

一种具有流量均衡性的 NoC 可重构容错路由算法

李 娇^{1,2}, 崇云锋¹, 徐海鹏¹, 郭润龙¹, 冉 峰^{1,2}

(1 上海大学 微电子研究与开发中心, 上海 200072;

2 上海大学 新型显示技术及应用集成教育部重点实验室, 上海 200072)

摘 要: 针对当前片上网络(NoC)容错路由算法不具有流量均衡的特性造成网络出现 congestion 的问题, 提出一种具有流量均衡策略的可重构容错路由算法. 首先, 对现有的故障模型进行分析, 提出一种具有故障感知能力的故障模型, 使得数据包在路由过程中能够掌握网络中的故障情况; 然后, 对算珠转向模型进行优化, 使其适用于存在故障的网络. 最后将优化后的转向模型应用到 LDBR 算法中, 使网络在具有容错能力的前提下能够降低网络 congestion. 实验结果表明, 在故障率较低时, 该算法与对比算法相比吞吐量提高了 9.8%, 数据包延时降低了 12.5%. 随着故障率的提高, 该算法依然具有优势.

关键词: 片上网络; 流量均衡; 可重构; 容错

中图分类号: TN402

文献标识码: A

文章编号: 1000-7180(2018)02-0015-07

A Reconfigurable Fault-tolerance Routing Algorithm with Traffic Balance Strategy in NoC

LI Jiao^{1,2}, CHONG Yun-feng¹, XU Hai-peng¹, GUO Run-long¹, RAN Feng^{1,2}

(1 Microelectronics R&D Center, Shanghai University, Shanghai 200072, China;

2 Key Laboratory of Advanced Display and System Application, Shanghai University, Shanghai 200072, China)

Abstract: The existed fault-tolerance routing algorithm in network-on-chip(NoC) without traffic balance strategy will cause congestion in network. Focusing on this issue, this paper propose a reconfigurable fault-tolerance routing algorithm with traffic balance strategy. Firstly, a fault model with fault-aware capability has been proposed after analyzed the existed fault models, packets can master the fault position based on this fault model Secondly, optimizing the abacus turn model for applying to network with faults. Finally, using the optimized turn model to LDBR algorithm for reduce network congestion. Experimental results show that the proposed algorithm has 9.8% higher throughput rate and 12.5% lower latency in average than contrast algorithm when the network is low fault rate. with the augment of fault rate, the present algorithm still has advantages.

Key words: network-on-chip; traffic balance; reconfigurable; fault-tolerance

1 引言

随着单个芯片中集成的知识产权核(Intellectual Property Core, IP Core)数目不断增加以及芯片特征尺寸的不断减小, NoC 发生故障的概率大幅提高, 为了解决故障带来的 NoC 性能损失, 容错路由算法被认为是一种保证 NoC 性能的有效方法^[1].

在容错路由算法的设计中, 首要保证的是网络不会发生死锁. 解决死锁的方法主要分为基于虚通道法和基于转向模型法, 而采用虚通道会带来较大的面积开销^[2], 不利于减小芯片的特征尺寸. 为此, 基于转向模型的理念, Chiu 提出了奇偶转向模型来解决死锁问题^[3], 并证明了, 只要数据包在路由过程中不会在网络中形成右边界就不会发生死锁. Wu

基于这种奇偶转向模型,提出一种容错路由算法^[4],这种故障区在形成过程中会有大量无故障节点被纳入到故障区中,造成了网络性能的损失.而且,该算法不允许故障节点出现在网络的左(右)边界以及与左(右)边界相邻列上,这在很大程度上降低了网络的容错能力.

为了提高网络的容错能力同时减少故障区中包含的无故障节点数,Fu 提出了一种基于无虚通道的容错路由算法^[5].这种算法提高了网络的容错能力.但是这种算法由于不具有可重构性,容易造成网络发生拥塞,增大了网络的延时.

文献[6]提出一种基于算珠转向模型(Abacus Turn Model)的可重构路由算法.该算法可根据网络中数据包的转向请求,来改变每个节点处禁止和允许的转向,从而使网络具有流量均衡性.但是该算法的容错能力较差,故障节点不允许出现在网络内部.

本文结合 Fu 所提出的故障模型具有牺牲较少无故障节点以及算珠转向模型具有均衡网络流量的特性,提出一种具有流量均衡策略的 NoC 可重构容错路由算法.实验结果表明,本文算法能够降低网络的延时,提高网络的吞吐量.

2 故障模型

在当前的故障模型中^[4-5],Fu 的故障模型的优点在于所生成的故障区牺牲较少的无故障节点,能够较大幅度地降低网络性能的损失.其故障区生成规则如下:

- (1)如果一个节点有 2 个及 2 个以上的邻居节点是故障节点,则该节点也变成故障节点;
- (2)如果一个节点在水平(垂直)方向上的邻居节点是故障节点,并且在垂直(水平)方向上有一个距离为 2 的节点是故障节点,则该节点也是故障节点.

如图 1 所示,根据 Fu 提出的故障模型的生成规则,在 2D-mesh 结构的网络中共有 9 种故障类型(FB-1~FB-9).本文根据 Fu 的故障模型,提出一种能够感知网络中故障区所在位置的故障模型.该故障模型对 Fu 提出的部分故障区(FB-2,FB-5 和 FB-8)进行故障区边界扩展,并将扩展的部分称之为扩展边界(Extended Boundary).当数据包处于扩展边界上时,通过扩展边界上的参考节点(Reference

Node)来感知目的节点与故障区的位置关系,选择一条最优的数据包传输路径.如图 1 所示,扩展规则如下:

- (1)对于 FB-2 故障类型,扩展边界从故障区西南角向下延伸,直到与其他故障节区或者网络的下边界相交,并将相交的节点记作参考节点 M;
- (2)对于 FB-8 故障类型,扩展边界从故障区西北角向上扩展,直到与其他故障节区或者网络的上边界相交,并将相交的节点记作参考节点 m;
- (3)对于 FB-5 故障类型,扩展边界从故障区西南角和东北角分别向下和向上扩展,直到与其他故障节区或者网络边界相交,并将相交的节点分别记作参考节点 M 和 m.

这种故障区的优点在于不会造成额外的无故障节点被牺牲.

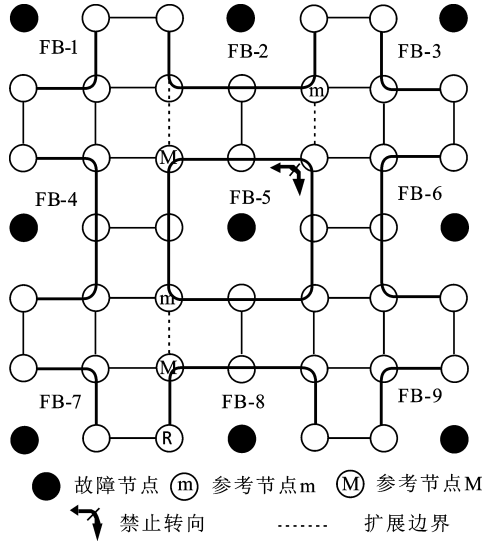


图 1 故障区类型

3 转向模型

转向模型是解决网络死锁的一种重要措施,由文献[3],只要将数据包在路由过程中形成的右边界从网络中移除,则网络就不会发生死锁,而在网络中形成右边界(本文以顺时针右边界为例),需要满足在网络的同一列上,允许由东向南(ES)转向的节点在允许由南向西(SW)转向的节点上方,且两个节点之间有连续的由北向南(NS)的链路.

文献[6]中提出的算珠转向模型是针对网络内部不存在故障节点的情况.其原理是在每列上分别添加一个顺时针“算珠”和一个逆时针“算珠”,通过限定算珠上方节点和下方节点禁止一些转向来避免

右边界的形成,如图2所示,虚线箭头为该节点处禁止的转向,其限定规则如下:

- (1)在顺时针“算珠”的上方数据包禁止 ES 转向,下方禁止 SW 转向;
- (2)在逆时针“算珠”的上方数据包禁止 NW 转向,下方禁止 EN 转向。

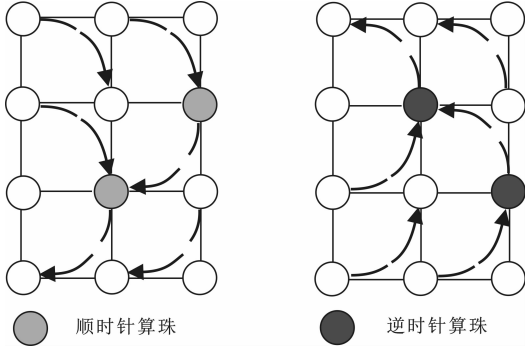


图2 算珠转向模型

这种转向模型的优点在于“算珠”的位置是可调的,使得每个节点可根据网络的通信所需要的转向来调整“算珠”所在位置,从而使网络具有流量均衡的特性。

但是,当网络内存在故障节点时,一个故障节点将同一列上无故障节点分成两个部分。如图3(a)所示,根据右边界的形成条件,两个部分之间不存在连续的链路,因此,即使允许 ES 转向的节点在允许 SW 转向的节点上方,只要保证这两个节点不在故障节点的同一侧,网络也不会形成右边界,所以这两个部分在避免形成右边界问题是相互独立的。如图3(b)所示,如果将算珠转向模型应用的存在故障的网络上时,故障节点上方的算珠必然也会影响故障节点下方节点的转向情况,从而造成不会形成右边界的转向也会被禁掉,从而影响了网络的性能。对此,当网络存在故障节点时,可在故障节点所在列的每一部分中,分别添加两种转向算珠,如图3(c)所示,使得数据包在一个部分中遵循算珠转向模型进行路由,而不受其他部分中算珠所在位置的影响。

然而,每一列上故障节点的个数和位置都是不同的,单纯通过添加“算珠”的方法已不再适用。为此,本文提出一种新颖的方法,该方法在每个节点处添加两个1比特的寄存器<clockwise, counter-clockwise>,当寄存器 clockwise 的值为“1”时代表顺时针算珠所在位置,寄存器 counter-clockwise 的值为“1”时代表逆时针算珠所在位置,这种方法精准

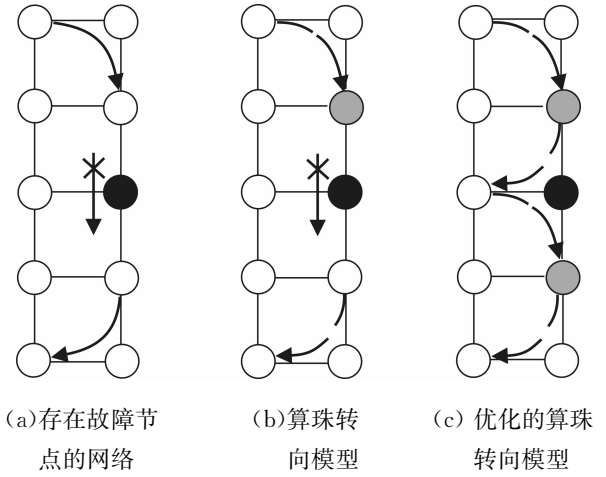


图3 存在故障节点的算珠转向模型

地将算珠转向模型应用的存在故障的网络中。每个节点处的<clockwise, counter-clockwise>的初始化规则如下:

- (1)网络上边界无故障节点处初始化为<0,1>;网络下边界无故障节点处初始化为<1,0>;
- (2)故障区上边界节点处初始化为<1,0>;故障区下边界节点处初始化为<0,1>;
- (3)当网络的上下边界与故障区的上下边界重合时,服从网络边界节点处的初始化规则;
- (4)其余非故障节点处初始化为<0,0>。

图4是根据本文所提出的故障模型及改进后的转向规则,对一个存在故障节点的网络进行初始化后的结构示例。其中黑色节点表示故障节点,白色阴影节点为遵循故障模型生成规则后禁用掉的节点,虚线部分为扩展边界。

网络初始化之后,如图4所示,每一列上从寄存器值为<0,1>的节点到<1,0>的节点之间所有节点为一个独立部分。在每个独立的部分中,为了响应数据包的转向请求,值为1的寄存器所处节点位置的变化规则与文献[6]中算珠变化规则相同,其规则如下所示:(以 clockwise 为例,counter-clockwise 的变化规则与 clockwise 的变化规则一样)

- (1)沿着列方向逐一节点变化,并将原先值为1的 clockwise 置为0,直到转向请求得到响应;
- (2)clockwise 值由1变到0需要禁止一个转向,在值为1的 clockwise 所在节点处,只有在没有数据包路由需要该将要禁止的转向时才能变为0;
- (3)当 clockwise 值为1处的节点及该节点上方、下方都有转向请求信号时,响应优先级首先是 clockwise 值为1所处节点转向请求,其次是该节点

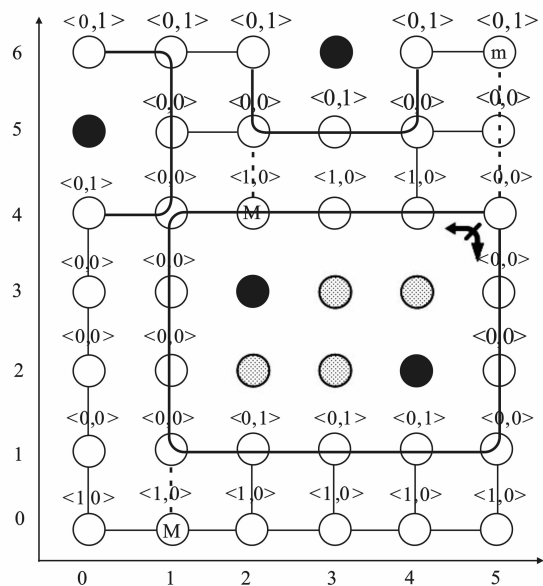


图4 初始化后的网络示例

上方的转向请求,最后是该节点下方的转向请求;

为了防止数据包垂直撞向故障区而导致网络的死锁,对 FB-5 和 FB-6 两种故障区边界节点上的寄存器值变化的限定,在这两种故障区上边界节点处的 clockwise 值固定为 1;下边界节点处的 counter-clockwise 值固定为 1. FB-5 右边界所在列,值为 1 的 clockwise 所在节点不能在 FB-5 故障区东北角节点的上方.扩展边界所在列,值为 1 的 clockwise 所在节点不能在参考节点 M 的上方;值为 1 的 counter-clockwise 所在节点不能在参考节点 m 的下方.

4 可重构容错路由算法

4.1 路由算法

根据所提出的故障区模型,网络中存在安全节点、故障节点、故障区边界节点以及扩展边界节点四种.其中只有故障节点是禁止数据包的接收和发送,其它三种节点处都允许发生数据包的路由.本文采用基于逻辑的分布式路由(Logic-Based Distributed Routing, LBDR)算法^[7],即在每个节点的东、南、西、北四个端口各增加三个比特的配置信息,其中包括两个比特的路由位(routing bits)和一比特的连通位(connectivity bits)信息,其中路由位表示当前节点处允许和禁止的转向,并随每个节点中两个寄存器值的变化而变化;连通位表示与每个输出端口相邻的节点是否为故障节点.根据数据包当前所在节

点的类型执行不同的路由策略,规则如下:

(1)当数据包处于安全节点时,数据包路由遵循 LBDR 算法.首先比较当前节点与目的节点的位置关系,选择能够减小数据包到目的节点距离的输出端口(如:目的节点在当前节点的东北方,所以东输出端口和北输出端口是可以作为数据包的输出端口);然后在所选择的输出端口中检查 LBDR 配置信息是否满足路由条件;最后选择满足条件的输出端口进行路由,当有两个输出端口满足条件时,数据包任意选择其中一个端口进行路由.

(2)当数据包处于故障区边界节点上时,因为存在故障区的原因,数据包依照 LBDR 算法数据包无法安全地到达目的节点,需要其它的路由策略.因为 FB-1 和 FB-7 可以看成是 FB-4 的一个部分,FB-3 和 FB-9 可以看成是 FB-6 的一个部分,而 FB-6 是 FB-4 的水平映射,FB-8 是 FB-2 的垂直映射,故当数据包处于故障区边界时,仅以 FB-2、FB-4 和 FB-5 为例讨论其路由策略,如图 5 所示.

对于 FB-2 故障类型,如图 5(a)所示,如果向东(包括东南方向和东北方向)的数据包被故障区西边界阻断时,那么数据包先路由到故障区西南角节点.当数据包达到西南角节点时,如果目的节点在故障区南边界所在行的下边(包括南边界所在行),恢复 LBDR 算法;否则,数据包路由到故障区东南角节点之后再恢复 LBDR 算法.相似的情况如图 5(b)和 5(c)所示,分别为向西的数据包被故障区东边界阻断,向东和向西的数据包被故障区南边界阻断.对于 FB-4 和 FB-5 故障类型的绕行规则与 FB-2 类似,如图 5(d)~5(j)所示,在此不再讨论.

(3)当数据包处于扩展边界节点上时(数据包位于扩展边界所在列的故障区边界上时,其在非绕行故障区时,路由策略与数据包处于扩展边界节点上的路由策略相同),路由策略如图 6 所示.图 6(a)表示数据包当前节点和目的节点都处于扩展边界所在列上,如 S1 分别向 D1 和 D2 发送数据包,D1 处于参考节点 M 的上方,数据包直接向下路由到 D1;D2 处于参考节点 M 的下方,所以数据包先向西路由,再恢复 LBDR 算法,因为数据包禁止 180°转向,所以数据包只能向下路由.同理对于 S2 向 D3 和 D4 发送数据包,因为 D3 在参考节点 m 的下方,D4 在参考节点 m 的上方,路由路径如图所示.

图 6(b)表示,数据包处于参考节点为 M 的扩

展边界上,目的节点在当前节点的西边(西南方向和西北方向)和东边(包括东南方向和东北方向)的情况.如 S3 分别向西边的 D5 和 D6 发送数据包,数据包先路由到西边相邻节点再恢复 LDBR 算法;当目的节点在当前节点东边时,如果目的节点在参考节点 M 的下方,数据包先路由到西边相邻节点再恢复 LDBR 算法(如 S3 向 D7 发送数据包),否则,数据

包根据 LDBR 算法进行路由.图 6(c)表示,数据包处于参考节点为 m 的扩展边界上,向西的数据包的路由策略与图 6(b)相同,向东的数据包,如果目的节点在参考节点 m 的上方,数据包先路由到西边相邻节点再恢复 LDBR 算法(如 S4 向 D10 发送数据包),否则,数据包根据 LDBR 算法进行路由.

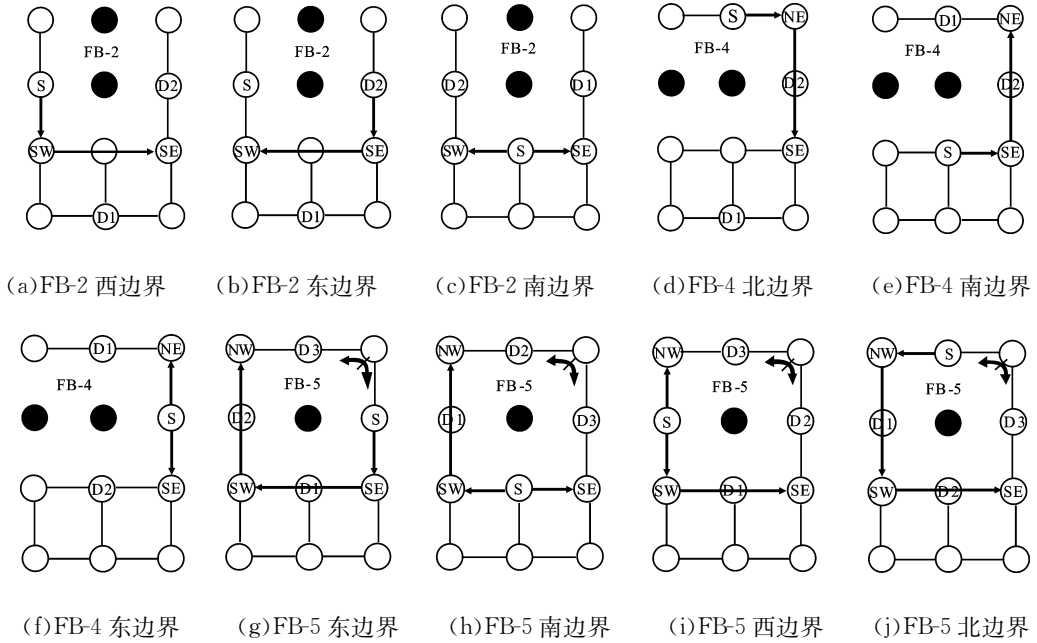


图5 故障区边界上的路由策略

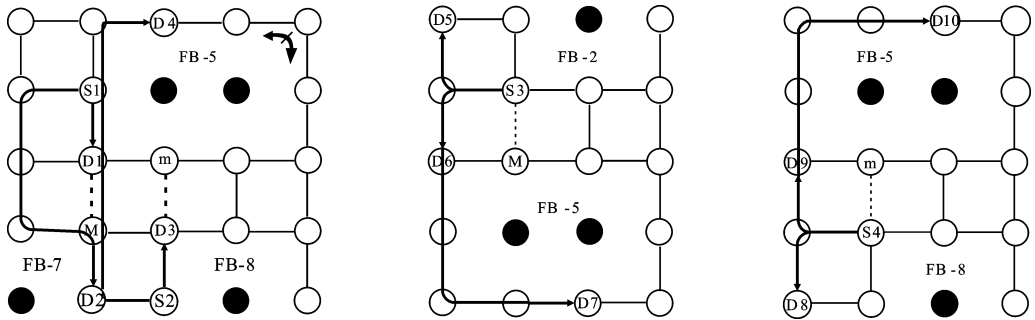


图6 扩展边界上的路由策略

4.2 死锁的避免

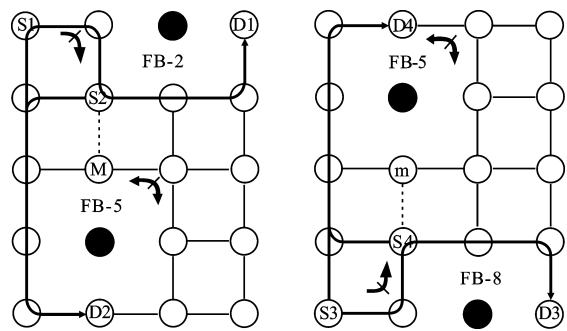
当网络内部没有故障节点时,本文算法与文献[6]是一样的,不会发生死锁的.

当网络中存在故障节点时,依据本文的算法,在安全节点处数据包路由遵循 LDBR 算法,不会产生右边界.当数据包垂直撞向故障区边界时发生敏感转向(所有可能产生右边界的转向,包括 ES 转向、SW 转向、EN 转向和 NW 转向)时,如 ES 转向发生

在 FB-2 和 FB-5 的西边界,EN 转向发生在 FB-8 西边界,本文通过对这两个故障区进行边界的扩展,使得数据包在这两种故障区西边界和扩展边界上能够感知网络中故障的存在,从而在该列上避免 SW 转向和 NW 转向的发生,所以不会形成右边界;SW 转向发生在 FB-5 和 FB-6 的北边界,NW 转向发生在 FB-5 和 FB-6 的南边界,本文通过对这两个故障区北边界和南边界节点处的寄存器状态进行限定,令

北边界节点中 clockwise 寄存器的值始终为 1,南边界节点 counter-clockwise 寄存器的值始终为 1,来分别在这两种故障区北边界上方禁止 ES 转向,南边界下方禁止 EN 转向,以达到禁止右边界形成的目的.

当两种敏感转向出现在同一列上同一部分时,如图 7 所示,在图 7(a)中,FB-2 的西边界会发生 ES 转向,FB-5 的北边界会发生 SW 转向,对于此,通过扩展边界具有感知故障的功能,在该列上禁止了 SW 转向.如 S2 向 D2 发送数据包时,因为 D2 在参考节点 M 的下方,数据包先向西路由再恢复 LDBR 算法,从而避免 SW 转向的发生,并且当 S1 向 D2 发送数据包时,如果 S1 处允许 ES 转向,数据包到达 FB-2 西边界,此时由于 D2 在参考节点 M 的下方缘故,数据包又要向西路由,产生了 180°转向,所以通过禁止参考节点 M 的上方所有节点中 clockwise 值不能为 1 的方法,禁止在该部分中发生 ES 转向.同理在 7(b)中,FB-8 的西边界会发生 EN 转向,FB-5 的南边界会发生 NW 转向,通过禁止在参考节点 m 的下方所有节点中 counter-clockwise 值不能为 1,禁止该部分发生 EN 转向.



(a)ES 转向和 SW 转向 (b)EN 和 NW 转向

图 7 两种敏感转向出现在同一列

当数据包绕行故障区时产生敏感转向,如在 FB-1 的东南角发生 EN 转向,但是在 EN 上方不会发生 NW 转向,所以不会形成右边界,根据数据包在故障区边界的绕行规则,只有在 FB-4 会形成右边界,但是因为 FB-4 位置的特殊性,数据包绕行过程中不会形成环路,所以也是无死锁的;在 FB-5 的东南角会发生 SW 转向和 EN 转向,本文通过扩展边界的绕行策略以及禁止东边界所在列的 clockwise 值为 1 的节点不出现在东北角上方的方法,禁止 ES 转向和 NW 转向出现在东北角上方,达到禁

止右边界形成的目的.

综上所述,当网络中任何位置出现故障节点时,通过本文所提出故障区模型和数据包路由策略,网络将不会发生死锁.

5 实验结果与分析

为了验证本文提出的算法,采用 NIRGAM 仿真器进行算法的仿真验证^[8].这种仿真器具有开放的源代码,可以针对 NoC 的特点进行合理的建模,是一个用于 NoC 网络性能仿真的专用仿真器.本文主要是从平均吞吐量和平均延时两个方面来衡量所提算法对网络性能的影响.

仿真是在 8×8 的 2D-Mesh 网络中进行,流量模型采用 uniform,每个数据包包含 3 个 flit,相邻节点之间数据包传输不使用虚通道,仿真时间持续 4 000cycles,并以 Fu's 算法作为参考对象.故障节点的选取既保证了故障节点分布于网络中各个位置又充分模拟各故障区在网络中的相对位置,确保实验能够包含网络中出现各种故障情况.

图 8 为两种算法的平均吞吐量分别在 5%左右的较低故障节点比例和 20%左右的较高故障节点比例下随通信负载的变化情况曲线图.当通信负载低于 60%时,由于负载较小,两种算法在不同故障率下其吞吐量是相同的,当通信负载大于 60%时,从图中可以看出本文算法相较于 Fu's 算法在吞吐量方面具有明显的优势.这是因为本文算法具有均衡网络流量的特点,当一个数据包有两个可选择输出端口时,数据包随机选取其中一个,从而达到降低网络堵塞的目的.

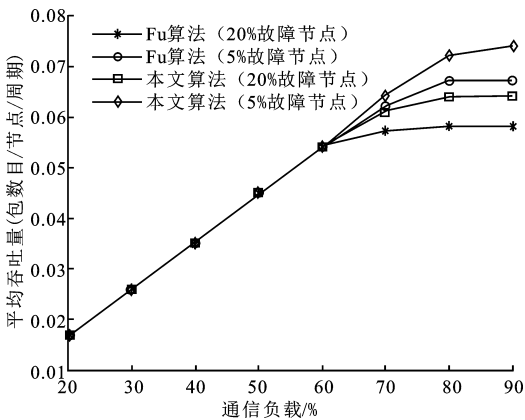


图 8 算法平均吞吐量比较

图 9 为两种算法在不同故障注入率下的平均延

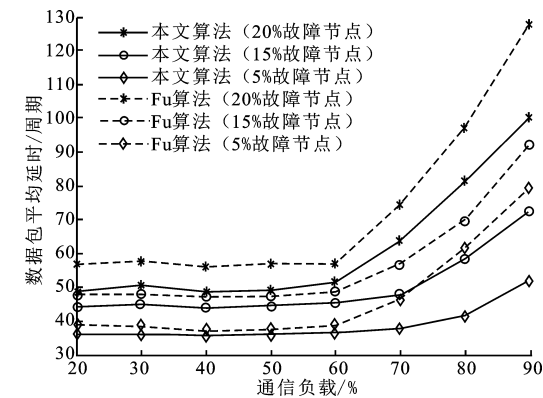


图 9 算法平均延时比较

时对比图。当网络中故障节点的比例为 5% 时,本文算法相比较于 Fu's 算法平均延时能降低 12.5%,在通信负载大于 70% 时优势更加明显。随着网络的故障节点比例的增加,本文算法和 Fu's 算法的数据包平均延时都会上升,但是本文算法具有降低网络拥塞的特点,从而降低了网络的平均延时,其中当故障比例高达 20% 左右时,本文算法相比较于 Fu's 算法平均延时能降低 10.1%。

6 结束语

本文提出一种具有流量均衡策略的可重构容错路由算法,该算法在网络中存在故障的前提下,较大幅度地保持算法的可重构性,并且针对网络中发生故障的位置不同,对数据包执行不同的路由策略,并保证数据包在网络中的传输是无死锁的。实验结果表明,本文采用的算法因为具有均衡流量的特点,故在网络吞吐量以及数据包平均延时方面明显地优于 Fu's 算法。

参考文献:

[1] 李娇,刘鹏,王良华,等. 一种带有均衡策略的无虚通道容错路由算法[J]. 微电子学与计算机, 2016, 33 (11):157-161.

[2] Zong W, Xu Q. Doart: a low-power and low-latency network-on-chip[C]//34th IEEE International Conference on Computer Design, Scottsdale, 2016:352-355.

[3] Chiu G M. The odd-even turn model for adaptive routing[J]. IEEE Transactions on Parallel and Distributed Systems, 2000, 11(7):729-738.

[4] Wu J. A fault-tolerant and deadlock-free routing protocol in 2D meshes based on odd-even turn model[J]. IEEE Transactions on Computers, 2003, 52 (9): 1154-1169.

[5] Fu B, Han Y, Li H, et al. ZoneDefense: a fault-tolerant routing for 2-D meshes without virtual channels [J]. IEEE Transactions on Very Large Scale Integration(VLSI) Systems, 2014, 22(1):113-126.

[6] Fu B, Han Y, Ma J, et al. An abacus turn model for time/space-efficient reconfigurable routing[J]. International Symposium on Computer Architecture, 2011,39 (3):259-270.

[7] Flich J, Rodrigo S, Duato J. An efficient implementation of distributed routing algorithms for noCs [J]. IEEE International Symposium on Networks-On-Chip, Newcastle upon Tyne, 2008, 3(5):460-475.

[8] Lavina J. Nirgam. a simulator for NoC interconnect routing and application modeling [EB/OL]. (2007-05-15). [2017-06-01]. <http://nirgam.ecs.soton.ac.uk/documentation>.

作者简介:

李 娇 女,(1975-),博士,讲师. 研究方向为高性能数字电路设计、集成电路可测试性设计.

崇云锋 男,(1991-),硕士研究生. 研究方向为片上网络容错技术.

冉 峰(通讯作者) 男,(1954-),教授,博士生导师. 研究方向为 SoC 设计、OLED 显示驱动等.

E-mail:ranfeng@shu.edu.cn.