Torus 网络中分布式自适应路由算法

顾华玺[1]，刘增基[1]，王 琨[2]，谢启明[1]

(1. 西安电子科技大学 综合业务网理论与关键技术国家重点实验室，陕西 西安 710071；
2. 西安电子科技大学 计算机学院，陕西 西安 710071)

摘要：基于转向模型提出适用于 Torus 网络的 3 种分布式自适应路由算法. 算法将物理网络逻辑上分为虚网络，分组路由按照预定的规则使用不同的虚网络，从而达到无死锁，无活锁的目的. 在二维 Torus 网络中实现这 3 种算法，仅需 3 条虚信道，这是目前 Torus 网络中实现无死锁自适应路由所需虚信道数目的最小值. 对所提算法的性能采用 OPNET 软件进行仿真，拓扑采用 8×8 2D Torus. 结果表明，与广泛用于实际系统的维序路由算法相比，这 3 种算法具备自适应性，在不同流量配置下都能提高网络的时延吞吐性能.

关键词：Torus 网络；路由；死锁；活锁；自适应

中图分类号：TN915.05 **文献标识码**：A **文章编号**：1001-2400(2006)03-0352-07

Distributed adaptive routing algorithms in Torus networks

GU Hua-xi[1], *LIU Zeng-ji*[1], *WANG Kun*[2], *XIE Qi-ming*[1]

(1. State Key Lab. of Integrated Service Networks, Xidian Univ., Xi′an 710071, China; 2. School of Computer Science and Technology, Xidian Univ., Xi′an 710071, China)

Abstract: Based on the Turn Model, three distributed adaptive routing algorithms are proposed for Torus networks, which split the physical network into virtual networks. According to the predefined rules, packets use different virtual networks on their way to destinations. In 2D Torus networks, only three virtual channels are needed. This is the minimum number of virtual channels to implement the adaptive routing algorithm in Torus networks. Simulations of the performance of three algorithms under different configurations are done by OPNET software. The results show that, compared with the popular dimension order routing algorithm, the three adaptive algorithm can achieve better performance under different traffic patterns because of adaptiveness.

Key Words: Torus networks; routing; deadlock; livelock; adaptiveness

直连网络(Direct Interconnection Network，简称 DIN)[1]是一种常见的网络拓扑形式，已经广泛应用于多处理器系统(Multi-processor)，多计算机系统(Multi-computer)，以及集群系统(cluster)中. 网格(Mesh)网络是人们较早研究的一种直连网络. 它结构规则，简单易于实现. 但是 Mesh 结构不对称，会极大地影响网络性能. Torus 网络是一种完全对称的直连网络拓扑形式，近年来针对它的研究越来越多[1,2]. Torus 网络具有很多优秀的网络特性，如规则对称性，路径多样性以及良好的扩展性. 因此它广泛应用于许多商用系统中，例如，2004 年底评出的全球超级计算机 TOP100 中排名首位的 IBM BlueGene/L 就采用 Torus 网络[3]；而另一家通信设备制造商，Avici 公司在其推出的世界上第一台太比特路由器中也采用 Torus 网络作为其交

收稿日期：2005-09-13
基金项目：国家 863 高技术研究发展计划资助项目(2002AA103062)
作者简介：顾华玺(1977-)，男，讲师，博士.

换网络拓扑[4].

路由算法是决定网络性能关键因素之一.避免活锁和死锁一直是路由算法设计中考虑的首要问题.其中活锁是指分组总不能到达目的的结点的情况,通过采用最短路径算法即可避免活锁的产生[1].而死锁是指分组对网络资源的申请形成一种环形依赖关系,构成死锁的所有分组由于无法打破这种环形依赖而无法继续路由的一种情形.如图 1 所示,分组 A,B,C 和 D 分别在结点 1,2,3 和 4 中的缓存里,而分组 A,B,C 和 D 的目的结点分别是结点 3,4,1 和 2,这样,4 个分组都无法申请到下一结点的缓存,因而都无法继续前进,形成死锁,导致整个网络立刻瘫痪.

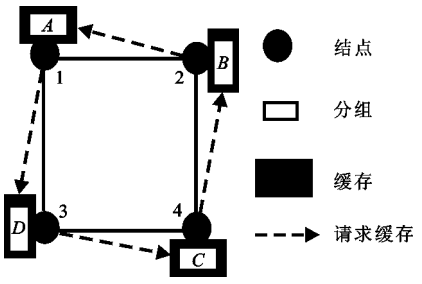

图 1　死锁示意图

多数研究人员通过在路由算法设计中破坏资源的环形依赖关系,达到无死锁的目的.例如,广泛应用于实际系统中的维序路由算法(Dimension Order,简称 DO)[1,2].该算法简单易于实现,但是它是确定性路由算法,不能充分利用网络资源,无法根据网络状态实现自适应路由.针对 Mesh 网络提出的转向模型(Turn Model)[5]克服了上述缺点,实现在 Mesh 网络中无死锁无活锁自适应路由算法.

但是 Torus 网络存在环绕信道,因而引入很多环路,使得转向模型无法直接应用到其中.本文中基于转向模型提出适用于 Torus 网络的 3 种分布式自适应路由算法 WF-T(West First for Torus),NF-T(Negative First for Torus)和 NL-T(North Last for Torus).通过引入虚信道将物理网络逻辑上划分为子网,制定相应的路由规则,从而达到破除环形依赖关系的目的,实现无死锁无活锁的自适应路由算法.在 2D Torus 网络中实现了这 3 种算法只需 3 条虚信道,文[6]中证明这是 2D Torus 网络实现无死锁自适应路由算法所需虚信道数目的最小值.

1　预　　备

为便于理解,首先介绍一些基本概念.

定义 1　直连网络拓扑图　可以抽象为图 G,是由一个非空有限集合 N 和集合 C 构成的二元组,记为 $G=(N,C)$.其中 N 为直连网络的结点集,元素 $n\in N$ 为直连网络中的一个结点.C 为直连网络的信道集,其元素 $c=<n_a,n_b>$,是直连网络中从 n_a 指向 n_b 的单向信道.

定义 2　n 维 Torus 网络　是由 $k_0\times k_1\times\cdots\times k_{n-1}$ 个结点构成,其中 k_i 表示第 i 维的结点数.网络上的每一个结点都可用一个 n 维向量 $(x_0,x_1,\cdots,x_{n-1})$ 表示.其中 $0\leqslant x_i\leqslant k_i-1$.结点 $(x_0,x_1,\cdots,x_{n-1})$ 和结点 $(y_0,y_1,\cdots,y_{n-1})$ 相连接的条件是 iff. $\exists i$ 使 $x_i=(y_i\pm1)\bmod k_i$ 而 $\forall j\neq i$,有 $x_j=y_j$.图 2 示出二维 Torus 网络结构的例子.

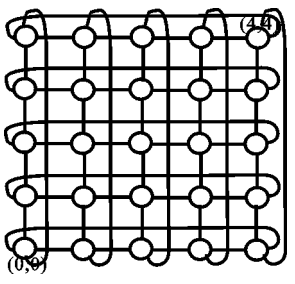

图 2　5×5 2D Torus 结构示意图

定义 3　正(负)信道　给定 2D Torus 网络 $G_t(N_t,C_t)$,若某一维度 i 上的一条有向信道 $<(x_0,x_1),(y_0,y_1)>$ 满足 $x_i=(y_i-1)\bmod k_i$(负:$x_i=(y_i+1)\bmod k_i$),则称该信道为正(负)信道,相应的方向为正(负)方向.正(负)信道的集合标记为 C^+ (C^-).

定义 4　左(右)信道　给定 2D Torus 网络 $G_t(N_t,C_t)$,若某一维度 i 上一条有向信道 $<(x_0,x_1),(y_0,y_1)>$ 满足 $x_i\leqslant\lfloor k_i/2\rfloor$ 且 $y_i\leqslant\lfloor k_i/2\rfloor$(右:$x_i\geqslant\lfloor k_i/2\rfloor$ 且 $y_i\geqslant\lfloor k_i/2\rfloor$),则称该信道为左(右)信道,左(右)信道的集合记为 $(C_t)^L$ ($(C_t)^R$).

定理 1　若直连网络 G 中的所有信道可按一定规则标号,而路由算法 R 沿着信道序号严格递增(或者递减)的顺序路由分组,那么该算法 R 是无死锁的[7].

2D Mesh 网络中的转向模型　转向模型是 Mesh 网络设计无死锁自适应路由算法的经典技术[1],近年来,研究人员基于转向模型设计出许多的算法[1,2,8~10].它的基本思想是通过分析分组在网络中可能的转向,

以及这些转向可能形成的环,禁止一定数量的转向来破除信道依赖关系. 如图 3 所示,通过在顺时针和逆时针两个方向分别禁止一个转向,就达到破除信道环形依赖关系的目的. 根据禁止的不同转向,形成 3 种无死锁、自适应性算法,即西优先(West First,简称 WF)、北最后(North Last,简称 NL)和负优先算法(Negative First,简称 NF).

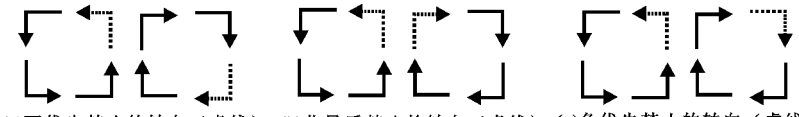

图 3 不同算法禁止的转向示意图

2 Torus 网络中的无死锁自适应路由算法

与 2D Mesh 网络不同,由于存在环绕信道,2D Torus 网络中存在很多环路. 因此上节中的 3 种路由算法 WF,NL 和 NF,应用于 2D Torus 网络时无法破除因网络结构产生的环路. 图 4 示出负优先算法应用于 2D Torus 网络时可能产生的环路. 如图所示的两个环路中,并未出现图 3(c)所示的禁止的转向,因而符合负优先路由算法. 但是在信道依赖图产生环路,算法是存在死锁的,为解决这一问题,引入虚网络.

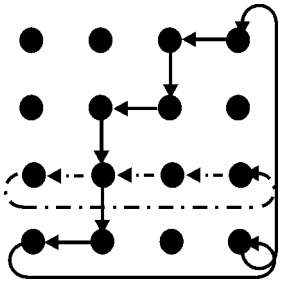

图 4 NF 在 2D Torus 中可能产生的环路

定义 5 虚网络 将物理信道逻辑上划分为若干条虚信道,每条虚信道有自己独立的缓存. 由结点和虚信道组成的网络称为虚网络,记为 $G_v(N,C_v)$.

为保证算法无死锁,将 2D Torus 网络 $G_t(N_t,C_t)$ 的物理信道划分为 3 条虚信道,并组成两个虚网络 1 和 2,分别记为 $G_v^1=(N_t,C_v^*)$ 和 $G_v^2=(N_t,C_v^{\#})$,其中 $C_v^*=C_v^0\cup\{C_v^1\cap[(C_v^+)^L\cup(C_v^-)^R]\}$;$C_v^{\#}=C_v^2\cup\{C_v^1\cap[(C_v^-)^L\cup(C_v^+)^R]\}$,$C_v^i$ 是虚信道 i 组成的集合. 图 5 示出一维环上虚网络的组成,为清楚起见,只画出负信道. 其中按照定义 G_v^1 由 C_v^1 中右信道和 C_v^0 组成,G_v^2 由 C_v^1 中左信道和 C_v^2 组成.

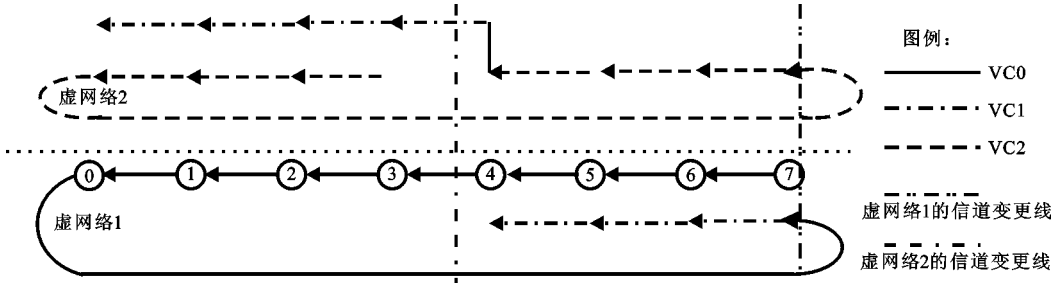

图 5 虚网络组成示意图

定义 6 信道变更线(Channel Alteration Line,简称 CAL) 当分组在某一虚网络中路由通过 CAL 后,将更换虚信道继续路由. G_v^1 中 x 维的信道变更线 CAL_{v1}^x 定义为位于结点 $(\lfloor k_x/2\rfloor-1,y)$ 和 $(\lfloor k_x/2\rfloor,y)$ 之间与信道 $<(\lfloor k_x/2\rfloor-1,y)(\lfloor k_x/2\rfloor,y)>$ 垂直的直线;而 CAL_{v2}^x 定义为位于结点 $(0,y)$ 和 (k_x-1,y) 之间与信道 $<(0,y)(k_x-1,y)>$ 垂直的直线,如图 5 所示. 类似的可定义 CAL_{v1}^y 和 CAL_{v2}^y.

定义 7 虚信道使用规则 当分组注入网络后,首先使用 C_v^0 中的虚信道路由,通过 CAL_{v1} 之后,使用 $(C_v^*-C_v^0)$ 中的虚信道;如果使用 $(C_v^*-C_v^0)$ 中的虚信道的分组发生 90 度转向,则进入 G_v^2 使用 C_v^2 路由,一旦通过 CAL_{v2},就使用 $(C_v^{\#}-C_v^0)$ 直到目的结点.

这样结合负优先算法在 Mesh 网络中禁止的转向，就得到 2D Torus 网络的无死锁自适应的负优先路由算法(NF-T)，描述如下：

NF-T Algorithgm/* Destination node (x_d, y_d), current node (x_c, y_c), input channel c_{in} */
Begin
$\Delta x=x_d-x_c$, $\Delta y=y_d-y_c$;
if $\Delta x<-(k_x-1)/2$　　$\Delta x=k_x+\Delta x$;
if $\Delta x>k_x/2$　　$\Delta x=k_x-\Delta x$;
if $\Delta y<-(k_y-1)/2$　　$\Delta y=k_y+\Delta y$;
if $\Delta y>k_y/2$　　$\Delta y=k_y-\Delta y$;
/* Select the output direction for the packet */
if $\Delta x=\Delta y=0$ send the packet to the local node and EXIT;
if $\Delta x<0$ and $\Delta y<0$ OutDir=$X-$ or $Y-$;
if $\Delta x>0$ and $\Delta y>0$ OutDir=$X+$ or $Y+$;
if $\Delta x<0$ and $\Delta y>0$ OutDir=$X-$;
if $\Delta x>0$ and $\Delta y<0$ OutDir=$Y-$;
if $\Delta x=0$ OutDir=$Y-$ or $Y+$;
if $\Delta y=0$ OutDir=$X-$ or $X+$;
/* Select the output virtual channel for the packet */
if $c_{in}=0$　/* if c_{in} is virtual channel 0 */
　　if OutDir will cross CAL_{v1}
　　OutVC=1;/* Select the virtual channel 1 for the packet */
　　else Out VC=0;
if $c_{in}=1$
　　if OutDir is vertical to the input direction
　　{if the current packet will cross CAL_{v2} along OutDir
　　OutVC=1;
　　else OutVC=2;}
　　else OutVC=1;
if $c_{in}=2$
　　if the current packet will cross CAL_{v2} along OutDir
　　OutVC=1;
　　else OutVC=2;
EXIT.

定理 2　NF-T 算法是 2D Torus 网络无死锁无活锁的路由算法.

证明　首先证明分组在 G_v^1 中路由不会产生死锁. 按如下规则为 G_v^1 中的虚信道分配序号：

① 结点 (x,y) 在 x 维的正向信道序号为 $K+M-n+i\times k_x$；

② 结点 (x,y) 在 y 维的正向信道序号为 $K+M-n+i\times k_y$；

③ 结点 (x,y) 在 x 维的负向信道序号为 $K-M-n+i\times k_x$；

④ 结点 (x,y) 在 y 维的负向信道序号为 $K-M-n+i\times k_y$.

其中 $K=k_x\times k_y$, $M=x+y$, $n=2$(维度), i 为虚信道号 0 或者 1.

图 6 示出在 4×3 Torus 网络中，按照上述标号规则对 G_v^1 中每条虚信道标号的结果. 为清晰起见，只画出 x 维的环绕信道，y 维的环绕信道也有类似结果. 从图中不难看出，按照 NF-T 在 G_v^1 中的路由规则，分组的输出虚信道序号总是大于输入虚信道序号. 因此 NF-T 在 G_v^1 中是严格按照序号递增的顺序选择路由的. 由定理 1 可得，NF-T 算法在 G_v^1 中是无死锁的.

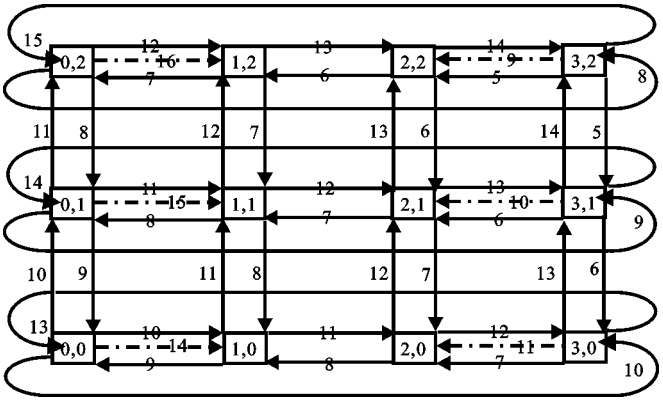

图 6 G_v^1 虚信道序号示意图

如图 5 所示,从 G_v^1 和 G_v^2 的定义可看出,利用 Torus 网络的规则对称性,将 G_v^2 沿负方向平移,直到 CAL_{v2} 与 CAL_{v1} 重合时,G_v^2 就可像 G_v^1 一样采用上述相同的标号规则,类似地可以证明 NF-T 在 G_v^2 中是无死锁的.

最后,由 NF-T 算法可得,分组产生后在 G_v^1 中路由,进入 G_v^2 后,将不再回到 G_v^1 中,即不会产生跨虚网络的环路. 综上可得,NF-T 算法是无死锁的.

另一方面,NF-T 算法总是使分组沿着最短路径路由,因此不会发生活锁. 定理 2 得证.

以上是 2D Torus 网络中 NF-T 算法的介绍,当 NF-T 算法应用到更高维的 Torus 网络中时,NF-T 需要更多的虚信道来保证无死锁. 分组首先在 0 和 1 维构成的平面上,按照 NF-T 算法进行路由,当完成 0 维(或者 1 维)路由后,继续在 1(或者 0)和 2 维构成的平面上同样按照 NF-T 算法继续路由,直到最后在 $n-1$ 和 n 维构成的平面到达目的结点. 在每个平面仍然分为两个虚网络,虚信道的使用规则同定义 7. 很容易证明 NF-T 在高维 Torus 网络是无死锁无活锁的.

与 NF-T 类似的,很容易得到 Tours 网络中的 NL-T 和 WF-T 算法,也是无死锁无活锁的自适应路由算法,限于篇幅,这里不再赘述.

3 性能分析

采用网络仿真软件 OPNET 对不同流量配置下所提出算法在 8×8 2D Torus 网络中的性能进行仿真. 业务源是经典假设的泊松(Poisson)源,即源中产生的分组到达服从泊松分布;分组长度服从 46 到 1 500 byte的均匀分布. 链路速率设为 1 Mbit/s. 仿真采用文献中常用 3 种流量模式[1,2,5~10],即均匀流量、热点流量和矩阵转置流量(transpose traffic pattern)模式. 在均匀流量下,每一结点等概率发送分组到其他结点. 热点流量模式下,一个或更多的结点被设为热点,它们将接收到比一般结点更多流量. 对单个热点的情况进行仿真,其位置随机产生,它将比其他结点多获得 10%的流量. 矩阵转置流量模式下,结点 (i,j) 只发送消息给结点(j,i).

采用平均端到端时延和归一化的吞吐来衡量算法性能,前者是指分组从注入网络到注出网络所经历的平均时间;后者是网络成功传送分组数与注入网络分组数之比. 3 种自适应算法都使用 3 条虚信道. 由于维序只能使用偶数条虚信道[1],对 VC 等于 2 和 4 的维序算法进行仿真. 一般来说,虚信道的数目越多网络性能越好. 但是相应的会增加实现的成本.

图 7 示出 4 种算法在均匀流量下的性能比较. 从图中可看出,在中低负载情况下,4 种算法无论是时延特性还是吞吐特性相近. 当网络负载超过 40%时,使用两条虚信道的 DO 开始达到饱和. 而 3 种自适应算法在 48%左右相继达到饱和. 使用 4 条虚信道的维序算法取得较好的性能,在 60%处才饱和. 这是因为维序路由算法包含能反映均匀流量特性的全局性信息,因此与自适应算法性能相当. 由于仿真中缺乏拥塞控制,4 种算法在过饱和点都出现吞吐下降的趋势,但这不是本文中所要研究的.

图 8 示出 4 种算法在热点流量下的性能比较. 与图 7 相比,由热点导致的时延增高在低流量(小于

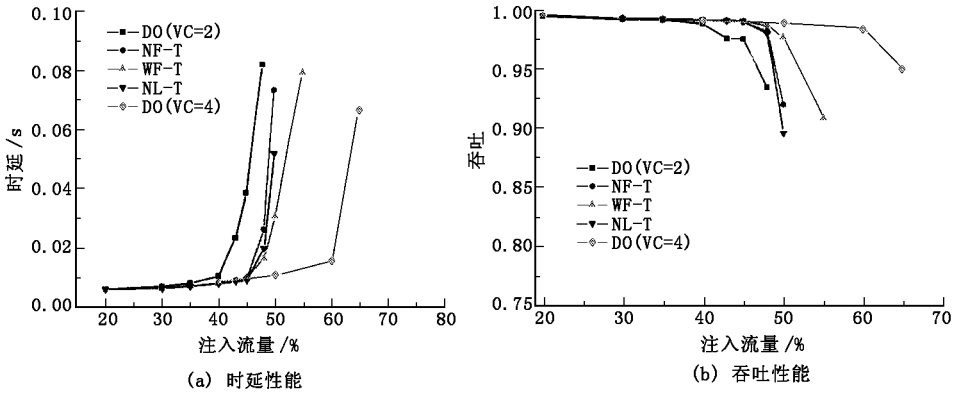

图 7　均匀流量下 4 种算法性能比较

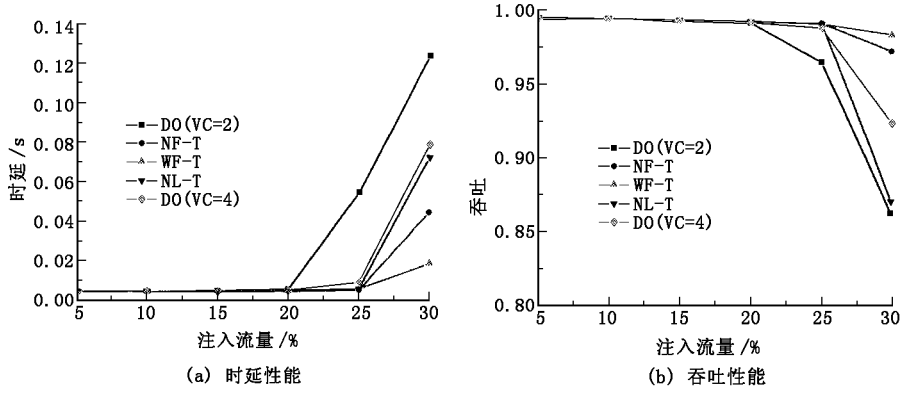

图 8　热点流量下 4 种算法性能比较

20%)负载下并不明显. 但是热点的存在使得 4 种算法的饱和点都提前. 从图中可看出,维序算法无论是使用两条还是 4 条虚信道,性能下降都非常快,分别在网络负载达到 20%和 25%左右就饱和,时延也都高于 3 种自适应算法. WF-T,NF-T 和 NL-T 算法由于存在自适应性,能够在一定程度上缓解热点流量的恶性作用,使分组绕开热点,从而减轻热点周围链路的负载,进而将饱和点推迟. 例如性能最好的 WF-T 比确定的维序路由算法吞吐性能改善 5%(VC 等于 4)和 10%(VC 等于 4).

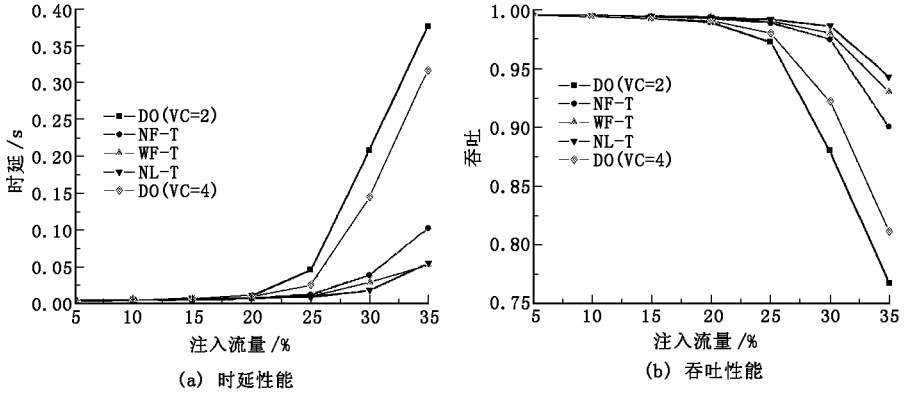

图 9　矩阵转置流量下 4 种算法性能比较

图 9 示出 4 种算法在矩阵转置流量下的性能比较. 从图中可看出,自适应算法的优越性更加明显. 即使是使用 4 条虚信道,维序算法在网络负载 25%左右已经开始饱和. 而 3 种自适应算法都是在 30%以后才开始饱和的,并且 3 种自适应算法的时延始终都低于维序算法. 图 8 和图 9 的结果表明,当流量不均匀时,具备

自适应的算法能够有效地提高网络性能.确定性算法只为每对源目的结点提供一条路径,因此无法应对不均匀的网络流量模式.

4 结 束 语

提出适合于 Torus 网络的无死锁自适应性能路由算法,通过引入虚信道,将物理网络逻辑上划分为虚网络,使分组按照预定的规则使用不同的虚网络进行路由,从而达到无死锁,无活锁的目的.仿真结果表明,由于算法引入自适应性,与当前流行的维序路由算法相比,提高网络性能.在均匀流量下,3 种自适应算法的饱和点分别在 45%,25%和 30%左右,而维序算法分别在 40%,20%和 20%左右(VC 等于 2)以及 60%,25%和 25%左右(VC 等于 4).下一步的工作将研究如何利用 3 种算法的自适应性,使其具有一定的容错能力,即在网络中存有故障结点的情况下,这 3 种算法能够继续工作.

参考文献:

[1] Dally W, Towles B. Principles and Practices of Interconnection Networks[M]. San Francisco: Morgan-Kaufmann Press, 2004.

[2] Liu Guqing, Chen Zhen, Qiu Zhiliang, et al. Research on the Link Traffic and Delay of a Direct Networks[J]. Journal of Xidian University, 2003, 30(7): 96-100.

[3] Adiga N R, Almasi G S. An Overview of the BlueGene/L Supercomputer[A]. Proc of Super Computing 2002[C]. Baltimore: IEEE, 2002. 1-22.

[4] Gu Huaxi, Qiu Zhiliang, Liu Zengji, et al. A New Fault-tolerant Routing Algorithm in a Novel Interconnection Network [J]. Journal of Xidian University, 2003, 30(7): 108-114.

[5] Glass C, Ni L. The Turn Model for Adaptive Routing[J]. Journal of ACM, 1994, 41(5): 874-902.

[6] Cypher R, Gravano L. Requirements for Deadlock-Free, Adaptive Packet Routing[J]. SIAM J on Computing, 1994, 23 (6): 1266-1274.

[7] Dally W J, Seitz C L. Deadlock-free Message Routing in Multiprocessor Interconnection Networks[J]. IEEE Trans on Computer, 1987, 36(5): 547-553.

[8] Chiu G. The Odd-even Turn Model for Adaptive Routing[J]. IEEE Trans on Parallel and Distributed Systems, 2000, 11 (7): 729-738.

[9] Jouraku A, Koibuchi M, Amano H, et al. Routing Algorithms Based on 2D Turn Model for Irregular Networks[A]. I-SPAN'02[C]. Manila: IEEE, 2002. 254-259

[10] Sun Y M, Yang C H, Chung Y C, et al. An Efficient Deadlock-free Tree-based Routing Algorithm for Irregular Wormhole-routed Networks Based on the Turn Model[A]. ICPP 2004[C]. Montreal: IEEE, 2004. 343-352.

(编辑:李维东)