

一种具有信元保序能力的 Clos 网络分布式调度算法

杨君刚^{1),2)} 鲍民权¹⁾ 刘增基¹⁾ 邱智亮¹⁾ 赵瑞琴¹⁾ 石增增¹⁾

¹⁾(西安电子科技大学综合业务网国家重点实验室 西安 710071)

²⁾(西安通信学院 西安 710106)

摘 要 分组交换三级 Clos 网络信元调度算法可分为集中式和分布式两种实现方式. 分布式调度具有良好的可扩展性, 适于在高速大容量环境中应用. 然而由于分布式调度会带来同一分组各个信元间的乱序问题, 给其实现带来困难. 该文提出了一种具有信元保序能力的三级 Clos 网络分布式调度算法. 该算法包括第一级的均匀负载分配、中间级的并行调度和第三级的按序输出调度三部分. 文中对算法的性能进行了严格的理论证明和相关的仿真分析, 表明该算法可以很好地解决传统分布式调度中的信元乱序问题, 具有良好的性价比.

关键词 三级 Clos 网络; 分布式控制; 调度算法; 信元保序

中图法分类号 TN915

A Distributed Scheduling Algorithm Maintaining Cells Order for Three-Stage Clos Networks

YANG Jun-Gang^{1),2)} BAO Min-Quan¹⁾ LIU Zeng-Ji¹⁾ QIU Zhi-Liang¹⁾

ZHAO Rui-Qin¹⁾ SHI Zeng-Zeng¹⁾

¹⁾(National Key Laboratory of Integrated Service Networks, Xidian University, Xi'an 710071)

²⁾(Xi'an Communication Institute, Xi'an 710106)

Abstract The cell scheduling algorithm for packet switching three-stage Clos networks can be implemented by centralized or distributed controlling scheme. The latter becomes more attractive as the switch becomes larger, because of its good scalability. However, this scheme may cause the cells of a flow mis-sequence, which limits its application. A distributed scheduling algorithm that could maintain cells order is proposed in this paper. It consists of three parts: load-balanced dispatch in the first stage, parallel scheduling in the second stage and scheduling cells in order at an output port in the third stage. The good performance and economy of this algorithm are shown by theoretical and simulation analysis in this paper.

Keywords three-stage Clos networks; distributed control; scheduling algorithm; maintaining cells order

1 引 言

Clos 网络^[1]由于其模块化性强和良好的可扩展

展能力, 成为实现大容量路由器交换网络的理想选择之一^[1-2]. 目前对分组交换三级 Clos 网络结构的研究可分为集中式^[2-6]和分布式^[7]两种. 集中式结构又可分为两大类: 无缓存(Bufferless)Clos 网络和缓

收稿日期: 2005-03-29. 最终修改稿收到日期: 2007-02-06. 本课题得到国家“八六三”高技术研究发展计划项目基金(2002AA103062)、中兴通信股份有限公司技术研究基金(ZXJS200609120159)、ISN 国家重点实验室开放课题(ISN8-03)资助. 杨君刚, 男, 1973 年生, 博士研究生, 主要研究方向为大容量路由器交换网络、下一代光网络关键技术等. E-mail: yangjg_xian@tom.com. 鲍民权, 男, 1959 年生, 副教授, 主要研究方向为宽带交换网络接入和交换技术等. 刘增基, 男, 1937 年生, 教授, 博士生导师, 主要研究领域为宽带网络关键技术等. 邱智亮, 男, 1965 年生, 教授, 博士生导师, 主要研究领域为宽带综合业务数字网、ATM 接入与交换技术、高性能路由器交换技术等. 赵瑞琴, 女, 1981 年生, 博士研究生, 主要研究方向为无线传感器网络、无线移动自组织网等. 石增增, 男, 1982 年生, 硕士研究生, 主要研究方向为光传输设备交换网络调度和保护技术等.

存式(Buffered)Clos 网络^[2]. 无缓存 Clos 网络的三级都采用空分交换结构(Space-Space-Space, 3S), 不带任何缓存, 这种交换结构的好处是交换网络实现简单, 但需要一个专门的分组调度(Packet Scheduling, PS)网络和调度算法来完成信息在网络中转发时的路径选择^[2-4]. 这样会增加网络的实现成本, 同时调度算法复杂, 实现困难; 缓存式 Clos 网络是在网络的第一级和第三级采用共享缓存方式, 中间级采用不带缓存的空分交换方式, 这样网络变为一个 MSM(Memory-Space-Memory) 结构^[5-6], MSM 结构需要第一(三)级为共享缓存方式, 极大地限制了网络的扩展能力^[8]. 因此, 集中式结构随着网络规模的增加, 会带来调度算法复杂性的提高, 降低其可扩展能力, 同时, 不利于网络的分布式实现, 增加了其实现难度^[8]. 为了克服集中式结构的缺点, 文献^[7]提出了一种分布式调度的三级 Clos 网络新结构——分布式 Clos 网络(Distributed Clos, D-Clos)和基于负载均衡的调度思想. 采用这种结构和调度思想可以把三级 Clos 网络中的调度问题分解为第一级的负载均衡和第二(三)级各个交换单元内部调度两个子问题来解决, 网络的三级在调度中不需要相互交换控制信息, 便于在多机架上实现; 同时后两级的调度可以利用现有单 Crossbar 网络调度算法十分丰富的研究成果^[9-14], 使网络具有良好的继承性. 然而分布式调度三级 Clos 网络可能引起同一分组的信元乱序, 解决信元乱序问题是分布式调度 Clos 网络需要解决的重要问题之一^[7]. 本文提出了一种 Clos 网络分布式调度算法——负载分配并发式调度算法(Load Dispatched and Volleyed Scheduling Algorithm, LDVSA), 该算法在保持分布式调度优点的基础上, 可以将信元按照进入交换网络的顺序输出, 很好地解决了信元乱序问题.

为了讨论方便, 本文假设整个交换网络的输入(出)端口数为 N , 三级 Clos 网络第一级为 $n \times n$ 交换单元, 单元数为 $r = N/n$ 个, 中间级交换单元为 $r \times r$, 个数为 n 个, 第三级交换单元为 $n \times n$, 个数为 r 个, 用 $C(n, n, r)$ 来表示该 Clos 网络^①. 分别用 $IM(i)$, $CM(k)$ 和 $OM(j)$ 表示第一级的第 i 个交换单元, 中间级的第 k 个交换单元和第三级的第 j 个交换单元. 本文讨论采用定长信元交换方式, 将按端口速率传输一个信元的时间称为一个时隙(Timeslot).

2 算法概述

三级 Clos 网络分布式调度算法分为三部分, 即

输入级的负载分配算法、中间级解决输出单元竞争的调度算法以及输出级解决输出端口竞争的调度算法^[7]. 相应的 LDVSA 调度算法也包括输入级的完全负载均匀分配算法(Full Load-Balanced Dispatch Algorithm, FLBDA)、中间级的轮询式 n -并行(Round-Robin based n -Parallel, RRnP)调度算法和输出级迭代式最老信元顺序输出(iterative Oldest Cell Order Output, iOCOO)调度算法三部分. LDVSA 结构如图 1 所示. 在输入级采用输入排队, 每个输入端口对到达各个 OM 的信元独自建立队列, 每个队列分为 n 个子队列, 对应 IM 的 n 个输出端口(也就是网络的 n 个中间级); 中间级的 RRnP 算法是由文献^[10]中的 K 并行调度算法(K -parallel scheduling algorithm)改进得到的. 这种算法在一个时隙内对 n 个中间级采用同样的调度安排. 调度算法首先按照轮询方式选择一个 CM, 然后根据该 CM 的排队情况进行计算得到本时隙的调度匹配. 在每个 CM 的一个输出端口只需要一个容量为 N 个信元的缓存就可以将同一分组的各个信元以不乱序的方式送到 OM 级; 原则上讲每个 OM 采用任何适用于单 Crossbar, 保证信元 FIFO(First In First Out, 先入先出)的调度算法, 都可以保证信元流以不乱序的方式输出. 为了实现简单, 本文采用由单 Crossbar 中 iOCF(iterative Oldest Cell First)算法^[13-14]进行简单修改得到的 iOCOO 调度算法来解决 OM 输出端口竞争问题. 下面对 LDVSA 算法的各个部分分别进行讨论.

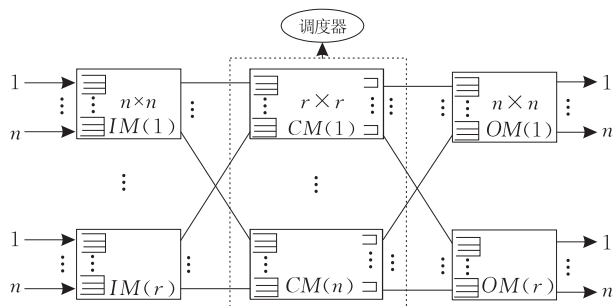


图 1 LDVSA 算法示意图

3 输入级完全负载均匀分配算法(FLBDA)

由输入级的队列组织结构可知, 一个输入级交换单元的每个输入端口总共有 $n \times r$ 个子队列, 称这

① 在本文中假设三级 Clos 网络第一级为 $n \times n$ 对称结构, 只是为了讨论方便, 本文算法同样适用所有 $C(n, K, r)$ 结构.

种队列为 VOSQ (Virtual Output Switch Queued 虚拟输出交换单元排队)。符号定义如下：

$VOSQ^{(l)}(i, k, j)$ ($1 \leq i \leq n, 1 \leq k \leq n, 1 \leq l \leq r, 1 \leq j \leq r$), 表示到达 $IM(l)$ 的第 i 个输入端口经过 $CM(k)$ 到达 $OM(j)$ 的信元在 i 端口的排队队列；

$L_{VOSQ^{(l)}(i, k, j)}$, 表示队列 $VOSQ^{(l)}(i, k, j)$ 的长度, 以信元为单位；

$C^{(l)}(l, i, j)$, 表示在时隙 t 到达 $IM(l)$ 输入端口 i 转发到 $OM(j)$ 的信元。

由于各个 IM 的对称性, 用 $IM(l)$ 来说明输入级负载分配算法, 在不发生混淆的情况下, 分别用 $VOSQ(i, k, j)$, $L_{VOSQ(i, k, j)}$ 和 $C^{(l)}(i, j)$ 来代替 $VOSQ^{(l)}(i, k, j)$, $L_{VOSQ^{(l)}(i, k, j)}$ 和 $C^{(l)}(l, i, j)$ 。不失一般性, 下面选择 $IM(l)$ 的第 i 个输入端口来讨论。

定义 1. 完全负载均匀分配算法 (FLBDA)。令集合 $M = \{k \mid k = \arg \min_{k \in [1, n]} \{L_{VOSQ(i, k, j)}\}\}$; $L_{ij} = \sum_{k=1}^n L_{VOSQ(i, k, j)}$, 集合 $\Gamma = \{j \mid j = \arg \max_{j \in [1, r]} \{L_{ij}\}\}$ 。在时隙 t , 输入端口 i 和该交换单元的输出端口 u 相连, $u = (i+t) \bmod n$ 。令 $t = hn + s$ ($0 \leq s < n, h$ 为自然数), 如果在时隙 hn 选择 $VOSQ(i, i, j)$ ($j \in \Gamma$) 的队头信元输出, 那么在时隙 t 选择 $VOSQ(i, u, j)$ 的队头信元输出; 输入端口 i 将 $C^{(l)}(i, j)$ 放到队列 $VOSQ(i, p, j)$ 中, 如果 $VOSQ(i, p, j)$ 满足以下两个条件之一: (1) 在时隙 $t-1$ 的末端 $p \in M$ 且是 M 的唯一元素; (2) 在时隙 $t-1$ 的末端, p 是集合 M 多个元素中的一个, p 是按轮询方向最接近 u 的值, 既若 $p = (i+t_1) \bmod n$, 那么 $t_1 = \min\{v \mid q = (i+v) \bmod n, q \in M, v > t\}$ 。

FLBDA 算法简单示意图如图 2 所示。

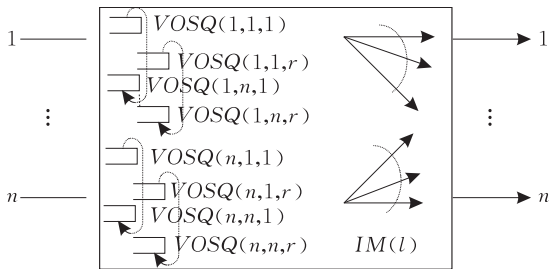


图 2 FLBDA 算法简单示意图

由定义 1 可以看出, FLBDA 算法是将到达信元 $C^{(l)}(i, j)$ 放到信元数最少的 VOSQ, 而信元输出是按 n 个时隙为周期的轮询方式选择, 这样有以下引理。

引理 1. 如果交换网络到达业务是可允许的 (admissible)^[9], 输入级采用 FLBDA 算法, 那么在

时隙 t 末, 当 $t = hn$ (h 为正整数) 时, 对于 $\forall p, q = 1, 2, \dots, n$, 有 $|L_{VOSQ(i, p, j)} - L_{VOSQ(i, q, j)}| \leq 1$, 当 t 为其它值时, 有 $|L_{VOSQ(i, p, j)} - L_{VOSQ(i, q, j)}| \leq 2$ 。

证明. 采用归纳法。假设输入级各个队列在时隙 0 时都是空的, 交换网络从时隙 1 开始到达信元, 信元到达在时隙的开始, 离开在时隙末端。用 t 表示时隙, 当 $t = 1, 2, \dots, n$ 时, 由于一个时隙内最多到达一个信元, 结合 FLBDA 算法很容易验证引理 1 成立; 设在 $t = T+1, T+2, \dots, T+n$ 时, 引理 1 都成立, 下面证明在 $t = T+n+1$ 时隙, 引理 1 仍然成立。设 $T \in [(h-1)n-1, hn-1]$, 任选两个 VOSQ 队列 $VOSQ(i, p, j)$ 和 $VOSQ(i, q, j)$ 进行分析。先证引理前半部分, 当 $T = (h-1)n-1$, 那么 $T+n+1 = hn$, 由假设知, 在 $T+1$ 时隙末, $|L_{VOSQ(i, p, j)} - L_{VOSQ(i, q, j)}| \leq 1$, 因为在 $T+1$ 到 $T+n+1$ 的 n 个时隙内 $VOSQ(i, p, j)$ 和 $VOSQ(i, q, j)$ 两个队列最多各自离开 1 个信元, 而两个队列最多各自到达 1 个信元, 且选择最短队列插入, 所以在 $T+n+1$, $|L_{VOSQ(i, p, j)} - L_{VOSQ(i, q, j)}| \leq 1$ 依然成立; 再证引理后半部分, 如果在时隙 $T+n+1$, $VOSQ(i, p, j)$ 和 $VOSQ(i, q, j)$ 都没有信元离开, 而信元到达最多一个, 还是先选择最短队列, 由假设在 $T+n$ 时隙末引理 1 成立, 所以在时隙 $T+n+1$ 末, 引理 1 依然成立; 如果在时隙 $T+n+1$ 离开的是 $VOSQ(i, p, j)$ 或 $VOSQ(i, q, j)$ 的队头信元, 不妨假设在时隙 $T+n+1$ 离开的是 $VOSQ(i, p, j)$ 队头信元。由 $T \neq (h-1)n-1$, 令 $(T+v) \bmod n = 0$, $0 < v < n$, 由假设, 在时隙 $T+v$ 末, $|L_{VOSQ(i, p, j)} - L_{VOSQ(i, q, j)}| \leq 1$, 如果 $L_{VOSQ(i, p, j)} \geq L_{VOSQ(i, q, j)}$, 显然在时隙 $T+n+1$ 末引理 1 成立; 如果 $L_{VOSQ(i, p, j)} = L_{VOSQ(i, q, j)} - 1$, 由于在时隙 $T+n+1$ 离开的是 $VOSQ(i, p, j)$, 如果在时隙 $T+v+1$ 到 $T+n+1$, $VOSQ(i, q, j)$ 没有信元离开, 那么 $|L_{VOSQ(i, p, j)} - L_{VOSQ(i, q, j)}| \leq 2$; 反之, $|L_{VOSQ(i, p, j)} - L_{VOSQ(i, q, j)}| \leq 1$ 。综上所述, 可以证明引理 1。证毕。

定义 2. 队间保序特性。如果对两个 FIFO 队列的信元从队头到队尾从小到大编号, C_1 和 C_2 是分属于两个队列的信元, 如果 C_1 先于 C_2 到达, 那么 C_1 的序号一定不大于 C_2 , 称这两个队列具有队间保序特性。

引理 2. 按照 FLBDA 分配的任意两个队列 $VOSQ(i, p, j)$ 和 $VOSQ(i, q, j)$ 在时隙 hn (h 为自然数) 末具有队间保序特性。

证明. 很显然, $VOSQ(i, p, j)$ 和 $VOSQ(i, q, j)$ 队列都是 FIFO 队列, 采用反证法。由引理 1, 不失一

般性,假设在时隙 $t=hn$ 末, $L_{VOSQ(i,q,j)} = L_{VOSQ(i,p,j)} + 1$ (如果两个队列等长可以按同样的方法证明). 设 $VOSQ(i,p,j)$ 和 $VOSQ(i,q,j)$ 队列的队尾信元 C_1 和 C_2 在时隙 hn 末不具有队间保序特性. 也就是 C_1 的序号小于 C_2 , 但 C_2 先于 C_1 到达. 假设 C_1 是在时隙 $(h-1)n+v (1 \leq v \leq n)$ 到达, 那么在 C_1 信元到达的前一个时隙末, $L_{VOSQ(i,q,j)} = L_{VOSQ(i,p,j)} + 2$; 从引理 1 的证明过程可以看出, $L_{VOSQ(i,q,j)} = L_{VOSQ(i,p,j)} + 2$ 的情况仅出现在 $(h-1)n$ 末, $L_{VOSQ(i,q,j)} = L_{VOSQ(i,p,j)} + 1$, 且在 $(h-1)n+1$ 到 $(h-1)n+v$ 时隙之间, $VOSQ(i,p,j)$ 队列有信元离开, 而 $VOSQ(i,q,j)$ 队列没有信元离开的情况, 按照 FLBDA 分配算法, 从时隙 $(h-1)n+v$ 到时隙 hn , 只要 $VOSQ(i,q,j)$ 队列非空 (按假设很显然), 肯定要离开一个信元, 而 $VOSQ(i,p,j)$ 不再会有信元离开, 所以在时隙 hn 有 $L_{VOSQ(i,q,j)} = L_{VOSQ(i,p,j)}$, 与假设 $L_{VOSQ(i,q,j)} = L_{VOSQ(i,p,j)} + 1$ 相矛盾. 证毕.

4 中间级调度算法

中间级交换单元采用带 VOQ 的输入排队, 首先按本单元输出端口组成 r 个 VOQ 队列, 每一个对应本单元的一个输出端口 (同样也对应网络的一个输出交换单元), 将一个 VOQ 进一步分为 n 个子队列, 每一个对应一个输入模块的一个输入端口, 称这种队列组织方式为组合虚拟输入输出排队 (Combined Virtual Input Output Queued, CVIOQ). 用 $CVIOQ^{(k)}(i,p,j)$ 表示从 $IM(p)$ 的第 i 个输入端口转发到 $CM(k)$, 从 $CM(k)$ 的第 j 个输出端口输出的信元队列, $VOQ^{(k)}(p,j)$ 表示 $CM(k)$ 的第 p 个输入端口存放到达 $CM(k)$ 第 j 个输出端口信元的队列组. 在不发生混淆时, 用 $CVIOQ(i,p,j)$ 代替 $CVIOQ^{(k)}(i,p,j)$. 很显然, 输入级 $VOSQ^{(p)}(i,k,j)$ 的输出信元就是 $CVIOQ^{(k)}(i,p,j)$ 的输入信元, 这两个队列相对应.

4.1 算法描述和相关性质

定义 3. 轮询式 n 并行调度算法 (RRnP). 在时隙 t , 按照轮询的方式, 选择一个 $CM(k) (k=t \bmod n)$, 按照 $CM(k)$ 在该时隙的排队情况计算一个匹配, 并将该匹配应用到所有 n 个中间级交换单元. $CM(k)$ 的匹配计算按以下规则: (1) 应用适合于 K 并行的调度算法^[10], 计算出一个输入输出端口的匹配 π ; 即对于 $CM(k)$, 如果 $(p,j) \in \pi$, 那么先选定 $VOQ^{(k)}(p,j)$, 很显然 $VOQ^{(k)}(p,j)$ 一定非空; (2) 在选定 $VOQ^{(k)}(p,j)$

的 n 个子队列中, 按以下规则确定一个子队列的队头信元输出: (a) 如果存在 $CVIOQ^{(k)}(i,p,j)$ 队列的长度大于 1, 那么选择该 $CVIOQ^{(k)}(i,p,j)$ 的队头信元; (b) 如果不存在满足 (a) 中条件的 i 值, 因为 $VOQ^{(k)}(p,j)$ 非空, 那么至少存在一个子队列非空, 如果非空的子队列是唯一的, 选择该队列队头信元; 如果非空的子队列有多个, 那么需要看其它中间交换单元相应的队列 $CVIOQ^{(q)}(i,p,j)$ 的非空情况, 选择其它 $n-1$ 个 CM 相对应子队列非空总数最多的一个 $CVIOQ^{(k)}(i,p,j)$ 队头信元; (3) 选定各个 $CVIOQ^{(k)}(i,p,j) (p=1,2,\dots,n)$ 后, 将同样的匹配在本时隙应用到其它的 $n-1$ 个中间交换单元.

定义 4. 队间 FIFO 特性. 如果对两个 FIFO 队列的信元从队头到队尾由小到大编号, 按照某种离开规则, 分别从两个队列中读取信元, 一个队列中序号小的信元离开时隙总不晚于另一个队列中序号比它大的信元离开时隙, 那么称这种排队和离开方式具有队间 FIFO 特性. 根据引理 1 和引理 2 以及定义 3 和定义 4 可以得出以下推论.

推论 1. 输入级采用 FLBDA 分配, 中间级采用 RRnP 调度算法的三级 Clos 网络, 中间级的输入排队满足以下特性: 对于 $\forall k_1, k_2 = 1, 2, \dots, n$, $CVIOQ^{(k_1)}(i,p,j)$ 和 $CVIOQ^{(k_2)}(i,p,j)$ 队列的长度最多相差一个, n 个 $CVIOQ^{(k)}(i,p,j) (k=1,2,\dots,n)$ 队列和 RRnP 调度算法具有队间 FIFO 特性.

证明. 因为输入级 $VOSQ^{(p)}(i,k,j)$ 的输出信元就是 $CVIOQ^{(k)}(i,p,j)$ 的输入信元, 由引理 1、引理 2 和定义 3、定义 4 很容易证明推论成立. 证毕.

引理 3. 如果在三级 Clos 网络中, 输入级采用 FLBDA 负载分配算法, 中间级采用 RRnP 调度算法, 那么在各个中间级交换单元的每个输出端口只需容量为 N 个信元的缓存就可以使经过交换网络的信元不乱序地进入到输出级各个交换单元.

证明. 如果在 $CM(k)$ 输出端口 j 为每个 $CVIOQ^{(k)}(i,p,j) (1 \leq i \leq n, 1 \leq p \leq r)$ 队列来的信元各自建立虚拟输入排队 (Virtual Input Queued) $VIQ^{(k)}(i,p) (1 \leq i \leq n, 1 \leq p \leq r)$, 显然这样的队列共有 $n \times r = N$ 个. 如果对 $CVIOQ^{(k)}(i,p,j) (k=1,2,\dots,n)$ 中的信元从队头到队尾按从小到大进行编号, 由推论, 当 $CVIOQ^{(i)}(i,p,j)$ 中标号为 v 的信元到达 $VIQ^{(i)}(i,p)$ 时, 其它 $CVIOQ^{(s)}(i,p,j) (s=1,2,\dots,n, s \neq i)$ 中编号为 $v-1$ 的信元肯定已经到达 $VIQ^{(s)}(i,p)$, 那么编号是 $v-1$ 的信元可以离开 $VIQ^{(k)}(i,p) (k=1,2,\dots,n)$ 不乱序地进入到输出

级,所以一个 $VIQ^{(k)}(i, p)$ 只需要一个信元容量的缓存. 因此, $CM(k)$ 的一个输出端口 j 需要 N 个信元容量的缓存. 证毕.

从引理 3 的证明过程,可以设计 $CM(k)$ 第 j 个输出端口的每个 $VIQ^{(k)}(i, p)$ 信元输出规则. 即只要任一 $VIQ^{(k)}(i, p)$ 有第 2 个信元到达,那么所有非空 $VIQ^{(k)}(i, p)$ ($k=1, 2, \dots, n$) 中的信元就可以输出到输出级. 由于 $RRnP$ 调度算法由一个调度器来完成,所以可以很简单地实现当一个 $VIQ^{(k)}(i, p)$ 有第 2 个信元到达时,通知所有的 $VIQ^{(k)}(i, p)$ ($k=1, 2, \dots, n$) 将缓存腾空. 前面讨论了 $RRnP$ 调度算法对整个交换网络的性能影响,下面重点分析一下 $RRnP$ 调度算法自身的一些特性.

4.2 $RRnP$ 调度算法的性能

为了更好地讨论 $RRnP$ 算法的性能,首先给出单 Crossbar 中调度算法的一些定义.

定义 5. 基于流水线式调度算法^[10] p PBS (Pipeline-Based Scheduling). 一个交换网络有 p 个子调度器,以 p 个时隙为周期,每一个时隙只有一个子调度器完成调度决定.

在单 Crossbar 中 p PBS 一般用在带有 VOQ 的输入排队中,将一个 VOQ 中的请求分散到 p 个子调度器,在每一个时隙,由一个子调度器根据在其中的请求,完成调度决定,各个子调度器轮流进行,这样可以降低对调度时间的要求,使调度器的调度时间从一个时隙变为 p 个时隙. 同时由于各个子调度器之间是相互独立的,可以保持单调度器调度算法的优点,图 3 示出了 3PBS 的一般工作原理.

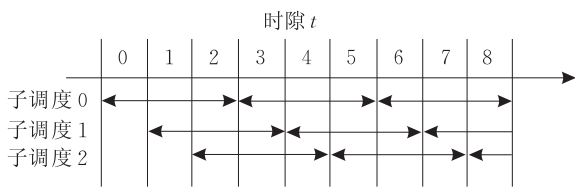


图 3 3PBS 的一般工作原理

一般来讲,所有实用的基于 MWM(最大权重匹配)和极大匹配的调度算法都可以采用流水线方式实现,同时能保持原有算法在吞吐量等方面的性能,如 DRRM(Dual Round-Robin Matching)^[4]、iSLIP^[12]、OCF(Oldest Cell First)^[13-14]等算法.

定义 6. 倍 K 调度^[10] (K -serial Scheduling). 在单 Crossbar 中,倍 K 调度是指调度器应用一个匹配连续 K 次才重新计算下一个匹配.

定义 7. K 并行匹配^[10] (K -parallel Matching).

在 K 个交换单元中,每一个时隙只由一个交换单元的调度器完成调度决定,并将得到的匹配用到所有 K 个交换单元.

定义 8. 调度能力相当. 同样输入业务通过两个采用不同调度算法的交换网络,如果在每一个时隙,一个交换网络转发的信元数总不少于另一个交换网络转发的信元数,那么就称这个交换网络的调度能力相当于另一个交换网络.

定理 1. 采用 FLBDA 和 $RRnP$ 调度算法的 $C(n, n, r)$ 网络的前两级和一个 $n \times n$ 的单 Corssbar 交换网络,如果三级 Clos 网络中间级各个交换单元和单 Crossbar 采用同类的调度算法,单 Corssbar 是基于流水线 n PBS,加速为 n 的倍 n 调度,那么这种三级 Clos 网络前两级的调度能力相当于单 Crossbar 网络.

证明. 采用 n PBS 的单 Crossbar 将每个 VOQ 的请求采用轮询方式分配到各个子调度器,因为 VOQ 中每个排队的信元对应一个请求,所以可以等价地看作将 VOQ 进一步分为 n 个子队列,每个子调度器根据相应队列的排队情况进行调度,用 $PVOQ(i, j, p)$ 表示 VOQ(i, j) 中对应子调度器 p 的子队列,用 $\sum_{r=1}^n CVIOQ^{(r)}(p, i, j)$ 来对应 $PVOQ(i, j, p)$,因为单 Corssbar 具有 n 倍的加速,采用的是倍 n 调度,所以在时隙 t 如果匹配中包含 $PVOQ(i, j, p)$,设 $PVOQ(i, j, p)$ 中的信元个数为 L ,那么将会有 $\min\{L, n\}$ 个信元从 $PVOQ(i, j, p)$ 队列中转发出去;在三级 Clos 网络中,由推论 1,可知 $\sum_{r=1}^n CVIOQ^{(r)}(p, i, j)$ 的信元被均匀地分到各个 $CVIOQ^{(r)}(p, i, j)$ ($r=1, 2, \dots, n$) 中;又因为 $RRnP$ 计算匹配的算法和单 Crossbar 的相同,所以在时隙 t , $RRnP$ 算法也将选择 $CVIOQ^{(r)}(p, i, j)$ 队列,那么很显然两个交换网络的调度能力相当. 证毕.

由定理 1 可以看出,只要将单 Crossbar 中性价比良好的调度算法进行简单的修改就可以应用到 $RRnP$ 中,同时可以保持原有算法在单 Crossbar 中良好的性价比. 在文献[10]中,已经证明很多实用于 Crossbar 的调度算法,经过简单修改,就可以应用在 k 并行调度中. 同时由于大多数调度算法可以通过流水线方式实现^[11],所以可以找到很多实用于 $RRnP$ 的调度算法,因此该算法具有良好的继承性.

4.3 $RRnP$ 算法的通信开销

图 4 是 $RRnP$ 调度器示意图,每一个子调度器

和一个中间级交换单元相对应,从图中和定义 3 可以得出,在每个时隙,首先由此时隙被选中的子调度器(称为主调度器)完成匹配计算,并将计算结果通过总线广播给各个子调度器,这需要 $r \log_2 r$ 比特的通信开销,如果需要(定义 3 规则(b)中情况),每一个子调度器要将已选中各个 VOQ 中的队列情况上传给本时隙的主调度器,用 1 个比特表示一个子队列空还是非空,最多需要 N 个比特,最后再由本时隙的主调度器将最终匹配结果广播给其它的子调度器,需要 $r \log_2 n$ 比特,所以整个 $RRnP$ 调度器在一个时隙内需要的通信开销为 $N + r \log_2 r + r \log_2 n$ 比特. 如果取 $N=64$, $r=n=8$,交换网络信元的长度为 64 个字节,线路速率为 10Gb/s,那么通信开销为 2.1875Gb/s,采用 2.5Gb/s 的数据总线就可以满足要求,这在目前技术下完全可以实现.

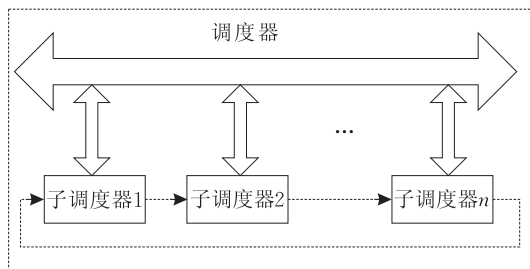


图 4 $RRnP$ 调度器示意图

5 输出级调度算法

在输出级,各个交换单元同样采用带有 VOQ 的输入排队,每一个输入端口排队分为 n 个子队列,对应本单元的 n 个输出端口. $iOCOO$ 调度算法是对启发式 $iOCF$ 调度算法^[13-14]进行简单修改得到的. 和 $iOCF$ 一样, $iOCOO$ 调度算法以信元在交换单元中排队等待的时间长短来划分信元优先级. 在交换单元中等待时间最长的信元称为最老信元(Oldest Cell),在输出端口发生竞争时,优先选择最老信元输出. $iOCOO$ 算法和 $iOCF$ 算法一样分为请求(request)、授权(grant)、接受(accept)和多次迭代四步. $iOCOO$ 算法和 $iOCF$ 算法唯一的区别在于第 2 步授权. 当一个输出端口接收到多个最老信元的请求时, $iOCF$ 算法是按轮询的方式来进行选择,而 $iOCOO$ 算法是按信元到达交换网络的先后顺序来进行选择,这是由 $iOCOO$ 算法特殊的应用场合所决定的. 很显然 $iOCOO$ 算法是一种极大匹配调度算法,由于 $OCOO$ 算法和 OCF 算法^[13-14]思路相同,所以 $OCOO$ 算法具有和 OCF 算法同样优秀的性

能. 另外, $iOCOO$ 算法仅在一个交换单元中实施,而交换网络中一个交换单元的端口数比整个交换网络的端口数要少得多,这样通信开销不会成为交换网络扩展的瓶颈,所以具有更好的可扩展性.

下面证明 $iOCOO$ 算法结合 FLBDA 算法和 $RRnP$ 算法可以保证进入三级 Clos 网络的信元以和输入同样的顺序从网络输出.

定理 2. 采用 LDVSA 算法调度的三级 Clos 网络能够保证输入到交换网络的信元以同样的顺序从交换网络输出.

证明. 采用 LDVSA 调度三级 Clos 网络是指输入级采用 FLBDA,中间级采用 $RRnP$,输出级采用 $iOCOO$ 算法,由推论 1 和 $RRnP$ 算法信元从中间级到输出级输出的规则可知,在每一个时隙,如果该时隙所有非空 $VIQ^{(p)}(i, j)$ ($p=1, 2, \dots, n$) 到达 $OM(j)$ 一个输出端口 q 的信元为 r 个 ($1 \leq r \leq n$),这 r 个信元肯定是在同一个时隙到达 $OM(j)$ 的 r 个输入端口,按照 $iOCOO$ 算法它们肯定是按序离开交换网络的,所以从交换网络 i 端口入到 q 端口出的信元通过 LDVSA 算法调度后,可以按和输入同样的顺序输出.

6 LDVSA 调度算法性能分析

6.1 理论分析

在文献[7]中说明了在分布式调度三级 Clos 网络中,研究的重点是第一级负载均衡算法的性能. LDVSA 算法是由第一级 FLBDA 负载分配算法、第二级 $RRnP$ 轮询调度算法和第三级 $iOCOO$ 算法结合而成. 由定理 1 和前面的算法描述可以得出, $RRnP$ 轮询调度算法和 $iOCOO$ 算法是在单 Crossbar 相应调度算法的基础上修改而来,保持着原有算法的良好性能,其性能保证的主要前提是要求第一级的负载分配算法可以保证到达它们的业务是允许的. 下面讨论 FLBDA 负载分配算法的性能.

定义 $\lambda_{i,x}$ ($1 \leq i, x \leq N$) 表示从交换网络输入端口 i 到达输出端口 x 信息的平均转发流量, $\gamma_{i,x,k}$ ($1 \leq k \leq n$) 为 $\lambda_{i,x}$ 从 $CM(k)$ 转发的百分比,称为 $CM(k)$ 对 $\lambda_{i,x}$ 的均衡系数. 有以下定理.

定理 3. 在分布式 Clos 网络中采用 FLBDA 负载分配算法,如果交换网络的输入业务是允许的,该网络的中间级和第三级各个交换单元的输入业务也是允许的.

证明. 从文献[7]知道,分布式调度 Clos 网络

第一级负载分配算法满足式(1),就可以保证当交换网络的输入业务是允许的,该网络的中间级和第三级各个交换单元的输入业务也是允许的.

$$\sum_{x=n(j-1)+1}^{jn} \sum_{i=1}^N \lambda_{i,x} \gamma_{i,x,k} \leq 1, \sum_{i=n(l-1)+1}^{ln} \sum_{x=1}^N \lambda_{i,x} \gamma_{i,x,k} \leq 1 \text{ 且} \\ \sum_{k=1}^n \gamma_{i,x,k} = 1 (1 \leq j, l \leq r, 1 \leq i, x \leq N, 1 \leq k \leq n) \quad (1)$$

由引理 1 可以知道,对于 $\forall p, q = 1, 2, \dots, n$,在任何时刻都有 $|L_{VOSQ^{(i,p,j)}} - L_{VOSQ^{(i,q,j)}}| \leq 2$,这说明 $\forall p = 1, 2, \dots, n$,都有 $L_{VOSQ^{(i,p,j)}} < \infty$. 否则,如果存在一个 p 有 $L_{VOSQ^{(i,p,j)}} = \infty$,那么对所有的 $p = 1, 2, \dots, n$,都有 $L_{VOSQ^{(i,p,j)}} = \infty$,那么按照 FLBDA 的定义和排队系统不稳定的定义,有到达 $IM(l)$ 第 i 输入端口转发到 $OM(j)$ 的信元个数在每个时限内都大于 1,这样违背了网络到达业务允许的要求,因此是不可能的. 同时,由于 FLBDA 算法对 $VOSQ^{(i,p,j)} (p = 1, 2, \dots, n)$ 队列的输出是按照轮询方式同等对待的,所以各个队列得到信元输出的机会是相同的. 这样在 FLBDA 算法和网络业务到达允许的情况下,有式(2)成立.

$$\begin{cases} \sum_{x=n(j-1)+1}^{jn} \lambda_{i,x} \gamma_{i,x,k} = \frac{1}{n} \sum_{x=n(j-1)+1}^{jn} \lambda_{i,x} \\ \sum_{i=n(l-1)+1}^{ln} \lambda_{i,x} \gamma_{i,x,k} = \frac{1}{n} \sum_{i=n(l-1)+1}^{ln} \lambda_{i,x} \end{cases} \quad (2)$$

同时,由网络业务是允许的有

$$\sum_{i=1}^N \lambda_{i,x} \leq 1, \sum_{x=1}^N \lambda_{i,x} \leq 1 \quad (3)$$

将式(2)和式(3)联合代入式(1)就会得出式(1)成立. 这样定理 3 可得证. 证毕.

6.2 仿真分析

为了进一步说明 LDVSA 算法的性能,对该算法在 OPNET 仿真平台下建立仿真模型,对其时延性能进行分析,并将其和 CRRD 算法^[5]以及 SLBSA 算法^[7]进行比较. 为了提高仿真的可信度,采用文献[15]的业务模型,将整个网络业务分为:非突发的高度均匀业务、非突发的低度均匀业务、非突发的低度不均匀业务和突发的高度均匀业务、突发的低度均匀业务、突发的低度不均匀业务 6 种业务类型,仿真结果如图 5 和图 6 所示.

从图 5 和图 6 可以看出,无论在突发业务还是非突发业务下,LDVSA 算法都继承了分布式调度算法的优点,和集中式的 CRRD 算法相比,在任何网络允许负载下,都是平稳的;而 CRRD 算法虽然在低负载时可以保证良好的网络性能,但是到了中高负载(负载在 0.6~0.7 以上)网络性能就会急剧

恶化. 和分布式 SBLSA 算法相比,LDVSA 算法时延稍大,这是因为在中间级采用 k 并行调度算法,相当于在单 Crossbar 中采用流水线方式实现调度,带来了实现的简单,但是会引入一定的时延代价,LDVSA 算法相对于 SLBSA 算法节省了网络成本.

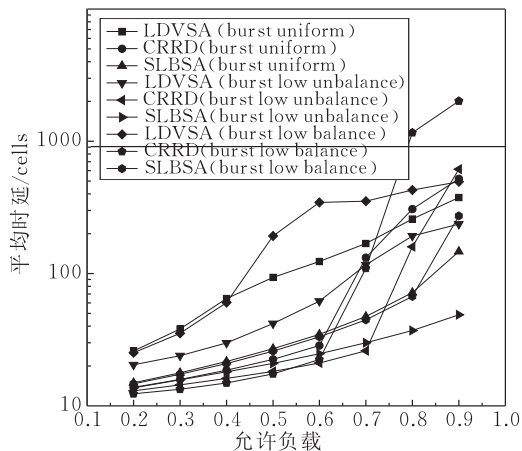


图 5 突发各种业务下 3 种不同算法性能对比图

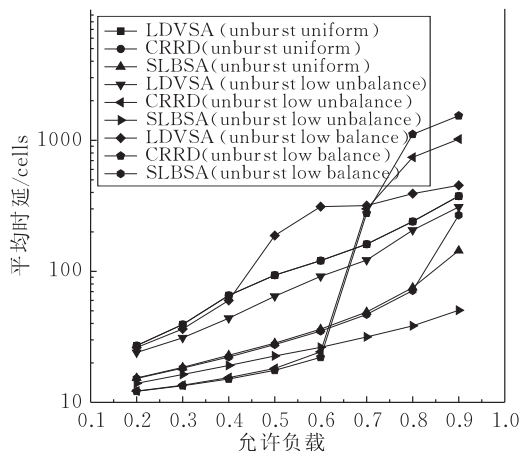


图 6 非突发各种业务下 3 种不同算法性能对比图

从前面的讨论可以看出,LDVSA 算法是一种建立在分布式 Clos 网络结构和负载均衡调度思想基础之上的调度算法,相比于传统的集中式调度算法,具有可扩展性好,调度算法实现简单,可以良好地继承现有单 Crossbar 交换网络调度算法成果等优点;相对于文献[7]所提出的 SLBSA 算法,很好地解决了信元乱序问题,不需要在网络输出端口加重排序缓存来进行信元重排序,降低了网络实现成本.

7 结 论

在三级 Clos 网络中采用分布式调度,可以充分发挥网络良好可扩展性的优势,调度算法实现简单,

具有良好的继承性,很适于在高速大容量环境下应用.然而,由于在分布式 Clos 交换网络结构中,中间级存在缓存会引起属于同一分组的各信元发生乱序问题.乱序问题的存在直接影响了 Clos 网络分布式调度的应用.本文提出了一种可以很好克服信元乱序问题的三级 Clos 网络分布式调度算法——LDVSA 算法.该算法继承了现有分布式调度算法的优点,不仅可以很好地解决信元乱序问题,同时因为该算法是在本地端口进行队列管理和指针轮询,不需要信息在不同单元的交互,算法实现不需要网络加速,所以不会使网络成本大幅度增加,具有很好的经济性.因此,该算法是一种具有良好性价比的调度算法.

参 考 文 献

- [1] Clos C. A study of nonblocking switching fabric networks. Bell Systems Technical Journal, 1953, 32(5): 406-424
- [2] Chao H J, Jing Zhi-Gang, Liew S Y. Matching algorithms for three-stage bufferless Clos network switches. IEEE Communication Magazine, 2003, 41(10): 46-54
- [3] Chao H J, Liew S Y, Jing Zhi-Gang. A dual-level matching algorithm for 3-stage Clos-network packet switches//Proceedings of the 11th Symposium on High Performance Interconnects, Stanford University. Palo Alto, California, 2003: 38-43
- [4] Chao H J. Saturn: A terabit switch using dual round robin. IEEE Communication Magazine, 2000, 38(12): 78-84
- [5] Oki E, Jing Z, Rojas-Cessa R, Chao H J. Concurrent round-robin-based dispatching schemes for Clos-network switches. IEEE/ACM Transactions on Networking, 2002, 10(6): 830-844
- [6] Pun K, Hamdi M. Distro: A distributed static round-robin scheduling algorithm for bufferless Clos-network switches//

Proceedings of the IEEE Globecom. Taipei, China, 2002: 2298-2302

- [7] Yang Jun-Gang, Qiu Zhi-Liang, Liu Zeng-Ji et al. Study on distributed scheduling algorithm in three-stage Clos networks. Acta Electronica Sinica, 2006, 34(4): 590-595 (in Chinese)
(杨君刚,邱智亮,刘增基等.三级 Clos 网络中分布式调度算法研究.电子学报, 2006, 34(4): 590-595)
- [8] Isaac Keslassy, Chuang S T et al. Scaling Internet routers using optics. Computer Communication Review, 2003, 33(4): 189-200
- [9] Dai J G, Prabhakar B. The throughput of data switches with and without speedup//Proceedings of the IEEE INFOCOM. Tel Aviv, Israel, 2000: 556-564
- [10] Mneimneh S, Sharma V, Siu Kai-Yeung. Switching using parallel input-output queued switches with no speedup. IEEE/ACM Transactions on Networking, 2002, 10(5): 653-665
- [11] Oki Eiji, Roberto Rojas-Cessa, Chao H J. A pipeline-based approach for maximal-sized matching scheduling in input-buffered switches. IEEE Communication Letters, 2001, 5(6): 263-265
- [12] McKeown N. iSLIP: A scheduling algorithm for input-queued switches. IEEE Transactions on Networking, 1999, 7(2): 188-201
- [13] McKeown N. Scheduling algorithms for input-queued cell switches [Ph. D. dissertation]. University of California, Berkeley, USA, 1995
- [14] Charny A. Providing QoS guarantees in input buffered crossbar switches with speedup [Ph. D. dissertation]. Massachusetts Institute of Technology (MIT), Massachusetts USA, 1998
- [15] Goudreau M W et al. Scheduling algorithms for input-queued switches: Randomized techniques and experimental evaluation//Proceedings of the IEEE INFOCOM. Tel Aviv, Israel, 2000: 1624-1643



YANG Jun-Gang, born in 1973, Ph. D. candidate. His current research interests include switching fabric in large capacity routers, key technologies of next generation optical networks, etc.

BAO Min-Quan, born in 1959, associate professor. His current research interests include accessing and switching technologies of broad band networks, etc.

LIU Zeng-Ji, born in 1937, professor, Ph. D. supervisor. His current research interests include key technologies of broad band networks, etc.

QIU Zhi-Liang, born in 1965, professor, Ph. D. supervisor. His current research interests include integrated service digital networks, ATM accessing and switching technologies, high performance router switching technologies, etc.

ZHAO Rui-Qin, born in 1981, Ph. D. candidate. Her current research interests include mobile Ad hoc and sensor networks, application of directional antennas in wireless networks, etc.

SHI Zeng-Zeng, born in 1982, M. S. candidate. His current research interests include scheduling and protecting technologies of switching fabric in optical transmission equipment, etc.

Background

This work is supported in part by the National High Technology Research and Development Program (863 Program) of China under grant No.2002AA103062, the Open Subject Program of National Key Laboratory of Integrated Service Network under grant No.ISN8-03, the Research Funding Subject of Zhong-Xing Corporation under grant No. ZXJS200609120159.

Switching fabric is the key element of a Router or Switch. It functioned as forwarding the cells from input port to correctly output port according their address. The performance of the switching fabric plays a very important role in the performance of Router or Switch.

The scheduling scheme is critical in a switching fabric, it is used to avoid competition for one output port. Clos network is a famous switching fabric, it possesses many merits such as good scalability, favorable network performance, etc. So the scheduling scheme in Clos network is a research

hotspot. However, the scheduling schemes used in current Clos networks are of high complexity and can't guarantee the networks performance. A distributed scheduling scheme is proposed to solve the above problems. Because the buffer is allocated at the central stage, the cells belong to the same packet through different central stages may suffer diverse queued delay, which cause the out-of-sequence of the cells. The mis-sequence is a serious problem faced by the distributed scheduling scheme. The main objective of this paper is to solve the out-of-sequence problem.

In this paper, a distributed scheduling algorithm that could maintain cells order is proposed. Theoretical analysis proves that the algorithm is capable of maintaining the cells order correctly. The good performance and economy of this algorithm are also shown by theoretical and simulation analysis in this paper.