

一种端口速率可变的共享存储器交换¹

汪卫章 葛 宁 冯重熙

(清华大学电子工程系 微波与数字通信国家重点实验室 北京 100084)

摘 要 针对局域网和城域网中的多种数据传输速率结构, 该文提出了一种共享存储器交换结构, 在普通共享存储器交换结构的基础上, 对支持可变的端口速率, 以及支持变长数据包交换进行了改进, 所提出的交换结构还具有自同步特点, 即各输入输出端口之间不需要全局同步; 同时还考虑了对变长数据包的队列管理。

关键词 共享存储交换, 缓冲区管理, 队列管理

中图分类号 TN919.3

1 引 言

在高速路由器中, 共享存储器交换和输入排队的全交叉点交换结构 (crossbar) 获得了广泛的应用^[1,2], 如 Cisco 等公司的路由器采取了全交叉点交换结构。输入排队的全交叉点交换的优点是交换核心的速率与端口速率相同, 但存储器的利用效率不高。Juniper 等公司的路由器采取了共享存储器交换结构^[3-7], 共享存储器交换的优点是存储器利用效率很高, 但核心速率远大于各端口线速, 对于存储器运行速度要求很高。

SDH(Synchronous Digital Hierarchy) 传输技术和以太网 (Ethernet) 传输技术在构建高性能数据网络中得到了最广泛的应用。SDH 的传输速率等级为 STM-1(155Mbit/s)、STM-4(622Mbit/s)、STM-16(2.5Gbit/s)、STM-64(10Gbit/s)。以太网的传输速率等级为 10M、100M、1G、10G。只有在 10G 接口上, SDH 才和以太网的传输速率一致。而在高性能的路由器、交换机中, 往往同时提供这两种标准数据接口, 以及提供同一标准下的不同等级接口。在设计交换结构时, 一般是设计具有相同传输速率的输入输出端口, 然后再对不同的接口速率进行适配。例如在以 622Mbit/s 为基础的交流矩阵中, 可以将 4 个 622M 端口汇聚成一个 2.5G 端口, 也可以将一个 622M 端口扩散成 4 个 155M 端口。但是在汇聚和扩散适配过程中会造成交换带宽的浪费, 以及加重了设备的复杂性。如将 155M 适配成 100M 端口时, 会造成 50% 的交换带宽浪费。本文提出了一种端口速率可变的共享存储器交换结构, 只要交换结构中的存储器带宽略大于各输入输出端口速率之和, 就能保证各端口的线速运行, 可用于交换各种速率等级的数据。

本文的结构如下: 第 2 节介绍端口速率固定的共享存储器交换结构, 以及对不同速率的适配方法。第 3 节提出新的端口速率可变的共享存储器交换结构并给出了可实现条件。第 4 节讨论了支持变长包的存储器管理和队列管理。

2 端口速率固定的共享存储器交换结构及其扩展

2.1 基本结构

端口速率固定的共享存储器交换结构如图 1 所示, 主要由复用器、解复用器、存储器、交换控制器 4 个部分组成。复用器将 N 路端口低速数据按时分复用方式复用为一高速数据, 解复用器的功能与之相反, 将一路高速数据还原为 N 路输出。存储器的功能是存储数据包, 交换控制器的功能可进一步分为时隙分配单元、存储器管理单元、队列管理单元、输出调度单元。

¹ 2001-09-18 收到, 2002-04-29 改回
国家自然科学基金资助项目 (69896240)

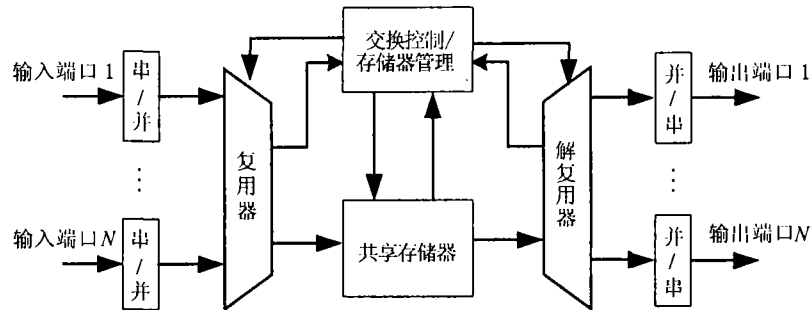


图 1 共享存储器交换的基本结构

时隙分配单元管理输入输出端口访问共享存储器的顺序，一般采用固定循环、读写交替。存储器管理单元负责存储器的访问，为了管理方便，存储器以固定大小的信元来存储数据包。数据包在进入交换矩阵之前被切分成信元，在离开交换矩阵之后重新组装成数据包。数据包在与信元的转换过程中会损失一定的带宽，文献 [8] 中给出了在 Internet 网中，当 IP 数据包以 64byte 的形式存储时，会损失 7% 的带宽。在交换矩阵的设计时，如果能将交换矩阵的总带宽（存储器的访问带宽）高于输入输出端口的带宽之和（>7%），基本上能弥补损失带宽。队列管理单元负责信元在存储器中的组织形式，可分为链表和非链表两种组织形式，主要是根据数据流信息、输入端口、输出端口、服务质量要求中的一种或多种信息将数据包组织成不同的队列。输出调度单元负责解决多个队列去往同一端口时的竞争，从而完成对数据流的带宽、时延等服务质量保证。在本文中仅关注单播的交换，没有考虑多播，有关多播的实现可见文献 [9-11]。

在图 1 中，共享存储器的总带宽等于各输入端口与输出端口的带宽之和，设输入输出端口的带宽为 B_l (运行)，则共享存储器的带宽为 $B_m = 2N \times B_l$ 。输入端口与输出端口的速率是严格相等的，各端口以时分复用的方式访问共享存储器，输入端口将数据包写入存储器，输出端口将数据包读出存储器。

2.2 基本结构中不同速率端口的适配

在实际应用中，在同一交换机或路由器中，往往必须接入不同数据速率的线路，这时就必须对端口与线路进行适配。适配有两种，一是从交换矩阵端口到高速线路的聚合，二是从交换矩阵端口到低速线路的扩展。图 2 中给出了对共享存储器交换矩阵聚合和扩展的示意图。在输入方，将低速的线路 1 至 M 复用到交换矩阵的输入端口 1，将高速的线路 P 解复用到交换矩阵的输入端口 $(N - M + 1)$ 至 N ；在输出方与之相反。

在图 2 中，在输出端，从高速端口 1 到低速线路 1 至 M 的扩展中，为了避免在解复用器 1 中使用较大的缓冲区，在交换矩阵内部的队列管理和输出调度上，要对每个输出线路维持单独的一个队列（在这里，我们对每个输出线路仅考虑一个队列，当支持优先级时，相应的要维持多个队列）；同时在输出调度时，必须保证在一定的时间范围内输出的去往各个线路的数据流速率不超过线路的物理速率，否则会造成该端口去往某一低速线路的数据量过多而导致缓冲区溢出。而在从端口 $(N - M + 1)$ 至 N 到高速线路 P 的输入聚合中，对于属于同一线路 P 的 M 个端口，可以共用一个队列 P 。所以在交换矩阵的端口适配中，交换矩阵的内部的队列管理和输出调度也必须作出相应的变化来适应这种适配。

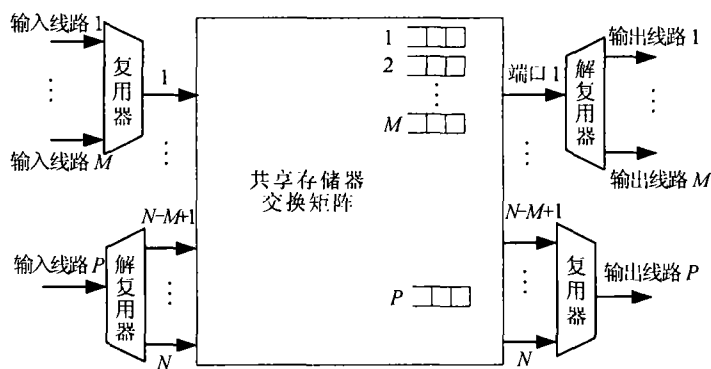


图 2 基本交换矩阵的端口适配

3 端口速率可变的共享存储器交换

3.1 基本结构

我们的目标是在图 1 中共享存储器的基本结构上做出两点改进，一是在输入输出接口上支持变长数据包；二是端口速率能在一定范围内可变。支持变长数据包能使外围输入输出接口处理变得更简单，省却了对数据包的切分和组装功能；端口速率可变使交换核心的运行与外部接口的速率无关，使之适合各种交换速率，有效地利用交换带宽。

改进后的共享存储器交换如图 3 所示，从结构上看，我们做了两点改变，一是在每个输入输出端口增加了一个异步 FIFO(先进先出存储器)；二是改变了以往的时分复用方式，在输入端与输出端各增加了一个竞争器，输入端对应共享存储器的写竞争，输出端对应共享存储器的读竞争。在存储器的组织方式上，变长的数据包仍以固定长度的信元存放(在此文中我们假设为 64byte)，数据包尾部不足一个信元长的占用整个信元的空间。对于存储器的访问仍以信元为基本单位，一次读写整个信元，不过在信元读写开始时由准备好的输入输出端口进行竞争决定由哪个端口占用读写时隙(输入端口的写竞争和输出端口的读竞争也可以放在一起竞争存储器时隙，可以适用于输入带宽与输出带宽不等的交换要求)。

在输入输出端口，为了保证对外的线速度运行，每个 FIFO 的长度应大于等于两个信元的长度。由于在输入输出端口采取了变长数据包的接口方式，为了避免输入 FIFO 溢出和输出 FIFO 读空(在一个数据包的传输中间)，我们提供的共享存储器的写带宽要大于或等于输入总带宽；读带宽要大于或等于输出总带宽，考虑到采用固定长度信元存储带来的带宽损失，还必须有 7% 的余量，注意到这个余量只是从统计上保证，如考虑在一定时间内包长全为 65byte 的数据包每个会占用两个 64byte 的交换时隙，这时只有交换矩阵的加速比为 2 时才能保证线速处理。在本文中的线速运行暂不考虑这种情况，所有线速处理均指统计意义上的处理。

同时我们必须注意到，在一个数据包的传输过程中，输入 FIFO 的读空或输出 FIFO 的写满是可以发生的，这是因为交换提供的访问带宽大于端口带宽，所以在交换控制中还必须保持对每个端口的读写状态信息。当输入读空或输出写满发生时，该端口不参与读写竞争，只有输入 FIFO 中的有效数据长度大于一个信元长度，这个端口才参与竞争写时隙；只有输出 FIFO 中的空闲空间长度大于一个信元长度，这个端口才参与竞争读时隙。

当采用图 3 中的端口速率可变的共享存储器交换结构时，由于在输入输出端采用了异步 FIFO，使得交换矩阵核心的运行时钟与端口以及端口之间的运行时钟独立，这使我们在设计交换系统时，避免了整个交换系统的时钟全同步要求。

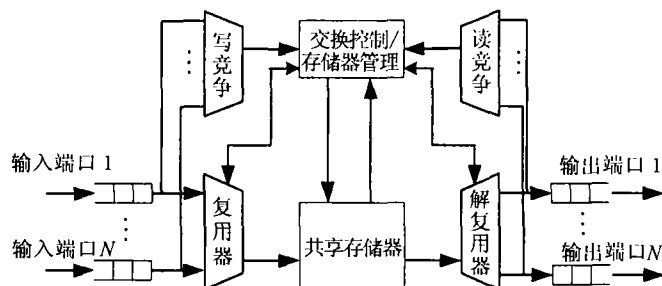


图3 端口速率可变的共享存储器交换结构

3.2 读写竞争的调度算法

在读竞争或写竞争的调度中的算法与常见的队列服务调度属于同一类调度算法，所以我们可以将队列服务调度中的一些算法移植过来，但是读写竞争的调度有其自己的特点，我们在下面给出了读写竞争中调度算法所必须遵从的准则。

设端口中 FIFO 的长度为 L_{fifo} ，单位为信元。由上面知 $L_{\text{fifo}} \geq 2$ ，对任一端口来说，在交换矩阵外的数据包是以整包的形式连续输入输出，而在内部，这个端口对存储器的访问是以信元为单位，间隔进行读写，所以只有输入 FIFO 中的数据量大于一个信元或输出 FIFO 中的空闲空间大于一个信元，这个端口才会参与竞争访问存储器时隙。对于输入 FIFO 来说，是连续写间隔读，而对输出 FIFO 是间隔写连续读。采用的调度算法所要遵从的准则就是，在一个数据包的传输过程中，要避免输入 FIFO 的写满和输出 FIFO 的读空。

设某一输入输出线路的带宽为 B_l ，设调度算法中访问该端口的最大间隔时间为 T_{max} ，则系统给该端口提供的最小带宽为 $B_{\text{min}} = C/T_{\text{max}}$ ， C 为信元长度。当 FIFO 长度为 $L_{\text{fifo}} = 2$ 时，调度算法必须满足：

$$C/T_{\text{max}} \geq B_l \text{ 或 } B_{\text{min}} \geq B_l \quad (1)$$

上式表明，在输入输出线路对 FIFO 的一个信元访问时间内，系统提供给该端口的访问带宽必须大于线路对 FIFO 访问的带宽。

当 FIFO 的长度 $L_{\text{fifo}} = N; (N \geq 2)$ 时，条件有所放松。设调度算法中连续访问该端口 N 次的最大时间为 $T_{N\text{max}}$ ，则系统给该端口提供的最小带宽为 $B_{\text{min}}(N-1)C/T_{N\text{max}}$ ， C 为信元长度。调度算法必须满足：

$$(N-1)C/T_{N\text{max}} \geq B_l \text{ 或 } B_{\text{min}} \geq B_l \quad (2)$$

下面以一个简单的轮询调度算法在读写竞争中的应用来举例：

将对存储器访问（读或写）的 F 个时隙组成一帧，设有 M 个输入或输出端口参与访问存储器。我们在 F 个时隙中对 M 个端口进行分配。使得分配结果满足 (1) 或 (2) 式，尽量使对同一个端口的访问时隙在 F 个时隙中均匀分布。 F 的一个时隙分配示意图如图 4 所示，图中画出了在 8 个时隙的帧中给 5 个端口分配时隙，其中端口 1 分配了一半的带宽，其余 4 个端口共享剩余带宽。在调度时，我们依照时隙分配图轮流访问各端口。

除了轮询调度算法外，我们还可采用其它一些调度算法，如最小队列优先等算法，无论采用哪一种算法，条件 (1) 或 (2) 式是必须满足的。由于轮询调度算法已经满足要求条件而且在调度算法中实现最简单，所以我们采用轮询调度。

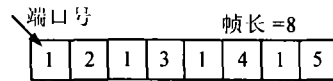


图4 时隙分配示意图

4 队列管理

缓冲区管理是指队列在存储器中的组织形式，为了方便缓冲区的管理，一般将缓冲区分成固定大小的缓冲片，我们称之为信元。对于变长数据包交换，有两种典型的方法^[12]，一是在入口板上将数据包分为固定大小的信元，然后提交给交换矩阵，在出口板上将信元组合成原数据包。这种方式简化了交换矩阵的设计，在交换矩阵内部只需关心信元而不必知晓数据包的存在，但是在出口板上需要大量存储空间来完成数据包的重组，特别是当不同入口不同流的信元间插到来时，而且出口板上必须等待属于同一数据包的所有信元到达后才发送，这也降低了出口线路的利用率。二是在交换矩阵内实现变长包的管理，虽然仍以信元方式存储数据包，但交换矩阵与入口板、出口板之间的接口为变长包接口，这样无需在出口上实现组合功能。这一节中我们主要关心的是交换矩阵的变长包队列管理。

队列的组织方式主要有两种，一种是线性表，一种是链表。线性表是指各个队列在存储器中独占一块区域，特点是结构简单，容错性好，但存储器利用率低。而链表是指所有的队列动态共享存储器，特点是存储器利用率高，但管理复杂，容错性差。在这里我们主要研究链表的组织方式。

链表的组织形式如图5所示。数据包按信元进行存储。存储器按照信元分成基本单元，每个单元附有一个指向下一个单元的指针。存储器中有两种单元：空闲和已占用单元。队列管理器将空闲单元组织为一个链表，称为空闲链表。已占用的单元按照不同的队列组织起来。每个队列需要两个指针，一个指向队列头，一个指向队列尾。

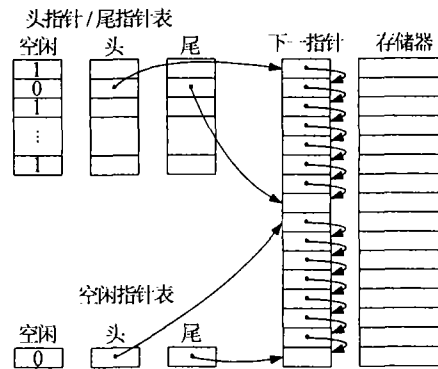


图5 链表的组织形式

在定长数据包交换中，当一个信元到来时，从空闲队列队头中分配一个单元存储该数据包，将它挂在所去队列的尾部，修改所去队列的尾指针和空闲队列的头指针。当一个队列中有信元离开时，我们将队列头指针所指向的信元复制到出口，同时将该信元挂在空闲队列尾，修改空闲队列尾指针和信元所在队列的头指针。当信元到达与信元离开相间时，可以简化这一过程^[13]：输出过程所产生的空闲存储单元并不归还空闲队列，而是直接给下一相邻输入过程使用；当输出过程不产生离队信元时，就在空闲队列上分配一个，当输入过程没有信元入队时，就将输出过程产生的空闲信元归还给空闲队列；这样在输出输入过程中都只有一次入队操作。

在变长数据包交换时,我们必须保持属于同一数据包的不同信元依次相邻的在队列中存放。由于输入端口是轮流访问存储器,不同输入端口的数据包可能间插到达。如果在数据包到达时一次分配它所需要的所有存储单元,这必然要多次访问空闲队列,这在高速的交换矩阵中是不可行的。

对于这一情况,我们采取了延迟入队技术。在具有 N 个输入口的交换矩阵中,我们保持 N 个临时队列与输入端口一一对应,临时队列的操作与普通队列一样。当数据包的信元到来时,我们根据它的输入接口先将它放入临时队列,当所属数据包的最后一个信元到达。我们将整个临时队列挂在数据包所去的队列队尾,同时清空临时队列。一个队列挂在另一队列队尾只涉及一个指针操作。我们保持 N 个临时队列,是因为同时最多只有 N 个输入接口有去往同一输出队列。

5 结 论

本文提出一种新的共享存储器交换结构,利用在每个输入输出端口增加大于两个信元长的 FIFO,提供了端口的自同步特性,避免了要求全局同步;通过动态分配读写共享存储器时隙,为不同速率的端口提供不同的带宽,从而实现端口速率在一定范围内可变。在支持变长包的队列管理上,采用了延迟入队方法,保证了来自同一端口同一数据包的不同信元在队列中相邻存放,不会与其它端口的数据包在队列中交叉存放,从而消除了在输出端口将信元重组成数据包的必要。

参 考 文 献

- [1] S. Keshav, R. Sharma, Issues and trends in router design, IEEE Communications Magazine, 1998, 36(5), 144-151.
- [2] N. K. Sharma, Modular design of a large sorting network, Proceedings of International Symposium on Parallel Architectures, Algorithms and Networks, Taipei, 1997, 362-368.
- [3] Ha, J. Nah, J. H. Lee, K. C. Park, Implementation of 32×32 shared buffer ATM switch and its control software, Asia Pacific Conference on Communications, Sydney, 1997, 1189-1193.
- [4] A. Jajszczyk, M. Roszkiewicz, J. Garcia-Haro, Comparison of ATM shared-memory switches, International Switching Symposium, Berlin, 1995, 409-413.
- [5] F. M. Chiussi, J. G. Kneuer, V. P. Kumar, Low-cost scalable switching solutions for broadband networking: the ATLANTA architecture and chipset, IEEE Communications Magazine, 1997, 35(10), 44-53.
- [6] K. Y. Eng, M. J. Karol, G. J. Cyr, M. A. Pashan, A 160-Gb/s ATM switch prototype using the concentrator-based growable switch architecture, IEEE International Conference on Communications, Washington, 1995, 550-554.
- [7] J. Li, An output-shared buffer ATM switch, International Conference on Computer Design, Austin, 1996, 14-148.
- [8] N. McKeown, How routers are really built, and how optics may play a part, Optics and Routing Seminar, Stanford, 2000.
- [9] N. K. Sharma, Review of recent shared memory based ATM switches, Computer Communications, 1999, 22(4), 297-316.
- [10] S. Kumar, D. Agrawal, On multicast support for shared-memory based ATM switch architecture, IEEE Network, 1996, 10(1), 34-39.
- [11] Y. Xiong, Impact of cell fanout distributions on multicast ATM switch performance, Electronics Letters, 1998, 34(6), 523-525.
- [12] K. Chan, S. Chan, K. Yeung, K. Ko, E. Wong, Clos-knockout: a large scale modular multicast ATM switch, IEEE Globecom, London, 1996, 1358-1362.

- [13] Y. S. Lin, C. B. Shung, Queue management for shared buffer and shared multi-buffer ATM switches, IEEE INFOCOM96, San Francisco, 1996, 688-695.

A VARIABLE PORT SPEED SHARED MEMORY SWITCH

Wang Weizhang Ge Ning Feng Chongxi

(Dept. of Electronic Engineering, Tsinghua University, Beijing 100084, China)

Abstract This paper describes a novel shared memory switch architecture in order to adapt to local area and metroplitan area network's multiple transport speed structures. This new switch can provide variable port speed, variable packet switch, and port's self-synchronization. This paper also describes queue management for variable packet.

Key words Shared memory switch, Buffer management, Queue management

汪卫章：男，1976 年生，博士生，研究方向：数据交换，ASIC 设计。

葛 宁：男，1971 年生，副教授，研究方向：数据交换，ASIC 设计，光通信。

冯重熙：男，1930 年生，教授，博士生导师，研究方向：光通信，宽带网络。