ISSN 1000-9825, CODEN RUXUEW
*Journal of Software*, Vol.19, No.7, July 2008, pp.1856−1864
DOI: 10.3724/SP.J.1001.2008.01856

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel/Fax: +86-10-62562563

# 一种公平服务的动态轮询调度算法*

扈红超[1+], 伊 鹏[1], 郭云飞[1], 李玉峰[1,2]

[1](国家数字交换系统工程技术研究中心,河南 郑州 450002)

[2](防空兵指挥学院 信息控制系,河南 郑州 450052)

## A Fair Service and Dynamic Round Robin Scheduling Algorithm

HU Hong-Chao[1+], YI Peng[1], GUO Yun-Fei[1], LI Yu-Feng[1,2]

[1](National Digital Switching System Engineering and Technological R&D Center, Zhengzhou 450002, China)

[2](Department of Information and Control, Air Defense Command College, Zhengzhou 450052, China)

+ Corresponding author: E-mail: huhongchao@gmail.com

**Hu HC, Yi P, Guo YF, Li YF. A fair service and dynamic round robin scheduling algorithm. *Journal of Software*, 2008,19(7):1856−1864.** http://www.jos.org.cn/1000-9825/19/1856.htm

**Abstract**: Scheduling policies are playing significant roles in guaranteeing the performance of core routing and switching devices. The limitations in complexities and extensibilities of current combined input and cross-point queueing switching fabric's scheduling policies are first analyzed. Then, based on this analysis, the principle for designing high extensible scheduling policies and the concept of virtual channel are proposed. Based on the principle and virtual channel, it comes up with a dynamic round robin scheduling algorithm-FDR (fair service and dynamic round robin), which is simple, and of high efficiency and fair service. FDR is based on round robin mechanism, whose complexity is only $O(1)$. It allocates the scheduling share for each virtual channel according to its current states, thus, FDR has good dynamic and real-time performance, and it can adapt to unbalanced traffic load network environment. Simulation results under SPES (switching performance evaluation system) show that FDR exhibits good delay, throughput and anti-burst performance.

**Key words**: switching fabric; scheduling policy; buffered crossbar; dynamic round robin; SPES (switching performance evaluation system)

**摘 要**: 调度策略是核心路由交换设备性能的重要保证.针对联合输入交叉节点排队(combined input and cross-point queuing,简称 CICQ)交换结构现有调度策略在复杂度或性能方面存在的缺陷,深入探讨了 CICQ 交换结构调度策略设计的基本准则,并提出了 CICQ 下虚拟通道的概念.基于基本准则和虚拟通道概念,提出一种简单、高效和公平服务的动态轮询调度策略——FDR(fair service and dynamic round robin).其算法复杂度为 $O(1)$,具有良好的可扩展性;并依据虚拟通道的状态为其分配调度份额,具有良好的动态实时性能,能够适应流量负载非均衡的网络环境.SPES(switching performance evaluation system)仿真结果表明,该算法具有良好的时延、吞吐量和抗突发性能.

* Supported by the National Natural Science Foundation of China under Grant No.60572042 (国家自然科学基金); the National High-Tech Research and Development Plan of China under Grant No.2005AA121210 (国家高技术研究发展计划(863)); the National Basic Research Program of China under Grant No.2007CB307102 (国家重点基础研究发展计划(973))

Received 2007-08-10; Accepted 2008-02-20

关键词: 交换结构;调度策略;带缓存交叉开关;动态轮询;交换系统性能仿真评价系统
中图法分类号: TP316 文献标识码: A

通信网络的多元化发展与骨干通信容量的迅速膨胀,不仅要求网络核心交换设备具备大规模的交换容量,同时要求交换设备能够在高速环境下稳定运行,并能够为不同业务提供有保障的服务质量(quality of service,简称 QoS).输出排队(output queued,简称 OQ)交换结构在性能上极具优势,然而要实现 $N\times N$ 规模的 OQ 交换结构,交换单元和存储单元都需工作于 $N$ 倍线路速率,这使得 OQ 交换结构在构建大容量交换系统时不具备良好的可扩展性.与 OQ 相比,输入排队(input queued,简称 IQ)交换结构的交换单元和存储单元均只需工作于线路速率[1],因而对于构建大容量交换系统是一种十分有效的解决方案,然而 IQ 交换结构需要集中控制式的调度机制,这种集中控制式的调度机制成为构建大容量交换系统的瓶颈.虽然理论上已经证明,基于最大权重匹配(maximum weight matching,简称 MWM)的调度策略在任意可容许业务流下可获得 100%的吞吐量,然而其复杂度是 $O(N^3\log N)$[1],对交换结构的规模过于敏感.虽然研究人员针对 IQ 交换结构提出种种解决方案,试图降低其实现复杂度,然而这种复杂度降低是以牺牲 IQ 交换系统的性能为代价的.

近年来,随着微电子技术的进步,在交换单元内部集成一定容量的缓存单元成为了现实[2],并成为交换结构技术领域的研究热点.基于带缓存交叉开关(buffered crossbar)构建的联合输入交叉节点排队(combined input-cross point-queued,简称 CICQ)交换结构由于性能优势更是备受关注[3].Javidi 等人证明了当输入输出端口的业务流到达率 $\lambda_{ij}\leq 1/N$ 时,在无须加速的条件下 CICQ 交换结构即可获得 100%的吞吐量[4].随后,Magill 等人证明了 CICQ 交换结构在 2 倍加速条件下可以通过分布式调度算法模拟支持 $k$ 个优先级的先入先出输出排队交换结构(first come first serve-output queuing,简称 FCFS-OQ)[5].Chuang 等人进一步证明了在 3 倍加速条件下带缓存 CICQ 交换结构可以通过分布式调度算法实现模拟任意调度算法的 OQ 交换结构[6].CICQ 交换结构优于 IQ 交换结构的关键在于,在每个交叉点(cross-point)集成了一定容量的缓存,从而将输入端口与输出端口的带宽资源冲突隔离开来,分布式和并行调度策略的设计与实现成为了可能.

针对目前 CICQ 调度策略在复杂度、稳定性或可扩展性等方面存在的缺陷,提出了一种公平服务的动态轮询调度策略(fair service and dynamic round robin,简称 FDR).其吞吐量接近 100%,且算法复杂度仅为 $O(1)$.FDR 的基本思想是为每个输入端口维护一个轮询指针和一个份额计数器.FDR 根据虚拟输出队列和对应交叉点缓存队列构成的虚拟通道的状态为其分配一定调度份额.当服务份额达到调度份额时,指针转向下一个虚拟输出队列.由于调度份额是基于当前系统中虚拟输出队列和交叉点缓存队列的实时状态得到的,这样就保证了去往不同目的端口业务流调度的公平性,并可以动态适应流量分布多变的网络环境.SPES(switching performance evaluation system)[7]仿真表明:FDR 具有良好的时延、吞吐量和抗突发性能.本文第 1 节首先介绍 CICQ 交换结构及其典型调度策略,并给出其调度策略设计的基本准则.第 2 节给出公平服务的动态轮询调度策略 FDR,并分析份额分配函数的选取准则.最后给出在 SPES 仿真平台下 FDR 算法的性能评估结果,并进行比较.

## 1 背景及相关研究

### 1.1 联合输入交叉节点排队交换结构

CICQ 交换结构采用带缓存交叉开关作为核心交换单元构建,为了解决队头阻塞问题,CICQ 交换结构通常采用虚拟输出排队机制.图 1 给出了一个典型的 $N\times N$ 规模的联合输入交叉节点排队交换结构模型.在该结构中,每个输入缓存单元从逻辑上被分割为 $N$ 个虚拟输出队列,每个虚拟输出队列对应于一个交叉点缓存队列.对该图中交叉节点队列的位置进行简单的调整后,就可以得到联合输入交叉节点排队交换结构更为直观的等效结构模型.可以看出:通过在交叉点集成一定容量的缓存,整个交换结构从逻辑上划分为 $N$ 个 $1\times N$ 和 $N$ 个 $N\times 1$ 子交换结构.

为了便于分析,在后文的讨论中假设 CICQ 交换结构是基于时隙(slot)的,每个分组为固定信元(cell)大小,一

个时隙为一个信元在交换单元中的传输时间(实际中的 ATM 数据分组为固定信元大小,而 IP 分组大小在 20 字节~1 500 字节之间,将分组大小固定为信元不影响性能分析).

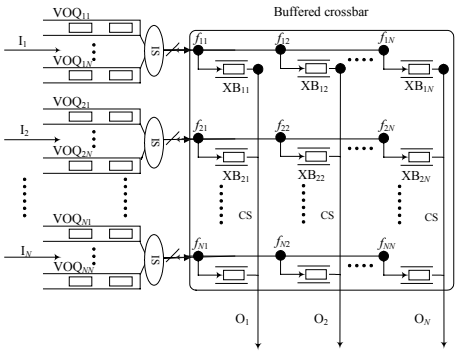


Fig.1 An $N\times N$ CICQ switch architecture

图 1 $N\times N$ 规模带缓存交叉开关交换结构模型

### 1.2 CICQ调度策略相关研究

基于 CICQ 设计的调度策略整体上可以分为两类:基于权重匹配的调度策略和基于轮询的调度策略.基于权重匹配的调度策略一般借助排序操作为每一输入端口匹配一个交叉点缓存队列,如 OCF-OCF(oldest cell first)[8]和 LQF(longest queue first)-RR 等.排序操作通常由比较器和 MUXes 来实现,当采用树形比较方式时,OCF-OCF 和 LQF-RR 的算法复杂度为 $O(\log N)$.从硬件实现的角度上讲,即使采用最优的实现方式,两个整数的比较操作依然需要 $O(\log B)$的延时,其中 $B$ 为 $\log L_{\max}$[9].对于 OCF-OCF 调度策略,$L_{\max}$ 为信元等待时间的最大值;对 LQF-RR 而言,$L_{\max}$ 为 VOQ 队列的最大长度.对于核心路由交换设备而言,缓存容量可以达到几十甚至上百兆,无论是 OCF-OCF 算法还是 LQF-RR 算法,$L_{\max}$ 值的大小都是无法忍受的.可见,OCF-OCF 和 LQF-RR 算法都不是构建大规模交换系统的有效解决方案.

针对 OCF-OCF 和 LQF-RR 算法复杂度过高的问题,研究人员提出了 SCBF(shortest crosspoint buffer first)[10]和 MCBF(most critical buffer first)[9].这些算法的共同点是都从交叉点缓存的状态出发考虑调度算法的设计,并将 $B$ 的值降为 $\log(N\cdot S)$,$S$ 为交叉点缓存的容量.SCBF 和 MCBF 算法的复杂度分别为 $O(N\log N)$和 $O(\log N)$.虽然仿真结果表明它们都能够在不同业务流模型下获得良好的时延性能,然而不足之处在于:由于在设计调度策略时仅仅考虑了交叉点缓存队列的状态而没有考虑输入端口虚拟队列的状态,因而会导致虚拟输出队列的不稳定性.分析结果表明,SCBF 和 MCBF 调度策略的稳定性比 OCF-OCF 和 LQF-RR 要差.此外,SCBF 和 MCBF 调度策略的复杂度依然对交换结构的规模比较敏感,并且无论采用交叉点缓存的片内排序还是片外排序都会给芯片引脚资源带来压力,因而在构建大规模交换系统时同样受到了限制.

基于轮询机制的调度策略有 RR-RR[11],DRR(differential round robin)[12]和 DRR-$k$(dual round robin)[13]等.虽然 RR-RR 算法的实现复杂度仅为 $O(1)$,但其不能提供良好的吞吐量、时延和公平性能.DRR 和 DRR-$k$ 是对 RR-RR 调度算法的改进,DRR 调度算法的基本思想是,为当前 VOQ 队列服务后,若当前 VOQ 队列长度依然大于其后 VOQ 队列长度的 2 倍,则轮询指针保持不变,否则轮询指针指向下一个 VOQ 队列;DRR-$k$ 算法引入了双轮询指针,并依靠主指针为每个 VOQ 队列分配固定的调度份额并进行更新,借助辅助指针提高吞吐量.DRR 和 DRR-$k$ 都在 $O(1)$的算法复杂度下获得较好的吞吐量和时延性能.然而它们的不足之处在于:设计调度策略时没有考虑交叉点缓存队列的状态;同时,主轮询指针的更新条件选择上过于简单,灵活性较差,抗突发性能受到了限制.

### 1.3 CICQ调度策略设计基本准则

带缓存 Crossbar 交换单元通过在每个交叉节点植入一定容量的缓存单元,从而使分布式与并行调度策略的实现成为现实.为了实现分布式的调度策略,每个输入与输出端口都要实现调度单元,这些调度单元并行工作

共同完成分组的调度任务.对于单个调度单元,算法的复杂度大为降低,如 CICQ 调度单元的复杂度基本在 $O(\log N)$数量级上,使支持组播交换、QoS 保证和构建更大规模的交换系统成为了可能.然而调度策略的这种分布式特性以及输入和输出调度策略的隔离,很难保证交换系统在每个时隙整体上都达到最佳匹配状态,如何设计出具有高吞吐量同时又能保留这种分布式特性的调度策略是目前 CICQ 交换结构研究的关键问题.

基于轮询机制的调度策略 RR-RR 公平地对待所有 VOQ 队列,仅考虑了 VOQ 中有无信元,而没有考虑到信元的多少与信元的状态,相对于高速业务流而言,低速业务流获得了较为及时的服务,造成了高速业务流与低速率业务流之间服务的不公平性.LQF-RR 和 OCF-OCF 对 VOQ 队列的长度和信元的等待时间进行了考虑,从而获得了性能上的提升,具有较好的稳定和时延性能;然而它们的算法复杂度与交换结构规模有关,限制了其可扩展性.SCBF 和 MCBF 对交叉点队列的状态进行了考虑,试图在所有输入与输出端口之间寻求一种最佳匹配,由于没有考虑 VOQ 队列的状态,虽然获得了良好的时延性能,然而和 RR-RR 一样,仍然存在不稳定现象.DRR 和 DRR-$k$ 算法对 RR-RR 算法轮询指针的更新规则进行了改进,在保持 $O(1)$算法复杂度的同时获得了较好的性能.

从以上分析可以看出,要提高 CICQ 交换系统的稳定性,在设计调度策略时必须考虑 VOQ 队列的状态;要获得交换系统整体性能的提升,还应使输入调度与交叉点调度之间具有一定的匹配关系;此外,调度策略的设计还应考虑到交换系统的可扩展性问题.这里的可扩展性包含两方面的内容:一是从硬件实现复杂度的角度上讲,要求硬件实现上要尽量简单,并应尽量避免比较和排序等复杂操作;二是调度策略的算法复杂度不应对交换结构的规模过于敏感,这是调度策略可扩展性的重要准则.可以看出,由于基于轮询机制设计的调度策略的算法复杂度仅为 $O(1)$,因而对于 CICQ 交换系统而言,是一类十分理想的调度模型.本文后面的内容从以上准则出发,重点研究基于轮询的调度策略,并提出具有可扩展性和良好性能的 FDR 调度算法.

## 2 公平服务的动态轮询调度策略 FDR

本节基于如图 1 所示的结构模型介绍 FDR 调度策略的设计.CICQ 结构的整个调度过程分为两个阶段:输入调度(input scheduling,简称 IS)和交叉点调度(cross-point scheduling,简称 CS).在每个时隙,每个 IS 调度单元从其输入端口的 $N$ 个 VOQ 队列中选出一个;每个 CS 调度单元从对应的 $N$ 个交叉点缓存(cross-point buffer,简称 XB)中选出一个.下面分别从输入调度和交叉点调度两个方面详细介绍 FDR 调度策略的设计.

### 2.1 定义及相关符号

为了便于描述,首先给出描述 $N\times N$ 规模 CICQ 交换结构的若干定义:

**定义 1**. 集合 $V=\{VOQ_{ij}|0\le i\le N,0\le j\le N\}$,其中,$VOQ_{ij}$ 缓存到达输入端口 $i$ 去往输出端口 $j$ 分组的虚拟输出队列.$v_{ij}(t)$为 $t$ 时刻 $VOQ_{ij}$ 的占有率,$0\le t\le T$,$T$ 为系统仿真时间,$0\le v_{ij}(t)\le C_q$,$C_q$ 为 VOQ 队列容量.

**定义 2**. 集合 $C=\{XB_{ij}|0\le i\le N,0\le j\le N\}$,其中,$XB_{ij}$ 二次缓存到达输入端口 $i$ 去往输出端口 $j$ 分组的交叉点队列,与 $VOQ_{ij}$ 一一对应.用 $l_{ij}(t)$表示 $t$ 时刻系统中 $XB_{ij}$ 队列长度,$0\le t\le T$,$0\le l_{ij}(t)\le C_l$,$C_l$ 为 $XB_{ij}$ 队列的容量.

**定义 3**. 传输到达输入端口 $i$ 去往输出端口 $j$ 分组的 $VOQ_{ij}$,$XB_{ij}$ 以及相关硬件资源,称为虚拟通道(virtual channel),记为 $VC_{ij}$,其状态用 $s_{ij}(t)$描述.

**定义 4**. 称 $VOQ_{ij}$ 在 $t$ 时刻是“候选的(eligible)”,若满足 $0<q_{ij}(t)\le C_q$ 且 $0\le l_{ij}(t)<C_l$,输入端口 $i$ 对应的候选 VOQs 的集合用 $E_i$ 表示,则 IS 为 $E_i$ 的函数.

**定义 5**. 称 $XB_{ij}$ 在 $t$ 时刻是“候选的”,若满足 $0<l_{ij}(t)\le C_l$,输出端口 $j$ 对应的候选 XBs 的集合用 $E'_j$ 表示,则 CS 为 $E'_j$ 的函数.

#### 2.1.1 输入调度 IS 的设计

FDR 为每一输入调度单元维护了一个轮询指针与一个份额计数器.分别用 $\boldsymbol{P}(t)=[p_i(t)]_{1\times N}$ 和 $\boldsymbol{Q}(t)=[q_i(t)]_{1\times N}$ 表示时刻 $t$ 输入端口轮询指针的位置向量和份额计数器的状态向量.现在讨论在 $t=n$ 时隙时,输入端口 $i$ 的 $p_i(n)$ 和 $q_i(n)$的更新机制.首先假设当前轮询指针的位置为 $p_i(n)=j$,IS 调度单元首先判断 $q_i(n)$是否大于 0,若 $q_i(n)>0$,并且满足 $VOQ_{ij}\in E_i$,则在 $n+1$ 时隙 $p_i(n+1)=j$,$q_i(n+1)=q_i(n)-1$;否则,$p_i(n)$和 $q_i(n)$按照如下规则进行更新:从 $j$ 开始,FDR 按照轮询规则找到第 1 个满足 $VOQ_{ik}\in E_i$ 的 $VOQ_{ik}$ 队列,即 $p_i(n+1)=k$;同时,FDR 依据对应虚拟通道 $VC_{ij}$

的状态 $s_{ik}(n+1)$按照规则$\Gamma$为其分配调度份额 $q_i(n+1)$,即 $q_i(n+1)=\Gamma(s_{ik}(n+1))$.从此刻开始,IS 持续为 $VOQ_{ik}$ 进行服务直到在 $t=n'$时刻,$q_i(n'+1)=0$ 或者 $VOQ_{ik}$ $E_i$.用伪码表示 IS 调度策略的调度过程如图 2 所示.

2.1.2 交叉点调度 CS 的设计

FDR 为每个交叉点调度单元 CS 维护两个轮询指针,主轮询指针和辅助轮询指针.用 $\boldsymbol{Z}_m(t)=[z_{mj}(t)]_{1\times N}$ 和 $\boldsymbol{Z}_a(t)=[z_{aj}(t)]_{1\times N}$ 表示 $t$ 时刻主轮询指针与辅助轮询指针的位置向量.讨论在 $t=n$ 时隙时,输出端口 $j$ 对应的交叉点调度单元两个指针的更新机制.首先假设主轮询指针和辅助轮询指针位置为 $z_{mj}(n)=i,z_{aj}(n)=i'$.CS 将输出端口 $j$ 对应的 XBs 分成了两类:被 IS 的轮询指针选中且阻塞的 XB 属于第 1 类,记为有序集合 $\boldsymbol{V}=\{XB_{ij}|i\in\boldsymbol{P}(n),l_{ij}(n)=C_l\}$;剩余的 XB 属于第 2 类,记为有序集合 $\overline{\boldsymbol{V}}$.在每个时隙内 $\boldsymbol{Z}_m(n)$和 $\boldsymbol{Z}_a(n)$的更新机制如下:若 $\boldsymbol{V}$ 不为∅(空集),则从 $i$ 开始,FDR 按照轮询规则从 $\boldsymbol{V}$ 中找到第 1 个满足 $XB_{kj}\in E'_j$ 的 XB,即 $z_{mj}(n+1)=k$;若 $\boldsymbol{V}$=∅或者没有满足条件的 XB,则从 $i'$开始,FDR 按照轮询规则从 $\overline{\boldsymbol{V}}$ 中找到满足 $XB_{kj}\in E'_j$ 第 1 个 XB,即 $z_{aj}(n+1)=k$;用伪码表示 CS 调度策略的调度过程如图 3 所示.

Input arbiter *IA*:

```
Input port i:
If (q_i(n)==0 || VOQ_ij ∉E_i ){
    For (index=0; index<N; index++)
    {
        k=(p_i(n)+index)%N;
        If(VOQ_ik∈E_i)
        {
            p_i(n+1)=k;
            q_i(n+1)=Γ(s_ik(n+1));
        }

    Else{
      q_i(n+1)=q_i(n+1)--;
    }
```

Fig.2 Pseudo-Code specifying IS arbiters

图 2 FDR 输入调度 IS 的伪码描述

Output arbiter *OA*:

```
Output port j:
flag=0;
If (V !=∅){
    For (index=0; index<N; index++){
        k=(z_mj(n)+index)%N;
        If(XB_kj∈E_j' && XB_kj∈V){
            flag=1;
            z_mj(n+1) = k;}}}
Else{
    For(index=0; index<N; index++){
        k=(z_aj(n)+index)%N;
        If(XB_kj ∈E_j'){
            z_aj(n+1) = k;}}
}
```

Fig.3 Pseudo-Code specifying CS arbiters

图 3 FDR 交叉点调度 CS 的伪码描述

2.1.3 规则$\Gamma$的讨论

在输入调度 IS 的工作过程中,当输入端口 $i$ 轮询到某一 $VOQ_{ij}$ 时,IS 按照规则$\Gamma$依据 $VC_{ij}$ 的状态 $s_{ij}(t)$为其分配本次轮询的调度份额,即 $q_{ij}(t)=\Gamma(s_{ij}(t))$,因此$\Gamma$的选取影响着 FDR 的性能.可以看出,若$\Gamma(s_{ij}(t))=1$,则该算法逼近为 RR-RR 调度策略;若$\Gamma(s_{ij}(t))=C$($C$ 为常量),则该算法逼近为 DRR-$k$ 调度策略.直观上看,如果当前虚拟通道的状态较为恶劣,比如具有较高的通道占有率、获得较少的服务率或分组平均延时较大等,为了缓解这种状况,该虚拟通道应分配更多的服务份额,因而$\Gamma$应是恶劣程度的递增函数.用 $x$ 表示恶劣程度的量化值,则 $q_{ij}(t)$为 $x$ 和$\Gamma$的函数,并且$\Gamma$应满足如下规则:

**规则 1**. $\Gamma(0)=0$.

**规则 2**. $\Gamma(\cdot)$是有界增函数,$\forall x_1<x_2,0\le\Gamma(x_1)\le\Gamma(x_2)\le C_q$.

**规则 3**. 为了支持变长分组交换,$\Gamma(\cdot)$应是可微函数并有$\Gamma'(\cdot)\ge 0$.

规则 1 表明,若当前 $VC_{ij}$ 的恶劣程度为 0,则不为其分配调度份额,例如当 $s_{ij}(t)$为 VOQ 队列长度时,‘0’表示 VOQ 队列为空,因而不需要为其分配调度份额,这是由 FDR 的调度机制决定的;规则 2 说明,$\Gamma$为恶劣程度的增函数,由于每次分配的调度份额不能超过 VOQ 队列的占有率,因而$\Gamma$为有界函数;规则 3 是对规则 2 的扩展,说明在变长分组交换下规则$\Gamma(\cdot)$是恶劣程度的可导可微函数,即对于任意 $x_0$,有$\Gamma(x_0+\Delta x)-\Gamma(x_0)=A\Delta x+o(\Delta x)$,其中 $A$ 为常数.$\Gamma(\cdot)$可以是线性的也可以是非线性的,只要满足规则(1)~规则(3)即可.不同的份额函数具有不同的硬件实现复杂度,下面分别针对两类函数进行分析和讨论.

**线性函数**.线性函数具有 $f(x)=ax+b$ 的形式,$0\le x\le C_q$,如图 4 所示,由规则 1 可知,$b=0$.由于 $s_{ij}(t)$是离散的,因而 $s_{ij}(t)$到 $q_{ij}(t)$的影射是有误差的,最大误差为一个信元的大小,这里采用向上取整的规则,即 $q_{ij}(t)=\lceil a\times s_{ij}(t)\rceil$.线性函数具有硬件实现简单、延时小等优点;其缺点是这种线性增长的灵活性差,对网络状态的变化反应较为“迟钝”.

**非线性函数**.与线性函数相比,非线性函数具有较好的动态特性,可以较为“灵敏”地感应网络状态的变化,如图 5 所示.其缺点是硬件上不易实现,硬件资源开销大.非线性函数有多项式函数、指数函数和对数函数等.多项式函数具有 $f(x)=a_nx^n+a_{n-1}x^{n-1}+\ldots+a_1x+a_0$ 的形式,由规则 1 可知,$a_0=0$,则 $q_{ij}(t)=\lceil a_nx^n+a_{n-1}x^{n-1}+\ldots+a_1x\rceil$.

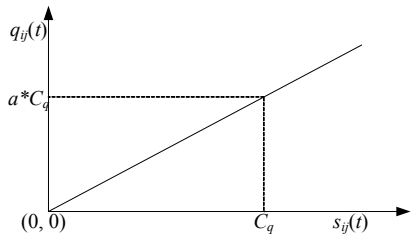


Fig.4 Linear allocating function curve

图 4 线性调度份额分配函数

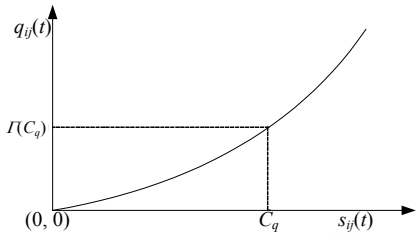


Fig.5 Nonlinear allocating function curve

图 5 非线性调度份额分配函数

可以看出,FDR 是一种“尽力而为”的调度策略.IS 调度阶段,FDR 依据交换系统的当前状态(而这种状态反映了网络中的流量分布)来分配虚拟通道的调度份额,并在输出阶段首先对拥塞的通道进行处理,因而具有良好的动态和公平特性.兼顾处理上的简单性和稳定性,这里选取 VOQ 队列的占有率作为 VC 的状态进行讨论,即 $s_{ij}(t)=v_{ij}(t)$.

## 3 性能评估

本节基于交换系统性能仿真评价系统 SPES 对 FDR 调度策略的时延、吞吐量和抗突发性等方面进行性能评估.为了便于说明 FDR 调度策略的性能,同时给出了 IQ 调度策略 *i*SLIP、OQ 调度策略 FQ 和 CICQ 调度策略 LQF-RR,SCBF-RR,MCBF 以及 RR-RR,DRR 和 DRR-*k* 的仿真结果.仿真采用 Bernoulli 和 ON-OFF 突发业务源模型,且业务流的到达过程均为可允许的,突发业务源的突发长度为 20.

### 3.1 时延性能仿真

时延性能仿真均采用 16×16 规模交换结构,系统仿真时间为 100 000 个时隙,业务流分布模型采用均匀分布(uniform)分布和 Diagonal 分布两种.用 表示输入端口 $i$ 总的业务流强度, 表示到达输入端口 $i$ 去往输出端口 $j$ 的业务流强度,则对于 Uniform 分布有 ;对于 Diagonal 分布,若 $j=i$,则 ,若 $j=(i+1) \bmod N$,则 ,否则 .

#### 3.1.1 Bernoulli 业务源

图 6 给出了在 Bernoulli 业务源 Uniform 与 Diagonal 分布下各种算法的时延性能仿真结果.可以看出:在 Bernoulli 业务源模型下,FDR 和 SCBF,LQF 以及 DRR-*k* 的性能相当,比 FQ 算法略差,优于 *i*SLIP,RR 和 DRR 算法.

#### 3.1.2 ON-OFF 突发业务源

图 7 给出了在 ON-OFF 突发业务源 Uniform 与 Diagonal 分布下各种算法的时延性能仿真结果.可以看出:在 ON-OFF 突发业务源模型下,FDR 算法可以与 SCBF 算法匹比,优于 *i*SLIP,RR,DRR 和 DRR-*k* 算法,其中 FQ 算法的性能最好,而 FQ 算法需要 $N$ 倍的加速作为代价.

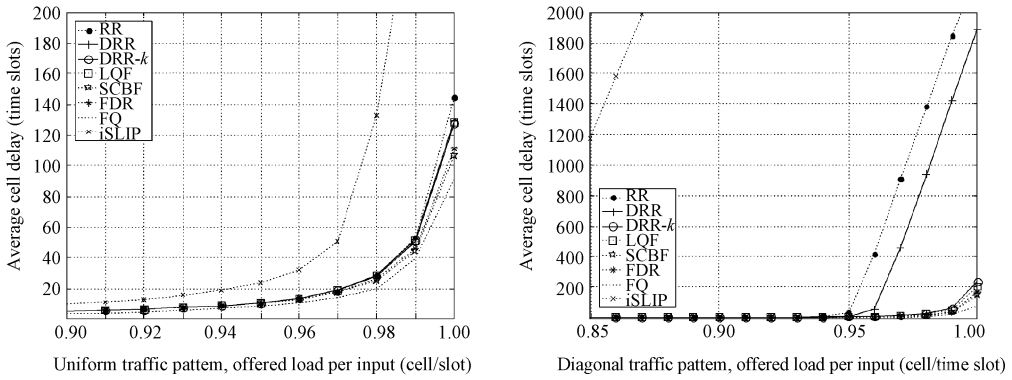


Fig.6 Delay performance of various schemes under Bernoulli uniform and diagonal traffic

图 6 贝努利业务源下各种算法的时延性能仿真结果

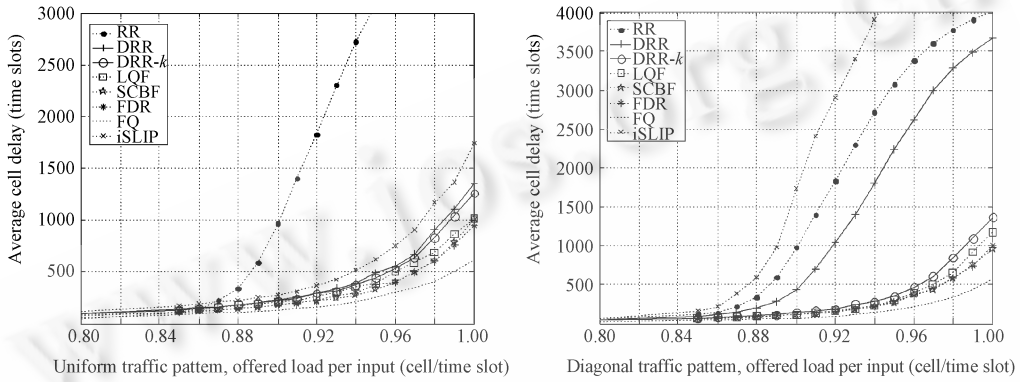


Fig.7 Delay performance of various schemes under ON-OFF burst uniform and diagonal traffic

图 7 突发业务源下各种算法的时延性能仿真结果

### 3.2 交叉点大小对系统性能的影响

图 8 给出了不同 XB 队列容量下 CICQ 交换系统的时延性能仿真结果,采用的是 Bernoulli Uniform 业务源分布模型.可以看出,XB 队列容量越大,系统的时延性能越好.

### 3.3 吞吐量仿真

采用 Hotspot 业务源分布模型来验证 FDR 调度策略的吞吐量性能.对于 Hotspot 业务源模型,用 $w$ 表示不均衡指数(unbalanced coefficient),$w\in[0,1]$,则 Hotspot 分布描述为:若 $j=i$, ,否则有

.

图 9 给出了 Bernoulli Hotspot 业务源分布模型下各种算法的吞吐量性能仿真结果.可以看出,FDR 和 DRR,LQF 以及 DRR-$k$ 都表现出良好的性能,其吞吐量接近 100%,优于 $i$SLIP,RR,DRR-$k$ 和 MCBF 算法.

### 3.4 抗突发性仿真

从全网的分组交换质量来看,总希望交换节点的调度策略对输出业务具有平滑作用,这样就可以减小到达下游节点业务流的突发性,因而调度策略的抗突发性能对全网的 QoS 保障具有重要意义.图 10 给出了在 ON-OFF 突发业务源 Uniform 和 Diagonal 分布下 FDR 算法的抗突发性能仿真结果,可以看出,FDR 调度策略的抗突发性能接近 RR 算法,因而能够对输出业务产生较好的平滑效果.

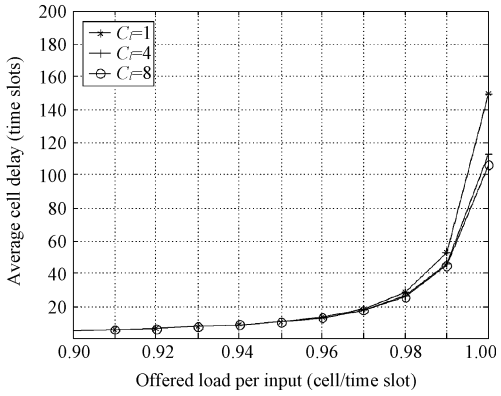


Fig.8 Delay performance under different XB size

图 8 不同 XB 大小下交换系统的时延性能

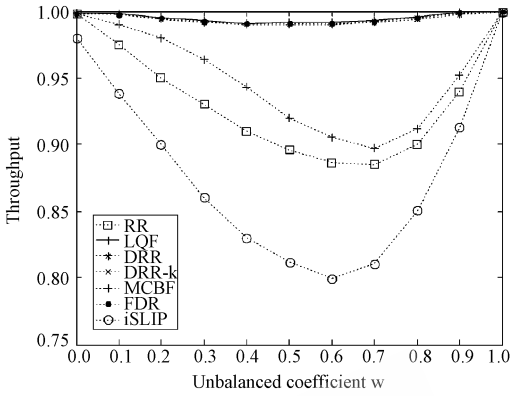


Fig.9 Throughput performance under Hotspot traffic

图 9 Hotspot 业务源下各种算法的吞吐量性能

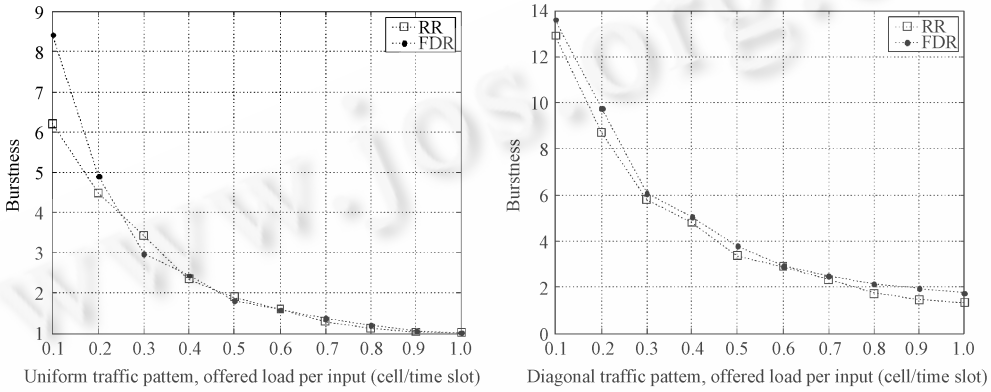


Fig.10 Anti-Burst performance of RR-RR and FDR algorithm under ON-OFF burst uniform and diagonal traffic

图 10 突发业务源下 RR 和 FDR 算法的抗突发性能

## 4 结束语

通过在每个交叉点集成一定容量的缓存单元,从而将 Crossbar 交换单元输入与输出端口之间的带宽冲突隔离开来,分布式与并行调度策略的实施成为了现实.我们分析了目前 CICQ 交换结构调度策略在可扩展性或性能方面存在的不足,给出了其调度策略设计的基本准则;以此准则为基础,提出了基于虚拟通道状态动态分配调度份额的 FDR 调度策略.以该成果为基础,如何构建支持组播交换的 CICQ 交换结构是下一步值得研究的课题.

**References**:

[1] McKeown N. Scheduling algorithms for input-queued cell switches [Ph.D. Thesis]. Berkeley: Dept. Elect. Eng. Comput. Sci., University of California, 1995.

[2] Texas Instruments. GS400.15-μm CMOS, Standard Cell/Gate Array. 2000. http://www.ti.com/

[3] Nabeshima M. Performance evaluation of a combined input- and crosspoint-queued switch. IEICE Trans. on Commun., 2000, E83-B(3):737−741.

[4] Javidi T, Magill R, Hrabik T. A high-throughput scheduling algorithm for a buffered crossbar switch fabric. In: Neuvo Y, ed. Proc. of the IEEE Int'l Conf. on Communications (ICC). Helsinki: IEEE Communications Society, 2001. 1586−1591.

[5] Magill B, Rohrs C, Stevenson R. Output-Queued switch emulation by fabrics with limited memory. IEEE Journal on Selected Areas in Communications, 2003,21(4):606−615.

[6] Iyer S, Chuang ST, McKeown N. Practical algorithms for performance guarantees in buffered crossbars. In: Znati T, ed. Proc. of the IEEE INFOCOM 2005. Miami: IEEE Communications Society, 2005. 981−991.

[7] Hu HC, Yi P, Guo YF. Design and implementation of high performance simulation platform for switching and scheduling. Journal of Software, 2008,19(4):1036−1050 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/19/1036.hml

[8] McKeown N, Mekkittikul A. Starvation free algorithm for achieving 100% throughput in an input queued switch. In: Lee D, ed. Proc. of the ICCCN'96. Rockville: IEEE Communications Society, 1996. 226−229.

[9] Mhamdi L, Hamdi M. MCBF: A high-performance scheduling algorithm for buffered crossbar switches. IEEE Communications Letters, 2003,7(9):451−453.

[10] Zhang X, Bhuyan LN. An efficient algorithm for combined input-crosspoint-queued (CICQ) switches. In: Shah R, ed. Proc. of the IEEE Globecom 2004. Dallas: IEEE Communications Society, 2004. 1168−1173.

[11] Rojas-Cessa R, Oki E, Jing Z, Chao JH. On the combined input-crosspoint buffered switch with round-robin arbitration. IEEE Trans. on Commun., 2005,53(11):1945−1951.

[12] Luo JZ, Lee Y, Wu J. DRR: A fast high-throughput scheduling algorithm for combined input-crosspoint-queued (CICQ) switches. In: George R, ed. Proc. of the IEEE MASCOTS 2005. Atlanta: IEEE Computer Society, 2005. 329−332.

[13] Zheng YF, Shao C. An efficient round-robin algorithm for combined input-crosspoint-queued switches. In: Dini P, ed. Proc. of the IEEE ICAS/ICNS 2005. Papeete: IEEE Computer Society, 2005. 23−28.

**附中文参考文献:**

[7] 扈红超,伊鹏,郭云飞.高性能交换与调度仿真平台的设计与实现.软件学报,2008,19(4):1036−1050. http://www.jos.org.cn/1000-9825/19/1036.hml

**扈红超**(1982－),男,河南商丘人,博士生,主要研究领域为高性能路由,交换技术.

**郭云飞**(1963－),男,教授,博士生导师,主要研究领域为高性能交换技术,下一代网络.

**伊鹏**(1977－),男,博士,讲师,主要研究领域为高性能交换,调度算法.

**李玉峰**(1976－),男,博士生,主要研究领域为宽带信息网络,高速路由器核心技术.