

WiNoC 中交叉开关仲裁及路由算法设计

欧阳一鸣¹, 张鹏¹, 王奇², 安鑫¹, 梁华国²

(1. 合肥工业大学计算机与信息学院, 安徽合肥 230009; 2. 合肥工业大学电子科学与应用物理学院, 安徽合肥 230009)

摘要: WiNoC(Wireless Network-on-Chip)中的无线路由器面临着比传统片上路由器更加严峻的拥塞问题,平衡有线/无线链路负载是当前无线片上网络的研究热点之一. 为此本文提出并设计了一种基于优先级的交叉开关仲裁方案 PbSA(Priority based Switch allocator), 其将优先级更高的无线数据包优先路由至无线路由器; 结合 PbSA 提出了拥塞感知的路由算法 CARA(Congestion-Aware Routing Algorithm), 该算法有效平衡有线/无线链路负载且避免死锁, 提高了数据包在网络中的路由效率. 此外, 本文还提出了新颖的虚通道划分方法, 它不仅减少了实现 PbSA 的硬件复杂度, 而且缓解了无线路由器发生拥塞时对整个网络的影响. 实验表明, 虽然本文的方案引入了较小的面积和功耗开销, 但是具有良好的流量自适应特性, 从而在低注入率或高注入率的情况下较好的提升了网络性能.

关键词: 无线片上网络; 拥塞避免; 交叉开关仲裁; 路由算法; 虚通道

中图分类号: TP302 **文献标识码:** A **文章编号:** 0372-2112(2021)03-0518-09

电子学报 URL: <http://www.ejournal.org.cn> **DOI:** 10.12263/DZXB.20191304

Switch Arbitration and Routing Algorithm Design in WiNoC

OUYANG Yi-ming¹, ZHANG Peng¹, WANG Qi², AN Xin¹, LIANG Hua-guo²

Abstract: Wireless routers in wireless network on chip confronts more severe congestion problem, so balancing wired/wireless loading has become research focus in WiNoC recently. We propose a priority-based switch arbitration scheme, in which data packets more suitable for transmission through wireless channels are routed to wireless routers; We propose a CARA(Congestion-Aware Routing Algorithm) combined with PbSA, which efficiently balance wired/wireless loading and simultaneously avoid deadlock, improving data routing efficiency. Besides, we propose a new virtual channel partition method, which decreases hardware complex for implementing PbSA, and mitigates the adverse effect caused by wireless router congestion. Evaluation shows that the scheme we proposed achieve fine data flow adaptivity with small area and energy overhead, so to improve network performance and routing efficiency under low or high injection rate.

Key words: WiNoC; congestion avoidance; switch arbitration; routing algorithm; virtual channel

1 引言

随着集成电路制造技术的不断发展, 预计在不久的将来单个芯片上可以集成数百甚至数千个处理核心, 传统的总线互联已经不能满足片上通信的需求^[1]. 虽然在 21 世纪初, 西方学者把宏观计算机网络的概念映射到单个芯片上, 提出了将 NoC(Network-on-Chip) 作为片上互联的基础架构来代替传统的总线互联^[2,3], 但是随着网络规模的不断增长, 长距离多跳传输限制了 NoC 的性能表现. 为了解决这一问题, 学者们利用 3D 互联^[4]、光互连^[5] 和射频互联^[6] 来加速传统 NoC 中的

多跳路由, 但是它们都存在一定程度上的局限性^[7].

无线互联是一个最具前景的解决方案, 它在传统 NoC 中的部分路由器上添加了无线收发单元, 使远距离节点对之间能够进行单跳无线传输^[8]. 虽然无线传输具有低延迟、高带宽的特性, 但是由于无线收发器开销大, 功耗高, 所以它并不能完全取代传统 NoC 中的有线传输, 只是作为传统 NoC 的优化方案^[9,10]. 小世界拓扑^[11] 是 WiNoC 的常用拓扑, 它将传统的 MESH 网络分为几个子网. 为叙述方便, 本文将配备有无线收发单元的路由器称为无线路由器. 图 1 是在本文的研究中所使用的拓扑结构. 已有研究表明, 将无线路由器放在每个

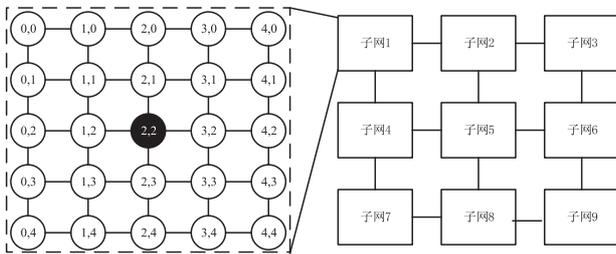


图1 15 × 15 MESH WiNoC架构

区域的中心可以使任何节点与无线路由器之间的通信距离最小化,从而保证了对无线介质的最高效利用^[12]. 图1中,黑色节点为无线路由器,与无线路由器处于同一子网的其余路由器称为有线路由器.网络中,每两个无线路由器之间都采用无线链路进行单跳无线通信^[13].

由于每个子网只拥有一个无线路由器,而子网内所有要使用无线链路的数据包都要经过该无线路由器,这就可能会导致大量数据包阻塞在子网中心的无线路由器处,从而致使片上通信瘫痪^[14,15].文献[16]中提出,若采用XY路由算法,则中心区域的片上路由器将比边缘区域的片上路由器有更大的路由压力,而无线路由器又恰巧在子网的中心位置,这就进一步加剧了无线路由器的拥塞问题.

针对上述问题,文献[14]提出了一种方法,该方法用SA(Simulate Anneal)算法找到无线路由器的最佳放置位置,然后利用 Δ 参数来平衡有线/无线负载.该方法的优点是实现简单,但是由于 Δ 的设置是静态的,所以不能自适应片上不断变化的流量情况,特别是突发流量情况.网络中的拥塞问题一直是NoC研究领域的热点问题,在传统NoC中它的大多数解决方案是引入拥塞传播网络或使用数据包携带拥塞信息,然后路由节点基于已知的拥塞信息来避免向拥塞区域发送数据包,但前一种方法不具有良好的可扩展性,后一种方法获得的拥塞信息不具有实时性^[17-19].

本文的方案旨在在不引入拥塞信息传播开销的前提下,既能充分发挥无线传输的优势,又能避免无线路由器拥塞.本文的主要贡献如下:

(1)提出并设计了一种新的基于优先级的交叉开关仲裁方案PbSA,该方案能够使跳距更长、数据包长度更大的无线数据包更快流入源子网的无线路由器.

(2)结合PbSA提出了CARA路由算法,该路由算法能够有效平衡有线/无线链路负载,能够在避免无线路由器拥塞的前提下,充分发挥无线链路的优势.

(3)结合PbSA提出了新的虚通道划分方法,该方法配合CARA能够有效解决头阻塞问题,提升路由效率.

2 本文的动机

WiNoC的引入主要是为了解决传统NoC中数据包进行长距离多跳有线传输的高延迟问题.但是由于无线链路传输的低并行性,无线路由器易发生拥塞,从而导致有些无线数据包产生了非常高的端到端延迟.这就违背了使用无线链路的初衷.相关文献表明,传输距离越长,数据包长度越大的数据包使用无线链路对整个片上网络的性能提升越明显^[7,14].利用这一特点,本文设计了PbSA,使传输距离更长,数据包长度更大的无线数据包优先通过交叉开关,从而使它能够更快的流入无线节点,使用无线链路.同时为了避免无线路由器发生拥塞且充分发挥无线链路的优势,我们需要对在同一段时间内注入相同子网的数据包进行分流(一部分使用无线链路,另一部分使用纯有线链路).分流的目的是为了在子网内长距离流量稀疏时,使利用无线链路增益较小的无线数据包也能使用无线链路,从而能够充分发挥无线链路的优势;而在子网内长距离流量密集时,又能过滤掉上述的无线数据包,只让利用无线链路增益较大的无线数据包使用无线链路.本文旨在在不引入拥塞传播网络的前提下,解决上述问题.

3 PbSA设计与实现

3.1 为数据包赋予优先级

我们记一个数据包 P 采用混合有线/无线链路传输至目的节点所需要的跳数为 A ,采用纯有线链路传输至目的节点所需要的跳数为 B ,那么 $B-A$ 的值就是该数据包采用无线链路代替长距离多跳有线链路所能减少的跳数,记该值为 C .

本文将任意两个子网之间采用无线链路通信所能节省的最大跳数,作为优先级变化的临界值,若 C 大于该值,则优先级加1,反之则减1.对于图1所示的拓扑结构,它的 C 值-优先级对应关系如表1.

表1 15 × 15 MESH WiNoC对应表

C 值	优先级	C 值	优先级
1	00	10	10
2	00	11	10
3	00	12	10
4	01	13	10
5	01	14	11
6	01	15	11
7	01	16	11
8	01	17	11
9	10	18	11

3.2 数据包格式

传统数据包^[18]的头 flit 中一般有 53bit 左右的 Free Bits(空闲位,空闲位大小与网络规模呈反相关,52bit 是在 32 × 32Mesh 中的空闲位大小^[18]).如图 2 所示,本文的方案在数据包头 flit 的空闲位中额外添加了五个字段,分别是:3bit 的 priority 字段,该字段与 packet_size 字段一起用于数据包竞争交叉开关,并根据 3.1 小结进行赋值;5bit 的 T_MAX 字段,该字段主要作为改变数据包属性的依据,其详细作用在第四节中有具体解释;4bit 的 packet_size 字段,用于标识数据包大小,单位为 flits;4bit 的 src_hub_id 字段,该字段指示源节点所处子网的无线路由器的位置;4bit 的 dst_hub_id 字段,该字段指示目的节点所处子网的无线路由器的位置.

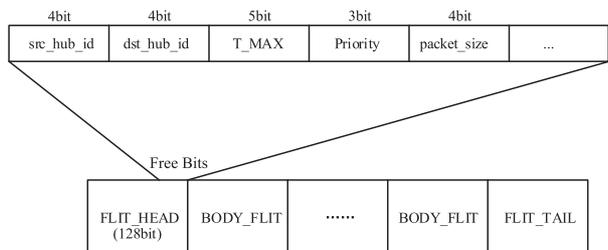


图2 数据包格式

3.3 虚通道划分

如图 3 所示,数据包 P2 为有线数据包,P1 和 P3 为无线数据包.因为无线路由器处发生拥塞,P1 和 P3 都被阻塞在当前路由器,但是由于路由器缓冲区都遵循先进先出的策略,那么即使 P2 之后的路由路径都是畅通的,也会导致 P2 被阻塞在当前路由器.为此,我们借鉴文献[20]的方法,将所有的无线数据包分配在 VC1,所有的有线数据包分配在 VC2.这样做还有利于路由器区分无线数据包和有线数据包,死锁避免,同时会降低本文实现 PbSA 的复杂度.

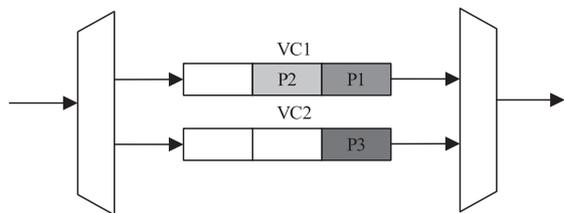


图3 虚通道宏观示意图

VC1 中包含两种类型的无线数据包,分别是处于源子网的无线数据包 S_WiP 和处于目的子网的无线数据包 D_WiP,VC2 中包含两种类型的有线数据包,分别是:(1) C 值小于等于 0 的数据包,它在源节点就被确定为有线数据包;(2) C 值大于 2 但是在源子网的路由过程中由于交叉开关竞争失败次数过多,导致总的等待时间超过 T_

MAX 后,由无线数据包转变为有线数据包.

3.4 Port_priority_Unit 设计

我们定义无线输出端口是采用 XY 路由算法,以当前路由器为源节点,以当前路由器所在子网的无线路由器为目的节点的输出端口.例如对于图 1 中的(0,4)路由器,它的无线输出端口为 E(东,以上北下南左西右东来定义)方向.

S_WiP 的头 flit 到达 VC1 后,它的优先级和数据包大小信息会读入 Port_Priority_Unit 中对应位置,尾 flit 离开缓冲区后该位置又被置 0.因为图 4 中我们假设当前路由器的 W 方向的端口为无线输出端口,所以 W 方向的输入缓冲区的 VC1 中只可能有 D_WiP,因此在图 4 中并没有 Input_W 的信息. information 为各个方向的输入缓冲区的 VC1 中无线数据包的优先级和数据包大小,ACK 信号在本方向输入端口的 Information 值最大时置 1,否则置 0. Empty 是一个 1bit 的寄存器,0 代表当前路由器中有无线数据包在缓冲区中,1 代表当前路由器中没有无线数据包在缓冲区中.

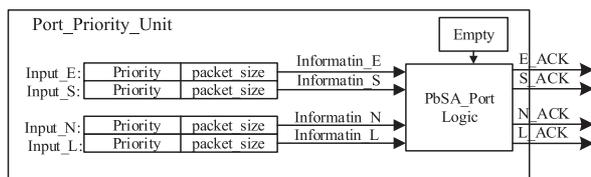


图4 Port_Priority_Unit单元设计

3.5 PbSA 总体设计与实现

图 5 是本文所提出的路由器基础架构,主要由 5 部分组成,分别是输入/输入端口、输入缓冲区、CARA、交叉开关(Crossbar)和 PbSA 组成.

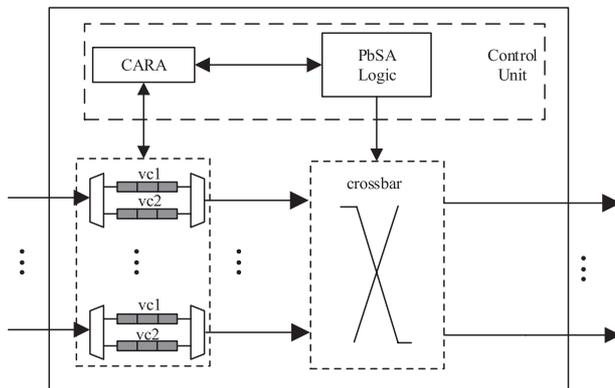


图5 路由器基础架构示意图

图 6 是以当前路由器的无线输出端口(W 方向输出端口)仲裁单元为例,本文所提出的 PbSA 的设计. PbSA 主要包括两个阶段的仲裁,且能在一个时钟周期内完成.第一个阶段是一个 2:1 的选择逻辑,称为 PbSA_VC Logic,它存在于每个输入端口之中,作用是从本端

口的两个虚通道请求信号中选择一个作为本端口的交叉开关请求信号;第二个阶段是从各个输入端口的请求信号中选择一个进行响应,它存在于每个输出端口中,主要由两部分组成,分别是四个 3 输入与门和有线数据包仲裁逻辑。本文为每个路由器的无线输出端口

的仲裁单元设置一个状态机,设置状态机的主要原因是为了防止数据包出现饥饿等待现象。状态机在 normal 状态下,无线输出端口优先分配给无线数据包,在 unnormal 状态下,无线输出端口优先分配给有线数据包。图 7 是状态机的状态转换图。

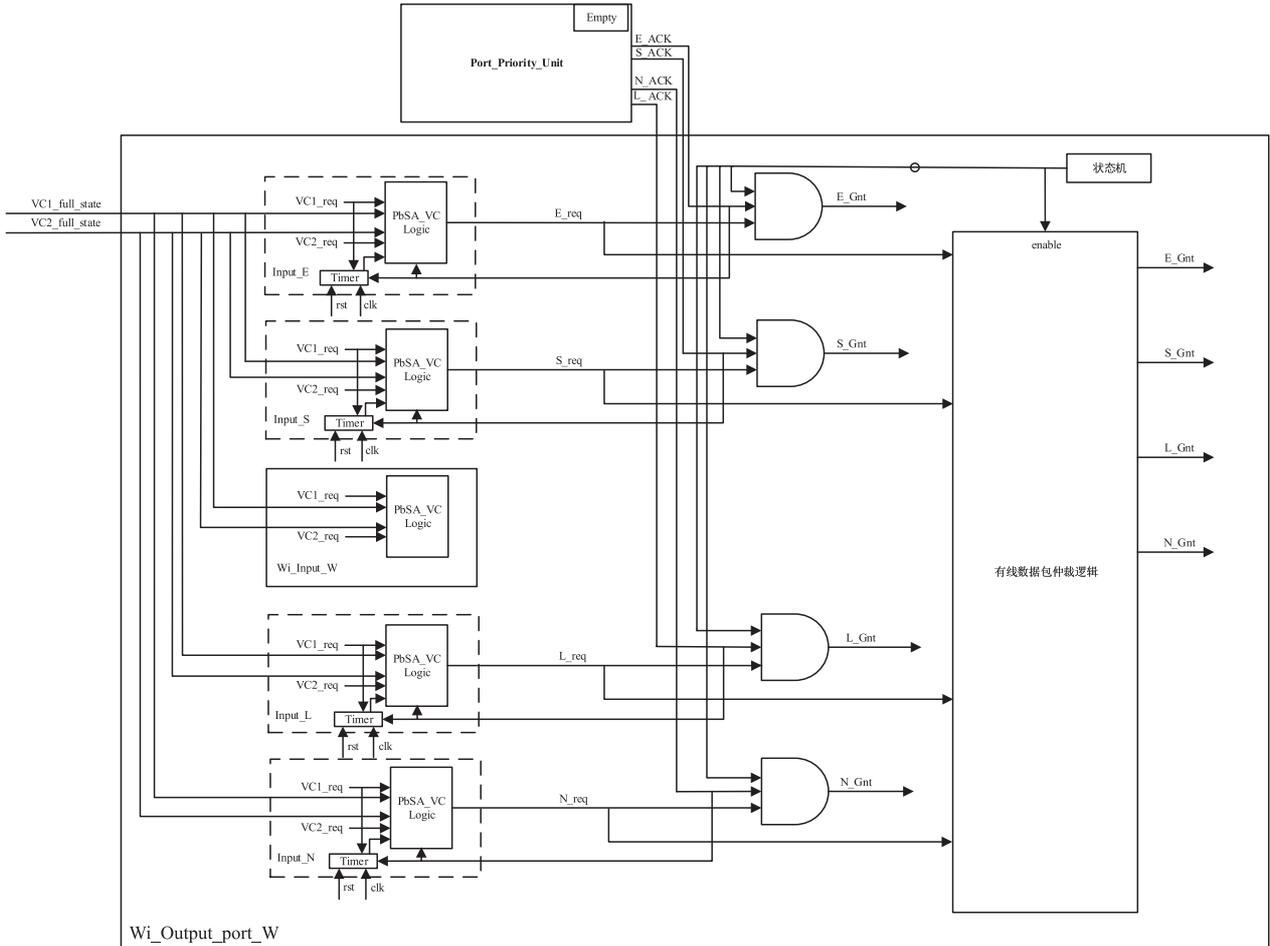


图6 基于输入端口的PbSA设计

算法 1 是 PbSA_VC Logic 的详细逻辑,其中 VC1_req 等于 1 表示当前输入端口的 VC1 有数据包请求交叉开关。VC1_full_state 等于 1 表示下一跳路由器对应缓冲区 VC1 满。Timer 和 T_MAX 的功能在第四节中有详细解释。图 6 中的与门只负责 S_WiP 的交叉开关仲

裁,它在状态机为 Normal 状态时有效。有线数据包仲裁逻辑主要负责有线数据包和 D_WiP 的仲裁。在该逻辑中,采用的是 round-robin 仲裁策略^[21]。图 6 中的所有 Gnt 信号(例如 E_Gnt, S_Gnt 等)在同一时间最多有一个为 1,其余全为 0。例如图 6 中的两个 E_Gnt 信号,只要有一个为 1,就代表当前 W 方向输出端口已授权给 E 方向输入缓冲区。

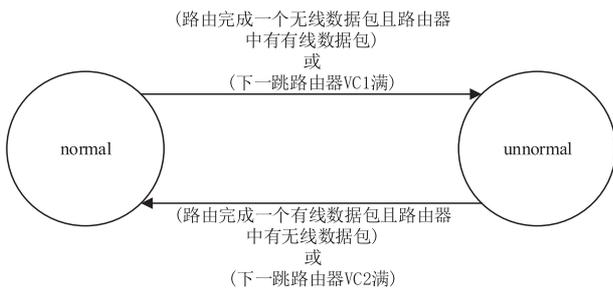


图7 状态转换图

算法 1 PbSA_VC 仲裁算法

Input: VC1_req, VC2_req, ACK, VC1_full_state, VC2_full_state, Timer, T_MAX

Output: req

- 1: if VC1_req = 0 and VC2_req = 1 and VC2_full_state = 0 then
- 2: 当前输入端口选择 VC2 的请求作为它的 req 信号
- 3: end if

```

4:  if VC1_req = = 1 and VC2_req = = 0 and VC1_full_state = = 0 then
5:    当前输入端口选择 VC1 的请求作为它的 req 信号
6:  end if
7:  if VC1_req = = 1 and VC2_req = = 1 then
8:    if Timer < T_MAX then
9:      if ACK = = 1 and VC1_full_state = = 0 and normal then
10:        当前输入端口选择 VC1 的请求作为它的 req 信号
11:      end if
12:      if VC2_full_state = = 0 and unnormal then
13:        当前输入端口选择 VC2 的请求作为它的 req 信号
14:      end if
15:    end if
16:  else
17:    if VC2_full_state = = 0 and unnormal then
18:      当前输入端口依次选择 VC1 和 VC2 请求作为它的 req
        信号
19:    end if
20:  end if
21: end if

```

4 CARA 路由算法

CARA 路由算法的引入主要是为了分流,避免无线路由节点发生拥塞. 本文在源节点路由器为每个无线数据包的头 flit 设置 5bit 的等待时间阈值 T_MAX (以周期为单位). 无线数据包在路由至源子网无线节点的过程中,若它由于和其它无线数据包竞争同一输出端口失败而引起的总的等待时间 T 超过了 T_MAX ,则改变该无线数据包为有线数据包. 这样做的原因是因为在本文 PbSA 的设计下,若一个无线数据包 T 较大,就代表在它之前已经有较多数量的且比该数据包对无线链路的利用收益更大的无线数据包流入了无线节点,那么为了避免无线路由器拥塞,应将该数据包转换为纯有线链路路由至目的节点.

为了使 T 不掺杂与有线数据包进行交叉开关竞争所等待的时间,从而使交叉开关争用较好的表达出当前网络的长距离流量情况. 如图 6 所示,本文为每个有线路由器的每个输入端口设计一个计时器 Timer,它的计时规则如下:(1)从 VC1_req 信号为 1 时开始初始计时;(2)在本端口的 ACK 信号为 1 且 VC1_full_state 为 0 时暂停计时;(3)ACK 信号为 0 或 VC1_full_state 为 1 继续开始计时;(4)在本端口 VC1 中的数据包获得交叉开关后完全终止计时;(5)数据包尾 flit 离开路由器时重置 Timer 为 0. 在整个计时过程中,若计时器超过了无线数据包的 T_MAX 字段,则在当前路由器改变该无线数据包为有线数据包后,经纯有线链路路由至目的节点.

4.1 T_MAX 值的设置

因为数据包使用无线链路的最终目的是为了减少端到端延迟,如果无线数据包在路由到源子网无线节

点的过程中由于交叉开关竞争失败而引起总的等待时间已经超过了它利用无线链路所能节省的时间,就

$$T_MAX = C \times (T_{wired_link} + T_{router}) + T_{wired_link} - \frac{DR_{wired_link}}{DR_{wireless_link}} \times T_{wired_link} \quad (1)$$

应该将该数据包转为纯有线链路路由至目的节点. 所以本文 T_MAX 的设置如式(1). T_{router} 表示无阻塞情况下,一个数据包通过路由器所需要的周期数; T_{wired_link} 表示一个数据包通过有线链路所需的周期数; DR_{wired_link} 表示有线链路的数据速率; $DR_{wireless_link}$ 表示无线链路的数据速率;

式(1)中除了 C 值之外,其余全为常量. 随着数据包 C 值的增加, T_MAX 值也增大,那么该数据包容忍交叉开关竞争失败的次数也就越多,从而能够利用无线链路传输的机会也就越大,这与本文的理念也是相吻合的. T_MAX 在源节点写入无线数据包的头 flit 中,而且如果该数据包在中间路由器产生了由交叉开关竞争失败所引起的等待时间 T_wait ,则该无线数据包的 T_MAX 字段重新赋值为 $T_MAX - T_wait$.

4.2 CARA 详细设计

CARA 逻辑上主要包括两个部分,一个是源节点处的路径选择策略,本文在源节点处中将 C 值大于 2 的无线数据包注入 VC1,使它拥有使用无线链路的潜力;另一个是在每个节点的路由计算单元,该部分的路由方案如下:(1)对于在 VC2 中的数据包,采用 YX 路由算法^[16]并输出至下一跳路由器的 VC2;(2)对于在 VC1 中的数据包 P1, P2. 若 P1 在 L_Wi (本地节点的无线输出端口方向的输入缓冲区)中,采用 YX 路由算法并输出至下一跳路由器的 VC1;若 P2 在 L_Wi 中,则申请无线输出端口并输出至下一跳路由器的 VC1,同时 Timer 开始计时. 当 $Timer > T_MAX$ 后,若 P2 还未路由至下一跳路由器,则对其采用 YX 路由算法并输出至下一跳 VC2.

对于 S_WiP 采用 XY 路由算法, D_WiP 采用 YX 路由算法的原因是即加快了两者的路由速度,减少了它们的相互影响,又防止了大部分路由器的无线输出端口方向的输入缓冲区的 VC1 的浪费. 因为如果对两者都采用 XY 路由算法,那么大部分路由器(例如图 1 中的(0,0), (1,0), (0,1), (1,1)等)的无线输出端口方向输入缓冲区的 VC1 永远是空闲的. 对于有线数据包(处于 VC2 中的数据包),则采用 YX 路由算法,这样做的原因有益于负载平衡^[16].

本文提出的路由算法是无死锁的,首先是有线数据包和无线数据包之间是不会产生死锁的,因为它们使用的是不同虚通道. 然后无线数据包之间也不会产生死锁,原因如下:(1)所有的 D_WiP 都使用 YX 路由算法,所以在它们之间不会产生死锁;(2)所有的 S_WiP 都使用 XY 路由算法,所以在它们之间也不会产生

死锁;(3)由于 D_WiP 和 S_WiP 不会竞争相同方向的输出端口,所以 D_WiP 和 S_WiP 之间没有依赖关系,它们之间也就不会产生死锁;

5 实验评估

5.1 仿真环境配置

本文在 Noxim^[21] 模拟器上进行扩展来评估所提出的 CASA(CARA + PbSA) 的性能. 本文在 15 × 15MESH 网络中,将 CASA 与基准 NoC 和文献[14]所提出的方案进行对比. 具体的仿真环境配置如表 2. 路由器的每个输入端口的缓冲区分为两个逻辑大小为 4 个 flits 的虚通道. 本文采用四种流量模式进行仿真,分别是 Transpose1, Transpose2, Random, Ulocal. 对于 Transpose1 流量模式,假设数据包源节点为 (x, y),那么它的目的节点为 (mesh_x-1-y, mesh_y-1-x). 这种流量模式的特点就是越靠近中间区域的路由节点注入网络的数据包的跳数就越小,但是从整体上来看,长距离多跳数据包占的比重更大. Transpose2 则是标准的转置流量模式,即 (x, y) 发送至 (y, x). Ulocal 是长距离流量模式,在这种流量模式下,长距离数据包的比重较大. 每次仿真包括 1000 个周期的预热阶段和 20000 个周期的仿真阶段. 为了减少实验结果的随机性,我们将每个实验参数都做 50 次仿真求平均值.

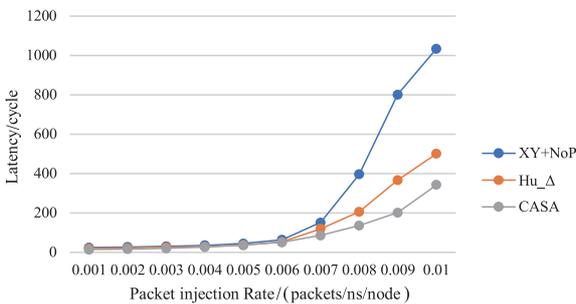
5.2 性能分析

图 8 是在上述四种流量模式下进行仿真所得到的延迟对比图. 四幅图中的横坐标是网络的注入率,纵坐标是以周期为单位的数据包平均延迟. CASA(CARA +

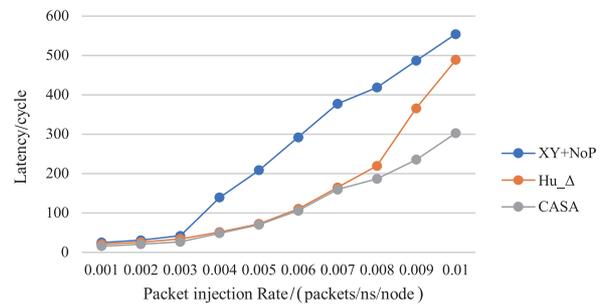
PbSA) 是本文所提出的方案,它的对比方案有两个,分别是 XY + NoP(即使用 XY 路由算法和两跳感知的拥塞处理策略)和 Hu_Δ(参考文献[14]提出的方案). 从图 8 中可以看出,在不同的流量模式下 WiNoC 较传统 NoC 都有一定的延迟改善,而且注入率越高,长距离流量比例越大,这种改善就越明显. Hu_Δ 与本文所提出的 CASA 方案都是在传统 NoC 中引入了无线链路. 从图 8 中可以看出,在不同流量模式下 CASA 较 Hu_Δ 在高注入率或低注入率下都有一定的延迟改善,尤其在 Transpose2 流量下可以明显发现在注入率大于 0.007 与注入率小于 0.005 时,CASA 与对比方案相比有较大的延迟改善. 这是因为 CASA 具有良好的流量自适应特性,即能够随着当前子网内长距离数据包流量的变化来自适应的选择合适的无线数据包使用无线链路. 而且在本文所提出的虚通道划分方法下,即使无线路由器发生拥塞,也不会致使网络瘫痪.

表 2 实验基本参数设置表

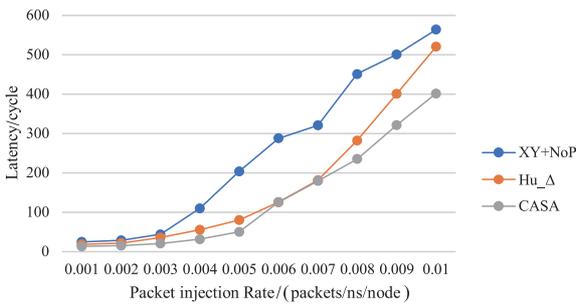
网络拓扑	15 * 15MESH(如图 1)
缓冲区大小/端口	8 个 flits
虚通道数量/端口	2
交换技术	虫孔路由
流量模式	Transpose1, Transpose2, Random, Ulocal
无线 MAC 机制	Token-packet
数据包大小	8 ~ 12 个 flits
路由算法	CARA
仿真时间	21000 cycles



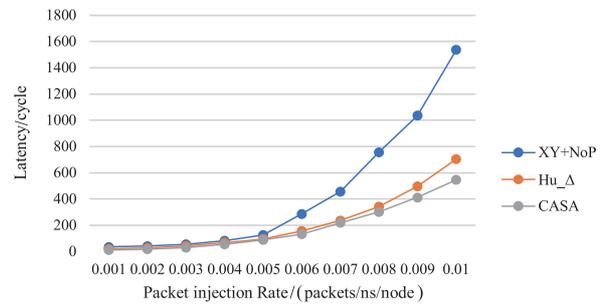
(a) Random Traffic



(b) Transpose1 Traffic



(c) Transpose2 Traffic



(d) Ulocal Traffic

图8 延迟对比图

饱和注入率也是评价网络性能的重要指标,饱和注入率越大说明网络对每个数据包规划的路由路径越合理.图9显示了在四种流量模式下,CASA和它的两种对比方案的吞吐量对比图.四幅图中的横坐标表示的是网络注入率,纵坐标是网络中平均每个节点的吞吐量.从图9中可以看出,在Random Traffic流量模式下,XY+NoP方案的饱和注入率为0.008,Hu_Δ方案的饱和注入率为0.01,CASA方案的饱和注入率为0.013;在Transpose1 Traffic流量模式下,XY+NoP方案的饱和注入率为0.004,Hu_Δ方案的饱和注入率为0.007,CASA方案的饱和注入率为0.009;在Transpose2 Traffic流量模式下,XY+NoP方案的饱和注入率为0.004,Hu_Δ方案的饱和注入率为0.006,CASA方案的饱和注入率为0.01;在Ulocal Traffic流量模式下,XY+NoP方案的饱和注入率为0.005,Hu_Δ方案的饱和注

入率为0.007,CASA方案的饱和注入率为0.009.由此可以看出本文的CASA方案在不同的流量模式下都有较好的饱和吞吐量表现,这依旧要归功于CASA算法的流量自适应特性.

为了用数据体现出CASA方案的流量自适应特性,本文在不同注入率和不同流量模式下,将整个仿真过程中使用过无线链路的无线数据包的C值进行统计得到图10.图中的横坐标是数据包注入率,纵坐标是整个仿真过程中使用过无线链路的无线数据包的C值的平均值.从图中可以看出,在同一流量模式不同注入率时,随着注入率的增加,C的平均值也在增加.而对于不同流量模式来说,长距离流量占比更大的流量模式,C的平均值就更大.这说明了CASA方案能够随着注入率或者子网内长距离流量的变化,自适应的选择合适的无线数据包使用无线链路.

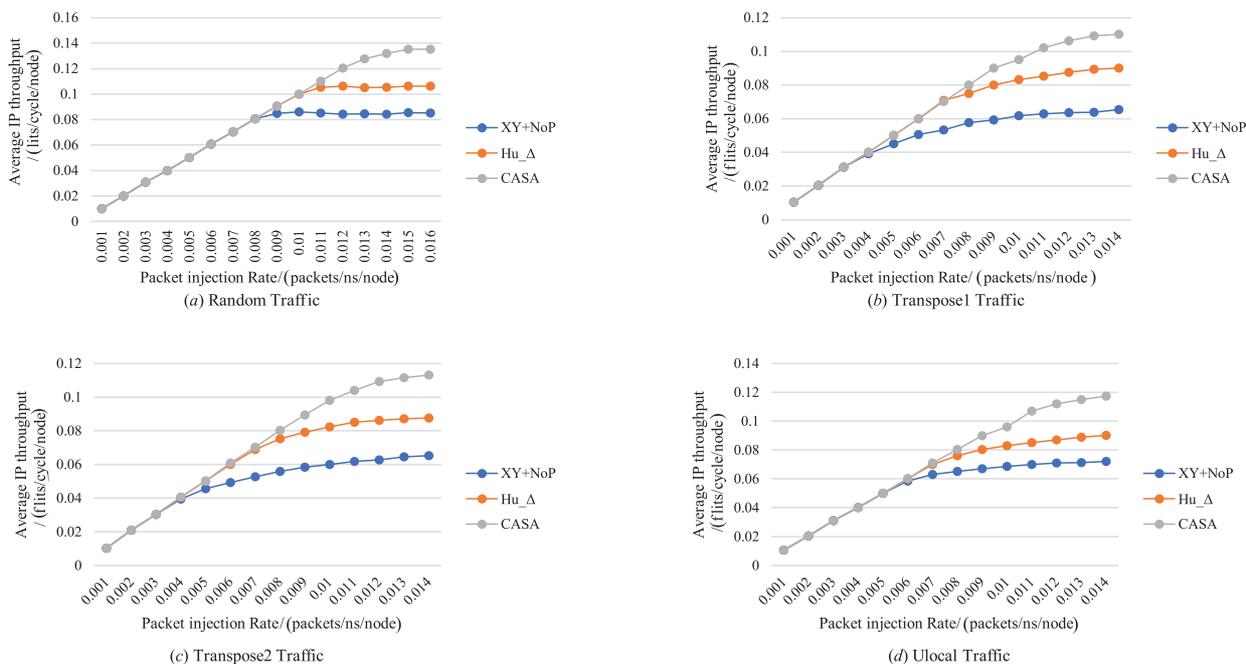


图9 吞吐量对比图

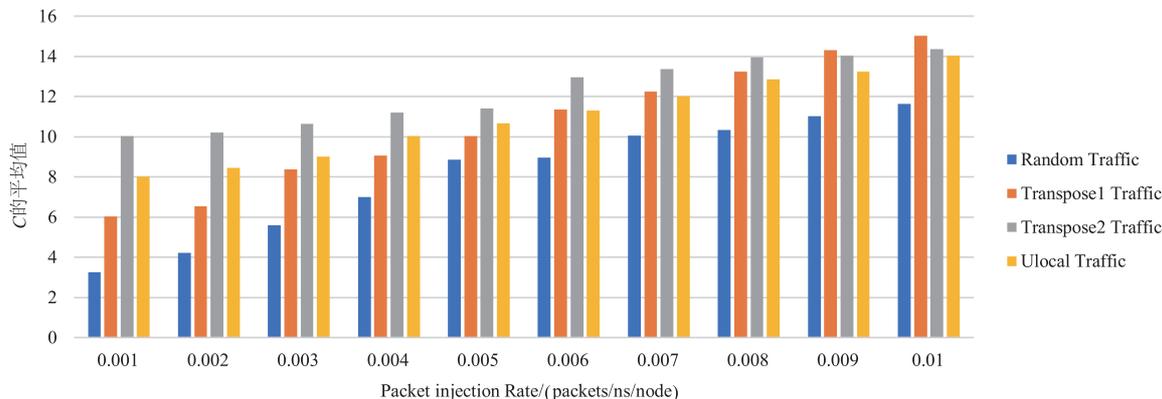


图10 C值分析图

5.3 功耗及面积分析

本文在传统 NoC 的基础上, 添加了无线链路及若干组件, 片上采用 Zig-Zag 天线和毫米波收发器, 面积开销分别约为 0.19mm^2 和 0.16mm^2 ^[13,22]. 本文在 DC 下对 Port_Priority_Unit 和 3.5 小结的仲裁单元做了 RTL 级仿真. 本文使用 Verilog HDL 实现上述组件后, 在 ModelSim 上进行功能仿真. 仿真完成后在 TSMC 65 nm 工艺下对编写好的 Verilog 代码进行了功耗与面积分析. Port_Priority_Unit 的占用面积为 74.88um^2 , 总功耗为 0.041962mw . 仲裁单元(除有线数据包仲裁逻辑)的占用面积为 351.6um^2 , 总功耗为 0.7012mw .

6 总结

WiNoC 中的无线路由器面临着严重的拥塞问题, 而拥塞会导致整个网络性能的急剧下降. 本文旨在在不引入拥塞信息传播开销的前提下, 每个子网能够根据当前子网内长距离流量和注入率情况, 自适应的选择合适的无线数据包使用无线链路. 在本文的 CASA (CARA + PbSA) 及虚通道划分的设计下, 不仅加快了无线数据包的端到端路由效率, 较好的平衡了有线/无线链路负载, 而且即使无线路由器发生了拥塞, 也不会致使网络瘫痪, 从而缓解了对整个网络的性能影响. 实验表明, 本文的方案虽然引入了较低的面积和功耗开销, 但是它不仅提高了网络吞吐量, 减少了数据包平均传输延迟, 而且具有良好的可扩展特性.

参考文献

- [1] Borkar S. Thousand core chips: a technology perspective [A]. Proceedings of the 44th annual Design Automation Conference [C]. San Diego: ACM, 2007. 746 – 749.
- [2] Dally W J. Route packets, not wires: on-chip interconnection networks [A]. Proceedings of the 38th Annual Design Automation Conference [C]. Las Vegas: IEEE, 2001. 684 – 689.
- [3] Benini L, De Micheli G. Networks on chips: A new SoC paradigm [J]. Computer, 2002, 35(1): 70 – 78.
- [4] Chao C H, Jheng K Y, Wang H Y, et al. Traffic and thermal-aware run-time thermal management scheme for 3D NoC systems [A]. 2010 Fourth ACM/IEEE International Symposium on Networks-on-Chip [C]. Grenoble: IEEE, 2010. 223 – 230.
- [5] Joshi A, Batten C, Kwon Y J, et al. Silicon-photonics networks for global on-chip communication [A]. 2009 3rd ACM/IEEE International Symposium on Networks-on-Chip [C]. La Jolla: IEEE, 2009. 124 – 133.
- [6] Chang M F, Cong J, Kaplan A, et al. CMP network-on-chip overlaid with multi-band RF-interconnect [A]. International Symposium on High Performance Computer Architecture [C]. Salt Lake City: IEEE, 2008. 191 – 202.
- [7] Mansoor N, Ganguly A. Reconfigurable wireless network-on-chip with a dynamic medium access mechanism [A]. Proceedings of the 9th International Symposium on Networks-on-Chip [C]. Vancouver: ACM, 2015. 1 – 8.
- [8] Deb S, Ganguly A, Pande P P, et al. Wireless NoC as interconnection backbone for multicore chips: Promises and challenges [J]. IEEE Journal on Emerging and Selected Topics in Circuits and Systems, 2012, 2(2): 228 – 239.
- [9] Rayess W, Matolak D W, Kaya S, et al. Antennas and channel characteristics for wireless networks on chips [J]. Wireless Personal Communications, 2017, 95(4): 5039 – 5056.
- [10] Mamaghani S M, Jamali M A J. A load-balanced congestion-aware routing algorithm based on time interval in wireless network-on-chip [J]. Journal of Ambient Intelligence and Humanized Computing, 2019, 10(7): 2869 – 2882.
- [11] Ganguly A, Chang K, Deb S, et al. Scalable hybrid wireless network-on-chip architectures for multicore systems [J]. IEEE Transactions on Computers, 2010, 60(10): 1485 – 1502.
- [12] Kim R G, Choi W, Liu G, et al. Wireless NoC for VFI-enabled multicore chip design: performance evaluation and design trade-offs [J]. IEEE Transactions on Computers, 2016, 65(4): 1323 – 1336.
- [13] Deb S, Chang K, Ganguly A, et al. Design of an efficient NoC architecture using millimeter-wave wireless links [A]. Thirteenth International Symposium on Quality Electronic Design [C]. Santa Clara: IEEE, 2012. 165 – 172.
- [14] Hu W H, Wang C, Bagherzadeh N. Design and analysis of a mesh-based wireless network-on-chip [J]. Journal of Supercomputing, 2015, 71(8): 2830 – 2846.
- [15] Rezaei A, Daneshtalab M, Palesi M, et al. Efficient congestion-aware scheme for wireless on-chip networks [A]. Parallel, Distributed and Network-Based Processing [C]. Heraklion: IEEE, 2016. 742 – 749.
- [16] Rohbani N, Shirmohammadi Z, Zare M, et al. LAXY: A location-based aging-resilient Xy-Yx routing algorithm for network on chip [J]. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 2017, 36(10): 1725 – 1738.
- [17] 黎建华, 吴宁, 胡永良, 等. 一种自适应的混合型无线 NoC 拓扑结构 [J]. 电子学报, 2016, 44(6): 1420 – 1428. LI Jian-hua, WU Ning, HU Yong-liang, et al. An adaptive hybrid wireless NoC topology structure [J]. Acta Electronica Sinica, 2016, 44(6): 1420 – 1428. (in Chinese)
- [18] Liu S, Chen T, Li L, et al. FreeRider: Non-local adaptive

- network-on-chip routing with packet-carried propagation of congestion information[J]. IEEE Transactions on Parallel and Distributed Systems, 2014, 26(8): 2272 – 2285.
- [19] 欧阳一鸣, 何鑫城, 梁华国, 等. 针对路径故障与局部拥塞的 NoC 容错路由算法[J]. 电子学报, 2016, 44(4): 920 – 925.
OUYANG Yi-ming, HE Xin-cheng, LIANG Hua-guo, et al. A fault tolerant routing algorithm aiming at a path fault and local congestion in NoC[J]. Acta Electronica Sinica, 2016, 44(4): 920 – 925. (in Chinese)
- [20] 欧阳一鸣, 陈静雯, 梁华国, 等. NoC 中负载均衡的 AVOQ 路由器设计[J]. 电子测量与仪器学报, 2017, 31(1): 92 – 98.
OUYANG Yi-ming, CHEN Jing-wen, LIANG Hua-guo, et al. Design of load-balancing AVOQ router in network-on-chip[J]. Journal of Electronic Measurement and Instrumentation, 2017, 31(1): 92 – 98. (in Chinese)
- [21] Catania V, Mineo A, Monteleone S, et al. Cycle-accurate network on chip simulation with noxim[J]. ACM Transactions on Modeling and Computer Simulation, 2016, 27(1): 1 – 25.
- [22] 欧阳一鸣, 钱昌, 杜高明, 黄正峰, 李建华, 梁华国. WiNoC 中 EF-ACK 容错无线接口设计[J]. 电子学报, 2018, 46(11): 2561 – 2570.
OUYANG Yi-ming, QIAN Chang, DU Gao-ming, HUANG Zheng-feng, LI Jian-hua, LIANG Hua-guo, et al. Design of an EF-ACK fault-tolerate wireless interface in WiNoC[J]. Acta Electronica Sinica, 2018, 46(11): 2561 – 2570. (in Chinese)

作者简介



欧阳一鸣 男, 1963 年生, 博士, 教授, 中国计算机学会高级会员, 研究方向: 片上网络 (NoC) 与片上系统 (SoC), 嵌入式系统的综合与测试, 数字系统设计自动化.
E-mail: oyymbox@163.com



张鹏 男, 1997 年生, 硕士研究生, 研究方向: 片上网络 (NoC) 与片上系统 (SoC), 无线片上网络.
E-mail: 917292287@qq.com