D2RA:基于混合路由的片上网络无死锁通信机制

肖灿文[1)] 汪玮[1),2),3)] 张晓云[4)] 李存禄[1)] 杨博[1),2),3)] 刘杰[1),2),3)] 车永刚[1)]

[1)](国防科技大学计算机学院，长沙 410073)

[2)](高端装备数字化软件湖南省重点实验室，长沙 410073)

[3)](国防科技大学并行与分布处理全国重点实验室，长沙 410073)

[4)](电子科技大学智能协同计算技术国家级重点实验室，成都 611731)

摘 要 片上网络通常采用受限的路由算法实现无死锁路由。维度气泡路由算法（Dimensional Bubble Routing Algorithm, DBRA）提供了另一种思路，仅需确保下游路由器的空闲空间大于剩余维度数即可实现无死锁传输，因此成为一种有前景的片上网络无死锁路由解决方案。然而，维度气泡路由算法在高维度片上网络中的一个显著缺点是，它需要大量的空闲缓存空间才能维持无死锁特性。具体来说，对于剩余 n 个维度的报文，下一跳路由器缓存队列中的可用缓冲区空间必须至少为 n，才能确保报文的无死锁传输。而维序路由（Dimension-Order Routing, DOR）虽无需额外缓冲区空间，但其存在显著局限性：严格要求按固定维度顺序完成路由（如先 X 维后 Y 维），完全丧失路径适应性，在网络局部拥塞时易因路径僵化导致报文堆积，大幅降低网络吞吐率与负载扩展性。基于此，本文提出了一种面向片上网络的新型无死锁完全适应性路由算法 D2RA。该算法结合了维度气泡路由算法和维序路由算法，在网络空闲缓存资源较少时采用维序路由，既能够充分发挥维度气泡路由路径多样性的优势，又能有效利用片上网络的缓存资源。该算法特别适用于 k-ary n-mesh 网络。与单独使用维度气泡路由算法相比，混合路由算法在采用维序路由时对下一步缓冲区的空闲空间需求大大降低并且维序路由不需要额外的缓冲区空间。本文证明了 D2RA 在任何 k-ary n-mesh 网络中均能保证无死锁，同时，实验结果也表明了该算法具有更优的性能及更强的负载和维度可扩展性：在 4-ary 3-mesh 网络中，D2RA 算法的报文平均延迟相较于 DBRA 算法最高有 96%的减少；较其他常用算法最高有 81.7%的减少。并且实验通过改变不同网络 VC 数目以及不同网络规模，验证了 D2RA 算法在 NoC 这种资源受限的场景的突出优势。

关键词 维度气泡路由；维序路由；混合路由策略；死锁；片上网络

中图法分类号 TP302

D2RA: A Deadlock-Free Communication Mechanism for Network-on-Chip Based on Routing Hybridization

XIAO Can-Wen[1)] WANG Wei[1),2),3)] ZHANG Xiao-Yun[4)] LI Cun-Lu[1)] YANG Bo[1),2),3)] LIU Jie[1),2),3)] CHE Yong-Gang[1)]

[1)](College of Computer Science and Technology, National University of Defense Technology, Changsha 410073)

[2)](Laboratory of Digitizing Software for Frontier Equipment, National University of Defense Technology, Changsha 410073)

[3)](Science and Technology on Parallel and Distributed Processing Laboratory, National University of Defense Technology, Changsha 410073)

[4)](Laboratory of Intelligent Collaborative Computing Technology, University of Electronic Science and Technology of China, Chengdu 611731)

本课题得到国家重点研发计划(2021YFB0300101)资助。肖灿文，博士，研究员，主要研究领域为计算机体系结构、高性能互连网络。汪玮（**通信作者**），博士研究生，中国计算机学会（CCF）会员，主要研究领域为通信优化和高性能计算。张晓云，博士，助理研究员，主要研究领域为高性能计算和片上网络。李存禄，博士，副研究员，主要研究领域为计算机体系结构、计算机网络。杨博，博士，副研究员，主要研究领域为通信优化和高性能计算。刘杰，博士，研究员，主要研究领域为通信优化、高性能计算和数值计算。车永刚，博士，研究员，主要研究领域为并行算法、性能评测和高性能计算。

Abstract Traditional deadlock avoidance techniques commonly employ restricted routing algorithms to achieve deadlock freedom for network-on-chip. Dimensional Bubble Routing Algorithm (DBRA) proposed a different deadlock avoidance theory which ensures deadlock-free transmission by guaranteeing that the free space in downstream routers is greater than the number of remaining dimensions, making it a promising deadlock-free routing solution for network-on-chip. However, a significant drawback of the DBRA algorithm in high-dimensional network-on-chip is that it requires a large amount of free buffer space to maintain its deadlock-free properties. Specifically, for a packet with *n* remaining dimensions, the available buffer space in the next hop queue must be at least *n* to ensure deadlock-free transmission. In contrast, Dimension-Order Routing (DOR) does not require additional buffer space but has significant limitations: it strictly mandates routing to be completed in a fixed dimensional order (e.g., X-dimension first, then Y-dimension), completely losing path adaptability. This rigidity in paths can easily lead to packet accumulation when the network is locally congested, significantly reducing network throughput and load scalability. This paper presents a novel deadlock-free fully adaptive routing algorithm for network-on-chip called D2RA. The algorithm combines DBRA and Dimensional Order Routing (DOR). It uses DOR when network buffer resources are scarce, thereby fully utilizing the path diversity of DBRA while efficiently using the on-chip network's buffer resources. The algorithm is particularly suitable for *k*-ary *n*-mesh networks. Compared to using DBRA alone, the hybrid routing algorithm significantly reduces requirements of free spaces in the next buffer when employing DOR, moreover, DOR does not require additional buffer space. This paper proves that D2RA guarantees deadlock-freeness in any *k*-ary *n*-mesh network, and experimental results show that it has superior performance, as well as stronger load and dimension scalability. In the 4-ary 3-mesh network, the D2RA algorithm achieves a reduction in average packet latency of up to 96% compared to the DBRA algorithm, and up to 81.7% relative to other popular algorithms. Furthermore, experiments conducted by varying the number of virtual channels (VCs) and network scale demonstrate the exceptional advantages of the D2RA algorithm in resource-constrained Network-on-Chip (NoC) environments.

Key words dimensional bubble routing; dimension-order routing; mixed routing strategy; deadlock; network-on-chip

1 引言

片上网络（Network-on-Chip, NoC）[1, 2, 3]通过路由器与链路的相互连接，构成了多核处理器中各个核心之间的互连通信基础架构，其性能对整个高性能计算系统至关重要。路由算法作为路由器转发功能的核心，决定了报文从源路由器到目的路由器的传输路径，直接关系到路由器的性能和网络传输效率[4]。

由于完全适应性路由策略具有可选路径最多的特点，可以充分利用网络资源，减少通信冲突，实现高效的通信，另外相对确定性路由，适应性路由策略对网络故障点具有更高的容忍度，现有的片上网络路由算法通常基于完全适应性路由采用印迹跟随（footprint）的虚拟通道选择[5]或采用机器学习的路径选择[6]等方式实现高效的路由选择。除了优化报文传输效率外，设计完全适应性路由算法必须解决一个关键问题：如何避免路由死锁。面向片上网络的完全适应性路由策略通常基于 Duato 原理[7]来实现无死锁路由。这些路由算法通常设置少量的虚拟通道作为逃逸通道，当报文可能发生死锁时，报文会被引导进入逃逸通道并采用无死锁的路由策略进行传输；而正常情况下，报文则会占用尽可能多的缓存资源，采用完全适应性路由策略进行传输。

Duato 死锁避免机制需要额外的虚拟通道支持，从而引入了额外的缓存资源开销[8]。已有研究提出了一种针对 *k*-ary *n*-mesh 网络的无死锁流控制理论，该理论避免了跨维度的死锁，并提出了完全适应性的维度气泡路由算法（Dimensional Bubble Routing Algorithm, DBRA）[9, 10]。该无死锁流控制理论允许使用任何最短路径适应性路由算法，无需额外的“逃逸”虚拟通道。研究表明，DBRA 能够确保任何 *k*-ary *n*-mesh 网络中无死锁。然而，依据 DBRA，对于具有 *n* 个剩余维度的报文，其下一跳

路由器队列中的可用缓冲区空间必须至少为 n（报文切片数量）。这个限制在以下情况下影响了 DBRA 的适用性：

（1）即便下一跳队列中已有$n-1$个可用缓冲区空间，具有 n 个剩余维度的报文仍无法进入下一跳路由器队列，这一约束可能对网络吞吐量产生负面影响。

（2）随着 n 值的增加，具有 n 个剩余维度的报文进入下一跳路由器队列的难度加大，尤其在网络拥塞时，具有 n 个剩余维度的报文可能会面临饥饿现象。

此外，维序路由（Dimension-Order Routing, DOR）虽无额外缓冲开销，且通过固定维度顺序（如先 X 维后 Y 维）确保路由无死锁，另一方面由于维序路由的路径单一，网络拥塞时报文阻塞严重，大幅降低网络吞吐率与负载扩展性。

上述方案的核心矛盾在于“资源需求”与“路径适应性”的失衡：DBRA 有适应性但需多缓冲，DOR 需少缓冲却无适应性，Duato 算法则依赖额外资源。为解决这一矛盾，本文提出新型无死锁完全适应性路由算法 D2RA，其核心设计思路在于融合 DBRA 与 DOR 的互补特性：当网络缓冲充足时，采用 DBRA 以充分利用路径多样性分散流量；当缓冲资源稀缺（下一跳缓冲＜剩余维度数）时，切换为 DOR 以规避阻塞，同时依托 DOR 的维度顺序约束维持无死锁。这种混合策略既无需额外虚拟通道，又能动态适配不同负载下的资源状态，高效平衡缓冲利用与路径适应性。并且，本文形式化证明了该算法在任何 k-ary n-mesh 网络中都具有无死锁特性。

2 相关研究

如何设计高效的片上网络路由算法，目前已开展了大量研究的工作。这些工作主要分为两大类型：

（1）基于启发式的适应性路由算法设计。这种方法使用静态网络状态信息作为判定拥塞的度量指标，并且这些路由算法主要从空间维度和时间维度上持续获取丰富的网络状态信息。

（2）基于学习式的适应性路由算法设计。这种方法主要借助机器学习技术来设计路由策略，通过不断从动态网络状态信息中学习和训练而得。

启发式的适应路由算法代表性研究有马胜等人提出了一种基于目的的自适应路由算法（Destination Based Adaptive Routing, DBAR）[11]，它根据当前节点和目的节点之间的最小象限内的X维度上和Y维度上的节点状态信息来选择输出端口，以减少区域内和区域外的干扰问题的产生。但是，DBAR路由算法构建的拥塞传播网络增加了路由实现复杂度和硬件开销；此外，拥塞传播过程存在延迟从而导致网络状态信息不准确。付斌章等人提出了一种足迹路由算法（Footprint）[5]，当端点拥塞时报文跟随之前同一目的的报文的路径传输报文从而动态隔离报文传输路径。该路由算法设计Footprint虚通道（Virtual Channel, VC）度量指标，该虚拟通道中存在与当前路由报文具有相同目的地的报文，这样保留路由历史足迹。

然而已有的启发式适应性路由算法所获得的网络状态信息中存在信息局限性强并且增加了设计复杂性等问题。

基于学习式的适应性路由算法设计主要是借助机器学习的方法设计路由算法。文章[12]提出了一种自适应路由的综合强化学习框架RELAR。RELAR框架适用于多种流量模式并解决多目标优化问题。文章[13]利用强化学习模型从多种候选路由算法中学习，并使用片上网络中的缓冲区和链路利用率信息来选择最佳路由算法，以提高片上网络的整体性能。张晓云等基于决策树学习模型提出了自适应路由算法DeTAR[6]。该算法采用决策树模型选择多种片上网络状态指标并对关键状态信息进行优先级排序，然后根据网络状态的数据集训练决策树模型，并将该模型输出的决策树转化为可实际部署的适应性路由算法。然而，这些基于机器学习的路由算法因硬件开销高且难以实际部署而未能广泛应用。

以上研究都采用Duato策略避免路由死锁。Duato方法通过混合完全适应性路由和逃逸路由实现网络无死锁路由。而DBRA算法不同于Duato策略，它采用流控策略保证了无死锁的完全适应性路由。然而，DBRA算法的缓冲区利用不均衡，剩余维度数目越多的报文越难流出。在本文中，我们将基于DBRA流控采用混合路由方式保证剩余维度数目多的报文在目标缓冲区只有一个空闲报文空间时也可以流出。

3 研究背景

在本节中，我们回顾文章[10]中使用的形式化符号，并引入了额外符号以描述本文提出的路由算法。相关符号表示紧密遵循经典工作[7]的文本，并根据互连网络的结构进行了简化定义。本文涉及主要的符号及其含义如表 1 所示。

表 1 符号表

符号	含义
N	互连网络中的节点集合
Q	互连网络中连接节点的队列集合
D	维度方向集合
u	当前节点（以维度空间坐标唯一表示）
v	目的节点（以维度空间坐标唯一表示）
d	某一维度方向
q	某一个缓冲队列，可用$\langle u,d\rangle$或$c(p)$表示
k(1)	表示该节点为某方向上的边界节点，当等于 k 时表示正方向边界节点，等于 1 时表示为负方向边界节点。
g	维度号
a	集合{−1, +1}中的非空元素
P	报文集合
C	某一个互连网络的配置集合
p	某一份报文
c	某一个互连网络中的一种配置
R	路由函数
F	流量控制函数
S	队列选择函数
z	剩余维度方向的数目
$cap(q)$	队列 q 的容量（按报文数量计）
$\langle R,F,S\rangle$	路由算法
$\Re^{\langle R,F,S\rangle}$	路由选择函数
$neighbor(u,d)$	节点 u 在 d 方向上下一跳的输入缓冲队列
$credit(c,q)$	队列 q 在配置 c 下可用的缓冲区空间数量
$dest(p)$	获取报文 p 的目的节点
$sign(\ )$	获取元素的正负号，即某个维度上的方向
$num(\ )$	获取集合中元素的个数
$RDS(u,v)$	节点 u 到节点 v 剩余维度的集合
$RDD(u,v)$	节点 u 到节点 v 剩余维度方向的集合

3.1 网络拓扑

本文专注于 k-ary n-mesh 网络，图 1 是一种典型的 k-ary n-mesh 网络，图 1(a)中，$k=4$，$n=2$。为了准确描述它们的特点，我们进行如下定义。

定义 1. 一个互连网络$\langle N,Q\rangle$，其中 N 是一组节点（路由器），Q 是连接节点的队列集合。

定义 2. 一个 k-ary n-mesh 网络是一个元组$\langle N,Q\rangle$，其中$N=N_1\times N_2\times...\times N_i\times...\times N_n$，是节点的集合，其中 $N_i=[1,2,...,k]$，$\forall i\in[1,2,...,n]$；$Q=N\times D$是队列的集合，其中$D=[0,1,...,n]\times\{+1,-1\}$是维度方向的集合。

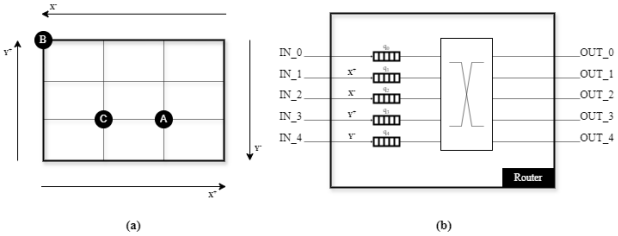

图 1 (a) 4-ary 2 mesh 网络 (b) 二维 mesh 网络的路由器微结构

我们用节点在 n 维空间的坐标作为它的唯一标记。$u=\langle u_1,u_2,...,u_i,...,u_n\rangle\in N$，其中$u_i$是节点 u 在第 i 维的坐标。另外，若$u_i=k$，则表示节点 u 是第 i 维正方向上的边界节点；若$u_i=1$，则表示节点 u 是第 i 维负方向上的边界节点。图 1 中，节点 A$=\langle 3,2\rangle$，节点 B$=\langle 1,4\rangle$既是 Y 维正方向上的边界节点又是 X 维负方向上的边界节点。若$d\in D$，那么$\langle u,d\rangle\in Q$表示节点 u 在 d 维度方向的输入缓冲队列。例如：图 1(b)中，q_1是 x 正方向上的输入队列。

我们定义节点 u 在第 i 维上的相邻节点为除了第 i 维的坐标差 1，其它维度上的坐标相同的节点。例如节点 C$=\langle 2,2\rangle$是节点 A 在 x 负方向上的相邻节点。

我们用维度号 g 和正负 1 的元组表示维度方向。+1 标示正方向，-1 标示负方向。为了方便描述，我们标记 x 正方向为$x^+=\langle 1,+1\rangle\in D$，$x$ 负方向为$x^-=\langle 1,-1\rangle\in D$，$y$ 正方向为$y^+=\langle 2,+1\rangle\in D$，$y$ 负方向为$y^-=\langle 2,-1\rangle\in D$，等等。同时，我们用特殊的维度 0 标记注入和排出：$0^+=\langle 0,+1\rangle$为注入方向，$0^-=\langle 0,-1\rangle$为排出方向。设$d\in D$，我们用函数$neighbor(u,d)$得到节点 u 在 d 方向上下一跳的输入队列。

在本文的设计中，路由器的输入队列被划分为基于报文大小的子队列（结构上等同于一个虚拟通道（Virtual Channel, VC）容纳一个报文，但报文不区分子队列即报文可以进入任何子队列），以消除报文的头阻塞（Head-of-Line, HOL）效应并提升路

由器的性能。然而，当子队列数目较大时，这种设计也引入了更为复杂的仲裁逻辑和交叉开关分配过程。在实际应用中，可以通过在每个时钟周期内仅检查所有子队列请求的一个子集，并且均匀轮换所有输入队列中的各个子队列，从而降低这一复杂度。因此，尽管增加了额外的仲裁阶段，路由器通过延迟可能会有所增加，但路由器的时钟频率和吞吐量依然能够保持稳定。此外，只要输入队列中的所有报文在开关仲裁过程中有均等的机会，路由器就能维持无头阻塞状态。在这种状态下，一个走完所有路由步的报文最终总能进入排出队列被吸收。

在以下的讨论中，我们假设所有路由器均为无头阻塞的，即每个路由器的输入队列都被划分为基于报文大小的子队列。这是我们提出的无死锁路由算法保证正确性的必要条件。

3.2 网络状态

为了定义网络的动态行为，我们首先建模网络的状态。

定义 3. 设 P 为报文的集合，且$\forall p\in P$，$dest(p)$为其目的节点。互连网络$\langle N,Q\rangle$的配置集合$C=\{c:P\mapsto Q\}$是一个函数集合，每个函数将一个报文映射到一个队列。$c(p)$标示在配置 c 下，报文 p 所在的队列。

因此，给定一个配置$c\in C$，其逆函数$c^{-1}=Q\mapsto \wp(P)$，其中$\wp(.)$表示幂集，即所有队列$q\in Q$中包含的报文的集合。设$cap(q)$表示队列 q 的容量（按报文数量计），则我们始终有：

$$num(c^{-1}(q))\leqslant cap(q) \tag{1}$$

3.3 路由算法

路由算法确定报文从源节点到目的节点在网络中传输的路径。图 2 为 DOR 路由算法的路由过程示例，图中黑色圆点为源节点，红色圆点为目标节点，带箭头的实线为从源节点到目标节点的路径。路由算法通常被分解为三个组件：路由函数、流量控制函数和选择函数。

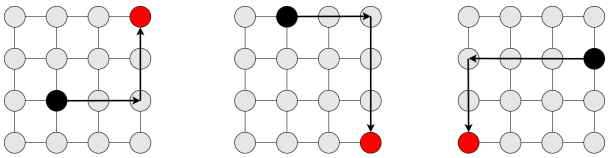

图 2 维序路由算法的路由过程

定义 4. 路由函数是一个映射 R：$N\times N\mapsto \wp(Q)$，其中对于每个位于当前节点 u 且目的地为 v 的报文，$R(u,v)$给出了该报文可以被路由到的备选队列集合。这里需要注意是可能会有多个候选队列。

定义 5. 流量控制函数是一个映射 F：$C\times P\times Q\mapsto\{True,False\}$，在配置$c\in C$下，$\forall p\in P$，$F(c,p,q)$决定是否允许将报文移动到队列$q\in Q$。

流量控制函数限制了报文的移动，包括注入队列中的报文，以及报文从一个节点推进到下一跳节点。与路由函数不同，流量控制函数依赖于当前的网络状态。

定义 6. 选择函数是一个映射 S：$\wp(Q)\mapsto Q\cup\{\perp\}$，它从候选队列集合中选择一个或零个“获胜”队列。

选择函数的功能是从路由函数确定的候选队列集合中，经过流量控制函数的进一步过滤，为下一跳路由节点选出一个或零个队列。文章[5, 6]的工作主要是优化选择函数。

定义 7. 给定一个路由算法$\langle R,F,S\rangle$及其路由函数、流量控制和选择函数，则其路由选择函数是一个映射$\Re^{\langle R,F,S\rangle}$：$C\times P\mapsto Q\cup\{\perp\}$。对于每个报文$p\in P$，在配置$c\in C$下，$p$ 的当前节点为 u，当前队列$\langle u,d\rangle=c(p)$，且其目的节点$v=dest(p)$，我们有：

$$\Re^{\langle R,F,S\rangle}(c,p)=S(q\in R(u,v)\mid F(c,p,q)=True) \tag{2}$$

（其中符号“|”表示条件，即保证流控函数为真。）

3.4 网络行为

为了定义网格网络的动态行为，需要对网络行为进行建模。

本文将网络在路由算法$\langle R,F,S\rangle$下的行为建模为一个自动机$\langle C,c_0,T:\ C\mapsto C\rangle$，其中状态是配置。在初始状态 c_0 中，所有报文都被放置在注入队列中：$\forall p\in P,\exists u\in N,c_0(p)=\langle u,0^+\rangle$。状态转换对应于报文从一个队列移动到另一个队列。

定义 8. 一个合法的配置序列$C_L=\{c_0,c_1,...,c_i,...\}\subseteq\wp(C)$，具有初始配置$c_0$，在路由算法$\langle R,F,S\rangle$下，是一组配置，其中：$\forall i,\exists p'\in P$且$\exists q=\Re^{<R,F,S>}(c_i,p')\neq\perp$，使得：

$$c_{i+1}(p)\triangleq\begin{cases}q, & if\ p=p'\\ c_i(p), & otherwise\end{cases} \tag{3}$$

C_L 描述了网络状态在每个周期内如何根据报文的位置演变。

例如，假设网络中有两个 y 方向相邻节点u_1、

u_2，在初始配置c_0下，报文p_1、p_2都在节点u_1的注入队列q_1中，其中报文p_1的流动方向为x正方向，报文p_2的流动方向为y正方向，并且设定当前路由算法$\langle R,F,S\rangle$规则为向y方向流动。所以在第二个周期时：$c_1(p_1)=c_0(p_1)$（即报文p_1未移动）；$c_1(p_2)=q_2$（即报文p_2注入至节点u_2的队列q_2）。

定义 9. 给定一个合法的配置$c_i\in C_L$，一个队列$q\in Q$，并令

$j=\max(\{l\mid l\leqslant i \ such\ that\ num(c_l^{-1}(q))>num(c_{l-1}^{-1}(q))\})$，$last:=C\times Q\mapsto P\cup\{\perp\}$定义如下：

$$last(c_i,q)\triangleq\begin{cases}c_j^{-1}(q)-c_{j-1}^{-1}(q), & if\ j>0\\ \perp, & otherwise\end{cases}\tag{4}$$

换句话说，$last(c_i,q)$表示在网络处于配置c_i时，进入队列q的最新报文。

例如，延续定义 8 示例的场景，假设合法配置序列$C_L=\{c_0,c_1,c_2,c_3\}$，对于各配置下队列q_2的报文情况如下：

1) 配置c_0：$c_0^{-1}(q_2)=\varnothing$（即此时队列$q_2$无报文流入），所以$num(c_0^{-1}(q_2))=0$；
2) 配置c_1：$c_1^{-1}(q_2)=\{p_2\}$（即此时队列q_2只有报文p_2流入），所以$num(c_1^{-1}(q_2))=1$；
3) 配置c_2：$c_2^{-1}(q_2)=\{p_2,p_3\}$（即此时队列$q_2$新增报文$p_3$流入），所以$num(c_2^{-1}(q_2))=2$；
4) 配置c_3：$c_3^{-1}(q_2)=\{p_2,p_3\}$（即此时队列$q_2$无报文流入），所以$num(c_3^{-1}(q_2))=2$。

现计算$last(c_3,q_2)$，即获取在网络处于配置c_3时，进入队列q_2的最新报文。

由于$num(c_3^{-1}(q_2))=num(c_2^{-1}(q_2))=2$，不满足定义条件，且$num(c_2^{-1}(q_2))>num(c_1^{-1}(q_2))$，故$l=2$,由此可得$j=2$。

由公式(4)得：$last(c_3,q_2)=c_2^{-1}(q_2)-c_1^{-1}(q_2)$，而从前面可以很容易得到两者差集为$\{p_3\}$。所以当网络处于配置$c_3$时，进入队列$q_2$的最新报文是$p_3$。

4 D2RA 路由算法

我们将维度气泡路由算法和维序路由算法结合起来，设计了一种新的混合路由算法，称为D2RA。我们定义一对路由和流量控制函数来形式化 D2RA 路由算法。首先，我们定义最短路径完全适应性路由函数为$R_{\min}$：$N\times N\mapsto P(Q)$。

$$R_{\min}(u,v)\triangleq neighbors(u,RDD(u,v))\tag{5}$$

其中$u,v\in N$，u为当前节点，v为目标节点，且$RDD(u,v)=\{\langle i,a\rangle\in D\mid sign(v-u)_i=a\}$是剩余维度方向的集合，$i\in[1,2,...,n]$，$a\in\{-1,+1\}$。函数$sign(v-u)_i$获取目标节点$v$与当前节点$u$在维度$i$上的坐标值差的符号位，若差值为正，$a$为$+1$；若差值为负，$a$为$-1$；若差值为 0，意味维度$i$没有剩余路由步。例如：$RDD(u,v)=\{x^+,y^+\}$意味着报文在$x^+$，$y^+$方向均剩余路由步。注意：这里用的是*neighbors*而不是*neighbor*，表示节点u在不同维度方向上下一跳的输入缓冲队列集合。

对于流量控制函数，我们关注基于“信用”的流量控制机制，该机制基于下一跳路由器队列中可用缓冲区缓存空间的数量做出流控决策。

定义 10. 设$credit(c,q)=cap(q)-num(c^{-1}(q))$表示队列$q\in Q$在配置$c\in C$下可用的缓冲区缓存空间数量。其中$credit(c,q)$的单位是报文。由债务函数$DEBT:C\times P\times Q\mapsto Z$引起的基于信用的流量控制函数$F_c^{DEBT}$，其定义如下：对于$\forall p\in P,q\in Q$，

$$F_c^{DEBT}(c,p,q)=\begin{cases}True, if\ debt(c,p,q)\leqslant credit(c,q)\\ False, \quad otherwise\end{cases}\tag{6}$$

这里需要注意，基于信用的流量控制函数完全由其债务函数来表征。

定义 11. 维度气泡流量控制函数F_c^{DBFC}是一个由债务函数$DBFC:C\times P\mapsto Z$引起的基于信用的控制函数，其中对于配置$c\in C$下的一个报文$p\in P$，位于队列$\langle u,d\rangle=c(p)$，并且目的地为$v=dest(p)$，我们有：

$$DBFC(c,p)\triangleq num(RDD(u,v))\tag{7}$$

我们定义债务函数$DBFC$等于报文的剩余维度方向集合的元素个数。例如：若报文p在配置c下在x^+，y^+方向均剩余路由步即：$RDD(u,v)=\{x^+,y^+\}$，那么$DBFC(c,p)=2$，这意味着流控函数F_c^{DBFC}在信用大于等于 2 时值为*True*否则为*False*，也就是说报文只有当下一级缓冲队列的空闲报文数（信用）大于等于 2 时才能进入。

定义了$R_{\min}$和F_c^{DBFC}函数后，我们将完全适应性维度气泡路由算法（DBRA）公式化为：

$$DBRA=<R_{\min},F_c^{DBFC},S>\tag{8}$$

同样地，我们可以公式化 *k*-ary *n*-mesh 网络的维序路由（DOR）算法。DOR 算法规定报文完成

某个维度上的所有路由步后才能进入下一个维度，并且哪个维度先走，哪个维度后走是确定的关系。若我们用数字 1 到 n 标示这种维度路由的先后次序，由于表示先后次序的数字与维度号一一映射，若我们用次序号作为维度的逻辑号，那么报文路由过程都可以认为是升序。为了简化讨论，我们假设本文中的 DOR 使用升序路由即先完成维度 1 的路由步，然后走维度 2 最后维度 n。

定义 12. 对于配置$c\in C$下的报文$p\in P$，当前队列$\langle u,d\rangle = c(p)$，目的节点$v = dest(p)$，令$RDS(u,v) = \{i \mid v_i - u_i \neq 0, i\in[1,2,...,n]\}$为剩余维度的集合，我们用$num(RDS(u,v))$表示这个集合的元素个数。例如：若报文在维度 1 和 2 上还有路由步即$RDS(u,v) = \{1,2\}$，那么$num(RDS(u,v)) = 2$。函数$\min(RDS(u,v))$得到集合$RDS(u,v)$中最小元素即最小剩余维度；令$g_{\min} = \min(RDS(u,v))$，例如：$RDS(u,v) = \{1,2\}$，那么$g_{\min} = 1$。定义$a = sign(v-u)_{g_{\min}}$和$d' = \langle g_{\min}, a\rangle$，$d'\in D$，$a\in\{-1,+1\}$，我们定义维序路由函数为：

$$R^{DOR}(u,v) \triangleq neighbor(u,d') \tag{9}$$

令队列$q = neighbor(u,d')$即队列q为节点u在d'方向上下一跳的缓冲队列，我们为 *DOR* 定义流量控制函数：

$$F_c^{DOR}(c,p,q) = \begin{cases} True, & if\ 1 \leqslant credit(c,q) \\ False, & otherwise \end{cases} \tag{10}$$

我们定义 D2RA 算法的路由函数为：

$$R^{D2RA}(u,v) = R_{\min}(u,v) \cup R^{DOR}(u,v) \tag{11}$$

在 D2RA 策略中，为了保持路由路径的多样性，路由过程尽可能按照 DBRA 路由策略进行报文的路径选择。

定义 13. 给定一个 *k*-ary *n*-mesh 网络$\langle N,Q\rangle$，配置$c\in C$下的报文$p\in P$，当前队列$\langle u,d\rangle = c(p)$，目的节点$v = dest(p)$，$q$是一个队列且$q\in R^{DOR}(u,v)\cup R_{\min}(u,v)$，我们定义 D2RA 的流量控制函数：

$$F_c^{D2FC}(c,p,q) = \begin{cases} True, & if(F_c^{DBFC} | F_c^{DOR}) = True \\ False, & otherwise \end{cases} \tag{12}$$

D2RA 路由选择函数定义为：

$$\Re^{D2RA} = S(\{q_0\in R_{\min} \mid F_c^{DBFC}(c,p,q_0) = True\} \cup \{q_1\in R^{DOR} \mid F_c^{DBFC}(c,p,q_0) = False\ and\ F_c^{DOR}(c,p,q_1) = True\}) \tag{13}$$

在流量控制和选择路由函数准备就绪后，我们就可以将 D2RA 公式化：

$$D2RA = < R^{D2RA}, F_c^{D2FC}, \Re^{D2RA} > \tag{14}$$

为了进一步阐述 D2RA 路由算法，我们用伪代码对它进行描述。

算法 1. D2RA 路由算法.

输入：报文的当前节点 $u = <u_1, u_2, \cdots, u_n>$，
报文的目标节点 $v = <v_1, v_2, \cdots, v_n>$
输出：下一跳目标队列 q

```
1.  E := {} //初始化可选队列集合 E 为空
2.  z = num(RDD(u, v))
    //计算报文的剩余维度方向集合的元素个数
3.  IF (t == 0)
4.      q := <u, 0> //报文完成路由将被吸收
5.  END IF
6.  ELSE //报文没有完成路由
7.  g := min (RDS(u,v)) //g 是最小维度号
8.      FOR i := 1 TO n DO //从维度 1 到维度 n 循环
        //该循环计算报文可以选择的目标队列
9.          a := sign(v −u)i
10.         IF a == +1 OR a == -1
11.             d := <i, a> //在 d 方向上剩余路由步
12.             q′ := neighbor(u, d)
                // q′为 u 在 d 方向上下一跳的缓冲队列
13.             m := credit(q′)
                // m 为队列 q′的空闲报文数
14.             IF m ≥ z //满足 DBRA 流控要求情况
15.                 E := E+{q′. high_priority}
16.             END IF
17.           ELSE IF g==i and m≥1   // meet with DOR
18.             E := E+{q′. low_priority}
19.           END ELSE IF
20.       END IF
21.   END FOR
22.     q := Select(E) //选择目标队列
23. END ELSE
    //未到目标节点的报文完成下一跳队列选择
```

算法 1.描述了 D2RA 算法的路由选择策略包括 DBRA 路由和 DOR 路由。其中，D2RA 算法赋予 DBRA 完全适应性路由策略高的优先级，保证了算法路由选择的多样性。另外，算法中路由的选择并不区分虚信道，DBRA 路由和 DOR 路由共享缓冲资源，这是和传统 Duato 策略不同的地方。同时，

从算法 1 可以看出：D2RA 算法的流控阀值是动态变化的。算法 1 首先判断下一步缓冲区是否满足 DBRA 流控条件（算法 1 中第 14 句），其中变量 m 和 z 的值都是不断变化的；若满足则按照 DBRA 路由选择队列；否则判断下一步缓冲区的维度是否满足 DOR 路由同时流控是否满足 DOR 流控要求。D2RA 算法根据不同的流控条件，动态地分配缓冲资源，减少网络拥塞，实现网络高效的通信。

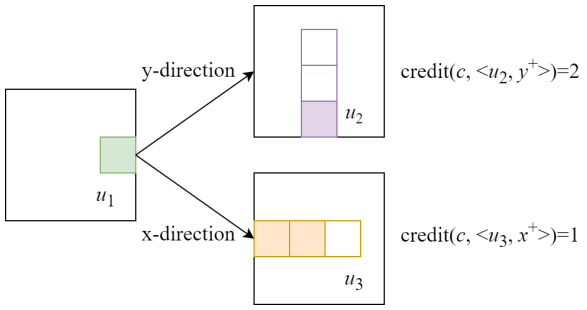

图 3 D2RA 路由选择示意图

图 3 是 D2RA 算法的一个例子。图 3 中，节点 u_1 的一个报文在 x^+和 y^+方向剩余路由步，因此 z=2。首先，在 x^+方向上，报文的下一步将进入的节点为 u_3，u_3 在 x^+方向上 $credit(c,\langle u_3,x^+\rangle)=1<z$，因此不满足 DBRA 流控，但 x 是报文剩余的最小维度且 $credit(c,\langle u_3,x^+\rangle)=1$满足 DOR 流控。因此队列 $\langle u_3,x^+\rangle$作为低优先级进入选择队列集合中。

再看 y^+方向上，报文的下一步将进入的节点为 u_2，u_2 在 y^+方向上 $credit(c,\langle u_2,y^+\rangle)=2=z$，满足 DBRA 流控。因此队列$\langle u_2,y^+\rangle$作为高优先级进入选择队列集合中。这样，最终报文将选择队列 $\langle u_2,y^+\rangle$ 作为下一步路由目标队列。若 $credit(c,\langle u_2,y^+\rangle)<2$，那么队列$\langle u_3,x^+\rangle$将成为报文的下一步路由目标队列。这个例子显示，D2RA 算法比 DBRA 算法提供了更多的路由选择。

在硬件开销方面，对照 DBRA，D2RA 只需增加 DOR 路径的仲裁，不需要增加额外的缓冲空间。我们通过图 4 详细分析比较 DBRA 和 D2RA 仲裁生成逻辑。

图 4(a)是 DBRA 输入端口生成第 i 维仲裁请求的逻辑。其中，u_i 和 v_i 进行比较若不相等输出逻辑 1 否则输出逻辑 0；同时报文的剩余维度数 z 和 i 维方向的下一步缓冲区的信用值进行比较，若信用值大于等于 z 则输出 1 否则输出 0。这两个比较器的输出经过与门生成第 i 维仲裁请求信号。图 4(b)和(c)是 D2RA 输入端口生成第 i 维仲裁请求的逻辑。其中，(c)图中 u_i 和 v_i 进行比较生成信号 r_i；(b)图中 r_1 是 u_1 和 v_1 进行比较生成的信号，其它信号类似。r_i 和 r_1 到 r_{i-1} 的反相值一起与操作生成选择信号 s_i，这意味着 r_1 到 r_{i-1} 这些信号中只要有 1 的逻辑那么 s_i 的值为 0。这个逻辑判断维度 i 是否为最低维度。(c)图中的选择器当$s_i=1$时输出 1 表示选择 DOR 流控；当$s_i=0$时输出 z 表示选择 DBRA 流控。选择器的输出和信用值进行比较，若信用值大于等于选择器的输出则输出 1 否则输出 0。然后这个输出值和 r_i 相与生成第 i 维仲裁请求信号。从上面的对比可知：D2RA 的仲裁请求逻辑相对 DBRA 需要增加一个大与门（采用多级二输入与门实现）用于生成选择信号，还需要增加一个选择器用于确定选用 DBRA 流控还是 DOR 流控。除了仲裁逻辑有变化外，D2RA 算法和 DBRA 算法的硬件实现都相同。因此，D2RA 算法的硬件实现只要在 DBRA 算法基础上增加少量硬件就可实现。

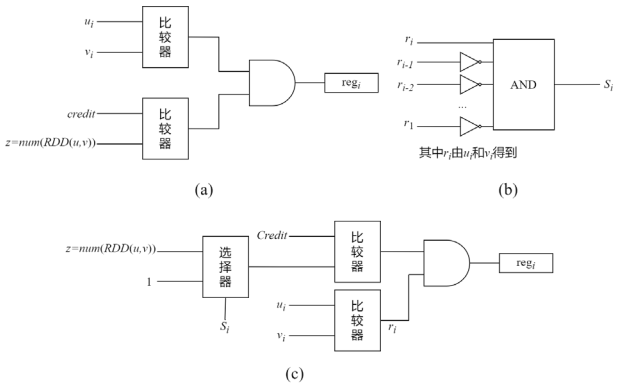

图 4 D2RA 与 DBRA 仲裁请求逻辑对比（(a)DBRA 输入端口生成第 i 维仲裁请求的逻辑；(b)和(c)是 D2RA 输入端口生成第 i 维仲裁请求的逻辑）

DBRA 路由和 DOR 路由单独分别布置在网络中，路由都没有死锁。然而这并不能推断出 DBRA 路由和 DOR 路由同时布置在一套网络也不会出现死锁。例如在同一套网络设置一个虚信道的情况下，同时采用 DOR 升序路由和降序路由就将出现图 5 所示的环结构，产生死锁现象。

同时，我们对比 D2RA 和 DBRA 算法路由，发现采用 DBRA 算法的所有网络边界节点总有空闲报文空间，而 D2RA 算法没有这个特征。

图 6(a)中，网络采用 DBRA 算法，在 X^+方向的边界节点 u_d 总会有空闲缓冲区保证只剩 X^+方向路由步的报文流动；图 6(b)中，网络采用 D2RA 算法，在 X^+方向的边界节点 u_d 的最后报文空间可以被按照 DOR 策略流入的报文占用。实际的证明场景比 DBRA 复杂很多。

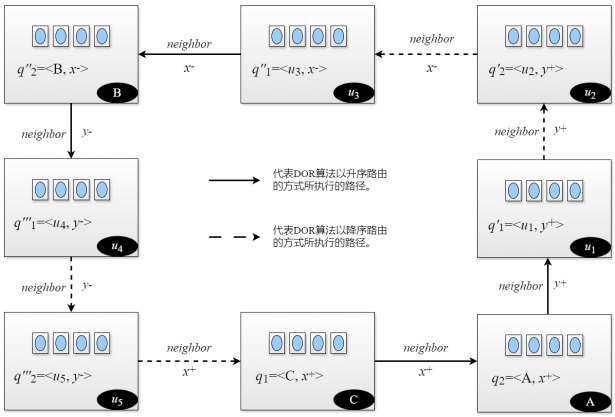

图 5 一个路由死锁场景

综上所述，我们需要理论证明 D2RA 算法的无死锁特性，给算法的合理性提供重要的理论支撑。

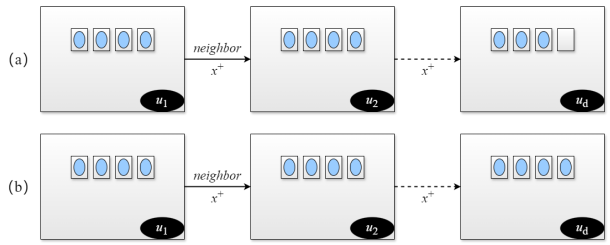

图 6 DBRA 算法和 D2RA 算法在边界队列中的区别（(a)为 DBRA 算法，(b)为 D2RA 算法）

5 D2RA 死锁避免机制

这一节的目标是证明D2RA的路由算法在任何 k-ary n-mesh 网络中都能避免死锁。这个结论将在**定理 1.**中正式建立，并通过五个支持性引理来证明。

5.1 相关表述

为清楚表述并证明 D2RA 的死锁避免机制，在此采用图论术语进行表述，相关表述如下。

表述 1. 令网络 $G=(V,E)$ 为 k-ary n-mesh 网络，其中顶点集 V 代表网络中所有路由节点集 N，边集 E 为包含所有相邻节点间的通道集合。

表述 2. 令有向图 $\mathcal{G}=(V_{CDG},E_{CDG})$ 为通道依赖图（Channel Dependency Graph, CDG），其中顶点集 $V_{CDG}=E$，边集 $(e_i,e_j)\in E_{CDG}$ 当且仅当报文可能从通道 e_i 直接进入通道 e_j。

表述 3. 每个节点 u 在维度方向 d 的输出通道 e 严格对应一个输出队列 q。

5.2 关键引理

引理 1. 给定一个带有D2RA路由算法的 k-ary n-mesh 网络 $\langle N,Q\rangle$，对于任意边界节点 u，其边界方向上的输出通道 e_u 在 CDG 中满足：

$$\nexists\, e_v\in V_{CDG}\ \ s.t.\ \ (e_u,e_v)\in E_{CDG} \tag{15}$$

引理 1.表述的意思是：采用 D2RA 路由算法的 mesh 网络中，假设节点 u 是 d 方向上的边界节点，报文 p 在节点 u 沿方向 d 的队列中，那么方向 d 肯定不是报文 p 的剩余方向。例如图 1 中，节点 B 是 y^+ 方向的边界节点，它的 y^+ 方向缓冲队列中的报文不会再有 y^+ 方向的路由步，即节点 B 在 y^+ 方向上的通道无出边。

证明.

设 $u=\langle u_1,u_2,...,u_i,...,u_n\rangle$ ， $v=dest(p)=\langle v_1,v_2,...,v_i,...,v_n\rangle$， $i\in[1,2,...,n]$。如果 $d=\langle i,+1\rangle$ 且节点 u 是 d 方向上的边界节点，那么 $u_i=k$。若 $d\in RDD(u,v)$，那么 $v_i>u_i$，因此 $v_i>k$。

如果 $d=\langle i,-1\rangle$ 且节点 u 是 d 方向上的边界节点，那么 $u_i=1$。若 $d\in RDD(u,v)$，那么 $v_i<u_i$，因此 $v_i<1$。这两种情况下 v_i 的值都与**定义 2.**矛盾，因此 $d\notin RDD(u,dest(p))$，即报文 p 没有 d 方向的路由步。

证毕.

引理 2. 给定一个带有 D2RA 路由算法的 k-ary n-mesh 网络 $\langle N,Q\rangle$，存在报文 $p\in P$，位于节点 u，$RDD(u,dest(p))=\{d\}$，对于 $\forall j\geq i$，$\forall q\in R^{D2RA}(u,dest(p))$ 有 $F_c^{D2FC}(c_j,p,q)=False$，设队列 $q'=neighbor(u,d)$，则 $credit(c_j,q')=0$。

换句话说，对于**引理 2.**，如果一个报文只在维度方向 d 上还有路由步，并且在当前路由节点上停滞，那么下一跳路由器队列 q' 应该已满。

证明.

因为 $RDD(u,dest(p))=\{d\}$，那么无论报文 p 下一步按 DBRA 路由还是按 DOR 路由，队列 q' 都是它的目标队列，因此 $q'\in R^{D2RA}(u,dest(p))$；由于对 $\forall q\in R^{D2RA}(u,dest(p))$ 有：$F_c^{D2FC}(c_j,p,q)=False$，所以 $F_c^{D2FC}(c_j,p,q')=False$，意味着对于 $\forall j\geq i$，$credit(c_j,q')<DBFC(c_j,p)=1$（因为 $RDD(u,dest(p))=\{d\}$ 意味着只要 q' 有一个空闲报文空间，在没有 HOL 问题前提下，报文 p 将进入），所以 $credit(c_j,q')=0$。

证毕.

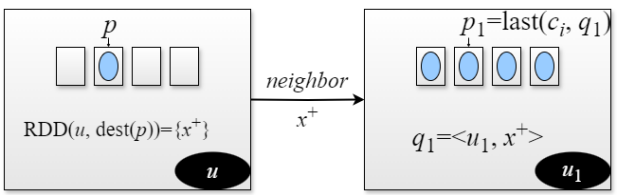

图 7 引理 2 的一个场景

图 7 中，报文 p 只在 X^+ 方向上剩余路由步，如果报文 p 始终不能离开节点 u，那么队列 q_1 应该总是满的。接下来，我们将分析最后进入队列 q_1 的报文 p_1 的特征。

引理 3. 给定一个带有 D2RA 路由算法的 k-ary n-mesh 网络 $\langle N,Q\rangle$，在节点 u 的输出通道 e_u 存在队列 $q=\langle u,d\rangle$，$d=\langle g_{\min},a\rangle$，$g_{\min}\in[1,2,...,n]$（$g_{\min}$ 指的是最小维度数），$a\in\{-1,+1\}$。对于 $\forall j\geq i$，$credit(c_j,q)=0$，$p=last(q)$，即通道 e_u 持续饱和（队列 q 一直满状态），报文 p 为最后进入的报文：

1) 如果 p 遵循 DBRA 路由进入 q，则 $RDD(u,dest(p))=\{d\}$；
2) 如果 u_d 是沿方向 d 的边界节点，则 $\min(RDS(u_d,dest(p)))\geq g_{\min}+1$；
3) 如果 u 不是沿方向 d 的边界节点，则 $\min(RDS(u,dest(p)))\geq g_{\min}$；
4) 存在通道序列 $e_{u_1}\to e_{u_2}\to\cdots\to e_{u_d}$ 满足：
   a) 有通道持续饱和（队列处于满状态）；
   b) 节点 u_d 是沿方向 d 的边界节点；
   c) 队列 $q_d=\langle u_d,d\rangle$，使得 $credit(c_j,q_d)=0$ 且 $\min(RDS(u_d,dest(q_d)))\geq g_{\min}+1$。

引理 3.的意思是：节点 u 在 d 方向上的队列 q 一直处于满状态，报文 p 是最后进入队列 q 的报文，那么：

1) 如果 p 是遵循 DBRA 路由进入的队列 q，那么报文 p 只剩下 d 方向上的路由步；
2) 如果节点 u_d 是 d 方向上的边界节点，那么报文 p 最小的剩余维度大于等于 $g_{\min}+1$；
3) 如果节点 u 不是 d 方向上的边界节点，那么报文 p 最小的剩余维度大于等于 $g_{\min}$；
4) 以节点 u 为起点沿 d 方向上的队列中，有队列一直处于满状态并且队列的最后报文的最小的剩余维度大于等于 $g_{\min}+1$。注意节点 u 在 d 方向上的队列 q 也是其中之一。

证明.

1） 因为对于 $\forall j\geq i$，$credit(c_j,\langle u,d\rangle)=0$，所以 p 不可能已经完成了路由，否则，p 一定会被网络吸收（因为不存在 HOL 问题），从而 $\exists j'\geq i$，使得 $credit(c_{j'},\langle u,d\rangle)>0$。如果报文 p 遵循 DBRA 路由进入队列 $q=\langle u,d\rangle$，则 p 只会在 d 方向上剩余路由步，否则在进入 q 之前报文 p 至少在两个方向上剩余路由步，按照 DBRA 流控，至少有 2 个空闲报文空间，报文才能进入 q，这将阻止 p 占据 q 中的最后一个空位。因此，$RDD(u,dest(p))=\{d\}$，因为 $d=\langle g_{\min},a\rangle$，意味着 $RDS(u,dest(p))=\{g_{\min}\}$。

2） 如果报文 p 遵循 DOR 路由进入队列 q，由于 DOR 使用升序路由，因此在报文 p 进入队列 q 之前，报文 p 的最小剩余维度为 $g_{\min}$，报文 p 进入 q 之后，报文 p 有可能完成维度 $g_{\min}$ 上的路由步（报文 p 进入 q 之前，在维度 $g_{\min}$ 上的剩余路由步为 1），因此 p 的最小剩余维度应该大于等于 $g_{\min}$ 即 $\min(RDS(u,dest(p)))\geq g_{\min}$。此外，根据**引理 1.**，如果 u 是 d 方向上的边界节点，那么队列 q 中的报文在 d 方向上不再有路由，因此 p 的最小剩余维度大于或等于 $g_{\min}+1$；如果 u 不是 d 方向上的边界节点，则 p 的最小剩余维度大于或等于 $g_{\min}$。

3） 由于 $credit(c_j,\langle u,d\rangle)=0$ 且 $\min(RDS(u,dest(p)))\geq g_{\min}$，因此如果 $\min(RDS(u,dest(p)))\geq g_{\min}+1$，则队列 $q=\langle u,d\rangle$ 满足需求，因此结论 4 是有效的，否则 $\min(RDS(u,dest(p)))=g_{\min}$。设队列 $q'=neighbor(u,d)$，那么 $q'=R^{DOR}(u,dest(p))$，报文 p 按照 DOR 路由将进入队列 q'，由于对于 $\forall j\geq i$，$credit(c_j,\langle u,d\rangle)=0$ 意味着 p 永远没有机会离开队列 $\langle u,d\rangle$，所以 $F_c^{D2FC}(c_j,p,q')=False$，这意味着报文 p 永远不能进入队列 q'，另一方面，由于只要队列 q' 有一个空闲报文空间，报文 p 按照 DOR 路由可以进入队列 q'，因此 $credit(c_j,q')=0$。令报文 $p'=last(q')$，根据 2）的结论可以推出 $\min(RDS(u',dest(p')))\geq g_{\min}$。我们对以节点 u' 为起点的 d 方向上的节点重复该过程，如果所有中间队列都是信用为 0 即队列已满且最小剩余维度为 $g_{\min}$ 无法满足结论 4，图 8 描述了这个场景。那么我们设 d 方向上边界节点为 u_d，由于前一级的信用一直为 0，因此 $credit(c_j,\langle u_d,d\rangle)=0$；设报文 $p_d=last(\langle u_d,d\rangle)$，根据**引理 1.**，报文 p_d 不再有 d 方向上的路由步，因此 $\min(RDS(u_d,dest(p_d)))\geq g_{\min}+1$。

证毕.

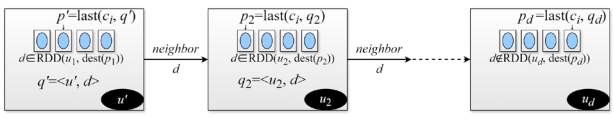

图 8 引理 3 结论 4 的一个场景

引理 3.的结论 4 中，我们从维度 $g_{\min}$ 推到了维度 $g_{\min}+1$，类似地，我们可以从维度 $g_{\min}+1$ 推到

维度 $g_{\min}+2$ 直到最高维度 n，形成有趣的递推过程，后面的证明中将用到这种方法。

引理4. 给定一个带有D2RA路由算法的 k-ary n-mesh 网络 $\langle N,Q\rangle$，$\forall p\in P$ 位于节点 u 上，如果 $RDS(u,dest(p))=\{n\}$（n 为最高维度），那么 $\exists j\geq i$ 和 $q\in R^{D2RA}(u,dest(p))$，使得 $F_c^{D2FC}(c_j,p,q)=True$。

引理4.表明，如果一个报文仅在维度 n 有剩余路由步且 n 是最高维度，则该报文最终会到达其目的地并被网络吸收。

D2RA 路由算法是最短路由策略，按照 D2RA 算法路由的报文若总能移动，那么报文最终将到达目标节点。

因此，若维度 n 的缓冲队列中的报文总可以流动，那么**引理 4.**成立。

我们以图 1 中的二维 mesh 为例，进行分析。y 维是二维 mesh 的最高维度。

图 9 展示了从节点 u_1 到节点 B 的过程中各节点 y^+ 方向上的队列情况。正如图 1 中所示，节点 B 是 y^+ 方向上的边界节点。

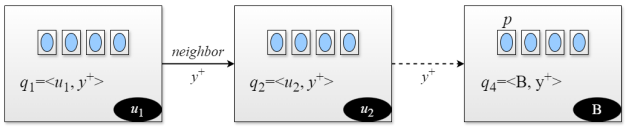

图 9 y^+方向上的队列

q_4 是 B 节点在 y^+ 方向上的队列，报文 p 占据了 q_4 的最后一个报文空间。如果报文 p 在进入队列 q_4 前还有 x 方向上的路由，那么按照 DOR 路由，报文 p 没有完成 x 方向上的路由不能进入队列 q_4；同样由于报文 p 在 x,y 方向都剩有路由，按照 DBRA 路由，队列 q_4 必须有 2 个以上的空闲报文空间，报文 p 才能进入队列 q_4。因此，报文 p 在进入队列 q_4 前不会有 x 方向上的路由。另一方面，报文 p 进入队列 q_4 后，不会再有 y^+ 方向上的路由（**引理 1.**），因此报文 p 将会被吸收。因此队列 q_4 总会出现空闲报文空间，这样在没有 HOL 前提下，只剩 y^+ 方向上路由步的报文总会流动。

下面我们采用反证法，先假设报文不能移动，然后推出矛盾的结论。

证明.

我们先假设存在只剩维度 n 上路由步的报文永远停留在当前节点，不能流出，然后证明这种情况与已有的引理矛盾。

假设 $\exists p\in P$，使得 $RDS(u,dest(p))=\{n\}$，对于 $\forall j\geq i$，$\forall q\in R^{D2RA}(u,dest(p))$，$F_c^{D2FC}(c_j,p,q)=False$ 即报文 p 永远不能移动。由于 $RDS(u,dest(p))=\{n\}$ 意味报文 p 只在维度 n 上还有路由步，这也意味报文 p 只在一个维度方向上还有路由步，因此我们可以设 $RDD(u,dest(p))=\{d\}$，队列 $q'=neighbor(u,d)$ 且队列 $q'=\langle u',d\rangle$，节点 u' 是节点 u 的相邻节点。根据**引理 2.**可以得到 $credit(c_j,q')=0$。令报文 $p'=last(q')$，如果 p' 按照 DBRA 路由进入 q'，则 $RDD(u',dest(p'))=\{d\}$（**引理 3.**）。

如果报文 p' 按照 DOR 路由进入 q'，由于维度 n 是最高维度且 DOR 使用升序路由因此 d 是报文 p' 路由的最后一个方向。而且，由于 $\forall j\geq i$，$credit(c_j,q')=0$，所以 p' 不可能完成路由。因此，我们推出 $RDD(u',dest(p'))=\{d\}$。事实是，对于 $\forall j\geq i$，$credit(c_j,q')=0$ 意味着报文 p' 没有机会离开队列 q'。令 $q''=neighbor(u',d)$，队列 q'' 是报文 p' 下一跳准备进入的队列，因此我们可以得出结论 $F_c^{D2FC}(c_j,p',q'')=False$。

如果我们对以节点 u' 为起点的 d 方向上节点重复该过程，我们将找到 d 方向的边界节点 u_d 和队列 $q_d=\langle u_d,d\rangle$，且 $credit(c_j,q_d)=0$，设报文 $p_d=last(q_d)$，由于报文 p_d 永远不能离开队列 q_d，因此报文 p_d 还存在路由步，根据前面的分析 d 方向是报文 p_d 唯一可能的路由方向，因此 $d\in RDD(u_d,dest(p_d))$。这与**引理 1.**矛盾。因此，假设不成立，这意味着**引理 4.**的结论有效。

证毕.

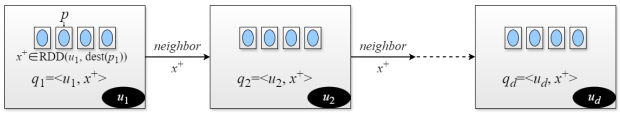

图 10 引理 5 在 2D *mesh* 中的一个场景

引理5. 给定一个带有D2RA路由算法的 k-ary n-mesh 网络 $\langle N,Q\rangle$，$\forall c_i\in C_L$，$\forall p\in P$ 位于节点 u 的队列 $c_i(p)$ 中，如果 $DBFC(c_i,p)=1$，那么 $\exists j\geq i$ 和 $q\in R^{D2RA}(u,dest(p))$，使得 $F_c^{D2FC}=(c_j,p,q)=True$。

引理 5.表明，D2RA 保证若一个报文只在一个维度方向上剩余路由步，那么这个报文最终会到达其目的地并被网络吸收。

图 10 描述了引理 5 在二维 *mesh* 中的一个场景。只剩 x^+方向路由步的报文 p 驻留在节点 u_1 的队列 q_1 中。当前 x^+方向上的队列都处于满状态。我们分析 x^+方向的边界队列 q_d。由于 q_d 的最后报文 *last*(c_j, q_d)不会再有 x^+方向路由步（引理 1），并且它没有

完成路由步，那么它只会剩余 y 方向的路由步。由于 y 方向是二维 *mesh* 最高维度，根据引理 4，它可以到达它的目标节点，因此 q_d 就会有空闲报文空间保证 x^+方向的报文流动，这样只剩 x^+方向路由步的报文 p 可以到达目标节点。

证明.

由于 $DBFC(c_i,p)=1$，这意味着报文 p 仅在一个维度上有剩余路由步。采用归纳法证明，设 $RDS(u,dest(p))=\{g\}$，通过对报文的剩余维度即 g 进行归纳证明。**引理 4.**中已证明 $g=n$，结论成立。假设当 $g>g_{\min}$，$g_{\min}\in[0,1,...,n-1]$ 时，**引理 5.** 成立，即 $\exists j\geq i$ 和 $q\in R^{D2RA}(u,dest(p))$，使得 $F_c^{D2FC}(c_j,p,q)=True$。

换句话说，如果剩余维度大于 $g_{\min}$，且只有一个剩余维度的报文最终可以朝目标节点移动。

对于 $g=g_{\min}$ 的情况，我们通过反证法来证明。

$\exists p\in P$，且 $g=g_{\min}$，满足 $\forall j\geq i$，$\forall q\in R^{D2RA}(u,dest(p))$，使得 $F_c^{D2FC}(c_j,p,q)=False$。

由于 $g=g_{\min}$，设报文 p 唯一剩余维度方向为 $d=\langle g_{\min},a\rangle$，$a\in\{-1,+1\}$，且队列 $q'=neighbor(u,d)$。

由于 $credit(c_j,q')=0$（**引理 2.**），$\exists q_1=\langle u_1,d\rangle$ 使得 $credit(c_j,q_1)=0$，且 $\min\left(RDS(u_1,dest(last(q_1)))\right)\geq g_{\min}+1$（**引理 3.**）。

设 $p_1=last(q_1)$，$\min\left(RDS(u_1,dest(p_1))\right)=g'_{\min}$（$g'_{\min}\geq g_{\min}+1$），节点 $v=dest(p_1)$，$a'=sign(v-u_1)_{g'_{\min}}$，$a'\in\{-1,+1\}$，维度方向 $d'=\langle g'_{\min},a'\rangle$，队列 $q_2=neighbor(u_1,d')$。

由于 $g'_{\min}$ 是报文 p_1 剩余维度中的最低维度，按照 DOR 路由策略，队列 $q_2=R^{DOR}(u_1,dest(p_1))$，因此 $q_2\in R^{D2RA}(u_1,dest(p_1))$。

由于 $credit(c_j,q_1)=0$，这意味着 p_1 永远不能离开 q_1，因此 $credit(c_j,q_2)=0$，反之，如果 q_2 中有一个空闲报文空间，p_1 按照 DOR 可以进入 q_2。

由于 $credit(c_j,q_2)=0$，$\exists q_d=\langle u_d,d'\rangle$ 使得 $credit(c_j,q_d)=0$，且 $\min\left(RDS(u_d,dest(last(q_d)))\right)\geq g'_{\min}+1$（**引理 3.**），由于 $g'_{\min}\geq g_{\min}+1$，因此 $\min\left(RDS(u_d,dest(last(q_d)))\right)\geq g_{\min}+2$。

从上面过程我们注意到最小的维度号每次增加 1，如果再重复该过程 $n-g_{\min}-2$ 次，可以得出 $\exists q_{a'}=\langle u_{a'},d_{a'}\rangle$ 使 $credit(c_j,q_{a'})=0$，且 $\min\left(RDS(u_{a'},dest(last(q_{a'})))\right)\geq n$。

由于 n 是最高维度，因此 $RDS(u_{a'},dest(last(q_{a'})))=\{n\}$ 即队列 $q_{a'}$ 的报文 $last(q_{a'})$ 只剩维度 n 上的路由步，由于 $credit(c_j,q_{a'})=0$ 意味队列 $q_{a'}$ 的报文 $last(q_{a'})$ 永远不能离开 $q_{a'}$，这与**引理 4.**矛盾。

这意味着**引理 5.**中的陈述在 $g=g_{\min}$ 时成立。

证毕.

5.3 核心定理

定理 1. 给定一个带有 D2RA 路由算法的 k-ary n-mesh 网络 $\langle N,Q\rangle$，$\forall c_i\in C_L$，$\forall p\in P$，位于节点 u，目标节点为 $v\in N$，$\exists j\geq i$，$\exists q\in R^{D2RA}(u,dest(p))$，使得 $F_c^{D2FC}(c_j,p,q)=True$。

定理 1.表明没有报文会被流量控制永久阻塞；换句话说，所有报文都可以通过 F_c^{D2FC} 的验证而最终到达其目标。

证明.

该证明通过对 $DBFC(c_i,p)$ 的归纳进行。当 $DBFC(c_i,p)=1$，**引理 5.**已表明**定理 1.**成立。假设对于 $DBFC(c_i,p)<z$，$z\in[2,...,n+1]$ 时，**定理 1.** 成立，即 $\exists j\geq i$，$\exists q\in R^{D2RA}(u,dest(p))$ 使得 $F_c^{D2FC}(c_j,p,q)=True$。换句话说，剩余维度方向数为 $z-1$ 或更少的报文最终可以朝目标节点移动。

对于 $DBFC(c_i,p)=z$ 的情况，我们采用反证法证明。假设 $\forall j\geq i$，$\forall q\in R^{D2RA}(u,dest(p))$，使得 $F_c^{D2FC}(c_j,p,q)=False$。

设维度 $g_{\min}=\min(RDS(u,v))$，$a=sign(v-u)_{g_{\min}}$，$a\in\{-1,+1\}$，维度方向 $d=\langle g_{\min},a\rangle$，队列 $q'=neighbor(u,d)$，由于 $g_{\min}$ 是报文 p 剩余维度中的最低维度，报文 p 按照 DOR 路由将进入队列 q'，因此队列 $q'=R^{DOR}(u,dest(p))$，队列 $q'\in R^{D2RA}(u,dest(p))$，从而 $F_c^{D2FC}(c_j,p,q)=False$，意味着 $\forall j\geq i$，$credit(c_j,q')=0$。

因此 $\exists q_1=\langle u_1,d\rangle$ 使得 $credit(c_j,q_1)=0$，且 $\min\left(RDS(u_1,dest(last(q_1)))\right)\geq g_{\min}+1$（**引理 3.**）。

如果我们再次将该过程重复 $n-z-g_{\min}$ 次，我们可以找到队列 $q_{a'}=\langle u_{a'},d_{a'}\rangle$，使得 $credit(c_j,q_{a'})=0$，$\min\left(RDS(u_{a'},dest(last(q_{a'})))\right)\geq g_{\min}+r$（$r>n-z-g_{\min}+1$）即 $\min\left(RDS(u_{a'},dest(last(q_{a'})))\right)>n-z+1$，这意味着集合 $RDS(u_{a'},dest(last(q_{a'})))$ 中最小的元素大于 $n-z+1$ 而最大的元素不会超过 n，因此 $DBFC(c_j,last(q_{a'}))<z$。由于 $credit(c_j,q_{a'})=0$ 意味着队列 $q_{a'}$ 的最后报文不可能离开 $q_{a'}$，这与假设剩

余方向为$z-1$或更少的报文最终能够朝目标节点移动矛盾。

因此，这意味着**定理 1.**对于$DBFC(c_j,p)=z$的情况成立。

证毕.

6 实验评估

本节我们从下面四个方面对D2RA算法进行综合评估。首先，我们对DBRA算法和D2RA算法的性能进行测试，比较算法改进前后的性能差别；然后我们全面比较D2RA算法和Duato算法以及不同优化策略算法的性能；接着，我们评估网络关键参数对D2RA算法的影响；最后，我们分析基于D2RA策略采用决策树实现的算法DeTAR_D2RA算法的性能。

6.1 实验方法

实验采用由Stanford大学开发的BookSim模拟器[14]进行性能评估。该模拟器采用结构化设计并提供大量的典型网络应用。我们在BookSim模拟器上实现了D2RA算法和相关算法以进行比较。

6.2 实验配置

表 2 Booksim配置参数

参数类型	参数配置
网络拓扑	4×4, 8×8, 16×16 2D mesh; 4×4×4 3D mesh
VC 数量	3
VC深度（flits）	8
报文大小	1,5,8 (flits)
流控机制	credit-based , wormhole
仲裁策略	基于优先级，round-robin
通信模式	uniform random; transpose; shuffle; tornado; neighbor; random permutation; bit-complement; bit-reverse
真实应用程序	PARSEC[15]
路由算法	D2RA；DBRA；Duato；DyXY; Footprint; DeTAR；DyXY_D2RA；Footprint_D2RA；DeTAR_D2RA

表 2 列出了实验模拟中关键参数的设置值。每个物理通道的 VC 数目缺省为 3，我们在评估缓冲资源对性能的影响时，将会改变 VC 数目。我们采用合成模式测试各个算法性能。这些合成模式包括：Uniform Random, Transpose, Shuffle, Tornado, Neighbor, Random Permutation, Bit complement 以及 Bit reverse。这些模式用于 NOC 压力测试的标准测试集。对于决策树生成的路由算法 DetAR_D2RA，我们采用真实应用程序 PARSEC 进行测试。为了评估算法的可扩展性我们选用了 4×4, 8×8, 16×16 2D mesh 以及 4×4×4 3D mesh 作为测试网络。网络注入率单位为 flits/cycle/node。比较的算法包括：

1） Duato 算法：经典的无死锁适应性路由算法。我们直接选用 BookSim 模拟器自带的 min_adapt 算法实现；
2） DBRA 算法：基于缓冲区预约的适应性路由算法，对于剩余维度数为 *n* 的报文需要下一级缓冲区的空闲报文数为 *n*;
3） DyXY 算法[16]：一个基于 Duato 策略加入拥塞反馈的适应性路由算法；
4） Footprint 算法：一个基于 Duato 策略用步径目标避免拥塞的适应性路由算法；
5） DeTAR 算法：一个基于 Duato 策略，采用决策树学习模型生成的适应性路由算法；
6） DyXY_D2RA 、Footprint_D2RA 和 DeTAR_D2RA 算法：基于 D2RA 策略的 DyXY 算法、Footprint 算法和 DeTAR 算法变种。

6.3 D2RA算法和DBRA算法性能评测

D2RA 算法混合 DBRA 路由策略和 DOR 路由策略实现的。D2RA 算法相对 DBRA 算法是否改善了性能，这一节我们通过实验进行验证。图 11 描述了 4×4×4 3D mesh 网络中各种通信模式下 D2RA 算法相对 DBRA 算法在不同注入率下的报文平均延迟的降低率。

1） 类似均匀流量通信模式：如图 11 所示：在 Uniform Random 和 Random Permutation 模式中，报文均衡的通过网络，呈现一种全局通信模式。在这些场景中，D2RA 展现了很明显的性能优势。具体来说，在 Uniform Random 模式下，当注入率=0.58 时， D2RA 算法的平均报文延迟相对 DBRA 算法减少了 62%。类似地，在 Random Permutation 模式下，当注入率=0.21 时，D2RA 算法的平均报文延迟相对 DBRA 算法减少了 33%。这表明 D2RA 算法在加入 DOR 策略后通过动态选择路径能更有效处理全局网络拥塞。

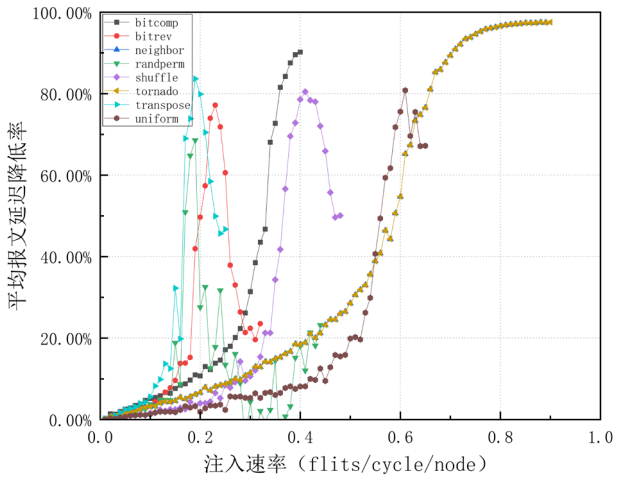

图 11 D2RA 算法较 DBRA 算法报文延迟降低

2）趋于局部流量模式：在 Neighbor 模式中，报文交换主要集中在邻近节点间。在这种模式下，D2RA 算法展现了明显的性能优势。当注入率=0.73 时，D2RA 算法的平均报文延迟相对 DBRA 算法减少了 93%。测试结果表明 D2RA 算法采用的混合通信模式可以有效减少局部通信冲突，快速释放网络资源，提高网络吞吐率。

3）非均匀流量模式：Transpose, Shuffle, Bit complement, Bit-reverse 和 Tornado 这些合成流量模式具有产生热点在特定网络区域密集流量等特点。D2RA 算法的性能始终表现较好，并且其负载可扩展性好。具体来说：在 bitcomp 模式下，注入率为 0.39 时，D2RA 算法的报文平均延迟比 DBRA 算法降低了 89%。在 Bit-rev 模式下，注入率为 0.23 时，D2RA 算法的报文平均延迟比 DBRA 算法降低了 77%。在 shuffle 模式下，在注入率为 0.49 时，D2RA 算法的报文平均延迟比 DBRA 算法降低了 40%。在 transpose 模式下，注入率为 0.21 时，D2RA 算法的报文平均延迟比 DBRA 算法降低了 70%。在 Tornado 模式下，注入率为 0.95 时，D2RA 算法的报文平均延迟比 DBRA 算法降低了 96%。（其核心原因在于：Tornado 模式作为非均匀流量，存在局部热点密集、多维度路由冲突集中的特性，4-ary 3-mesh 网络中多数报文剩余维度数为 2~3，而注入率 0.95 使网络缓冲利用率超 90%，多数节点缓冲空闲空间仅 0~1，远低于 DBRA “下一跳缓冲≥剩余维度数”的要求，导致报文频繁阻塞形成“阻塞链”，延迟剧增；而 D2RA 在缓冲不足时自动切换至无需额外缓冲的 DOR 模式，快速疏导 DBRA 阻塞的报文，同时优先尝试 DBRA 的路径多样性以避免 DOR 路径僵化引发的二次拥塞，动态平衡路径适应性与资源效率，最终相对 DBRA 大幅减少了报文平均延迟。）

6.4 D2RA算法和Duato算法及变种算法性能评测

D2RA 算法和 Duato 算法都是混合路由策略，采用了相同的最短路径完全适应性路由和 DOR 路由。本节我们比较 D2RA 算法和 Duato 算法及它们的变种算法的性能。

图 12 描述了 D2RA 算法，Duato 算法，DyXY 算法和 Footprint 算法以及 D2RA 算法的变种算法 DyXY_D2RA 和 Footprint_D2RA 算法在 4×4×4 3D mesh 中的性能比较。为了直观显示算法的性能趋势，我们采用对报文平均延迟进行取对数处理。测试表明在各种测试场景中无论是均匀模式（如：Uniform 等）还是非均匀模式如：Bit-complement 等），D2RA 算法的性能都优于 Duato 算法，D2RA 算法的的平均报文延迟增长更平缓。其中在 Bit-complement 模式下最为明显。这显示 D2RA 算法相对 Duato 算法在缓冲资源动态分配以及减少网络拥塞表现更好。另一方面，DyXY_D2RA 和 Footprint_D2RA 算法相对 DyXY 算法和 Footprint 算法在各种测试场景下性能有明显的提升。例如：在通信密集的高冲突模式如：Transpose，Shuffle 以及 Bit-reverse 模式测试中，DyXY_D2RA 和 Footprint_D2RA 算法相对 DyXY 算法和 Footprint 算法能获得更高的饱和注入率。在全局均匀流量模式如 Uniform 和 Random permutation 模式下，DyXY_D2RA 和 Footprint_D2RA 算法相对 DyXY 算法和 Footprint 算法在注入率不断增加的情况下，报文平均延迟增加相对平缓。在 Neighbor 这种具有局部通信特点的模式下，由于采用 D2RA 策略，DyXY_D2RA 和 Footprint_D2RA 算法获得了更高的吞吐率和更低的网络延迟。这些测试表明 DyXY 算法和 Footprint 算法在融入 D2RA 路由策略后能优化路由决策，更高效的管理缓冲资源，减少冲突引发的网络延迟。

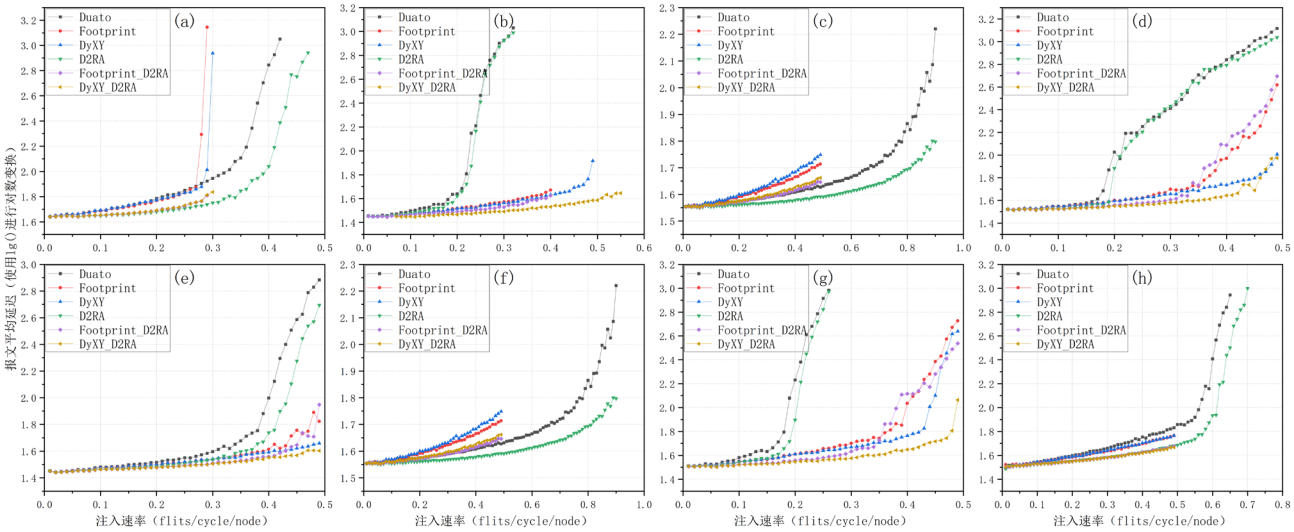

图 12　4-ary 3-mesh 网络（4×4×4 3D mesh 网络）算法结果对比图（(a) bitcomp 模式；(b) bitrev 模式；(c) neighbor 模式；(d) randperm 模式；(e) shuffle 模式；(f) tornado 模式；(g) transpose 模式；(h) uniform 模式）

6.5　不同网络配置参数对算法性能的影响

在这部分测试中，我们比较不同网络 VC 数目以及不同网络规模下 D2RA 算法和 Duato 算法性能。

6.5.1　VC 数目的影响

图 13 描述了在 8×8mesh 网络中，不同 VC 数目下 D2RA 算法相对 Duato 算法在各种流量模式下的吞吐率提升。图 13 显示随着 VC 数目的增加，这个提升值在下降。采用 2VC 时，D2RA 算法的吞吐率实现了平均 27%的提升。其中，在 Tornado 模式下实现 67%的提升。当 VC 数目增加到 4 和 8 时，缓冲资源的增加大大缓减了 Duato 算法的拥塞状况，Duato 算法性能明显提高，相对来说，D2RA 算法的性能提高不大，D2RA 算法的吞吐率的平均提升率分别下降到了 4.7%和 3.6%。

这种趋势正是因为 Duato 算法依赖“自适应性路由+专用逃逸 VC”实现无死锁，VC 数目较少（如 2VC）时，有限逃逸 VC 易成瓶颈，报文争抢冲突率高、吞吐率受限，而 VC 数目增加（如 4VC、8VC）后，逃逸 VC 与自适应 VC 竞争缓解、空闲率提升，性能瓶颈逐步解除；D2RA 算法优势源于“DBRA+DOR”动态适配策略，VC 较少（缓冲受限）时，DOR 模式可规避 DBRA 高缓冲要求、减少报文堆积，优势显著，VC 增加（缓冲充足）时，D2RA 切换至 DBRA 模式频率升高，路由逻辑与 Duato 趋同，动态调度优势弱化。同时低 VC 场景下网络资源紧张使 D2RA 资源调度优势放大，高 VC 场景下 Duato 依托多 VC 提升吞吐率、与 D2RA 效率差距缩小，最终导致 D2RA 相对 Duato 的吞吐率提升值下降，该现象也印证 D2RA 核心价值聚焦于 VC 受限、缓冲敏感的 NoC 场景。

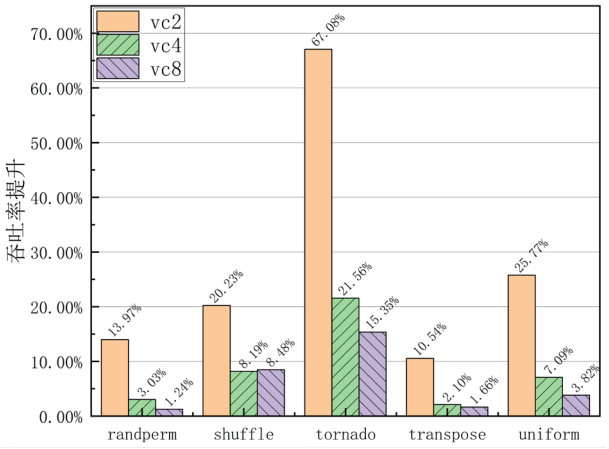

图 13　D2RA 在不同 VC 下的吞吐率提升值

6.5.2　网络规模的影响

图 14 描述了网络规模分别为 4×4，8×8，16×16 时，D2RA 算法相对 Duato 算法在各种流量模式下的吞吐率提升。图 14 表明随着网络规模的扩大，D2RA 算法的性能提升持续增加。具体来说，4×4 规模下，吞吐率平均提升了 1.4%；8×8 规模下，吞吐率平均提升了 8%；16×16 规模下，吞吐率平均提升了 21%。随着网络规模的扩大，资源冲突更加频繁。Duato 算法采用静态逃逸方法的路由策略将降低资源利用率。而 D2RA 算法采用动态配置缓冲资源模式能够更好地平衡资源使用，从而提升性能。

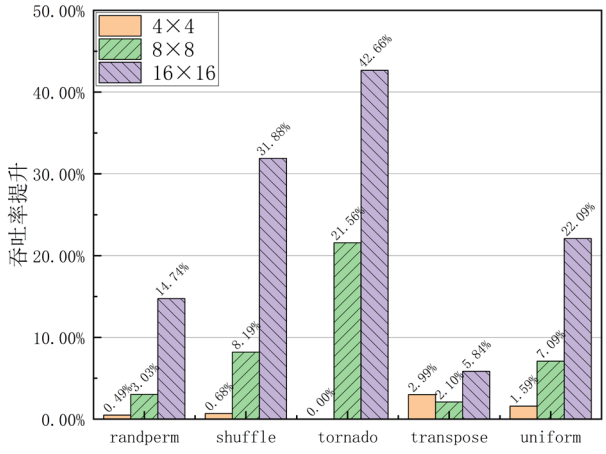

图 14 不同网络规模下的吞吐率

6.6 DeTAR_D2RA算法和DeTAR算法性能评测

图15描述了在8×8mesh网络中，对于PARSEC应用负载，DeTAR_D2RA 相较 DeTAR 在不同基准上的性能变化。在报文延迟上平均提升仅 0.41%，且收益主要集中在具有更强突发性或热点特征的基准（如 canneal 2.38%、blackscholes-small 1.71%），多数应用改进不足 0.3%，甚至在 bodytrack 上出现轻微退化。这一结果说明，在拥塞较轻、VC/缓冲资源充足或通信模式较均匀的场景下，DeTAR 在融入 D2RA 路由策略后收益不明显；但在高竞争、高突发的工作负载中，D2RA 路由策略能够有效规避局部拥塞、减少阻塞链条，从而带来显著延迟下降。进一步分析发现，D2RA 的优势与 VC 数、缓冲深度和包长分布高度相关，在 VC 受限或缓冲较小的配置下，其相对收益预计将进一步放大。此外，从稳定性角度看，D2RA 与 DeTAR 的机器学习决策逻辑并不冲突，结合后在所有基准中均未引入死锁风险，且尾延迟分布趋于收敛，体现了良好的可用性与集成潜力。

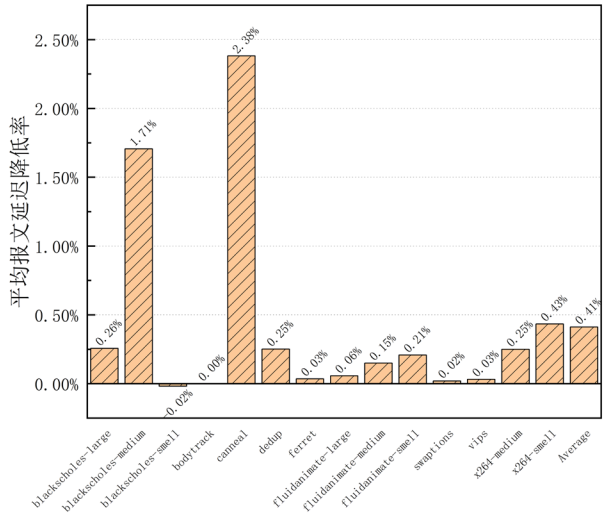

图 15 DeTAR_D2RA 相对 DeTAR 的平均报文延迟降低率

7 结 论

在本文中，我们提出了一种新的路由算法，名为 D2RA，用于 *k*-ary *n*-mesh 网络，该算法结合了维度气泡路由和维序路由。在 D2RA 中，DBRA 路由和 DOR 路由共享缓冲区资源。此外，即使在下一个队列中只有一个空闲的报文空间，具有 *n* 个剩余维度的报文仍然可以申请进入下一个队列，并按照 DOR 进行路由。我们通过形式化证明的方式，证明了本文的主要理论结果：D2RA 在任何 *k*-ary *n*-mesh 网络中都能保证无死锁。并且 D2RA 算法可以和目前流行的路由设计方法无缝融合。实验结果也进一步证实了 D2RA 算法具有更优的性能及更强的负载和可扩展性。

D2RA 算法为我们面向通用网络设计高效路由提供了一个新思路。我们可以通过融合网络已有的多种路由策略设计新的混合路由算法来改善通信性能。路由算法的融合需要考虑算法之间的性能互补性以及实现的难易度，最好可以像 D2RA 算法实现无缝对接。例如：我们以前的工作实现了完全适应性路由算法 TADBR[17]，该算法对下一步缓冲区的需求是 2n(n 是剩余维度数)个报文空间，该算法与 DBRA 算法一样：报文剩余维度数越大越难进入下一步缓冲区。文献[18]的路由算法保证在同一个环内只要存在一个气泡(Bubble)，报文的路由就不会死锁。它对缓冲区空间需求较少，但不能实现适应性路由。下一步我们将研究 TADBR 和文献[18]的路由策略的结合实现 Torus 网络的新型路由算法。

参 考 文 献

[1] Abbasi R, Vahid J. SBCT-NoC: ultra low-power and reliable simultaneous bi-directional current-mode transceiver for network-on-chip interconnects. IEEE Transactions on Nanotechnology, 2023, 22: 777-784.

[2] Agarwal S, Goel K, Sinha M, Deb S. Mitigation of phase transitions in self-organizing NoC for stable queueing dynamics. IEEE Transactions on Computers, 2025, 74: 623-636.

[3] Ma Rui-Yang, Huang Jia-Yi, Zhang Shi-Jian, Xie Yuan, Luo Guo-Jie. NoCFuzzer: automating NoC verification in UVM. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 2025, 44(1): 371-384.

[4] Zhang Xiao-Yun, Dong De-Zun, Li Cun-Lu, Wang Shao-Cong, Xiao Li-Quan. A survey of machine learning for network-on-chips. Journal of Parallel and Distributed Computing, 2024, 186: 104778.

[5] Jin Kang, Li Cun-Lu, Dong De-Zun, Fu Bin-Zhang. HARE: history-aware adaptive routing algorithm for endpoint congestion in networks on chip. International Journal of Parallel Programming, 2019, 47:433-450.

[6] Zhang Xiao-Yun, Wang Yao-Hua, Dong De-Zun, Li Cun-Lu, Wang Shao-Cong, Li Quan-Xiao. DeTAR: A decision tree-based adaptive routing in networks-on-chip//Proceedings of the 29th International European Conference on Parallel and Distributed Computing, Limassol, Cyprus, 2023: 352-366.

[7] Duato J. A necessary and sufficient condition for deadlock-free routing in cut-through and store-and-forward networks. IEEE Transactions on Parallel and Distributed Systems, 1996, 7(8): 841-854.

[8] Ebrahimi M, Daneshtalab M. Ebda: a new theory on design and verification of deadlock-free interconnection networks// Proceedings of the 2017 ACM/IEEE 44th Annual International Symposium on Computer Architecture, Toronto, Canada, 2017: 703-715.

[9] Xiao Can-Wen, Zhang Min-Xuan, Dou Yong, Zhao Zhi-Tong. Dimensional bubble flow control and fully adaptive routing in the 2-d mesh network on chip//Proceedings of the 2008 IEEE/IFIP International Conference on Embedded and Ubiquitous Computing, Shanghai, China, 2008:353-358.

[10] Xiao Can-Wen, Yang Yue, Zhu Jian-Wen. A sufficient condition for deadlock-free adaptive routing in mesh networks. IEEE Computer Architecture Letters, 2015, 14(2):111-114.

[11] Ma Sheng, Jerger N E, and Wang Zhi-Ying. Dbar: an efficient routing algorithm to support multiple concurrent applications in networks-on-chip//Proceedings of the 2011 38th Annual International Symposium on Computer Architecture, San Jose, USA, 2011:413-424.

[12] Wang Chang-Hong, Dong De-Zun, Wang Zi-Cong, Zhang Xiao-Yun, Zhao Zhen-Yu. Relar: a reinforcement learning framework for adaptive routing in network-on-chips//Proceedings of the 2021 IEEE International Conference on Cluster Computing, Portland, USA, 2021:813-814.

[13] Reza M F, Le T T. Reinforcement learning enabled routing for high-performance networks-on-chip//Proceedings of the 2021 IEEE International Symposium on Circuits and Systems, Daegu, Republic of Korea, 2021: 1-5.

[14] Jiang N, Becker D, Michelogiannakis G, Balfour J, Towles B Shaw D.E.. A detailed and flexible cycle-accurate network-on-chip simulator//Proceedings of the 2013 IEEE International Symposium on Performance Analysis of Systems and Software, Austin, USA, 2013:86-96.

[15] Bienia C, Kumar S, Singh J P, Li K. The parsec benchmark suite: characterization and architectural implications. //Proceedings of the 2008 International Conference on Parallel Architectures and Compilation Techniques, Toronto, Canada, 2008: 72-81.

[16] Li Ming, Zeng Qing-An, Jone W B. Dyxy - a proximity congestion-aware deadlock-free dynamic routing method for network on chip//Proceedings of the 2006 43rd ACM/IEEE Design Automation Conference, San Francisco, USA, 2006:849-852.

[17] Xiao Can-Wen, Zhang Min-Xuan, Guo Feng. Dimensional bubble flow control and adaptive routing algorithm in torus networks. Journal of Computer Research and Development, 2007, 44(9): 1510-1517 (in Chinese)
(肖灿文, 张民选, 过锋. 环网中的维度气泡流控与自适应路由算法. 计算机研究与发展, 2007, 44(9):1510-1517)

[18] Dai Yi, Lu Kai, Ma Sheng, Su Jin-Shu, Li Dong-Sheng. Bubble-swap flow control. ACM Transactions on Architecture and Code Optimization, 2025, 22(1): 1-26.

XIAO Can-Wen, Ph.D., professor. His main research interests include computer architecture and high-performance interconnection networks.

WANG Wei, Ph.D. candidate. His main research interests include communication optimization and high performance computing.

ZHANG Xiao-Yun, Ph.D., assistant researcher. Her main research interests include high performance computing and network-on-chip.

LI Cun-Lu, Ph.D., associate professor. His main research interests include computer architecture and computer network.

YANG Bo, Ph.D., associate professor. His main research interests include communication optimization and high performance computing.

LIU Jie, Ph.D., professor. His main research interests include communication optimisation, high performance computing and numerical computing.

CHE Yong-Gang, Ph.D., professor. His main research interests include parallel algorithms, performance reviews and high performance computing.