# Torus 网络中基于中心距离的完全自适应路由算法

虞志刚[1],向 东[2],王新玉[1]
(1.清华大学计算机科学与技术系,北京 100084; 2.清华大学软件学院,北京 100084)

**摘 要:** Torus 网络凭借其优越的结构特性,引起了工业界和学术界的广泛关注.高效、无死锁的路由算法设计是互连网络研究的一个重要方面.针对 Torus 网络实现自适应路由所需虚通道数目多的缺点,提出了自适应路由算法 Gear,该算法基于中心距离的方法来限制虚通道的使用,在虚切通交换下仅需两条虚通道即可为 Torus 网络提供无死锁自适应路由.通过仿真对所提算法的有效性进行了验证,结果表明,在同等情况下算法 Gear 的性能较经典的维序路由和 Duato 协议具有非常明显的优势.

**关键词:** Torus 网络; 路由算法; 虚通道; 自适应路由
**中图分类号:** TN915.5 **文献标识码:** A **文章编号:** 0372-2112 (2013) 11-2113-07
**电子学报 URL:** http://www.ejournal.org.cn **DOI:** 10.3969/j.issn.0372-2112.2013.11.001

# Fully Adaptive Routing in Torus Networks Based on Center Distance

YU Zhi-gang[1], XIANG Dong[2], WANG Xin-yu[1]
(1. *Department of Computer Science and Technology, Tsinghua University, Beijing* 100084, *China*;
2. *School of Software, Tsinghua University, Beijing* 100084, *China*)

**Abstract:** Torus networks win lots of industrial and academic attention by virtue of the superior architecture proprieties. The design of efficient deadlock-free routing algorithms is an important aspect of interconnection networks research. Against the problem that torus networks need numbers of virtual channels to support adaptive routing, we propose an adaptive routing algorithm: Gear, which needs only 2 virtual channels to support deadlock-free adaptive routing in Virtual Cut-Through switched Torus. Gear implements fully adaptive routing by constraining the use of some special virtual channels on the concept of Center Distance. We verify the efficiency of the algorithm with simulation. The results show that, in the same circumstances, the advantage of proposed Gear over classic Dimension-Order Routing and Duato's Protocol is very apparent.

**Key words:** Torus networks; routing algorithm; virtual channel; adaptive routing

## 1 引言

互连网络(Interconnection Networks)取代由于电气限制而达到性能极限的总线技术,成为解决现代数字系统系统级通信问题的通用方案[1].互连网络主要由三个要素来描述:拓扑结构、交换机制和路由算法[2].Torus 网络是一种完全对称的拓扑结构,具有很多优良特性,如网络直径小,所有节点度相同,结构简单,路径多以及可扩展性好等.因此被广泛应用于商用系统中,如 2004 年超级计算机 TOP 500 中排名第一的 IBM Blue Gene/L 就采用了 3-D Torus 网络[3];最新的超级计算机 IBM Blue Gene/Q 和 2011 年 TOP500 中排名第一的 K Computer 就分别采用了 5-D Torus 和 6-D Torus 作为系统互连结构[4].因此在 Torus 网络中设计高效、无死锁路由算法至关重要.

交换机制决定怎样给分组分配通道、缓冲区等网络资源.分组交换要求在做路由决策之前,分组必须被完整接收,而虚切通交换(Virtual Cut-Through Switching, VCT)[2,27]不同,引入了流水线式消息传输,在路由器完全接收分组之前,分组直接跨到下一个路由器上.VCT 交换被广泛用于超级计算机中,如商用超级计算机 Cray XT/Cascade 就采用了 VCT 交换技术[4,28].如未特别说明,下文所设计路由算法都针对 VCT 交换网络.

路由算法决定了每个消息或分组将在网络中传输的路径,它负责将分组正确无误的发送到目的节点.确定性路由算法,分组在任意节点对之间总是提供相同的路径,而与网络状态无关.该算法简单,但路径唯一,因此路径中有一条通道或节点发生故障时,分组就不能

收稿日期:2013-01-10;修回日期:2013-02-28;责任编辑:蓝红杰
基金项目:国家自然科学基金(No.61170063,No.60910003)

被正确传输.适应性路由算法[1,2,5]在做路由决策时要考虑当前网络的状态.通常分组在任意节点对之间都有多条路径供选择,所有分组均匀的使用各条通道,使得网络流量更加均衡,有利于网络性能的提高.

## 2 相关工作

转弯模型理论[7,8]和平面自适应算法[9]通过限制路由实现无死锁路由,在一定程度上损失了自适应度.付斌章[10]提出了一种用于 Mesh 网络的路由算法 AbTM,该算法不需要虚通道就可实现可重构路由.与 Mesh 网络相比,Torus 网络中引入的环绕通道虽减少了网络直径和平均距离,但环绕通道在每一维引入了环,使得针对 Torus 网络设计无死锁自适应路由异常困难[4,6,11].尤其在虚通道资源有限的情况下.

Dally[1,6]提出了适用于 Torus 网络的维序路由算法(Dimension Order Routing, DOR):需 2 条虚通道,分组严格按照特定次序跨越所有的维,当分组跨过环绕通道后,使用另外一条虚通道.但该算法为确定性路由,使得系统性能受限.顾华玺[12]将转弯模型扩展到 Torus 网络,仅需 3 条虚通道实现部分自适应路由.Gravano[1]提出了两种完全自适应路由算法分别需要 5 和 8 条虚通道.Duato[13]提出了一种完全自适应路由算法的设计策略,在无死锁确定性路由算法的基础上,增加一条虚通道提供完全自适应性即可.如果以 DOR 作为 Torus 网络基本的路由算法,再增加一条虚通道实现完全自适应路由,共需要 3 条虚通道.为了后续表述方便,称这一算法为 Duato 协议.

## 3 预备知识

**定义 1(Torus)** $k$ 元 $n$ 立方有 $k_1 \times k_2 \times \cdots \times k_{n-1} \times k_n$ ($k_i = k$, $\forall i \in [1, n]$)个节点,$k$ 是第 $i$ 维的节点数,$k \geqslant 2$.每个节点 $X$ 的坐标由 $n$ 位坐标($x_1, \cdots, x_{n-1}, x_n$)定义,其中 $0 \leqslant x_i \leqslant k-1$, $\forall i \in [1, n]$.$X$ 和 $Y$ 相邻的充要条件是:存在 $j$ 使得 $y_j = (x_j \pm 1) \bmod k$,而对任意的 $i \neq j$ 且 $1 \leqslant i, j \leqslant n$,有 $y_i = x_i$.$k$ 元 $n$ 立方由于增加了环绕通道,使得网络更加规整,如图 1 所示为一个 6 元 2 立方(即 6×6 Torus)网络,网络完全对称.

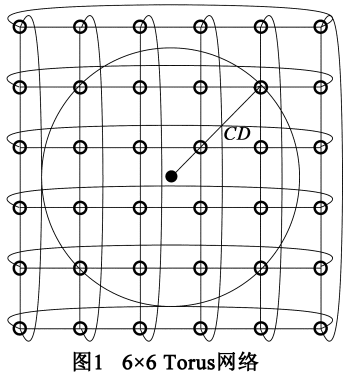


图1 6×6 Torus网络

**定义 2(虚通道)** 虚通道[14]是互连网络研究中常用概念,每条虚通道具有独立的缓冲区,而多条虚通道分时复用一条物理通道.从逻辑上看,每条虚通道就像使用一条低速工作的物理通道一样.虚通道可以用来降低消息延迟,提高网络吞吐量.既然允许多条分组以时分复用的方式共享一条物理通道,分组就可以继续发送而不必阻塞.但虚通道的引入是有代价的,它增加了路由器的硬件开销[22],使得路由调度更加复杂.

**定义 3(死锁)** 死锁是指当多个分组等待路由转发时,各自占用部分资源又相互等待对方释放资源时,出现的所有的分组都永久阻塞的一种网络状态[6,15,16].

图 2 示出了一个死锁的例子,在图($a$)所示的网络中,同时存在四个分组,分组 $A$ 的源节点为 $n_1$,目的节点为 $n_3$,当前分组占用通道 $c_1$ 申请通道 $c_2$;分组 $B$ 源节点为 $n_2$,目的节点 $n_4$,当前占用 $c_2$ 申请 $c_0$;依次类推,分组 $C$ 占用 $c_3$ 申请 $c_4$;分组 $D$ 占用 $c_4$ 申请 $c_1$.这样分组 $A, B, C, D$ 即形成了图 2($b$)所示的通道间的循环依赖关系,所有分组都将不能前进,构成了死锁.

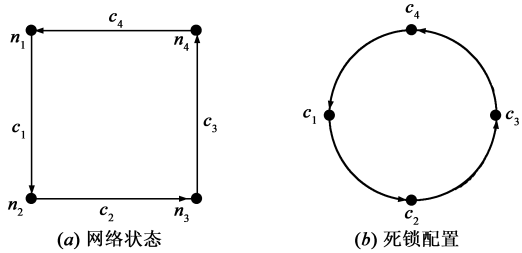


图2 死锁实例

**定义 4(Torus 网络维序路由算法)** 在路由分组时,严格按照递增(或递减)次序跨越所有维,只有在上一维的偏移量降为零后才再转入下一维进行路由.Torus 网络的维序路由[17,18]需要使用 2 条虚通道来避免死锁.分组在跨越某一维的环绕通道之前使用虚通道 1;跨越之后就使用虚通道 2.

**定义 5(中心距离)** 中心距离,如图 1 所示,是指网络中任意节点与坐标中心的距离,用 $CD$ 表示.$k$ 元 $n$ 立方网络中,任意节点 $x = (x_1, x_2, \cdots, x_n)$, $0 \leqslant x_i \leqslant k-1$, $\forall i \in [0, k-1]$与网络坐标中心$((k-1)/2, (k-1)/2, \cdots, (k-1)/2)$的距离记为:

$$CD(x) = \sqrt{\sum_{i=1}^{n} \left(\frac{k-1}{2} - x_i\right)^2}$$

$CD(x)$为节点 $x = (x_1, x_2, \cdots, x_n)$到坐标中心之间直线距离.$k$ 元 $n$ 立方网络中,网络的最大距离为 $(k-1)\sqrt{n}/2$;$k$ 为奇数时,网络的最小 $CD = 0$;$k$ 为偶数时,最小 $CD = \sqrt{n}/2$.

## 4 基于中心距离的完全自适应路由算法

### 4.1 设计思想

本节提出了一种基于虚切通交换 Torus 网络的路

由算法:Gear*,该算法只需 2 条虚通道即可提供完全自适应、最短路径路由.下面将详细叙述算法的设计思想,具体算法以及无死锁证明.

Gear 算法基于网络中心距离的概念,对通道的使用进行限制.算法中要用到两条虚通道,为了表述方便,用 $VC_1$, $VC_2$ 分别表示虚通道 1 和虚通道 2.

**规则 1**:如果分组从源到目的节点不需要过环绕通道,$VC_1$ 提供完全自适应路由.

**规则 2**:如果分组从源到目的节点不需要过环绕通道,$VC_2$ 提供维序路由.

**规则 3**:如果分组从源到目的节点需要过环绕通道(1 条或者更多),$VC_1$ 提供从 $A$ 到 $B$ 的路径,当且仅当满足 $CD(A) \leqslant CD(B)$.

**规则 4**:如果分组从源到目的节点需要过环绕通道,分两种情况:(1) $VC_2$ 提供从 $A$ 到 $B$ 的路径,若 $CD(A) > CD(B)$;(2)用 $l$ 表示分组需要过环绕通道的最低维,若分组在维 $l$ 的边界节点,则提供该维环绕通道.

规则 1、2 表示,对于从源节点到目的节点不需要过环绕通道的分组,$VC_2$ 提供连接的、无死锁的确定性路由—维序路由,$VC_1$ 提供所有可以使分组向目的节点靠近的路径—完全自适应路由.$VC_2$ 保证分组最终能够传递到目的节点,$VC_1$ 为路由提供自适应性,提高网络性能.规则 3 和 4 为需要过环绕通道的分组提供了路径选择.如果分组需要过环绕通道,$VC_1$ 使分组向中心距离不小于当前节点的节点路由;$VC_2$ 不仅提供中心距离减少的路径,当分组在需要过环绕通道的最低维的边界节点时,还提供通过环绕通道的路径.

## 4.2　算法描述

依据 4.1 节规则,算法 1 为适用于 *n-D* Torus 网络的 Gear 算法.首先,计算当前节点和目的节点各维上的偏移,若所有维的偏移均为零,那么分组到达目的节点;否则,调用 *checkWraplink*()计算分组从当前节点到目的节点需要通过环绕通道的次数:若 $c = 0$,分组不需要过环绕通道,调用 *noWraplink*()为分组选择输出路径;若 $c \neq 0$,调用函数 *Wraplink*()为分组选择输出路径.

如算法 2 所示,*noWraplink*()示出了不需要过环绕通道的分组的路由过程,$VC_1$ 通道为分组提供完全自适应路由,也就是说,对于任意维 $i$,若 $0 < off_i \leqslant k/2$ 或 $off_i < -k/2$,提供 $VC_{i,1}+$($VC_{i,1}+$ 表示第 $i$ 维正方向上 $VC_1$ 通道);若 $-k/2 \leqslant off_i < 0$ 或 $off_i > k/2$,提供 $VC_{i,1}-$.而 $VC_2$ 通道为分组提供维序路由,用 $j$ 表示偏移不为零的最低维,若 $off_j > 0$($off_j < 0$),则提供 $VC_{j,2}+$($VC_{j,2}-$).选择函数 *select*()在所有可选路径中选择一条作为分组的下一跳.

**算法 1**　适用于 *n-D* Torus 网络的 Gear 算法 Input: Coordinates of the current node $(c_1, c_2, \cdots, c_{n-1}, c_n)$ and the destination node $(d_1, d_2, \cdots, d_{n-1}, d_n)$

Output: Selected output channel

```
{
    (1) S: = Ø, off_i: = d_i - c_i, ∀ i∈[1, n];
    (2) if off_i = 0, ∀ i∈[1, n]. Return channel: = internal;
    (3) c: = checkWraplink(off_1, off_2, ⋯, off_{n-1}, off_n);
    (4) if c = 0, S: = Fully-adaptive-routing-noWraplink();
    (5) else S: = Fully-adaptive-routing-Wraplink();
    (6) Return channel: = select(S).
}
```

**算法 2**　*n-D* Torus 网络完全自适应路由函数 noWraplink() Input: Coordinates of the current node $(c_1, c_2, \cdots, c_{n-1}, c_n)$ and the destination node $(d_1, d_2, \cdots, d_{n-1}, d_n)$

Output: Set of available output channels

```
{
/* If the packet needn't traverse any wraparound links, the virtual channel
VC2 provides Dimension Order Routing, while the virtual channel VC1 provides fully adaptive routing */
    1) S: = Ø, Let j denotes the first dimension whose offset is not zero;
    2) for(i: = 1, i < = n, i ++)
        {
          a) off_i: = d_i - c_i,
          b) if off_i > 0, S: = SU{VC_{i,1}+};
          c) if off_i < 0, S: = SU{VC_{i,1}-};
          d) if i = j,
                d1) if off_j > 0, S: = SU{VC_{j,2}+};
                d2) if off_j < 0, S: = SU{VC_{j,2}-};
        }
    3) Return S.
}
```

算法 3 中 *Wraplink*()示出了需要过环绕通道分组的路由过程,此时 $VC_1$ 和 $VC_2$ 为分组提供完全自适应路由,但这些路径必须要满足算法对 $VC_1$ 和 $VC_2$ 通道的限制,函数 $VC_1$_ *Restrain*()和 $VC_2$_ *Restrain*()将对提供的路径进行判断,若满足要求则返回该路径;否则返回 Ø.步骤(*b*)到(*c*)即实现了路由过程.根据规则 4,设分组需要过环绕通道的最低维为 $l$,那么若分组当前在第 $i = l$ 维的边界节点,那么 $VC_2$ 通道为分组提供过环绕通道的路径.具体过程如步骤(*d*)所示,如果 $c_i = 0$,那么为分组提供环绕通道 $VC_{i,2}-$;若 $c_i = k-1$,那么为分组提供环绕通道 $VC_{i,2}+$.

* Gear:齿轮,由于算法两条虚通道的提供的路径互不重叠且相互补充,如齿轮般契合,故而命名.

**算法 3** *n-D* Torus 网络完全自适应路由函数 Wraplink() Input: Coordinates of the current node $(c_1, c_2, \cdots, c_{n-1}, c_n)$ and the destination node $(d_1, d_2, \cdots, d_{n-1}, d_n)$

Output: Set of available output channels

{

/* If the packet need traverse one or more wraparound links */

1) S: = Ø, Let l denotes the first dimension along which the packet needs to traverse a wraparound link

2) for($i$: = 1, $i$ < = $n$, $i$ ++)

{

a) $off_i$: = $d_i - c_i$;

b) if $0 < off_i \leqslant k/2$ or $off_i < -k/2$,

S: = SU$VC_1$_*Restrain*($VC_{i,1}$ +),

S: = SU$VC_2$_*Restrain*($VC_{i,2}$ +);

c) if $-k/2 \leqslant off_i < 0$ or $off_i > k/2$,

S: = SU$VC_1$_*Restrain*($VC_{i,1}$ −),

S: = SU$VC_2$_*Restrain*($VC_{i,2}$ −);

d) if $i = l$,

d1) if $c_i = 0$, S: = SU{$VC_{i,2}$ −};

d2) if $c_i = k - 1$, S: = SU{$VC_{i,2}$ +};

}

3) Return S.

}

/* Note: $VC_1$_*Restrain*(*channel*) means that: if *CD*(*curr*) ⩽ *CD*(*next*), Return {*channel*}; else Return Ø. $VC_2$_*Restrain*(*next*) means that: if *CD*(*curr*) > *CD*(*next*), Return {*next*}; else Return Ø. Here *curr* donates the current node where the packet is, next donates the node the channel directs to. */

### 4.3 无死锁证明

无活锁、饿死和死锁是对路由算法最基本的要求，无活锁是指分组不会在目的节点周围游荡；无饿死是指分组申请的资源不会总是被其他的分组占用；无死锁表示分组不会在网络中永久阻塞，也就是说分组最终能够到达目的节点[20,21]. 解决活锁和饿死问题相对比较简单，最短路径路由是解决活锁最常用的方法，而使用正确的资源分配策略会消除饿死[22]. 死锁是迄今为止最难解决的问题，处理死锁有三种策略：死锁预防、死锁避免和死锁恢复[23,24]. Gear 采用正确的资源分配策略，并提供最短路径路由，故不存在活锁和饿死. 因此，本节只给出 Gear 算法无死锁证明.

Dally 首次提出使用通道相关图[6]概念来分析确定性路由算法的无死锁特性，Duato 发展了 Dally 的无死锁理论，并将其扩展到自适应路由算法无死锁特性的研究中去，给出自适应路由算法无死锁的充分必要条件：存在一个连接的路由子函数且其扩展通道相关图中没有环路[13]. Verbeek 对 Duato 关于自适应路由算法无死锁的充分必要条件进行修正，并给出了新的充分必要条件[25,26]及相应的死锁检测算法(www.cs.ru.nl/~freekver/PS_WHS/index.html). 在路由算法设计过程中，本文使用该理论对算法进行了检测，结果表明完全自适应路由算法 Gear 是无死锁的.

## 5 试验结果与性能分析

为了研究本文提出的完全自适应路由算法 Gear 的性能，开发了一个微片级互连网络模拟器. 模拟器实现了以上算法，并实现了 Torus 网络维序路由算法 DOR 及 Duato 协议. 通过比较以上算法在同一模拟器下的性能，来验证 Gear 在高效利用资源实现高性能路由方面的改进与提高.

实验中采用平均延迟(Average Latency)以及标准化可接受流量(Normalized Accepted Traffic)来衡量算法性能. 延迟是指从分组开始发送到被目的节点接收所经历的时间. 单个分组的延迟是不重要的，在多数情况下，平均延迟直接关系到网络性能. 标准化可接受流量是指吞吐率除以网络饱和负载，在一定的负载条件下，较低的延迟和较高的标准化可接受流量意味着更好的性能.

实验所用到的网络结构有 4×4、8×8、16×16，使用三种不同的流量模型来检验算法性能：均匀流量模型、对位传输模型和热点流量模型. 在均匀流量模型下，每个节点等概率发送分组到其他节点；在对位传输模型下，节点$(i,j)$只发送分组到$(j,i)$；在热点流量模型下，本文对单个节点设为热点，其位置随机产生，将比其他节点多获得 10% 的流量.

下面将分别从流量模型、网络规模和通道使用率三个方面对算法 Gear 的性能进行研究.

### 5.1 流量模型

本节在 8×8Torus 网络(本试验中消息长度默认为 16 个微片，虚通道缓冲区大小默认为 16 个微片)中，讨论算法在不同流量模型下的性能：比较了在不同流量模型下，2 条虚通道下 Gear 与 DOR 以及 3 条虚通道下，Gear 和 Duato 协议在各流量模型下的性能比较. Gear 提供完全自适应路由，DOR 为确定性路由，为比较算法的性能，将 Gear 与 Torus 网络完全自适应路由算法—Duato 协议进行对比. Duato 协议需要 3 条虚通道来实现完全自适应路由，其中 2 条虚通道提供逃逸通道，另外 1 条通道提供自适应通道，这是迄今为止在 Torus 中实现自适应路由所需虚通道的最小值. 为了公平起见，Gear 也使用 3 条虚通道.

如图 3 示出了均匀流量模型，使用 2 条虚通道情况下算法 Gear、DOR 和使用 3 条虚通道情况下 Gear 和 Duato 协议的性能曲线. 从图中可以看出：(1)在同等条件下，Gear 分别优于 DOR 和 Duato 协议：当标准化实用负载达到 0.3 时 DOR 趋于饱和，Gear(2*VC*)的饱和点在 0.48；Duato 协议在 0.5 时趋于饱和，而 Gear(3*VC*)的饱和

点在 0.6 左右.综上,Gear(2*VC*)性能较 DOR 提高了 67%;Gear(3*VC*)性能较 Duato 协议提高了 20%;(2)Gear 只需 2 条虚通道可以为分组提供完全自适应路由,Duato 协议需要使用 3 条虚通道才能为分组提供完全自适应路由.但是从图中不难看出,Gear 在 2 条虚通道情况下,性能与 Duato 协议差别不大,仅为 4%;(3)另外在网络流量不断增大的情况下,算法 Gear 为分组提供完全自适应路由,性能一直优于 DOR;Gear(3*VC*)增加 1 条通道用于完全自适应路由,使整体性能大幅提升,特别在高负载情况下性能改善更加明显.

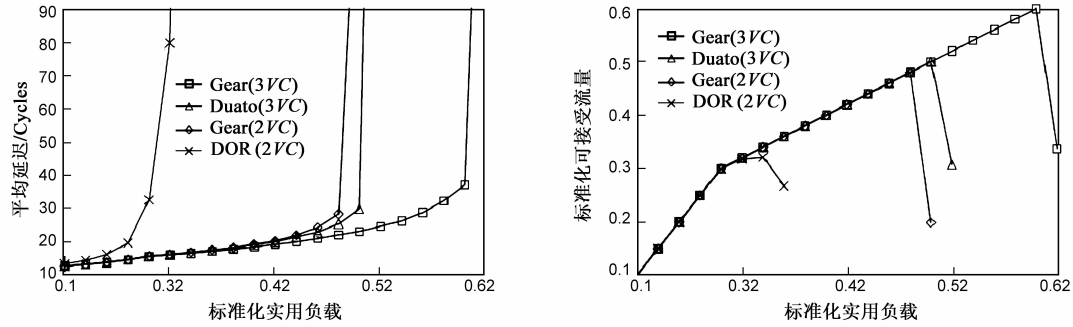



**图3　均匀流量模型下Gear, DOR, Gear(3*VC*)和Duato协议的性能比较**

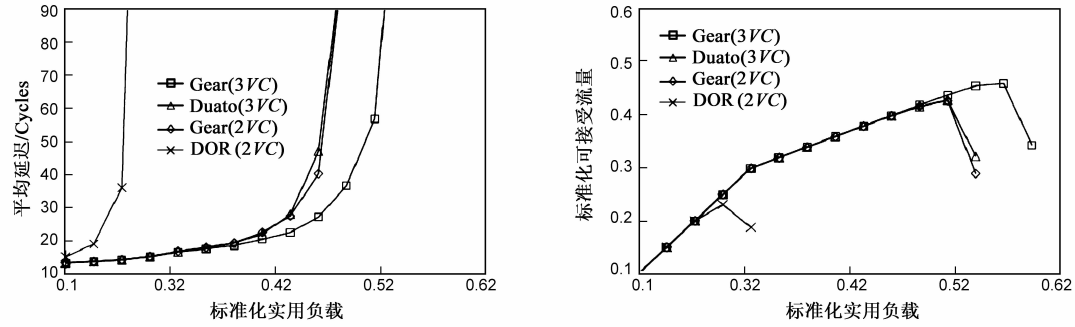



**图4　对位传输模型下Gear, DOR, Gear(3*VC*)和Duato协议的性能比较**

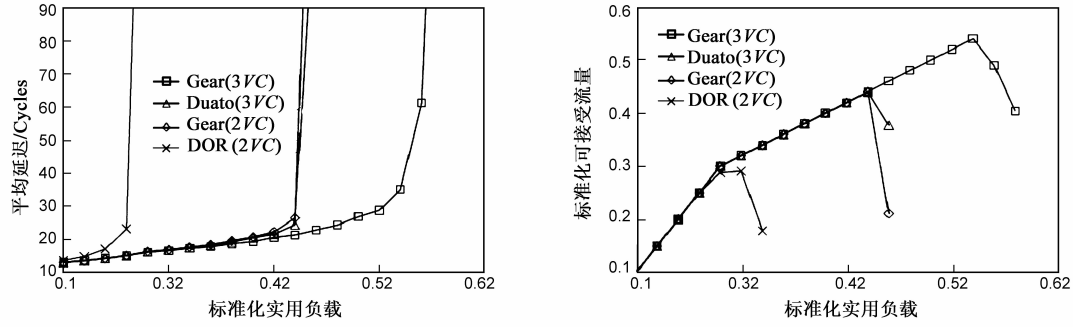



**图5　热点传输模型下Gear, DOR, Gear(3*VC*)和Duato协议的性能比较**

图 4 和图 5 分别示出了在对位传输模型和热点模型下,算法 Gear(2*VC*)与 DOR,Gear(3*VC*)与 Duato 协议的性能曲线的性能比较.从图中可以看出,无论在哪种流量模型下,Gear 为所有分组提供所有的最短路径,而 DOR 只提供一条确定路径,从而获得了性能上的极大改善.在对位传输模型下,Gear 的饱和点为 0.4,DOR 的饱和点为 0.2,提高了 1 倍.在热点模型下,Gear(0.45)较 DOR(0.25)提高了 80%,也就是说,在网络负载较大时,Gear 能够为分组提供自适应度,分组可以选择所有最短路径进行路由,从而能够有效的减少拥塞和消息延迟,提高网络吞吐,改善网络性能.如图 5 所示,Duato 协议在 0.44 趋于饱和,而 Gear 在 0.55 趋于饱和,提高了 25%左右.另外,不管在低负载还是高负载情况下,算法 Gear 性能都优于 Duato 协议,特别是在高负载情况下,优势更加明显.在对位传输模型下,Gear 的性能也优于 Duato 协议,性能改善 12.5%.

## 5.2　网络大小

本节将研究网络规模对路由算法性能的影响.随

着网络规模的增大,算法对网络规模的敏感性成为评价算法性能的一个指标.本节在均匀流量模型下,采用不同网络规模进行性能比较.

图 6 示出了网络规模为 4×4,8×8,16×16 三种情况下算法 Gear 和 DOR 的性能比较.通过观察不难发现,随着网络规模的增大,算法 Gear 和 DOR 的饱和点都减小,但 Gear 的性能一直优于 DOR.在 4×4 网络条件下,Gear(饱和点 0.48)较 DOR(饱和点 0.34)性能提高 41.2%;在 8×8 网络条件下,Gear(饱和点 0.5)较 DOR(饱和点 0.3)性能提高 67%;在 16×16 网络条件下,Gear(饱和点 0.42)较 DOR(饱和点 0.25)性能提高 68%.

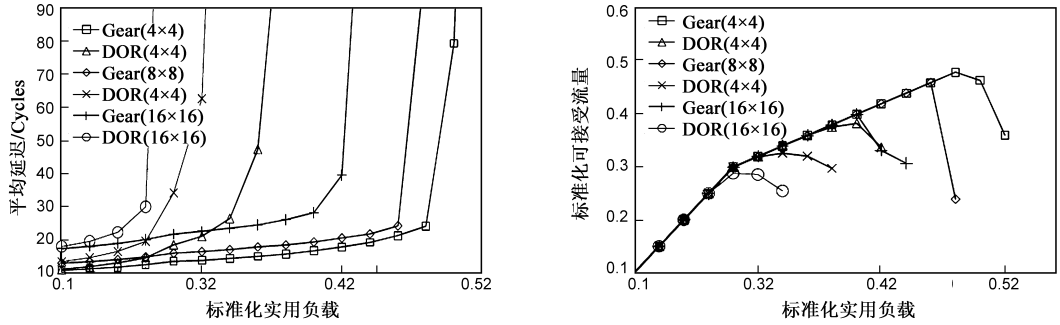



图6 不同网络规模下算法Gear和DOR的性能比较

### 5.3 通道使用率

本节将进一步分析算法 Gear,并从通道使用率的角度来解释,Gear 性能优于 DOR 的原因.本节实验在均匀流量模型下,在相同消息注入率时,分析 Gear 和 DOR 对虚通道 1 和虚通道 2 的使用情况.如表 1 所列(表中"X"表示网络已饱和时,数据未统计),在相同注入率下,Gear 对两条通道的使用更加均衡.如在负载为 0.35 的情况下,Gear 对 $VC_2$ 的使用率 34.05% 较 DOR(7.63%)提高了 3.5 倍.然而,从整体上来说,两条通道的使用率还不够均衡,因此在后续工作中,将根据这一现象,对虚通道的使用进一步优化.

表 1 均匀流量模型下虚通道使用率

| 注入率 | Gear(2*VC*) | | DOR(2*VC*) | |
|---|---|---|---|---|
| | $VC_1$ | $VC_2$ | $VC_1$ | $VC_2$ |
| 0.1 | 88.85% | 11.15% | 93.23% | 6.77% |
| 0.15 | 79.74% | 20.26% | 93.13% | 6.87% |
| 0.2 | 72.19% | 27.81% | 93.14% | 6.86% |
| 0.25 | 68.02% | 31.98% | 93.16% | 6.84% |
| 0.3 | 66.11% | 33.89% | 92.87% | 7.13% |
| 0.35 | 65.95% | 34.05% | 92.37% | 7.63% |
| 0.4 | 61.50% | 38.50% | X | X |

## 6 结论

Torus 网络在当今商用超级计算机中得到了广泛应用,同时也引起了学术界的广泛研究.路由算法是决定网络性能的关键因素之一,设计高效无死锁自适应路由算法是提高网络性能的重要方面.本文基于网络中心距离的概念,通过限制部分虚通道的方法,为 Torus 网络设计了无死锁自适应路由算法 Gear,仅利用 2 条虚通道实现了 Torus 网络的完全自适应路由.这是迄今为止,在 Torus 网络中实现完全自适应路由的所需虚通道数的最小值.仿真试验表明,在同等条件下,与 DOR 和 Duato 协议相比,算法 Gear 获得了不低于 20% 的性能改善,具有更低的网络延迟以及更高的网络吞吐;另外,Gear 能够更好的适应网络规模变化,且均匀使用 2 条虚通道.

### 参考文献

[1] Dally W J, Towles B. Principles and Practices of Interconnection Networks[M]. San Francisco CA: Morgan Kaufmann, 2004.112 – 321.

[2] Duato Jose, Yalamanchili Sudhakar, et al. Interconnection Networks: An Engineering Approach[M]. San Francisco, CA: Morgan Kaufmann, 1997.1 – 204.

[3] Adiga N R, Blumrich M A, et al. Blue gene/l torus interconnection network[J]. IBM Journal of Research and Development, 2005, 49(2): 265 – 276.

[4] Luo Wei, Xiang Dong. An efficient adaptive deadlock – free routing algorithm for torus networks[J]. IEEE Transaction on Parallel and Distributed Systems, 2012, 23(5): 800 – 808.

[5] Chen Jun, Xu Du, et al. A positive-first and negative-first fault-tolerant routing schemes for concave and convex faults[A]. International Conference on Future Computer and Communication [C]. Wuhan: IEEE Computer Society, 2010.53 – 58.

[6] Dally W J, Seitz G L, Deadlock-free message routing in multiprocessor interconnection networks[J]. IEEE Transaction on Computers, 1987, 36(5): 547 – 553.

[7] Glass C J, Ni L. The turn model for adaptive routing[J]. Journal of the ACM, 1994, 41(5): 874 – 902.

[8] Chiu Geming. The odd-even turn model for adaptive routing [J]. IEEE Transaction Parallel and Distributed Systems, 2000,

11(7):729 – 738.

[9] Chien A, Kim J H. Planar-adaptive routing: low – cost adaptive networks for multiprocessors[J]. Journal of The ACM, 1995, 42(1):91 – 123.

[10] Fu Binzhang, Han Yinhe, et al. An abacus turn model for time/space-efficient reconfigurable routing [A]. 38th International Symposium on Computer Architecture[C]. San Jose, CA: ACM Sigarch Computer Architecture News, 2011.259 – 270.

[11] Xiang Dong, Wang Qi, et al. Deadlock-free fully adaptive routing in tori based on a new virtual network partitioning[A]. 37th International Conference on Parallel Processing[C]. Portland, Oregon: IEEE Computer Society, 2008.612 – 619.

[12] 顾华玺,刘增基,等. Torus 网络中分布式自适应路由算法[J].西安电子科技大学学报(自然科学版),2006,33(3):352 – 358.
Gu Huaxi, Liu Zengji, et al. Distribute adaptive routing in torus networks[J]. Journal of Xidian University (Science), 2006, 33(3):352 – 358. (in Chinese)

[13] Duato Jose. A new theory of deadlock-free adaptive routing in wormhole networks[J]. IEEE Transaction on Parallel and Distributed Systems, 1993, 4(12):1320 – 1331.

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

[15] Ascia G, Catania V, et al. Implementation and analysis of a new selection strategy for adaptive routing in networks-on-chip[J]. IEEE Transaction on Computers, 2008, 57(6):809 – 820.

[16] Schwiebert L, Jayashimha D N. A necessary and sufficient condition for deadlock-free wormhole routing[J]. Journal of Parallel and Distributed Computing, 1996, 32(1):103 – 117.

[17] Ramanujam R. S, Lin B. Weighted random routing on torus networks[J]. IEEE Computer Architecture Letters, 2009, 8(1):1 – 4.

[18] Linder D H, Harden J C. An adaptive and fault tolerant wormhole routing strategy for k-ary n-cubes[J]. IEEE Transaction on Computers, 1991, 40(1):2 – 12.

[19] Xiang Dong. Deadlock-free adaptive routing in meshes with fault-tolerance ability using channel overlapping [J]. IEEE Transaction on Dependable and Secure Computing, 2011, 8(1):74 – 88.

[20] Xiang Dong, Zhang Yueli, et al. Practical deadlock-free fault-tolerant routing in meshes based on the planar network fault model[J]. IEEE Transaction on Computers, 2009, 58(5):620 – 633.

[21] Xu Yi, Zhao Bo, et al. Simple virtual channel allocation for high throughput and high frequency on-chip routers[A]. 16th International Symposium on High Performance Computer Architecture [C]. Bangalore, India: IEEE Computer Society, 2010.1 – 11.

[22] 杨盛光,李丽,等.面向能耗和延时的 NoC 映射方法[J].电子学报,2008,36(5):937 – 942.
Yang Shengguang, Li Li, et al. An energy and delay-aware mapping method of NoC[J]. Acta Electronica Sinica, 2008, 36(5):937 – 942. (in Chinese)

[23] Matsutani H, Koibuchi M, et al. Fat H-Tree: A cost-efficient tree-based on chip network[J]. IEEE Transaction on Parallel and Distributed Systems, 2009, 20(8):1126 – 1141.

[24] 赵宏智.2D Mesh 片上网络中交换机服务性能影响的研究及其拓扑改进[J].电子学报,2009,37(2):294 – 298.
Zhao Hongzhi. Study of the impact of switch service performance on 2d mesh network on chip and its improved topology[J]. Acta Electronica Sinica, 2009, 37(2):294 – 298. (in Chinese)

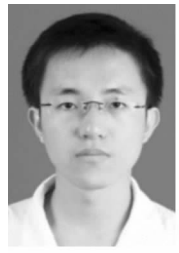
[25] Verbeek Freek, Schialtz Julien. A comment on a necessary and sufficient condition for deadlock-free adaptive routing in wormhole networks [J]. IEEE Transaction on Parallel Distribute System, 2011, 22(10):1775 – 1776.

[26] Verbeek Freek, Schialtz Julien. On necessary and sufficient conditions for deadlock-free routing in wormhole networks [J]. IEEE Transaction on Parallel Distribute System, 2011, 22(12):2022 – 2032.

[27] 马立伟,孙义和.片上网络拓朴优化:在离散平面上布局与布线[J].电子学报,2007,35(5):906 – 911.
Ma Liwei, Sun Yihe. Network-on-Chip topology optimizations: floor-plan and routing on discrete plane[J]. Acta Electronica Sinica, 2007, 35(5):906 – 911. (in Chinese)

[28] Matsutani H, Koibuchi M, et al. Prediction router: A low-latency on-chip router architecture with multiple predictors[J]. IEEE Transaction on Computers, 2011, 60(6):783 – 799.

## 作者简介



**虞志刚**　男,1989 年生于安徽省宿松县.现为清华大学计算机系博士生.研究方向为并行与分布式计算、片上网络路由.
E-mail: yuzg@live.com

**向　东**　男,1966 年生于重庆市.现为清华大学软件学院教授,博士生导师,杰出青年.研究方向为集成电路测试、分布式计算、容错计算.
E-mail: dxiang@tsinghua.edu.cn