一种在多维分组交换结构中使用的基于死锁恢复策略的自适应路由算法

朱旭东 李乐民 许 都
(电子科技大学宽带光纤传输与通信系统技术重点实验室 成都 610054)

摘 要：在高性能路由器中采用多维分组交换结构是解决可扩展性的一种方法。在实现这种交换结构时，内部路由算法是决定其性能的一项重要技术。该文提出了一种分布式死锁检测机制，它在检测死锁时与交换结构的内部节点队列长度结合，仅考虑本地节点的信息就可实现死锁检测。基于这种检测策略提出了一种新的自适应路由算法 QDAR(Queue length based Deadlock recovery Adaptive Routing)。文中分析了这种内部路由算法在三维 torus 多维分组交换结构中的应用性能。通过改变节点中的缓存器容量，节点间互连物理通道上的虚拟通道个数对算法进行了性能仿真。与现有几种路由算法进行了性能比较。

关键词：内部路由算法，交换结构，多维分组交换结构，死锁恢复

中图分类号：TP393 **文献标识码**：A A **文章编号**：1009-5896(2005)11-1801-05

A Deadlock Recovery Based Adaptive Routing Algorithm for Multi-dimensional Switching Fabric

Zhu Xu-dong Li Le-min Xu Du
(*Key Laboratory of Broadband Optical Fiber Transmission and Communication Networks, UESTC of China, Chengdu* 610054, *China*)

Abstract Scalable switching fabrics can be done on implementing high performance routers by employing multi-dimensional packet switching fabrics. The internal routing algorithm in the switching fabric is a key technology. This paper proposes a new distributed deadlock detection strategy, which combines with queue length on each node without other information required except local information. Based on this technology, a fully adaptive routing algorithm——QDAR(Queue length based Deadlock recovery Adaptive Routing) have been designed. The performance is assessed on 3-dimensional torus architecture. Effect of the buffer length, the number of virtual channels and variable traffic types has been analyzed. Performance evaluation through comparing with other routing algorithms has been done.

Key words Routing algorithm, Switching fabric, Multi-dimensional packet switching fabric, Deadlock recovery

1 引言

直接互连结构(Direct Interconnection Network，DIN)在大型多处理器系统(Massively Parallel Processor，MPP)、多个计算机间的互连和共享缓存多处理器系统中都有着广泛应用。这类结构具有较高的并行处理特性和较好的扩展性。随着VLSI(Very Large Scale Integration)技术的发展，在芯片中可提供更大缓存容量和实现更高的处理速度。直接互连结构已经被考虑应用到高性能路由器的交换结构中[1-3]。由于这类结构在实现时一般采用的是由多个节点以对等方式构成多维拓扑结构，因此，在本文中我们通称这类结构为多维分组交换结构(Multi-dimension Packet Switching Fabric, MPSF)。

MPSF 中的路由算法用来完成把分组从输入端口传送到对应的输出端口，与此同时，好的路由算法能充分利用交换结构的资源(缓存和物理链路)，达到高吞吐率、低时延的目标。路由算法是 DIN 结构中的一个重要的关键技术。

分组数据交换中对吞吐率的要求更高[4]，在直接互连结构中，由于不同目标端口间存在内部冲突，降低了整个结构的吞吐率，为此，可通过增加内部各个节点的处理速度和互连带宽来解决。但这易增加物理实现难度。在现有文献中一般采用虚拟通道策略[5]，来降低内部碰撞引起的分组丢弃。

2004-05-25 收到, 2004-11-25 改回
国家自然科学基金(60372011)和国家 863 T 比特路由器项目资助课题

具体实现时每条物理链路对应的链路缓存器，划分为V个队列，每个队列i对应物理链路上的虚拟通道vc_i。这种策略避免了单个缓存中由于队头阻塞引起的阻碍其他业务流使用空闲物理链路带宽的问题。虚拟通道的另一个作用是解决DIN结构中存在的死锁问题，如何利用虚拟通道技术实现高吞吐率、低时延的目标是路由算法需要解决的问题。有一类解决方法是采用死锁避免的策略，即分组在传送时限制对部分缓存资源的使用[6-8]。该类策略资源利用率不高，同时由于限制了对部分链路的使用，使得分组路由自适应性降低。为了达到对存储器的充分利用，提出了另一类方法，死锁恢复策略。该类算法中，分组可以任意使用各个维上的资源，若发生死锁，用恢复的策略来解除死锁。由于死锁与阻塞存在很大的共性，正确判断死锁需要复杂的统计。简单基于时间的策略由于受分组长度的变化，时间门限值不易确定，而且存在死锁分组同步恢复引起过度恢复的问题。文献[9]提出的一种死锁恢复策略，采用时间门限与令牌结合的方法，当节点中的分组等待时间大于死锁门限时，分组将请求一个在整个结构中流动的令牌，只有获得令牌的死锁分组才能使用恢复死锁用的资源，在实现上较复杂。文献[10]采用寻找死锁分组构成的树的根节点的策略来恢复死锁，解决了死锁过度恢复的情形。但在实现时需要检测所有输出物理链路的状态信息，同时需要分析相邻节点阻塞分组的信息来确定根节点，加大了算法实现的难度。文献[11]主要针对已经发现的死锁分组如何快速传送到目标节点进行了改进。

我们考虑到交换结构中死锁的出现概率同业务负载强度存在很大联系[10, 12]，而在现有文献中还没有结合负载强度的死锁恢复策略。所以，本文提出了一种同业务负载强度相结合的死锁检测策略，其基本思想是在检测死锁时除了对分组等待时间进行检测，同时结合对内部节点上的队列的长度进行判断。在实现时基于流量控制信息，每个节点可以独立检测死锁。由于不同路径上的业务负载不同，所以这种方法可以部分解决几个死锁分组同步进入死锁恢复而产生的过度恢复。这是因为死锁是由于多个分组路径成环引起的，而移动死锁环中的一个分组就可以实现死锁恢复。在分布式检测时，由于各个节点的死锁检测信息并不相同，所以检测的结果不完全相同，从而可以避免死锁环上的分组同时被检测为死锁分组。

基于此死锁检测策略，我们实现了一种新的自适应路由算法—QDAR(Queue length based Deadlock recovery Adaptive Routing)。在文中仿真分析了算法在三维 torus 直接互连结构的中应用时的吞吐率时延特性。通过改变缓存器长度，虚拟通道的个数，交换结构内部加速因子进行了仿真分析，与其它几种算法进行了仿真比较。最终获得了一些对用多维结构构建 $T(T=10^{12})$比特路由器的有意义的结论。

本文第 2 节介绍所采用交换结构的模型以及交换节点的简单示意图。第 3 节叙述死锁检测方法和 QDAR 路由算法。第 4 节为仿真结果和性能分析。第 5 节总结全文。

2 交换结构模型

一个n维拓扑结构可用$(k_1,k_2,\cdots,k_i,\cdots,k_n)$来表示，其中$k_i$表示维$i$的直径(节点个数)。本文中我们采用一个三维torus结构，整个结构的内部节点间互连采用双向链路。在本文中提到的所有的内部节点为构建交换结构的基本交换单元(Switch element)，在本文中用节点来指代。每一维上最远节点间存在物理链路连接构成一个圆环(torus)结构。torus结构中的每个内部交换节点的规模为$(2n+1)\times(2n+1)$。即每个物理链路对应crossbar的一个交换端口。其中n是结构维数(Dimension)。图 1 是三维torus结构中的交换单元结构示意图。内部交换单元上处理分组采用两级调度策略，工作流程如下：

第 1 步 在收到一个分组后根据分组的虚拟通道号VCID(Virtual Channel Identification)放入对应的虚拟通道缓存队列中，由‘VC 分配单元’根据目标地址和下一跳节点的虚拟通道状态信息分配下一跳虚拟通道。不同类型路由算法返回虚拟通道个数不同，在自适应路由算法中可以选择多个输出虚拟通道，但在确定路由算法中返回的虚拟通道数就只有一个。

第 2 步 所有在第 1 步中获得虚拟通道号的分组和上一个时隙已经获得的虚拟通道未释放的分组一起竞争输出物理链路。在此，输出物理链路竞争采用文献[13]中的请求-应答-确认调度机制来解决物理链路请求冲突。

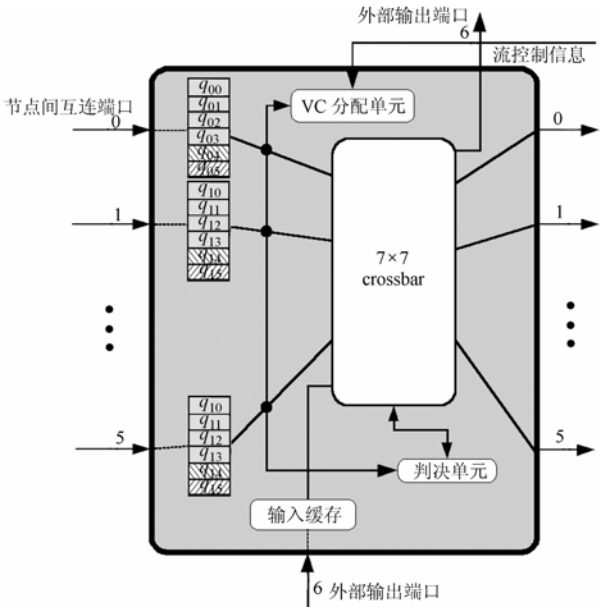

图 1 交换单元结构示意图

3　死锁检测机制和 QDAR 路由算法

本节描述了同业务负载强度相结合的死锁检测机制，以及基于这种检测机制而提出的 QDAR 路由算法。在算法描述前，我们对虚拟通道和对应的节点上的缓存作如下的设定。

物理通道用下面的式子表示：$p_{(n_s,n_d)}$，设定，如果 $s>d$ 则是正向链路，反之，为负向链路。每一条物理通道总共包含 V 个虚拟通道，分成两类：普通虚拟通道(NVC, Normal Virtual Channel)和逃逸虚拟通道(EVC, Escape Virtual Channel)。我们沿用类似文献[6]中的方式来标识各个虚拟通道。如 $vc_{i,+,k,(x_0,x_1,x_2)}$ 表示目标节点是 (x_0,x_1,x_2)，在 i 维上的一个正向虚拟通道，其中，如果 $0\le k<2$，表示是逃逸虚拟通道，$k\ge 2$ 为普通虚拟通道。i+表示 i 维上的正向通道，同理，i – 就是指 i 维上负向通道。

在 QDAR 中，每个方向上的物理链路上普通虚拟通道数 $C_{\mathrm{NVC}}\ge 1$，表示为 $vc_{i,\pm,k,(x_0,x_1,x_2)}$ 其中 $k\ge 2$；逃逸通道数 $C_{\mathrm{EVC}}=2$。如果是正方向，就是下面两个：$vc_{i,+,0,(x_0,x_1,x_2)}$ 和 $vc_{i,+,1,(x_0,x_1,x_2)}$；负方向是：$vc_{i,-,0,(x_0,x_1,x_2)}$ 和 $vc_{i,-,1,(x_0,x_1,x_2)}$。文中，torus 结构中每维上的最远节点间的链路称为 WR(Wrap Round)链路。

3.1　死锁检测恢复策略

下面首先描述基于排队长度和等待时间的死锁检测策略。该策略应用两个参数，分别是：排队队列长度门限($\mathrm{QL_{th}}$)和等待时间门限($\mathrm{WT_{th}}$)。当交换节点上队列长度大于$\mathrm{QL_{th}}$时，就检测该队列头分组的等待时间，如果分组等待时间大于$\mathrm{WT_{th}}$，认为该分组死锁。也就是在这种检测机制中，当负载强度较大时，内部节点上的队列排队长度QL增加的速度较快，则检测到的死锁分组数增加，这符合在业务负载强度大时分组发生死锁的概率增加的规律。另外为防止在业务负载较小时存在死锁(这种情形发生概率较小)，为此设置了最大超时时限T_{out}。当分组等待时间大于T_{out}时同样认为该分组死锁。在检测到结构中出现死锁时，需要应用死锁恢复机制来解决死锁。我们采用的是积极(Progressive)死锁恢复策略，即不丢弃死锁分组，而是有效利用已有的带宽资源来传递死锁分组。我们设定每个虚拟通道对应缓存空间的上限为$\mathrm{QL_{max}}$。这些参数需满足下面关系式：

$$\begin{cases}\mathrm{QL_{th}}\le \mathrm{QL_{max}}\\ \mathrm{WT_{th}}\le (\mathrm{QL_{th}}+\mathrm{QL_{max}})/\mathrm{Channel\_speed}\\ T_{\mathrm{timeout}}>\mathrm{WT_{th}}\end{cases}\qquad(1)$$

其中 Channel_speed 表示交换节点间物理链路的传送速率。

3.2　QDAR 路由算法

根据上一节中的死锁检测的思想，我们提出了一种基于死锁恢复策略的路由算法QDAR。在实现时采用分布式机制，即每个节点根据相邻节点状态信息和分组头的路由信息来独立进行路由计算，交换结构中的每个节点都进行下面的路由计算。设分组Pk从输入端口I_{src}传送到输出端口O_{dest}，那么在交换结构中就是从节点n_{src}到n_{dest}，每个节点的路由算法用来确定下一跳节点n_{next}。下面是节点n_{cur}中计算下一跳的过程：

步骤 1　如果分组 Pk 是死锁分组，跳转到步骤 4。

步骤 2　如果分组Pk是普通分组，则任意选择更接近n_{dest}的节点(y_0,y_1,y_2)作为下一跳节点n_{next}，分组可以使用的虚拟通道就是位于物理链路 $p_{(n_{\mathrm{cur}},n_{\mathrm{next}})}$ 上的虚拟通道，即 $vc_{i,\pm,k,(y_0,y_1,y_2)}$ 其中 $k\ge 2$。在此可能存在多个下一跳节点，我们采用随机方法选择一个。步骤 2 保证分组可选择源、目标节点间所有最短路径。

步骤3　如果步骤2中物理链路上所有虚拟通道都无效，分组等待下一时隙。同时，T_{elapsed}增加 1 个时隙。

步骤 4　如果 Pk 是死锁分组，采用维序路由 DOR (Dimension Order Routing)。而且所选的通道必须为逃逸通道。在选择虚拟通道时必须判断分组是否经过了 WR 链路，如果没有，则选择 $vc_{i,\pm,0,(x_0,x_1,x_2)}$;反之，则选择 $vc_{i,\pm,1,(x_0,x_1,x_2)}$。

为了说明QDAR路由算法，我们给出了图 2 在二维torus多维交换结构中的路由示意图。路由从源节点(0,1)到目标节点(2,5)。根据排列组合理论，如果分组从源到目标节点各维上的差值分别为 $\Delta x,\Delta y,\Delta z$。那么它可选择的路径数为$L_{\mathrm{path}}$：

$$L_{\mathrm{path}}=\frac{P_{(\Delta x+\Delta y+\Delta z)}^{(\Delta x+\Delta y+\Delta z)}}{P_{\Delta x}^{\Delta x}P_{\Delta y}^{\Delta y}P_{\Delta z}^{\Delta z}}\qquad(2)$$

式(2)中 P 表示计算排列数。例如，在 m 个样本空间中取出 n 个样本进行排列，可排列的总数为：$P_m^n=m\times(m-1)\times(m-2)\times\cdots\times(m-n+1)$，$(m\ge n)$。在图 2 中从源节点到目标节点，可选择的路径总数为 $L_{\mathrm{path}}=(2+4)!/(2!4!)=15$。

沿用文献[6]中的分析方法，由于QDAR是最短路径算法，所以实现该路由算法，在一条双向物理链路仅需 5 个虚拟通道就可实现。即虚拟通道 $vc_{i,+,0,(x_0,x_1,x_2)}$，$x_i>k/2$ 和虚拟通道 $vc_{i,-,1,(x_0,x_1,x_2)}$，$x_i<k/2$, 其中i={0,1,2}不会被使用，可以省略。

QDAR 算法采用队列长度作为附加的一个指标判断死锁，从而把业务负载强度与死锁恢复结合。由于在输入业务

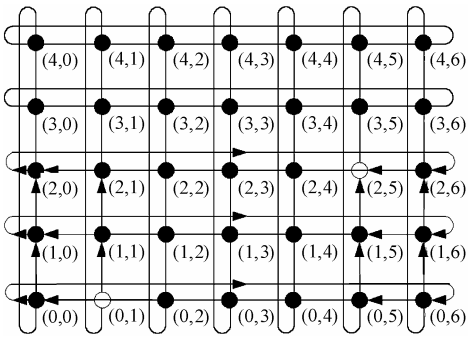

图 2　从节点(0，1)到节点(2，5)的路径示意图

量大时死锁概率增加，因此把业务量和死锁恢复的机制结合起来是一种死锁判断的好的尝试。在路由算法实现时仅使用本地节点信息，在分布式结构中可降低硬件实现的难度，从而提高死锁判断和恢复的速度。因为虚拟通道数与维数和节点数无关，所以路由算法易于扩展。所需的通道个数少，对于虚拟通道管理单元的实现简单。在我们建议的策略中由于各个分组属于趋向不同节点的业务流，所以队列长度的增长是异步的。这样就可以实现异步死锁恢复机制，部分解决由于同步恢复造成的过度恢复问题。

由于 QDAR 基于死锁恢复机制，使得它对结构的依赖性不强，易于实现结构扩展，可应用到其它不规则多维交换结构中。

4 仿真结果

我们使用 OPNET 仿真工具构建了一个三维 torus 交换结构。交换结构采用虫孔路由(wormhole)策略，即每个入端口的包，在进入交换结构前被切分成多个相同长度的分组(flit)，在第一个分组中保留路由和控制信息，其余分组仅需随着第一个分组经过的路径传送。每个分组在内部节点上的处理时间和通道上的传送时间分别为一个单位时隙(cycle)。连接各个交换节点的物理链路的传送速率为每个时隙发送一个分组。如果结构采用加速因子 s，那么每个单位时隙处理的分组个数为 s 个。我们采用统计平均每个节点每时隙接收到的分组数 τ 作为交换结构的吞吐率。在没有指明业务类型时采用的都是随机均匀业务，即分组发送到每个输出端口的概率相同，都为 $1/N$ (N 为交换结构的端口数)。三维结构的规模为(4,16,5)，则 MPSF 结构为 320×320，即 N=320。每个端口的业务独立同分布，为均匀分布(Uniform)变长包，包平均长度为 20 个分组。

4.1 虚拟通道数和缓存大小改变对交换结构性能的影响

本节通过改变每个物理链路上的虚拟通道个数和每个节点的缓存大小来分析QDAR路由算法的性能。图 3 是仿真结果。在输入业务量较大时，缓存器长度QL_{max}为 200 个分组时性能最差。这种情形是由于这种结构的特性造成的。因为随着中间节点的缓存增加，在业务量很大时需要转发的中间业务负载始终处于接近饱和状态，内部冲突加剧，死锁概率增加。反之，如果中间节点的缓存小，分组在每个节点的本地输入缓存处堵塞。对DIN内部转发业务影响较小，反而能达到比业务负载较大时更好的业务转发能力。所以在此缓存容量小的结构比缓存容量大的结构具有更高的吞吐率。这也证明了死锁恢复的路由算法在业务负载接近饱和时，性能将变差。为提高整个结构的吞吐率，仅依靠提高中间节点缓存容量是没有效果的。在后面的仿真中我们尝试了采用增加节点间链路的带宽和交换单元处理速度的方法来提高其性能。

图 4 是虚拟通道的增加对性能的影响，在业务负载较小时，虚拟通道数 VC=5 时，有更小的时延。但在业务负载超出一定值时，随着 VC 的增加，分组平均时延略有增加。这是由于虚拟通道可以增加路由灵活性，在业务量较小时可提供好的自适应路由，降低分组平均时延。但是当业务负载接近饱和时，灵活的路由反而增加了转发业务的负载强度，导致更高的死锁概率，造成性能降低。

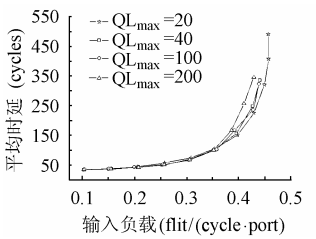

图 3 改变队列长度对性能的影响

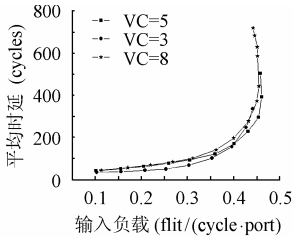

图 4 改变虚拟通道数对性能的影响

4.2 交换结构内部加速对性能的影响

为了分析交换结构内部加速(speedup)对于整个结构性能的影响，我们设定下面 3 个参数进行仿真实验：节点的处理速度增加倍数 PSS；节点间链路提速倍数 BWS 和虚拟通道个数 VC。图 5 是仿真结果。从结果图中可以看出在物理链路提速 BWS=2，交换节点处理速度不变 PSS=1 时性能比不提速时有所提高；在此基础上我们增加虚拟通道 VC=4，吞吐率获得进一步提高。在 PSS=2，BWS=2，VC=3 系统性能更进一步大幅提升。但在此基础上增加虚拟通道数对性能提升非常小。

在提速时可提高系统的性能。在物理链路提速时，可以通过增加虚拟通道的个数来提高物理链路带宽的利用率，从而提高整个结构的性能。虚拟通道只有在物理链路带宽不是瓶颈时才能起到提高系统性能的目的。

4.3 改变包的平均长度对性能的影响

我们采用不改变死锁恢复的各个参数设置，仅改变包的平均长度(PKLEN)来分析 QDAR 路由算法的吞吐率。图 6 是仿真结果图。横坐标为输入端口的业务负载强度，纵坐标为

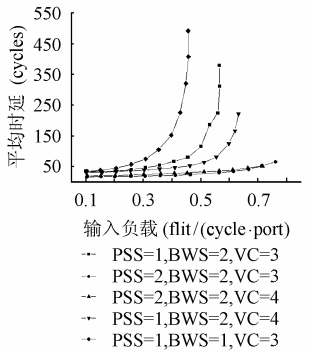

图 5 交换结构内部加速对性能的影响

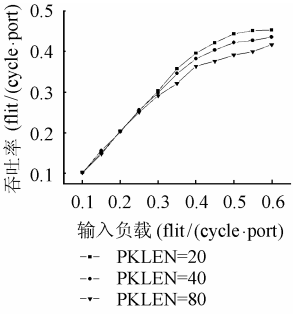

图 6 包长度改变对性能的影响

输出业务强度，即吞吐率。变长包的平均长度分别为 20 个分组，40 个分组，80 个分组，QDAR 算法的吞吐率变化不大。这是因为，虽然，在包长度的增加会在短时间内使得缓存的长度增加，但是由于 QDAR 算法采用的是分析连续负载强度的策略，而在包长度增加时，在相同的业务负载下，包较长的业务各个包之间的间隔时间也同时变大。所以，实际缓存的长度并不会连续快速增加。因此，包长度的改变对 QDAR 算法的性能影响很小。

4.4　不同路由算法间的性能比较

我们把QDAR与下面几种路由算法进行了比较：分别是源路由算法RLB[8]，自适应路由算法star-channel[6]和确定路由算法DOR[14]。从图 7，图 8 仿真结果可以看出QDAR算法可以在平均时延和吞吐率上优于这 3 种不同类型的路由算法。

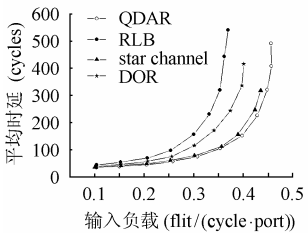

图 7　不同路由算法时的时延-吞吐率曲线

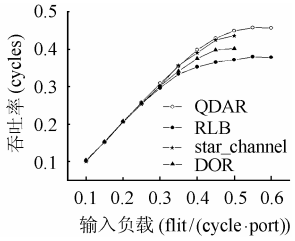

图 8　不同路由算法时的负载-吞吐率曲线

5　结束语

本文提出了在死锁检测时同业务负载相结合的策略，在此基础上提出了一种基于死锁恢复的路由算法。把这种算法应用到三维 torus 结构中进行了仿真分析。得到了一些关于采用三维 torus 结构构造 T 比特级别路由器的有意义的结论。首先，在 QDAR 路由算法中，当交换节点虚拟通道数或缓存容量超出一定值后，增加虚拟通道数或缓存容量不能提高交换结构的性能，在业务负载过大时反而会降低其性能；其次，交换结构内部提速可以提高整个结构的性能；最后，相比其他路由算法，QDAR 有更优的性能。由于允许任意选择通路，所以该算法并不限制在三维 torus 互连结构中使用，还可以应用到其他不规则 MPSF 结构中。

参 考 文 献

[1] Dally W J. Scalable switching fabrics for Internet routers. 1999, Computer Systems Laboratory, Stanford University and Avici Systems Inc., http://www.avici.com.

[2] Park J S, Davis N J. The folded hypercube ATM Switch[C]. IEEE International Conference on Networking, Colmar, France, 2001: 370 – 379.

[3] Mir N F. An efficient switching fabric for next-generation large-scale computer networking[J]. *Elsevier Computer Networks*, 2002, 40(2): 305 – 315.

[4] Dally W J. Introduction to Interconnect network. 2000, http://cva.standford.edu/ee482course.

[5] Dally W J. Virtual channel flow control[J]. *IEEE Trans.on Parallel and Distributed Systems*, 1992, 3(3): 194 – 205.

[6] Luis G, Gusatavo D. Adaptive deadlock- and livelock-free routing with all minimal paths in torus networks[J]. *IEEE Trans.on parallel* and *Distributed systing.*, 1994. 5(12): 1233 – 1251.

[7] Tedd N, Johnsson L S. ROMM routing on mesh and torus networks[C]. 7th Annual ACM Symposium on Parallel Algorithms and Architectures SPAA'95, Santa Barbara, California, 1995: 275 – 287.

[8] Singh A, Dally W J, *et al.*. Locality-preserving randomized oblivious routing on torus networks[C]. ACM Symposium on Parallel Algorithms and Architectures (SPAA), Winnipeg, Manitoba, Canada, 2002: 1 – 12.

[9] Pinkston T M. Flexible and efficient routing based on progressive deadlock recovery[J]. *IEEE Trans on Computers*, 1999, 48(7): 649 – 669.

[10] Lopez P, Martnez J M, Duato J. A very efficient distributed deadlock detection mechanism for wormhole networks[C]. IEEE proceedings, The 4th International Symposium on High Perf. Computer Arch., Las Vegas, USA, Feb.1998: 57 – 66.

[11] Song Y H, Pinkston T M. A new mechanism for congestion and deadlock resolution[C]. Proc., International Conference on Parallel Processing, Vancouver, Canada, Aug. 2002: 81 – 90.

[12] Duato J P, Pinkston T M. A general theory for deadlock-free adaptive routing using a mixed set of resources[J]. *IEEE Trans on Parallel and Distributed Systems*, 2001, 12(12): 1219 – 1235.

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

[14] Ni L M, McKinley P K. A survey of wormhole routing techniques in direct network[J]. *IEEE Computer*, 1993, 26(2): 62 – 76.

朱旭东：　男，1974 年生，博士，主要从事宽带网络中分组数据交换结构、结构中的调度策略、路由算法等的研究. E-mail: zhuxudong@std.uestc.edu.cn.

李乐民：　男，1932 年生，教授，博士生导师，中国工程院院士，目前主要研究方向为宽带通信网.E-mail: lml@std.uestc.edu.cn.

许　都：　男，1968 年生，副教授，主要从事网络性能分析、核心路由器体系结构中的关键技术研究.