时隙间迭代的输入队列交换机 Round-Robin 调度算法*

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 Round-Robin Scheduling Algorithm by Iterating Between Slots for Input-Queued Switches

WU Jun+, CHEN Qing, LUO Jun-Zhou

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

+ Corresponding author: Phn: +86-25-83795595, Fax: +86-25-83794838, E-mail: j_wu@seu.edu.cn, http://cse.seu.edu.cn

Received 2003-09-10; Accepted 2004-07-06

Wu J, Chen Q, Luo JZ. A Round-Robin scheduling algorithm by iterating between slots for input-queued switches. *Journal of Software*, 2005,16(3):375–383. DOI: 10.1360/jos160375

Abstract: Input-Queueing is becoming increasingly used for high-bandwidth switches and routers for its scalability, but it needs an elaborate scheduling algorithm to achieve good performance. Round-Robin algorithms have been extensively investigated due to its simplicity and parallelism. However, the present Round-Robin algorithms suffer from poor performance under nonuniform and burst traffic. This paper proposes a Round-Robin algorithm named iSLOT, which can approximate the maximum matching algorithms by iterating the scheduling decision between slots and using the randomness of the queue length. Simulation results show that iSLOT not only is stable under uniform i.i.d traffics, but also outperforms the existing round-robin algorithms under burst and nonuniform traffics in throughput and delay performance.

Key words: switch; input-queueing; Round-Robin; throughput

摘 要: 输入队列因具有良好的可扩展性而广泛应用于高速交换机和路由器中,但输入队列需要精心设计调度算法以获取较好的性能.Round-Robin 算法因其简单性和并行性而得到广泛的研究,但现有的 Round-Robin 算法在突发流量和非均匀流量下的负荷-延迟性能较差.提出了调度决策在时隙间进行迭代的思想,并利用队列长度具有随机性的特点设计了能近似最大匹配的 Round-Robin 算法——iSLOT.仿真结果表明,iSLOT 不仅在均匀流量下是稳定的,在非均匀流量和突发流量下的吞吐率及延迟性能均远好于现有的 Round-Robin 算法.

* 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)); the China Specialized Research Fund for the Doctoral Program of Higher Education under Grant No.20030286014 (高等学校博士学科点专项科研基金); the Jiangsu Provincial Key Laboratory of Network and Information Security under Grant No.BM2003201 (江苏省网络与信息安全重点实验室资助); the Foundation of Excellent Doctoral Dissertation of Southeast University under Grant No.YBJJ0408 (东南大学优秀博士论文基金)

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,Petri 网;陈晴(1973－),女,博士生,主要研究领域为高性能网络协议和算法;罗军舟(1960－),男,博士,教授,博士生导师,主要研究领域为高性能网络与协议工程,网络安全,网络管理,网格计算.

关键词: 交换机;输入队列;Round-Robin;吞吐率
中图法分类号: TP301 文献标识码: A

随着 Internet 用户的增长和多媒体信息的广泛使用,Internet 上的数据流量急剧增长,这对 Internet 主干的速度提出了很高的要求.近几年,随着光通信技术的发展,尤其是随着 DWDM 技术的成熟,Internet 主干链路的容量已不再是问题.然而,构成 Internet 基础寻径/交换结构的交换机(或路由器),由于主要仍采用存储-转发工作方式,而其中存储器工作速率的发展远远跟不上链路容量的发展,因此导致交换机的交换速率成为限制 Internet 提速的瓶颈.

传统的交换机大多采用输出队列结构,即分组缓存在交换机的输出端.虽然输出队列结构的交换机具有最优的性能,但由于缓存器的工作速率必须 N 倍于输入链路的速度,不适于高速交换的场合.另一可选的方案是采用输入队列结构.输入队列分组缓存在输入端,存储器可以工作在链路速率,可扩展性大大优于输出队列结构.但由于分组缓存在输入端所造成的 HOL(head of line)阻塞现象导致输入队列交换机采用 FIFO 调度时的吞吐率只能达到 58.6%[1],因此,输入队列交换机在 20 世纪 90 年代以前未得到广泛研究.近几年,随着链路速率和存储器速率矛盾的日益突出,输入队列交换机引起了研究人员的普遍重视.文献[2,3]的结果显示,采用虚输出队列(virtual output queueing,简称 VOQ)和二分图加权最大匹配调度算法,输入队列交换机可以在任何 i.i.d 流量下达到 100%的吞吐率.最近,文献[3]给出了一类较简单且在任何满足强大数律流量下稳定的最大匹配算法 MNCM.但加权最大匹配算法的复杂度一般为 $O(N^3\log N)$,即使 MNCM 类最大匹配算法也有 $O(N^{2.5})$的复杂度,且难以并行执行,不具备实用性.

目前,输入队列交换机的一个研究热点是设计低复杂度的并行算法,这方面现已取得许多成果.这些算法大致可分为两类,一类是近似最大匹配算法,另一类是极大匹配算法.Tassiulas 在文献[5]中提出了一种近似最大匹配的随机算法,但由于该算法每次调度决策只使用了所有决策空间的两个样点,所以虽然该算法在 i.i.d 流量下是稳定的,但算法的延迟吞吐性能很差.APSARA[6,7]和 SERERA[6,8]可以看做是该算法的改进.APSARA 采用了 $O(N^2)$个样点,取得较好的性能.由于所有 $O(N^2)$个样点的权可以并行计算,因此算法的时间复杂度只有 $O(N)$.但需要 $O(N^2)$个权计算模块和一个权仲裁器,硬件的复杂度高.SERERA 利用了输入模式的随机性,只采用两个样点同样取得了较好的性能,但其复杂度稍高 $O(N\log N)$且不能并行执行.随机算法的另一缺点是算法存在饥饿性.PIM,iLQF,iOCF 和各种 RR(Round-Robin)[9−16]等并行迭代算法属于极大匹配算法.其中,由于 RR 仲裁器只需简单的循环优先操作,硬件实现简单,因此得到了深入的研究.

基本的 RR 算法由于存在指针同步现象,其最大吞吐率小于 65%[9].iSLIP[9]算法从解决指针同步入手,对基本 RR 算法做了改进,使得算法在均匀的 i.i.d 流量下达到了 100%的吞吐率.文献[10]对指针同步问题做了进一步的探讨,提出了去同步更为彻底的 FIRM 算法,因此 FIRM 算法的延迟-吞吐性能要优于 iSLIP.DRR(dual Round-Robin)[11−13]是一种两相的 Round-Robin 算法,与 iSLIP 和 FIRM 等三相的 RR 算法相比,输入输出端口间的通信量大大减少,同时,文献[13]证明了 DRR 算法在均匀 i.i.d 流量下的吞吐率为 100%.这类极大匹配算法虽然硬件实现简单,与近似最大匹配算法相比更适合高速的场合,但 RR 算法有着致命的缺陷:在非均匀流量和突发流量下算法延迟-吞吐性能差,尤其是在非均匀流量下,吞吐率只能达到 80%左右.为了克服这一缺陷,文献[15]中提出了 EDRR(exhaustive dual Round-Robin)算法,该算法在一种非均匀流量下(本文称为弱对角流量)的吞吐率达到了 95%以上,但 EDRR 在均匀流量下的性能很差.同时,我们通过仿真发现,EDRR 算法在一种更扭曲的流量下(本文称为对角流量)其吞吐率同样只能达到 85%左右.

本文针对现有 RR 算法的问题,提出了调度决策在时隙间进行迭代的思想,设计了 iSLOT 算法.仿真结果表明,iSLOT 算法不仅在均匀 i.i.d 流量下是稳定的,在非均匀流量下的吞吐率达到 95%以上,并且在突发流量下,iSLOT 算法的性能要远好于其他 RR 算法.甚至在均匀 i.i.d 流量下,iSLOT 算法的延迟-吞吐性能也好于 iSLIP,FIRM 等经典的 RR 算法.

1 Round-Robin 算法原理

文献[14]首次提出 VOQ 的队列组织方式,将输入队列交换机的调度问题转化成二分图的匹配问题,并设计了并行迭代算法 PIM,使得输入队列交换机的吞吐率在均匀 i.i.d 流量下提高到了 95%以上.但 PIM 算法的实现需要随机数产生器,难以硬件实现,所以利用循环优先级仲裁来替代随机仲裁的 RR 算法得到广泛而深入的研究.目前,用于输入队列交换机的 RR 算法可分为两类,一类是三相的 RR 算法,其每一迭代步执行下述 3 个步骤:

Step1(Request). 每一输入端口为该端口的非空 VOQ 发送服务请求至输出端;

Step2(Grant). 每一输出端口在收到的请求服务的输入端口中,以循环优先级方式选择最接近输出端指针的输入端口,发送允许服务信息至该输入端口,并根据第 3 步的结果修改指针;

Step3(Accept). 每一输入端口在收到允许服务的输出端口中,以循环优先级方式选择最接近输入端指针的输出端口,发送接受服务信息至该输出端口.修改该输入端口的指针,并将找到的输入-输出匹配用于配置交换阵列.

基本的 RR 算法、iSLIP 和 FIRM 等算法属于三相的 RR 算法.这些算法的区别仅在于 Step2 中各输出端指针的修改策略.iSLIP 和 FIRM 由于去了指针的同步而具有较好的性能(关于指针同步、去同步的讨论请参见文献[9,10,16]),并且文献[13]证明了 iSLIP 算法仅需迭代一步即可在均匀 i.i.d 流量下获得 100%的吞吐率.

DRR 和 EDRR 算法属于两相的 RR 算法,即每个迭代步执行下述两个步骤:

Step1(Request). 每一输入端口在所有非空 VOQ 中,以循环优先级方式选择最接近该输入端指针的 VOQ,为其发送请求服务信息至相应的输出端口,并根据第 2 步的结果修改指针;

Step2(Grant). 每一输出端口在收到的请求服务的输入端口中,以循环优先级方式选择最接近输出端指针的输入端口,发送接受服务信息至该输入端口.修改该输出端口的指针,并将找到的输入-输出匹配用于配置交换阵列.

DRR 和 EDRR 算法的区别在于指针的修改策略不同.当找到一对输入-输出匹配后,DRR 算法将相应的输入端和输出端指针递增 1,而 EDRR 算法将相应的指针停留在与该匹配对应的位置上.因此,DRR 是一种公平服务的策略,而 EDRR 属于竭力服务的策略.公平服务的策略能够很好地服务于均匀流量,因此 DRR 算法在均匀的 i.i.d 流量下是稳定的[13].

两相 RR 算法与三相 RR 算法相比,由于输入输出间少了一次信息交互,端口间的通信量大为减少,可以支持更高的链路速率.三相算法可以通过多次迭代进一步改善性能,但由于链路速率越来越快,导致留给调度决策的时间也越来越短.例如,1Tbps 的链路速率,若分组长度为 1000bits,那么留给调度的时间仅有 1ns,显然多步迭代在这种高速场合下是不可行的.因此,下文提及的 RR 算法均指迭代一次的 RR 算法.

2 ISLOT(inter-slot iterative)算法

RR 算法可以通过多次迭代改善性能,说明一次迭代不能保证 100%地找到极大匹配.另一方面,由于每个时隙抵达每个输入端的分组至多一个,而且每个输入端也至多只有一个分组接受服务,这说明前后两个时隙的 VOQ 状态变化不大,也就是说,前一个时隙的匹配边在下一个时隙几乎也是匹配的.因此,若下一个时隙的决策在前一个决策的基础上进行迭代,将提高找到极大匹配的概率,从而提升算法的性能.基于这一观察,我们首先设计了基本的时隙间迭代调度算法.

2.1 基本的时隙间迭代算法

算法 1 是一个两相的 RR 算法,算法包括输入端的请求和输出端的应答两个阶段.显然,其复杂性与其他两相 RR 算法相同.它也采用了竭力服务的策略,与 EDRR 的区别在于,EDRR 是通过 RR 指针的修改策略隐式地保留了前一个时隙的可用匹配,而算法 1 对前一个时隙的可用匹配采用了显式的保留(保留在数组 *Sch* 中).采用显式保留的优点是,下一个时隙的决策可以在前一个决策的基础上作进一步迭代获得.时隙间迭代的思想也可用于三相的 RR 算法,限于篇幅,这里省略.

算法 1. 基本的时隙间迭代算法.

$Q[0..N-1][1..N]\ldots Q[i][j]$表示输入端口 i 去往输出端 j 的分组队列长度;

$p\_Req[0..N-1]$	输入端指针 $p_Req[i]$取值于{1,2,…,N};初始时可取任意值;
$p\_Grt[1..N]$	输出端指针 $p_Grt[i]$取值于{0,1,…,N–1};初始时可取任意值;
$r\_State[1..N][0..N-1]$	收到的请求状态寄存,$r\_State[i][j]$=1 表示输出端口 i 收到来自输入端 j 的服务请求;
$Sch[0..N-1]$	匹配寄存,$Sch[i]=j$ 表示输入端 i 和输出端 j 的一对匹配,j 取值于{1,2,…,N};
$out\_Busy[1..N]$	输出端口在一个时隙开始时状态寄存,$out\_Busy[i]$=1 表示在时隙开始时输出端 i 已有匹配;

```
Input_port i:
If (Sch[i]==0)
   For (j=1;j<=N;j++)
   If ((Q[i][p_Req[i]])&&(out_Busy[p_Req[i]]==0))
                r_State[p_Req [i]][i]=1;
                break;
      }
      else p_Req[i]=p_Req[i]%N+1;
```

```
Output_port i:
for (j=0;j<N;j++)
      if (r_State[i][p_Grt[i]]){
            {Sch[p_Grt[i]]=i;
            out_Busy[i]=1;
            break;
      }
      else p_Grt[i]=(p_Grt[i]+1)%N;
```

Sch 中记录的匹配用于交换机的调度,然后输入端修改状态值为下一时隙的决策做准备.

```
Input_port i:
If (Q[i][Sch[i]]>0){
      Q[i][Sch[i]]=Q[i][Sch[i]]−1;
      If (Q[i][Sch[i]]==0){
            p_Req[i]=p_Req[i]%N+1;
            out_Busy[Sch[i]]=0;
            Sch[i]=0;
      }
}
```

2.2 算法1的性能分析

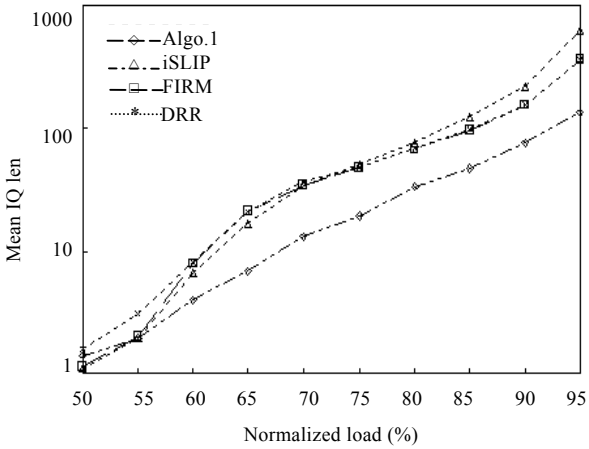


Fig.1 The performance of Algorithm 1 under uniform i.i.d traffic

图 1 算法 1 在均匀 i.i.d 流量下的性能

从前面的讨论可知,算法 1 在每个时隙找到极大匹配的概率应该大于 EDRR 等其他 RR 算法,因此应该具有较好的延迟-吞吐性能.我们利用仿真实验验证了这一分析.具体仿真环境请见第 4 节.图 1 给出了 iSLIP,FIRM, DRR 和算法 1 在均匀 i.i.d 流量下的队长-负荷曲线.从中可以看出,算法 1 的队长明显地小于其他几种算法.图 2 给出了算法 1 在一种非均匀流量(对角流模式)下的队长-负荷曲线.虽然,算法 1 在对角流量下的性能并不理想,但其在各个负荷时的队长还是要小于其他的 RR 算法.因此,仿真的结果支持我们关于时隙间迭代有助于找到极大匹配的分析.

算法 1 与其他 RR 算法相比,性能有了较大的改进,虽然它仍未克服 RR 算法在非均匀流量下性能差的弱点,但它说明时隙间迭代可以有效地提

升 RR 算法的性能.

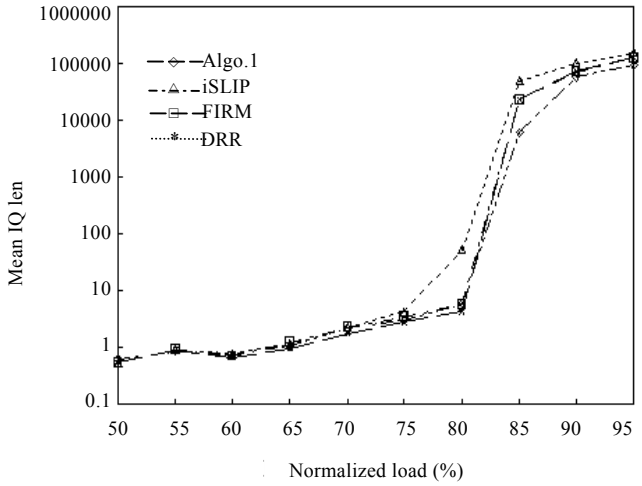


Fig.2 The performance of Algorithm 1 under diagonal traffic
图 2 算法 1 在对角流量下的性能

2.3 iSLOT算法

目前的研究结果表明,包括 PIM,iSLIP,FIRM,DRR 在内的所有极大匹配算法都在非均匀流模式下表现欠佳.而近似最大匹配的算法却可以在各种 i.i.d 流量下取得较好的性能,因此要使得 RR 算法能够适应非均匀流量就必须设计出在某种程度上能够近似最大匹配的 RR 算法.基于这一考虑,我们在算法 1 的基础上设计了能以较大概率找到最大匹配的 RR 算法——iSLOT.具体算法如下:

iSLOT 算法.

out_Busy[*i*]表示与输出端口 *i* 相应的匹配将服务的时隙数.其他变量定义与算法 1 相同.

```
Input_port i:
If (Sch[i]==0)
    For (j=1;j<=N;j++)
    If ((Q[i][p_Req[i]])&&(out_Busy[p_Req[i]]==0)){
            r_State[p_Req[i]][i]=1;
            break;
      }
            else p_Req[i]=p_Req[i]%N+1;
```

```
Output_port i:
for (j=0;j<N;j++)
      if (r_State[i][p_Grt[i]]){
            Sch[p_Grt[i]]=i;
            out_Busy[i]=Q[p_Grt[i]][i]/2;
            break;
      }
      else p_Grt[i]=(p_Grt[i]+1)%N;
```

Sch 中记录的匹配用于交换机的调度,然后输入端修改状态值为下一时隙的决策做准备.

```
Input_port i:
If (Q[i][Sch[i]]>0){
      Q[i][Sch[i]]=Q[i][Sch[i]]−1;
      Out_Busy[Sch[i]]=Out_Busy[Sch[i]]−1;
      if(Out_Busy[Sch[i]]==0){
            p_Req[i]=p_Req[i]%N+1;
            Sch[i]=out_Busy[Sch[i]]=0;
      }
}
```

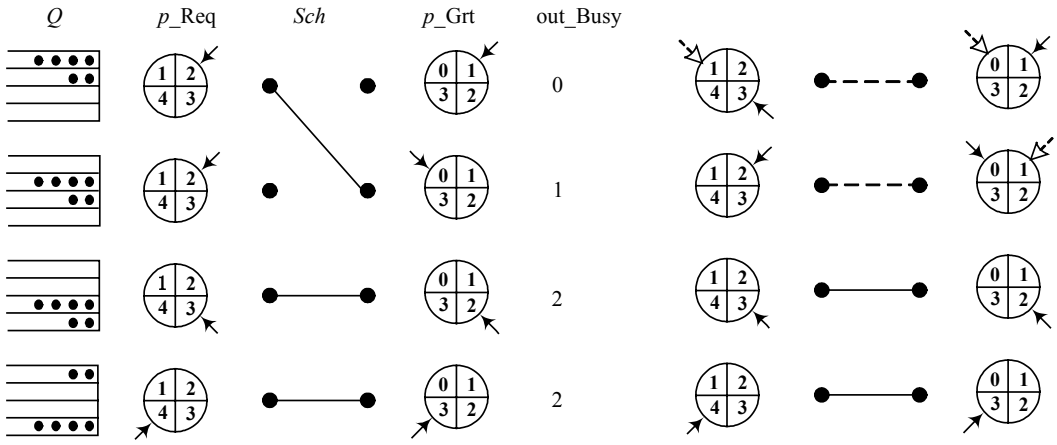
iSLOT 算法与算法 1 的主要区别在于,它采用了介于完全公平服务和竭力服务之间的一种服务方式,即数组 *Sch* 中所保留的前一时隙的匹配边与算法 1 不同.当一个输出端口找到一个 VOQ 作为匹配时,该输出端口将

为该 VOQ 服务 1/2 队长个时隙,然后算法将强行释放这一匹配边.这一服务策略不仅改善了算法 1 的公平性,而且与时隙间迭代相结合,使得 iSLOT 算法具有了近似最大匹配的能力.下面我们对 iSLOT 算法的机理作进一步的分析.

2.4 iSLOT算法的性能分析

算法 1 归根结底还是极大匹配算法.它之所以在均匀 i.i.d 流量下取得很好的性能,是由于在均匀流量下负荷较高时,每个输入端口的各个 VOQ 几乎都不空,这样,输入输出端口所构成的二分图几乎是完全的,在完全的二分图上极大匹配也就是最大匹配.但在非均匀流量下,尤其是对角流这种非常扭曲的流量下,每个输入端只有少数的 VOQ 非空.在这种情况下,极大匹配常常不是最大匹配,这就是算法 1 和其他极大匹配算法不能很好地调度非均匀流量的原因.但算法 1 又不同于其他极大匹配算法,它是在前一个调度决策的基础上作进一步迭代以获得当前时隙调度的,因此,只要能从前一个时隙的极大匹配中跳出来,即释放掉导致不能收敛到最大匹配的某些匹配边,就有可能使得在下一个时隙的迭代中找到最大匹配.当然,判断哪些边是导致不能收敛到最大匹配的边是较为困难的,RR 算法之所以受到重视是因为它的简单,故而复杂的算法是不可取的.所以,我们在 iSLOT 算法的设计中采用了简单的随机释放策略,即利用了各队列队长的随机性.每当一个输出口找到一个 VOQ 作为匹配时,该输出口将为该 VOQ 服务 1/2 队长个时隙,由于队长是随机的,算法运行一段时间后一个调度决策中匹配边的释放也就具有一定的随机性,同时,利用队长作为匹配边释放的依据使得算法具有一定的自适应特性.

图 3 的例子可以进一步说明 iSLOT 的工作过程.图 3 中的 A 给出了算法在时隙 t–1 时找到匹配后的状态.此时的匹配有 3 条匹配边,是一个极大匹配但不是最大匹配.用这个匹配配置交换阵列后,算法将对系统状态进行修改.由于 out_Busy[2]=1,在 t–1 时隙服务完之后,out_Busy[2]将变为 0,因此,根据 iSLOT 算法,即使输入端 0 还有一个去往输出端 2 的分组,该匹配边(输入端口 0-输出端口 2)均将被释放,同时,p_Req[0]将移动到 3.时隙 t 时,为说明简单起见,我们假设时隙 t 开始时没有新的分组到达.由于输入端口 0 没有去往输出端 3 和 4 的分组,所以 p_Req[0]将滑动到 1,从而向输出端口 1 发出服务请求.同样地,输入端 1 将向输出端 2 发出请求,而输入端 2 和 3 由于有上一个时隙保留的匹配,故不发出请求信息.所以,输出端口 1 和 2 将各收到一个服务请求,并将响应这一请求.因此,时隙 t 的迭代将找到两个新的匹配边,如图 3 中的 B 的虚线边所示,再加上上一个时隙保留下的两个匹配边构成了时隙 t 的调度决策.显然这一匹配是最大匹配.而若采用算法 1,由于时隙 t–1 时已找到了极大匹配,且服务完后没有被服务的队列变空,所以这个极大匹配将被保留至时隙 t.而在极大匹配的基础上,无论再进行怎样的迭代也不会对结果有任何改进,因此,算法 1 在时隙 t 时仍然只能采用极大匹配.



A: The found matching and variables state of time slot t–1

B: The found matching and variables state of time slot t

Fig.3 Illustration of an execution of iSLOT algorithm

图 3 iSLOT 算法工作过程示例

通过上述分析可知,iSLOT 算法已不单纯是极大匹配算法,在很大程度上已能近似最大匹配算法,因此我们有理由相信,iSLOT 将能够很好地调度非均匀流量.同时,由于 iSLOT 算法采用了两相 RR 算法的框架,每个端口每个时隙只需作一次循环优先仲裁,且各端口的运算是并行执行的,故而复杂度是 $O(1)$.另一方面,循环优先仲裁器只需简单的逻辑即可实现[9],现已广泛应用于高速交换机中,因此 iSLOT 算法具有很好的实用性.

3 仿真结果

为了验证 iSLOT 算法的性能,我们对其进行了详尽的仿真.在给出仿真结果之前,我们先介绍本文仿真环境的设置.

3.1 交换机及流量模型

(1) 交换机模型

我们所仿真的交换机是 32×32 且所有链路具有相同的速率.交换机采用 VOQ 存储组织方式,每个 VOQ 的容量足够大即不发生缓冲溢出.不同端口间不进行缓冲共享.

(2) 流量模型

负荷ρ采用正规化的负荷,即$\rho\in[0,1]$.本文的仿真实验使用了文献中常见的 4 种流量模型:

均匀流量:分组的到达过程是 Bernoulli i.i.d 的. $\lambda_{ij}=\rho/N$, $\forall i,j$, λ_{ij} 表示抵达输入端口 i 去往输出端口 j 的分组速率.

对角流量:分组到达过程是 Bernoulli i.i.d 的.对任一输入端口 i, $\lambda_{ii}=2\rho/3$, $\lambda_{ii+1}=\rho/3$,对其他输出端口 j, λ_{ij}=0.这是一种非常扭曲的流量,每个输入端除了两个 VOQ 有分组到达外,其他队列皆空闲.

弱对角流量:分组到达过程是 Bernoulli i.i.d 的.对任一输入端口 i, $\lambda_{ii}=2\rho/3$,对于其他输出端口 j, $\lambda_{ij}=\rho/(3\times(N-1))$,这也是一种非均匀流量,但与对角流量相比要稍均匀些.在有些文献中,也将这种流量称为对角流量,本文中为了与上述的对角流量区别,我们称其为弱对角流量.

突发流量:分组的到达过程不再是 i.i.d 的,而是符合一种两状态的马尔可夫过程,即 ON-OFF 模型.当一个输入端口处于 ON 状态时有分组到达,当处于 OFF 状态时将没有分组的抵达.ON 的长度服从均值 32 的几何分布,OFF 的长度服从均值 $(1-p)/p$ 的几何分布,其中参数 p 用来调节流量的负荷.目前的研究表明,Internet 上的流量具有一定的自相似性,因此突发流量模型更接近实际流量.

我们主要考察了吞吐率和延迟两方面的性能.在估计算法的吞吐率时,算法对考察的负荷运行 8 000 000 时隙,若 4 000 000 时隙后,系统的队长不显著增长即认为算法在这一负荷是稳定的.算法的延迟性能本文以队长-负荷曲线表示,分组的平均延迟可以由平均队长通过 Little 公式进行估算.在获得算法在某一负荷下的队长时,算法在该负荷下运行 1 000 000 时隙.

3.2 iSLOT算法的吞吐率

表 1 给出了各种 RR 算法在不同流量下的最大吞吐率.iSLIP,FIRM 和 DRR 这 3 种 RR 算法在均匀的 i.i.d 流量下都是稳定的,即达到了 100%的吞吐率.但它们在两种非均匀流量下吞吐率较小,都未超过 85%.EDRR 是针对 DRR 在非均匀流量下性能差这一弱点所做的改进,从表中可以看出,EDRR 在非均匀流量下的吞吐率的确好于 DRR,尤其是在弱对角流量下,EDRR 的吞吐率高达 96.8%.但 EDRR 在均匀流量下的吞吐率也只有 96.5%,且过渡态长.而 iSLOT 算法不仅在均匀 i.i.d 流量下的吞吐率达到 100%,在两种非均匀流量下的吞吐率都达到 96%以上,因此就吞吐率性能而言,iSLOT 要优于其他 RR 算法.

Table 1 The throughput of kinds of RR algorithms under different traffics

表 1 各种 RR 算法在不同流量下的吞吐率

	Uniform traffic	Diagonal traffic	Weakly diagonal traffic
iSLIP	1.0	0.812	0.776
FIRM	1.0	0.838	0.774
DRR	1.0	0.836	0.757
EDRR	0.965	0.855	0.968
iSLOT	1.0	0.962	0.975

3.3 延迟-吞吐率性能

延迟性能是输入队列调度算法的一个重要性能指标,它不仅关系到交换机所需配置的存储器大小,而且直接决定网络的服务质量,因此我们对 iSLOT 算法在各种流量下的延迟性能作了详尽的仿真.如图 4~图 7 所示.

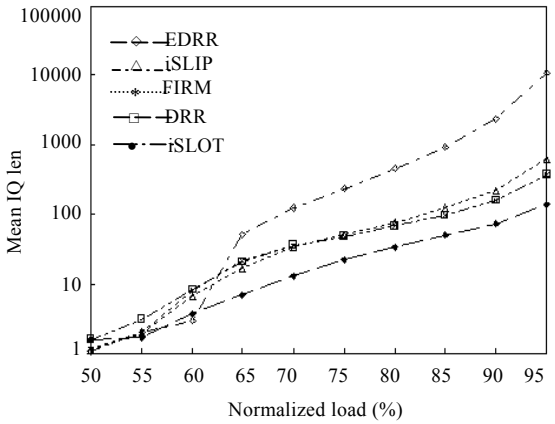


Fig.4 IQ length-load curve under uniform i.i.d traffic
图 4 均匀 i.i.d 流量下的队长-负荷曲线

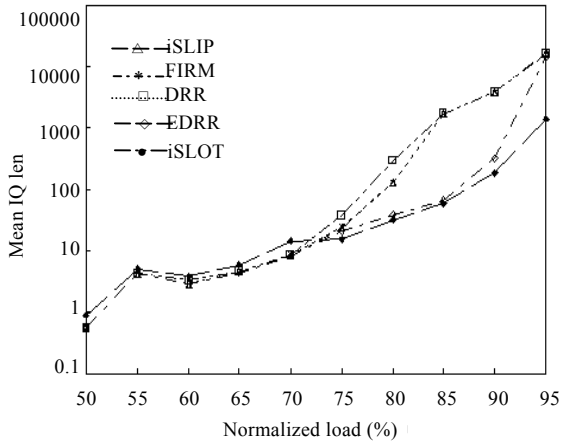


Fig.5 IQ length-load curve under burst traffic
图 5 突发流量下的队长-负荷曲线

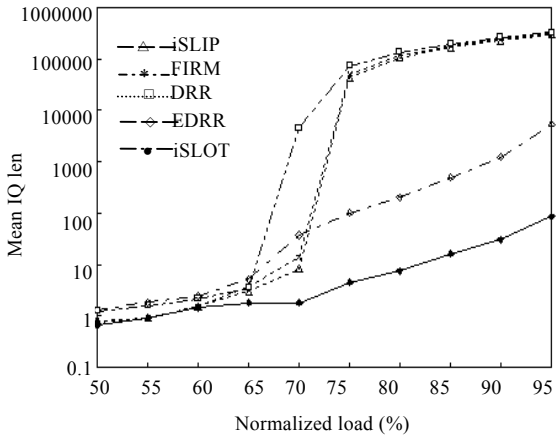


Fig.6 IQ length-load curve under weakly diagonal traffic
图 6 弱对角流量下的队长-负荷曲线

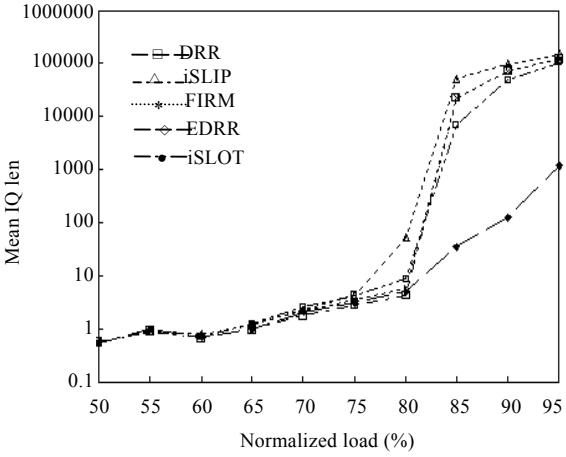


Fig.7 IQ length-load curve under diagonal traffic
图 7 对角流量下的队长-负荷曲线

图 4 是各种 RR 算法在均匀 i.i.d 流量下的队长-负荷曲线.从中可以看出,EDRR 算法虽然在非均匀流量下比 DRR 的性能有所提高,但是以付出均匀流量下的延迟性能为代价的.不仅吞吐率下降了 3%左右,而且高负荷时延迟比 DRR 高出了近 30 倍.而 iSLOT 算法在均匀流量下的延迟性能却明显好于 iSLIP,FIRM 等经典算法,在负荷大于 80%时,iSLOT 算法的队长大约是 FIRM 和 DRR 的 1/2,只有 iSLIP 队长的 1/3.

图 5 比较了各种 RR 算法在突发流量下的延迟性能.突发流量由于更接近网络上的真实流量,因此算法在突发流量模型下的性能显得尤为重要.图 5 的结果显示,iSLIP,FIRM 和 DRR 在突发流量下延迟性能差,与均匀 i.i.d 流量相比,队长急剧增长.这主要是由于这 3 种算法采用了完全的公平服务策略而这一策略,与分组的突发到达不相匹配造成的.EDRR 和 iSLOT 对突发流量却显得较不敏感,尤其是 iSLOT 在突发流量下仍然保持了很好的性能.在 90%的负荷下,iSLOT 的队长是 EDRR 队长的 60%;在 95%的负荷时,iSLOT 队长不到 EDRR 队长的 1/10.

图 6 和图 7 给出了算法在两种非均匀流量下的性能.iSLIP,FIRM 和 DRR 在非均匀流量下表现出极差的性

能.而 EDRR 算法在弱对角流量下性能有了很大的改善,但在对角流量下几乎仍然和 iSLIP 等算法在一个水平上.而 iSLOT 算法与它们相比却有了本质的提高,在 95%的负荷时,队长只在 10^2 数量级,这是极大匹配算法所不能达到的.这一结果验证了我们前面的分析——iSLOT 算法已不仅是极大匹配算法,它能够在很大程度上近似最大匹配算法.

4 结束语

本文提出的 iSLOT 是一种两相的 Round-Robin 算法,它的主要特点是:(1) 提出了调度决策在时隙间进行迭代的思想,令下一个时隙的决策在上一个时隙决策的基础上进行迭代,提高了一步迭代找到极大匹配的可能性.(2) 通过随机释放匹配边来及时跳出极大匹配.这两点相结合使得 iSLOT 算法具备了近似最大匹配的能力.仿真实验表明,iSLOT 与其他 RR 算法相比,不仅在突发流量下和非均匀流量下的性能有了质的突破,而且在均匀流量下吞吐率达到 100%的同时,算法的延迟性能也有明显改善.

我们下一步的工作将研究 iSLOT 算法思想在区分服务网模型下的应用,并适当改动 iSLOT 算法以使其能够提供区分服务.

References:

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

[2] Mckeown N, Mekkittikul A, Anantharam V, Walrand J. Achieving 100% throughput in an input-queued switch. IEEE Trans. on Communications, 1999,47(8):1260−1267.

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

[4] Tabatabaee V, Tassiulas L. MNCM a new class of efficient scheduling algorithms for input buffered switches with no speedup. In: Matta I, ed. Proc. of the IEEE INFOCOM. San Francisco: IEEE Communications Society, 2003. 1406−1413.

[5] Tassiulas L. Linear complexity algorithms for maximum throughput in radio networks and input queued switches. In: Guerin R, ed. Proc. of the IEEE INFOCOM. San Francisco: IEEE Computer Society Press, 1998. 533−539.

[6] Giaccone P, Prabhakar B, Shah D. Towards simple, high-performance schedulers for high-aggregate bandwidth switches. In: IEEE Computer Society, ed. Proc. of the INFOCOM 2002. San Francisco: IEEE Press, 2002. 1160−1169.

[7] Giaccone P, Shah D, Prabhakar B. An implementable parallel scheduler for input-queued switches. IEEE Micro, 2002,22(1):19−25.

[8] Shah D, Giaccone P, Prabhakar B. Efficient randomized algorithms for input-queued switch scheduling. IEEE Micro, 2002,22(1): 10−18.

[9] McKeown N. The iSLIP scheduling algorithm for input-queued switches. IEEE Trans. on Networking, 1999,7(2):188−201.

[10] Serpanos DN, Antoniadis PI. FIRM: A class of distributed scheduling algorithms for high-speed ATM switches with multiple input queues. In: Katzela I, ed. Proc. of the IEEE INFOCOM. Tel Aviv: IEEE Communications Society, 2000. 548−555.

[11] Chao HJ, Park JS. Centralized contention resolution schemes for a large-capacity optical ATM switch. In: IEEE Communication Society, ed. Proc. of the IEEE ATM Workshop. Fairfax: IEEE Press, 1998. 11−16.

[12] Chao HJ. Saturn: A terabit packet switch using dual Round-Robin. IEEE Communication Magazine, 2000,38(12):78−84.

[13] Li Y, Panwar S, Chao HJ. On the performance of a dual Round-Robin switch. In: Ammar M, ed. Proc. of the IEEE INFOCOM. Anchorage: IEEE Communications Society, 2001. 1688−1697.

[14] Andersen T, Owicki S, Saxe J, Thacher C. High speed switch scheduling for local area networks. ACM Trans. on Computer Systems, 1993,11(4):319−352.

[15] Li Y, Panwar S, Chao HJ. The dual Round-Robin matching switch with exhaustive service. In: Gunner C, ed. Proc. of the IEEE Workshop on High Performance Switching and Routing. Kobe: IEEE Communications Society, 2002. 58−63.

[16] Jiang Y, Hamdi M. A fully desynchronized Round-Robin matching scheduler for a VOQ packet switch architecture. In: Gunner C, ed. Proc. of the IEEE Workshop on High Performance Switching and Routing. Kobe: IEEE Communications Society, 2002. 58−63.