ISSN 1000-9825, CODEN RUXUEW
Journal of Software, Vol.20, No.8, August 2009, pp.2289−2297
doi: 10.3724/SP.J.1001.2009.03327

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

一种交错编码的多重门限调度算法*

伊 鹏[1+], 汪斌强[1], 陈庶樵[1], 李 挥[2]

[1](国家数字交换系统工程技术研究中心,河南 郑州 450002)

[2](北京大学 深圳研究生院 集成微系统重点实验室,广东 深圳 518055)

Interleaving Coded Multi-Threshold Scheduling Algorithm

YI Peng[1+], WANG Bin-Qiang[1], CHEN Shu-Qiao[1], LI Hui[2]

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

[2](Key Laboratory of Integrated Microsystems, Shenzhen Graduate School, Peking University, Shenzhen 518055, China)

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

Yi P, Wang BQ, Chen SQ, Li H. Interleaving coded multi-threshold scheduling algorithm. *Journal of Software*, 2009,20(8):2289−2297. http://www.jos.org.cn/1000-9825/3327.htm

Abstract: An interleaving coded multi-threshold scheduling (ICMTS) algorithm is proposed in this paper. Since the ICMTS algorithm uses the interleaving coded thresholds of two stage queues as the scheduling weights, it can systematically evaluate the scheduling demands of both the input queues and the crosspoint queues. By segmenting the queue length as multiple thresholds, the hardware resource of this algorithm can be largely decreased. It is proved that a CICQ (combined input-crosspoint-queued) switch operating with the ICMTS algorithm can achieve 100% throughput with a speedup of two. To facilitate hardware implementation, a simplified maximal ICMTS scheme is also presented with a time complexity of $O(\log N)$. Simulation results show that even the simplified ICMTS scheme can obtain better performance than the existing algorithms.

Key words: switch architecture; scheduling algorithm; combined input-crosspoint-queued; buffered crossbar

摘 要: 提出一种交错编码的多重门限调度算法(interleaving coded multi-threshold scheduling,简称 ICMTS).该算法将前、后级队列门限标记交错编码作为权值表征输入调度过程前、后两级队列的整体调度需求,根据交错编码的权值对前级虚拟输出队列进行优化调度判决,并通过多重门限机制降低算法的硬件资源开销.采用流模型证明当加速因子为 2 时,ICMTS 算法可获得 100%的吞吐量,并给出 ICMTS 算法的工程简化设计方案,复杂度为 $O(\log N)$.仿真仿真结果表明,采用 ICMTS 算法的工程简化方案即可获得比现有算法更优的调度性能.

关键词: 交换结构;调度算法;联合输入交叉节点排队;带缓存交叉开关

中图法分类号: TP393 **文献标识码**: A

调度算法用于决定交换结构在拥塞时优先服务的报文,因此对于提高路由交换设备的性能十分关键.近年

* Supported by the National High-Tech Research and Development Plan of China under Grant Nos.2007AA01Z218, 2008AA01Z214 (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 2007-10-20; Revised 2008-02-01; Accepted 2008-03-14

来,联合输入交叉节点排队(combined input-crosspoint-queued,简称 CICQ)交换结构成为研究热点,基于 CICQ 交换结构提出多种调度算法.虽然这些调度算法与传统输入排队(input queued,简称 IQ)交换结构的调度算法相比在综合性能方面获得了一定的提升,但是其算法设计均未能充分结合 CICQ 交换结构的特点,忽略了输入调度过程与前、后两级队列同时紧密相关的特性,仅根据单级队列的拥塞状态进行判决,因此不利于优化调度性能.

为此,本文提出一种交错编码的多重门限调度算法(interleaving coded multi-threshold scheduling,简称 ICMTS),该算法采用多重门限机制衡量 CICQ 交换结构前、后两级队列的拥塞程度,通过将前、后级队列门限标记交错编码作为权值表征输入调度过程各虚拟输出队列的调度需求.多重门限机制有利于降低算法的硬件资源开销,两级队列门限标记交错编码作为权值使得 ICMTS 算法兼顾了前、后两级队列的整体调度需求,不仅能够有效地控制输入排队长度的增长,提高调度算法的稳定性,而且能够充分利用交换结构的输出带宽,优化分组交换调度处理的平均时延.流模型分析表明,当加速因子为 2 时,ICMTS 算法对于所有满足强大数定律(strong law of large number,简称 SLLN)的可容许到达业务均可以获得 100%的吞吐量.为了便于硬件实现,我们给出了 ICMTS 算法的工程简化设计方案,其复杂度仅为 $O(\log N)$.本文采用典型的到达业务模型对 ICMTS 算法和其他多种算法进行仿真比较.仿真结果表明,采用 ICMTS 算法的简化设计方案即可获得比现有算法更优的性能.

本文第 1 节对 CICQ 交换结构进行描述,并对基于该结构的典型算法进行回顾和评述.第 2 节提出 ICMTS 调度算法,采用流模型证明当加速因子为 2 时,ICMTS 算法可获得 100%的吞吐量,并给出 ICMTS 算法的工程简化设计方案.第 3 节对 ICMTS 算法和其他典型算法进行仿真比较.第 4 节是结论.

1 背景及相关研究

1.1 CICQ交换结构

交叉开关由于具有无阻塞特性,实现简单,并且有成熟的商用芯片可直接应用,因而在目前商用路由器交换结构设计中广为应用[1,2].然而随着网络技术的发展,路由交换设备必须能够支持更高的端口速率和更密集的端口数量,基于交叉开关构建的交换结构因此面临着越来越多的挑战.输出排队(output queued,简称 OQ)交换结构虽然在提供服务质量(QoS)保障方面极具优势,通过简单的调度机制即可获得高吞吐量和良好的时延性能,然而交换单元和存储单元的 N 倍加速问题使其在高速环境下应用受限[3].输入排队(IQ)交换结构虽然无须硬件加速,并且通过采用虚拟输出排队(virtual output queued,简称 VOQ)机制也可以获得高吞吐量[4],但是其调度过程必须从全局的角度协调交换结构所有输入端口和输出端口的带宽使用,因而算法复杂度较高,在高速环境下不易于硬件实现[5].比较而言,联合输入输出排队(CIOQ)交换结构对输出排队机制和输入排队机制进行了较好的折衷,Chuang 等人证明了该交换结构在 2 倍加速时能够完全模拟 OQ 交换结构[6,7],然而 CIOQ 交换结构模拟 OQ 交换结构同样需要集中式的匹配算法,依然具有极高的算法复杂度,因此仅具有理论意义.

一种新的思路是,在交叉开关内部的交叉节点增设小容量的缓存单元,而且近年来随着芯片设计技术的进步和工艺水平的提高,在交叉开关内部实现小容量的节点缓存并不困难[8].目前,带缓存交叉开关已经成为交换结构领域新的研究热点,基于带缓存交叉开关构建的联合输入交叉节点排队(CICQ)交换结构因其性能优势更是备受关注[9].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 served output queued,简称 FCFS-OQ)[10];Chuang 等人证明了在 3 倍加速条件下,CICQ 交换结构可以通过分布式调度算法实现模拟采用任意调度算法的 OQ 交换结构[11];我们证明了当空分复用扩展因子为 2 时,CICQ 交换结构无须加速即可模拟采用任意调度算法的 OQ 交换结构[12].CICQ 交换结构模型如图 1 所示,它采用带缓存交叉开关作为核心交换单元,在交换结构的每个输入端口分别设置一个输入缓存单元.为了避免发生队头阻塞,输入缓存单元通常采用虚拟输出排队机制,即每个输入缓存单元在逻辑上被划分为 N 个虚拟输出队列,分别用于缓存到达 N 个输出端口的分组.

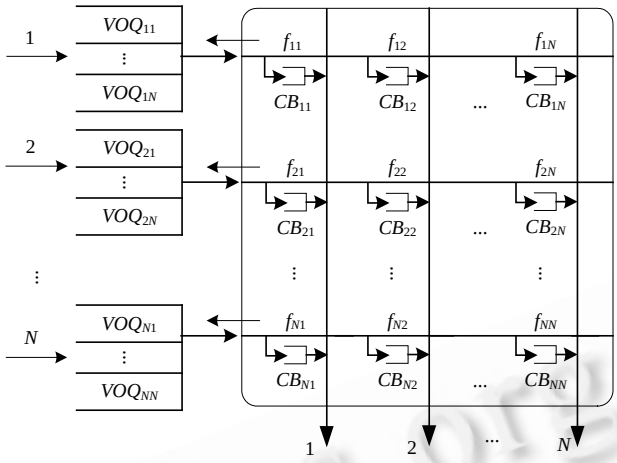


Fig.1 An $N\times N$ CICQ switch model

图 1 $N\times N$ 的 CICQ 交换结构模型

CICQ 交换结构在结构形态上与 IQ 交换结构十分相似,与 IQ 交换结构相比,CICQ 交换结构由于交换单元具有缓存能力,能够隔离交换结构输入端口与输出端口的带宽资源冲突,从而使得交换单元的每个输入端口和输出端口可以相对独立地使用内部连接带宽资源,避免了集中控制式的调度判决,因此,其调度机制与 IQ 交换结构相比将大为简化.下文用 VOQ_{ij} 标识输入缓存单元源端口为 i、目的端口为 j 的虚拟输出队列,用 CB_{ij} 标识与 VOQ_{ij} 和输出端口 j 相关联的节点缓存.为了防止节点缓存溢出导致分组丢失,对每个 CB_{ij} 维护一个流控状态信号 f_{ij} 用于对 VOQ_{ij} 进行流量控制,当 CB_{ij} 被写满时,f_{ij}=1,禁止 VOQ_{ij} 中的分组输出;当 CB_{ij} 未被写满时,f_{ij}=0,允许 VOQ_{ij} 中的分组输出.

1.2 典型调度算法分析

CICQ 交换结构的分组调度被分割为输入调度过程和交叉节点调度过程两个部分,其中:输入调度过程根据交叉节点缓存的流控状态信号对缓存于虚拟输出队列中的分组进行调度,选择合适的分组送往对应节点缓存;交叉节点调度过程负责调度核心交换单元内部具有相同输出端口的一组节点缓存队列,并从中选取合适的分组从交换结构的输出端口输出.由于 CICQ 交换结构输入调度过程不仅决定交换结构输入端口的带宽分配,而且还关系到输出端口的带宽使用效率,目前普遍认为输入调度算法的设计对于提升系统的交换性能尤为关键.因此,本文主要研究 CICQ 交换结构输入调度算法的设计.

基于 CICQ 交换结构已经提出多种调度算法,其中典型的算法有 RR-RR[13],OCF-OCF[14],LQF-RR[15],SCBF[16],MCBF[17]等.为了便于分析,我们将带缓存交叉开关内部具有同一输出端口 j 的所有节点缓存抽象为一个逻辑队列 LQ_j,并按照虚拟输出队列和逻辑队列的连接关系对 CICQ 交换结构的缓存队列进行重新组合,则 CICQ 交换结构可抽象成如图 2 所示的一个两级排队系统.RR-RR 算法在输入调度过程采用简单轮询机制,复杂度为 $O(1)$,十分便于硬件实现,但缺点是性能较差.OCF-OCF 算法和 LQF-RR 算法在输入调度过程均根据第 1 级虚拟输出队列的拥塞程度进行调度判决,前者以分组的等待时间作为调度权值选取所有虚拟输出队列首部具有最长等待时间的分组获得服务,后者以虚拟输出队列的排队长度作为调度权值选取具有最大排队长度的虚拟输出队列获得调度.两种算法的复杂度均为 $O(\log N)$.但就实现层面而言,两种算法均需要采用复杂度为 $O(\log B)$的二输入比较器作为基本硬件单元,其中 $B=\log L_{\max}$($L_{\max}$ 表示比较器输入参数的最大值).由于虚拟输出队列具有较大的缓存容量,两种算法的比较参数值都很大,因此比较仲裁电路的实现复杂度较高.为了降低实现复杂度,SCBF 算法和 MCBF 算法在输入调度过程选择以带缓存交叉开关内部缓存资源的使用情况作为比较对象,以逻辑队列的排队长度作为调度权值优先选取对应逻辑队列最为空闲的虚拟输出队列获得服务.虽然它们的算法复杂度也是 $O(\log N)$,但由于带缓存交叉开关内部缓存容量远小于虚拟输出队列的缓存容量,其比较仲裁

电路的输入参数值远小于 OCF-OCF 算法和 LQF-RR 算法,因此可以明显降低调度算法的硬件实现复杂度.

虽然这些调度算法与传统 IQ 交换结构的调度算法相比在综合性能方面获得了一定的提升,但是其算法设计均未能充分结合 CICQ 交换结构的特点,忽略了输入调度过程与前后两级队列同时紧密相关的特性.OCF-OCF 算法和 LQF-RR 算法仅依据第 1 级虚拟输出队列的状态进行调度判决,忽略了输入调度过程第 2 级逻辑队列接收分组的需求.采用这类调度算法的 CICQ 交换结构即使与空逻辑队列相关联的虚拟输出队列也不能优先获得服务,因此会导致交换结构输出带宽资源的浪费.SCBF 算法和 MCBF 算法仅依据第 2 级逻辑队列的状态进行调度判决,忽略了输入调度过程第 1 级虚拟输出队列发送分组的需求,采用这类调度算法的 CICQ 交换结构即使十分拥塞的虚拟输出队列也不能优先被调度输出分组,因此虚拟输出队列的排队长度即便在可容许的输入流量下也会迅速增长,不仅增加了交换结构的输入缓存资源需求,而且降低了调度算法的稳定性.因此不难看出,现有算法仅依据单级队列状态进行判决不利于优化 CICQ 交换结构的调度性能.

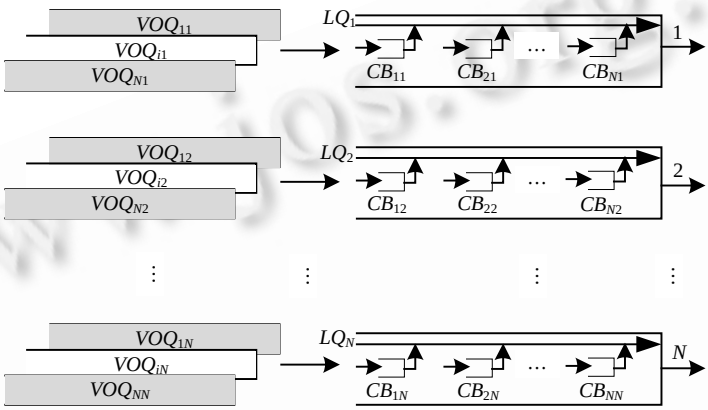


Fig.2 Two stage queued system model for CICQ switches

图 2 CICQ 交换结构的两级排队系统抽象模型

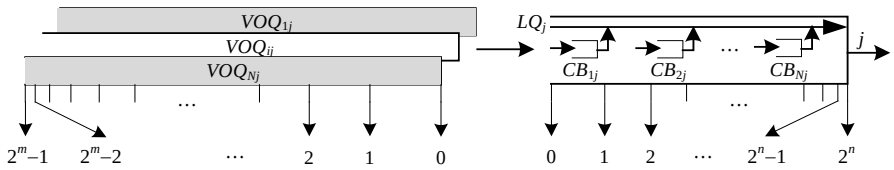
2 ICMTS调度算法

针对现有算法存在的不足,本节结合 CICQ 交换结构的特点,从前、后两级队列整体调度需求的层面提出一种交错编码的多重门限调度算法 ICMTS.通过流模型证明当加速因子为 2 时,该算法可获得 100%的吞吐量,并给出了 ICMTS 算法的工程简化设计方案.为了便于分析和表述,文中假定交换结构各端口到达业务均为定长包,标记为 *cell*,以线路速率传输一个 *cell* 所需的时间称为一个时隙.

2.1 算法描述

ICMTS 算法依据虚拟输出队列的拥塞程度确定前级队列的发送分组需求,依据逻辑队列的空闲程度确定后级队列接收分组的需求,根据交换结构前后两级队列整体调度需求进行调度判决,其调度权值的确定机制如图 3 所示.为了降低算法设计中比较仲裁电路的硬件复杂度,ICMTS 算法在界定队列拥塞程度和空闲程度时采用了多重门限机制,虚拟输出队列的门限标识用 $b_{m-1}b_{m-2}\ldots b_1b_0$ 表示,逻辑队列的门限标识用 $a_na_{n-1}\ldots a_1a_0$ 表示,其中 $m=kn$,因为通常虚拟输出队列的缓存容量要大于逻辑队列.通过对两级排队系统模型的分析发现:前级队列排队长度变化对其输出分组需求的影响会随队列拥塞程度的增加而增强;后级队列排队长度变化对其接收分组需求的影响会随队列空闲程度的增加而增强.ICMTS 算法在门限划分时随着排队长度的增加,虚拟输出队列采用先稀疏后密集的划分机制,逻辑队列采用先密集后稀疏的划分机制,采用这种划分机制使队列门限标识可以更加灵敏地表征队列发送和接收分组的需求.为了获得高吞吐量,ICMTS 算法在确定 CICQ 交换结构整体调度需求时将逻辑队列门限标识的最高位作为调度权值最高位,从而保证空逻辑队列对应的虚拟输出队列具有更高的调度优先级.调度权值的其他各比特由虚拟输出队列门限标识和逻辑队列门限标识的其他比特交错

编码构成,标记为 $a_nb_{m-1}\ldots b_{m-k}a_{n-1}b_{m-k-1}\ldots b_{m-2k}a_{n-2}b_{m-2k-1}\ldots b_ka_1b_{k-1}\ldots b_0a_0$.



VOQ threshold marker: $b_{m-1}b_{m-2}\ldots b_1b_0$ LQ threshold marker: $a_na_{n-1}\ldots a_1a_0$

Scheduling weight: $a_nb_{m-1}\ldots b_{m-k}a_{n-1}b_{m-k-1}\ldots b_{m-2k}a_{n-2}b_{m-2k-1}\ldots b_ka_1b_{k-1}\ldots b_0a_0$

Fig.3 Scheduling weight arbitrate mechanism for ICMTS algorithm

图 3 ICMTS 算法调度权值确定机制

下文将用 W_{ij} 标识虚拟输出队列 VOQ_{ij} 的调度权值,下面给出 ICMTS 调度算法的具体描述:

1) 将所有满足 f_{ij}=0 且 W_{ij}>0 的虚拟输出队列标 VOQ_{ij} 识为合格队列.

2) 在合格队列中选择所有 W_{ij} 最高位为 1 的 VOQ_{ij},使其与尽可能多的输出端口建立匹配并输出分组.

3) 未获得匹配的输入端口选择具有最大 W_{ij} 值的合格队列获得调度输出分组.

4) 更新流控状态信号 f_{ij}.

ICMTS 算法优先将所有合格的虚拟输出队列与尽可能多的空逻辑队列进行匹配,使交换结构的出口带宽资源得以充分利用,为获得高吞吐量提供了保证.未匹配的输入端口以调度权值作为判决依据,使逻辑队列非空时拥塞的虚拟输出队列优先获得调度,保证了算法的稳定性.如何保证 W_{ij} 最高位为 1 的 VOQ_{ij} 与尽可能多的输出端口建立匹配关系是 ICMTS 算法的关键问题,下面对该匹配机制进行详细说明.将所有输入端口的集合 X 和所有空逻辑队列的集合 Y 看作二分图的顶点集中两个互不相交的子集,如果 W_{ij} 最高位为 1,则说明输入端口 i 和空逻辑队列 LQ_j 之间有一条边连接.令 x_i 表示输入端口 i 具有的边连接数,令 y_j 表示逻辑队列 LQ_j 建立匹配的个数,匹配机制首先将所有 y_j 值初始化为 0,并按照 x_i 值和 y_j 值递增的顺序分别对合格的输入端口和逻辑队列进行排序.对于每一输入端口 i,在逻辑队列序列中选择第 1 个与输入端口 i 具有边连接关系的逻辑队列建立匹配,然后更新 y_j 值并重新对逻辑队列按照 y_j 值递增的顺序进行排序.与 IQ 交换结构的最大尺度匹配(maximum size match,简称 MSM)机制相比,ICMTS 算法的匹配机制并不要求建立一对一的匹配关系,因此其复杂度与 MSM 机制相比大为降低.图 4 给出了 ICMTS 算法匹配过程的伪码.由图可知,匹配过程的复杂度为 $O(N\log N)$,而选择具有最大调度权值队列的复杂度为 $O(\log N)$.因此,ICMTS 算法的复杂度为 $O(N\log N)$.

Initialization: $y_j \leftarrow 0$ (j=1,2,…,N)
1) sort inputs in the increasing order of x_i
2) **for** each sorted $input_i$ with an edge **do**
3) find the first j in the sorted LQ list such that there is an edge between $input_i$ and LQ_j
4) $y_j=y_j+1$
5) resort LQ in the increasing order of y_j

Fig.4 Match pseudocode for ICMTS algorithm

图 4 ICMTS 算法匹配过程伪码

2.2 吞吐量分析

流模型分析方法对于分析和验证交换结构和调度算法的吞吐量十分有效,因此近年来许多文献均采用该方法进行交换结构和调度算法的吞吐量分析[15,16,18].流模型分析方法的基本思路是,将交换结构及其到达过程和离去过程抽象为一个流量模型,并对系统中离散的到达离去事件按照统计方法求取流极限,如果对于特定业务,流量模型中离去过程和到达过程的流极限值相等则说明交换结构可获得 100%的吞吐量.参考文献[18]中的定理及其证明,本节采用流模型分析 ICMTS 算法的吞吐量性能.

令 $A_{ij}(n)$表示到第 n 时隙所有到达输入端口 i,目的端口为 j 的 $cell$ 个数,$A_{ij}(0)$=0.我们假定到达过程{$A_{ij}(\cdot)$,

$i,j=1,\ldots,N\}$以概率 1 满足强大数定律,即

$$\lim_{n\to\infty}\frac{A_{ij}(n)}{n}=\lambda_{ij},\ i,j=1,...,N \tag{1}$$

如果λ_{ij}满足公式(2),则到达业务$\{A_{ij}(\cdot),i,j=1,\ldots,N\}$被称为可容许的.

$$\sum_i\lambda_{ij}\le 1 \text{ and } \sum_j\lambda_{ij}\le 1 \tag{2}$$

令 $D_{ij}(n)$表示到第 n 时隙所有离开交换结构输出端口 j 的 *cell* 个数,$D_{ij}(0)=0$.根据文献[18]的定理 1 可知,如果 $D_{ij}(n)$以概率 1 满足公式(3),则交换结构称为速率稳定的(rate stable),可以获得 100%的吞吐量.

$$\lim_{n\to\infty}\frac{D_{ij}(n)}{n}=\lambda_{ij},\ i,j=1,...,N \tag{3}$$

令 $Z_{ij}(t)$表示 t 时刻缓存于 VOQ_{ij} 和 CB_{ij} 中流的总量,$Z(t)$是以 $Z_{ij}(t)$为元素构成的流量矩阵.根据文献[18]中的定理 3 和引理 1 可知,若要证明交换结构是速率稳定的,则仅需证明对于任意 $Z(t)>0$,$Z'(t)\le 0$,其中 $Z'(t)$为 $Z(t)$的导数.直观上来说,也就是证明当交换结构处于拥塞时,可容许的到达业务缓存队列中流的总量是非递增的.

定理 1. 当加速因子为 2 时,ICMTS 算法对于任意满足强大数定律的可容许业务均可获得 100%的吞吐量.

证明:令 $B_j(n)$表示在第 n 时隙缓存在 LQ_j 的 *cell* 总数,$B_j(n)=0$ 说明逻辑队列为空.对于任意一个时隙 n,如果 $Z_{ij}(n)>0$,则存在以下两种情况:(1) $B_j(n)>0$;(2) $B_j(n)=0$.

对于第 1 种情况,任意工作保持(work-conserving)交叉节点调度算法均可以将一个 *cell* 从输出端口 j 输出.令 $M_j(t)=\sum_i Z_{ij}(t)$ 表示 t 时刻所有目的端口为 j 的流的总量,则

$$M_j(n+1)-M_j(n)\le\sum_i(A_{ij}(n+1)-A_{ij}(n))-1 \tag{4}$$

结合公式(3),运用流模型的求极限运算可得:

$$M'_j(t)=\sum_i Z'_{ij}(t)\le\sum_i\lambda_{ij}-1\le 0 \tag{5}$$

对于第 2 种情况,ICMTS 算法会尽可能多地匹配空逻辑队列,并根据匹配关系将输入端缓存中的 *cell* 送往节点缓存.如果 LQ_j 获得匹配,则得到与第 1 种情况相同的结果.如果 LQ_j 未获得匹配,则存在另一空逻辑队列 $LQ_{j'}$ 满足 $Z_{ij'}(n)>0,j'\ne j$ 与输入端口 i 匹配.可以推断,$LQ_{j'}$ 唯一匹配于输入端口 i,否则调度机可以重新将输入端口 i 匹配于 LQ_j 以匹配更多的空逻辑队列.因此,输入端口 i 必有一个 *cell* 被输出,令 $L_i(t)=\sum_j Z_{ij}(t)$ 表示 t 时刻所有输入端口为 i 的流的总量,则

$$L_i(n+1)-L_i(n)\le\sum_j(A_{ij}(n+1)-A_{ij}(n))-1 \tag{6}$$

结合公式(3),运用流模型的求极限运算可得:

$$L'_i(t)=\sum_j Z'_{ij}(t)\le\sum_j\lambda_{ij}-1\le 0 \tag{7}$$

定义 $C_{ij}(t)=L_i(t)+M_j(t)$,公式(5)和公式(7)式中任何一个成立均可推出

$$C'_{ij}(t)=L'_i(t)+M'_j(t)\le\sum_j\lambda_{ij}+\sum_i\lambda_{ij}-1 \tag{8}$$

因此,对于加速因子为 2 的 ICMTS 算法可得:

$$C'_{ij}(t)\le\sum_j\lambda_{ij}+\sum_i\lambda_{ij}-2\le 0 \tag{9}$$

根据文献[18]定理 2 的证明,由 $C'_{ij}(t)\le 0$ 可推得 $Z'(t)\le 0$.因此,当加速因子为 2 时,ICMTS 算法可以保证 CICQ 交换结构是速率稳定的,可以获得 100%的吞吐量. □

2.3 简化方案

虽然已经证明 ICMTS 调度算法可以获得良好的吞吐量性能,但是其匹配过程仍需采用集中控制机制,各个输入端口的调度判决存在较强的相关性,因此复杂度偏高,在高速环境下不易于硬件实现.就调度算法的复杂度和可扩展性层面而言,最好是各个输入端口能够彼此独立地作出调度判决.为此,本文给出了 ICMTS 算法的工程简化设计方案,简称为 SICMTS(simplified interleaving coded multi-threshold scheduling)方案.SICMTS 方案不要求建立空逻辑队列和输入端口之间的最大匹配,它采用全分布式调度机制,在各个输入端口并行选取具有最

大调度权值的虚拟输出队列获得服务.具体机制如下:

1) 将所有满足 f_{ij}=0 且 W_{ij}>0 的虚拟输出队列 VOQ_{ij} 标识为合格队列.

2) 在合格队列中选择具有最大 W_{ij} 值的合格队列获得调度输出分组.

3) 更新流控状态信号 f_{ij}.

各个端口独立判决可能会导致某些空逻辑队列获得多个匹配而其他空逻辑队列却未获得匹配,这在一定程度上会影响到部分输出端口的带宽资源利用效率.然而,由于调度权值以其最高位单独用于标识空逻辑队列,因此空逻辑队列在调度判决过程中具有绝对优先权.这样,即使在所有输入端口完全同步地选择同一空逻辑队列的最坏情况下,最多需要 N 个时隙即可让所有输出端口的带宽资源获得充分利用.由于 SICMTS 方案仅需在各个输入端口并行选择具有最大调度权值队列获得服务,在采用典型的树型比较电路实现时,其算法复杂度为 $O(\log N)$,硬件复杂度为 $O(\log N\cdot\log m+n+1)$.与现有算法相比,SICMTS 方案采用多门限机制降低了算法的硬件复杂度,而且调度权值采用的交错编码机制全面考虑了前、后两级队列的整体调度需求,不仅能够充分利用交换结构的输出带宽,而且可以有效地控制系统前级队列排队长度的增加,因此可以期望获得更好的综合性能.

3 仿真分析

本节通过仿真实验手段从稳定性和有效性两个方面对 SICMTS 方案进行性能评测.其中,稳定性通过采用与文献[16]类似的仿真测试方法来衡量,有效性通过算法在不同业务分布条件下的时延特性来验证.为了便于分析比较,本节同时还对 OCF-OCF,LQF-RR,SCBF-OCF 和 MCBF 等多种典型调度算法进行了仿真.仿真平台采用系统级设计方法和面向对象技术编程实现.所有仿真实验采用 16×16 的交换结构,仿真业务选取 ON-OFF 突发业务源.业务源经过交换后的输出端口分布分别采用 uniform 分布($\lambda_{ij}=\lambda_i/16$)和 diagonal 分布($\lambda_{ii}=2\lambda_i/3$, $\lambda_{i|i+1|}=\lambda_i/3$),其中,突发业务的突发长度为 20.仿真实验通过改变突发过程的平均空闲长度来调整负载ρ.仿真器工作的时间粒度为一个时隙,仿真时间为 200 000 个时隙.

图 5 给出了上述典型算法在 ON-OFF 突发业务源 uniform 分布($\lambda_{ij}=\lambda_i/N$)条件下与 SICMTS 方案的时延特性比较.图 6 给出了上述典型算法在 ON-OFF 突发业务源 diagonal 分布$\lambda_{ii}=2\lambda_i/3$,$\lambda_{i|i+1|}=\lambda_i/3$)条件下与 SICMTS 方案的时延特性比较.由图可见,当业务负载较重时,SICMTS 方案可以获得优于其他算法的时延性能.

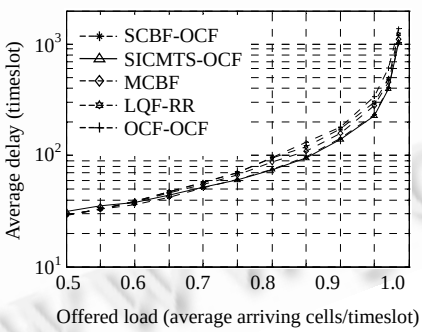


Fig.5 Average delay for SICMTS scheme under uniform traffic

图 5 SICMTS 方案在 uniform 分布条件下的平均时延

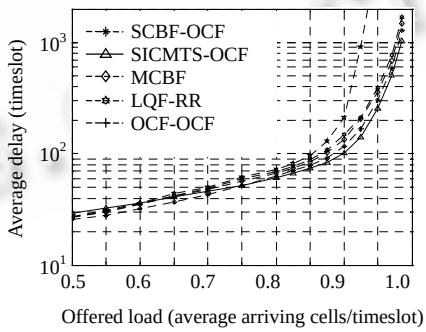


Fig.6 Average delay for SICMTS scheme under non-uniform traffic

图 6 SICMTS 方案在 diagonal 分布条件下的平均时延

图 7 给出了典型算法与 SICMTS 方案的稳定性比较.根据参考文献[16]可知,稳定性可以通过 uniform 分布($\lambda_{ij}=\lambda_i/N$)条件下向量$\|L(n)\|$值的大小来衡量,$\|L(n)\|$值越小,说明对应调度算法的稳定性越高.其中,$\|L(n)\|$的定义如下:

$$\|L(n)\|=\sqrt{VOQ_{1,1}(n)^2+\ldots+VOQ_{1,N}(n)^2+\ldots+VOQ_{N,1}(n)^2+\ldots+VOQ_{N,N}(n)^2} \tag{10}$$

||$L(n)$||体现了所有虚拟输出队列在时隙 n 的整体占用情况.由图 7 的仿真结果可知,SICMTS 方案的||$L(n)$||值比其他算法要小,因此具有更好的稳定性.

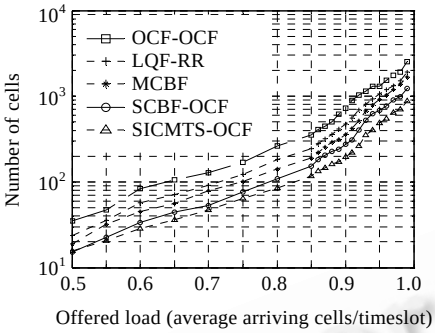


Fig.7 Stability simulation for SICMTS scheme under uniform traffic

图 7 SICMTS 方案在 uniform 分布条件下的稳定性仿真

4 结束语

针对 CICQ 交换结构现有调度算法仅根据单级队列的拥塞状态进行判决,忽略了输入调度过程与前、后两级队列同时紧密相关的特性,本文提出一种兼顾前、后两级队列整体调度需求的交错编码多重门限调度算法(ICMTS).流模型分析表明,当加速因子为 2 时,ICMTS 算法对于所有满足强大数定律(SLLN)的可容许到达业务均可以获得 100%的吞吐量.为了便于硬件实现,文中给出了 ICMTS 算法的工程简化设计方案 SICMTS,其复杂度仅为 $O(\log N)$,并且比现有方案需要更少的硬件资源.仿真结果表明,SICMTS 方案具有良好的稳定性与有效性,整体性能优于现有算法,对于带缓存交叉开关交换结构的调度算法设计具有参考意义.

致谢 在此,我们向对本文的工作给予支持和建议的同行,尤其是国家数字交换技术研究中心的汪斌强教授、兰巨龙教授和张兴民教授领导的研发团队中的同学和老师表示感谢.

References:

[1] Cisco Systems, Inc. Cisco 12000 series Internet routers. http://www.cisco.com

[2] Partridge C, Carvey PP, Burgess E, Castineyra I, Clarke T, Graham L, Hathaway M, Herman P, King A, Kohalmi S, Ma T, Mcallen J, Mendez T, Milliken WC, Pettyjohn R, Rokosz J, Seeger J, Sollins M, Storch S, Tober B, Troxel GD, Waitzman D, Winterble S. A 50-Gb/s IP router. IEEE/ACM Trans. on Networking, 1998,6(3):237−248.

[3] Kesidis G, McKeown N. Output-Buffer ATM packet switching for integrated-services communication networks. In: Proc. of the IEEE ICC'97. Montreal: IEEE, 1997. 1684−1688.

[4] Anderson TE, Owicki SS, Saxe JB, Thacker CP. High-Speed switch scheduling for local-area networks. ACM Trans. on Computer Systems, 1993,11(4):319−352.

[5] Chang CS, Chen WJ, Huang HY. On service guarantees for input buffered crossbar switches: A capacity decomposition approach by Birkhoff and von Neumann. In: Proc. of the IEEE IWQoS'99. London: IEEE, 1999. 79−86.

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

[7] Stoica I, Zhang H. Exact emulation of an output queueing switch by a combined input output queueing switch. In: Proc. of the 6th IEEE/IFIP IWQoS'98. Napa: IEEE, 1998. 218−224.

[8] Singhal V, Le R. High-Speed buffered crossbar switch design using Virtex-EM devices. 2000. http://www.xilinx.com/xapp/xapp240.pdf

[9] Yi P, Wang BQ, Guo YF. Providing QoS guarantees in buffered crossbars with space-division multiplexing expansion. In: Proc. of the IEEE GLOBECOM 2006. San Francisco: IEEE, 2006. 1−6.

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

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

[12] Yi P, Wang BQ, Guo YF, Li H. Providing QoS guarantees in a novel switch architecture. Acta Electronica Sinica, 2007,35(7): 28−35 (in Chinese with English abstract).

[13] Yoshigoe K, Christensen K, Jacob A. The RR/RR CICQ switch: Hardware design for 10-Gbps link data rate. In: Proc. of the IEEE Int'l Performance, Computing, and Communications Conf. IEEE, 2003. 481−485.

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

[15] Javidi T, Magill R, Hrabik T. A high-throughput scheduling algorithm for a buffered crossbar switch fabric. In: Proc. of the IEEE ICC 2001. IEEE, 2001. 1581−1587.

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

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

[18] Dai JGJ, Prabhakar B. The throughput of data switches with and without speedup. In: Proc. of the IEEE INFOCOM 2000, Vol.3. IEEE, 2000. 556−564.

附中文参考文献:

[12] 伊鹏,汪斌强,郭云飞,李挥.一种可提供 QoS 保障的新型交换结构.电子学报,2007,35(7):28−35.

伊鹏(1977－),男,湖北黄冈人,博士,讲师,主要研究领域为路由交换技术.

陈庶樵(1973－),男,博士,副教授,主要研究领域为网络安全,路由算法.

汪斌强(1963－),男,博士,教授,博士生导师,主要研究领域为宽带信息网,高速路由器核心技术.

李挥(1964－),男,博士,副教授,主要研究领域为交换技术,集成电路设计技术.