

文章编号:1001-9081(2005)07-1491-03

一种支持变长分组的 CIOQ 交换结构

张树旗¹, 贾树恒²

(1. 信息工程大学 国家数字交换系统工程技术研究中心, 河南 郑州 450002;

2. 河南农业大学 基础科学学院, 河南 郑州 450002)

(copper2002@sohu.com)

摘 要: 在分析了组合输入输出排队结构的基础上, 对传统 CIOQ (Combined Input-Output Queued) 的输出队列进行扩展和在内部交换结构中采用并行传送的方式, 实现了交换调度的分布式操作和内部无加速的 CIOQ 交换; 又通过将输出队列的状态信息反压到输入端和在输出端采取基于整包调度的算法, 实现了对变长分组的交换, 减小了定长信元交换中分组切割和重组的开销。

关键词: 交换结构; 调度算法; 组合输入输出排队 (CIOQ); 变长分组交换

中图分类号: TP393.07 **文献标识码:** A

CIOQ-based variable packet switches

ZHANG Shu-qi¹, JIA Shu-heng²

(1. National Digital Switching System Engineering & Technology R&D Center, Information Engineering University, Zhengzhou Henan 450002, China;

2. College of Basic Science, Henan Agriculture University, Zhengzhou Henan 450002, China)

Abstract: By extending the output queue (OQ) in traditional Combined Input-Output Queued (CIOQ) and using parallel transmission method in the internal switching architecture, the switching fabric could work in non-speedup with distributed operation of scheduling. By feeding the OQ's state back into the input, the switching architecture was able to fit for variable packets, and the expenses caused by segmentation and reassembling the packets in cell-based switching fabric were decreased.

Key words: switching architecture; scheduling algorithm; Combined Input-Output Queued (CIOQ); variable packets switching

0 引言

由于网络通信量的不均匀性, 导致在很短时间内可能有大量的数据分组同时到达, 因此要求交换系统具有好的缓冲能力, 对到来的数据分组进行排队服务。在交换的输出端设置缓存队列, 当一个分组到达输入端口后被立即送到输出队列, 这种输出排队的方式对所有流量模式具有最佳的时延和吞吐量率性能; 但是当 N 个输入要同时去往同一个输出队列时, 若要求不丢失分组就需要 N 倍线路带宽的存储器, 这限制了交换的容量和可扩展性。在交换的输入端设置缓存队列可以减小对存储器带宽的要求; 但采用单 FIFO (First In First Out) 作为缓存队列结构时, 这种输入排队方式存在队列头部阻塞问题, 使得最大的吞吐率只能达到 58.6%^[1]。采用虚拟输出排队 (Virtual Output Queuing, VOQ) 的方式虽然可以提高最大吞吐率到 100%, 但需要高效的集中式的调度来保证其性能。虽然组合输入输出排队 (Combined Input and Output Queuing, CIOQ) 的方式通过采用某种匹配算法在加速比为 2 的条件下也可以达到 100% 的吞吐率^[2], 但是这些匹配算法都要求集中式的调度, 这在实际的实现中复杂度过高且不宜于进一步扩展。本文介绍了一种能采用分布式调度的 CIOQ 交换结构, 该结构降低了实现的复杂度, 方便了系统的扩展。

现有的交换调度算法大都是基于定长信元设计的, 而在实际的 IP 网络中传输的都是变长的分组, 因此在交换过程中

需要将分组进行切割和重组, 使得交换的开销大大增加。虽然文献[3]中对输入排队交换模型下基于分组的调度算法进行了研究, 但在内部交换中仍然是以定长信元处理的。本文中介绍的调度算法是基于整包实现的, 可以完全避免分组的切割与重组。

1 分布式调度的 CIOQ 结构

输出排队能提供良好的交换性能, 但要求交换内部加速 N 倍; 输入排队虽然没有加速要求, 但对交换的性能不能很好地保证。组合输入输出排队 (CIOQ) 将到达的分组在输入和输出端分别进行缓存, 使得在低加速因子的条件下便能实现好的交换性能。

1.1 传统的 CIOQ 结构

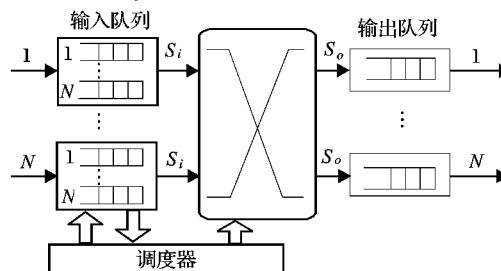


图1 组合输入输出排队, 输入加速因子 S_i , 输出加速因子 S_o 。

传统的 CIOQ 交换结构如图 1 所示, 主要由三部分组成:

收稿日期: 2004-12-19; 修订日期: 2005-03-08 基金项目: 国家 863 计划项目 (2003AA103510)

作者简介: 张树旗 (1975-), 男, 黑龙江林口人, 硕士研究生, 主要研究方向: 高性能网络; 贾树恒 (1977-), 男, 河南驻马店人, 助教, 主要研究方向: 电子信息系统。

输入队列、交换选路结构和输出队列。设输入加速因子 S_i 为一个周期内一个输入队列能够传输的分组数目; 输出加速因子 S_o 为一个周期内能够送到输出队列的分组数目。 S_i 和 S_o 可以在 1 和 N 之间取任意值, 当两者都取 1 时等同于典型的输入排队系统; 当 $S_i = 1, S_o = N$ 时系统等同于典型的输出排队系统。而内部交换选路结构的加速比 $S (S < N)$ 是输出加速因子和输入加速因子的比值, 即 $S = S_o / S_i$ 。

传统的 CIOQ 结构中交换选路结构大都采用交叉开关矩阵, 它的调度算法是在输入和输出间选择无冲突的匹配, 这个匹配在每个时隙内计算, 而且只能采用集中的方式^[4]。同时为了实现交叉开关高的利用率和无冲突匹配, 现在普遍采用内部加速的方法。但是加速要求内部集中调度器必须比原先运行快 S 倍, 而且缓存必须提供 $(S + 1)/2$ 倍的吞吐率。若能在各个端口采用分布式的调度, 则上述问题将会变得简单易行。

1.2 分布式调度的 CIOQ 结构

在图 1 的结构中, 若采用分布式的调度, 需要在每个端口设置一个调度器且彼此间互不交互信息。但是当多个输入端的分组要去往同一个输出端口时, 在输出队列处会产生输入冲突。解决输入冲突的方法有两种: 1) 输出队列采用一个 N 倍线路带宽的共享缓存; 2) 将每个输出端口的共享缓存都分成 N 个队列, 一个队列只对应于一个输入, 即分布式缓存。这种结构将简化调度的实施——将原先的集中式调度器变成了 $2N$ 个独立的部分, 由它们共同完成系统的调度。若输出端通过反压信号将输出队列的状态反馈到输入端, 则集中调度和分布式调度从长的时间上考虑是等价的。但在分布式调度的 CIOQ 结构中, 加权轮询 (WRR) 和严格优先级调度能够在输入和输出队列中实施而不影响交换吞吐率, 因而也不需要内部加速来弥补低效率的调度策略。同时 $2N$ 个调度器可以并行工作, 而且并不要求 $2N$ 个调度器一定工作在同一时刻, 这就使得系统传输分组不必以定长为单位。另外, 直接以变长分组交换也能消除内部加速的其他障碍, 如补偿分组不是分段的整数倍时的分割开销; 同时也减少了入口线卡的重组装置和缓存。

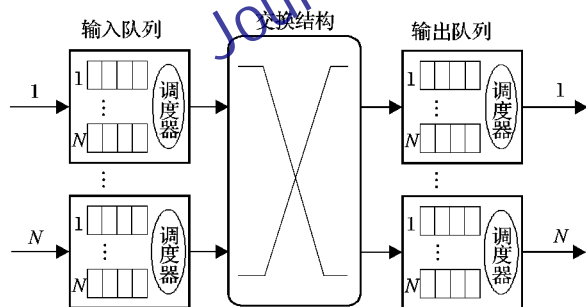


图2 分布式调度的 CIOQ 结构

采用如图 2 所示结构用来直接交换变长分组, 在每个输入端设置单独的 VOQ 队列用来缓存未能送交输出缓存的分组; 在每个输出端设置 N 个单独队列用来缓存不同输入端送来的分组。当一个输出队列满时, 发送一个反馈信号给输入端来避免输出缓存的溢出。当分组 P_{ij} (i 端口输入, j 端口输出) 进入到系统中, 它先被缓存在输入端 VOQ 队列 A_{ij} ($j \in [1, N], i$ 不变) 中, 当 i 端口的调度器收到输出端 j 的第 i 个队列 B_{ij} ($i \in [1, N], j$ 不变) 送来的选择反压信号时, 使能 P_{ij} 传输到队列 B_{ij} 等待调度输出。这样, 通过队列交换重组后 CIOQ 变成了两级排队结构, 即在输入端是按目的端口排队的 VOQ 结构, 在输出端是按源端口排队的并行输出排队结构。在这种结构中, 我们将调度器分别放置在每个端口。在输入端口

调度器根据输出缓存的状态信息选择哪个 VOQ 队列得到服务输出; 而输出端调度器对输出队列中的分组进行整包调度, 提高交换的性能。

2 基于分组的调度算法

WRR 调度算法^[5]对每一个队列设置一个权值, 轮询到该队列时, 根据其权值发送一定量的数据。但是 WRR 是与“字节”打交道的, 而不是与“包”打交道。每一轮轮询, 调度机都访问所有的队列, 当访问到某一队列时, 根据权值决定从该队列调度多少个字节输出。由于 WRR 以“字节”分配带宽, 而 IP 网络不能发送半个包, 必须寻找可实现的改进或近似方法。

在定长信元交换中, 调度器周期地更新状态, 更新周期等于两定长信元的到达时间差 (时隙)。而变长分组中, 分组长度不定, 那么调度器将不能再以固定的时间更新。因此, 要基于分组调度, 需要首先确定传输分组的开始和结束。输入队列控制器首先检查队列中到达的数据头标识 SOP (Start of Packet), 然后等待该分组的尾标志 EOP (End of Packet) 到来。若收到 EOP, 则控制器生成一个完整包标识送给输入调度器; 若收到另一个 SOP (前一个 SOP 丢失, 不完整包), 则丢弃已接收的分组数据, 接收下一个分组。每个输入调度器按照收到的完整包标识, 调度器采用如下策略来实现分组调度:

- 1) 每个输入的 VOQ 每收到一个 EOP 标志时, 将该队列号发送给调度器一次, 调度器收到后将它存放在一个有序列表中。若列表中已有内容, 则新收到的插到列表尾部。
- 2) 每个输出队列都设置一个阈值 T , 当空闲缓存大于 T 时, 输出队列向输入调度器发送一个允许输出标志。
- 3) 输入调度器每收到一个允许输出标志执行以下操作: 从列表头开始依次将列表中的队列号同输出队列送来的允许输出标志相比较, 相匹配的队列就将缓存的分组送到输出队列等待输出。而本次匹配的队列号从列表中删除, 未匹配依次前移。

在每个输入端的调度器同时执行, 互不交互, 完全实现分布式调度。在输出端的调度器也同输入端一样完全分布式执行, 采用基于整包的调度策略调度输出:

- 1) 每个输出端的每个队列, 收到一个完整包时产生一个队列标识, 存放在一个单队列列表中, 若同时有两个队列产生标识, 则列表按队列序号先后写入; 若非空, 则新的标识加在列表尾部。
- 2) 调度器循环读取列表的内容, 每次从列表头队列标识的队列中输出一个分组, 并将这个标识从队列头部删除。

3 性能分析

在定长信元 (cell) 交换模式下, 到达的分组都被分割成定长的信元, 在交换完成后重组最初的形式离开系统。在交换中采用定长信元使得调度算法的执行比变长分组容易, 但是用定长信元交换主要存在以下两方面的不足: 1) 需要一个专用的输入分割模块, 将到达输入端的分组分割成信元, 在输出端这些信元再被重新组合, 这增大了系统的操作开销。2) 由于一个信元不能包含不同分组的数据, 因而在信元分割时就可能产生不完整的信元, 这就需要信元填充, 浪费了带宽。例如, 一个 64 字节的分组, 若采用 40 字节的信元, 那么总的带宽损失就是 $(64 - 40)/64 \approx 37\%$ 。而采用基于变长分组的交换, 操作完全以整包进行, 减少了分组切割和重组的开销; 而且仅需要对分组添加简短的头尾标识, 与定长分组的填充相比, 提高了带宽的利用率。

由输入端调度算法可知,只要可用的空闲输出缓存超过阈值 T ,输入 VOQ 的分组就能得到服务输出到输出缓存。而在输出队列中,一个到达分组只要在前面的分组服务结束后就能得到服务输出,同时由于输出队列长度受控于阈值 T 而不会出现无限长的情况,因此新到达的分组在一段有限时间后一定能得到服务。在一段有限时间后输出队列的空闲缓存空间一定会大于 T ,同时只要输入端的缓存足够大而不至于引起输入队列溢出而丢弃分组,输入的分组就一定能到达输出队列。因此,在输入缓存足够大的情况下到达的完整分组不会在系统中拥塞丢弃,整个交换系统可以达到 100% 的吞吐率。而在均匀流量模式下,这种 CIOQ 模型在负载 50% ~ 90% 时,平均时延比理想的 OQ 交换仅有稍微的增加,如图 3 所示。

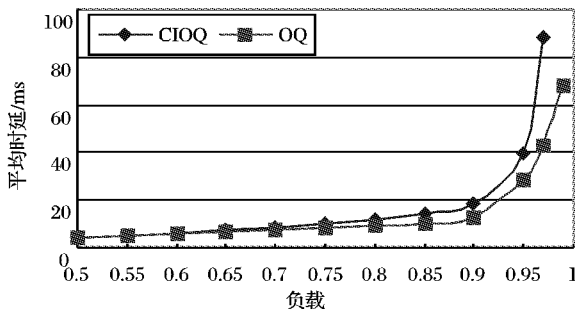


图3 CIOQ 和理想 OQ 的负载和平均时延比较

对于每个输出端的平行队列,设队列容量为 M 。由于我们对变长分组采用整包调度,每次以一个分组为单元,因此每次调度输出时队列中至少要有有一个完整的分组存在,这样队列容量 M 最小为一个最大分组长度。若队列容量 $M = MTU$,则输出缓存的阈值 T 应该满足:当正在输出的分组完全离开队列的同时队列中应该至少还有一个完整的分组可以被再次调度输出。因此,理想的情况下阈值 T 应该等于 $M/2$,即一个最大分组长度的一半。但是在实际操作中取 M 大于 MTU ,即 $M = 3k$ 字节, $T = MTU$ 。当输出队列中空闲空间小于 T 时输出队列输出一个分组,若这时空闲队列仍小于 T ,则继续从输出队列

中输出分组;若队列空闲空间大于 T 时,则在继续输出该队列中分组的同时再向输入端的调度器发送允许输出标识,那么在该队列为空时输入的分组已经存储到了队列中,可以继续连续输出,避免了工作不守恒状态的出现。若初始时输出队列空闲空间大于 T ,那么输入队列的分组就直接写到输出队列中输出。该结构相当于采用了两级流水线操作,实现了系统的守恒工作。

4 结语

文章在分析传统 CIOQ 交换结构的基础上对输出队列和内部交换互连部分进行扩展,实现了交换调度的分布式操作,降低了调度实现的复杂度。同时在扩展 CIOQ 结构中通过反馈控制和基于包调度的 WRR 算法,在保证 100% 吞吐率的情况下实现了对变长分组的交换。另外,可以适当修改输入端的队列分布规则使得这种 CIOQ 交换结构实现对区分服务的 QoS 支持。

参考文献:

- [1] KAROL M, HLUCHYJ M, MORGAN S. Input versus output queuing on a space division switch[J]. IEEE Transactions on Communication, 1988, 35(12): 1347-1356.
- [2] PRABHAKAR B, MCKEOWN N, STAN-CSL-TR-97-738, On the Speedup Required for Combined Input and Output Queued Switching, Stanford University, 1997.
- [3] MARSAN M, BIANCO A, GIACCONE P, et al. Packet Scheduling in Input-Queued Cell-Based Switches [A]. INFOCOM 2001 [C], 2001. 1085-1094.
- [4] CHRYSOS N. TR 325, Design Issues of Variable-Packet-Size, Multiple-Priority Buffered Crossbars [R]. Heraklion, Crete, Greece, 2003.
- [5] SHIMONISHI H, SUZUKI H. Analysis of Weighted Round Robin Cell Scheduling and Its Improvement in ATM Networks [J]. IEICE Transactions on Communications, 1998, E81-B(5): 910-927.

(上接第 1490 页)

假定输入端口 1 中有 4 个流分别输出到端口 1 到 4,输入端口 2 中有 4 个流分别输出到端口 5 到 8,流 $f(1,1)$, $f(1,2)$, $f(1,3)$, $f(1,4)$ 的预约带宽分别为 40%, 30%, 20%, 10%, 流 $f(2,5)$, $f(2,6)$, $f(2,7)$, $f(2,8)$ 的预约带宽分别为 40%, 30%, 20%, 10%, 改变各流的输入负载,我们看各流实际得到的带宽,见图 5。

可见,当输入负载小于 25% 时,各流得到的带宽相同,当输入负载大于 40% 时,各流按照预约的比例得到带宽。当输入负载在 25% 和 40% 之间时,例如当负载为 30% 时,流 $f(1,3)$, $f(2,7)$ 得到的带宽分别为 26.8%, 流 $f(1,4)$, $f(2,8)$ 得到的带宽分别为 13.4%, 分别多得到 6.8% 和 3.4%, 因此剩余的带宽按照预约的比例在各流间公平的分配。

4 结语

传统的 crossbar 交换结构广泛用于各种现代路由器中,但传统的 crossbar 交换结构在提供良好的 QoS 方面存在着很大不足,本文在传统的交换结构基础上提出并讨论了一种新的交换结构——CICQ (combined input and crosspoint buffered queuing) 交换结构,这种交换结构相比传统的交换结构不但在各种输入流下能提供更好的吞吐率、更高的效率,其吞吐率

接近输出排队交换结构,而且能够实现良好的 QoS 能力。本文提出了在 CICQ 交换结构下实现分布式的 DRR 加权公平调度算法的方案,这一方案具有良好的可扩展性,可以适应高速的网络环境。仿真结果显示,分布式的 DRR 调度方案能够实现各个流在输入排队情况下的公平调度

参考文献:

- [1] PAREKH A, GRILLER R. A generalized processor sharing approach to flow control - The single node case [A]. Proc IEEE InfoCom [C], 1992. 915-924.
- [2] SHREEDHAR M, VARGHESE G. Efficient fair queuing using deficit round robin [A]. SIGCOMM [C], Boston, 1995.
- [3] MCKEOWN N, ANANTHARAM V, WALRAND J. Achieving 100% throughput in an input-queued switch [A]. Proc InfoCom'96 [C], 1996. 296-302.
- [4] STEPHENS D, ZHANG H. Implementing Distributed Packet Fair Queuing in a Scalable Switch Architecture [A]. Proc INFOCOM [C], 1998.
- [5] KATEVENIS M, PASSAS G, SIMOS D, et al. Variable Packet Size Buffered Crossbar (CICQ) Switches [A]. Proceedings of IEEE International Conference on Communications (ICC 2004) [C]. Paris, France, 2004.