输入缓冲交换开关的多步调度策略*

孙志刚，卢锡城

(国防科学技术大学 计算机学院，湖南 长沙 410073)

E-mail：zgsun1@263.net

http://www.nudt.edu.cn

摘要：输入缓冲交换开关已经在越来越多的ATM交换机和高性能路由器中使用.对于独立的信元到达，VOQ(virtual output queueing)技术与LQF(longest queue first)和OCF(oldest cell first)等加权调度算法的结合使用可以使利用交换开关的吞吐率达到100%.然而LQF和OCF等加权调度算法过于复杂，无法用硬件实现.提出了多步调度策略，使得用硬件实现加权调度算法成为可能.在该策略下，对于独立的信元到达，LQF算法仍可以达到100%的利用开关吞吐率，并具有良好的延时特性.

关键词：输入缓冲交换开关；加权调度算法；多步调度策略

中图法分类号：TP393 **文献标识码**：A

由于硬件实现简单，输入缓冲交换开关已经在越来越多的ATM交换机和高性能路由器中使用[1-3].当输入端口使用单一的FIFO(first in first out)排队机制时，HOL(head of line)阻塞使得开关吞吐率最多只能利用58%[3-6]，因此，在目前输入缓冲的交换设备中，开关一般采用VOQ(virtual output queueing)技术，即每个输入端口为到达不同输出端口的信元设置不同的FIFO队列.VOQ技术的采用消除了HOL阻塞.[3]

我们将只使用VOQ队列空满标志进行调度决策的调度算法称为非加权算法，将使用其他信息，如VOQ队列长度、信元等待时间等进行调度决策的算法称为加权调度算法.加权调度算法实现复杂，但性能优于非加权调度算法.非加权调度算法，如iSLIP[4]，只有在信元到达服从一致、独立、同分布的贝努里过程时才可以100%地利用开关吞吐率[7].然而，在实际情况中，信元的到达是非一致的.对于独立的、非一致的信元到达过程，只有加权调度算法，如LQF[4]，OCF[5]和LPF[6]等，才可以100%地利用开关带宽.

对于$N\times N$的交换开关，LQF和OCF算法的复杂性为$O(N^3\log N)$[4,5]，LPF算法的复杂性为$O(N^{2.5})$[6].由于硬件实现加权调度算法需要大量的高速比较器和比较器之间的快速互连，因此十分困难.为了解决这一问题，本文提出了多步调度策略.该策略虽然不能降低算法的复杂性，但可以在保证算法性能的前提下，使算法以较慢的速度，在较长的时间内执行，因此减小了调度器实现的难度.

1 输入缓冲开关模型

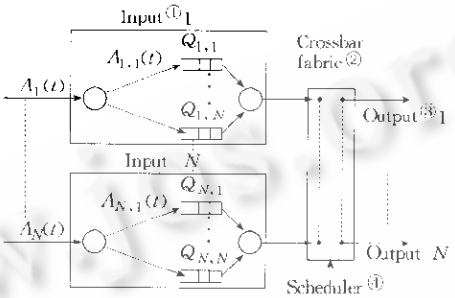
采用VOQ技术的输入缓冲开关模型如图1所示，开关包含N个输入端口和N个输出端口，

* 收稿日期：1999-12-09；修改日期：2000-04-17

基金项目：国家自然科学基金资助项目(69933030)

作者简介：孙志刚(1973－)，男，江苏连云港人，博士，主要研究领域为计算机网络与通信；卢锡城(1946－)，男，江苏靖江人，教授，博士生导师，中国工程院院士，主要研究领域为高性能计算，计算机网络，分布处理.

到达输入端口的包是定长的,称为信元.开关交换一个信元的时间为一个时间槽,第 n 个时间槽开始的时间称为时刻 n,$n=0,1,2,\ldots$,输入端口的信元到达服从离散、各态遍历的过程 $A_i(n)$,速率为 λ_i,$1\leqslant i\leqslant N$.每个时间槽开始有 0 或 1 个信元到达每个输入端口.若输入端口 i 到达一个目的端口为 j 的信元,$1\leqslant j\leqslant N$,那么该信元将被放入 FIFO 队列 $Q(i,j)$ 中.记时刻 n 队列 $Q(i,j)$ 的长度为 $L_{i,j}(n)$,输入端口 i 中目的输出端口为 j 的信元到达过程为 $A_{i,j}(n)$,速率为 $\lambda_{i,j}$,因此有 $\lambda_i=\sum_{j=1}^{N}\lambda_{i,j}$.



①输入,②交换阵列,③输出,④调度器.

Fig. 1 Module of VOQ switches

图1 采用VOQ队列技术的交换开关模型

定义 1. 对于信元到达过程的集合 $A(n)=\{A_i(n);1\leqslant i\leqslant N\}$,若 $\sum_i\lambda_{i,j}<1$,$\sum_j\lambda_{i,j}<1$,则称 $A(n)$ 是可接受的(admissible),否则称 $A(n)$ 是不可接受的.

定义 2. 到达过程是一致的(uniform)当且仅当各输入端口到达过程有相同的速率,且信元到每个目的输出端口的概率是一致的.

调度器的功能是确定每个时间槽内输入、输出端口之间的连接关系.调度算法实际上是寻找二部图 (X,Y,E) 的一个匹配 M[4,5].其中 X,Y 分别代表输入、输出端口的集合.E 代表 X,Y 之间边的集合.若 $L_{i,j}(n)>0$,则边 $e_{i,j}\in E$.对于非加权调度算法,边 $e_{i,j}$ 的权 $w_{i,j}$ 为 1;对于加权调度算法,$w_{i,j}$ 按一定的规则给出.例如,对于 LQF 算法,$w_{i,j}=L_{i,j}(n)$,M 按最大权匹配算法[8]求出.

每个时间槽结束后,若边 $e_{i,j}\in M$,则队列 $Q(i,j)$ 头部的信元将被送往输出端口 j,队列的长度减 1.

定义 3. 对于某种信元到达过程,如果交换开关输入队列长度的数学期望不会无限地增长,那么称这个开关是稳定的.即 $E\left[\sum_{i,j}L_{i,j}(n)\right]<\infty$,$\forall\ n$.

定义 4. 设开关的各个输入端口的信元到达是独立的,如果对于任何可接受的到达过程,开关都是稳定的,那么称开关可以 100%地利用吞吐率.

2 多步调度策略

2.1 单步调度策略

为了提高效率,目前所有的输入缓冲交换开关在第 n 个时间槽对信元交换的同时,调度器根据各 VOQ 的状态确定开关在第 $n+1$ 个时间槽内的拓扑,并将配置信息写入开关.第 n 个时间槽结束后,开关根据预先的配置迅速改变内部拓扑,并启动第 $n+1$ 个时间槽的交换.我们将上述方式称为单步调度.

单步调度对调度器的速度要求很高.例如,对于端口速率为 2.5Gbps、信元长度为 64 字节、

16×16的交换开关来说,每个时间槽为200ns左右,调度器要在这200ns内获得256个VOQ的状态,执行调度算法并将调度结果写入交换开关.由于时间短,目前调度器中实现的都是简单的非加权调度算法[2,3],然而这些算法并不能保证开关是稳定的.

2.2 多步调度策略

多步调度策略将$k(k>1$,称为步长)个时间槽划分为一个阶段.调度器在每个阶段对开关的拓扑结构配置一次,因此调度器可以有足够的时间(k个时间槽)执行较复杂的加权调度算法.

由于调度器是以阶段为基本的时间单位进行工作,它只关心开关在kn时刻的状态,$n=0,1,2,\ldots$,因此,从调度器的角度来看,多步调度有如下特点:

- 队列$Q(i,j)$的到达率为$k\lambda_{i,j}$,每个阶段开始有m个信元到达,$0\leqslant m\leqslant k$.
- 若某阶段将开关输入i与输出j相连,那么该阶段内有q个信元从输入i交换到输出j.

$$q=\begin{cases}k & \text{当 } L_{i,j}(n)\geqslant k\\ L_{i,j}(n) & \text{其他}\end{cases},\quad n\text{为该阶段开始时刻.} \qquad (*)$$

我们将在多步调度策略下实现的LQF算法称为SLQF(stride longest queue first).下面,我们以SLQF算法为例来分析多步调度策略的性能.

2.3 SLQF算法的稳定性

由式(*)可以看出,当VOQ队列长度小于k时,部分交换带宽将被浪费.因此,k越小越好.然而从一个较长的过程来看,当端口负载较低、VOQ队列较短时,一个阶段所浪费的带宽将在后续阶段中补足.当负载较高,VOQ队列较长时,SLQF对交换带宽的利用与LQF算法相一致.下面的定理1证明了当信元到达是独立分布时,SLQF算法可以100%地利用开关的吞吐率.

定理1. 当信元到达是独立分布时,SLQF算法可以100%地利用交换开关的吞吐率.

证明见附录.

我们在8×8的开关上对LQF,SLQFk($k=2,4,8,16,64$)和iSLIP等7种算法的性能进行了模拟,SLQFk表示步长为k的SLQF算法.模拟的时间长度为2×10^5个时间槽.算法对带宽的利用律用平均每个时间槽开关交换的信元数p($0\leqslant p\leqslant8$,精确到0.001)来衡量.

文献[9]的研究表明,具有长期相关性的自相似(self-similar)交通模型比独立分布的交通模型更能反映出路由器端口的报文到达情况.与独立分布的模型相比,自相似模型具有更大的突发性.我们用两状态的burst过程来模拟输入端口的报文到达.虽然该过程不能严格模拟自相似模型,但通过选取合适的burst长度和目的端口转移概率,两状态burst过程可以模拟自相似信元流突发性较强的特点.

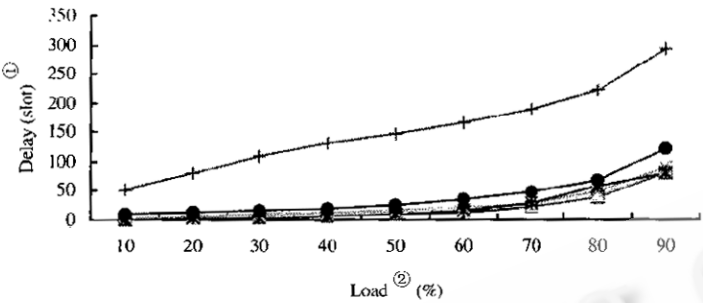
模拟结果如图2所示.图2表明:

(1) LQF和SLQFk($k=2,4,8,16,64$)算法的p值完全相同.

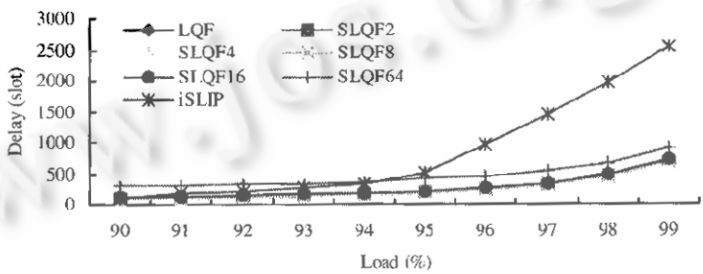
(2) 当负载小于0.90时,iSLIP算法的p值与LQF等算法的p值相同,这是因为低负载时信元在VOQ中很少积累,任何调度算法在性能上都等价于随机调度算法.

(3) 当负载大于0.90时,iSLIP算法固有的Round Robin指针同步问题[7]减小了算法对带宽的利用率,因此,iSLIP算法的p值小于LQF和SLQF算法的p值.例如,当负载为0.99时,iSLIP算法的p值为7.847,而LQF和SLQF算法的p值为7.916.

因此,SLQF算法对开关的吞吐率与LQF算法一致,优于iSLIP算法.



(a) Case 1: load<0.9
(a) 情况 1:负载小于 0.9



(b) Case 2: load>0.9
(b) 情况 2:负载大于 0.9
①延时(时间槽),②负载.

Fig. 2 Delay comparison of LQF, SLQF and iSLIP algorithms
图 2 LQF,SLQF 和 iSLIP 算法的延时特性比较

2.4 SLQF 算法步长的选取

SLQF 算法的延时特性会随着步长的增加而变差,这有以下几个原因:

(1) 当 k 大于 VOQ 长度时,一个阶段内部的带宽浪费使部分信元通过开关的时间后推.

(2) 调度器利用第 n 个阶段开始时的 VOQ 长度计算第 $n+1$ 个阶段开关的拓扑.负载较高时,k 越大,第 n 和 $n+1$ 阶段开始时的 VOQ 长度差别越大(在 n 阶段开始时最长的 VOQ 队列在第 $n+1$阶段开始时可能不再是最长的队列),选取较短队列中信元交换的概率越大,信元的平均延时的增加也越大.

算法的延时特性可用所有被交换信元的平均延时加以衡量.图 2 显示了各种算法的负载延时曲线.从图中可以看出:

(1) 当 k 不大于开关端口数 N 时,SLQF 算法的延时特性与 LQF 相当.当步长加大到 $2N$ 和 N^2 时,SLQF 的延时特性逐渐变差.

(2) 高负载时,SLQF 算法的延时特性优于 iSLIP 算法.

高负载时,VOQ 队列非空,SLQF 的延时特性优于 iSLIP 算法的原因是,前者选取较长队列中的信元交换,后者按 Round Robin 方法选取信元交换,而选取较长队列中的信元交换有利于降低信元的平均延时.

以上结果表明,当 SLQF 算法的步长等于 N 时比较合适.

3 总 结

本文提出输入缓冲交换开关的多步调度策略,并以SLQF算法为例,证明了采用多步调度策略的调度算法可以100%地利用交换开关的吞吐率.通过模拟可知,当步长为N时,SLQF算法的性能与LQF算法接近.由于SLQF算法易于实现,因此多步调度策略是有意义的.文献[5]中指出,LQF算法可能会带来VOQ中信元等待时间无穷大的问题,而具有相同复杂性的OCF算法可以避免上述现象的发生,因此研究多步调度策略下的OCF算法是我们下一步的工作.

同时,由于自相似过程可以较真实地反映报文到达的实际情况,我们还将对开关在自相似报文到达时的性能作进一步的分析.

References:

[1] Keshav, S., Sharma, R. Issues and trends in router design. IEEE Communication Magazine, 1998,36(3):144~151.

[2] Partridge, C., Carvey, P., Burgess, E., *et al*. A 50-Gbps IP router. IEEE/ACM Transactions on Networking, 1998,6(3):237~247.

[3] McKeown, N., Izzard, M. The tiny tera: a packet switch core. IEEE Micro, 1997,17(1):26~33.

[4] McKeown, N., Anantharam, V., Walrand, J. Achieving 100% throughput in an input-queued switch. In: Proceedings of the IEEE Infocom'96. San Francisco, 1996. http://tiny_tera.stanford.edu/~nickm/papers.html.

[5] Nick, McKeown, *et al*. A starvation-free algorithm for achieving 100% throughput in an input queued switch. In: Proceedings of the ICCCN'96. 1996. http://tiny_tera.stanford.edu/~nickm/papers.html.

[6] Mekkittikul, A., McKeown N. A practical scheduling algorithm to achieve 100% throughput in input queued switches. In: Proceedings of the IEEE Infocom'98. San Francisco, 1998. http://tiny_tera.stanford.edu/~nickm/papers.html.

[7] McKeown, N. iSLIP: a scheduling algorithm for input-queued switches. IEEE Transactions on Networking, 1999,7(3):188~201. http://tiny_tera.stanford.edu/~nickm/papers.html.

[8] Xie, Zheng, Li, Jian-ping. Network Algorithms and Complexity Theories. Changsha: Press of National University of Defence Technology, 1995 (in Chinese).

[9] Paxson, V., *et al*. Wide-Area traffic: the failure of poisson modeling. IEEE/ACM Transactions on Networking, 1995,3(3):226~244.

[10] Meyn, K. Stability of queueing networks and schduling policies. IEEE Transactions on Automatic Control, 1995,40(2):251~260.

附中文参考文献:

[8] 谢政,李建平.网络算法与复杂性理论.长沙:国防科学技术大学出版社,1995.

附录 SLQF算法的稳定性证明

设SLQF算法步长为k,为了简洁,下面将第n个阶段开始的时刻kn记为n,$n=0,1,2,\ldots$

基本定义:

(1) 开关的状态向量$\overline{L}(n)\equiv(L_{1,1}(n),L_{1,2}(n),\ldots,L_{1,N}(n),\ldots,L_{N,1}(n),\ldots,L_{N,N}(n))^T$,其中$L_{i,j}(n)$为队列$Q(i,j)$在$n$时刻的长度.

(2) 信元到达开关的速率向量$\overline{\lambda}\equiv K^{-1}(\lambda_{1,1},\lambda_{1,2},\ldots,\lambda_{1,N},\ldots,\lambda_{N,1},\ldots,\lambda_{N,N})^T$,其中$\lambda_{i,j}$为队列$Q(i,j)$的到达速率.

(3) 开关的信元到达向量$A(n)\equiv(A_{1,1}(n),A_{1,2}(n),\ldots,A_{1,N}(n),\ldots,A_{N,1}(n),\ldots,A_{N,N}(n))^T$,其中

$$A_{i,j}(n)=\begin{cases}p & \text{第}n\text{个阶段开始有}p\text{个信元到达队列}Q(i,j)\\ 0 & \text{其他}\end{cases},\quad 1\leqslant p\leqslant K.$$

(4) 开关 VOQ 队列的服务向量 $\overline{S}(n)\equiv(S_{1,1}(n),S_{1,2}(n),\ldots,S_{1,N}(n),\ldots,S_{N,1}(n),\ldots,S_{N,N}(n))^T$,其中

$$S_{i,j}(n)=\begin{cases}K & \text{第 } n \text{ 个阶段内输入 } i \text{ 与输出 } j \text{ 相连}\\ 0 & \text{其他}\end{cases}.$$

根据 $S_{i,j}(n)$的定义,有 $\sum_{i=1}^{N}S_{i,j}(n)=K$ 和 $\sum_{j=1}^{N}S_{i,j}(n)=K$,因此 $\|\overline{S}(n)\|^2=NK^2$.

(5) 近似的下一状态向量 $\overline{Y}(n+1)=\overline{L}(n)-\overline{S}(n)+\overline{A}(n)$,实际的下一状态向量 $\overline{L}(n+1)$与 $\overline{Y}(n+1)$的关系为

$$L_{i,j}(n+1)=Y_{i,j}(n+1)+\begin{cases}k-L_{i,j}(n) & 0\leqslant L_{i,j}(n)<S_{i,j}(n)\\ 0 & \text{其他}\end{cases},\quad 1\leqslant i,j\leqslant N.$$

(6) SLQF 算法就是找一个服务向量 $\overline{S}^*(n)$,对于任何其他服务向量 $\overline{S}^t(n)$,有 $L^T(n)\overline{S}^*(n)\geqslant L^T(n)\overline{S}^t(n)$.

引理 1. $\forall\ (L(n),\overline{\lambda}),\quad L^T(n)\overline{S}^*(n)\geqslant L^T(n)\overline{\lambda}$.

证明:上式等价于 $L^T(n)(\overline{S}^*(n)/K)\geqslant L^T(n)(\overline{\lambda}/K)$,$\overline{S}^*(n)/K$ 和 $\overline{\lambda}/K$ 分别是 N 阶置换矩阵和双随机矩阵行向量的组合.由于所有双随机矩阵组成一个凸集,所有极点都为置换矩阵,又因为对于任何服务矩阵 $\overline{S}^t(n)$(对应于凸集中的点 $\overline{S}^t(n)/K$),有 $L^T(n)(\overline{S}^*(n)/K)\geqslant L^T(n)(\overline{S}^t(n)/K)$,因此对于任何双随机矩阵 A,都有 $L^T(n)(\overline{S}^*(n)/K)\geqslant L^T(n)A$,所以,$L^T(n)(\overline{S}^*(n)/K)\geqslant L^T(n)(\overline{\lambda}/K)$. □

为了简洁,下面将 $\overline{S}^*(n)$简写为 $\overline{S}(n)$.

引理 2. $\forall\ \overline{\lambda},E(\overline{L}^T(n+1)\overline{L}(n+1)-\overline{L}^T(n)\overline{L}(n)\,|\,\overline{L}(n))\leqslant 2NK^2$.

证明:

$$\begin{aligned}\overline{L}^T(n+1)\overline{L}(n+1)&=\overline{L}^T(n)\overline{L}(n)+(\overline{L}(n)-\overline{S}(n)+\overline{A}(n))^T(\overline{L}(n)-\overline{S}(n)+\overline{A}(n))-\overline{L}^T(n)\overline{L}(n)\\&\quad-2\overline{L}^T(n)(\overline{S}(n)-\overline{A}(n))+(\overline{S}(n)-\overline{A}(n))^T(\overline{S}(n)-\overline{A}(n)),\end{aligned}$$

考虑引理 1 的结果:

$$\begin{aligned}E(\overline{L}^T(n+1)\overline{L}(n+1)-\overline{L}^T(n)\overline{L}(n)\,|\,\overline{L}(n))&=-2\overline{L}^T(n)(\overline{S}(n)-K\overline{\lambda})+\sum_{i,j}(K\lambda_{i,j}-S_{i,j})^2\\&\leqslant\sum_{i,j}(K\lambda_{i,j}-S_{i,j})^2\leqslant 2NK^2.\end{aligned}$$

□

引理 3. $E(\overline{L}^T(n+1)\overline{L}(n+1)-\overline{L}^T(n)\overline{L}(n)\,|\,\overline{L}(n))\leqslant-\varepsilon\|\overline{L}(n)\|+2NK^2$,其中 $\varepsilon>0$,$\forall\ \overline{\lambda}<K(1-\beta)\overline{\lambda}_m$,$0<\beta<1$,$\overline{\lambda}_m$为任意的速率向量,且 $\|\lambda_m\|^2=N$.

证明:

$$\begin{aligned}\overline{L}^T(n)(K\overline{\lambda}-\overline{S}(n))&\leqslant\overline{L}^T(n)(K\overline{\lambda}_m-\overline{S}(n))-\overline{L}^T(n)K\overline{\beta\lambda}_m\\&\leqslant-K\beta\overline{L}^T(n)\overline{\lambda}_m\|\overline{L}(n)\|/|\overline{L}(n)\|,\end{aligned}$$

设 $L_{\max}(n)=\max(L_{i,j}(n),1\leqslant i,j\leqslant N$,$\lambda_{\min}=\min(\lambda_{mi,j},1\leqslant i,j\leqslant N)$,则 $\overline{L}^T(n)\overline{\lambda}_m\geqslant L_{\max}*\lambda_{\min}$.又因为 $\|\overline{L}(n)\|\leqslant\sqrt{N^2*L_{\max}^2(n)}=N*L_{\max}(n)$,因此

$$\begin{aligned}E(\overline{L}^T(n+1)\overline{L}(n+1)-\overline{L}^T(n)\overline{L}(n)\,|\,\overline{L}(n))&\leqslant\overline{L}^T(n)(K\overline{\lambda}-\overline{S}(n))+2NK^2\\&\leqslant-K*\beta*L_{\max}(n)*\lambda_{\min}*\|\overline{L}(n)\|/(N*L_{\max}(n))+2NK^2\\&\leqslant-\varepsilon|\overline{L}(n)\|+2NK^2,\end{aligned}$$

其中 $\varepsilon=K\beta\lambda_{\min}/N$. □

引理 4. $E(\overline{L}^T(n+1)\overline{L}(n+1)-\overline{L}^T(n)\overline{L}(n)\,|\,\overline{L}(n))\leqslant-\varepsilon\|\overline{L}(n)\|+4NK^2$,其中 $\varepsilon>0$,$\forall\ \overline{\lambda}<K(1-\beta)\overline{\lambda}_m$,$0<\beta<1$,$\overline{\lambda}_m$为任意的速率向量,且 $\|\lambda_m\|^2=N$.

证明:

$$\begin{aligned}&E(\overline{L}^T(n+1)\overline{L}(n+1)-\overline{L}^T(n)L(n)\,|\,L(n))\\&\quad=E(\overline{L}^T(n+1)\overline{L}(n+1)-\overline{Y}^T(n+1)\overline{Y}(n+1)\,|\,\overline{L}(n))+E(\overline{Y}^T(n+1)\overline{Y}(n+1)-\overline{L}^T(n)\overline{L}(n)\,|\,\overline{L}(n))\\&\quad\leqslant-\varepsilon|\overline{L}(n)|+2NK^2+E(\overline{L}^T(n+1)\overline{L}(n+1)-\overline{Y}^T(n-1)\overline{Y}(n+1)\,|\,\overline{L}(n)),\end{aligned}$$

其中 $\varepsilon=K\beta\lambda_{\min}/N$.

又因为 $L_{i,j}(n+1)=\overline{L}_{i,j}(n-1)+\begin{cases}k-L_{i,j}(n) & 0\leqslant L_{i,j}(n)<S_{i,j}(n)\\ 0 & \text{其他}\end{cases}$, $1\leqslant i,j\leqslant N$,

因此,

$$\begin{aligned}&E(\overline{L}^T(n+1)\overline{L}(n+1)-\overline{\tilde{L}}^T(n+1)\overline{\tilde{L}}(n+1)\mid\overline{L}(n))\\&\quad-\sum_{i,j}E((L_{i,j}(n+1)+\tilde{L}_{i,j}(n+1))^*(L_{i,j}(n+1)-L_{i,j}(n+1)))\\&\quad\leqslant\sum_{i,j}(2K\lambda_{i,j}(S_{i,j}(n)-L_{i,j}(n))-(S_{i,j}(n)-L_{i,j}(n))^2\\&\quad\leqslant\sum_{i,j}2K\lambda_{i,j}{}^*K\leqslant 2NK^2,\end{aligned}$$

因此,$E(L^T(n+1)\overline{L}(n+1)-\overline{L}^T(n)\overline{L}(n)\mid\overline{L}(n))\leqslant-\varepsilon\|\overline{L}(n)\|+4NK^2$. □

SLQF 算法的稳定性证明:

设 $V(\overline{L}(n))=\overline{L}^T(n)\overline{L}(n)$,则 $V(\overline{L}(n))$为二阶李亚普诺夫函数.由引理 4 可知,存在 $M,\varepsilon>0$,使得 $E(V(\overline{L}(n+1)-V(\overline{L}(n)\mid\overline{L}(n))\leqslant-\varepsilon\|\overline{L}(n)\|+M$.其中 $\varepsilon=K\beta\lambda_{\min}/N$,$M=4K^2N$.因此由文献[10]可得,交换开关是稳定的.

Multi-Step Scheduling Strategy in Input-Queued Switches*

SUN Zhi-gang, LU Xi-cheng

(*School of Computer, National University of Defence Technology, Changsha* 410073, *China*)

E-mail: zgsun1@263.net

http://www.nudt.edu.cn

Abstract: Input-Queued switches are increasingly used in ATM switches and high performance routers. It has been proved that combining VOQ (virtual output queueing) technology and some weighted scheduling algorithms, such as LQF (longest queue first) and OCF (oldest cell first), the switch throughput can reach 100% for all cell arrivals with independent distributions. But the algorithms of LQF and OCF are so complicated that they cannot be easily implemented in hardware. A multi-step scheduling strategy proposed in this paper makes it possible to implement the weighted scheduling algorithms in hardware. It is also proved that the switches based on the LQF by introducing the multi-step scheduling strategy can still get 100% throughput and the better delay property for arrivals with independent distributions.

Key words: input-queued switch; weighted scheduling algorithm; multi-step scheduling strategy

* Received December 9, 1999; accepted April 17, 2000

Supported by the National Natural Science Foundation of China under Grant No. 69933030