高速 IP 路由器中输入排队调度算法综述*

庞 斌[1+], 贺思敏[1], 高 文[1,2,3]

[1](中国科学院 计算技术研究所,北京 100080)
[2](哈尔滨工业大学 计算机科学与工程系,黑龙江 哈尔滨 150001)
[3](中国科学院 研究生院,北京 100039)

A Survey on Input-Queued Scheduling Algorithms in High-Speed IP Routers

PANG Bin[1+], HE Si-Min[1], GAO Wen[1,2,3]

[1](Institute of Computing Technology, The Chinese Academy of Sciences, Beijing 100080, China)
[2](Department of Computer Science and Engineering, Harbin Institute of Technology, Harbin 150001, China)
[3](Graduate School, The Chinese Academy of Sciences, Beijing 100039, China)
+ Corresponding author: Phn: 86-10-82649316, Fax: 86-10-82649298, E-mail: bpang@jdl.ac.cn
http://www.jdl.ac.cn
Received 2002-06-06; Accepted 2002-12-10

Pang B, He SM, Gao W. A survey on input-queued scheduling algorithms in high-speed IP routers. *Journal of Software*, 2003,14(5):1011~1022.
http://www.jos.org.cn/1000-9825/14/1011.htm

Abstract: Most high-speed IP routers exploit cell-based switching fabrics, whose scalability and performance are mainly affected by queuing scheme and scheduling algorithm. Input-queued router is referred to as an ideal structure in terms of scalability. However, it needs an efficient scheduling algorithm to guarantee throughput and delay. Several input-queued scheduling algorithms are surveyed in this paper. The scheduling algorithms are classified into four classes: maximum size matching, maximum weight matching, stable marriage matching, and deterministic scheduling algorithm. The similarities and the difference of different algorithms in mechanisms of each class are described, and their performances are compared. Finally, the future directions and possible open problems are discussed.
Key words: router; switch fabric; queuing scheme; input queued; scheduling algorithm; matching

摘 要: 高速 IP 路由器一般采用基于定长信元的交换结构,其可扩展性和性能分别受排队策略和调度算法的影响.基于输入排队策略的路由器具有良好的可扩展性,但需要一个有效的调度算法的支持,才能保证吞吐率和延迟等性能.主要讨论输入排队调度算法,将现有的调度算法分为 4 类:最大(无权重)匹配、最大权重匹配、稳定婚姻匹配和确定型调度.对每一类算法,从技术特点和性能指标两个方面进行比较和分析.最后给出了输入排队

* Supported by the National Natural Science Foundation of China under Grant No.69983008 (国家自然科学基金); the National High-Tech Research and Development Plan of China under Grant No.2001AA112100 (国家高技术研究发展计划(863)); the Knowledge Innovation Program in CAS under Grant No.KGCXZ-103 (中国科学院知识创新工程)

第一作者简介: 庞斌(1971－),男,山东临沂人,博士生,主要研究领域为计算机网络,多媒体通信技术.

© 中国科学院软件研究所 http://www.jos.org.cn

调度算法的发展趋势.

关键词: 路由器;交换结构;排队策略;输入排队;调度算法;匹配

中图法分类号: TP393 **文献标识码**: A

在过去的 10 年中,随着宽带技术和多媒体应用的迅速发展,Internet 对高性能路由器的需求越来越高.与早期的基于总线结构的路由器不同,现在的高速路由器一般采用基于定长信元的交换结构,以实现快速的分组转发.在这种结构下,交换结构的配置由调度器决定.变长的分组首先在输入端口被分成定长的信元,然后通过交换结构传送到输出端口,最后在输出端口,信元被重新组装成分组,等待发送到输出链路[1].

高速路由器的可扩展性和服务质量(quality of service,简称 QoS)主要受排队策略和调度算法的影响.排队策略(如输入或输出排队)决定如何缓冲到达的分组,主要影响路由器的可扩展性.调度算法保证路由器的可预测 QoS 性能,如吞吐率、延迟、抖动等[2].由于输出排队存在可扩展性问题,高速路由器一般采用输入排队(虚拟输出排队)或组合输入/输出排队结构.在一个非单纯的输出排队的结构下,保证路由器性能的关键在于设计一个有效的调度算法,根据输入(和输出)队列的状态信息决定哪个输入端口能够通过交换结构传送信元,从而避免交换结构的访问冲突.我们将此类算法称为输入排队调度算法.

现有的输入排队调度算法可以分为以下几类:根据匹配算法的不同,可以分为最大匹配和稳定婚姻匹配;根据加速比,可以分为有加速比和无加速比;根据排队策略,可以分为虚拟输出排队和组合输入/输出排队;根据调度算法的实现,可以分为集中式和分布式.实际上,对于某个具体的调度算法,根据不同的分类标准,又可分属多个不同的类.在本文中,我们将现有的输入排队调度算法大致分为以下 4 类:最大(无权重)匹配、最大权重匹配、稳定婚姻匹配和确定型调度算法.

在本文中,我们假设路由器的交换结构以同步方式运行,即在相等的时间间隔内执行调度算法,而且内部不含存储器.有关利用交叉开关的缓冲区实现分布式调度的算法见文献[3].此外,我们主要讨论单播通信模式下的调度算法.有关组播通信模式下的调度算法见文献[4].

本文第 1 节给出输入排队调度算法的研究背景.第 2 节主要对现有的调度算法进行分类并比较不同算法的技术特点和性能.第 3 节介绍调度算法的发展趋势.最后总结全文.

1 研究背景

1.1 路由器排队策略

传统的路由器一般基于输出排队(output queueing,简称 OQ).在这种结构下,到达输入端口的信元马上被交换到相应的输出端口.OQ 的优点是能够提供最优的吞吐量和延迟控制,其调度算法(如通用处理器共享 PGPS[5])在理论上已得到深入的研究且实现简单.但为了保证 OQ 的正常运行,交换结构的内部带宽和输出队列的存储器访问速率必须是输入端口链路速率的 N 倍(如果输入速率不同,则为端口速率之和),即要求 N 倍的加速比,这里,N 是输入端口数.随着链路速率或端口数的增加,在现有的工艺水平下很难实现高速的基于 OQ 的交换结构.

为了克服 OQ 结构的可扩展性问题,高速路由器考虑采用输入排队(input queueing,简称 IQ).到达的信元首先被保存在输入端口的缓冲区中,然后通过调度算法决定信元何时通过交换结构传送到输出端口.IQ 的优点是不需要加速比,但是存在链头阻塞问题(head of line blocking):如果队列链头的信元被阻塞,同队列到其他输出端口的信元就不能被转发.研究表明,当端口数较多时,在所有输出均匀分布的 Bernoulli 到达下,IQ 只能达到 58.6%的吞吐率[6].

解决输入排队链头阻塞问题的一种简单方案是文献[7]提出的虚拟输出排队(virtual output queueing,简称 VOQ).在这种结构下,每个输入端口为每个输出设置一个队列,从而消除了链头阻塞并保持加速比为 1.理论研究和仿真实验都表明,一个采用最大权重匹配调度算法(见第 2.2 节)的 VOQ 路由器可以到达 100%的吞吐率.在本文中,如不特别指出,输入排队路由器均采用 VOQ 排队方式.但是,VOQ 路由器的一个不足是很难提供 QoS

© 中国科学院软件研究所 http://www.jos.org.cn

保证,原因在于信元的转发不仅与输入端口的通信量有关,而且受调度算法的影响.

提高 VOQ 路由器性能的一种方法是利用加速比,这需要在输入和输出端口都设置缓冲区.这种结构称为组合输入/输出排队(combined input/output queueing,简称 CIOQ).研究表明,加速比为 2 的 CIOQ 路由器能够完全仿真一个 OQ 路由器(见第 2.3 节).这样,我们可以利用 CIOQ 继承 OQ 的吞吐率和延迟特性.但是,要在 CIOQ 路由器中实现 QoS 保证,关键在于设计一个有效的配置交换结构的调度算法.

1.2 VOQ和CIOQ路由器逻辑结构

VOQ 路由器的逻辑结构如图 1 所示.它主要由 4 部分组成:输入端口、输出端口、交换结构和调度器.我们假设输入和输出端口的数量都是 N 且数据传输速率相同.输入端口采用 VOQ,共有 $N\times N=N^2$ 个队列.分组在进入交换结构前已被分成定长信元.时间被分成等长的时间片.

输入端口 $i(1\le i\le N)$的信元到达是一个离散时间的随机过程 $A_i(t)$.在一个时间片 t 内,最多有一个信元到达一个输入端口.到达输入端口 i 且输出端口是 j 的信元放入队列 Q_{ij}.在第 t 个时间片内,队列 Q_{ij} 的长度表示为 $L_{ij}(t)$.

我们定义 $A_{ij}(t)$为输入 i 到输出 j 的到达过程,其到达速率为 λ_{ij},到达过程的集合 $A(t)=\{A_i(t), 1\le i\le N\}$.若输入和输出都在负载范围以内,即

$$\sum_{i=1}^{N}\lambda_{ij}<1,\forall j \text{ 和 } \sum_{j=1}^{N}\lambda_{ij}<1,\forall i$$

成立,则 $A(t)$被认为是容许的,否则就是非容许的.我们定义通信量矩阵 $\Lambda=[\lambda_{ij}]$.

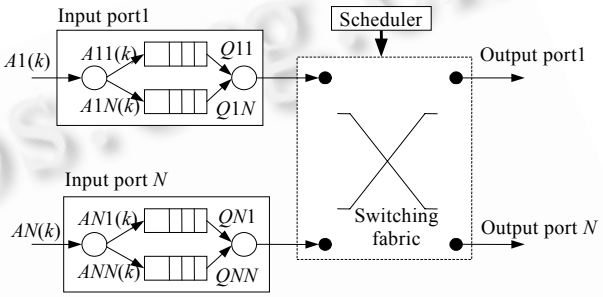

Fig.1 System structure of VOQ router

图 1 VOQ 路由器系统结构

我们用 $N\times N$ 服务矩阵 $S(t)=[s_{ij}(t)]$表示时间片 t 时交换结构的配置,其元素表示调度结果:

$$s_{ij}(t)=\begin{cases}1, & \text{如果有信元从输入}i\text{到输出}j\\ 0, & \text{否则}\end{cases}, \tag{1}$$

矩阵的限制条件(即交换结构的传输限制)是

$$\sum_{i=1}^{N}s_{ij}(t)\le 1,\ \sum_{j=1}^{N}s_{ij}(t)\le 1. \tag{2}$$

CIOQ 和 VOQ 路由器的结构基本相同.但由于加速比大于 1,因此在每个输出端口也有队列.在 CIOQ 结构下,我们将时间片分成更小的阶段(phase).在一个阶段内完成一次信元的调度和传输.

1.3 性能评价

在进行性能评价时,如果我们说信元到达是一个独立的过程,是指它满足以下两个条件:(1) 每个输入端口的信元到达是独立同分布的;(2) 每个输入端口的信元到达独立于其他输入端口.如果信元的到达过程具有相同的速率,并且目的端口均匀分布在所有的输出端口,我们称信元的到达是均匀的.

性能评价常用的指标包括吞吐率和信元的延迟.吞吐率是指单位时间内路由器转发的信元数量.延迟是指信元从到达路由器到离开所经历的时间.我们说一个路由器是稳定的,是指输入队列长度的期望值不能无限增长,即

$$E\left[\sum_{ij}L_{ij}(t)\right]<\infty,\forall t$$

成立.如果一个路由器在所有独立的和容许的到达下都是稳定的,我们说这个路由器能够达到 100%的吞吐率.

2 现有的输入排队调度算法

我们将现有的调度算法分为最大(无权重)匹配、最大权重匹配、稳定婚姻匹配和确定型调度算法 4 类.其

© 中国科学院软件研究所 http://www.jos.org.cn

中最大匹配、最大权重匹配和稳定婚姻匹配都是二分图的匹配算法,而确定型调度算法主要基于矩阵分解.

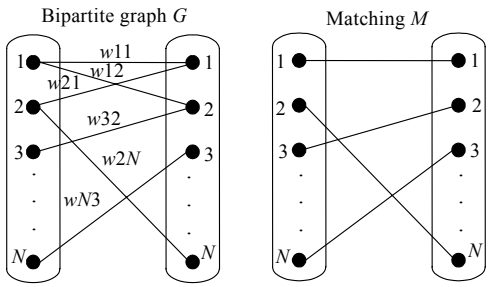

Fig.2 Bipartite graph description of IQ scheduler

图 2 输入排队调度算法的二分图描述

除了确定型调度算法以外,其他调度算法都把交换结构看成一个二分图 $G=[V,E]$(如图 2 所示),这里顶点集合 V 可以分为两个子集:(1) 左顶点子集 V_1,其元素 v_{1k} 表示输入端口 k;(2) 右顶点子集 V_2,其元素 v_{2k} 表示输出端口 k.边集 E 表示从输入到输出端口可能的传输(如从 v_{1i} 到 v_{2j} 的边表示有信元从输入 i 到输出 j).边的权重记为 w_{ij},它可以表示队列中是否有信元,队列长度或队列头信元的等待时间等信息.当输入 i 没有到输出 j 的信元时, $w_{ij}=0$.根据二分图 G 的边的权重组成的 $N\times N$ 矩阵称为权重矩阵 $W=[w_{ij}]$.我们定义时间片 t 时权重矩阵 $W(t)=[w_{ij}(t)]$.

二分图 G 的匹配定义为边集 E 的子集 M,具有性质:M 中没有两条边有公共顶点.因此,如果 M 是一个匹配,那么每一个左顶点最多与 M 的一条边关联,类似地,每一个右顶点最多与 M 的一条边关联.这说明一个二分图的匹配满足交换结构的传输限制条件(见式(2)).

最大匹配(maximum size matching,简称 MSM)是指边数达到最大,而最大权重匹配(maximum weight matching,简称 MWM)是指边的权重之和达到最大.由于这两种算法具有复杂度高、硬件实现复杂等缺点,在实际应用中,我们一般用极大匹配(maximal matching)近似最大匹配.所谓的极大匹配是指在当前已完成的匹配下,无法再通过增加未完成匹配的边的方式来增加匹配的边数或权重.稳定婚姻匹配根据每个信元的优先级别来调度信元.

2.1 基于最大匹配的算法

2.1.1 最大匹配算法(MSM)

我们可以直接用二分图的 MSM 算法解决调度中的匹配问题.目前已知的渐进复杂性最好的该类算法可以达到 $O(N^{2.5})$[8].MSM 采用 1 位的队列占用作为边的权重:当队列中有信元时,相应的边的权重为 1;否则为 0.仿真实验表明[9],MSM 在均匀的独立到达下可以实现 100%的吞吐率.但其也具有以下缺点:(1) 在容许的非均匀通信量下, 可能导致不稳定和不公平;(2) 在非容许的通信量下,可能导致饿死;(3) 算法实现起来过于复杂且运行时间长.

2.1.2 极大匹配算法

在实际应用中,我们一般用启发式算法来解决二分图的匹配问题.这些算法具有实现简单、运算速度快等优点,但只能实现极大匹配.启发式调度算法要经过多次迭代才能找到极大匹配,每次迭代包括 3 个步骤.在一个时间片开始时,所有的输入和输出都初始化为未匹配,只有那些直到一次迭代结束都未能完成匹配的输入和输出才能留到下一次迭代.这 3 个步骤是:

(1) 请求(request):每个未完成匹配的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.

(2) 响应(grant):一个输出端口可能收到多个输入端口发来的请求信号.每个未完成匹配的输出端口从收到的请求中选择一个输入端口并向其发送响应信号.

(3) 接受(accept):每个未完成匹配的输入端口可能收到多个输出端口的响应信号.输入端口从响应信号中选择一个输出端口并向其发送接受信号.

PIM(parallel iterative matching)[10]是第一个采用多次迭代实现输入排队调度的算法.它利用随机的方法选择请求或响应信号.PIM 平均经过 $O(\log N)$次迭代后收敛到极大匹配,并且能够保证所有的请求会被响应.但存在以下不足:首先,在高速情况下实现随机选择存在一定的困难[11];其次,对非容许的通信量,它可能导致连接间的不公平;最后,在单次迭代的情况下,只能达到 63%的吞吐率,仅略高于 FIFO(first in first out)队列.

RRM(round robin matching)[11]算法采用轮转优先算法调度输入和输出端口.在 RRM 算法中,每个输出(入)

© 中国科学院软件研究所 http://www.jos.org.cn

端口有一个跟踪最高优先级输入(出)端口的响应(接受)指针.RRM 算法的主要机制是:在第(2)步,输出端口按固定轮转顺序从它的优先级列表中选择当前优先级最高的元素,然后通知所有输入它的请求是否被响应.指向最高优先级的响应指针增加 1(模 N),移到下一个位置;第(3)步,输入端口按固定轮转顺序从它的优先级列表中选择当前优先级最高的元素,然后发出接受信号.指向最高优先级的接受指针增加 1(模 N),移到下一个位置.RRM 解决了 PIM 中的两个问题:过于复杂和不公平.但是在高负载下,RRM 会变得不稳定,原因在于输出端口响应指针更新的规则不够合理,在负载较高的情况下会出现同步现象,从而导致其最大吞吐率只能达到 50%.

iSLIP(iterative SLIP)[11]算法在 RRM 的基础上进行了改进,以减少同步现象.iSLIP 与 RRM 不同之处在于第(2)步:输出端口的响应指针只有在被接受后才移动到下一个位置.与 PIM 算法相比,iSLIP 算法的主要优点是实现简单且具有高吞吐率.实验结果表明,在均匀的独立 Bernoulli 到达下,即使是单次迭代,iSLIP 算法也能达到 100%的吞吐率,明显高于 PIM 算法[12].

其他调度算法还有 FIRM[13]和 DRRM(dual round robin matching)[14].FIRM 与 iSLIP 的不同之处在于:当输出端口的响应信号被输入端口拒绝时,响应指针将指向该输入端口,从而保证该输入端口的请求信号在下一个时间片内被优先响应.DRRM 算法只需要两次信号交换,即请求和响应.

在以上算法中,在一个时间片内,一次迭代建立的匹配在随后的迭代过程中不能被改变,即使有更好的匹配出现也不行.为了克服这种“局部最大”的限制,Goudreau 等人在文献[15]中提出了 Shakeup 算法,目的是实现“全局最大”.但 Shakeup 需要更多次的迭代才能收敛,在一个实际系统中,该算法是否可行仍然是一个开放的问题.

PIM,iSLIP 和 FIRM 等算法通过多次迭代实现对 MSM 算法的近似,因此具有和 MSM 算法类似的性能,如在均匀的独立到达下,所有算法经过多次迭代都可以实现 100%的吞吐率,但对非均匀通信量会变得不稳定.

2.2 基于最大权重匹配的算法

本节首先给出最大权重匹配算法;其次,讨论极大权重匹配算法,并从技术特点和复杂度两个方面进行比较;随后,我们给出了最大权重匹配算法在 3 个方面的扩展:支持变长分组、多个节点和多类通信量;最后介绍理论研究结果并给出小结.

极大权重匹配算法又分为有记忆和无记忆两类.我们知道,在一个时间片内,最多有一个信元到达(离开)一个输入(输出)端口.这表明队列的长度或路由器的状态在相邻的时间片内不会发生大的变化.因此,在一个时间片内权重最大的匹配在随后的几个时间片内仍然是最大,这就是调度算法的记忆特性.有记忆的极大权重匹配算法就是利用这种特性,根据时间片 t 的最大权重匹配决定 t+1 时的最重匹配,而无记忆的极大权重匹配算法每次都重新开始发现最重匹配.

2.2.1 最大权重匹配算法(MWM)

MWM 是对最大匹配的扩展,在计算边的权重时考虑队列超过 1 位的性质,如队列长度或排队等待时间等.目前,解决这类问题的最有效的算法其渐进复杂度是 $O(N^3\log N)$ [16].

LQF(longest queue first)和 OCF(oldest cell first)[9,17]是较早提出的利用 MWM 实现输入排队调度的两种算法.LQF 把权重 $w_{ij}(t)$设为队列长度 $L_{ij}(t)$,而 OCF 的权重 $w_{ij}(t)$是队列 Q_{ij} 头信元的等待时间.在容许的通信量下,这两种算法都能达到 100%的吞吐率.在非容许通信量下,LQF 有可能出现某个输入端口被饿死的现象.然而,OCF 在任何情况下都不可能出现饿死的现象.为了克服 LQF 算法硬件实现复杂的缺陷,文献[18]提出 LPF(longest port first)算法,它的权重 $w_{ij}(t)$采用端口占有,定义为队列长度的函数:

$$w_{ij}(t)=\begin{cases}R_i(t)+C_j(t), & L_{i,j}(t)>0\\ 0, & \text{否则}\end{cases}, \tag{3}$$

这里,$R_i(t)=\sum_{j=1}^{N}L_{i,j}(t)$, $C_j(t)=\sum_{i=1}^{N}L_{i,j}(t)$.实际上,LPF 的权重并不确切地等于队列的长度,这使得 LPF 能够同时利用最大匹配和最大权重匹配算法的优点.与 LQF 算法一样,LPF 能在均匀和非均匀的通信量下到达 100%的吞吐率,但 LPF 算法的渐进复杂度为 $O(N^{2.5})$,略低于 LQF.

© 中国科学院软件研究所 http://www.jos.org.cn

2.2.2 无记忆的极大权重匹配算法

LQF 算法和 OCF 算法在硬件实现上相当复杂而且运行时间较长,为了克服这个缺点,文献[19]提出两种启发式算法: iLQF 和 iOCF.这两种算法采用类似于 PIM 的多次迭代的方法,但请求信号的长度由 1 位变成多位.其他调度算法还有 iLPF,RPA(reservation with preemption and acknowledgement)和 MUCS(matrix unit cell scheduler).iLPF[18]是一种近似 LPF 的算法,目的是为了提高 LPF 的运行速度,匹配算法采用基于预排序和仲裁的启发式算法.RPA[20]的权重与 LQF 一样,都是队列长度,但匹配算法采用基于预留向量的启发式算法.MUCS[21]的匹配算法采用基于贪婪矩阵的启发式算法,权重定义为

$$w_{ij}(t)=\frac{L_{ij}(t)}{\sum_{k=1}^{N}L_{ik}(t)}+\frac{L_{ij}(t)}{\sum_{k=1}^{N}L_{kj}(t)}. \tag{4}$$

表 1 对以上 5 种算法进行了比较.文献[22]指出,对最大权重匹配算法性能影响最大的因素是如何计算权重,而对算法复杂度影响最大的是实现匹配的启发式算法.

Table 1 Comparison of the maximal weight matching algorithms

表 1 最大权重匹配算法比较

Algorithm	Weight	Matching algorithms	Complexity
iLQF	Queue length	Iterative matching	$O(N^2\log_2 N)$
iOCF	Cell age	Iterative matching	$O(N^2\log_2 N)$
iLPF	Port occupancy	Preordering and arbitration	$O(N^2)$
RPA	Queue length	Reservation vector	$O(N^2)$
MUCS	MUCS length	Matrix greedy	$O(N^3)$

2.2.3 有记忆的极大权重匹配算法(随机调度算法)

为了找到一个更好的匹配,调度器需要更长的时间才能完成多次迭代.但是,随着链路速率或端口数的增加,可供算法运行匹配计算的时间实际上越来越短.为了解决这个矛盾,文献[23,24]提出利用记忆特性和随机化算法近似 MWM,其目的是提供一种在性能上和 MWM 相似,同时在硬件上易实施的算法.这类算法又可称为随机调度算法.

随机化算法的基本思想是:决策过程不是基于全部的状态,而是基于一个小的随机抽取的样本空间,从而大大简化了决策的过程.在文献[23]中,Tassiulas 最早将随机化应用到 VOQ 调度.我们将此算法称为 TASS 算法:

(1) 设 $S(t)$表示时间片 t 时交换结构的服务矩阵.

(2) 在时间片 $t+1$,从 $N!$个可能的匹配中随机均匀地选择一个匹配 $R(t+1)$.

(3) 我们从匹配 $S(t)$和 $R(t+1)$中选择权重较大的一个作为时间片 $t+1$ 的服务矩阵.

TASS 算法在任何容许的独立 Bernoulli 到达下都可以实现 100%的吞吐率.但实验结果显示,TASS 算法的延迟性能很差[24].这是由于 TASS 算法在实现匹配时只考虑了迭代间的记忆特性.实际上,一个匹配的大部分权重集中在少数几个边(重边)上,因此记住重边(heavy edge)比记住匹配更重要.根据这一观察,Giaccone 等人提出了 LAURA 算法[24]:

(1) 设 $S(t)$表示 LAURA 在时间片 t 时交换结构的服务矩阵.

(2) 在时间片 $t+1$,使用 RANDOM 过程产生匹配 $R(t+1)$.

(3) 将 $S(t+1)$=MERGE($R(t+1)$,$S(t)$)作为时间片 $t+1$ 的服务矩阵.

其中,RANDOM 过程随机地生成一个匹配,MERGE 过程根据所有属于匹配 $S(t)$和 $R(t+1)$的边组合成一个最大权重匹配.LAURA 算法的复杂度为 $O(N\log^2 N)$,低于 MWM.其他基于随机化的近似算法还包括 APSARA 和 SERNA[24].APSRA 算法主要利用调度的记忆特性,而 SERNA 算法主要利用记忆特性和最近到达的信息.所有这些算法(LAURA,APSARA 和 SERNA)在容许的独立 Bernoulli 到达下,都可以达到 100%的吞吐率,并且其延迟接近 MWM.

我们将随机化近似算法的特点总结如下:(1) 随机化的方法避免了一个时间片内的多次迭代过程;(2) 如果系统状态在相邻时间片之间变化不大,则可利用这些状态信息进一步简化算法;(3) 仿真实验结果表明[24],利用随机化方法生成的近似算法在吞吐率和延迟方面都有良好的性能.但是,在高速情况下利用随机的方法(如

© 中国科学院软件研究所 http://www.jos.org.cn

LAURA)生成一个匹配非常困难,因此,这类算法硬件实现的复杂性需要进一步的研究.

2.2.4 算法的扩展

最大权重匹配算法的扩展包括 3 个方面:从定长信元到变长分组、从单个节点到多个节点以及从单类通信量到多类通信量.

在 VOQ 路由器中,变长的分组要分成定长的信元才能通过交换结构,然后在输出端口重新组装成分组.但是经过调度后,属于同一分组的信元可被其他分组的信元分隔开,因此增加了对缓冲区的要求且操作复杂.为了克服这一问题,文献[25]将现有 MWM 算法扩展成基于分组的模式.我们将该算法称为 PB(packet based)-MWM,其基本思路是:对一个由 n 个信元组成的分组,如果输入端口开始传送该分组的第 1 个信元,那么在随后的 $n-1$ 个时间片内,始终保持该输入-输出连接.在任何容许的独立的 Bernoulli 到达下,只要分组的长度是有限的,PB-MWM 算法就可以达到 100%的吞吐率.文献[26]将文献[25]的信元到达模式由 Bernoulli 扩展到更普通的容许模式,并发现存在一个反例,使得 PB-MWM 算法变得不稳定.为了在分组模式下实现稳定的调度,文献[26]提出一种基于“等待”的 MWM 算法,并证明了算法的稳定性.

文献[27]的作者发现,LQF 和 LPF 算法在多节点的环境下不能达到稳定.为了解决这个问题,文献[27]提出了 LIN(longest-in-network)算法.但是 LIN 的计算复杂度高且不易扩展到更一般的通信量模式.在文献[28]中,Leonardi 等人首先提出一类调度算法($F(x)$-max-scalar),将目前的算法由面向单类通信量模式扩展到多类,然后将该算法由面向单节点扩展到多个互连的节点.无论是单/多节点,在容许的多类通信量下,该算法都是稳定的.

2.2.5 理论研究

对输入排队路由器调度算法性能的理论研究主要基于 Lyapunov 函数和流体模型这两种方法.

在稳定性方面,文献[9,29]应用 Lyapunov 函数来发现在独立的信元到达下调度算法的稳定区域,而文献[30]将信元的到达扩展到更一般的模式,并利用流体模型证明:(1) 任何采用最大权重匹配算法的 VOQ 路由器都可以达到 100%的吞吐率;(2) 在加速比大于 2 的情况下,任何采用极大权重匹配算法的 CIOQ 路由器都可以达到 100%的吞吐率.

在延迟方面,文献[31]利用 Lyapunov 函数分析了最大权重匹配算法信元延迟的均值以及队列长度的均值和方差.文献[32]将文献[31]的结果从最大权重匹配扩展到一类近似最大权重匹配算法 1-APRX.在这类算法中,一个调度算法的权重 W 与最大权重匹配调度算法的权重 W^* 相差最多 $f(W^*)$,这里,$f(W^*)$ 是一个次线性函数.这两类算法权重的差异记为“近似距离”.在任何容许的独立 Bernoulli 到达下,1-APRX 算法可以达到 100%的吞吐率.1-APRX 算法的延迟限度与近似距离呈线性关系,即权重的差异越小,近似算法的性能越好.因此,我们可以利用近似距离指导近似最大权重匹配算法的设计.

2.2.6 小 结

我们从算法的复杂度和性能两个方面对最大匹配和最大权重匹配算法做一个简单的比较.(1) 在算法复杂度方面.MSM 和 MWM 的复杂度分别为 $O(N^{2.5})$和 $O(N^3\log N)$,在硬件实现上非常复杂且运行时间长.但是采用多次迭代的近似算法平均经过 $\log N$ 次迭代就可以收敛到极大匹配,因此得到实际的应用;(2) 在性能方面.MSM 在均匀的通信量下可以到达 100%的吞吐率,但在非均匀的通信量下,算法就会变得不稳定.而对 MWM 来说,只要通信量是容许的,无论是否均匀,都可以达到 100%的吞吐量.近似算法具有类似的性能.

2.3 基于稳定婚姻的算法

稳定婚姻问题是一种二分图的匹配,最早由 Gale 和 Shapley 提出[33].已有的解决该问题的算法是 GSA(gale-shapley algorithm),算法复杂度的下限是 $\Omega(N^2)$[34].在输入排队调度算法中,GSA 算法利用输入和输出端口定义的优先清单寻找稳定的输入-输出匹配.优先清单主要用来解决输入/输出端口的访问冲突.我们说一个匹配是稳定的,是指所有已完成匹配的输入和输出端口,在没有完成匹配的输出和输入端口集合中不能发现一个端口,其优先级比已匹配的端口要高.

MUCFA(most urgent cell first algorithm)[35]算法利用 GSA 算法和输入/输出优先清单在输入/输出端口之间

© 中国科学院软件研究所 http://www.jos.org.cn

发现一个稳定婚姻匹配.输出端口 j 根据队列 Q_{ij} 头信元的紧急值为输入端口 i 赋一个优先值.信元的紧急值定义为在仿真的 FIFO 输出排队队列中,排在该信元前面的信元的个数.同样,每个输入端口根据头信元的紧急值为每个输出端口赋一个优先值并建立相应的优先清单.采用 MUCFA 的 CIOQ 路由器在加速比为 4 时能够准确地仿真一个 FIFO OQ 路由器.

在 JPM(joined preferred matching)[36]和 CCF(critical cell first)[37]算法中,输入优先清单的信元分别按到达时间的反序和输出占用的增序排列.一个信元的输出占用是指该信元的目的输出队列中等待转发的信元的个数.与 MUCFA 算法一样, JPM 和 CCF 算法的输出优先清单的信元按紧急值排列.JPM 和 CCF 算法从两个方面加强了 MUCFA 算法的结论:(1) 只需要 2 倍的加速比;(2) 允许仿真 FIFO 以及其他输出排队的调度算法.

LOOFA(lowest output occupancy first algorithm)[38]算法的输入优先清单与 CCF 算法相同.输出优先清单按信元到达时间排列.当加速比为 2 时,采用 LOOFA 的路由器是连续工作(work-conserving)的,因此能够提供与 OQ 相同的吞吐率.此外,LOOFA 可以在传输流级限制每个分组的传输延迟.表 2 主要从技术特点和算法性能两个方面比较了以上算法.

Table 2 Comparison of stable marriage algorithms

表 2 稳定婚姻算法比较

Algorithm	Input preference list	Output preference list	Speedup	Complexity
MUCFA	Urgent value	Urgent value	4	$\Omega(N^2)$
JPM	Arrival time	Urgent value	2	$\Omega(N^2)$
CCF	Output occupancy	Urgent value	2	$O(N)$
LOOFA	Output occupancy	Arrival time	2	$O(N^2)$

在文献[39]中,作者提出在无加速比基础上为输入排队路由器提供 QoS 保证的 3 种算法,分别利用加权信用、有效信元和有效等待时间作为边的权重,用来保证带宽预留、信元延迟和公平共享非保留的交换容量等 QoS 指标.调度算法也是基于稳定婚姻匹配.

最后,我们将稳定婚姻和最大(权重)匹配作比较[19]:(1) 稳定婚姻与最大(权重)匹配有很大的不同,目前还不清楚稳定婚姻的带宽利用率以及是否会导致饿死,而我们知道 GSA 算法会偏重于输入或输出的某一方;(2) 稳定婚姻问题和 GSA 算法通常定义为找到在输入和输出间的一个完全匹配方案,而在交换结构下并不总能找到完全匹配方案,结果可能导致稳定婚姻数的大幅降低[34];(3) 与多次迭代算法不同,在 GSA 算法中,一个已建立的连接在随后的迭代中可能被拒绝;(4) 稳定婚姻不一定有最大权重,而最大权重匹配也不一定是稳定婚姻[39].

实际上,最大(权重)匹配与稳定婚姻算法匹配过程都包括两部分:输出选择和输入选择.在输出端口,两种算法都是根据某种优先级别选择输入端口(即输入选择).但是在输入端口,当有多个输出端口同时选择一个输入端口时,两种算法作出输出选择的方式不同:最大(权重)匹配一般采用随机的或顺序的方式,而稳定婚姻算法仍然是根据信元的优先级.研究表明[40],利用优先级实现的输入排队调度算法能够获得更好的最大吞吐率和延迟限度.因此在设计算法时,应重点考虑如何利用信元的有用信息和要保证的 QoS 指标来设计信元的优先级.

2.4 确定型调度算法

确定型调度算法的基本思想是在 N 个时间片内确定地服务一个队列一次.一种可能的确定型调度算法的实现是在时间片 t 时,输入端口 i 和输出端口($(t+i) \bmod N$)相连,而无论输入端口 i 是否有要发送的信元.研究表明,对一个均匀的独立 Bernoulli 到达,这种简单的轮转算法可以达到 100%的吞吐率.

本文介绍的确定型调度算法主要基于由 Chang 等人提出的 Birkhoff-von Neuman 输入排队交换机(以下简称 BvN 交换机)[41].这种交换机的调度算法采用 Birkhoff 和 von Neuman 矩阵分解方法.具体而言,对通信量矩阵 $\Lambda=[\lambda_{ij}]$,如果满足条件:

$$\sum_{i=1}^{N}\lambda_{ij} \le 1, j=1,2,...,N\ ,\quad \sum_{j=1}^{N}\lambda_{ij} \le 1, i=1,2,...,N\ , \tag{5}$$

那么存在一个正数集 φ_k 和服务矩阵集 S_k,k=1,…,K,其中 $K \le N^2-2N+2$,使得

$$\Lambda \le \sum_{k=1}^{K}\varphi_k S_k\ ,\quad \sum_{k=1}^{K}\varphi_k = 1\ . \tag{6}$$

© 中国科学院软件研究所 http://www.jos.org.cn

矩阵分解的计算复杂度为 $O(N^{4.5})$.根据得到的分解,交换机可以按照服务矩阵 S_k 的权重 $\varphi_k,k=1,\ldots,K$,调度连接的模式 S_k.BvN 交换机使用的在线调度算法是 PGPS[5]算法的一个简化版本.特别地,如果对所有的 $k,\varphi_k=1/K$,那么算法按周期 K 产生一个周期性的连接模式序列.研究表明[41],如果分配的带宽大于每个输入-输出连接的到达速率,BvN 交换机可以达到 100%的吞吐率.但由于不能事先确定通信量矩阵 Λ,因此只能通过测量或估计的方法来获得这些信息.此外,矩阵分解算法过于复杂,不适合端口数多的交换机.

在文献[42]中,Chang 等人提出了一种较简单的两级交换机体系:负载平衡的 BvN 交换机(load balanced Birkhoff-von Neuman,简称 LB-BvN 交换机).第 1 级执行负载平衡,第 2 级采用 BvN 输入排队交换结构,对已完成负载平衡的通信量按照一个周期性的连接模式序列进行交换,周期等于输入/输出端口数 N.具体而言,设服务矩阵 S 是一个单循环的 $N\times N$ 置换矩阵.一个典型的单循环置换矩阵是一个循环移位矩阵:当 $j=(i+1) \bmod N$ 时,$s_{i,j}=1$;否则,$s_{i,j}=0$.在式(6)中,我们设 $S_k=S^k,\varphi_k=1/N$,其中 $k=1,\ldots,N$.由于 S 是一个单循环置换矩阵,S^N 是一个单位矩阵,BvN 交换机中的类 PGPS 算法只是按周期 N 设置连接模式.此外,在每 N 个时间片中,每个输入-输出连接就得到一个时间片,即分给每个输入-输出连接的速率是 $1/N$.这意味着如果 LB-BvN 交换机想要达到 100%的吞吐率,就必须要求到达第 2 级的通信量是均匀的,而这正是第 1 级所要完成的工作.

LB-BvN 交换机具有可扩展性好(算法的在线复杂度是 $O(1)$)、硬件实现复杂度低等优点,但有可能破坏同一输入端口信元的 FIFO 顺序关系,即造成信元失序.为了消除信元失序,Chang 等人在文献[43]中对两级交换机结构作了改进(如图 3 所示):在第 2 级之后增加了一个重排序和输出缓冲区,负责对失序的信元进行重排序和保存等待输出的信元.同时,在第 1 级之前增加一个传输流分离器和负载平衡缓冲区(图 3 中的 VOQ_1).我们以这种结构为例来说明 LB-BvN 交换机的运行过程.

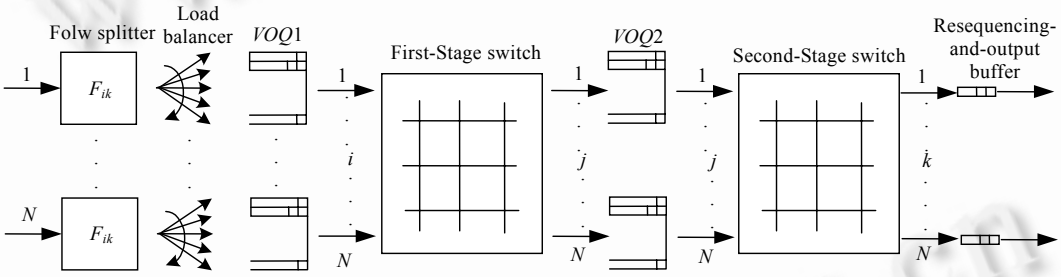

Fig.3 Two-Sage switch architecture

图 3 两级交换机结构

第 1 级的输入称为外部输入 EI_i(external input),$i=1\ldots N$.第 1 级的输出称为内部输出 IO_j(internal output),与第 2 级的输入,即内部输入 II_j(internal input)相连,$j=1\ldots N$.最后,第 2 级的输出称为外部输出 EO_k(external output),$k=1\ldots N$.一个信元在 LB-BvN 交换机内要经过以下步骤:

(1) 外部输入 EI_i 的一个信元首先被传输流分离器划归不同的传输流 F_{ik},这里,k 是信元的目的外部输出端口 EO.因此,每个 EI 最多可能有 N 个传输流,分别对应于 N 个 EO.

(2) 负载平衡器以轮转的方式将 F_{ik} 的所有信元发送到 N 个 VOQ_1(对应 N 个 IO)中.

(3) 第 1 级交换结构按确定的次序服务 VOQ_1 的队列.当一个队列接受服务时,信元离开 VOQ_1 并通过交换结构到达 VOQ_2 然后在 VOQ_2 排队等待服务.

(4) 第 2 级交换结构按确定的次序服务 VOQ_2 的队列.当一个队列接受服务时,信元离开 VOQ_2 并通过交换结构到达重排序和输出缓冲区.

为了限制失序信元数量,文献[43]还提出了两种解决方案,分别基于 FCFS(first come first served)和 EDF(earliest deadline first).FCFS 方案要求在第 2 级输入前的 VOQ_2 中增加复杂抖动控制机制,而 EDF 方案要从第 2 级输入队列中找出时戳最小的信元,这在一个高速交换机中很难实现.

在文献[44]中,Keslassy 等人从两个方面对以上结构作了进一步的改进:(1) 提出一种全帧优先(full frame first,简称 FFF)算法,使得信元的平均延迟比一个理想的 OQ 多一个常数,因此具有与 OQ 相同的吞吐率;(2) FFF

© 中国科学院软件研究所 http://www.jos.org.cn

算法采用一种三维队列(3DQ)结构来避免信元的失序,因此不需要重排序和输出缓冲区.但是,3DQ 排队结构比 VOQ 复杂,需要更多的缓冲区.

Chang 等人提出的两级交换结构不仅能够达到 100%的吞吐率并保证延迟限度,而且不需要加速比和复杂的调度算法,因此在设计高性能并保证延迟的交换机时,这种两级交换结构具有重要的参考价值.此外,这种两级交换结构不仅可以应用于电子的 Internet 路由器,而且还可以应用于光交换结构[44].但是,这种两级交换结构仍然存在一些开放的研究问题,其中包括如何设计一个简单的能够消除失序现象的调度算法,如何在确定型调度算法下实现灵活的带宽分配和 QoS 保证.

3 输入排队调度算法的发展趋势

本节主要从交换机体系结构和网络体系结构这两个方面讨论输入排队算法的发展趋势.

现有的交换机根据体系结构的不同可分为集中式和分布式两种.集中式的优点是调度方案简单、易于实施.但是,很难在一个短的时间片内实现调度.分布式的交换机由多个并行的带独立调度器的交换结构组成.一个负载平衡算法将每个输入端口的信元发送到某个交换结构,然后由交换结构决定何时转发信元到输出端口.分布式结构的最大优点是降低了交换结构的输入/输出链路速率,从而增大了时间片,因此可以使用复杂的调度算法.现有的采用分布式结构的交换机包括 ADSA[45]和 PPS(parallel packet switch)[46,47].与 PPS 相比,ADSA 具有算法简单和控制信息少的优点.但这种分布式结构性能需要在实际网络环境中做进一步的测试,其负载平衡算法也需要深入研究.在网络体系结构方面,DiffServ(differentiated services)[48]通过在聚集通信量的水平上提供 QoS 而被认为是下一代 Internet 结构的基础.前面介绍的大部分调度算法只考虑输入-输出对之间的 QoS 保证,并没有考虑如何在聚集上提供 QoS 和分配带宽.在将来的调度算法研究中,应考虑 DiffServ 中已定义的各种服务种类.关于这方面的研究见文献[28].

此外,在文献[49,50]中还提出了基于帧(frame-based)或包迹(envelope-based)的输入排队调度算法.其基本思想是将同一个队列中相邻的若干信元(分组)组成帧(包迹),从而增加调度算法时间片,使复杂调度算法的实现成为可能.这也是提高输入排队路由器可扩展性的一种可行的方法.

总之,未来的输入调度算法首先要适应网络带宽的快速增长,具有良好的可扩展性和性能,不能成为网络传输的瓶颈,同时,在延迟、公平性和服务类型的多样化等方面也要有较好的保证.

4 结 论

网络带宽技术和多媒体应用的不断发展对网络互联的核心设备——路由器的性能要求越来越高.输入排队策略解决了路由器的可扩展性问题,但其性能受输入排队调度算法的制约.如何设计一个既能提供高吞吐率又能在硬件上实现简单的输入排队调度算法,是保证路由器性能的关键.

本文首先综述了输入排队调度算法,将现有的算法分为 4 类:最大匹配、最大权重匹配、稳定婚姻匹配和确定型调度.然后对每一类算法从技术特点和性能两个方面进行比较,指出它们的相似性和不同之处.在这 4 类算法中,最大匹配、最大权重匹配和稳定婚姻匹配都是基于二分图的匹配算法,能实现 100%的吞吐率.稳定婚姻匹配算法通过优先调度算法使输入排队路由器能够提供与输出排队路由器类似的服务质量保证.利用随机化算法实现的近似最大权重匹配避免了多次迭代,能更快地发现一个最大权重匹配.确定型调度算法主要基于矩阵分解技术,采用两级交换结构,在无加速比的情况下提供 100%的吞吐率和保证延迟限度,因此对未来的高性能路由器设计具有重要的参考价值.

References:

[1] Bux W, Denzel WE, Engberson T, Herkersdorf A, Luijten RP. Technologies and building blocks for fast packet forwarding. IEEE Communication Magazine, 2001,39(1):70~77.

[2] Nong G, Hamdi M. On the provision of quality-of-service guarantees for input queued switches. IEEE Communications Magazine, 2000,38(12):62~69.

© 中国科学院软件研究所 http://www.jos.org.cn

[3] Javidi T, Magill R, Hrabik T. A high-throughput scheduling algorithm for a buffered crossbar switch fabric. In: Neuvo Y, ed. Proceedings of the IEEE International Conference on Communications (ICC). Helsinki: IEEE Communications Society, 2001. 1586~1591.

[4] Prabhakar B, McKeown N, Ahuja R. Multicast scheduling for input-queued switches. IEEE Journal on Selected Areas in Communications, 1997,15(5):855~866.

[5] Parekh AK, Gallager RG. A generalized processor sharing approach to flow control in integrated service networks: the single-node case. IEEE/ACM Transactions on Networking, 1993,1(3):344~357.

[6] Karol M, Hluchyj M, Morgan S. Input versus output queueing on a space division switch. IEEE Transactions on Communication, 1988,35(12):1347~1356.

[7] Tamir Y, Frazier G. Dynamically-Allocated multi-queue buffer for VLSI communication switches. IEEE Transactions on Computers, 1992,41(6):725~737.

[8] Hopcroft J E, Karp RM. An $n^{5/2}$ algorithm for maximum matching in bipartite graphs. SIAM Journal on Computing, 1973,1.2: 225~231.

[9] McKeown N, Mekkittikui A, Anantharam V, Walrand J. Achieving 100% throughput in an input-queued switch. IEEE Transactions on Communication, 1999,47(8):1260~1267.

[10] Anderson T, Owicki S, Saxes J, Thacker C. High speed switch scheduling for local area networks. ACM Transactions on Computer Systems, 1993,11(4):319~352.

[11] McKeown N. The iSLIP scheduling algorithm for input-queued switches. IEEE/ACM Transactions on Networking, 1999,7(2): 188~201.

[12] McKeown N, Anderson TE. A quantitative comparison of scheduling algorithms for input-queued switches. Computer Networks and ISDN Systems, 1998,30(24):2309~2326.

[13] Serpanos DN, Antoniadis PI. FIRM: A class of distributed scheduling algorithms for high-speed ATM switches with multiple input queues. In: Sidi M, ed. Proceedings of the IEEE INFOCOM. Tel Aviv: IEEE Communications Society, 2000. 548~555.

[14] Chao HJ. Saturn: A terabit packet switch using dual round robin. IEEE Communications Magazine, 2000,38(12):78~84.

[15] Goudreau MW, Kolliopoulos SG, Rao SB. Scheduling algorithms for input-queued switches: Randomized techniques and experimental evaluation. In: Sidi M, ed. Proceedings of IEEE INFOCOM. Tel Aviv: IEEE Communications Society, 2000. 1634~1643.

[16] Tarjan RE. Data structures and network algorithms. SIAM, 1983.

[17] Mekkittikui A, McKeown N. A starvation-free algorithm for achieving 100% throughput in input-queued switches. In: Lee D, ed. Proceedings of the IEEE International Conference on Computer Communications and Networks (ICCCN). Rockville, MA: IEEE Communications Society, 1996. 226~231.

[18] Mekkittikui A, McKeown N. A practical scheduling algorithm to achieve 100% throughput in input-queued switches. In: Akyildiz I, ed. Proceedings of the IEEE INFOCOM. San Francisco: IEEE Communications Society, 1998. 792~799.

[19] McKeown N. Scheduling algorithms for input-queued switches [Ph.D. Thesis]. University of California at Berkeley, 1995.

[20] Marsan MA, Bianco A, Leonardi E, Milla L. RPA: A flexible scheduling algorithm for input buffered switches. IEEE Transactions on Communications, 1999,47(12):1921~1933.

[21] Duan H, Lockwood JW, Kang SM, Will JD. A high performance OC12/OC48 queue design prototype for input buffered ATM switches. In: Hasegawa T, ed. Proceedings of the IEEE INFOCOM. Kobe: IEEE Communications Society, 1997. 20~28.

[22] Marsan MA, Bianco A, Giaccone P, Leonardi E, Neri F. Input-Queued router architectures exploiting cell-based switching fabrics. Computer Networks, 2001,37(5):541~559.

[23] Tassiulas T. Linear complexity algorithms for maximum throughput in radio networks and input queued switches. In: Akyildiz I, ed. Proceedings of the IEEE INFOCOM. New York: IEEE Communications Society, 1998. 533~539.

[24] Giaccone P, Prabhakar B, Shah D. Towards simple, high-performance schedulers for high-aggregate bandwidth switches. In: Kermani P, ed. Proceedings of the IEEE INFOCOM. New York: IEEE Communications Society, 2002. 1160~1169.

[25] Marsan MA, Bianco A, Giaccone P, Neri F. Packet scheduling in input-queued cell-based switches. In: Sengupta B, ed. Proceedings of the IEEE INFOCOM. Anchorage: IEEE Communications Society, 2001. 1085~1094.

© 中国科学院软件研究所 http://www.jos.org.cn

[26] Ganjali K, Keshavarzian A, Shah D. Input queued switches: Cell switching vs. packet switching. In: Bauer T, ed. Proceedings of the IEEE INFOCOM. San Francisco: IEEE Communications Society, 2003. http://www.stanford.edu/~yganjali/#Publications.

[27] Andrews M, Zhang L. Achieving stability in networks of input-queued switches. In: Sengupta B, ed. Proceedings of the IEEE INFOCOM. Anchorage: IEEE Communications Society, 2001. 1673~1679.

[28] Leonardi E, Mellia M, Marsan MA, Neri F. On the throughput achievable by isolated interconnected input-queueing switches under multiclass traffic. In: Kermani P, ed. Proceedings of the IEEE INFOCOM. New York: IEEE Communications Society, 2002. 1605~1614.

[29] Leonardi E, Mellia M, Neri F, Marsan MA. On the stability of input-queued switches with speedup. IEEE/ACM Transactions on Networking, 2001,9(1):104~118.

[30] Dai JG, Prabhakar, B. The throughput of data switches with and without speedup. In: Sidi M, ed. Proceedings of the IEEE INFOCOM. Tel Aviv: IEEE Communications Society, 2000. 556~564.

[31] Leonardi E, Mellia M, Neri F, Marsan MA. Bounds on average delays and queue size averages and variances in input-queued cell-based switches. In: Sengupta B, ed. Proceedings of the IEEE INFOCOM. Anchorage: IEEE Communications Society, 2001. 1095~1103.

[32] Shah D, Kopikare M. Delay bounds for the approximate maximum weight matching algorithm for input queued switches. In: Kermani P, ed. Proceedings of the IEEE INFOCOM. New York: IEEE Communications Society, 2002. 1024~1031.

[33] Gale D, Shapley LS. College admission and the stability of marriage. American Mathematical Monthly, 1962,69:9~15.

[34] Gusfield D, Irving R. The Stable Marriage Problem: Structure and Algorithms. The MIT Press, 1989.

[35] Prabhakar P, Mckeown N. On the speedup required for combined input and output queued switching. Technical Report, Stanford CSL-TR-97-738, 1997.

[36] Stoica I, Zhang H. Exact emulation of an output queueing switch by a combined input and output queueing switch. In: Knightly E, ed. Proceedings of the IEEE IWQoS. Napa: IEEE Communications Society, 1998. 218~224.

[37] Chuang ST, Goel A, McKeown N. Matching output queueing with a combined input/output-queued switch. IEEE Journal on Selected Areas in Communications, 1999,17(6):1030~1039.

[38] Krishna P, Patel NS, Charny A, Simcoe RJ. On the speedup required for work-conserving crossbar switches. IEEE Journal on Selected Areas in Communications, 1999,17(6):1057~1066.

[39] Kam AC, Siu KY. Linear-Complexity algorithms for QOS support in input-queued switches with no speedup. IEEE Journal on Selected Areas in Communications, 1999,17(6):1040~1056.

[40] Weller T, Hajek B. Scheduling nonuniform traffic in a packet-switching system with small propagation delay. IEEE/ACM Transactions on Networking, 1997,5(6):813~823.

[41] Chang CS, Chen WJ, Huang HY. Birkhoff-von Neumann input buffered crossbar switches. In: Sidi M, ed. Proceedings of the IEEE INFOCOM. Tel Aviv: IEEE Communications Society, 2000. 1614~1623.

[42] Chang CS, Lee DS, Jou YS. Load balanced Birkhoff-von Neumann switches Part I: One-stage buffering. Computer Communications, 2002,25(6):611~622.

[43] Chang CS, Lee DS, Lien CM. Load balanced Birkhoff-von Neumann switches Part II: Multi-Stage buffering. Computer Communications, 2002,25(6):623~634.

[44] Keslassy I, McKeown N. Maintaining packet order in two-stage switches. In: Kermani P, ed. Proceedings of the IEEE INFOCOM. New York: IEEE Communications Society, 2002. 1032~1041.

[45] Wang W, Dong L, Wolf W. A distributed switch architecture with dynamic load-balancing and parallel input-queued crossbars for terabit switch fabrics. In: Proceedings of IEEE INFOCOM. New York: IEEE Communications Society, 2002. 352~361.

[46] Iyer S, Awadallah A, McKeown N. Analysis of a packet switch with memories running slower than the line-rate. In: Sidi M, ed. Proceedings of the IEEE INFOCOM. Tel Aviv: IEEE Communications Society, 2000. 529~537.

[47] Iyer S, McKeown N. Making parallel packet switches practical. In: Sengupta B, ed. Proceedings of the IEEE INFOCOM. Anchorage: IEEE Communications Society, 2001. 1680~1687.

[48] Blake S, Black D, Carison M, Davies E, Wang Z, Weiss W. An architecture for differentiated services. IETF RFC 2475, 1998.

[49] Bianco A, Franceschinis M, Ghisolfi S, Hill AM, Leonardi E, Neri F, Webb R. Frame-Based matching algorithms for input-queued switches. In: Aoyama T, ed. Proceedings of the IEEE Workshop on High Performance Switching and Routing (HPSR). Kobe: IEEE Communications Society, 2002.

[50] Kar K, Lakshman TV, Stiliadis D, Tassiulas L. Reduced complexity input buffered switches. In: Proceedings of the Hot Interconnects VIII. 2000. http://www.bell-labs.com/user/stiliadi/publications.html.

© 中国科学院软件研究所 http://www.jos.org.cn