ISSN 1000-9825, CODEN RUXUEW
Journal of Software, Vol.19, No.12, December 2008, pp.3207−3217
DOI: 10.3724/SP.J.1001.2008.03207

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

一种维序的基于组合输入输出排队的并行交换结构*

戴 艺+, 苏金树, 孙志刚

(国防科学技术大学 计算机学院,湖南 长沙 410073)

A Parallel Packet Switch Achieving In-Order Cell Delivery with Combined-Input-and-Output Queuing Switches

DAI Yi+, SU Jin-Shu, SUN Zhi-Gang

(School of Computer, National University of Defense Technology, Changsha 410073, China)

+ Corresponding author: E-mail: y_dai@163.com

Dai Y, Su JS, Sun ZG. A parallel packet switch achieving in-order cell delivery with combined-input-and-output queuing switches. *Journal of Software*, 2008,19(12):3207−3217. http://www.jos.org.cn/1000-9825/19/3207.htm

Abstract: An in-order queuing (IOQ) PPS architecture proposed in this paper uses a small fixed-size buffer in the demultiplexor to distribute traffic equally among switch planes, with central combined input-and-output queuing (CIOQ) switch planes under the control of a single scheduler that applies the same matching at each of the parallel switch planes during each cell slot. This operation is called synchronous scheduling. It is proved that the round robin demultiplexing algorithm along with synchronous scheduling guarantees cells of a flow can be read in order from the output queues of the switch planes. Furthermore, by using a synchronous scheduling called strict longest queue first (SLQF) algorithm this scheme reduces considerably not only the amount of state information required by the scheduler, but the communication overhead required to achieve cell reordering. Compared with existing PPS designs, IOQ PPS (in-order queuing parallel packet switch) is more practical to implement in hardware because of its simple implementation mechanisms, as the experimental results demonstrate, and it offers the best delay performance.

Key words: switch architecture; IOQ PPS (in-order queuing parallel packet switch); CIOQ switch; parallel packet switch; in-order cell delivery

摘 要: 提出一种按序排队(in-order queuing,简称 IOQ)PPS 体系结构,通过在分流控制器引入固定尺寸的缓冲区,实现负载在每个交换平面的均匀分配;中间层组合输入输出排队(combined input-and-output queuing,简称 CIOQ)交换平面受控于中央调度器,在每个时间槽(time slot),中央调度器将同一种匹配实施到每一个交换平面,称之为同步调度策略.可以证明,在该体系结构下,轮询(round robin)分派算法配合同步调度策略可以保证同一条流的信元按序从交换平面读出.进一步提出了严格最长队列优先同步调度算法,极大地减少了中央调度器需要维护的状态信息和信元重定序开销.与目前主流的 PPS 设计相比,IOQ PPS(in-order queuing parallel packet switch)实现机制简单,

* Supported by the National Natural Science Foundation of China under Grant No.90604006 (国家自然科学基金); the National Basic Research Program of China under Grant No.2003CB314802 (国家重点基础研究发展计划(973))

Received 2007-02-13; Accepted 2007-09-19

易于硬件实现.模拟结果表明,IOQ PPS 具有最优的延迟性能.

关键词: 交换结构; IOQ PPS(in-order queuing parallel packet switch);组合输入输出排队交叉开关;并行报文交换;信元按序发送

中图法分类号: TP393 文献标识码: A

有统计显示,商业 DRAM 访存速度每 18 个月速度增长 10%,而链路速率每 7 个月就增长了 100%[1].随着链路速率从 OC-192(10Gb/s)到 OC-768(40Gb/s)甚至达到了 OC-3072(160Gb/s),在电交换结构(electronic switch)中,缓冲报文已经变得很困难甚至是不可能的[2].例如,当链路速率为 160Gb/s 时,需要缓冲存储器在不到 1ns 的时间里完成对 40 字节信元的写入和读出操作,而目前 DRAMs 的随机访问时间为 50ns.交换结构所需要的大容量缓冲区用 DRAM 实现,只有容量非常小的缓冲区才使用高速 SRAM 实现(目前还不支持大容量片上 SRAM)[3].DRAM 技术的发展速度远远落后于摩尔定律[4].一些新的存储技术,例如 RAMBUS[5],SDRAMs 以及 DDRAMs 具有很快的 I/O 时间,但这些技术并不能降低存储器的随机访问时间.输出排队(OQ)交换结构的内部互连带宽和缓冲存储器带宽必须 N 倍于输入端口链路速率(若输入端口链路速率不同,则为端口速率之和).如果不使用特殊的存储器件作为输出缓冲区,OQ 交换结构甚至不能应用于 G 比特网络[6].在输入排队(IQ)/组合输入输出排队(CIOQ)交换结构中,输入端口到输出端口的匹配是通过交换矩阵调度算法实现的,算法复杂度至少是 $O(N^2)$,随着端口规模的扩大,IQ/CIOQ 交换结构需要更多的时间来计算匹配以调度所有的交换端口.例如,链路速率为 10-Gpbs,信元长度为 64 字节时,必须在 51ns 以内调度全部的端口.对于较大的端口数目 N(N>16),调度算法的性能难以达到 10-Gbps 以上链路速率的要求.尽管通过采用基于帧的调度[7]、流水化调度[8]、确定型调度[9]能够有效提高 IQ/CIOQ 交换结构调度算法的性能,但由于单个芯片所能封装的引脚数目有限,将单级 IQ/CIOQ 交换结构扩展到 T 比特以上交换容量几乎是不可能的.

通常采用多个低速交叉开关来构建 T 比特交换网络,并行报文交换 PPS[2,3,10−14]在过去 6 年里一直被认为是降低交换系统存储带宽需求,提高交换速率及交换容量的有力手段.PPS 包括 N 个分流/重组控制器和 K 个中间层低速交换模块,这些交换模块并行工作,各自独立地交换报文.PPS 设计存在的主要问题是:如何以较低的通信开销保持每条流报文的顺序,使得 PPS 系统易于实现.目前,PPS 信元重组算法需要分流控制器、中间层交换平面以及重组控制器三者之间的相互协作来保证每条流的顺序,它们两两之间通信的复杂性以及大量状态信息的维护使得 PPS 中的关键算法难以硬件实现.

1 相关工作

PPS 的交换思想最先由文献[15]提出来,交换系统包括 N 个分流/重组控制器和一个 $NK\times NK$ 的核心交换开关.分流控制器采用基于虚拟连接(VC)的 SCIMA 算法分派流量,为了满足每条流的 QoS 需求,SCIMA 算法在分流控制器端将单个报文流分离成多个子流并动态调度这些子流以避免内部链路的拥塞.SCIMA 算法只能为有限数目的流指定子连接,其余的流则被分派到单独的内部链路.流的分离和子流在内部链路的负载均衡将造成报文乱序.因此提出一种报文重组协议,即使在子连接之间存在巨大延迟差异的情况下,报文重组协议仍能够以正确的顺序重组每条流的报文并能恢复每条子流单个报文的丢失[15].后继研究者进一步提出并评估了多种报文重组协议,主要包括全序序列、SCIMA+AFR 和基于流数量等级这 3 种协议,它们不但可以正确地重组每条流的报文,还能够解决每个子连接多个报文的丢失问题[16].这些 PPS 设计[15,16]最初应用于 ATM 交换机,由于 SCIMA 算法在连接建立阶段需获知连接的速度,因此需要维护大量的状态信息(例如内部链路的剩余带宽,分离 VC 的数量,核心交换平面的反馈信息等等),这使得 SCIMA 算法难以应用于 G 比特网络.文献[17]提出一种考虑了报文流级的流量分布的 PPS,分流器端的分派算法假设流的速率就是该流第一个报文的到达速率,并进一步假设流的速率一直不会改变,这些假设显然不适用于随网络拥塞状况而改变速率的 TCP/IP 流.

斯坦福大学 Iyer 等人提出一种集中式 PPS(centralized PPS),并证明当加速比 $S\geq 2$ 时,采用 FCFS-OQ 交换平面,分流/重组控制器不设缓冲区的集中式 PPS 能够仿真 FCFS-OQ 交换结构[2].特别地,当加速比 $S\geq 3$ 时,集中式

PPS 能够提供 QoS 保证[2].集中式 PPS 采用集中式信元分派算法,分流控制器在每个仲裁周期和中央仲裁器通信,中央仲裁器为每一个到达 PPS 系统的信元选择交换平面,这需要每个分流控制器和中央仲裁器之间的控制通路运行在线速.集中式调度方法通信复杂度为 $o(N\log N+2N\log K+NK)$,硬件实现复杂[3];不能在各交换平面之间公平分配流量,存储资源使用低效.为了保持报文的顺序,集中式 PPS 需要交换平面实现反馈机制,因而无法用现有的交叉开关构建交换平面[2].为了解决以上问题,Iyer 通过在分流/重组控制器引入小型的固定尺寸的缓冲区提出一种分布式 PPS(distributed PPS),每个分流控制器独立执行信元分派算法,降低了通信复杂度[2],之后的 PPS 设计均采用分布式信元分派算法.分布式 PPS 不能仿真 FCFS-OQ 交换结构,因而不能提供 QoS 保证[18].此外,重组控制器存在“死锁”现象(没有任何信元可以读出而不违反报文流的顺序)[18]难以实现信元按序发送.文献[3]提出一种 VIQ PPS,通过在重组控制器端引入固定尺寸的 VIQ(virtual input queues)队列实现了信元按序发送.VIQ PPS 的缺陷在于信元可能被重组控制器延迟发送,这也是 VIQ PPS 难以控制信元延迟提供 QoS 保证的原因[3].本文提出一种 IOQ PPS(in-order queuing parallel packet switch),分流控制器引入固定尺寸的缓冲区,实现了负载在每个交换平面的均匀分配;采用 CIOQ 交换平面,进一步降低了存储器带宽需求;利用 CIOQ 交换平面队列特性,实现了信元按序发送并消除了分流控制器和交换平面之间以及交换平面和重组控制器之间的通信开销.

2 IOQ PPS 体系结构

IOQ PPS 体系结构如图 1 所示,分流控制器通过引入运行在线速的固定尺寸的缓冲区,采用轮询的方式将同一条流的信元均匀分派到 K 个 CIOQ 交换平面,中间层 CIOQ 交换平面在中央控制器的同步调度下将信元交换到相应的输出队列,最后由不带缓冲区的重组控制器按照每条流的顺序将信元从输出队列输出到正确的输出端口.每个端口通过内部链路和 K 个交换平面相连,内部链路速率为 $S(R/K)$,其中 S 为内部链路加速比,R 为外部端口速率.

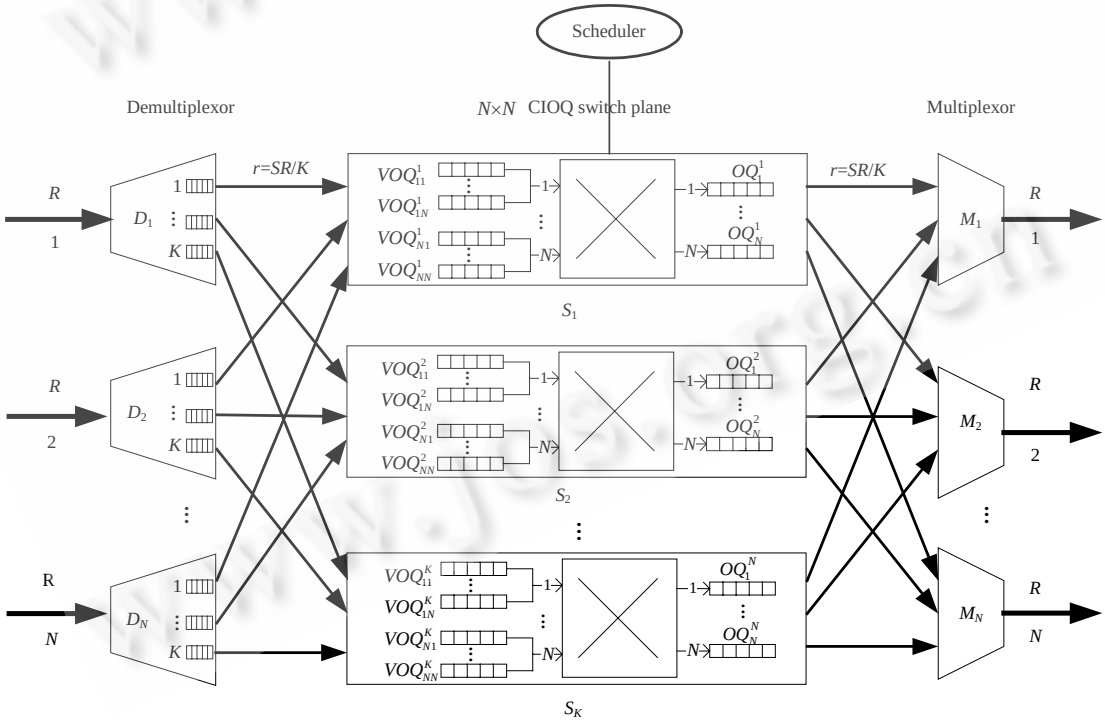


Fig.1 Architecture of an IOQ PPS based on CIOQ switches

图 1 基于 CIOQ 交叉开关的 IOQ PPS 体系结构

定义 1. 在速率为 R 的链路发送或接收一个信元所耗费的时间为外部时间槽(external time slot).

定义 2. 在速率为 SR/K 的链路发送或接收一个信元所耗费的时间为时间槽(time slot).

定义 3. 在每个时间槽,调度算法仅计算一种匹配 M,并将 M 实施到 K 个交换平面,这就是同步调度策略.

PPS 体系结构受限于两种约束:输出链路约束和输入链路约束[2],这两种约束定义如下:

定义 4. 外部输入端口每$\lceil K/S \rceil$个外部时间槽至多往同一交换平面发送一个信元,这是因为内部链路速率要比外部输入端口慢 K/S 倍.我们称这种约束为输入链路约束(input link constraint).

定义 5. 每个交换平面每$\lceil K/S \rceil$个外部时间槽至多发送一个信元到同一输出端口,这是因为内部链路速率要比外部输出端口慢 K/S 倍.我们称这种约束为输出链路约束(output link constraint).

3 IOQ PPS 实现方案

3.1 信元分派算法

IOQ PPS 分流控制器体系结构如图 2 所示,缓冲区由 K 个深度为 N 个信元的 FIFO(first in first out)队列构成,对应于 K 个交换平面.分流控制器 i 为 N 个输出端口保持 N 个独立的轮询指针 $P_1,\ldots,P_N$,指针的取值范围为 $\{1,\ldots,K\}$.如果指针 $P_j=l$,则表明下一个目的端口为 j 的信元将会被分派到交换平面 l,图 2 显示了连续到达的 6 个信元在分流控制器缓冲区的分布情况,C_m^j 表示第 m 个外部时间槽到达、目的端口为 j 的信元.当最后一个目的端口为 1 的信元 C_6^1 进入缓冲队列时,信元 C_1^1,C_3^2,C_2^2 已经依次到达中间层交换平面,如图 2 灰色部分所示.分流控制器为每条流维护独立的轮询指针,实现每条流在交换平面的均匀分布,称为轮询分派算法.没有引入缓冲区的分流控制器在执行轮询分派算法时可能会违反输入链路约束.分流控制器缓冲区的引入不仅满足输入链路约束的需要,更重要的是消除了系统加速比.

我们用 E 表示以外部时间槽为度量单位的时间,t 表示时间.如果外部链路速率为 R,信元长度为 P 字节,那么每个信元需要花费时间 P/R 到达,$t=EP/R$.

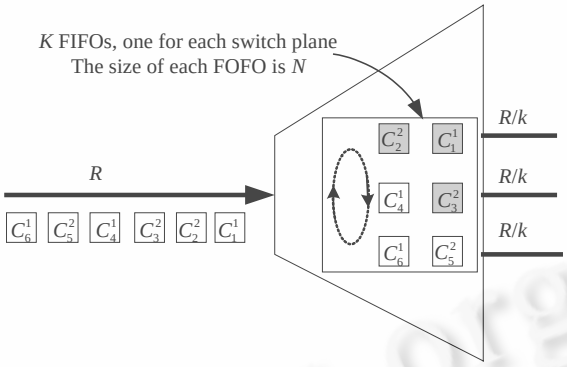


Fig.2 Architecture of a demultiplexer (K=3, N=2, S=1)

图 2 分流控制器体系结构(K=3,N=2,S=1)

引理 1. 在 E 个外部时间槽内,分流控制器 i FIFO 队列 $Q(i,l)$中的信元个数 $D(i,l,E)$满足:当 $E\le N$ 时,$D(i,l,E)\le E$;当 $E>N$ 时,$D(i,l,E)<\frac{E}{K}+N$.

证明:既然分流控制器以轮询方式分派每条流的信元,那么分流控制器每接收 K 个到达同一输出端口的信元,恰能向每个交换平面发送一个信元.令 $S(i,E)=\sum_{j=1}^{N}\overline{S}(i,j,E)$,其中 $\overline{S}(i,j,E)$ 表示分流控制器 i 在任意 E 个外部时间槽内接收到目的端口为 j 的信元数目,$S(i,E)$表示在时间间隔 E 内分流控制器 i 接收的全部信元数目,显然 $S(i,E)\le E$.当 $E>N$ 时,有

$$D(i,l,E)\le\sum_{j=1}^{N}\left\lceil\frac{\bar{S}(i,j,E)}{K}\right\rceil\le\left\lceil\sum_{j=1}^{N}\frac{\bar{S}(i,j,E)}{K}\right\rceil+N-1=\left\lceil\frac{S(i,E)}{K}\right\rceil+N-1\le\left\lceil\frac{E}{K}\right\rceil+N-1<\frac{E}{K}+N,$$

当 $E\le N$ 时,同理可得 $D(i,l,E)\le E$.

定理 1. 在没有内部链路加速比的情况下,缓冲区大小为 NK 个信元的分流控制器能够执行轮询分派算法而不发生缓冲区溢出现象.

证明:由 $t=EP/R$,重写引理 1 为 $D(i,l,t)\le Rt/PK+N$,表示在时间 t 内,写入到分流控制器每个 FIFO 队列的信元数目小于等于 $Rt/PK+N$.由此可将每个 FIFO 队列表示为漏桶源(leaky bucket source),平均速率为 $\rho=R/PK$,桶深为 $\sigma=N$ 个信元的漏桶模型.每个 FIFO 的发送速率为 $\mu=R/PK$,由漏桶源的定义可知[17],桶深为 N 的 FIFO 缓冲区不会发生溢出.

每个分流控制器维护一个运行于线速为 R,大小为 NK 个信元,组织成 K 个 FIFO 队列的缓冲区.以端口数目 $N=1024$,信元长度为 64 字节,交换平面数目 $K=10$ 为例,缓冲区大小为 5Mb,采用目前的 SRAM 技术可将该缓冲区集成在芯片上.另一方面,由于 IOQ PPS 采用的是 CIOQ 交换平面,不受"N 倍加速比"限制,以目前的 SRAM 技术,在缓冲存储器的读写速率和数据宽度方面也完全能够达到要求.因此,带缓冲的 IOQ PPS 结构是可行的. □

3.2 交换平面调度算法

在 IOQ PPS 体系结构里,中央调度器执行同步调度策略,集中控制 K 个交换平面调度信元到输出队列.在本节,我们主要证明使用轮询分派算法和同步调度策略的 IOQ PPS 具有按序排队特性,从而为不带缓冲的重组控制器按每条流的顺序从输出队列读取信元提供了可能.第 1.4 节介绍了重组控制器操作的细节.

我们用 $f(i,j)$表示从输入端口 i 到输出端口 j 的流,$C(i,j)$表示从输入端口 i 到输出端口 j 的信元,VOQ_{ij}^{l} 表示交换平面 l 的 VOQ_{ij} 队列,OQ_j^l 表示交换平面 l 的输出队列 OQ_j.

定义 6. 对于任意输出端口 j,在每个时间槽结束后,只要交换平面存在目的端口为 j 的信元,那么一定存在一条流的 d 个最老信元(oldest cell)以循环(round robin)方式位于 OQ_j 队列头的位置,其中 $1\le d\le K$,那么称 IOQ PPS 具有按序排队特性.

引理 2. 如果使用轮询分派算法和同步调度策略,那么对于任何流 $f(i,j)$,在时间槽 T 结束后,存在 $1\le l\le K$,VOQ_{ij}^{l} 是时间槽 T 结束后最后一个接收到信元 $C(i,j)$的 VOQ_{ij} 队列,当 $l<z\le K$ 时,VOQ_{ij}^{z} 队列长度为 L,当 $1\le z\le l$ 时,VOQ_{ij}^{z} 队列长度为 $L+1$,信元按序以循环方式分布在 VOQ_{ij}^{z} 队列.特别地,当 $l=K$ 时,所有 $VOQ_{ij}s$ 队列长度相同.

证明:采用对时间槽的归纳法证明.

对第一个时间槽,引理 2 显然成立.

假设在时间槽 T 结束后,引理 2 成立,那么只需证明在时间槽 $T+1$ 结束后,引理 2 依然成立.考虑以下两种情况:

情况 1. 在时间槽 T 结束后,所有交换平面的 $VOQ_{ij}s$ 队列非空,假设 VOQ_{ij}^{a} 是时间槽 T 结束后最后一个接收到信元 $C(i,j)$的 VOQ_{ij} 队列,当 $a\le z\le K$ 时,VOQ_{ij}^{z} 队列长度为 L,当 $1\le z\le a$ 时,VOQ_{ij}^{z} 队列长度为 $L+1$,信元按序以循环方式分布在 VOQ_{ij}^{z} 队列.在时间槽 $T+1$ 内,同步调度算法将同一种匹配实施到每个交换平面后,所有 $VOQ_{ij}s$ 队列的长度要么全部保持不变,要么全部减 1.不失一般性,假设在时间槽 $T+1$ 内有 d 个信元从 M_i 发送到各交换平面,显然,$1\le d\le K$.按照轮询分派算法,这 d 个信元按照流 $f(i,j)$的顺序以循环方式依次缓冲到队列 $VOQ_{ij}^{(a+1)\bmod K}$...$VOQ_{ij}^{(a+d)\bmod K}$.因此,存在 $1\le l=(a+d)\bmod K\le K$,使得引理 2 在时间槽 $T+1$ 结束后依然成立.

情况 2. 在时间槽 T 结束后,至少有一个 VOQ_{ij} 队列为空.由引理 2 可知,其他非空 VOQ_{ij} 队列长度至多为 1.在时间槽 $T+1$ 内,同步调度算法将同一种匹配实施到每个交换平面后,所有 $VOQ_{ij}s$ 队列的长度要么全部保持不变,要么全部变成 0.同上可证引理 2 在时间槽 $T+1$ 结束后,依然成立. □

引理 3. 如果使用轮询分派算法和同步调度策略,那么在时间槽 T 结束后,对于任何流 $f(i,j)$,要么所有信元分

布在不同的交换平面,要么 K 个最老信元分布在不同的交换平面.

证明:假设在时间槽 T 结束后,存在 VOQ_{ij} 为空,那么由引理 2 可知,对于所有的交换平面 l,VOQ_{ij}^{l} 队列长度至多为 1,因此流 $f(i,j)$所有的信元分布在不同的交换平面.

使用反证法证明 K 个最老信元分布在不同的交换平面的情况.假设在时间槽 T 结束后,所有 VOQ_{ij} 队列非空,且 K 个最老信元不是分布在不同的交换平面,那么一定存在某个 VOQ_{ij} 队列 VOQ_{ij}^{l} 包含 K 个最老信元中的 2 个信元 C_1 和 C_2,令 C_1 在时间槽 T_1 到达,C_2 在时间槽 T_2 到达,按照轮询分派算法,从 T_1 到 T_2 的时间段里恰有流 $f(i,j)$的 K–1 个信元到达输入端口 i,故 T_2 到达的 C_2 不在 K 个最老信元之列,这与假设相矛盾. □

定理 2. 使用轮询分派算法和同步调度策略,可以满足按序排队特性.

证明:采用对时间槽的归纳法证明.

对第 1 个时间槽,定理 2 显然成立.

假设在时间槽 T 结束后,定理 2 成立,那么只需证明在时间槽 T+1 结束后,定理 2 依然成立.由引理 3 得知,对于任何流 $f(i,j)$,在一个时间槽结束后,要么所有信元分布在不同的交换平面,要么 K 个最老信元分布在不同的交换平面.那么在时间槽 T+1,同步调度算法将另一条流 $f(b,j)$的 d'个最老信元交换到不同的 $OQ_j s$ 队列,$1\le d'\le K$.由引理 2 得知,这 d'个信元按序以循环方式分布在 OQ_j^{l} 队列中.在时间槽 T 结束后,由定理 2 可知,只要输出队列 $OQ_j s$ 不全为空,就存在流 $f(a,j)$的 d 个最老信元循环分布在 OQ_j^{l} 队列头中,其中 $1\le d\le K$.在时间槽 T+1 内,M_j 将流 $f(a,j)$的 d 个最老信元发送到输出端口 j.在时间槽 T+1 结束后,流 $f(b,j)$的 d'个最老信元循环分布在 OQ_j^{l} 队列头中.定理 2 在时间槽 T+1 结束后,依然成立. □

3.3 重组控制机制

第 3.2 节已经证明使用轮询分派算法和同步调度策略,可以满足按序排队的特性.在这一节我们详细介绍分流控制器和中央调度器之间以及中央调度器和重组控制器之间怎样相互协同实现信元按序发送.重组控制机制描述如下:

1. 只要流 $f(i,j)$缓存在队列 $VOQ_{ij}s$ 中的信元数目变成 0,分流控制器 D_i 将轮询指针 P_j 重置为 1.该策略可以保证每条流的最老信元总是被分派到第一个交换平面,简化了信元重组过程.
2. 中央调度器为每个输出端口 j 保持一个 FIFO 列表 L_j,用于记录每次执行匹配 M 时交换到输出端口 j 的信元数目(由引理 3 可知,每次调度交换的信元数目要么是 K 个,要么是全部的信元).因此,每当执行到输出端口 j 的匹配时,中央调度器将参数 p 添加到 FIFO 列表 L_j 的尾部.参数 p 就是非空 VOQ 队列的数目.若 $p\ne K$,则将索引 j 通告给 D_i,轮询指针 P_j 被重置为 1,其中$(i,j)\in M$ 表示当前匹配选择调度 VOQ_{ij} 队列.
3. 既然每条流的最老信元在第一个交换平面,那么重组控制器 M_j 只需在每个时间槽从中央调度器的 FIFO 列表 L_j 检索参数 p,然后从输出队列 OQ_j^{1} 开始依次读出 p 个输出队列头信元.中央调度器和重组控制器之间的通信复杂度为 $O(N\log K)$.

3.4 进一步降低通信开销

提出一种严格最长队列优先 SLQF(strict longest qsueue first)同步调度算法,进一步降低了系统通信开销.中央调度器只根据第一个交换平面的队列状态执行 SLQF 算法,在每个时间槽将计算得到的匹配 M 实施到 K 个交换平面.SLQF 算法与单个交叉开关执行的 LQF 算法[18]类似,也是一种循环调度算法,与 LQF 算法不同的是,每次循环,SLQF 算法只发送队列长度大于等于为 2 的 VOQ 队列的调度请求.

对于第一个交换平面的任意 VOQ_{ij} 队列,$(i,j)\in M$,表示 VOQ_{ij} 队列长度至少为 2,因此一定有流 $f(i,j)$的 K 个信元交换到输出端口 j.在每个时间槽,重组控制器 M_j 按 OQ_j^{1} 到 OQ_j^{K} 的顺序依次读出 K 个信元即可保证信元按序发送.但 SLQF 算法存在“饿死”现象,例如,某个队列只有一个信元并且再也没有信元到达,而其他所有队列的长度开始为 2 且每个信元时间都有信元到达,这样第一个队列就永远得不到服务.为了调度那些在长时间内得不

到服务的队列长度为 1 的 VOQ_{ij}^1 队列,SLQF 同步调度算法采用了一种超时机制:

如果 VOQ_{ij}^1 队列长度为 1 且等待时间超过了预设阈值 τ,则 SLQF 算法在每次循环将 VOQ_{ij}^1 队列权重设置为最高并为其发送调度请求.

(1) 在时间槽 T 结束后,若 $(i,j)\in M$ 且 $p\neq K$,中央调度器通告分流调度器 D_i 将轮询指针 P_j 重置为 1,并将参数对 (p,h) 记录在 FIFO 列表 C_j 中,其中 h 为时间槽 T 结束后 OQ_j^1 队列信元个数(在时间槽 T+1 结束后,本次调度的 p 个信元缓冲到 OQ 队列完毕).每过一个时间槽,所有的参数 h 自减 1,在 h=0 的下一个时间槽即时间槽 $T+h$ 开始时,中央调度器向重组控制器 M_j 发送通信请求信号.

(2) 当重组控制器 M_j 从 OQ_j^1 队列读出每条流的第 1 个信元时,检查中央调度器是否有通信请求.若通信请求信号有效,M_j 接收参数 p,然后依次从 OQ_j^1 到 OQ_j^p 读出 p 个头信元;否则依次从 OQ_j^1 到 OQ_j^K 读出 K 个头信元.

SLQF 同步调度算法进一步降低了 IOQ PPS 的通信开销,只有在“饿死”现象发生时才启动一次通信,而不是在每个时间槽.

4 性能模拟与分析

我们使用斯坦福大学开发的 SIM 模拟器[19]进行模拟实验.SIM 模拟器是为模拟单个交叉开关而设计的,我们对 SIM 模拟器进行修改构成 PPS 交换环境.开发的模拟模型包括 OQ,iSLIP IQ[20],集中式 PPS[2],分布式 PPS[2],VIQ PPS[3],IOQ PPS.OQ 和 iSLIP 作为起参照作用的单级交换结构具有与 PPS 系统相同的聚合交换容量.除本文以外,只有文献[3]通过模拟对目前主流的 PPS 系统性能进行比较,文献[3]中的实验结果表明,VIQ PPS 在使用分布式调度的 PPS 系统中具有最优延迟性能,而本文的模拟结果显示,IOQ PPS 延迟性能优于 VIQ PPS.

4.1 流量模型及实验参数

SIM 模拟器支持多种流量源,本文采用了两种具有代表性的流量模型:1) 贝努利一致流,服从贝努利到达过程,独立同分布,目的端口均匀分布于所有的输出端口;2) 突发流,突发长度为 10,在忙-闲周期(busy-idle periods)突发信元,目的端口以连续突发或者一个信元接一个信元的方式分布于所有的输出端口.由于 Internet 流量具有突发特性,突发流量模型更接近于真实的网络流量.

在所有的模拟实验中,控制变量为负载,响应变量为系统平均延迟(包含队列延迟和传输延迟).负载范围为 50%~95%,以 5%的幅度递增;模拟时间为 200 000 个时间片(slots),一个时间片等于一个外部时间槽;实验参数为 N=8,K=4,S=1.实验类型包括:1) 延迟实验,在突发流量模型和贝努利一致流量模型下模拟了 OQ,iSLIP IQ[20],集中式 PPS[2],分布式 PPS[2],VIQ PPS[3],IOQ PPS 的系统平均延迟.2) 加速比实验,将 IOQ PPS 内部链路加速比分别设置为 S=1,S=2,S=3 时,在贝努利流量模型下模拟 IOQ PPS 系统平均延迟,该实验评估内部链路加速比对 IOQ PPS 延迟性能的影响.3) 交换平面数目实验,在贝努利流量模型下模拟了当交换平面的数目分别为 K=4,K=6,K=8,K=10 时,IOQ PPS 系统平均延迟.分两种情况来进行该实验($r=S(R/K)$):第 1 种情况,内部链路加速比保持不变,内部链路速率 r 随交换平面数目的改变而改变;第 2 种情况,随着交换平面数目的改变,通过改变内部链路加速比 S 保持内部链路速率 r 恒定不变.该实验全面评估了交换平面数目对 IOQ PPS 延迟性能的影响.

4.2 建立模拟模型

在 OQ 和 iSLIP IQ 交换模型中,输入/输出端口的缓冲区大小分别设置成无穷.iSLIP 调度算法的循环次数为 4.以下各种 PPS 模型交换平面缓冲区大小均设置成无穷.对于集中式 PPS,理论上内部链路速率加速比 $S\geq 2$ 时,集中式 PPS 能够仿真 FCFS-OQ 交换结构[2],但文献[2]并没有给出具体的可描述的集中式信元分派算法可用于建模.因此,我们将集中式 PPS 建模为具备和 FCFS-OQ 相同的交换性能,仅存在内部传输延迟的差别.分布式 PPS 建模为分流控制器采用轮询分派算法;中间层 OQ 交换平面为每个到达的信元实行延迟均等操作[2],延迟均等操作延迟交换信元直至信元所经历的延迟到达最大值(NK 个外部时间槽);中间层 OQ 交换平面及重组控制器根据信元到达时间戳服务信元;分流/重组控制器分别设有固定尺寸为 NK 个信元的缓冲区.分布式 PPS 重组

控制器端存在“死锁”现象[18],当“死锁”发生时,重组控制器排空 K 个缓冲区的头信元,解除“死锁”现象,但这将造成流的乱序和信元的丢失并增加重组算法的复杂度.VIQ PPS 建模为分流控制器采用轮询分派算法并设有固定尺寸为 NK 个信元的缓冲区;中间层交换平面采用 OQ 交叉开关;重组控制器为每一个输入端口维护一个轮询指针指示下一个读取的 VIQ 队列,若相应的 VIQ 队列为空,则等待直到该队列变成非空,重组控制器端设有 K 个缓冲块对应 K 个交换平面,每个缓冲块设有 N 个 VIQ 队列对应 N 个输入端口,重组控制器为每一个输入端口维护一个轮询指针指示下一个读取的 VIQ 队列,为了保证信元的顺序,当相应的 VIQ 队列为空时,则等待直到该队列变成非空,重组控制器以循环方式选择轮询指针.IOQ PPS 建模为分流控制器采用轮询分派算法并设有大小为 NK 个信元的缓冲区;中央调度器采用 SLQF 同步调度算法;如果没有特殊声明,中间层交换平面采用不带加速比的 CIOQ 交叉开关;重组控制器不带缓冲区.为了保证 OQ 交叉开关的正常运行,OQ 交换平面加速比均为 N.

4.3 模拟结果分析

图 3 显示了在贝努利一致流量模型下各种 PPS 系统的性能,最大模拟负载为 98%.集中式 PPS 具有最低延迟性能.这是因为理论上集中式 PPS 可以仿真 FCFS-OQ 交换结构,因而具有与 OQ 交换结构相当的延迟性能(仅存在报文传输延迟的差异).但这种仿真假设集中式信元分派算法能够实时获取所有交换平面的全局信息,并根据信元的历史调度信息在每个仲裁周期处理所有输入端口的调度请求,庞大的通信开销使得集中式调度算法无法实现.分布式 PPS 具有最差延迟性能,这是因为大部分的信元延迟都发生在其执行的“延迟均等”操作上.图 3 显示了当负载达到 93%以上时,集中式 PPS 具有比 iSLIP 更好的延迟性能,这说明在高负载的情况下 CIOQ 交换结构呈现出了不稳定的特性.VIQ PPS 是目前具有最佳延迟性能的 PPS 系统(集中式 PPS 只具备理论上的最低延迟).图 3 显示了 IOQ PPS 在 80%以下负载的情况时具有比 VIQ PPS 更低的平均系统延迟;但在 80%以上负载时,IOQ PPS 的延迟性能低于 VIQ PPS.这是因为 VIQ PPS 使用的是比 CIOQ 更为稳定的 OQ 作为中间层交换平面,但 OQ 交换平面运行速率必须是内部链路速率的 N 倍,因此在本实验中,OQ 交换平面加速比为 8,而 CIOQ 交换平面没有任何加速比.图 3 同样显示了 IOQ PPS 只需使用加速比为 2 的 CIOQ 作为交换平面,就能在高负载的情况下保持稳定,并表现出比 VIQ PPS 更优的延迟性能.图 4 显示了在突发流量模型下各种 PPS 系统的性能.当负载达到 85%以上时,iSLIP 和没有加速比的 IOQ PPS 已经变得不稳定;当负载达到 90%以上时,所有交换结构都表现出不稳定性.可见真实网络流量的突发特性对交换结构的性能有着极大的影响.

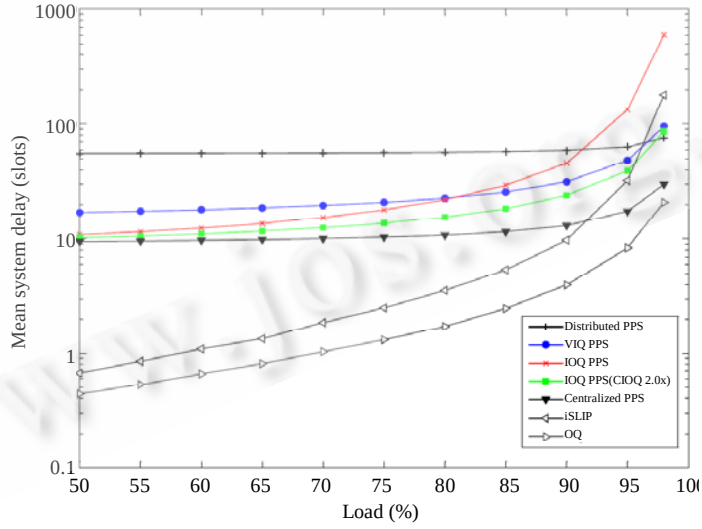


Fig.3 Delay performance for OQ, iSLIP and various PPS's (Bernoulli_iid_uniform)

图 3 OQ,iSLIP 以及各种 PPS 系统平均延迟特性(贝努利一致流量模型)

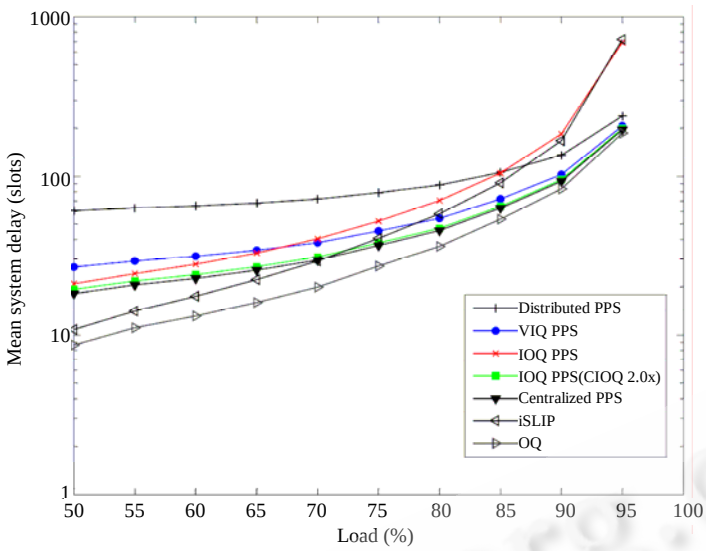


Fig.4 Delay performance for OQ, iSLIP, and various PPS's (bursty)

图 4 OQ,iSLIP 以及各种 PPS 系统平均延迟性能(突发流量模型)

图 5 显示了在贝努利一致流量模型下内部链路加速比的实验结果,IOQ PPS 采用 CIOQ2.0x 交换平面.当内部链路加速比为 2x 时,IOQ PPS 延迟性能得到了显著提高:负载低于 85%时,延迟几乎减少了一半;负载达到 85%以上时,延迟减少了一半以上;负载等于 98%时,延迟减少了将近 2/3.将内部链路速率加速比进一步增加到 3x 时, IOQ PPS 延迟性能只得到了少量提升.在 PPS 系统中,内部链路加速比直接影响着中间层交换平面加速比,例如,CIOQ 交换平面至少运行在与内部链路相同的速率,OQ 交换平面操作速率必须 N 倍于内部链路速率.因此,PPS 系统在增加内部链路加速比的同时必然增强对交换平面的性能要求,从而加大了硬件开销.

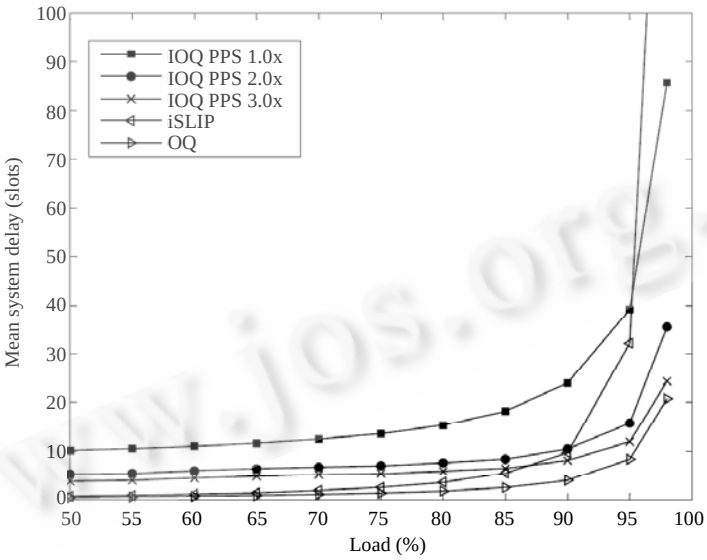
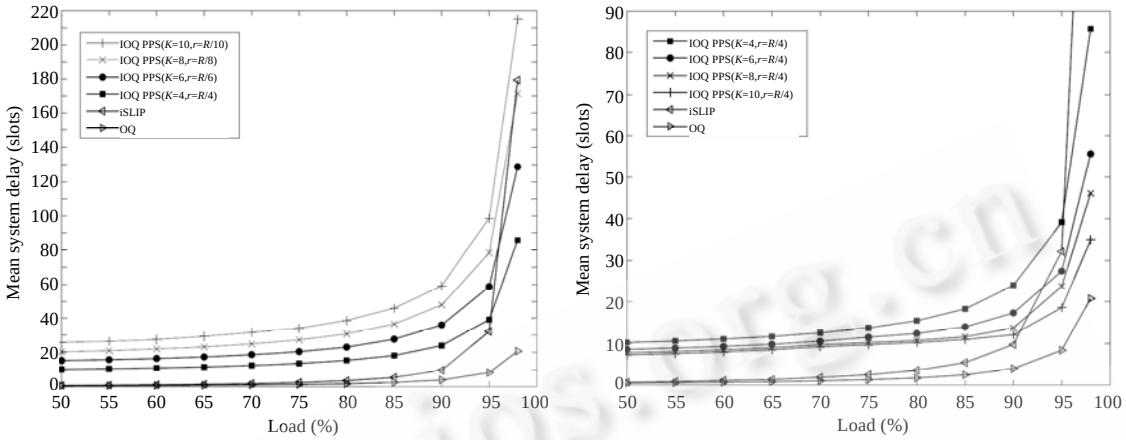


Fig.5 Delay performance for IOQ PPS for speedup experiment

图 5 IOQ PPS 加速比实验的延迟性能

图 6 显示了在贝努利一致流量模型下交换平面数目的实验结果,IOQ PPS 采用 CIOQ2.0x 交换平面.图 6(a)

显示了当内部链路加速比不变时(S=1),IOQ PPS 的系统平均延迟随着交换平面数目(K)的增加而增加,这是因为内部链路速率 $r=R/K$,交换平面数目(K)越大,内部链路速率(r)越低,从而导致系统平均延迟增加.这表明内部链路速率的降低抹煞了增加交换平面所带来的好处;与交换平面数目相比,内部链路速率对 IOQ PPS 延迟性能的影响更大.另一方面,图 6(b)显示了当内部链路速率 r 不变时($r=R/4$),IOQ PPS 系统平均延迟随着交换平面数目(K)的增加而降低,这是因为更多的交换平面增加了 IOQ PPS 的平行度.在这种情况下,信元大部分的延迟发生在分流控制器端.



(a) Case 1 (b) Case 2

(a) 第 1 种情况 (b) 第 2 种情况

Fig.6 Delay performance for IOQ PPS for number of switch planes experiment

图 6 IOQ PPS 交换平面数目实验的延迟性能

5 结束语

本文提出一种维序的 PPS 体系结构 IOQ PPS.与目前主流的 PPS 设计相比,IOQ PPS 信元分派及信元重组算法简单,易于硬件实现.分流控制器通过引入固定大小为 NK 个信元的缓冲区实现了负载在每个交换平面的均匀分配.采用 CIOQ 交叉开关作为中间层交换平面,与目前广泛应用于 PPS 设计的 OQ 交换平面相比,CIOQ 交换平面不需要加速比,解决了 OQ 交换平面“N 倍加速比”问题,降低了硬件开销.通过广泛的理论分析和全面的模拟实验,本文对 IOQ PPS 性能进行了深入的评测.从理论上证明了使用轮询分派算法和同步调度策略的 IOQ PPS 具有按序排队特性,从而实现了同一条流的信元经过交换平面并行交换后不乱序.进一步提出 SLQF 同步调度算法,极在地降低了报文重定序通信开销:只有在“饿死”现象发生时才启动一次中央调度器到分流/重组控制器的通信,而不是在每个时间槽.延迟实验模拟结果显示,当 CIOQ 交换平面加速比等于 2 时,IOQ PPS 比目前主流的 PPS 系统更稳定,信元延迟更低.最后模拟分析了内部链路加速比和交换平面数目对 IOQ PPS 延迟性能的影响.下一步工作包括如何在 IOQ PPS 中提供 QoS 保证.

References:

[1] McKeown N. Memory for High Performance Internet Routers, Presentation to Micron. 2003. http://tiny-tera.stanford.edu/~nickm/talks/Micron_Feb_2003.ppt

[2] Iyer S, McKeown NW. Analysis of the parallel packet switch architecture. IEEE/ACM Trans. on Networking, 2003,11(2):314−324.

[3] Aslam A, Christensen K J. A parallel packet switch with multiplexors containing virtual input queues. Computer Communications, 2004,27:1248−1263.

[4] Cuppu V, Jacob B, Davis B. A performance comparison of contemporary DRAM architectures. In: Proc. of the 26th Int’l Symp. on Computer Architecture (ISCA’99). 1999. 222−233. http://www.ece.umd.edu/~blj/papers/isca99.pdf

[5] http://www.rambus.com

[6] Iyer S, Kompella R, McKeown N. Analysis of a memory architecture for fast packet buffers. In: Proc. of the IEEE Workshop on High Performance Switching and Routing. 2001. 368−373. http://klamath.stanford.edu/~nickm/papers/mmahpsr01.pdf

[7] Bianco A, Giaccone P, Leonardi E, Neri F. A framework for differential frame-based matching algorithms in input-queued switches. In: Proc. of the IEEE INFOCOM. Piscataway: Institute of Electrical and Electronics Engineers Inc., 2004. 1147−1157.

[8] Oki E, Rojas-Cessa R, Chao H J. A pipeline-based maximal-sized matching scheme for high-speed Input-buffered switches. IEICE Trans. on Communications, 2002,E85-B(7):1302−1311.

[9] Chang CS, Lee DS, Jou YS. Load balanced Birkhoff-von Neumann switches part I: One-Stage buffering. Computer Communications, 2002,25(6):611−622.

[10] Chiussi F, Khotimsky D, Krishnan S. Generalized inverse multiplexing of switched ATM connections. In: Proc. of the IEEE GLOBECOM. Piscataway: Institute of Electrical and Electronics Engineers Inc., 1998. 3134−3140.

[11] Khotimsky D, Krishnan S. Evaluation of open-loop sequence control schemes for multi-path switches. In: Proc. of the IEEE ICC. Piscataway: Institute of Electrical and Electronics Engineers Inc., 2002. 2116−2120.

[12] Mneimneh S, K. Siu. Scheduling unsplittable flows using parallel switches. In: Proc. of the IEEE ICC. Piscataway: Institute of Electrical and Electronics Engineers Inc., 2002. 2410−2415.

[13] Khotimsky D, Krishnan S. Towards the recognition of parallel packet switches. In: Proc. of the Gigabit Networking Workshop in Conjunction with IEEE INFOCOM. Piscataway: Institute of Electrical and Electronics Engineers Inc., 2001.

[14] Iyer S, McKeown N. Making parallel packet switches practical. In: Proc. of the IEEE INFOCOM. Piscataway: Institute of Electrical and Electronics Engineers Inc., 2001. 1680−1687.

[15] Iyer S, Awadallah A, McKeown N. Analysis of a packet switch with memories running slower than the line rate. In: Proc. of the IEEE INFOCOM. Piscataway: Institute of Electrical and Electronics Engineers Inc., 2000. 529−537.

[16] Khotimsky D, Krishnan S. Evaluation of open-loop sequence control schemes for multi-path switches. In: Proc. of the IEEE ICC. Piscataway: Institute of Electrical and Electronics Engineers Inc., 2002. 2116−2120.

[17] Cruz R L. A calculus for network delay, Part I: Network elements in isolation. IEEE Trans. on Information Theory, 1991,37(1): 114−131.

[18] Mc Keown N. Scheduling algorithms for input-queued cell switches [Ph.D. Thesis]. Berkeley: University of California, 1995.

[19] http://klamath.stanford.edu/tools/SIM/. 2002.

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

戴艺(1980－),女,湖南邵东人,博士生,主要研究领域为计算机网络体系结构,交换技术,调度算法.

孙志刚(1973－),男,博士,副研究员,CCF高级会员,主要研究领域为计算机网络与通信,交换技术,调度算法.

苏金树(1962－),男,博士,教授,博士生导师,CCF 高级会员,主要研究领域为计算机网络,信息安全.