一种优化指针策略的输入排队调度算法①

申　宁[1]　李　俊[1]　倪　宏[2] (1. 中国科学技术大学 网络传播系统与控制重点实验室 安徽 合肥 230027; 2. 中国科学院 声学研究所 国家网络新媒体工程技术研究中心 北京 100190)

摘　要： 针对FIRM(fcfs in round-robin matching)算法在处理非均匀业务时，延时和丢包性能出现缺陷的问题，在 FIRM 算法的基础上设计并实现了一种优化指针策略的 low-FIRM(longest oldest weighted FIRM)算法。该算法根据队列长度和队首信元等待时间的权值修改输入端口的轮询指针，使得权值大的队列趋于优先服务，从而优化了在非均匀业务下的调度性能。接着给出了 low-FIRM 算法的性能分析和仿真，与 iSLIP(iterative round-robin matching with slip)算法、FIRM 算法进行了比较。仿真结果表明，low-FIRM 与经典算法相比，在均匀业务下的性能近似，而在非均匀业务下性能有了较大的提升。

关键词： 交换结构；虚拟输出队列；FIRM；调度算法；优化指针策略

Pointer Strategy Optimized Scheduling Algorithm for Input Queued Switches

SHEN Ning[1], LI Jun[1], NI Hong[2] (1. Key Lab of Anhui Network Communication System and Control, USTC, Hefei Anhui 230027, China; 2. DSP Center of Institute of Acoustics, Chinese Academy of Sciences, Beijing 100190, China)

Abstract: To resolve the performance decrease of FIRM under non-uniform traffic, a pointer strategy optimized algorithm named low-FIRM is proposed. This scheduling algorithm modifies round-robin pointers of each input queue according to the weight of the length and waited time of each virtual output queue, which makes the queue with largest weight tend to be served first. Low-FIRM is described and its performance is compared with those of iSLIP and FIRM. Simulation results show that low-FIRM achieves high performance under both of the uniform and non-uniform traffic.

Keywords: switching fabric; virtual output queue (VOQ); fcfs in round-robin matching (FIRM); scheduling algorithm; pointer strategy optimized

1　引言

随着 Internet 用户的增多和多媒体应用的广泛普及，网络数据流量急剧增长，因此对网络主干的速度提出了更高的要求。近年来，随着光纤通信等技术的不断发展，主干链路的容量已经不是问题，然而，作为互联网重要设备的交换机仍采用存储-转发的工作方式，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。

影响交换机交换性能的主要因素是交换结构、队列结构以及调度算法。目前设计开发高性能网络路由器内部交换结构的技术已趋于成熟，相比传统的共享内存和共享总线，Crossbar 交换结构在提供比较高的吞吐率的同时，可扩展性也得到保证。虚拟输出队列(virtual output queue，简称 VOQ)能够解决分组缓存在输入端所造成的 HOL(head of line)阻塞现象[1]，

① 基金项目:国家科技支撑计划(2008BAH28B04);安徽省高校自然科研项目(KJ2008A106)
收稿时间:2010-04-12;收到修改稿时间:2010-05-19

同时又保留了输入排队实现简单、可扩展性好的优点。针对输入排队的 Crossbar 结构，研究者提出来一系列低复杂度的极大匹配算法：并行迭代匹配(parallel iterative matching，简称 PIM)算法[2]、基本轮转算法(round-robin matching，简称 RRM)算法[3]、滑动多次迭代轮转匹配算法(iterative round-robin matching with slip，简称 iSLIP)算法[3]、先到先服务轮转匹配算法(fcfs in round-robin matching，简称 FIRM)算法[4]等。PIM 算法改善了最大匹配算法(maximum size matching，简称 MSM)算法存在的“饿死”现象，但 PIM 算法一次迭代的吞吐率只有 63%[5]；RRM 算法克服了 PIM 算法的复杂性和不稳定性，但是存在指针同步的问题，其最大吞吐率小于 65%[3]；iSLIP 算法和 FIRM 算法都是从解决指针同步入手，对基本的 RR 算法做了改进，使得算法在均匀流量下达到了 100%的吞吐率，FIRM 算法在延迟和吞吐性能上略优于 iSLIP 算法，但是这两种算法在非均匀流量下，调度性能不佳[6]。

本文第 2 节介绍了 FIRM 算法，分析了其在处理非均匀业务流上存在不足的原因；第 3 节提出了一种基于 FIRM 的改进算法 low-FIRM(longest oldest weighted FIRM)，阐述了其设计思想和具体实现，对其性能进行了详尽的分析；第 4 节给出了改进算法的性能评估结果，主要是在延时和丢包性能上与经典的 iSLIP 和 FIRM 算法进行了比较；第 5 节总结全文。

2 FIRM算法原理

FIRM 算法是一种极大匹配(maximal matching)算法，其目标是通过多次迭代找到一个极大匹配。在一个时隙开始时，所有的输入和输出端口都初始化为未匹配。每个输入端口存在一个指向最高优先级输出端口的指针 RRI，同样，每个输出端口也存在一个指向最高优先级输入端口的指针 RRO。算法在迭代收敛或者迭代 i 次后停止，进入下一时隙。迭代收敛是指在新的一轮迭代中没有新的匹配出现。

FIRM 算法每次迭代的执行步骤：

第一步：请求。所有未匹配的输入端口向每一个非空 VOQ 对应的、未匹配的输出端口发出请求信号。

第二步：准许。如果未匹配的输出端口 j 接收到请求，则按照 Round Robin 规则找到最接近 RRO_j 的输入端口，并向该输入端口发送准许信号。当且仅当该准许在第一次迭代的第三步被输入端口接受以后，RRO_j 才会被更新为准许端口的下一个输入端口。

第三步：接受。如果未匹配的输入端口 i 接收到准许，按照 Round Robin 规则找到最接近 RRI_i 的输出端口进行接受，建立输出匹配。当且仅当这是第一次迭代时，RRI_i 才会被更新为匹配输出端口的下一个端口，对于发出准许信号而未匹配的输出端口，其 RRO 指向未接受其许可的输入端口[7]。

FIRM 算法的第三步中，每个输入端口公平的对待所有输出端口的准许，并不考虑相应 N 个 VOQ(输入端口和输出端口数均为 N)的长度和队首信元的等待时间。按照这一服务准则，每个 VOQ 都具有相同的输出速率，在均匀业务流的情况下，每个 VOQ 的输入速率相同，因此这种公平服务的调度算法可以达到很好的性能，然而，在非均匀业务流的情况下，不同 VOQ 的输入速率存在很大的差异，如果对每个 VOQ 仍然采用公平服务的原则，速率低的 VOQ 能够得到及时的服务，而速率高的 VOQ 中信元则会不断积累，表现为信元延时和丢包率上升。

3 low-FIRM算法

low-FIRM 算法通过修改输入端口优先级指针 RRI 的更新策略，使得该算法不仅保持 FIRM 算法在处理均匀流量的优势，而且在处理非均匀流量时，调度性能也有较大提升。

3.1 low-FIRM 算法的实现

基于 FIRM 算法，low-FIRM 算法进行了如下的修改：每次调度的第一次迭代中，将输入端口 i 的接受指针 RRIi 指向第 i 个输入端口的 N 个 VOQ 队列中队长最长的输出端口，如果存在不止一个最大长度的 VOQ 队列，则指向等待时间最长的那个 VOQ 对应的输出端口。

假设用 $VOQ_{i,j}$ 表示输入端口为 i，输出端口为 j 的虚拟输出队列，$L_{i,j}$ 表示 $VOQ_{i,j}$ 的队列长度，$O_{i,j}$ 表示 $VOQ_{i,j}$ 的队首信元等待时间，则该算法相当于为每个 VOQ 设定了一个基于 L 和 O 的权值 $W_{i,j}$，RRI_i 应指向 N 个 VOQ 中权值最大的队列。比较权值大小的方法是：若 $L_{i,j}$ 大于 $L_{i,j'}$ 时，则 $W_{i,j}$ 大于 $W_{i,j'}$；当 $L_{i,j}$ 等于 $L_{i,j'}$ 时，若 $O_{i,j}$ 大于 $O_{i,j'}$，则 $W_{i,j}$ 大于 $W_{i,j'}$。

输入端口 i 使用 low-FIRM 算法完成一次迭代的实现伪代码如下：

if (本次迭代是第一次迭代)

RRI_i 指向 $W_{i,j}$ 最大的输出端口;

向所有非空 VOQ 对应的输出端口发送请求信号;

等待直到收到输出端口的信号;

if (信号为准许信号)

选择最接近 RRI_i 的输出端口发送接受信号;

向其他端口发送拒绝信号;

if (本次迭代时第一次迭代)

RRI_i 更新为准许端口的下一个输出端口;

连接建立;

else

收到信号为拒绝信号，判断是否进入下一次迭代;

输出端口 j 使用 low-FIRM 算法完成一次迭代的实现伪代码如下:

等待直到收到输入端口请求信号;

选择最接近 RRO_j 的输入端口发送准许信号;

向其他端口发送拒绝信号;

等待直到收到输入端口信号;

if (信号为接受信号)

if (本次迭代时第一次迭代)

RRO_j 更新为接受端口的下一个输入端口;

连接建立;

else

if (本次迭代时第一次迭代)

收到信号为拒绝信号，RRO_j 更新为未接受

其准许的输入端口;

判断是否进入下一次迭代;

3.2 low-FIRM 算法的性能分析

通过上述指针策略的优化，low-FIRM 算法在每次迭代的接受步骤中，输入端口倾向于接收队列长度最长并且队首信元等待时间最长的输出端口的准许。

low-FIRM 算法在处理均匀流量时，由于每个 VOQ 的队列长度和队首信元等待时间趋于相等，即每个队列的权值趋于相等，接受指针的更新策略与 FIRM 算法相似，因此调度性能同 FIRM 算法近似。

low-FIRM 在处理非均匀流量时，权值 W 最大的 VOQ 倾向于最先输出，这使所有 VOQ 队列长度和队首信元等待时间趋于平衡，可以较好的改善调度算法的延时和丢包性能。

图 1 和图 2 描述了一个 3×3 的交换结构采用 low-FIRM 调度算法进行两次迭代的工作情况，假设每个 VOQ 的信元容量为 4 个。可以看出，该算法通过在第一次迭代的接受步骤中将输入端口 RRI 指针指向权值 W 最大的 VOQ 对应的输出端口，使得两次迭代后，权值最大的 VOQ 均得到匹配，输出信元。

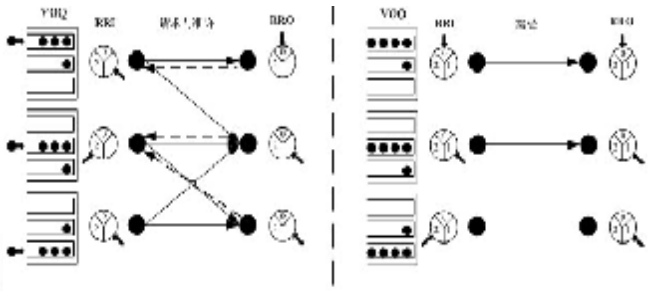


图 1　low-FIRM 算法第一次迭代

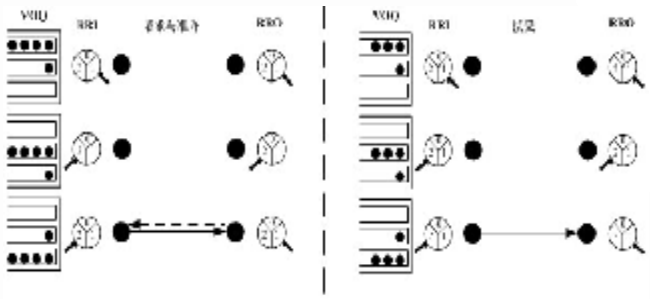


图 2　low-FIRM 算法第二次迭代

图 3 描述了同样初始状态采用 FIRM 算法进行两次迭代的调度结果。在下一时隙，如果 VOQ2,2 和 VOQ3,3 继续有信元到达，由于超出了队列容量，会造成丢包，同时，这两个虚拟输出队列的队首信元的延时继续累加，使队列的平均延时加大。

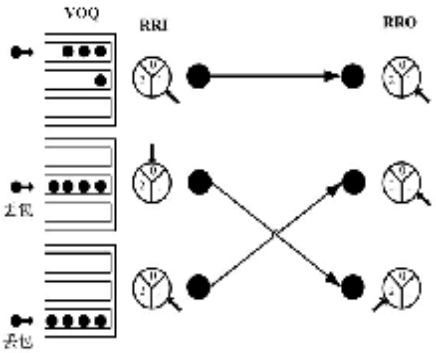


图 3　FIRM 算法两次迭代调度结果

4 算法仿真

由于 iSLIP 算法性能优于 PIM 和 RRM 算法，并且已经被应用于高速路由器[8]，而 FIRM 算法性能略高于 iSLIP，因此，本文主要对 iSLIP、FIRM 和 low-FIRM 三种算法进行详尽的仿真和性能对比。仿真时间设为 100000 个时隙，三种算法的迭代次数均为 5。

4.1 交换机模型

本文所仿真的交换机的端口配置为 32×32，采用 VOQ 的存储组织方式，每个 VOQ 的最大信元容量为 500 个，不同 VOQ 间相互独立，不进行缓冲共享。

4.2 信元到达模式

仿真实验中实现了两种信元到达模式：

(1) 独立同分布到达过程，即 Bernoulli 过程。每一时隙，信元以固定的概率到达，到达的概率与信元的历史情况无关。

(2) 突发到达过程。信元的到达过程采用两状态的马氏过程，即 ON-OFF 模型。ON 状态表示有信元到达，连续到达的突发信元的目的端口相同，反之，OFF 状态表示没有信元到达。ON 状态的保持时间服从均值为 64 的几何分布，OFF 状态的保持时间服从均值为 64(1 - λ) / λ 的几何分布。其中 λ 表示流量的负荷。

4.3 流量分布模式

λ ∈(0, 1]为归一化的流量负载，本文采用如下常用的流量分布模型[9-11]：

1) 均匀分布模型。若输入端口有信元进入，信元的目的端口为 32 个输出端口中任意一个的概率均为 λ /32。

2) 强对角分布模型。若输入端口 i 有信元进入，设信元的目的端口为 j 的概率为 P_{ij}，则：

$$P_{ij}=\begin{cases}\frac{2}{3}, j=i\\ \frac{1}{3}, j=i+1\\ 0, else\end{cases} \quad (1)$$

3) 弱对角分布模型。若输入端口 i 有信元进入，设信元的目的端口为 j 的概率为 P_{ij}，则：

$$P_{ij}=\begin{cases}\frac{2}{3}, j=i\\ \frac{1}{3\times(32-1)}, j\neq i\end{cases} \quad (2)$$

4.4 性能指标

仿真实验中用来衡量算法性能的指标有：

1) 平均延时。设算法在时间区间[0, 100000]内共输出了 n 个信元，其中第 i 个通过交换结构的信元在 VOQ 队列中等待的时间为 $delay_i$ 个时隙，1≤i≤n，则平均延时 mean_delay 为：

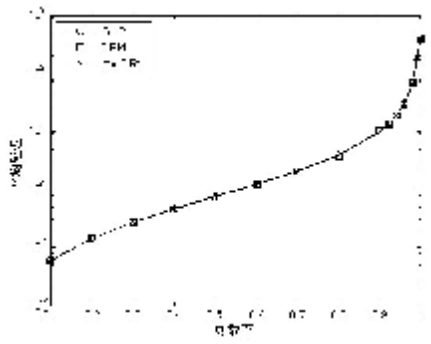
$$mean\_delay=\frac{\sum_{i=1}^{n} delay_i}{n} \quad (3)$$

2) 丢包率。当信元到达队列已满的 VOQ 时，该信元会被丢弃。设算法在时间区间[0, 100000]内共输出了 n 个信元，共丢弃了 drop_num 个信元，则丢包率 drop_rate 为：

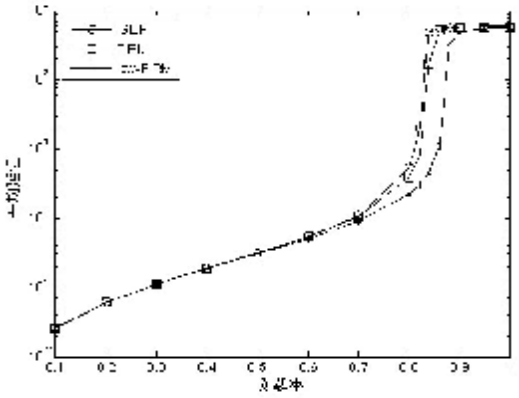
$$drop\_rate=\frac{drop\_num}{n} \quad (4)$$

4.5 延时性能

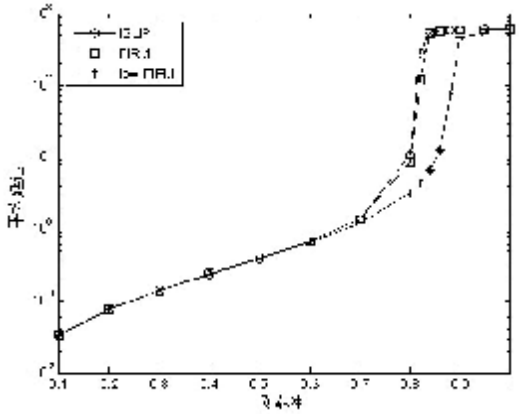
图 4 是 low-FIRM 算法同经典的 iSLIP 算法、FIRM 算法的延迟性能的比较。从图 4(a)中可以看出，在均匀流量模型下，low-FIRM 算法与经典算法一样能取得较好的性能。由图 4(b)和图 4(c)可知，在强对角流量和弱对角流量下，low-FIRM 算法在负载率为 0.7 到 0.9 之间的平均延时性能比两种经典算法有较大的提高，负载率在 0.8 至 0.9 之间时，平均延时降低了一到两个数量级。在突发流量下，low-FIRM 算法的延时性能比经典算法有所提升，在负载率为 0.92 时，平均延时比 iSLIP 算法降低了 15%，如图 4(d)所示。



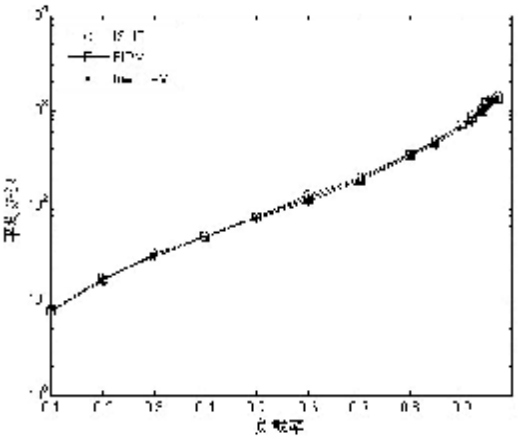
(a) 均匀独立同分布流量



(b)　强对角独立同分布流量



(c)　弱对角独立同分布流量



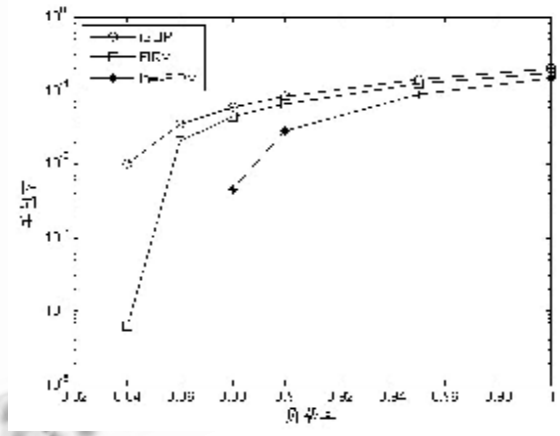
(d)　突发流量

图 4　不同流量模型的平均延时-负载曲线

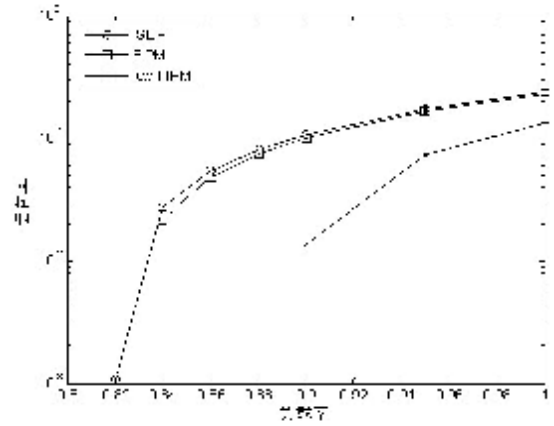
4.6 丢包性能

在均匀流量模型下，low-FIRM 算法与经典算法一样，丢包率为零。

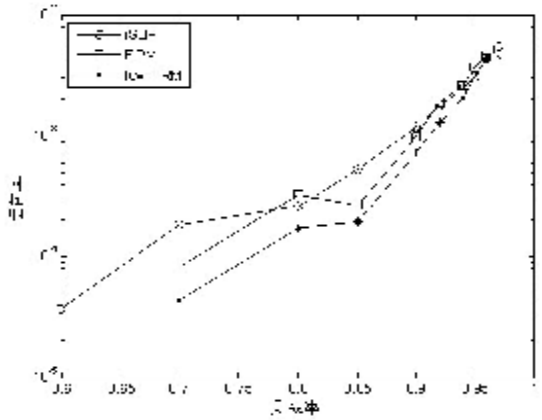
图 5 是 low-FIRM 算法同经典的 iSLIP 算法、FIRM 算法在非均匀流量下丢包性能的比较。由图 5(a)可知，在强对角流量下，low-FIRM 算法出现丢包的负载率比经典算法高 4%，并且丢包率减小了 99%。在弱对角流量下，low-FIRM 算法出现丢包的负载率比 iSLIP 算法高 8%，并且丢包率减小了 87%，如图 5(b)所示。而从图 5(c)可以看出，在突发流量下，low-FIRM 算法出现丢包的负载率比 iSLIP 算法高 10%，并且丢包率减小了 77%。对比图 4(d)和图 5(c)，可以看出新算法在处理突发流量时，对丢包性能的提升高于延迟性能的提升，出现这种现象是由于突发流量保持 ON 状态时，输入端口连续到达多个目的端口相同的信元，相应的 VOQ 出现队列满而导致丢包的概率高于其他几种流量。综上所述，改进算法不仅能保证在更高的负载率时才出现丢包现象，而且在各种流量模型和负载率下，丢包性能都明显优于两种经典算法。



(a)　强对角独立同分布流量



(b)　弱对角独立同分布流量



(c) 突发流量

图 5 不同流量模型的丢包率-负载曲线

5 结语

作者通过对经典的 FIRM 算法的指针策略的优化，设计了一种 low-FIRM 算法，它针对 FIRM 算法在非均匀流量下的性能缺陷进行了改进。仿真结果表明，low-FIRM 算法不仅保持了经典算法在均匀流量下的优良性能，而且在对角流量和突发流量等非均匀流量下性能有了较大的提升，同时算法的复杂度与 FIRM 算法相近，可用在骨干网核心路由器中。

参考文献

1 Karol M, Hluchyj M, Morgan S. Input versus output queueing on a space division switch. Proceeding of the Global Telecommunications Conference. New York, IEEE, 1987:659－665.

2 Anderson T, Owicki S, Saxes J. High speed switch scheduling for local area networks. ACM Transactions on Computer Systems, 1993.11(4):319－346.

3 Mckeown N. The iSLIP scheduling algorithm for input-queued switchs. IEEE Trans. on Networking, 1999.7(2):188－201.

4 Serpanos DN, Atoniadis PI. FIRM: A class of distributed scheduling algorithms for high-speed ATM switches with multiple input queues. Proceedings of the Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Tel Aviv, Isreal: IEEE, 2000:548－555.

5 Giaccone P, Shah D, Prabhakar B. An implementable parallel scheduler for input-queued switches. IEEE Micro, 2002.22(1):19－25.

6 Mekkitikul A, Mckeown N. A practical scheduling algorithm to achieve 100% throughput in input-queued switches. Proceedings of the Seventeenth Annual Joint Conference of the IEEE Computer and Communications Societies. San Francisco: IEEE, 1998: 792－799.

7 杨黎莉, 蒋震艳, 杜新华. 一类基于Crossbar的分布式调度算法的分析与比较. 计算机工程, 2002. 28(10):96－98.

8 姜小波,杜小伟. 一种高速 crossbar 调度算法及性能分析. 计算机应用, 2010,30(1):101－103.

9 Li Y, Panwar S, Chao HJ. The dual round-robin matching switch with exhaustive service. Proceedings of the IEEE HPSR 2002, Kobe, Hyogo, 2002:58－63.

10 Li Y, Panwar S, Chao HJ. Exhaustive service matching algorithm for input queued switches. Proceedings of the IEEE HPSR 2004, Phoenix, Arizona, 2004:253－258.

11 Bianco A. A framework for differential frame-based matching algorithms in input queued switches. Proceedings of the IEEE INFOCOM 2004, Hong Kong, 2004:1147－1157.