为输入缓存型 Crossbar 结构中的重要算法，该算法不仅可用于 ATM 交换或 IP 交换中，还可构成多级交换中的基本交换单元。因此，对于星上交换芯片受限的情况，可考虑采用该算法。

参 考 文 献

[1] 《卫星与网络》杂志. 超高速因特网卫星[OL]. http://tech.sina.com.cn/t/2007-10-10/16331784404.shtml, 2007, 10.

[2] 黎军, 周诠. 卫星 ATM 交换系统中一种连接允许控制算法的改进[J]. 宇航学报, 2006, 27(3): 513–517.
Li Jun and Zhou Quan. Connection admission control in satellite ATM switching system: A new improved strategy [J]. *Journal of Astronautics*, 2006, 27(3): 513–517.

[3] 陈锡生. ATM 交换技术[M]. 北京: 人民邮电出版社, 2000: 59–61.
Chen Xi-sheng. ATM Switching Technology[M]. Beijing: People Posts and Telecommunications Press, 2000: 59–61.

[4] Kim Hakyong and Kim Kiseon. Performance analysis of the multiple input-queued packet switch with the restricted Rule[J]. *IEEE/ACM Trans. on Networking*, 2003, 11(3): 478–487.

[5] 张志群, 魏激波, 丁炜. 基于螺旋线的 Round-Robin Crossbar 调度算法[J]. 电子与信息学报, 2003, 25(6): 816–823.
Zhang Zhi-qun, Wei Ji-bo, and Ding-Wei. A R-R Crossbar scheduling algorithm based on spirality[J]. *Journal of Electronics & Information Technology*, 2003, 25(6): 816–823.

[6] 魏利华, 唐玉华. Crossbar 输入排队调度算法的研究[J]. 计算机应用与软件, 2006, 23(3): 22–24.
Wei Li-hua and Tang Yu-hua. Research of scheduling input-queue algorithms with Crossbar[J]. *Computer Applications and Software*, 2006, 23(3): 22–24.

[7] 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(10): 1078–1082.
Sun Zhi-gang, Su Jin-shu, and Lu Xi-cheng. ISP: A high performance crossbar arbitrating algorithm [J]. *Chinese J. Computers*, 2000, 23(10): 1078–1082.

[8] 赵增辉, 李文江. 基于 VOQ 输入缓存交换系统调度算法研究[J]. 无线电通信技术, 2006, 32(6): 59–61.
Zhao Zeng-hui and Li Wen-jiang. Study on scheduling algorithms for VOQ input-buffer switches[J]. *Radio Communications Technology*, 2006, 32(6): 59–61.

[9] Nick McKeown. The iSLIP scheduling algorithm for input-queued switches[J]. *IEEE/ACM Trans. on networking*, 1999, 7(2): 188–201.

张 怡： 女，1982 年生，工程师，研究方向为卫星数据传输与处理.

周 诠： 男，1965 年生，研究员，博士生导师，研究方向为卫星数据传输与处理.

黎 军： 男，1975 年生，工程师，研究方向为卫星数据传输与处理.