线路速率缓存的重端口交换机方案及行为分析*

吴 俊+, 陈 晴, 罗军舟

(东南大学 计算机科学与工程系 网络室,江苏 南京 210096)

A Scheme and Behavior Analysis of Duplicated Ports Switches with Line Rate Buffers

WU Jun+, CHEN Qing, LUO Jun-Zhou

(Network Laboratory, Department of Computer Science and Engineering, Southeast University, Nanjing 210096, China)

+ Corresponding author: Phn: 86-25-3795595, Fax: 86-25-3794838, E-mail: j_wu@seu.edu.cn

http://cse.seu.edu.cn

Received 2002-11-15; Accepted 2003-06-04

Wu J, Chen Q, Luo JZ. A scheme and behavior analysis of duplicated ports switches with line rate buffers. *Journal of Software*, 2003,14(12):2060~2067.

http://www.jos.org.cn/1000-9825/14/2060.htm

Abstract: Nowadays, Internet is facing two challenges simultaneously: the need of a higher switching speed and the provision of a QoS guarantee. The former requires that memories in switches/routers should work at a line rate, and the latter requires that switches should mimic OQ (output queuing) switches. The present CIOQ (combined input-output queuing) scheme for switches needs a speedup of 2. In this paper, a novel design scheme based on parallel technique is proposed, which is called DPS (duplicated ports switch). DPS enables the internal fabric of switches to work at the line rate. A theoretical proof shows that each port with a duplicated number of 2 is sufficient to mimic the OQ switches and is necessary in practice.

Key words: switch; input-queueing; output-queueing; scheduling algorithm

摘 要: 现今 Internet 中的交换机/路由器面临着高交换速率和提供 QoS 保证的双重挑战.前者要求交换机/路由器的缓冲存储器尽可能地以链路速率工作,后者要求交换机能够完全模仿 OQ(output queuing)交换机的行为.而目前的 CIOQ(combined input-output queuing)设计方案需要交换机内部加速 2 倍.提出了采用并行技术的重端口交换机(duplicated ports switch,简称 DPS)设计方案.该方案可以使交换机工作于输入链路的速率且其行为与 OQ 交换机的行为等价,并证明了为完全模仿 OQ 交换机行为,端口重数为 2 是充分必要的.

关键词: 交换机;输入对列;输出队列;调度算法

中图法分类号: TP393 **文献标识码**: A

随着光通信技术的发展和多媒体技术的广泛使用,如今的Internet面临着两个主要问题:日益增加的网络链

* Supported by the National Natural Science Foundation of China under Grant No.90204009 (国家自然科学基金); the National Grand Fundamental Research 973 Program of China under Grant No.G1998030402 (国家重点基础研究发展规划(973))

第一作者简介: 吴俊(1970－),男,江苏扬州人,博士生,讲师,主要研究领域为高性能网络,协议工程.

路传输速率和提供 QoS 保证.解决这两个问题都涉及网络的基础交换/寻径结构.这种基础交换/寻径结构是由交换机或路由器互连而成,因此支持高速率且提供QoS保证的交换机/路由器成了下一代Internet实现的主要瓶颈之一.交换机/路由器中负责将输入链路上的数据转发到输出链路的部件称作交换结构(switching fabric).它是交换机的核心,直接决定了交换机的行为.交换结构一般由交换阵列和缓存器组成.现有的交换机的交换结构大多是基于输出队列(output queuing)技术的,即缓存器设置在交换阵列的输出端.虽然输出队列交换机仅需简单的分组调度算法,如 FIFO,即可获得很好的性能,但难以扩展,即对 $N\times N$ 的交换结构而言,交换阵列和缓存需要工作于 N 倍的线路速率,不适用于高速率或多端口的场合,因此采用输入队列(input queuing)的交换机成为研究的热点.输入队列技术在分组到达交换机/路由器时将其缓存在输入端口的缓存中,输出端口不设置缓存.与OQ(output queuing)相比,IQ 最大的优点是交换结构的工作速率与线路速率相同,避免了 OQ 难以规模化的问题.但 IQ 技术也存在着两个弱点:(1) 存在 HOL(head of line)阻塞[1],影响交换机的吞吐率;(2) 难以控制分组的转发时延,即不易提供 QoS 支持.

目前,输入队列的 HOL 阻塞问题基本得到了解决,文献[2,3]中给出了基于虚输出队列和最大权匹配的 LQF 和 LPF 算法,并证明了这两种算法具有渐近 100%的吞吐率.但由于输入队列不仅存在输入端的竞争(HOL 阻塞),而且存在输出端的冲突,单纯地使用 IQ 技术难以控制分组转发的时延;另一方面,现有的提供 QoS 保证的输出调度算法(如 WFQ,WF^2Q 等)都是针对 OQ 交换机设计的.因此,既要使 IQ 交换机获得 100%的吞吐率,又要能提供 QoS 保证,就必须改进 IQ 交换机的设计,使其能完全模仿 OQ 交换机的行为.一种折衷的方案是将 $N\times N$ 交换结构的工作速率提高 $s(1<s<N)$倍,这样,在同一时刻,若有到 N 个分组去往同一输出端口,那么将可以有 s 个立刻转发至输出端口,剩余的 $N-s$ 个将留在输入端,因此输入、输出端都必须设有缓存,故称为 CIOQ(combined input and output queuing).文献[4]证明了 CIOQ 模仿 OQ 的充分条件是 $s\geq 4$,且 s 与交换机的端口数 N 无关.文献[5]给出了分析 CIOQ 交换机稳定性的一般方法,并证明了当 $s=2$ 时,RRD(random rate driven)和 LQD(longest queue driven)等多种调度算法均能获得 100%的吞吐率.文献[6]证明了,当 $s=2$ 时,采用 LOOFA(lowest occupancy output first algorithm)算法的 CIOQ 交换机是工作守恒的,并给出了分组的延迟界.文献[7]证明了,当 $s\geq 2$ 时,CIOQ 模仿 OQ 是充分的.文献[8]最终证明了 $N\times N$ 的 CIOQ 交换机模仿 OQ 交换机的充分必要条件是 $s=2-1/N$,即 CIOQ 模仿 OQ 交换机的实用加速下限是 2.

自 20 世纪 80 年代以来,网络的线路速率以平均每 7 个月提高 1 倍的速度增长,Tbps 的网络已处在实验中,甚至已在向 Pbps(10^{15} bps)的网络迈进[9].而商品存储器的读写速度却只以平均每 18 个月提高 1.1 倍的速率增长.这导致存储器成为下一代交换机实现的主要瓶颈.虽然加速的 CIOQ 方案只需加速 1 倍即可完全模仿 OQ 交换机的行为,但在将来的高速网络中,要对交换机的内部结构进行加速,其代价是巨大的,甚至是不可行的.基于此,本文提出了一种无须加速 CIOQ 交换机的设计方案——重端口交换机 DPS(duplicated port switch).

现有的高速分组交换机为了避免处理不定长分组引起的复杂性,通常在交换机内部将不定长分组切分成定长的短分组(类似于 ATM 的信元)进行交换,然后在输出端进行重新组装.因此下文约定“分组”是指定长分组,交换结构的工作时间划分成若干等长的时隙,分组仅在每个时隙的开始时到达.

1 DPS 交换结构

输入队列交换机的 HOL 阻塞严重影响了交换机的吞吐率.造成 HOL 阻塞的原因是,由于每个输入端口只有一条输入队列,在每个输入队列中的分组存在着转发冲突,正是这种冲突导致了 FCFS 服务规则的 HOL 阻塞.VOQ(virtual output queuing)技术可以成功地解决输入端的冲突问题,即在每个输入端口建立 N 个队列($N\times N$ 交换机),每个队列对应一个输出端口,这样,每个输入端口分组的所有转发可能都呈现在队列头部,从而避免了输入端的冲突.因此,在此基础上选择适当的调度算法可以获得 100%的吞吐率.但 VOQ 技术同样受到每个输出端口每个时隙只能接受一个分组的限制,因而该技术同样受到输出端冲突的困扰,从而难以控制分组的转发时延.解决输出冲突的途径是增加交换机的交换带宽,如加速的 CIOQ 技术.但如前所述,在高速网络中对存储器进一步加速是困难的,而交换机中的 cross-bar 交换结构则是容易大规模化的部件,因此我们提出了一个利用并行技术增加交换带宽的设计方案——重端口交换机 DPS.该方案与 VOQ 类似,通过输入端采用多队列来解决或减

轻输入端的冲突,同时通过增加交换带宽来解决输出端的冲突.

如图 1 给出的 2×2 的 DPS-2 交换机所示,一个 d 重 $N\times N$ 的 DPS,每个输入、输出端口都有 d 个缓冲存储器,每个存储器均有与交换结构的接口,即交换结构是 $dN\times dN$ 的,记为 DPS-d.由于 DPS 通过并行增加了交换带宽,输出端也需要缓存,因此也采用 CIOQ 技术.其工作过程如下:在每个时隙的开始时,若输入端口有分组抵达,则按一定的入队规则进入该端口 d 个队列中的一个进行排队;调度器决定交换阵列的配置,将相应的分组由输入端转发至输出端;输出调度器根据调度规则(如 FIFO,WFQ 等)决定离开的分组.

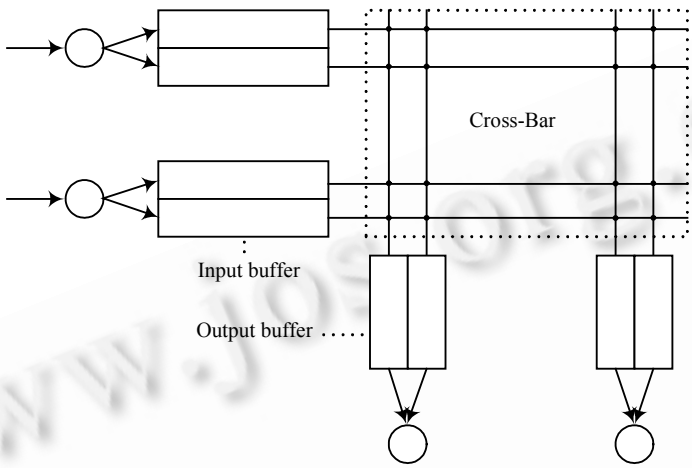


Fig.1 Illustration of a 2×2 DPS-2 switch architecture
图 1 2×2 的 DPS-2 交换机示意图

DPS 由于每个存储器的工作速率、交换阵列的速率都与线路速率相同,因此对调度算法有如下限制:

(1) 每个输入存储器每个时隙只能转发一个分组;

(2) 每个输出缓存器每个时隙只能接收一个分组.

这是 DPS 与加速的 CIOQ 的主要区别,加速 d 倍的 CIOQ 中输入和输出存储器每个时隙可以执行 d 次读或写操作.虽然 DPS-d 和加速 d 倍的 CIOQ 都将交换机的交换带宽增加到 d 倍,但 DPS 对交换带宽的加速要弱于加速的 CIOQ 方案.如图 2 所示,输入端口 1 有两个待转发分组,加速 2 倍的 CIOQ 可以在一个时隙内转发这两个分组,而 DPS-2 由于这两个分组处于同一个缓存中,所以在一个时隙内只能转发其中的一个.因此,下面要解决的问题是:在任何输入流模式下,对于任意 $N\times N$ 交换机 DPS-d 完全模仿 OQ 交换机的行为所需的最小 d 是多少.其中,CIOQ 交换机的行为与 OQ 交换机行为等价是指[6]:在相同的输入下,CIOQ 交换机分组的离开时间及离开顺序与 OQ 交换机的分组离开时间和离开顺序相同.

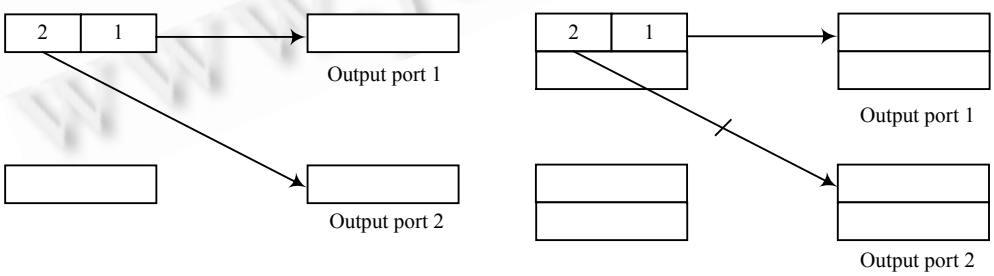


Fig.2 Comparison between the two CIOQ schemes
图 2 两种 CIOQ 方案的比较

2 入队规则及调度算法

2.1 入队规则

设 DPS-d 输入端口 i 的队列为 $Q_i[0],Q_i[1],\ldots,Q_i[d-1]$.在时隙 t,输入端口 i 有分组 p 到达,则 DPS-d 的输入队列的入队规则如下:

(1) p 进入队列 $Q_i[j]$, $j=t$ (mod) d,其中 mod 表示取模运算;

(2) p 存放在 $Q_i[j]$的最前面.

由上一节的介绍可知,DPS 的能力弱于加速的 CIOQ 的主要原因是 DPS 可能存在更多的输入端冲突,因此,输入队列的入队规则就显得非常重要.下一节我们将证明在上述的简单入队规则下,DPS-2 模仿 OQ 交换机的行为是充分的.

2.2 OQ交换机的行为

OQ 交换机的行为可以用两种输出队列的排队规则来描述[8]:

(1) FIFO,即先来先服务队列;

(2) PIFO(push in first out),分组按优先级插入队列,总是队头元素先接受服务.

仅提供 Best-Effort 服务的 OQ 交换机一般采用 FIFO 调度规则,其特点是,当分组到达交换机时,该分组的离开时间即可被指定,即后抵达的分组不会影响先抵达分组的离开时间.提供 QoS 保证的交换机目前采用的算法可分为两类:工作守恒的算法和非工作守恒算法[10].前者一般在分组抵达交换机/路由器后立即送往输出队列,并以输出链路服务于该分组的虚服务时间或期望的该分组到达时钟作为该分组接受服务的优先数,典型的有 WFQ,Virtual Clock[11,12]等.后者如 jitter-EDD,stop and go[13]等,其所用的优先数与工作守恒的调度算法类似,不同的是,在分组抵达后并不立即送往输出队列,而是等待一段时间,以控制抖动的积累.因此,在这两类调度算法下,分组在送到输出队列后,其优先数将不会随着随后到达的分组而变化.这就是所谓的单调特性[7],即新到达分组不影响已到达分组接受服务的相对顺序.显然,这一单调特性可以用 PIFO 服务规则来刻画.

下文我们假设当 DPS 模仿 FIFO 交换机时,DPS 具有计算 FIFO 交换机中分组离开时间的能力,即当分组 p 到达时,DPS 赋予 p 一个离开时间用 $D[p]$表示.当 DPS 模仿 PIFO 交换机时,在 DPS 中每一输出端口使用一个输出优先权表来跟踪被模仿的 OQ 交换机中分组离开顺序[8],每一滞留在交换机内的分组 p 在相应的优先权表内有一个表项,用 $D[p]$表示分组 p 在相应的优先权表中的位置.

2.3 调度算法

文献[7,8]显示稳定匹配算法能够保证离开时间为 0 的分组优先转发至输出端口,但 DPS-d 每个输入和输出端口有 d 个队列,且每个队列均有与交换结构的接口,因此,我们将稳定匹配算法进行扩充,以适应 DPS.$N\times N$ 的 DPS-2 调度算法如下:

Step 1. 初始时,未匹配的输入队列集 $IQ=\{Q_1[0],Q_1[1],\ldots,Q_n[0],Q_n[1]\}$,未匹配的输出队列集 $OQ=\{O_1[0],O_1[1],\ldots,O_n[0],O_n[1]\}$,未匹配的输出端口集 $O=\{1,2,\ldots,n\}$, $flag[1]=flag[2]=\ldots=flag[n]=0$($flag[i]=0$ 表示 $O_i[0]$未匹配);

Step 2. 为 O 中的每一元素 i 生成端口 i 允许发送信号送至 IQ 中的每一元素;

Step 3. 对所有 $Q_i[j]\in IQ$,若 $Q_i[j]$收到输出端口 k 的允许信号,则将 $Q_i[j]$中所有待转发至输出端口 k 的分组离开时间 $D[p]$发送至 $OQ\cap\{O_k[0],O_k[1]\}$;

Step 4. 对所有 $O_k[j]\in OQ$,若 $j=1$ 且 $flag[k]=0$,则 $O_k[j]$响应在收到的分组请求中离开时间第二小的分组,否则 $O_k[j]$响应在收到的分组请求中离开时间最小的分组,并将响应信号发至相应的输入队列;

Step 5. 对所有 $Q_i[j]\in IQ$,若收到 $O_k[m]$响应的分组 p 是该队列中收到响应的分组位置最前的一个,则 $Q_i[j]$选择 p 与 $O_k[m]$匹配;并将 $Q_i[j],O_k[m]$分别从 IQ 和 OQ 中删去,若 $m=0$,则置 $flag[k]=1$,若 $OQ\cap\{O_k[0],O_k[1]\}=\varnothing$,则将 k 从 O 中删去;

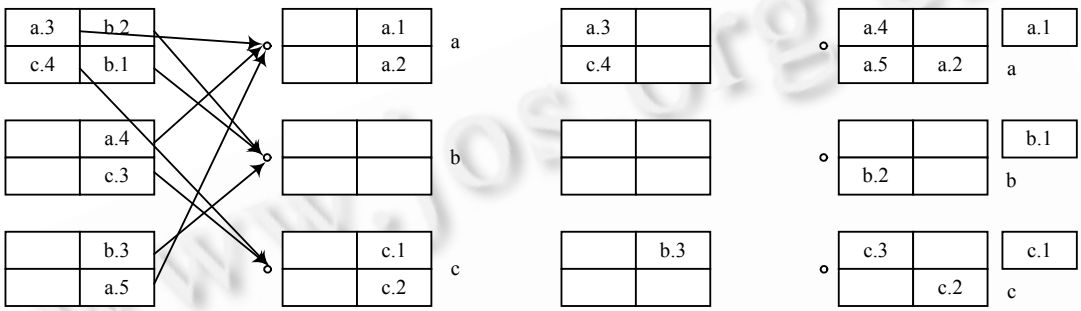
Step 6. 若 Step5 有新的匹配且 IQ 和 O 均非空,则转 Step2,否则迭代结束.

图 3 是该算法执行过程的示意图,图 3(a)是算法开始时队列所处的状态,图 3(b)是迭代一步后找到的分组与输出队列的匹配,其中分组 a.1 中的 a 表示转发端口,1 表示离开时间(或优先数).输入端口 1 的分组 a.3 虽然是请求转发至输出端口 a 的离开时间(或优先数)最小的分组,但由于其前面的分组 b.2 也收到了响应,所以 a.3 未获匹配;而输入端口 3 的分组 b.3 虽然处于队列的最前端,但由于输入端口 1 的分组 b.1 和 b.2 的离开时间(或优先数)小于 b.3,导致 b.3 在这次调度中也未能匹配.从该例容易看出,一个分组在上述调度算法结束后有 3 种可能:

(1) 该分组被转发至相应的输出端口;

(2) 与该分组同一队列、位置处于该分组前面的某个分组被转发,如图 3 中的分组 a.3;

(3) 与该分组不在同一输入队列但与该分组有相同转发端口且离开时间(或优先数)小于该分组的某两个分组被转发,如图 3 中的分组 b.3.



(a) The buffer state before scheduling (a) 调度前的缓存状态

(b) The buffer state after scheduling (b) 调度后的缓存状态

Fig.3 Illustration of the scheduling algorithm execution

图 3 调度算法执行过程示意图

加速 2 倍的 CIOQ 采用的稳定匹配算法的复杂性与交换阵列的规模 N 和输入队列中分组数有关.DPS-2 所采用的修改后的稳定匹配算法中队列内的分组数与之相同,交换阵列的规模扩大了 1 倍,但两个队列是并行执行的,因此算法复杂度与稳定匹配算法相同.并且,由于加速 2 倍的 CIOQ 交换机在每个时隙须执行稳定匹配算法两次,而 DPS-2 只须执行 1 次匹配算法,因此 DPS-2 总的调度复杂性小于加速 2 倍的 CIOQ 方案.

3 DPS-2 的充分性

对于 DPS 来说,不为整数的端口重数 d 没有实际意义,而 d=1 即为输入队列交换机,显然,不能模仿 OQ 的行为,因此从现实的角度看,d≥2 是必要的.下面我们将证明,当 d=2 时 DPS 可以充分模仿 OQ 交换机的行为.在给出 DPS-2 的充分性结论之前,先引入几个证明中将用到的概念:

定义 1(output cushion)[8]. 任何时隙,设处于输入队列中的分组 p 的转发端口为 k,p 的 Output Cushion 定义为输出端口 k 的队列(O_k[0]和 O_k[1])中离开时间(或优先数)小于 $D[p]$的分组个数,记为 $OC[p]$.

定义 2(input thread)[8]. 任何时隙,处于输入队列中的分组 p 的 Input Thread 定义为该输入队列中处于 p 前面的分组个数,记为 $IT[p]$.

定义 3(slackness). 任何时隙,处于输入队列中的分组 p 的 Slackness 记为 $S[p]$,$S[p]=OC[p]-2\times IT[p]$.分组 p 的 Slackness 表明了分组 p 被调度的紧迫程度,$S[p]$越小,被调度的紧迫性越高.

根据第 1 节给出的输入队列入队规则,在 d=2 时输入端口 i 的两个队列轮流有入队操作.下面约定在任何时隙输入端口 i,有入队操作的队列记为 i_{IN},无入队操作的队列记为 i_{OO}.用 $p\in i_{IN}$ 和 $p\in i_{OO}$ 表示相应队列中的分组.

引理 1. DPS-2 在第 1 节给出的调度算法和入队规则下,任一输入端口 i,在任一时隙 t 有 $\forall p\in i_{IN}$:$S[p]>0$(不包括 t 时隙新抵达分组)且 $\forall p\in i_{OO}$:$S[p]\geq 0$.

证明:用归纳法.

当 t=0 时,i_{OO} 和 i_{IN} 为空,引理结论显然成立.

设 $t=n$ 时引理成立,则:

(1) 对 $\forall p\in i_{OO}$,根据调度算法的性质,p 有 3 种可能.(a) p 被调度,则 n 时隙结束以后,p 已离开输入队列,无须再讨论 $S[p]$.(b) 与 p 同一队列中处于 p 前面的某个分组被调度,那么 $IT[p]$减 1,而 p 的目的输出端口有一个分组离开即 $OC[p]$至多减 1,因此,n 时隙后,$S[p]$至少增加 1.如图 3 中的分组 a.3,时隙开始时 IT[a.3]=1,OC[a.3]=2,S[a.3]=2−1×2=0.调度结束后,IT[a.3]=0,OC[a.3]=1,S[a.3]=1−0×2=1.(c) 与 p 有相同目的输出端口而相对离开时间小于 p 的两个分组获得匹配,那么 $OC[p]$增加 2,减去离开输出端口的一个分组,n 时隙后 $S[p]$增加 1.如图 3 中的分组 b.3,时隙开始时,IT[b.3]=0,OC[b.3]=0,S[b.3]=0−0×2=0,调度结束后,IT[b.3]=0,OC[b.3]=1,S[b.3]=1−0×2=1.由归纳假设可知,n 时隙开始时,$\forall p\in i_{OO}:S[p]\geq 0$;而 n 时隙结束时,$S[p]$至少增加 1,即 $S[p]>0$.由入队规则,n 时隙的 i_{OO} 是 n+1 时隙的 i_{IN},即在 n+1 时隙开始时,$\forall p\in i_{IN}:S[p]>0$.

(2) 对 $\forall p\in i_{IN}$,由于新抵达的分组插在队列的最前端,因此 $IT[p]$增加 1.与(1)类似,p 也有 3 种可能:(a) p 被调度,无须再考虑;(b) 与 p 同一队列中处于 p 的前面的某个分组被调度,那么 $IT[p]$又减 1,而 p 的目的输出端口有一个分组离开即 $OC[p]$至多减 1,因此 n 时隙后 $S[p]$至多减少 1;(c) 与 p 有相同目的输出端口而相对离开时间小于 p 的两个分组获得匹配,那么 $OC[p]$增加 2,减去离开输出端口的一个分组,n 时隙后 $S[p]$减少 1.对于新抵达的分组 p_{new},$IT[p_{new}]$为 0,所以,p_{new} 要么被转发至输出端口,要么 $S[p_{new}]\geq 1$(同(1)中的可能(c)).由归纳假设可知,在 n 时隙开始时,$\forall p\in i_{IN}:S[p]>0$,因此 n 时隙结束时 $S[p]\geq 0$.由入队规则,在 n+1 时隙开始时,$\forall p\in i_{OO}:S[p]\geq 0$.

综合(1)(2)可知,若 $t=n$ 时引理成立,则 $t=n$+1 时引理亦成立.证毕. □

定理 1. 若 $N\times N$ 的 DSP-2 工作于第 1 节所述的入队规则和调度算法下,则 DSP-2 与采用 FIFO 输出调度的 OQ 交换机行为等价.

证明:用归纳法.

当时隙 $t=0$ 时,定理显然成立.

设直到时隙 $t=n$ 时,DSP-2 完全模仿了 OQ 交换机的行为.因此,对于输入队列中的任何分组 p,若其离开时间为 0,那么其 $OC[p]=0$.

当 $t=n$+1 时,我们先证明任意输入队列中离开时间为 0 的分组最多只有 1 个.下面分两种情况加以讨论:

(1) 该队列有入队操作.由引理 1 可知,该队列的任一分组 p:$S[p]>0$,所以 $OC[p]>0$,即 p 的离开时间不为 0.因此,该队列中至多只有新入队的分组离开时间为 0.

(2) 该队列无入队操作.若该队列有多于 1 个分组的离开时间为 0,不失一般性,设有 p_1,p_2 的离开时间为 0,且 p_1 在队列中的位置处于 p_2 之前,那么 $OC[p_2]=0$,而 $IT[p_2]>0$,因此 $S[p_2]<0$.这与引理 1 矛盾,所以,在该队列中也最多只有 1 个分组的离开时间为 0.

这保证了输入队列端没有冲突.下面只要证明调度算法能将离开时间为 0 的分组在这一时隙调度至输出端口.由于每个输出端口每个时隙最多只能离开 1 个分组,因此,输入端口离开时间为 0 的分组在输出端也没有冲突.而且同(2)的讨论,$S[p]$为 0 的分组必然处于某个队列的最前面.这样,根据调度算法的性质,所有处于输入端的离开时间为 0 的分组均可以在这一时隙被调度,即可以模仿 OQ 的离开时间和顺序.而原本已在输出队列中的离开时间为 0 的分组显然也能模仿 OQ 的离开时间和顺序.因此,在 $t=n$+1 时隙,DPS-2 也完全模仿了 OQ 交换机的行为.定理得证. □

为了能在 DPS-2 交换机上应用 WFQ,WF^2Q 等提供 QoS 保证的输出队列调度算法,仅能模仿 FIFO 的 OQ 交换机行为是不够的.然而这一类算法都属于 PIFO 队列调度规则,具有单调特性,即后抵达分组不改变队列中其他分组的相对关系.这使得新抵达交换结构输入端的分组不影响输入队列中其他分组的 OC 值,保证了引理 1 在模仿 PIFO 交换机时同样成立.因此,定理 1 的证明也适用于模仿 PIFO 的情况,只是分组离开优先权的计算算法不同.因此我们有以下定理:

定理 2. 若 $N\times N$ 的 DSP-2 工作于第 2 节所述的入队规则和调度算法之下,那么 DPS-2 与采用 PIFO 输出调度规则的 OQ 交换机行为等价.

证明:同定理 1 的证明,此处从略. □

如前所述,无论是工作守恒或是工作不守恒的现有分组 QoS 调度算法都属于 PIFO 服务规则,因此,定理 1

和定理 2 说明,本文提出的重端口交换机在端口重数大于等于 2 时可以充分模仿任何输出队列交换机的行为.

4 扩展性讨论

可扩展性是进行高性能交换机设计时所要考虑的一个重要方面.影响交换机设计扩展性的因素有 3 个方面:一是对存储器带宽的要求;二是对供电系统的要求;三是对芯片集成度的要求.其中最关键的是存储器的带宽,这是目前高性能交换机设计的瓶颈.DPS 方案由于缓存工作于连路速率,因此在对存储器的要求方面,DPS 的扩展性优于传统的 OQ 和内部加速的 CIOQ 设计方案.从耗能的角度来看,交换机的耗能和交换机的规模及交换机的工作速率成正比.DPS 的交换阵列虽然比加速的 CIOQ 所需要的大,但由于 DPS 交换阵列的工作速度低,因此,其总的能耗与加速的 CIOQ 相仿.从对集成度的要求来看,DPS 所需要的交换阵列较大,对芯片集成度的要求较高.但这并不影响 DPS 的应用,正如 CIOQ 对芯片的集成度要求远高于 OQ 但并不妨碍 CIOQ 的广泛使用一样.这主要有如下两方面的原因:

(1) 由于集成电路技术正按摩尔定律揭示的速度发展,而交换机端口数的需求却不会有太大的变化,因此,芯片集成度的限制是交换机设计中的次要因素.

(2) 高性能交换机主要应用于网络核心交换场合.核心交换机的特点是链路速率很高且端口数少.因此,DPS 方案非常适用于核心交换机场合.即使对于端口数较多的边缘交换机,DPS 在实现时也可以采取一些技术来减小交换阵列的规模.比如,可以在缓冲器和交换阵列之间采用合路和分路器,这样,$N\times N$ 的 DPS-d 交换机的交换阵列将是 $N\times N$ 的,而不是 $dN\times dN$ 的.或者可以采用文献[14]提到的多片交换阵列方案,即用多片小的交换阵列互联成一个大的交换阵列.

因此,综合来看,DPS 的扩展性比加速的 CIOQ 方案要好.

5 结束语

本文的主要贡献在于:(1) 提出了一种无须加速且与 OQ 交换机行为等价的交换机设计方案——DPS-d;(2) 证明了 d=2 是 DPS 模仿 OQ 交换机行为的充分条件,从实际应用角度看也是必要条件.

DPS 方案最终实现的困难与加速的 CIOQ 相同,即调度算法的复杂度较高且不易并行实现.因此,进一步的研究可以有两个方向:(1) 设计并行的稳定匹配算法,这需要设法解开输出端所需信息的耦合;(2) 根据 DPS 结构的特点,设计新的能保证分组延迟的调度算法(不一定完全模仿 OQ 的行为),使其具有较低的复杂度.

References:

[1] Karol M, Hluchyj M, Morgan S. Input versus output queuing on a space division switch. IEEE Transactions on Communications, 1987,35(12):1347~1356.

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

[3] Mekkittikul A, Mckeown N. A practical algorithm to achieve 100% throughput in input-queued switches. In: Guerin R, ed. Proceedings of the IEEE INFOCOM'98. San Francisco: IEEE Computer Society Press, 1998. 792~799.

[4] Prabhakar B, Mckeown N. On the speedup required for combined input- and output-queued switching. In: Proceedings of the 1998 IEEE International Symposium on Information Theory. Cambridge: IEEE Information Theory Society Press, 1998. 165.

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

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

[7] Stoica I, Zhang H. Exact emulation of an output queuing switch by a combined input output queuing switch. In: Guerin R, ed. Proceedings of the 6th International Workshop on Quality of Service. Napa: IEEE Communication Society Press, 1998. 218~224.

[8] Chuang ST, Goel A, McKeown N, Prabhakar B. Matching output queuing with a combined input/output queued switch. IEEE Journal on Selected Areas in Commutations, 1999,17(6):1030~1039.

[9] Kleinrock L. Nomadic computing and smart spaces. IEEE Internet Computing, 2000,4(1):52~53.

[10] Zhang H. Service disciplines for guaranteed performance service in packet-switching networks. Proceedings of the IEEE, 1995, 83(10):1374~1396.

[11] Parekh AK, Gallager RG. A generalized processor sharing approach to flow control in integrated services networks the multiple node case. IEEE/ACM Transactions on Networking, 1994,2(2):137~150.

[12] Zhang L. Virtual clock: A new traffic control algorithm for packet switching networks. ACM SIGCOMM Computer Communication Review, 1990,20(4):19~29.

[13] Golestani S. Congestion-Free transmission of real-time traffic in packet networks. In: Silvester J, ed. Proceedings of the IEEE INFOCOM'90. San Francisco: IEEE Computer Society Press, 1990. 527~542.

[14] Han MS, Jeon YI, Lee WS, Park KC. Simple iterative matching for input and output buffered switch with multiple switching planes. In: Lee MM, ed. Proceedings of IEEE the 4th International Conference on ATM and High Speed Intelligent Internet Symposium. Seoul: IEEE Communication Society Press, 2001. 163~167.

敬 告 作 者

《软件学报》创刊以来,蒙国内外学术界厚爱,收到许多高质量的稿件,其中不少在发表后读者反映良好,认为本刊保持了较高的学术水平.但也有一些稿件因不符合本刊的要求而未能通过审稿.为了帮助广大作者尽快地把他们的优秀研究成果发表在我刊上,特此列举一些审稿过程中经常遇到的问题,请作者投稿时尽量予以避免,以利大作的发表.

1. 读书偶有所得,即匆忙成文,未曾注意该领域或该研究课题国内外近年来的发展情况,不引用和不比较最近文献中的同类结果,有的甚至完全不列参考文献.

2. 做了一个软件系统,详尽描述该系统的各个方面,如像工作报告,但采用的基本上是成熟技术,未与国内外同类系统比较,没有指出该系统在技术上哪几点比别人先进,为什么先进.一般来说,技术上没有创新的软件系统是没有发表价值的.

3. 提出一个新的算法,认为该算法优越,但既未从数学上证明比现有的其他算法好(例如降低复杂性),也没有用实验数据来进行对比,难以令人信服.

4. 提出一个大型软件系统的总体设想,但很粗糙,而且还没有(哪怕是部分的)实现,很难证明该设想是现实的、可行的、先进的.

5. 介绍一个现有的软件开发方法,或一个现有软件产品的结构(非作者本人开发,往往是引进的,或公司产品),甚至某一软件的使用方法.本刊不登载高级科普文章,不支持在论文中引进广告色彩.

6. 提出对软件开发或软件产业的某种观点,泛泛而论,技术含量少.本刊目前暂不开办软件论坛,只发表学术文章,但也欢迎材料丰富,反映现代软件理论或技术发展,并含有作者精辟见解的某一领域的综述文章.

7. 介绍作者做的把软件技术应用于某个领域的工作,但其中软件技术含量太少,甚至微不足道,大部分内容是其他专业领域的技术细节,这类文章宜改投其他专业刊物.

8. 其主要内容已经在其他正式学术刊物上或在正式出版物中发表过的文章,一稿多投的文章,经退稿后未作本质修改换名重投的文章.

本刊热情欢迎国内外科技界对《软件学报》踊跃投稿.为了和大家一起办好本刊,特提出以上各点敬告作者.并且欢迎广大作者和读者对本刊的各个方面,尤其是对论文的质量多多提出批评建议.