

片上网络中面向报文有序传输的 自适应路由算法研究*

姜燕¹, 张广艳²

(1. 湖北医药学院 信息管理与信息系统系, 湖北 十堰 442000; 2. 清华大学 计算机科学与技术系, 北京 100084)

摘要: 与确定性路由算法相比, 自适应路由算法可以提高片上网络的通信性能, 但是报文可能会无序到达。在目的节点对报文排序将会导致严重的面积和计算开销, 甚至可能会抵消采用自适应路由算法带来的性能增益。为此, 首先提出一种部分自适应路由算法, 以满足报文的有序到达; 然后描述了对该算法提供支持的路由器硬件结构; 最后在二维片上网络下对该算法及确定性和自适应路由算法进行了性能评估和比较。与 XY 算法相比, 该算法显著降低了报文延时, 提升了饱和点。同时讨论了对路由器面积和功耗影响。虽然路由器的功耗有所上升, 但是由于报文交付性能提升, 所以每个 flit 的能耗增长可忽略不计。

关键词: 片上网络; 自适应路由; 报文; 硬件结构; 延时; 饱和点

中图分类号: TP393 **文献标志码:** A **文章编号:** 1001-3695(2016)04-1240-05

doi: 10.3969/j.issn.1001-3695.2016.04.062

Research on adaptive routing algorithm for orderly transfer of packet in network on chip

Jiang Yan¹, Zhang Guangyan²

(1. Dept. of Information Management & Information System, Hubei Medical College, Shiyan Hubei 442000, China; 2. Dept. of Computer Science & Technology, Tsinghua University, Beijing 100084, China)

Abstract: Although adaptive routing algorithms promise higher communication performance of network on chip, as compared to deterministic routing algorithms, they suffer from the out-of-order packet delivery problem. In the context of network on chip, the area and computational overhead of ordering packets at the destination is high and may reverse any gain achieved through the use of adaptively of the routing algorithm. To solve this problem, this paper firstly proposed a partially adaptive routing scheme which ensured in-order delivery of the packets in a message. Then, it described the hardware architecture of a router that supports the proposed scheme. Finally, it evaluated and compared the performance of this scheme with both deterministic as well as adaptive routing algorithms for network on chip. As compared to the XY routing algorithm, this technique significantly reduced the packet delay and improved the saturation point. It also discussed the impact on router area and power dissipation. Although the power consumption of routers increase, the increase in energy consumption per flit is negligible as a result of the improved packet delivery performance.

Key words: network on chip(NOC); adaptive routing; packet; hardware architecture; delay; saturation point

0 引言

片上网络(network on chip, NOC)^[1,2]的总体性能取决于多个网络因素, 如拓扑结构、路由算法、流量控制及交换技术等^[3]。其中, 路由算法可分为确定型算法、遗忘型算法及自适应算法^[4]。一般来讲, 自适应路由算法的通信性能(报文延时和吞吐量)强于确定型路由算法, 当网络负载较高时尤其如此^[5]。在自适应路由算法中, 一条消息的多个报文可能没有按照次序到达目的节点。对已知的一对源节点和目的节点, 鉴于路径长度和路径拥塞程度不同, 报文到达目的节点时次序可能发生混乱。在部分应用场景中, 要求报文被接收时的次序与

在源节点发送时相同。例如, 在分布式共享内存系统(DSM)中, 通信系统中的消息在传递时次序发生混乱会给缓存一致性协议的功能造成影响, 因为该协议发挥作用的前提是假设报文有序到达^[6]。如果报文到达目的节点时次序混乱, 则必须在目的节点处利用缓存对其重新排序。报文重新排序将会造成缓存区和通信时间开销。这一开销可能会使高性能通信效益完全丢失甚至作废^[7]。

还有一个问题就是需要选择合适的缓存尺寸。除非空间容量无限, 否则用于重新排序的缓存可能被填满导致无法再接收其他报文。这将导致报文丢失, 于是需要端到端 ACK/NACK 协议进行恢复。然而这些协议在网络资源利用和拥塞方面的开销非常严重^[8]。

收稿日期: 2014-11-24; **修回日期:** 2015-01-13 **基金项目:** 国家自然科学基金资助项目(61272055/F020803); 湖北省教育厅科学技术研究项目(B2013119)

作者简介: 姜燕(1977-), 女, 硕士, 副教授, 主要研究方向为片上网络、大数据处理(2725710435@qq.com); 张广艳(1976-), 男, 副教授, 博士, 主要研究方向为大数据处理、网络存储、分布式计算。

文献[9]对自适应路由算法中报文的有序传输进行了研究。其基本思路是只允许每对源节点和目的节点间存在非重叠路径,并要求无序报文在目标路由器的输入端口中等待。目的路由器必须在将报文传输给内核之前,从不同的输入端口缓存中按次序读取报文。然而该方法可能导致死锁。此外,如果对该方法进行改进以解决死锁问题,则会严重影响底层路由功能的自适应性,因为所有相交的路由路径必须被切断。另外,文献[10]中的方法试图解决动态虚拟信道(VC)分配引发的排序问题。在该算法中,由于同一数据流中的报文可以不受约束地请求下跳路由器中的任意虚拟信道,所以可能影响报文有序传输的性能,即使对确定型路由算法也是如此。目前还没有一种实用算法可有效处理片上网络的报文传输次序混乱问题。为此,本文提出了一种部分自适应路由算法,可满足消息的报文按序到达。本文利用合成流量及基于真实应用的一组流量场景来评估本文算法的性能。仿真实验结果表明了本文方法的有效性。

1 本文算法

1.1 基本思路

一般而言,当源节点突发性地将一大组相关报文发往目的节点时便会出现报文传输次序混乱问题。本文用“消息”表示源节点向单个目的节点发送的一组报文。当采用自适应路由算法时,属于同一消息的报文到达目的节点的次序可能与发送次序不同。这是因为报文序列中传输较早的报文可能通过拥塞严重的路径进行传输,于是到达目的节点的时间晚于发送时间在其之后的报文。

本文方法的基本思路是保证同一消息的不同报文依次交付,同时允许同一数据源和目的节点间的不同消息可以沿着不同的路径传输。通过要求同一消息的不同报文采用同一路径传输即可保证依次交付。此外,同一数据源和目的节点间的不同消息会造成阻塞。也就是说,假设同一数据源和目的节点间有两个消息 M_i 、 M_{i+1} ,于是,在接收到上一报文 M_i 的确认消息前 M_{i+1} 不得开始传输,如图1所示。图1中, M_i 、 M_{i+1} 是具有相同源节点和目的节点的两个连续报文。当源节点接收到目的节点关于上一报文 M_i 的确认消息后,才能开始传输 M_{i+1} 。这可保证不同路由路径上的消息报文按序交付。在上述策略中,消息内路由算法不属于确定型算法。即同一消息中的所有报文沿着相同路径传输但是路由路径由首个报文自适应确定。

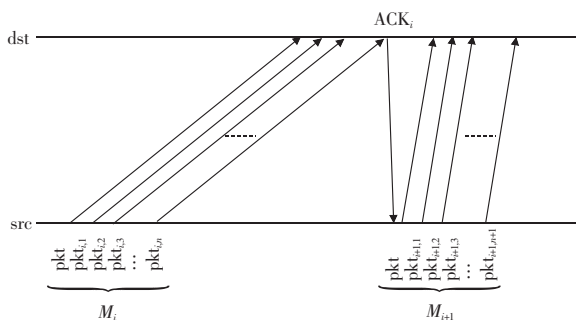


图1 报文阻塞

1.2 算法内容

为了使同一消息中的所有报文采用同一路径,本文使用如下策略。报文的首部 flit 增加一个 ID(msg_id)来唯一标志报

文所隶属的消息。同时向首部增加另一区段(seq_no),以记录消息中报文的位置。图2给出了本文算法使用的首部 flit 包含的部分区段。从图中可见,msg_id 包含源节点和目的节点 ID,并带有消息计数器。这可保证两个不同消息(由不同的源节点生成或发往不同的目的节点)的 msg_id 不同。

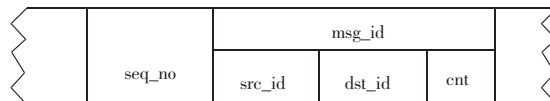


图2 首部 flit 格式

每个路由器配有一张表,称为有序表(IOT)。IOT 表用于维护通过该路由器的各个消息的路径。可使用一种简单的哈希函数,利用消息的 ID 来计算该消息的入口地址。IOT 存储了正在传输该消息报文的被选输出路径。路由算法的伪代码如下所示。IOT 表是有限的,因此可维护有限数量消息的路径信息。

```

1  OutputPort RoutingAlgorithm( HeaderFlit h)
2  {
3      OutputPort output;
4
5      if( h.seq_no == 1 ) //消息的首个报文
6      {
7          OutputPort outputs[ ] = RoutingFunction( h.dst );
8          if( ! IOT.full() )
9          {
10             output = SelectionFunction( outputs );
11             IOT.insert( h.msg_id, output );
12         }
13         else
14             output = outputs[ F( h.msg_id ) ];
15     }
16     else
17         if( IOT.Search( h.msg_id ) )
18             output = IOT[ h.msg_id ];
19         else
20         {
21             OutputPort outputs[ ] = RoutingFunction( h.dst );
22             output = outputs[ F( h.msg_id ) ];
23         }
24
25     return output;
26 }

```

算法以入向报文的首部 flit 为输入,以报文传输时利用的输出端口为输出。如果报文是消息的首个报文(第5行),则利用报文的目的节点(h.dst)为参数触发路由函数。RoutingFunction 函数返回一组可用于到达目的节点的输出端口(存储于向量输出中)。如果 IOT 表未满足(第8行),则利用拥塞感知选择函数^[11]从路由函数返回的可用输出端口集合中选择一个端口(第10行)。

然后利用将报文 ID 与被选输出端口关联起来的新元素来更新 IOT 表,并用 IOT 表传输同一消息中的所有后续报文(第11行)。如果 IOT 表满(第13行),则利用函数 F 确定性地选择输出(第14行),这一选择只与消息 ID 有关。也就是说,函数 F 对同一 msg_id 总是返回同一输出。对消息的非首个报文处理如下:如果后续报文隶属的消息已经在 IOT 表中登记(第17行),则从 IOT 表中提取将要使用的输出端口(第18行);否则,调用路由函数(第21行),根据消息 ID 从可用输出端口集合中确定性地选择一个输出端口(第22行)。

生成的路由算法可以看做是确定型路由算法和自适应路

由算法的综合体。实际上,对这些消息来说,路径选择完全取决于消息 ID。对于可以在 ITO 中分配到元素的消息来说,这是自适应算法。此时,虽然作出路由决策时利用了处理消息首个报文时获得的网络状态信息,仍使用了自适应路由算法。当然,本文算法的有效性取决于 IOT 的尺寸。另一方面,IOT 尺寸会影响路由器成本(硅面积和功耗)。下文将会进行性能和硬件分析。

2 对路由器架构的影响

第 1 章讨论的算法必须集成到路由器。本章描述算法部署情况,并评估延时、面积和功耗方面的性能,在评估时以部署了 XY 路由算法的路由器作为比较对象。

2.1 算法的硬件部署

图 3 对本文路由算法的可能硬件部署进行了详细描述。模块的主要输入用后续报文的首部 flit 表示。

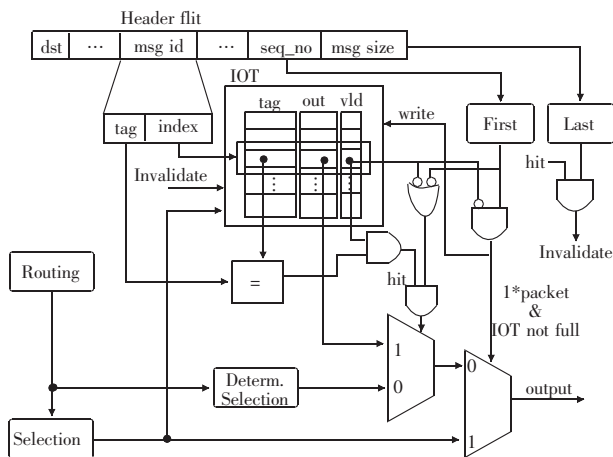


图 3 本文路由算法部署模块框图

图 3 中只给出了首部 flit 中被路由逻辑使用的相关区段。Dst 区段表示目的节点的地址, msg id 表示当前消息的标志符, seq. no 表示该消息的报文序号。模块输出为报文必须发往的输出端口。Msg id 用于明确 IOT 的地址。准确地说, msg id 分为标记和索引。索引部分用于选择 IOT 的元素, 该元素存储了同一消息中所有报文必须采用的输出端口。

只对消息的首个和末尾报文更新 IOT 表(写入), 对消息的其余报文只能以读取模式访问 IOT 表。本文首先分析消息首个报文的数据流。First 模块检查入向 flit 是否是消息首个报文的 flit。如果是, 且被选元素空闲(即有效位 vld 为 0), 则把 Selection 选择函数的输出存储到 IOT 表; 否则, 如果被选元素分配给不同的消息(即 vld 为 1), 则采用确定型选择策略(当与首个多路转换器的选择端口相连的“与”门输出为 0 时)。

对其余报文, First 模块的输出为 0, 因此模块的输出对应于首个多路转换器的输出。此时可能有两种情况: 如果 msg id 的标记字段不等于 IOT 中存储的标记, 或者如果被选择的元素的 vld 位为 0, 则存在错失(miss), 于是采用确定型路由; 否则(即标记匹配且 vld = 1), 表示命中并, 将 IOT 表被选元素的 out 字段所存储的值作为输出端口。最后, 对最后一个报文, 将 Last 模块的输出(若当前 flit 为消息最后一个报文的首部 flit, 则对其进行设置)与命中信号一并使用, 令被选 IOT 元素无效。

2.2 合成结果

本文系统用 VHDL 语言进行设计, 利用 Synopsys 设计编译器进行合成, 并在 TSMC 公司 90 纳米技术库中进行映射, 部署了三种基于虫孔的路由器, 即 XY 确定型路由函数(XY)、Odd-Even^[12]路由函数(OE)及与 Odd-Even 路由函数集成的有序传输机制(OE + IO)。

后面两种路由器部署了部分自适应路由策略。对这两种路由器, 当路由函数返回两个或两个以上输出时, 将会利用选择策略以便选择连通输入端口中被占用的缓存最少的输出。经设计, 所有路由器工作频率为 1 GHz, 采用的流水线机制基于如下三个阶段, 即 FIFO、路由/选择、仲裁/纵横。

所有部署中的 FIFO 缓存深度为 6 个 flits, flit 尺寸为 64 位。对于 OE + IO, 假设不同部署的 IO 表尺寸不同, 可表示为 OE + Io n, 其中 $n = 2, 4, 8, 16$, 表示 IOT 表的元素数量。表 1 给出了 XY 路由器面积和功耗增长比例。从表中可见, 视 IOT 表尺寸不同, 本文方法在面积和功耗方面的开销为 4% ~ 19%。虽然算法的有效性随着 IOT 尺寸的增长而增长, 但是实验结果表明, 当 IOT 有四个以上元素时, 并不能带来显著的性能提升, 此时, 面积和功耗方面的开销分别低于 7% 和 6%。

表 1 XY 路由器的面积和功耗增长百分比 / %

路由器	面积	功耗	路由器	面积	功耗
OE	2	2	OE + IO8	10	11
OE + IO2	4	4	OE + IO16	18	19
OE + IO4	7	6			

3 仿真实验

本章利用合成流量数据及从真实应用中提取出来的流量数据来评估本文方法的性能。首先, 测量具有相同源节点和目的节点的两个连续报文注入网络后到达目的节点时次序发生混乱这一事件的频率, 对报文无序传输问题进行分析; 然后, 在多种流量场景下进行全面的性能分析, 强调平均通信延时降低及饱和点增加这两方面的性能提升, 同时分析和讨论了对能耗产生的影响; 最后, 结合一个实例对本章作总结。

3.1 测量无序报文的数量

首先, 评估报文到达目的节点时次序发生混乱的频率。假设有一个 8×8 mesh 拓扑结构, 路由算法为 Odd-Even^[12], 报文尺寸为 4 flit, 流量模式为均匀模式和 1 型反转模式。图 4 给出了随机策略和缓存区选择策略下到达目的节点后次序混乱(OoO)的报文比例。如果路由函数返回多个输出端口, 则随机选择策略随机选择输出端口, 而缓存区水平策略选择与输入端口相连(相邻路由器中), 且被占用的缓存最小的输出端口。

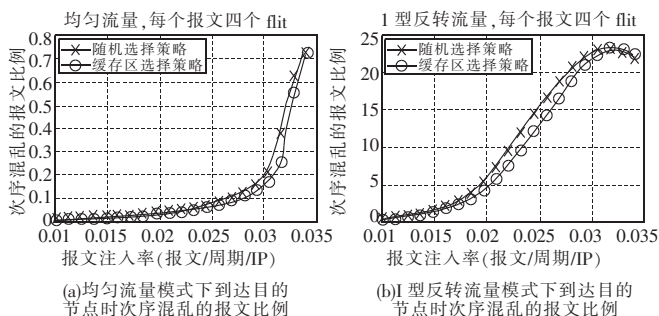


图 4 到达目的节点时次序混乱的报文比例

从图中可见, OoO 比例与流量场景有着密切关系。在均匀流量场景下不到 1%, 但是在反转场景下超过 20%。具体来说, 采用拥塞感知型选择策略时(如缓存区选择策略), OoO 比例低于采用随机选择策略时。根据文献[10]可知, 到达目的节点发生次序混乱的报文比例还与源节点和目的节点间的距离有关。如笔者预期, 当路由路径变长时, 次序混乱的概率也会增加。对于 8×8 mesh 网络, 均匀和反转流量模式下, 平均路由路径长度分别为 5 跳和 12 跳。这在一定程度上解释了两种场景下 OoO 值差异的原因。

3.2 性能分析

然后, 本文比较 XY 路由、Odd-Even 路由及采用了本文有序传输策略的 Odd-Even 路由的性能。考虑 8×8 mesh 拓扑结构, 分析不同合成流量场景及消息长度下的平均报文延时。在分析时考虑了所有注入消息为阻塞型消息的情景。这就是说, 即使采用确定型路由策略, 源节点在接收到前一传输消息的确认消息前不得注入新的消息。

图 5 给出了比特反转流量场景下不同 pirs^[5] 及消息包含 2、4 和 8 个报文时各种路由策略的平均延时。从图中可见, 当消息尺寸下降、IOT 尺寸上升时, 本文算法的有效性增加。无论消息尺寸如何, 相对于确定型路由算法的性能优势均非常明显。具体来说, 消息尺寸为 8、4 和 2 个报文时, 平均延时分别下降 33%、50% 和 56%。与 Odd-Even 相比, 消息尺寸较小时的差异并不明显, 但是当消息尺寸增加时差异显现。之所以出现这一现象, 是因为当消息尺寸增加时, 本文算法的自适应性下降, 自适应性也未得到有效利用。其原因如下: a) 平均而言, IOT 表的普通元素被占用的时间变长, 因此新的消息发现 IOT 元素不可用的概率增加, 于是消息必须按照确定型路由传输, 导致自适应性下降; b) 自适应的精细化水平下降, 也就是说, 由首个报文自适应选择的路由路径被使用很长时间, 而网络状态在发生变化, 可能导致初始选择不再最优。

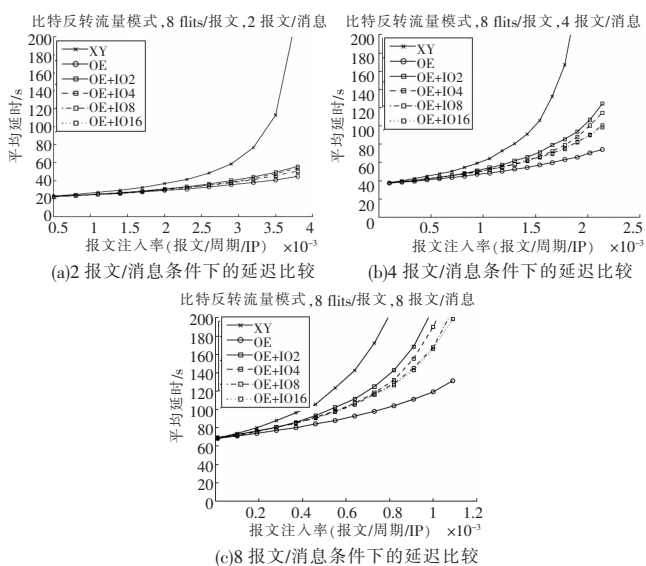


图 5 不同流量场景及消息尺寸条件下的延时变化情况

图 6 给出了本文方法在不同消息尺寸条件下与 XY 算法相比每个 flit 的平均能耗百分比增加值。每个 flit 的能量计算为总能耗和注入网络的 flit 的总数量之比。如笔者预期, 当消息尺寸增加时, 每个 flit 的平均能量上升。当 IOT 表只有 2 或

4 个元素时, 无论消息尺寸如何, 能量开销平均来说均不到 1%; 当 IOT 表有 8 或 16 个元素时, 能量开销平均来说分别不到 1.5% 和 2.5%。在部分情况下能量开销百分比为负。这就是说, 虽然路由器消耗了更多的能量, 但是因为注入的流量在更少时间内处理完, 所以消息总能量下降。当消息尺寸较大时, 对 Hot-spot、Shuffle 和 Transpose 等流量来说能效非常明显。

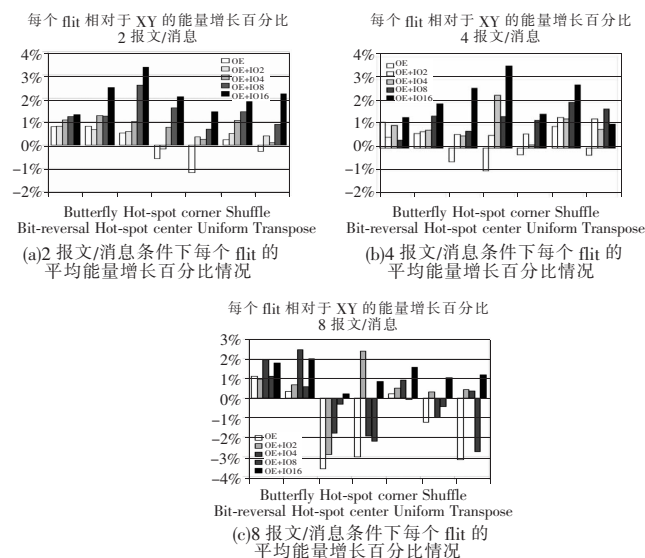


图 6 相对于 XY 在不同消息尺寸条件下每个 flit 的平均能量增长百分比情况

3.3 实例分析: 多媒体系统

上述分析基于合成流量场景。本节利用复杂的异构系统来评估本文方法的有效性, 如图 7 所示。系统包括如下子系统^[13]: a) MMS, 包括 H. 263 视频编码器、H. 263 视频解码器、mp3 音频编码器和 mp3 音频解码器的一个通用多媒体系统; b) MIMO-OFDM 接收器; c) PIP 和多窗口显示应用(MWD)。在该案例研究中, 内存流量限制为有序流量, 而所有剩余通信中的报文可以无序到达目的节点。

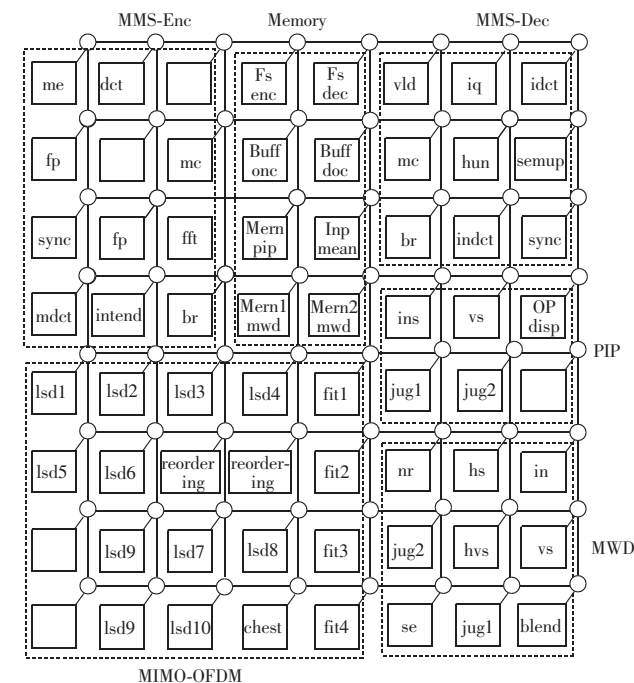


图 7 由多媒体子系统、MIMO-OFDM 接收器、PIP 和 MWD 模块组成的异构系统

图 8 给出了本文方法与 XY 相结合时最大吞吐量的平均增长幅度。MMS-Enc 和 MIMO-OFDM 的提升效果最为明显,因为它们映射到更多的内核上。这表明网络规模越大,自适应性越强,性能越有优势。图 9 给出了每个 flit 的能量增长百分比情况。观察结论与合成流量情景类似,路由逻辑开销导致每个 flit 的能量总体上呈上升趋势。然而在部分情况下(如 MMS-Enc 和 MIMO-OFDM),每个 flit 的能量下降。

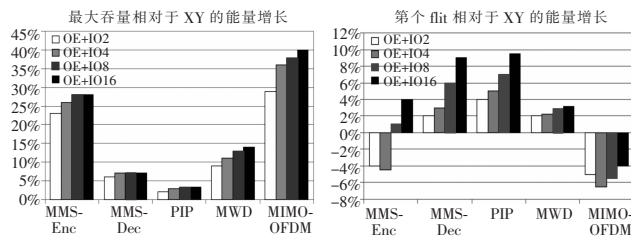


图 8 最大吞吐量相对于 XY 的平均增长情况

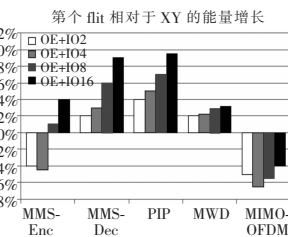


图 9 每个 flit 的能量相对于 XY 的增长百分比情况

4 结束语

本文提出一种方法可保证不牺牲底层路由算法自适应性的情况下保证报文按序交付。与 Odd-Even 路由算法及可有效保证所有报文按序交付的一种确定型路由算法进行了性能评估和比较。与确定型路由算法相比,本文方法可有效提升目标流量场景下的平均延时和饱和点性能,同时评估了对路由器面积和功耗的影响。虽然路由器的功耗增加,但是由于报文交付性能提升,所以每个 flit 的能耗增长可忽略不计。在部分情况下,每个 flit 的能耗甚至会下降。本文下一步研究的重点主要是降低本文算法部署时的逻辑复杂度,以及其他真实的流量场景下评估本文方法的性能。

参考文献:

- [1] 张帅,宋凤龙,王栋,等.多核结构片上网络性能——能耗分析及优化方法[J].计算机学报,2013,36(5):988-1003.
- [2] 冯超超,张民选,李晋文,等.一种可配置双向链路的片上网络容错偏转路由器[J].计算机研究与发展,2014,51(2):454-463.
- [3] 付斌章,韩银和,李华伟,等.面向高可靠片上网络通信的可重构

路由算法[J].计算机辅助设计与图形学学报,2011,23(3):448-455.

- [4] 孙光,张媛媛,李勇,等.基于虚拟化片上网络的二级优先顺序路由算法[J].清华大学学报:自然科学版,2011,51(3):416-419.
- [5] Palesi M, Holsmarkk R, Kumar S, et al. Application specific routing algorithms for networks on chip[J]. IEEE Trans on Parallel and Distributed Systems, 2009, 20(3): 316-330.
- [6] Ikram S, Asher D, Akkawi I, et al. A framework for specifying, modeling, implementation and verification of SOC protocols[C]//Proc of the 27th IEEE International System-on-Chip Conference. [S. l.]: IEEE Press, 2014: 268-273.
- [7] Ortin-Obon M, Ramini L, Tatenguem F H, et al. A complete electronic network interface architecture for global contention-free communication over emerging optical networks-on-chip[C]//Proc of the 24th Edition of the Great Lakes Symposium on VLSI. [S. l.]: ACM Press, 2014: 267-272.
- [8] Rahmati D, Murali S, Benini L, et al. Computing accurate performance bounds for best effort networks-on-chip[J]. IEEE Trans on Computers, 2013, 62(3): 452-467.
- [9] Murali S, Atienza D, Benini L, et al. A multi-path routing strategy with guaranteed in-order packet delivery and fault-tolerance for networks on chip[C]//Proc of the 43rd Annual Design Automation Conference. New York: ACM Press, 2006: 845-848.
- [10] Lis M, Shim K S, Cho M H, et al. Guaranteed in-order packet delivery using exclusive dynamic virtual channel allocation, MIT-CSAIL-TR-2009-036[R]. Cambridge: Computer Science and Artificial Intelligence Laboratory, Massachusetts Institute of Technology, 2009.
- [11] Ascia G, Catania V, Palesi M, et al. Implementation and analysis of a new selection strategy for adaptive routing in networks-on-chip[J]. IEEE Trans on Computers, 2008, 57(6): 809-820.
- [12] Chiu G M. The odd-even turn model for adaptive routing[J]. IEEE Trans on Parallel and Distributed Systems, 2010, 11(7): 729-738.
- [13] Hu Jingcao, Marculesu R. Energy-and performance-aware mapping for regular NoC architectures[J]. IEEE Trans on Computer-Aided Design of Integrated Circuits and Systems, 2005, 24(4): 551-562.