针对高速交换结构的广义极大匹配调度算法

徐 扬[1], 唐 毅[1], 文振 [2], 刘 斌[1]
(1. 清华大学计算机科学与技术系, 北京 100084; 2 深圳大学计算机系, 广东深圳 518055)

摘 要: 调度算法是决定交换结构性能和实现复杂度的重要因素, 极大匹配算法在这两方面存在不足. 本文提出一类广义极大匹配(EMM)算法, 使用不同权值参数能够派生出不同子类的算法. 对广义极大匹配算法的研究从两方面展开, 首先在 2 倍数据加速比下证明任何 EMM(2)算法都能取得 100% 的吞吐量, 并通过仿真表明能够取得与理想输出排队相近的延时性能; 其次在没有加速比的条件下通过仿真表明具有 2 个以上权值参数的广义极大匹配算法能够大大提高极大匹配算法的吞吐量性能.

关键词: 交换结构; 加速比; 调度算法; 极大匹配; 吞吐量
中图分类号: TP393.05 文献标识码: A 文章编号: 0372-2112 (2007) 10-1809-08

Extended Maximal Matching Algorithm in High-Speed Switches

XU Yang[1], TANG Yi[1], WEN Zhen-kun[2], LIU Bin[1]
(1. *Department of Computer Science and Technology, Tsinghua University, Beijing* 100084, *China*;
2. *Department of Computer Science and Technology, Shenzhen University, Shenzhen, Guangdong* 518055, *China*)

Abstract: Scheduling algorithms make a great impact on the performance and implementation complexity of switch architecture. Traditional maximal matching (MM) algorithm cannot get a proper balance between these two factors, so in this paper we propose a new kind of Extended Maximal Matching (EMM) algorithm. By using different weight parameters, EMM algorithm can derive different kinds of algorithms. We prove that any EMM(2) algorithm with data speedup of 2 can deliver 100% throughput, and show it can also achieve almost the same delay performance as ideal Output Queueing (OQ). Furthermore, under the situation of non-speedup, through simulation we show EMM algorithms, with more than two weight parameters, can greatly increase the throughput performance of MM algorithm.

Key words: switch architecture; speedup; scheduling algorithm; maximal matching; throughput

1 引言

输出排队结构(Output Queueing, OQ)[1]是一种理想交换结构, 能够取得 100% 的吞吐量和最低的平均信元延时. 由于输出排队结构中输出队列及交换网络需 N 倍的加速比(N 为交换结构端口数), 硬件实现复杂度高、可扩展性差, 较少在高速路由器中使用. 输入输出组合排队结构(Combined Input/Output Queueing, CIOQ)[2~4]的加速比介于 1 和 N 之间, 由输入端、交换网络及输出端三部分组成(如图 1 所示). 每个输入端有一个由 N 个 VOQ(Virtual Output Queue)组成的缓存, 每个输出端有一个输出队列. 输入排队结构(Input Queueing, IQ)仅在输入端缓存数据[5,6], 交换过程无需加速比. 除了在输出端没有队列外, 其他部分都与输入输出组合排队结构相同.

后两种结构只需较低的加速比, 可扩展性好, 但需要高效的调度算法来解决信元对端口的冲突竞争. 此时调度问题可以抽象成二分图的匹配问题, 已有的调度算法可以分为几大类. 其中最大权重匹配(Maximum Weight Matching, MWM)算法被证明在没有加速比时能够取得 100% 的吞吐量[5], 但计算复杂度高, 达到 $O(N^3)$; 最大尺寸匹配(Maximum Size Matching, MSM)[5]算法的计算复杂度也达到 $O(N^{2.5})$, 因此二者都不适用于高速环境.

与前两类算法相比, 极大匹配(Maximal Matching, MM)[7~9]算法的计算复杂度仅为 $O(N\log_2 N)$, 如 PIM[8], iSLIP[7], DRRM[9]等. 其主要缺点是在非均匀业务模型下最大吞吐量只有 80%[10,11]. 研究表明使用 2 倍加速比, 极大匹配算法能够取得 100% 的吞吐量[3]和良好的延

收稿日期: 2006-05-18; 修回日期: 2007-07-12
基金项目: 国家自然科学基金(No. 60373007, 60573121); 中国-爱尔兰科学技术合作研究基金(No. CI-2003-02); 高等学校博士点基金(No. 20040003048); 清华大学 985 基金(No. JCpy2005054); 教育部培育基金(No. 705003)

时性能[12, 13]. 但 2 倍加速比的引用在提高性能的同时也增加了系统的实现复杂度. 极大权重匹配算法[6]在极大匹配算法的请求过程中考虑了权重因素, 请求信号的宽度由 1 位变成多位, 从而提高了算法的吞吐量性能. 但缺点是占用更多的通信开销, 同时在迭代过程中需要使用比较器对不同权重的请求进行仲裁, 因此实现复杂度较大.

本文提出了一类新的广义极大匹配(Extended Maximal Matching, EMM) 调度算法, 并从两方面来解决系统实现复杂度和性能之间的矛盾. (1) 提出数据加速比的概念以降低具有加速比的交换结构的实现复杂度; 提出并证明在 2 倍数据加速比下能够取得 100% 吞吐量的 EMM(2) 算法; (2) 仿真表明只要采用适当的权值参数, 广义极大匹配算法在没有任何加速比的情况下也能够取得很好的性能.

本文结构如下: 第 2 部分对交换模型进行定义; 第 3 部分提出广义极大匹配算法并给出一种具体实现; 第 4 部分对具有 2 倍数据加速比的 EMM(2) 算法的性能进行分析证明; 第 5 部分对广义极大匹配算法进行仿真分析; 最后第 6 部分总结全文.

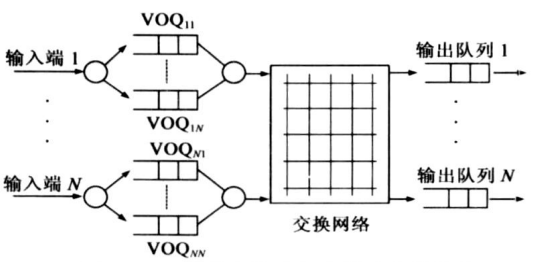


图 1 $N\times N$ 输入输出组合排队交换结构

2 交换模型

考虑图 1 所示的交换结构, 假设系统中的所有信元都是定长信元, 线速传输一个信元的时间为一个时隙. 下面对本文用到的一些符号进行定义:

VOQ_{ij}: 输入端 i 中缓存去往输出端 j 的信元的队列;

$A_{ij}(n)$: 时隙 n 结束时, 累计到达 VOQ_{ij} 的信元数目;

$Z_{ij}(n)$: 时隙 n 开始时, VOQ_{ij} 中信元的数目;

$\pi(n)$: $N\times N$ 转置矩阵, 表示时隙 n 时交换网络的连接状态: 如果输入端 i 与输出端 j 匹配, $\pi_{ij}(n)=1$, 否则 $\pi_{ij}(n)=0$.

当到达过程 $\{A_{ij}(\cdot),\ i,j=1,\ \cdots,\ N\}$ 以概率 1 满足式(1)时, 称到达过程服从强大数定律, 并称 λ_{ij} 为 VOQ_{ij} 的信元平均到达速率.

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

如果输入流量的平均到达速率满足式(2), 称为是可接纳的

$$\sum_i\lambda_{ij}\leqslant 1,\ \sum_j\lambda_{ij}\leqslant 1\qquad i,j=1,\ \cdots,N \tag{2}$$

如果在任何可接纳输入流量下, $\lim_{n\to\infty}\sum_{i,j}Z_{ij}(n)$ 都有上界, 称交换结构是稳定的, 或者说能够取得 100% 的吞吐量.

3 广义极大匹配算法

3.1 广义极大匹配(Extended Maximal Matching)

匹配: 给定一个二分图 $G=\langle V, E\rangle$, V 为顶点集合, E 为边集合; 设 $M\subseteq E$, 若 M 中任意两条边在 G 中都不相邻, 则称 M 为 G 的一个匹配.

极大匹配: 匹配 M 是二分图 G 的一个极大匹配, 当且仅当对于 G 中的任意一条边, 至少有一个端点在 M 中.

赋权二分图: 一个二分图 $G=\langle V, E\rangle$, 如果每条边都有一个权重值, 则这个二分图称为赋权二分图.

基于权重 k 的二分子图: 由赋权二分图 G 中所有顶点及权重不小于 k 的边所组成的图, 称作是图 G 的基于权重 k 的二分子图, 简记为 G_k.

投影: 边集合 $E1$ 与边集合 $E2$ 的交集称作 $E1$ 在 $E2$ 上的投影.

定义 **1** 基于权重 k 的广义极大匹配: EMM(k)

设 M 是赋权二分图 G 中一个匹配, G_k 为 G 基于权重 k 的二分子图. 如果 M 在图 G_k 边集上的投影是 G_k 的一个极大匹配, 称 M 是二分图 G 的基于权重 k 的广义极大匹配, 简记为 EMM(k).

EMM(k) 的一种非严谨但更加通俗直观的定义是: 对于权重大于等于 k 的边, 该匹配是一个极大匹配. 容易看出, EMM(1) 就是传统意义上的极大匹配算法.

定义 **2** 基于权重 $k_1, k_2, \cdots, k_n$ ($k_1<k_2<\cdots<k_n$) 的广义极大匹配: EMM($k_1, k_2, \cdots, k_n$)

对一个二分图 G, 如果一个匹配同时满足 EMM(k_1), EMM(k_2) … EMM(k_n), 称这个匹配为基于权重 $k_1, k_2, \cdots, k_n$ 的广义极大匹配, 简记为 EMM($k_1, k_2 \cdots k_n$).

图 2(a) 给出一个 3×3 输入排队结构的例子, 二分图边的权值对应 VOQ 的队列长度. 图 2(b) 给出二分图

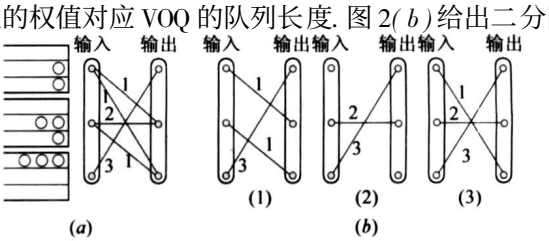


图 2 同一个二分图中的三种不同广义极大匹配

的三种不同匹配, 第一个匹配为 EMM(1, 3), 第二个匹配为 EMM(2, 3), 第三个匹配为 EMM(1, 2, 3).

3.2 广义极大匹配算法及其实现

由 3.1 中的定义可见, 广义极大匹配是多种匹配的一个集合, 不同的权值参数可以派生出不同种类的匹配. 我们将实现 EMM(L_1, L_2, …, L_t) 匹配的算法称为 EMM(L_1, L_2, …, L_t) 算法①.

下面在 iSLIP 算法基础上, 给出 EMM(L_1, L_2, …, L_t) 算法的一种具体实现: MWiSLIP(L_1, L_2, …, L_t) 算法.

算法执行分为 t 个阶段: 第一阶段完成长度大于等于 L_t 的 VOQ 的调度, 得到一个 EMM(L_t) 匹配; 依此类推在第 m 个阶段完成长度在[L_{t-m+1}, L_{t-m+2}]区间的 VOQ 的调度, 得到一个 EMM(L_{t-m+1}, …, L_t) 匹配; 最终在 t 个阶段后得到一个 EMM(L_1, L_2, …, L_t) 匹配.

每个输入端 i 为每个阶段 m 维护一个指针 $AP_{i,m}$, 每个输出端 j 也为每个阶段 m 维护一个指针 $GP_{i,m}$. 所有指针的初始状态都指向第一个输出(或输入)端口.

每个阶段的调度过程由输入输出间的多次迭代组成, 每次迭代(设第 m 阶段)包括三个步骤:

步骤 1　请求. 每个未匹配的输入端 i, 检查是否存在长度位于[L_{t-m+1}, L_{t-m+2})区间内的 VOQ(当 $m=1$ 时, 长度区间为[L_t, +∞)), 并向对应于这些 VOQ 的输出端发送请求.

步骤 2　响应. 每个未匹配的输出端 j, 从所有请求中选取优先级最高的(从指针 $GP_{i,m}$所指位置开始第一个遇到的请求)进行响应. 仅当该响应在本阶段第一次迭代中就被确认时, $GP_{i,m}$更新指向被响应输入端的下一个端口.

步骤 3　确认. 每个未匹配的输入端 i, 从所有响应中选取优先级最高的(从指针 $AP_{i,m}$所指位置开始第一个遇到的响应)进行确认. 仅当确认发生在本阶段第一次迭代中时, $AP_{i,m}$更新指向被确认输出端的下一个端口.

MWiSLIP 算法可以看成是由多个 iSILP 算法串行叠加而成, 其算法复杂度主要取决于调度阶段的数量(即权值参数的数量)和每个阶段中迭代的次数. iSLIP 算法的平均迭代次数为 $O(\log_2 N)$, 因此具有 t 个权值参数的 MWiSLIP 算法平均迭代次数为 $O(t\cdot\log_2 N)$, 可见仅当其具有较少的权重参数时才有实用价值.

4　利用广义极大匹配算法降低加速比

从理论上证明取得 100% 吞吐量的极大匹配算法需要 2 倍加速比[3]. 能否找到比 2 倍加速比实现更简单, 同时又能从理论上保证 100% 吞吐量的(广义)极大匹配算法?

4.1 数据加速比

具有 s 倍加速比的交换结构在每个时隙内要完成 s 次调度和 s 次交换. 如果在一个时隙内只进行一次调度, 但是允许交换 s 次, 无疑会简化交换系统的实现, 我们将这种交换结构称作具有 s 倍数据加速比.

与具有 s 倍加速比的交换结构相比, 具有 s 倍数据加速比的交换结构有如下特点:

1. 从数据的传输速率来说, 两种结构相同;

2. 从交换网络重配置频率来说, 前者是后者的 s 倍;

3. 对于调度算法的单次执行时间来说, 前者的调度算法执行时间是后者的 1/s; 当采用复杂度相似的调度算法时, 后者更容易实现;

那么在具有 2 倍数据加速比但没有调度加速比的交换结构中, 极大匹配算法 EMM(1)能否保证 100% 的吞吐量?

考虑图 3 所示的具有 2 倍数据加速比的 3×3 交换结构, 设当前时隙为 t, 交换结构中信元的到达过程为:

当 t mod 3= 0 时, VOQ_{12}, VOQ_{22}以及 VOQ_{32}有信元到达;

当 t mod 3< 0 或> 0 时, VOQ_{21}和 VOQ_{33}有信元到达.

显然输入流量满足可接纳条件. 接下来给出交换结构中使用的调度算法:

当 t mod 3= 0 时, 输入端 1 与输出端 2 匹配;

当 t mod 3< 0 或> 0 时, 输入端 2 与输出端 1 匹配, 输入端 3 与输出端 3 匹配.

图 3 给出交换结构从时隙 0 开始的运行情况, 可见上面的调度算法是一个 EMM(1) 算法. 由于 VOQ_{22}和 VOQ_{32}自始至终都得不到服务, 随着时间的推进, 最终导致系统不稳定. 从这个反例可知 2 倍数据加速比的 EMM(1)算法并不一定能保证 100% 的吞吐量. 但在引入广义极大匹配算法的概念之后, 我们可以得到定理 1.

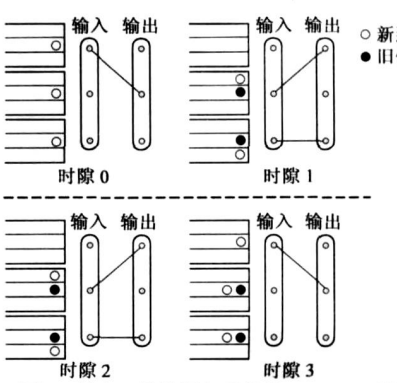


图 3　具有 2 倍数据加速比的 EMM(1)算法示例

① 广义极大匹配中的权值参数可采用不同的指标, 如队列长度和队头信元等待时间; 本文只讨论队列长度做为权值参数的情况.

4.2 具有 2 倍数据加速比的 EMM(2) 算法

定理 **1** 当输入流量可接纳并满足强大数定律时, 任何一个 EMM(2) 算法在 2 倍数据加速比下都能够取得 100% 的吞吐量.

定理 1 的证明思路是首先给出一个能够模拟 2 倍数据加速比交换结构的并行交换结构; 通过证明该并行交换结构能够取得 100% 的吞吐量间接证明定理 1.

图 4 是一个具有 M 个交换平面的 $N\times N$ 并行交换结构, 由三部分组成: N 个解复用器、M 个交换平面和 N 个复用器. 在解复用器、交换平面、复用器的实现中采用不同的算法, 能够得到不同的并行交换结构.

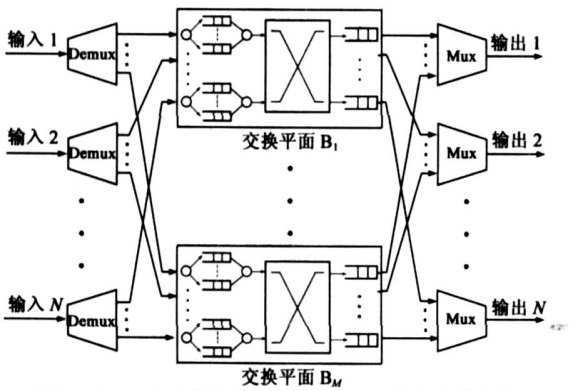


图 4 由 M 个子交换平面组成的 $N\times N$ 并行交换结构

假设存在一个具有 s 倍数据加速比的虚拟交换结构 A; B 为拥有 s 个交换平面的并行交换结构, B_1, B_2, …, B_s 为 B 中 s 个交换平面. 下面给出一种能够用并行结构 B 模拟①交换结构 A 的模拟算法(EA 算法). 为便于描述, 给每个用到的符号都增加一个上标, 例如交换结构 A 中的 VOQ_{ij} 用 VOQ_{ij}^{A} 表示.

设当前时隙为 n, 模拟算法 EA 由三部分组成:

(1) 解复用原则: 对 B 中每个输入端 i, 当有一个新信元 c 到达时(假设去往输出端 j), 如果在虚拟交换结构 A 中 $Z_{ij}^{A}(n)=0$, 将信元 c 送往交换平面 B_1(存入 $VOQ_{ij}^{B_1}$); 否则按照轮询原则(按照 B_1, B_2, …, B_s 的顺序), 将信元 c 送往其中一个交换平面;

(2) 交换平面调度原则: B 中所有交换平面都采用与 A 相同的匹配, 即 $\pi^{B_1}(n)=\cdots=\pi^{B_s}(n)=\pi^{A}(n)$;

(3) 复用原则: 对于每个输出端 j, 如果在虚拟交换结构 A 中有一个信元 c 离开, 在 B 中复用器 j 从 s 个交换平面的输出队列中挑选出信元 c, 将其送往输出链路.

定理 **2** 任何一个具有 s 倍数据加速比, 交换网络连接序列为 $\pi(\cdot)\{\pi(0),\pi(1),\cdots\}$ 的交换结构, 都可以通过 EA 算法用一个有 s 个交换平面的并行结构来模拟, 这 s 个交换平面没有加速比, 并且交换网络连接序列仍然为 $\pi(\cdot)\{\pi(0),\pi(1),\cdots\}$.

定理 2 的详细证明过程参见附录 1.

仅考虑数据加速比 $s=2$ 的情形. 令 A 表示数据加速比为 2 的交换结构, B 为使用 EA 算法模拟 A 的并行结构, B_1 和 B_2 是 B 中的两个交换平面. 下面给出几个引理:

引理 **1** 如果交换结构 A 中信元到达过程可接纳并满足强大数定律, 交换平面 B_2 中信元到达过程满足下式:

$$\sum_i \lambda_{ij}^{B_2}\leqslant\frac{1}{2},\quad \sum_j \lambda_{ij}^{B_2}\leqslant\frac{1}{2}\qquad i,j=1,\cdots,N \tag{3}$$

证明: 根据 EA 算法解复用原则, 当 $s=2$ 时 B_1 和 B_2 中的到达过程满足:

$$\lambda_{ij}^{B_1}\geqslant\lambda_{ij}^{B_2}$$

由于 $\lambda_{ij}^{B_1}+\lambda_{ij}^{B_2}=\lambda_{ij}^{A}$, 可得

$$\lambda_{ij}^{B_2}\leqslant\frac{1}{2}\lambda_{ij}^{A} \tag{4}$$

根据可接纳和强大数定律的定义, 有下式成立

$$\sum_i \lambda_{ij}^{A}\leqslant 1,\quad \sum_j \lambda_{ij}^{A}\leqslant 1 \tag{5}$$

将式(4)带入式(5), 可以得到式(3). 引理得证.

引理 **2** 一个匹配 π 在交换结构 A 中是 EMM(2), 当且仅当 π 在交换平面 B_2 中是 EMM(1).

证明: 根据 EA 算法, 任意时隙, 集合 $\{VOQ_{ij}^{A}(n)\mid Z_{ij}^{A}(n)\geqslant 2,\ i,j=1,\cdots,N\}$ 中的元素与集合 $\{VOQ_{ij}^{B_2}(n)\mid Z_{ij}^{B_2}(n)\geqslant 1,\ i,j=1,\cdots,N\}$ 中的元素一定一一对应. 因此对于 A 中的 EMM(2) 算法一定是 B 中的 EMM(1) 算法, 反之亦然. 引理得证.

引理 **3** 在任意时隙 n, 交换平面 B_1 输入端信元的数目一定大于交换平面 B_2 输入端信元的数目, 并且信元数目之差有上界, 满足下式:

$$0\leqslant\sum_{i,j}Z_{ij}^{B_1}(n)-\sum_{i,j}Z_{ij}^{B_2}(n)\leqslant N^2 \tag{6}$$

证明: 根据 EA 算法中的解复用原则和交换平面调度原则, 在任意时隙 n, 都有下式成立

$$0\leqslant Z_{ij}^{B_1}(n)-Z_{ij}^{B_2}(n)\leqslant 1,\quad i,j=1,\cdots,N \tag{7}$$

将式(7)对所有 i,j 求和, 可得到式(6), 引理得证.

引理 **4** 当到达过程满足式(8)时, 任何无加速比的 EMM(1) 算法都能取得 100% 的吞吐量, 且交换结构稳定.

$$\sum_i \lambda_{ij}\leqslant\frac{1}{2},\quad \sum_j \lambda_{ij}\leqslant\frac{1}{2}\qquad i,j=1,\cdots,N \tag{8}$$

证明: 我们利用文献[3]中的流量模型方法证明.

① 如果每个信元在具有相同输入的两个交换结构中的离开时间完全一致, 则称其中一个交换结构能够模拟另一个交换结构.

令 $L_i(n)=\sum_j Z_{ij}(n)$, $M_j(n)=\sum_i Z_{ij}(n)$, 并令

$$C_{ij}(n)=L_i(n)+M_j(n) \tag{9}$$

令 $C'(n)$ 表示 C 在时隙 n 时的导数.

由文献[3]可知当

$$Z_{ij}(n)>0 \text{ 时, } C'_{ij}(n)\leqslant \sum_k \lambda_{ik}+\sum_k \lambda_{kj}-1 \tag{10}$$

将式(8)带入式(10), 可得

$$\text{当 } Z_{ij}(n)>0 \text{ 时, } C'_{ij}(n)\leqslant 0 \tag{11}$$

令 $f(n)$ 表示 $Z(n)$ 和 $C(n)$ 的笛卡儿乘积, 即

$$f(n)=\langle Z(n), C(n)\rangle=\sum_{i,j} Z_{ij}(n)C_{ij}(n)$$

$$=\sum_{i,j,k}(Z_{ij}(n)Z_{ik}(n)+Z_{ij}(n)Z_{kj}(n))$$

令 $f'(n)$ 表示 f 在时隙 n 时的导数, 即

$$f'(n)=\sum_{i,j,k}Z'_{ij}(n)Z_{ik}(n)+\sum_{i,j,k}Z_{ij}(n)Z'_{ik}(n)$$

$$+\sum_{i,j,k}Z'_{ij}(n)Z_{kj}(n)+\sum_{i,j,k}Z_{ij}(n)Z'_{kj}(n)$$

$$=2\sum_{i,j}Z_{ij}(n)C'_{ij}(n)$$

根据式(11)可得

当 $Z_{ij}(n)>0$ 时, $f'(n)\leqslant 0$

因此当 $f(n)>0$ 时, $f'(n)\leqslant 0$ (12)

由于 $f(0)=0$, 根据文献[3]中引理 1, 式(12)说明对于几乎所有 n, $f(n)$ 都等于 0, 也即 $Z(n)=0$. 因此系统稳定, 算法能够取得 100% 的吞吐量. 引理得证.

下面对定理 1 进行证明:

令 $\pi(\cdot)$ 表示交换结构 A 中由任意一个 EMM(2)算法所得到的交换网络连接序列, 根据引理 2, 可知在交换平面 B_2 中 $\pi(\cdot)$ 是一个满足 EMM(1)的交换网络连接序列.

根据引理 1, 当交换结构 A 中的到达过程可接纳并且服从强大数定律时, 交换平面 B_2 中的到达过程满足式(3). 结合引理 4 可知交换平面 B_2 在 $\pi(\cdot)$ 连接序列下是稳定的, 即 $\lim_{n\to\infty}\sum_{i,j}Z^{B_2}_{ij}$ 有界.

将 $\lim_{n\to\infty}\sum_{i,j}Z^{B_2}_{ij}(n)$ 带入式(6), 可知 $\lim_{n\to\infty}\sum_{i,j}Z^{B_1}_{ij}(n)$ 同样有界. 由此可以得到, B_1 交换平面与 B_2 交换平面都是稳定的, 因此并行交换结构 B 是稳定的.

由于并行交换结构 B 完全模拟交换结构 A, 因此交换结构 A 也是稳定的, 说明任意一个 EMM(2)算法在交换结构 A 中都能够取得 100% 的吞吐量. 定理得证.

定理 1 为具有 2 倍数据加速比的交换结构调度算法设计提供了理论依据, 表明只要是满足 EMM(2)条件的算法, 即无论是 EMM(2), EMM(1, 2) 或是 EMM(2, 3), 都能够取得 100% 的吞吐量.

5　仿真实验

本节对具有 2 倍数据加速比和无加速比的广义极大匹配算法进行仿真, 仿真中采用 MWiSLIP 算法做为广义极大匹配算法的具体实现.

5.1　算法参数与仿真环境设置

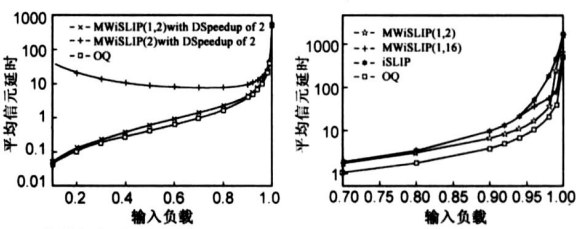
MWiSLIP 算法复杂度与权值参数个数有关, 仿真中只考虑权值参数个数较少的情况, 将 MWiSLIP 算法每个阶段允许的最大迭代次数设为 4, 整个算法的最大迭代次数设为 6. 对于普通 iSLIP 算法, 最大迭代次数设为 4.

交换结构端口数设为 8, 输入业务采用三种不同的模型: 贝努利均匀模型、非均匀模型, 以及突发模型.

5.2　贝努利均匀模型

(1)　2 倍数据加速比

图 5(a)为具有 2 倍数据加速比的 MWiSLIP(1, 2)、MWiSLIP(2)算法, 以及 OQ 交换结构在贝努利均匀模型下的延时性能, 可见所有算法都能取得 100% 的吞吐量. OQ 始终具有最低的平均信元延时. MWiSLIP(2)算法在低负载时的延时性能很差, 这是由于当负载很低时, 输入端 VOQ 的长度都很短, 而当 VOQ 长度为 1 时, 一直要等到长度达到 2 之后才会被 MWiSLIP(2)算法调度. 而与之相对应的 MWiSLIP(1, 2)算法在 2 倍数据加速比下能够很好地工作, 并取得和 OQ 交换结构非常接近的延时性能.



(a) 2 倍数据加速比(图中 DSpeedup 指数据加速比)　　(b) 无加速比

图 5　广义极大匹配算法在贝努利均匀业务下的延时性能

(2)　无加速比

无加速比时 MWiSLIP(1, 2)算法、MWiSLIP(1, 16)算法、iSLIP 算法以及 OQ 交换结构在贝努利均匀模型下的延时性能如图 5(b)所示(只画出负载从 0.7 到 1.0 区间的延时性能). 两种 MWiSLIP 算法与 iSLIP 算法在低负载时区别不是很大, 但在负载高于 0.9 时, MWiSLIP 算法较 iSLIP 算法性能有明显优势.

5.3　贝努利非均匀模型

令 ρ 表示每个输入端的总负载, $\rho_{i,j}$ 表示从输入端 i 到达去往输出端 j 的业务负载, w 表示非均匀系数, 则非均匀模型中 $\rho_{i,j}$ 的定义如式(13)所示

$$\rho_{i,j}=\begin{cases}\rho\left(w+\dfrac{1-w}{N}\right), & \text{当 } i=j\\ \rho\dfrac{1-w}{N}, & \text{当 } i\neq j\end{cases} \tag{13}$$

文献[10, 11]研究表明 iSLIP 等算法在 $w=0.6$ 时吞

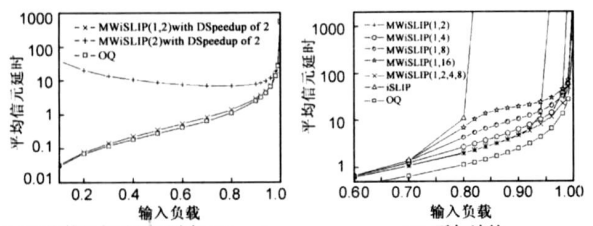
吐量最多只能达到 80% 左右. 在这里为了更好地观察算法之间的性能差别, 令 $w = 0.6$.

(1) 2 倍数据加速比

2 倍数据加速比下 MWiSLIP(1, 2) 和 MWiSLIP(2) 算法的延时性能如图 6(a) 所示. 可见 OQ 交换结构仍具有最低的平均延时, MWiSLIP(1, 2) 算法的延时性能非常接近于 OQ, 而 MWiSLIP(2) 算法在低负载时延时仍然较大.

(2) 无加速比

图 6(b) 是几种算法在无加速比时的延时性能曲线, 可见 iSLIP 算法只能取得 80% 的吞吐量, MWiSILP(1, 2) 能够取得 94% 的吞吐量, MWiSLIP(1, 4) 能够取得 98% 的吞吐量, MWiSLIP(1, 8) 和 MWiSLIP(1, 16) 算法能够取得接近 100% 的吞吐量.



(a) 2 倍数据加速比 (图中 DSpeedup 指数据加速比) (b) 无加速比

图 6 广义极大匹配算法在贝努利非均匀业务下的延时性能

图中不同权值参数的 MWiSLIP 算法延时曲线有很强的规律性. 总的来说, 具有 2 个权值参数的 MWiSLIP 算法, 当第一个参数取 1 时, 第二个参数越大, 在低负载时的延时性能越差, 在高负载时的延时性能越好; 反之在低负载时的延时性能越好, 但在高负载时性能越差.

上述试验现象是因为低负载时输入端 VOQ 的平均长度都很短. 若第二个参数取值很大, 在算法执行时的作用发挥不出来 (因为没有长度超过这个参数值的队列), 此时的 MWiSLIP 算法就退化成了普通的 iSLIP 算法; 而当第二个参数取值较小时, 可以在低负载时更早地发挥作用.

在高负载时, 队列平均长度都很长, 可以认为所有算法的第二个参数都发挥了作用. 这时第二个参数值取得越大, 算法就能更好地先服务权重大的边, 得到的匹配权值也就越大, 根据文献[14] 中的结论, 平均信元延时就更小.

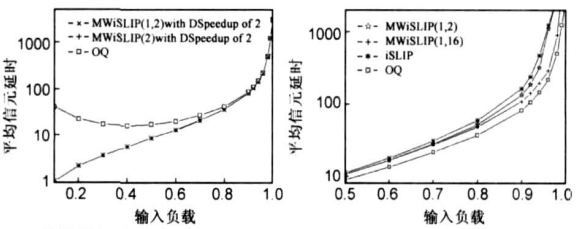
因此要想同时在低负载和高负载下取得较好的性能, 可以采用具有更多权值参数的 MWiSLIP 算法, 例如图 6(b) 中的 MWiSLIP(1, 2, 4, 8) 算法.

5.4 突发模型

令突发长度 $b = 10$, 每一串突发信元到达 N 个输出端口的概率均等.

(1) 2 倍数据加速比

图 7(a) 给出了 2 倍数据加速比的 MWiSLIP(1, 2)、MWiSLIP(2) 算法和 OQ 交换结构在突发业务下的延时性能. MWiSLIP(1, 2) 算法与 OQ 的延时性能几乎完全相同.



(a) 2 倍数据加速比 (图中 DSpeedup 指数据加速比) (b) 无加速比

图 7 广义极大匹配算法在突发业务下的延时性能

(2) 无加速比

无加速比下 MWiSLIP(1, 2)、MWiSLIP(1, 16) 算法, iSLIP 算法以及 OQ 交换结构的延时性能如图 7(b) 所示. 低负载时 MWiSLIP(1, 16) 算法的延时性能与 MWiSLIP(1, 2) 和 iSLIP 算法相当, 而在中高负载时则要好于它们.

6 结论

本文结合最大权重匹配算法和极大匹配算法的特点, 提出了一类广义极大匹配算法, 并给出了一种具体实现: MWiSLIP. 通过采用不同的权值参数, 广义极大匹配算法可以派生出不同子类的调度算法.

基于广义极大匹配算法, 文章首先证明了任何 EMM(2) 算法只需要 2 倍数据加速比就能够从理论上保证 100% 吞吐量, 降低了原先 EMM(1) 算法同时需要 2 倍数据加速比和 2 倍调度加速比的要求. 仿真结果表明 MWiSLIP(1, 2) 算法在 2 倍数据加速比下能够取得接近 OQ 的延时性能.

本文同时还在没有任何加速比的条件下对具有两个权值参数的 MWiSLIP 算法进行了仿真. 结果显示其信元延时性能要大大好于普通的 iSLIP 算法.

附录 1 定理 2 的证明

一、 准备工作

令具有 s 倍数据加速比的交换结构为 A, 并行交换结构为 B, B 的 s 个交换平面分别为 $B_1, B_2, \cdots, B_s$. 并行结构的运行满足 EA 算法的要求.

给每个交换网络输入端的信元 c 定义一个序列号 $sn(c, n)$, 表示时隙 n 开始时信元 c 在交换结构 A 的 VOQ 中的位置. 当信元 c 处于 VOQ 队头时序列号为 0, 处于 VOQ 队列中第二个位置时序列号为 1, 依此类推.

在此有 3 点需要强调: (1) 仅在每个时隙的到达阶段中, 每个信元的序列号才重新计算, 在每个时隙的调度交换过程中, 虽然信元在 VOQ 中的位置可能发生变

化, 但是序列号并不改变; (2)序列号只对输入端信元有意义, 对于输出端信元, 序列号并没有定义. (3)不管在交换结构 A, 还是在并行结构 B 中, 序列号都根据交换结构 A 进行计算.

根据序列号的不同, 在每个时隙将交换结构输入端的信元分成 s 类, 令 $class(c, n) = sn(c, n) \bmod s$, 代表信元 c 在时隙 n 的分类号.

在数据加速比为 s 的交换结构中, 有以下性质.

性质 1: 每个信元在到达交换结构输入端之后, 其 *class* 取值始终保持不变.

根据数据加速比的定义, 可以很容易地得到性质 1, 因此每个信元的分类号 $class(c, n)$ 可以简写为 $class(c)$.

性质 2: 交换结构 A 每个 VOQ 中任意 s 个连续信元都属于不同的分类, 并且分类号相邻.

性质 3: 当并行交换结构 B 利用 EA 算法模拟交换结构 A 时, 交换平面 B_1 中的所有信元都属于第 0 类, 交换平面 B_2 中的信元都属于第 1 类, …, 交换平面 B_s 中的信元都属于第 $s-1$ 类.

证明: 设在时隙 n 之前性质 3 成立,

在时隙 n, 对于任意输入端 i, 如果有新信元 c 到达 (不失一般性, 假设去往输出端 j), 分两种情况考虑:

(1) $Z_{ij}^{A}(n) = 0$: 显然信元 c 的序列号为 0, 属于第 0 类. 根据 EA 算法的解复用规则, 信元被送往交换平面 B_1. 因此时隙 n 中性质 3 仍然成立;

(2) $Z_{ij}^{A}(n) > 0$: 假设与信元 c 具有相同源/目的端口的上一个信元属于分类 k 并去往了交换平面 B_{k+1}. 由性质 2 可知, 信元 c 所属类别为 $(k+1) \bmod s$, 而根据 EA 算法解复用原则信元将送往交换平面 $B_{(k+1) \bmod s+1}$. 由此可知在时隙 n 中性质 3 仍然成立.

结合(1), (2), 可以得到性质 3.

二、 定理 2 的证明

证明定理 2 的关键是证明任何一个信元 c 在并行结构 B 中到达输出端的时间与在交换结构 A 中到达输出端的时间一致(并行结构输出端的复用原则在这里暂不需要考虑, 只要信元按时到达输出端自然能够找到合适的复用原则使其按时离开并行结构).

初始条件: 显然在第 0 个时隙, 上述结论正确.

归纳假设: 假设在时隙 n 之前, 任何一个信元在并行交换结构 B 中到达输出端的时间与在交换结构 A 中到达输出端的时间一致.

在时隙 n 新信元到达阶段完成之后, 考虑交换结构 A 中任意一个输出端 j, 不失一般性, 假设输入端 i 与输出端 j 相匹配, 即 $\pi_{ij}^{A}(n) = 1$. 令 $X_{ij}^{A}(n)$ 表示此时交换结构 A 中 VOQ_{ij} 的长度.

(1)如果 $X_{ij}^{A}(n) = 0$, 在交换结构 A 中没有信元送往输出队列 j; 同样在并行交换结构 B 中也没有信元被送往输出队列 j;

(2)如果 $0 < X_{ij}^{A}(n) < s$, A 中 VOQ_{ij}^{A} 中的所有 $X_{ij}^{A}(n)$ 个信元都被送到输出队列 j, 根据性质 2 这 $X_{ij}^{A}(n)$ 个信元都属于不同的分类, 又根据性质 3 可知这 $X_{ij}^{A}(n)$ 个信元都在 B 中不同的交换平面中. 根据 EA 算法的交换平面调度原则, B 中所有交换平面都采用与 A 相同的匹配, 即 $\pi_{ij}^{B_1}(n) = \cdots = \pi_{ij}^{B_s}(n) = \pi_{ij}^{A}(n) = 1$, 因此这 $x_{ij}^{A}(n)$ 个信元在时隙 n 都被送往各自所在交换平面的输出队列;

(3)如果 $X_{ij}^{A}(n) \geqslant s$, A 中 VOQ_{ij}^{A} 中队头 s 个信元都会被送到输出队列 j, 根据性质 2 这 s 个信元都属于不同的分类, 因此这 s 个信元都在 B 中不同的交换平面中. 根据 EA 算法交换平面调度原则, B 中所有交换平面都采用与 A 相同的匹配, 因此所有 $X_{ij}^{A}(n)$ 个信元在时隙 n 中都被送往各自所在交换平面的输出队列中.

根据归纳假设, 定理 2 得证.

参考文献:

[1] Karol M, Hluchyj M, Morgan S. Input versus output queueing on a space-division packet switch[J]. IEEE Transactions on Communications, 1987, 35(12): 1347–1356.

[2] Chuang S, Goel A, McKeown N, Prabhakar B. Matching output queueing with a combined input/output-queued switch[J]. IEEE J Select Areas Commun, 1999, 17(6): 1030–1039.

[3] Dai J G, Prabhakar B. The throughput of data switches with and without speedup[A]. IEEE INFOCOM 2000[C]. Tel Aviv, Israel: IEEE Computer and Communications Societies, 2000. 556–564.

[4] Prabhakar B, McKeown N. On the speedup required for combined input and output queued switching[J]. Automatica, 1999, 35(12): 1909–1920.

[5] McKeown N, Mekkittikul A, Anantharam V, Walrand J. Achieving 100% throughput in an input-queued switch[J]. IEEE Transactions on Communications, 1999, 47(8): 1260–1267.

[6] McKeown N. Scheduling algorithms for input-queued cell switches[D]. Ph. D. dissertation Univ California, Berkeley, 1995.

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

[8] Anderson T E, Owicki S S, Saxe J B, Thacker C P. High-speed switch scheduling for local-area networks[J]. ACM Transactions on Computer Systems, 1993, 11(4): 319–352.

[9] Li Y, Panwar S, Chao H J. On the performance of a dual

round-robin switch[A]. IEEE INFOCOM 2001[C]. Anchorage, USA: IEEE Computer and Communications Societies, 2001. 1688-1697.

[10] Rojas-Cessa R, Oki E, Jing Z, Chao H J. CIXB-1: combined input-one-cell-crosspoint buffered switch[A]. IEEE HPSR 2001[C]. Dallas, USA: IEEE Communications Society, 2001. 324-329.

[11] Rojas-Cessa R, Oki E, Chao H J. CIXOB-k: combined input-crosspoint-output buffered packet switch[A]. IEEE GLOBECOM 2001[C]. San Antonio, USA: IEEE Communications Society, 2001. 2654-2660.

[12] Benson K D. Throughput of crossbar switches using maximal matching algorithms[A]. IEEE ICC 2002[C]. New York, USA: IEEE Communications Society, 2002. 2373-2378.

[13] Shah D. Maximal matching scheduling is good enough[A]. IEEE GLOBECOM 2003[C]. San Francisco, USA: IEEE Communications Society, 2003. 3009-3013.

[14] Shah D, Kopikare M. Delay bounds for approximate maximum weight matching algorithms for input queued switches[A]. IEEE INFOCOM 2002[C]. New York, USA: IEEE Computer and Communications Societies, 2002. 1024-1029.

作者简介:

徐 扬 男, 1979 年 5 月出生于甘肃省兰州市. 2001 年 7 月毕业于北京邮电大学计算机科学与技术学院, 获工学学士学位; 2007 年 1 月毕业于清华大学计算机系, 获工学博士学位. 主要研究方向: 高速路由器体系结构、交换结构与调度算法、高速网络安全等.

E-mail: xy01@mails.tsinghua.edu.cn

唐 毅 男, 1983 年 3 月生于湖北武汉. 在清华大学计算机系攻读博士学位. 主要研究方向为流分类查找、以及高速网络安全.

E-mail: tangy05@mails.tsinghua.edu.cn

文振 男, 1962 年 3 月生于广东信宜, 清华大学计算机科学与技术专业毕业, 获工学硕士学位. 现为深圳大学信息工程学院计算机系副教授. 主要研究领域为多媒体信息处理技术、流媒体技术、图像处理技术.

刘 斌 男, 生于山东, 清华大学计算机系教授, 博士生导师. 主要研究领域为交换技术、网络处理器、流量工程、高速网络安全等.