

文章编号: 2095-2163(2022)10-0001-09

中图分类号: TP393

文献标志码: A

# 基于动态调整的弹性片上网络路由算法

李悦瑶<sup>1</sup>, 胡海洋<sup>1</sup>, 王奇<sup>1</sup>, 安鑫<sup>1,2</sup>, 李建华<sup>1,2</sup>

(1 合肥工业大学 计算机与信息学院, 合肥 230601; 2 合肥工业大学 情感计算与先进智能机器安徽省重点实验室, 合肥 230601)

**摘要:** 片上网络具有良好的可拓展性和并行性,能够应对多核处理器的各种通信需求。路由算法对片上网络的性能和效率有较大的影响。网络流量不均衡可导致拥塞和热点,严重影响整个网络的性能。当前学界已提出了本地感知、区域感知和全局感知的自适应路由算法来缓解拥塞问题。然而,这些算法依然存在一些问题,如近视、非全局最优、开销高。本文提出了基于强化学习中 sarsa 奖惩机制思想的路由算法,简称 TCRA。TCRA 针对 Mesh 网络中无死锁的路由限制来确定每一跳的可选输出端口,并基于 sarsa 模型进行非最短路由的策略选择和拥塞值更新。此外,TCRA 中还引入了基于延迟的阈值更新机制,通过增加一些阈值的限制来动态调整拥塞值,从而平衡网络中的流量分布。实验结果表明,与传统的区域感知路由算法相比,TCRA 能够将网络延迟平均减少 30%,并降低 13% 的功耗。与传统的基于强化学习的路由算法相比,TCRA 也能够平均降低 20% 的延迟,并具有较低的路由器面积开销。

**关键词:** 片上网络; 无拥塞路由; 自适应路由算法; 强化学习; sarsa 模型

## Resilient on-chip network routing algorithm based on dynamic tuning

LI Yueyao<sup>1</sup>, HU Haiyang<sup>1</sup>, WANG Qi<sup>1</sup>, AN Xin<sup>1,2</sup>, LI Jianhua<sup>1,2</sup>

(1 School of Computer Science and Information Engineering, Hefei University of Technology, Hefei 230601, China;

2 Anhui Province Key Laboratory of Affective Computing and Advanced Intelligent Machine, Hefei University of Technology, Hefei 230601, China)

**[Abstract]** The on-chip network has good scalability and parallelism, which can cope with the various communication needs of multi-core processors. Routing algorithms have a significant impact on the performance and efficiency of the on-chip network. Unbalanced network traffic can lead to congestion and hotspots, and seriously affect the performance of the entire network. Locally-aware, area-aware and globally-aware adaptive routing algorithms have been proposed to alleviate congestion problems. However, these algorithms still suffer from some problems such as myopia, non-global optimality and high overhead. In this paper, a routing algorithm is proposed based on the idea of sarsa reward and punishment mechanism in reinforcement learning, referred to as TCRA. The optional output port of each hop is determined by TCRA for deadlock-free routing constraints in Mesh networks, and policy selection and congestion value update are performed for non-minimal routing based on the sarsa model. In addition, a delay-based threshold update mechanism is introduced in TCRA to dynamically adjust the congestion value to balance the traffic distribution in the network by adding some threshold limits. Experimental results show that TCRA is able to reduce network latency by an average of 30% and power consumption by 13% compared to traditional area-aware routing algorithms. TCRA is also able to reduce latency by an average of 20% and has a lower router area overhead than traditional reinforcement learning based routing algorithms.

**[Key words]** on-chip network; congestion-free routing; adaptive routing algorithms; reinforcement learning; sarsa models

## 0 引言

随着新兴技术的不断发展,不可拓展问题研究日渐成为时下的研究热点。针对这种现象,片上网络(network on chip, NoC)则应运而生。片上网络有良好的可扩展性和并行性。当前,片上网络已经成为片上多核处理器、GPU 等多核平台的主导型通信方案<sup>[1]</sup>。

片上网络拓扑分为规则拓扑和不规则拓扑,而规则拓扑又以 Mesh 和 Torus 最为经典。其中较为常见的就是 Mesh 结构,设计框架详见图 1。在 Mesh 片上网络中,每个位于中间位置的路由器都有 4 个输入/输出端口以及 1 个本地端口。各个节点以水平和垂直方式相互连接。Mesh 拓扑结构简单,可拓展性好,且具有路径多样性。

先前的工作提出了很多克服拥塞负面影响的策

**基金项目:** 安徽省重点研究与开发计划(202004d07020004); 安徽省自然科学基金项目(2108085MF203)。

**作者简介:** 李悦瑶(1997-),女,硕士研究生,主要研究方向:片上网络。

**通讯作者:** 李建华 Email: jhli@hfut.edu.cn

**收稿日期:** 2022-03-04

略,主要分为3种。第一种<sup>[2]</sup>,结合局部网络状态设计路由算法来避免拥塞。第二种<sup>[2]</sup>,获取用于设计路由算法的全局网络状态来避免拥塞。第三种<sup>[2]</sup>,提出了端到端的流量控制策略,以减少在拥塞的情况下向网络注入数据包。具体来说,第三种方法<sup>[2]</sup>是通过设计一些选择策略,能够在路由数据包的过程中绕过拥塞节点。输出选择应该从一组预定的通道中选择一个通道来将消息转发到下一跳。如何从一组替代路径中选择并不拥挤的路径是很大的挑战。

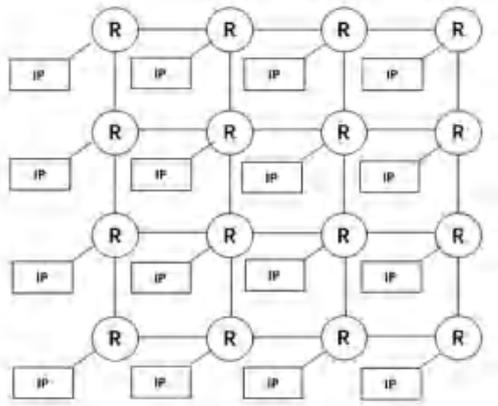


图1 4×4 Mesh 拓扑

Fig. 1 4×4 Mesh topology

强化学习(Reinforcement learning, RL)是机器学习的范式和方法论之一。随着多学科融合的发展,强化学习也得到了更多来自机器学习领域以外的关注。强化学习的原理机制如图2所示。



图2 强化学习

Fig. 2 Reinforcement learning

如今,绝大多数基于强化学习的路由算法都是以Q-Learning模型为基础展开的,但是相比于Q-Learning, sarsa学习更适用于在线学习,也更加保守。使用Q-Learning还会带来因过于贪婪而降低实际效果的风险。

本文的目标是能够充分利用强化学习的奖惩原理,克服常见的Q-learning算法的弊端,并将这种理念应用于片上网络中,提出利用其它模型来设计一种新的路由算法,该算法简称TCRA。

## 1 相关工作

片上网络中的网络负载不平衡会导致吞吐量降

低和网络延迟增加。自适应路由算法能够结合当前网络环境拥塞情况,有意识地避开拥塞区域的路由器,并做出最佳的路径选择判断。下面将围绕本文研究内容阐释分析关于路由算法的一些工作。

### 1.1 自适应路由算法

为了达到避免拥塞的目的,自适应路由算法可以分为3类,分别是:本地感知自适应路由算法、区域感知自适应路由算法、全局感知自适应路由算法。对此将给出探讨分述如下。

本地感知自适应路由算法仅仅使用当前路由器所在节点的局部信息作为拥塞的衡量手段。Kim等人<sup>[3]</sup>使用相邻节点的缓冲区的可用程度作为拥塞指标,而Singh等人<sup>[4-5]</sup>则根据输出队列的长度来作为拥塞指标。Wang等人<sup>[6]</sup>提出了一种路由由独立的并行缓冲区结构的管理方案。该方法使用相邻节点的空闲并行缓冲区的数量作为拥塞度量指标,选用的仲裁策略是优先考虑来自高度拥塞的上游节点的数据包,并将其路由到不太拥塞的下游节点。

Gratz等人<sup>[7]</sup>提出区域拥塞感知路由算法(RCA)。该方法在轻量级监控网络上传递从远程节点收集的拥塞信息。在每一跳的过程中,将传入的拥塞信息与本地拥塞信息聚合,然后再进一步传播到网络中。在上述方法的基础上,Ma等人<sup>[8]</sup>提出了另一种区域自适应路由算法DBAR,是在选择路由时考虑了有关目的节点的信息,减少了聚合噪声。Xu等人<sup>[9]</sup>又提出一种基于区域树的路由算法,能够通过区分拥塞区域和非拥塞区域来分别选择XY路由算法和基于树的路由算法,从而在保证无死锁前提下有效降低延迟率。An等人<sup>[10]</sup>通过在拥塞位置2跳之内的回溯,来重新规划路线。

全局感知自适应路由技术的现有方法相对较少。Manevich等人<sup>[11]</sup>提出了自适应切换维度有序路由(ATDOR)的NoC架构,该架构实现了一个二级网络,将每个节点的拥塞信息传输到每一个专用节点,利用其来为网络中的每个源目的地对选择XY或YX。Ramanujam等人<sup>[12]</sup>提出了DAR算法,该算法添加了一个单独的网络,用于与网络的其余部分通信其本地拥塞信息。然后每个节点确定由其来为候选输出端口之间的特定目的地分配的流量。Ramakrishna等人<sup>[13]</sup>提出GCA路由算法,GCA通过在数据包头部中嵌入“捎带”状态信息来消除对边带网络的需要。

### 1.2 基于RL的路由算法

基于Q-Routing的模型已经在等多篇文献

中<sup>[14-16]</sup>进行了研究,但是在片上网络背景下的研究却仍然少见。Feng 等人<sup>[17]</sup>提出了一种可重构容错偏转路由算法 (FTDR)。该算法通过强化学习方法使用 2 跳故障信息来重新配置路由表。同时利用 Q-Routing 方法来容忍故障并在每对交换机之间找到一条存在的路径。Puthal 等人<sup>[18]</sup>也在这种模型的基础上提出了 C-Routing 方法。这是一种基于层次集群的自适应路由,通过在 2 条不同的路径上发送数据包到目的地,在核心之间均匀分布流量。Ebrahimi 等人<sup>[19]</sup>提出一种片上网络中高度自适应路由算法的摄取感知学习模型 (HARAQ),该算法在每对源交换机和目标交换机之间提供了广泛的替代路径,以便在网络中更自适应地路由消息。Farahnakian 等人<sup>[20]</sup>提出一种优化的 Q-Learning 模型并使用动态编程根据从每个输出通道发送消息的延迟估计在网络上进行流量分配的优化。Samala 等人<sup>[21]</sup>提出一种基于强化学习路由算法,旨在解决一些故障网络问题。算法主要是根据变化的链路和路由器故障的数量来动态地缩放网络规模,达到检查选择路径最优的目的。Reza<sup>[22]</sup>提出基于强化学习思想的自配置片上网络,主要考虑的是产热热点问题。

绝大多数基于强化学习的路由算法都是以 Q-Learning 模型为基础展开的,但是如前所述,sarsa 学习<sup>[23]</sup>相比较于 Q-Learning 更适用于在线学习,也更加保守。使用 Q-Learning 会有因为过于贪婪而降低实际效果的风险。

## 2 基于 sarsa 奖惩思路的阈值限制路由算法

### 2.1 TCRA 整体架构

本文提出了 TCRA 算法。该算法首先基于 sarsa 的奖惩机制以路由器延迟时间作为衡量标准对路由表中的数据进行更新。同时增加 2 个阈值,分别为延迟阈值和更新阈值。只有当路由器实际延迟时间大于延迟阈值时才对上游路由器中的对应表值进行更新。当某个上游路由器在超过更新阈值时间内一直没有发生更新,那么将该路由器中的值进行初始化。

图 3 给出了 TCRA 机制的整体工作流程。在数据包途经的每一跳路由器上,首先根据当前节点和目标节点之间的位置关系,判断出所有可用的端口,然后有 90% 的概率选择其中表值最小端口作为输出端口。当数据包到达下一跳的路由器时,下游路由器能够记录到该跳的传输延迟时间。将这个时间

与延迟阈值进行对比以确定是否将该时间反馈给上游路由器。除此之外,更新阈值用来判断任一节点是否长期未更新,来判断是否将表值重置。

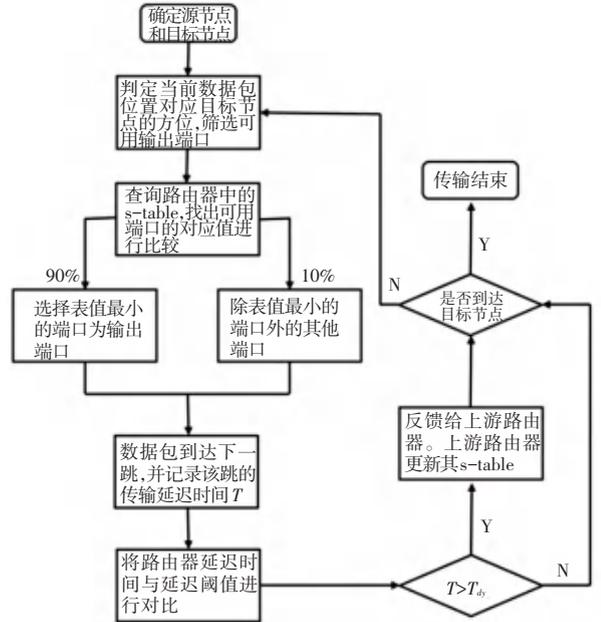


图 3 方法流程图

Fig. 3 TCRA flow chart

### 2.2 一种高效的 s-table 数值更新策略

TCRA 算法的前提是创建无死锁路由,可选择虚拟通道以及采取强化学习的学习动作,接下来将对此研究做专题论述。

#### 2.2.1 死锁避免方法

本文的实验是应用在 2D-Mesh 网络上面的。在 Mesh 网络中,不可避免一个问题就是死锁问题,由于自适应路由的非规律性选择,很容易在网络中产生循环依赖,从而造成网络死锁。为了解决这个问题,本文使用了 mad-y 方法<sup>[24]</sup>。在这种方法中,x 维度上只有一个通道,而 y 维度上有 2 个虚拟通道 (VC)。图 4 说明了 mad-y 网络中使用的路由器。

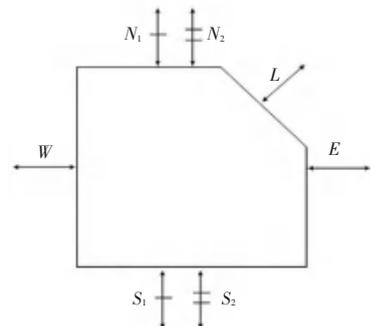


图 4 mad-y 路由器结构

Fig. 4 mad-y router architecture

mad-y 方法禁止一些固定的转弯来避免 2D-NoC 中的死锁。如图 5 所示。如果能够从每个循环中消除至少一圈,参见图 5(a),那么就可防止 90°转弯中所产生的循环依赖的可能。当涉及到 0°转弯时,包含有 2 种类型:改变虚拟通道的转弯和不改变虚拟通道的转弯。同样为了避免产生循环依赖,该方法只允许从  $VC_1$  到  $VC_2$  的转弯,参见图 5(b)。由于 mad-y 方法是一种最小路由算法,完全禁止 180°转弯。但是,本文参考了文献[19]提出了一种允许 180°转弯的新路由方法。该方法修改了 mad-y 的编号机制,并禁止了一些可能导致死锁的转弯,在此基础上增加了一些 180°转弯,改进了输出通道的可选择性。由图 5(c)可知,图 5 中显示了允许的 180°转弯。

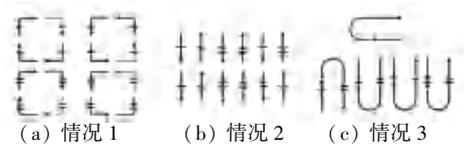


图 5 路由器可用转弯

Fig. 5 Routers available for turning

为了在避免死锁的同时实现最大的适应性,对于输入通道和目标开关位置的每个组合,参考文献[19]所提出的方法,筛选了所有符合条件的 0°、90°和 180°转弯。当消息通过输入通道之一到达时,该到达路由器能够确定一个或多个潜在的输出通道来传递消息。路由决策基于当前交换机和目标交换机的相对位置(即在以下 8 种情况之一内:北(N)、南(S)、东(E)、西(W)、东北(NE)、西北(NW)、东南(SE)和西南(SW)。所有允许的可选择输出通道,见表 1。

表 1 目标路由器对于位置以及可用输出口

Tab. 1 Location and available output ports for target routers

	N	S	E	W	NE	NW	SE	SW
L	$N_1, N_2, S_1, W$	$N_1, S_1, S_2, W$	$N_1, N_2, S_1, S_2, E, W$	$N_1, S_1, W$	$N_1, N_2, S_1, S_2, E, W$	$N_1, S_1, W$	$N_1, N_2, S_1, S_2, E, W$	$N_1, S_1, W$
$N_1$	$N_2, S_1, W$	$S_1, S_2, W$	$N_2, S_1, S_2, E, W$	$S_1, W$	$N_2, S_1, S_2, E, W$	$S_1, W$	$N_2, S_1, S_2, E, W$	$S_1, W$
$N_2$	-	$S_2$	$S_2, E$	-	$S_2, E$	-	$S_2, E$	-
$S_1$	$N_1, N_2, S_1, W$	$N_1, S_1, S_2, W$	$N_1, N_2, S_1, S_2, E, W$	$N_1, S_1, W$	$N_1, N_2, S_1, S_2, E, W$	$N_1, S_1, W$	$N_1, N_2, S_1, S_2, E, W$	$N_1, S_1, W$
$S_2$	$N_2$	-	$N_2, E$	-	$N_2, E$	-	$N_2, E$	-
E	$N_1, N_2, S_1, W$	$N_1, S_1, S_2, W$	$N_1, N_2, S_1, S_2, E, W$	$N_1, S_1, W$	$N_1, N_2, S_1, S_2, E, W$	$N_1, S_1, W$	$N_1, N_2, S_1, S_2, E, W$	$N_1, S_1, W$
W	$N_2$	$S_2$	$N_2, S_2, E$	-	$N_2, S_2, E$	-	$N_2, S_2, E$	E

### 2.2.2 路由表设计及结构

TCRA 算法的关键之一是根据每个路由器上的路由表选择最合适的输出口进行数据包的传送。由于不需要所有可能的数值,所以该路由表已经在最大程度上减少了额外的面积开销。在每一个输入端口对应的表格行中,可用输出口都有对应的数值,作为最大传输延迟的映射,能够反映该段时间内选择某个端口的拥塞程度。具体路由表设计见表 2。表 2 中,行表示可选输出口,列表示目标路由器对应于当前位置的方向。

表 2 s-table 表格

Tab. 2 s-table tables

	$N_1$	$N_2$	$S_1$	$S_2$	E	W
N						
S						
E						
W						
NE	3	4	8	7	4	10
NW						
SE						
SW						

最初, s-table 单元仅仅对于最小路径输出通道设置初始值为零,以强制无拥塞网络的最小路径路由。在图 5 中可以看到,由于 mad-y 方法中不允许某些转弯,因此某些单元格为空。

### 2.2.3 路由表更新策略及 sarsa 公式详细介绍

TCRA 算法是以强化学习的 sarsa-learning 算法为基本思想。本文中路由问题中的动作是根据目标区域的每一跳可能会产生的传输时间决定的,以尽量选取最优的输出端口。为了给传入数据包选择最佳的输出口,该方法保留了一个名为 s-table 的查找表。s-table 的每个单元格都包含目标方向和可选的输出通道的值。单元格中的值越高,代表输出通道的区域越拥挤。s-table 中的表值表示从所选输出口向相应方向发送数据包的延迟时间产生的映射值,参见表 2。每个路由器都有一个 s-table 来选择最优输出口。每个路由器在发送一个数据包后都会等待并接收一个反馈消息。反馈消息为从上游路由器选定输出口发送数据包开始,到下游路由器接收到该数据包的时间为止所产生的延迟时间。这

里需用到的数学公式为:

$$S_d(i, j) = S_d(i, j) + \alpha(T_{ij} + \gamma S_d(j, k) - S_d(i, j)) \quad (1)$$

其中,  $\alpha$  表示学习率, 也就是网络考虑当前所采取的动作对其当前整体的影响有多大;  $\gamma$  表示折扣率, 即与动作反馈相关的一个变量,  $\gamma$  若选取为 0, 则该变量不对系统造成任何影响。公式(1) 主要包含有 4 个不同的变量。片上网络中的数据包每行进一步, 表值都会根据公式(1) 至多进行一次更新。

在公式(1) 中,  $T_{ij}$  的值用路由器延迟时间来衡量, 即数据包从上游路由器输出端口输出开始到下游路由器输出端口输出结束的整个时间。

除此之外, TCRA 还增加了一个阈值限制条件, 否则每一跳都更新带来的额外开销是不可估计的, 具体的限制机制将在下一节介绍。

## 2.3 基于延迟的阈值更新

### 2.3.1 延迟阈值详细介绍

TCRA 是应用在 2D-Mesh 网络上面的。当数据包从上游路由器传输到下游路由器时, 如何将数据包延迟时间重新传给上游路由器, 这就需要增加额外的线路来传输数据。在文中, TCRA 忽略增加的额外线路带来面积开销方面的优化, 仅仅考虑减少每次更新所带来的功耗开销以及对整体网络环境的影响。对此, 首先提出延迟阈值的概念。当到达下游路由器的延迟时间没有达到延迟阈值时, 那么可以认为这一跳的整个的传输过程没有遇到拥塞问题, 此时此刻上游路由器中的 s-table 中该方向该数据端口的值仍旧比较低, 所以不需要进行一次更新, 也就不需要将该反馈消息传回上游路由器。如果下次有同样情况发生时, 仍旧可以优先选择该输出端口进行传输。

反之, 如果传输延迟时间大于延迟阈值, 则意味着数据包在传输过程中遇到了一些阻塞, 此时把消息反馈给上游路由器。上游路由器根据这些反馈消息, 更新路由表的值。因为拥塞, 所以 sarsa 算法会产生负反馈, 对应的值会相应增加。这样, 下次再出现相同选择的情况下, 该端口就不是最优选择了。

延迟阈值的设定目标是控制路由表更新的频率, 过高的阈值设定会降低网络传输的路径灵活度, 而过低的阈值设定会导致频繁的路由表更新, 造成过高的信息传递开销。研究中通过实验来测定合适的阈值。在一定的流量模式和拓扑结构下, 数据包在网络中传输的最大延迟会在一定范围内波动, 其平均值称之为  $T_{max}$ , 接下来分别将延迟阈值设定为

$0.6T_{max}$ ,  $0.8T_{max}$ ,  $T_{max}$ ,  $1.2T_{max}$  和  $1.4T_{max}$  下。以  $T_{max}$  为标定的归一化结果如图 6 所示。

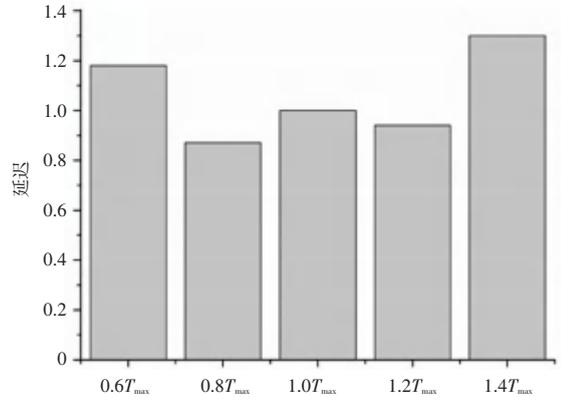


图 6 基于  $T_{max}$  的延迟

Fig. 6 Delay based on  $T_{max}$

实验结果表明, 当把延迟阈值设置为最大延迟时间的 0.8 倍时, 基本是最佳选择。

### 2.3.2 更新阈值详细介绍

研究发现增加了延迟阈值后会造成另外一个问题, 即如果该端口对应的数值因为上次的负反馈而持续停留在一个很高的结果的话, 该端口就可能存在处于长期不更新、也不被选择的状态。

因此提出更新阈值  $Tud$  的概念, 当 s-table 中的某个值超过更新阈值时间后都没有进行过更新, 就认为该端口已经恢复至空闲状态, 则将其对应的值重新初始化。

$Tud$  的设定是为了防止部分节点处于停滞状态、不再参与路由表更新, 研究中在该实验部分将运行周期始终设定为 10 000, 而  $Tud$  的值在设定为总运行周期的 1/10, 1/20, 1/50, 1/100, 1/200 和 1/500 后的网络延迟表现如图 7 所示, 其中以总运行周期的 1/10 为标定进行归一化。

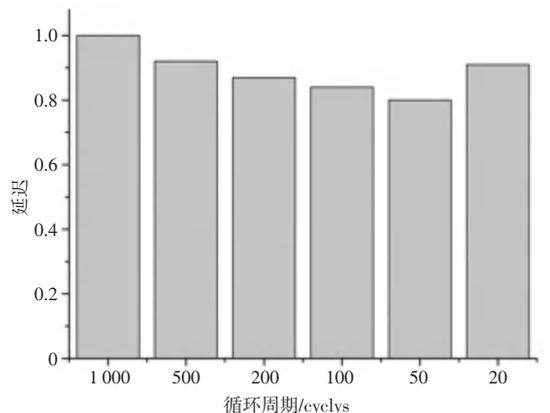


图 7 基于总运行周期 1/10 的延迟

Fig. 7 Delay based on 1/10<sup>th</sup> of the total operating cycle

图 7 中, 设置更新阈值为总运行周期的 1/200

时性能最佳。增加了更新阈值概念后,网络浪费率会有相对应的较少。

## 2.4 非贪婪选择策略

为了避免出现因为过于贪婪而带来的可能不良后果,在本文中设置这样一个机制,90%的情况下,仍旧保持选择对应值最小的输出端口进行路由,而还有10%的可能是选择其他的输出端口。对于这些剩余的其他端口,能够被选中的概率是完全相等的。具体如式(2)所示:

$$P_d(i, j) = \begin{cases} 1 - \varepsilon & \min S_d(i, j) \\ \varepsilon & \text{其他} \end{cases} \quad (2)$$

其中,  $\varepsilon$  表示随机的概率。在本文中,将其设定为0.1。因此90%的情况下都会选择表值中数值对应最小的端口作为输出端口,但是也有10%的概率会选择其他端口。对于第二种情况,余下的端口被选择的概率是相等的。

## 3 实验及结果分析

### 3.1 实验方法

本文对于 TCRA 算法的评估,主要在功耗开销以及延迟两个方面进行了实验。仿真实验分为2个部分。对此做探讨论述如下。

(1) 开销与功耗。实验选择的工具是 Design Compiler,由 Synopsys 公司开发,并使用 Verilog HDL 语言进行硬件逻辑综合。通过在 90 nm TSMC 库下使用 Design Compiler 合成修改后的通讯模块,工作电压为 1 V,工作频率为 1 GHz。

(2) 延迟。使用 noxim 片上网络模拟器进行了仿真以确定每个网络的延迟特性。数据包长度均匀分布在 1~5 个 flits 之间。对于所有路由器,每个输入通道的缓冲器大小为 8 个 flit。请求成功率定义为向网络接口成功注入消息的次数与尝试注入次数之比。模拟器预热了 12 000 个周期,然后平均性能测量了另外 200 000 个周期。使用了 2 个合成流量模型,分别是均匀随机流量模型和热点流量模型。具体参数配置见表 3。

### 3.2 开销与功耗分析

表 4 分别对比了 4 种方案的面积开销、平均功耗和最大功耗。表 4 中,对照组选取的是 DBAR、C-Routing 和 HARAQ。

从面积开销上看,TCRA 和采用类似原理的 HARAQ 算法相当,区别在于选择策略和路由计算模块的硬件有所差异;而在面积开销这方面,TCRA 处于 DBAR 和 C-Routing 开销之间。对于未分区片

上网络来说,DBAR 的性能与传统局部自适应算法类似,因此只需要有限跳内的非全局信息的聚合。而 C-Routing 算法虽然不增设虚拟通道,但是需要一个较大的路由来满足频繁更新的要求。

表 3 仿真模拟器参数配置

Tab. 3 Simulation simulator parameters configuration

参数	配置
Topology	8×8 2D-Mesh 14×14 2D-Mesh
Buffer size	8 flits
Package size	1~5 flits
Simulation	200 000 cycles
Traffic pattern	Hotspot& Uniform
Virtual channel	1(y direction)

表 4 面积开销与功耗分析比较

Tab. 4 Comparison of area overhead and power consumption analysis

Network platforms	Area/mm <sup>2</sup>	Avg. Power/W (dynamic & static)	Max. Power (dynamic&static)
DBAR	6.794	2.41	3.33
C-Routing	6.954	2.52	3.46
HARAQ	6.822	2.81	3.06
TCRA	6.863	2.75	2.91

TCRA 由于路由表的维持和重路由的增加,所以平均功耗较 C-routing 和 DBAR 要高出 10% 以上;而由于 HARAQ 也采用了大型路由表,在平均功耗方面与 TCRA 基本相当。最大功耗是在热点模式下读取的,热点周边的拥塞会导致大量数据包需要更长的等待时间才能到达终端,从而极大地增加了动态功耗。相较于工作原理类似 HARAQ 算法,TCRA 在面积开销方面略高,但是由于算法的合理性,能够缓解热点周边的拥塞情况,在热点模式下 TCRA 展示了更好的性能,较 HARAQ 方案降低了约 5% 的最大功耗。

### 3.3 延迟分析

为了评估本文提出方案的有效性,做了与 DBAR、C-Routing、HARAQ 以及 Q-RT<sup>[25]</sup> 算法的比较实验。各对比算法分析比较结果见表 5。DBAR 是利用局部拥塞信息的自适应路由,C-Routing 是利用 Q-learning 技术的自适应路由,而 HARAQ 在 Q-learning 模型的基础上通过拥塞值的计算增加了多条路径并改进了选择策略。Q-RT 是最近提出了动态检测全局网络拥塞状态、并利用 Q-learning 方法寻找最佳路径的方法。

表 5 各对比算法分析比较

Tab. 5 Analysis and comparison of each comparison algorithm

	最小路由	q-learning 模型	sarsa 模型	设定阈值限制
DBAR	√	×	×	×
C-Routing	√	√	×	×
HARAQ	×	√	×	×
Q-RT	×	√	×	×
TCRA	×	×	√	√

为了与 TCRA 进行对比,4 种对比方法均采用了基于 mad-y 的完全自适应路由函数。接下来将针对在 2 种不同流量模式下产生的平均延迟进行分析。

3.3.1 uniform 流量模式下的性能评价

在 uniform 流量模式中,路由器向具有均匀分布的其他路由器发送消息。图 8 和图 9 中展示了 2 种网络尺寸下的平均延迟时间与平均消息注入速率之间的函数关系。

结果表明,整体来说 TCRA 与其他方法相比均有一定的提升。在低负载情况下,TCRA 与 Q-Routing 方案 (HARAQ, C-Routing 和 Q-RT) 和 DBAR 的性能相当。当负载增加时,DBAR 将呈现一个较大的延迟的增加,无法处理高负载情况下的路由,而其他方案则能够通过学习而实现一个更有效的路由策略。其中,TCRA 会导致最低的延迟。因为不仅能比其它 2 个方案更有效地分配流量,而且还增加了选择性的消息传输方式。

在实际情况下,在 DBAR 和 C-Routing 中,消息的传递使用最小路径,因此在这种流量下,相应的数据包更加有向整个网络中心路由的趋势,从而会在网络中创建大型的永久热点。因此相应地,通过网络中心传输的消息将比使用任何非最小路径的消息延迟得多。由于 TCRA 可以重新路由消息,因此就可以避开拥塞区域,该方案相对来说要比其他方案的性能好得多。采用最小路由和非最小路由并结合智能选择策略,在 8 × 8 网络中 TCRA 在接近饱和点处的平均网络延迟比 C-Routing、DBAR、HARAQ 和 Q-RT 分别降低了 24.8%、34%、7.5% 和 14%。

3.3.2 hotspot 流量模式下的性能评价

在 hotspot 流量模式下,选择一个或多个路由器作为热点,在常规均匀流量的基础上接收额外的流量。本次仿真中,给定一个热点百分比  $H$ ,一条新生成的消息以额外的  $H\%$  的概率直接发送到每个热点路由器。在 8 × 8 和 14 × 14 2D-Mesh 网络模型下,分别模拟了 (4,4) 和 (7,7) 两个热点交换机的热

点流量。在本文的模拟仿真中,设定了  $H = 10\%$ 。其结果如图 10 和图 11 所示。

从图 10 中可以看出,与其他路由方案相比,TCRA 在性能方面表现更佳。以 8 × 8 网络为例,在接近饱和点处的性能改善分别为 C-Routing、DBAR、HARAQ 和 Q-RT 的 21%、33%、5% 和 14.8%。通过观察得知,在 8 × 8 网络中,与 DBAR 相比,下降幅度平均高达 24.4%。实验结果表明,采用非最小方案和合理的强化学习策略可以有效地分配流量。

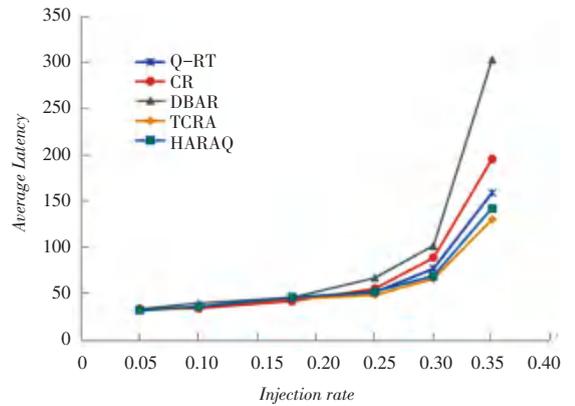


图 8 uniform 流量模式下 8 × 8 Mesh 网络延迟

Fig. 8 8 × 8 Mesh network latency in uniform traffic mode

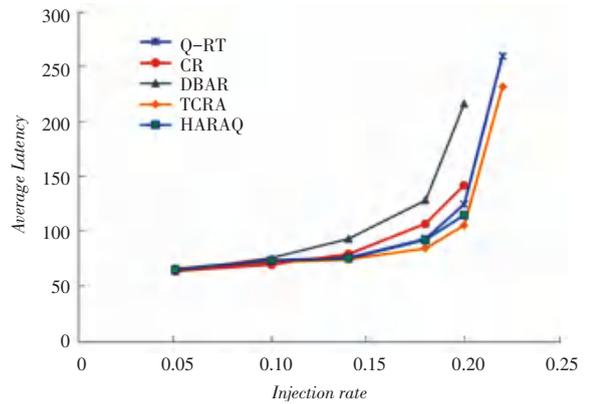


图 9 uniform 流量模式下 14 × 14 Mesh 网络延迟

Fig. 9 14 × 14 Mesh network latency in uniform traffic mode

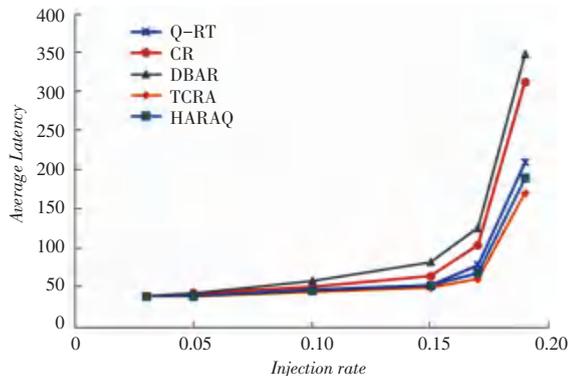


图 10 hotspot 流量模式下 8 × 8 Mesh 网络延迟

Fig. 10 8 × 8 Mesh network latency in hotspot traffic mode

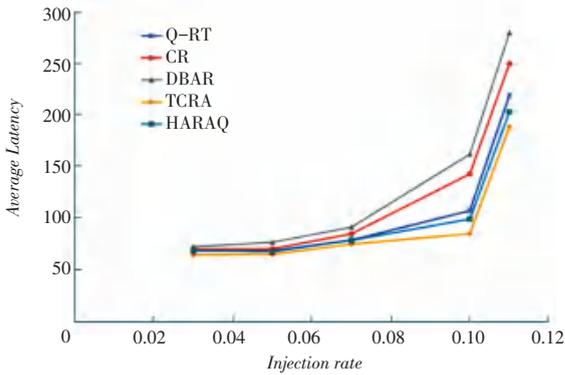


图 11 hotspot 流量模式下  $14 \times 14$  Mesh 网络延迟

Fig. 11  $14 \times 14$  Mesh network latency in hotspot traffic mode

## 4 结束语

片上网络中的网络负载不平衡会导致吞吐量降低和网络延迟增加。而路由算法是有效改进片上网络负载平衡不佳的方法,自适应路由算法能够结合当前网络环境拥塞情况,有意识地避开拥塞区域的路由器,并做出最佳的路径选择判断。

为了能够最大限度地降低网络延迟,本文提出了一种与强化学习方法相结合的片上网络无拥塞路由算法 TCRA。主要是通过  $y$  通道多增加一条虚拟通道并增加转弯限制,由此可以得到每一跳的可选方向和路径,同时增加了延迟阈值和更新阈值,能够有选择性地 将拥塞信息回传给下游路由器,还增设了非贪婪路径选择策略。实验结果表明,采用非最小方案和合理的强化学习策略可以有效地避开拥塞路径,而阈值的设定又能够直接减少额外开销,以最大程度地提高片上网络的系统性能。

## 参考文献

- [1] BENINI L, De MICHELI G. Networks on chips: A new SoC paradigm[J]. Computer, 2002, 35(1):70-78.
- [2] TANG Minghua, LIN Xiaola, PALESI M. Local congestion avoidance in network-on-chip[J]. IEEE Transactions on Parallel and Distributed Systems, 2016, 27(7):2062-2073.
- [3] KIM J, PARK D, THEOCHARIDES T, et al. A low latency router supporting adaptivity for on-chip interconnects[C]// 42<sup>nd</sup> Design Automation Conference. Anaheim, CA, USA: IEEE, 2005:559-564.
- [4] SINGH A, DALLY W J, GUPTA A K, et al. GOAL: a load-balanced adaptive routing algorithm for torus networks[C]//30<sup>th</sup> Annual International Symposium on Computer Architecture. San Diego, CA, USA:IEEE, 2003:194-205.
- [5] SINGH A, DALLY W J, TOWLES B, et al. Globally adaptive load-balanced routing on tori[J]. IEEE Computer Architecture Letters, 2004, 3(1):2.
- [6] WANG Chifeng, HU W H, BAGHERZADEH N. Scalable load

- balancing congestion-aware network-on-chip router architecture [J]. Journal of Computer and System Sciences, 2013, 79(4): 421-439.
- [7] GRATZ P, GROTT B, KECKLER S. Regional congestion awareness for load balance in networks-on-chip[C]// 2008 IEEE 14<sup>th</sup> International Symposium on High Performance Computer Architecture. Salt Lake City, UT, USA:IEEE, 2008:203-214.
- [8] MA Sheng, JERGER N E, WANG Zhiying. DBAR: an efficient routing algorithm to support multiple concurrent applications in networks-on-chip [C]// 2011 38<sup>th</sup> Annual International Symposium on Computer Architecture (ISCA). San Jose, CA, USA:IEEE, 2011:413-424.
- [9] XU C, et al. A region-tree based fault-tolerant routing algorithm for network-on-chip [C]// 2022 IEEE 2<sup>nd</sup> International Conference on Power, Electronics and Computer Applications (ICPECA). IEEE, 2022:72-76.
- [10] AN Jianfeng, YOU Hongjun, SUN Jinhua, et al. Fault tolerant XY-YX routing algorithm supporting backtracking strategy for NoC [C]//2021 IEEE Intl Conf on Parallel & Distributed Processing with Applications, Big Data & Cloud Computing, Sustainable Computing & Communications, Social Computing & Networking (ISPA/BDCLOUD/SocialCom/SustainCom). IEEE, 2021: 632-635.
- [11] MANEVICH R, CIDON I, KOLODNY A, et al. A cost effective centralized adaptive routing for networks-on-chip [C]// 2011 14<sup>th</sup> Euromicro Conference on Digital System Design. Oulu, Finland:IEEE, 2011:39-46.
- [12] RAMANUJAM R S, LIN B. Destination-based adaptive routing on 2D mesh networks [C]// 2010 ACM/IEEE Symposium on Architectures for Networking and Communications Systems (ANCS). La Jolla, CA, USA:IEEE, 2010:1-12.
- [13] RAMAKRISHNA M, KODATI V K, RRATZ P V, et al. GCA: Global congestion awareness for load balance in networks-on-chip [J]. IEEE Transactions on Parallel and Distributed Systems. IEEE, 2016,27(7): 2022-2035.
- [14] KUMAR S, MIKKULAINEN R. Dual reinforcement Q-routing: An on-line adaptive routing algorithm [C]//Proceedings of Artificial Neural Networks in Engineering. Austin: The University of Texas.1997:231-238.
- [15] BOYAN J A, LITTMAN M L. Packet routing in dynamically changing networks: A reinforcement learning approach [C]// Advances in Neural Information, Denver, Colorado: NIPS,1994: 671-678.
- [16] MAJER M, BOBDA C, AHMADINIA A, et al. Packet Routing in Dynamically Changing Networks on Chip [C]//19<sup>th</sup> IEEE International Parallel and Distributed Processing Symposium. Denver, CO, USA:IEEE, 2005:8.
- [17] FENG Chaochao, LU Zhonghai, JANTSCH A, et al. A reconfigurable fault-tolerant deflection routing algorithm based on reinforcement learning for network-on-chip [C]// Third International Workshop on Network on Chip Architectures. Atlanta, Georgia, USA:ACM, 2010:11-16.
- [18] PUTHAL M K, SINGH V, GAUR M S, et al. C-Routing: An adaptive hierarchical NoC routing methodology [C]// 2011 IEEE/IFIP 19<sup>th</sup> International Conference on VLSI and System-on-Chip. Hong Kong, China:IEEE, 2011:392-397.