



南京大學

本科畢業論文

院 系 電子科學與工程

專 業 微電子科學與工程

題 目 無線片上網絡通信協議設計優化

年 級 15 學 號 151180032

學生姓名 高茜

指導老師 李麗 職 稱 教授

提交日期 2019 年 5 月 9 日

南京大学本科生毕业论文（设计）中文摘要

毕业论文题目：无线片上网络通信协议设计优化

电子科学与工程 院系 微电子科学与工程 专业 15 级 本科生姓名：高茜

指导教师（姓名、职称）：李丽 教授

摘要：

片上网络 (NoC) 是实现片上多核系统 (SoC) 的经典架构。无线片上网络 (WiNoC) 延迟低，布局布线简单，可以减少传统 NoC 架构互联带来的能耗，具有成为下一代主流通信平台的潜力。本研究旨在加深对 WiNoC 的无线通信协议的了解，并对传统基于时分复用的介质访问控制 (MAC) 机制进行算法层面的优化，通过对无线通信需求的感知实现动态 MAC 机制，从而改善网络性能。

本文主要包括对无线片上网络通信架构的模型介绍、对国内外研究现状的总结展望、对传统无线介质访问机制的算法分析、基于指数平滑法的无线通信需求预测、动态介质访问机制的模拟仿真等内容。其中，对动态 MAC 机制的效果分析包括：不同流量模型、不同网络拓扑、不同路由算法、不同令牌持有时间、以及模拟实际应用的性能仿真。通过无线 MAC 机制的改进优化，网络性能在上述提及的各种综合复杂环境条件下均获得了较好的实验结果：相比传统 MAC 机制，采用动态 MAC 机制后网络平均延迟降低，进入饱和点的数据包注入率提高，体现出动态 MAC 机制的优越性。

关键词：无线片上网络；动态 MAC 机制；时分复用；指数平滑法

南京大学本科生毕业论文（设计）英文摘要

THESIS: Robust and Traffic Aware Medium Access Control Mechanisms for Wireless Network-on-Chip Architectures

DEPARTMENT: School of Electronic Science and Engineering

SPECIALIZATION: Microelectronic Science and Engineering

UNDERGRADUATE: Qian Gao

MENTOR: Professor Li Li

ABSTRACT:

NoC(Network on Chip) is a classic architecture for implementing SoC(System on Chip). Wireless NoC(WiNoC) has low latency and simple layout and wiring, which can reduce the energy consumption caused by the interconnection of traditional NoC architecture. As a result, it has the potential to become the next generation of mainstream communication platform. This research aims to deepen the understanding of WiNoC wireless communication protocol, to optimize the traditional time division based media access (MAC) mechanism at the algorithm level, and to realize dynamic MAC mechanism so as to improve network performance.

The main contents of this paper include the introduction of the wireless network on chip communication architecture model, the summary of the research status worldwide, the analysis of traditional wireless media access mechanism, the prediction of wireless communication demand based on exponential smoothing method, and the simulation of dynamic media access mechanism. Among them, the simulations of dynamic MAC mechanism are completed under various conditions: different traffic patterns, different network topologies, different routing algorithms, different token holding time, etc. Simulation of performance in practical application is also included. Through the improvement and optimization of wireless MAC mechanism, the network performance has obtained better experimental results under the various conditions mentioned above. Compared with traditional MAC mechanism, the average delay of network decreases after adopting dynamic MAC mechanism, and the network enters the saturation point at higher packet injection rate, reflecting the superiority of the proposed dynamic MAC mechanism.

KEY WORDS: Wireless NoC;MAC protocol;TDMA;Exponential Smoothing

目 录

第 1 章 绪论	1
1.1 研究背景及意义	1
1.2 论文主要工作及创新点	2
第 2 章 WiNoC 通信架构与研究现状	3
2.1 WiNoC 通信架构	3
2.1.1 通信链路	3
2.1.2 路由器	3
2.1.3 网络接口	6
2.2 WiNoC 关键问题研究现状	6
2.2.1 网络拓扑结构	6
2.2.2 路由算法	7
2.2.3 无线通信技术	7
2.2.4 无线介质访问协议	8
第 3 章 WiNoC MAC 机制研究	10
3.1 基于时分复用的 MAC 机制	10
3.1.1 单无线信道 MAC 机制	10
3.1.2 多无线信道 MAC 机制	12
3.2 动态 MAC 机制	13
3.2.1 无线信道的需求感知	14
3.2.2 无线信道的需求预测	14
3.2.3 基于指数平滑法的动态 MAC 机制	17
3.2.4 动态切换 MAC 机制	19

第 4 章 WiNoC 性能仿真	20
4.1 仿真平台介绍	20
4.2 仿真参数说明	22
4.3 仿真结果	23
4.3.1 不同流量模型下仿真结果	23
4.3.2 不同无线拓扑下仿真结果	26
4.3.3 不同路由算法下仿真结果	28
4.3.4 不同令牌持有时间下仿真结果	29
4.3.5 模拟实际应用仿真结果	30
4.4 实验结论	31
第 5 章 总结与展望	33
5.1 总结	33
5.2 有待解决的问题及展望	33
5.2.1 指数平滑法中参数的选择	33
5.2.2 无线利用率问题	34
5.2.3 令牌循环总时间的问题	34
参考文献	35
致谢	40

第 1 章 绪论

1.1 研究背景及意义

随着半导体技术的发展，片上系统（SoC）的集成度将显著提高，传统基于总线的互连架构将难以满足某些特定应用的功能要求：对于需要进行频繁并行通信的系统，总线架构将无法提供通信所需的带宽。因此，研究人员引入了嵌入式交换网络，即片上网络（NoC）来互连 SoC 中的 IP 模块。相比于总线互联架构，NoC 的设计空间和可扩展性更高：研究者可以实现不同的网络拓扑、路由算法、仲裁策略等。此外，片上网络具有良好的并行通信能力，并且可以跨时钟域。^[1]

然而，实现高效能的 NoC 架构通信也面临着诸多挑战。一方面，传统片上网络采用金属线进行互联。随着 NoC 规模不断扩大，金属线的长度和网络的布线复杂度将不断增加，这将使互联带来的能耗变高。国际技术半导体发展路线图（ITRS）数据显示，在 0.13 微米的制作工艺下，金属导线功耗将占据 NoC 总功耗的 50% 以上。在未来，这一比例还会继续上升。另一方面，传统 NoC 通信采用多跳传输，即数据包到达路由器后要经过存储、运算、仲裁后转发给下一个路由器，直到传输到目的节点。因此，当源节点与目的节点距离较远时，数据包在传输过程中要经过多个路由器，从而带来传输延迟。研究人员采用三维^[2]，光学^[3]和具有多波段射频（RF）互连（RF-I）^[4]的 NoC 结构来降低延迟和功耗，但无法解决跨芯片互连的难题。此外，三维和光学 NOC 必须克服技术和制造方面的挑战，才能实现大规模生产。虽然可以使用现有的 CMOS 技术来构建 RF-I NoC，但这种 NoC 架构需要铺设片内传输线以用作波导。

随着硅集成天线在几十到一百兆赫的毫米波范围内工作特性的确定，基于天线的通信成为了一种可行的芯片内和芯片间通信技术。此外，在光学频率下工作的碳纳米管（CNT）具有优异的发射和吸收特性，与天线特性十分相似^[5]。这些发现为基于片上天线和收发器的无线 NOC（Wireless NoC，WiNoC）设计提供了新的思路。片上无线通信链路不仅减少了传统 NoC 架构远距离多跳通信的延迟，而且消除了三维、光学、RF-I NoC 中的复杂互连路由和布局问题，表明了无线片上网络（WiNoC）作为未来主流通信平台的潜力。

1.2 论文主要工作及创新点

本论文主要分为 4 个部分：

第二章对 WiNoC 的通信架构与研究现状进行简单介绍；

第三章重点分析 WiNoC 中的无线介质访问控制协议。传统的基于令牌介质访问机制 (Token-based MAC) 方案无需信号同步，适用于片上多核系统，在网络负载较高时能高效协调无线资源。但是，由于网络流量分布的不均衡，各无线节点对无线信道的需求会有较大的不同。因此，本论文提出了一种动态 MAC 机制，结合时间序列预测算法，通过当前时刻或过去时刻不同无线节点对无线信道的使用情况预测下一周期内的使用情况，再据此将每个无线节点持有令牌的时隙长短按比例分配，以提高无线信道利用率；

第四章的仿真实验结果表明，这种动态 MAC 机制有利于实现低延迟的片上通信；

第五章对论文工作进行总结并提出待解决的问题与未来工作展望。

论文创新点：

1. 为 WiNoC 中的无线节点加入通信情况感知与预测模块
2. 采用指数平滑法对无线节点通信需求进行预测
3. 根据预测结果对 MAC 机制进行动态切换
4. 根据预测结果对每个无线节点持有令牌的时隙进行按比例分配

第 2 章 WiNoC 通信架构与研究现状

2.1 WiNoC 通信架构

无线片上网络由三个主要模块组成。第一个是物理上连接节点并实现通信的链路（**Link**）。第二个模块是路由器（**Router**），它从共享链路接收数据包，并根据每个数据包中含有的目的地址，将数据包转发到另一个共享链路。最后一个模块是网络接口（**Network Interface, NI**）。它在 IP 核和网络之间建立逻辑连接，每个 IP 可以有不同的网络接口协议。

2.1.1 通信链路

通信链路连接网络中的路由器。链路可以包括一个或多个物理信道，每个信道由一组线路组成。链路的实现涉及源节点和目标节点之间同步协议的定义。该协议可以通过在通信期间设置的专用线路或通过诸如 FIFO 的其他方法来实现^[38]。数据在通信链路上传输时会带来延迟。因此，在网络中的节点之间提供快速，可靠和低功耗的互连非常必要。

微片（flit）是链路层面的概念，它是形成数据包和数据流的基本单元。在大多数情况下，flit 的大小由物理单位衡量（例如 16 bits），这是在网络中传输的最小数据量。微片包含 3 部分：头微片（Head Flit），体微片（Body Flit）和尾微片（Tail Flit）。头微片中存储了数据包的路由信息，体微片是数据包的负载段，尾微片则用于释放路由路径。

2.1.2 路由器

路由器由本地端口，多个输入端口和输出端口，连接输入和输出端口的交换矩阵组成。其中，输入端口用来连接到共享 NoC 信道，而输出端口用来连接到可能的其他共享信道。除了上述物理连接，路由器还包含一个逻辑块，用于实现流量控制策略（路由，仲裁等）和 NoC 移动数据的整体策略。

- 流控制策略表征数据流沿 NoC 的分组移动，因此它涉及全局（网络级）和本地（路由器级）两个层面。可以通过在流控制策略中采取特定措施（例如通

过避免数据流通过 NoC 内的某些路径) 来确保无死锁路由。此外, 通过优化 NoC 资源使用 (通道, 带宽等), 可以提高 NoC 通信效率。例如, 可以通过引入虚拟信道 (virtual channel) 来减少拥塞, 或者可以通过选择最小路径路由来减小传输延迟。

- 路由算法根据数据包标头中可用的路由信息 来选择到达路由器的数据包的输出端口。不同路由算法对 NoC 整体性能和开销有较大影响。在确定性路由中, 源节点与目的节点之间的数据传输在不同实验情况下总是选择相同的路径。例如, 在 XY 路由中, 数据包首先沿着行 (X), 然后沿列 (Y) 向目的地移动, 反之亦然。然而, 在自适应路由算法中, 路径的选择会考虑到链路拥塞。这涉及链路负载的动态评估。Glass 和 Ni^[37] 提出的 Negative First (NF) 和 West First (WF) 算法是自适应路由算法的例子。
- 缓冲策略是当网络拥塞并且无法立即转发数据包时用于在路由器中存储信息的策略。缓冲策略 (缓冲区的数量, 位置和大小) 对网络流量和 NoC 整体性能有重要影响。此外, 缓冲区占路由器整体区域的比例很大。在路由器中, 可能只有有一个缓冲区, 由所有输入端口共享; 也可能每个输入或输出端口都有一个缓冲区。

传统路由器

图 2.1 是一个用于二维网格结构的传统路由器基本架构, 它包含 5 个可以双向传输的数据端口: 本地 Local, 东 East, 南 South, 西 West, 北 North。其中, E,S,W,N 四个端口分别与对应方向的路由器相连, L 端口进行与本地 IP 核之间的数据传输。构成路由器的主要结构有数据缓冲区 (buffer), 仲裁器 (Arbiter), 数据链路和交叉开关。其中, 仲裁器包含路由逻辑与流控制逻辑电路。

无线路由器

无线路由器在传统路由器的基础上增加了无线接口 (Wireless Interface,WI)。无线接口由收发器 (transceiver) 相关电路和片上天线 (antenna) 组成。考虑到片上天线的设计开销, 多个传统路由器可以共享一个无线接口, 即无线电集电器 (radio hub), 如图 2.2 中多个灰色节点连接到同一个黑色的无线接口。集电器与传统路由节点

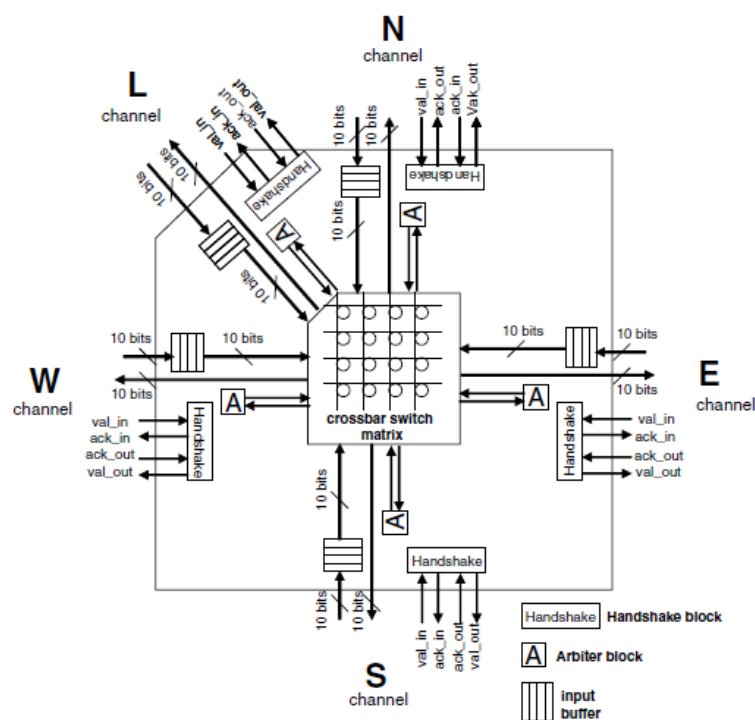


图 2.1: 传统路由器组成架构

通过有线链路进行连接，即图 2.3 中的蓝色箭头，将接收到的数据存储在缓冲区，经过判断后，如果选择无线传输，则将数据包再传输到天线缓冲区 (tx_{CH0})，通过无线信道 (wireless channel) 传输到另一个集电器的天线接收端，存储在接收端的缓存区 (rx_{CH0})。无线传输通常用于距离较远的两节点间通信，从而减少 NoC 通信平均延迟。

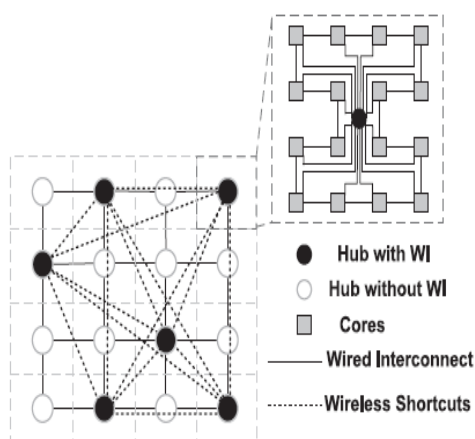


图 2.2: 包含无线路由器的 NoC 架构

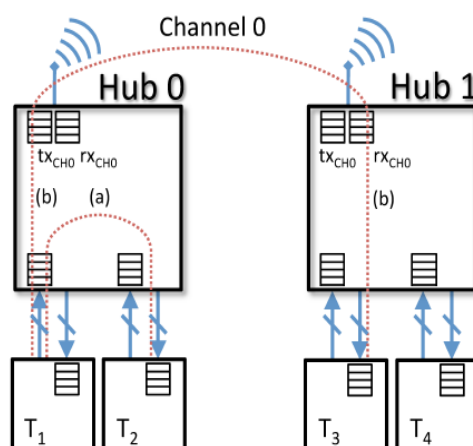


图 2.3: 无线接口内部结构

2.1.3 网络接口

网络接口 (NI) 也被称为网络适配器 (Network Adapter, NI), 它在 IP 核和网络之间建立逻辑连接。每个 IP 可以具有关于网络的不同接口协议。网络接口的存在使计算和通信之间的分离成为可能。网络接口分为前端和后端, 前端处理核心请求, 后端处理网络协议, 如组装和拆解包, 重新排序缓冲区, 实现同步协议, 在存储方面帮助路由器等。

2.2 WiNoC 关键问题研究现状

国外对无线片上网络的研究大约从 2009 年开始, 国内则研究较少。由于 WiNoC 由传统 NoC 发展而来, 在一些研究内容上, 如网络拓扑、路由算法、流控制策略等, WiNoC 借鉴了 NoC 中已有的研究成果。不同的是 WiNoC 还涉及对无线通信的研究, 如天线、收发器设计, 无线通信协议设计等。

2.2.1 网络拓扑结构

拓扑结构反映了路由器在 WiNoC 中的分布以及它们之间的连接方式。拓扑结构大体分为两类: 规则型拓扑 (例如传统基于网格 mesh 的拓扑) 和不规则型拓扑 (例如基于小世界 small world 的拓扑)。

D.Zhao^[6] 提出了基于 CMOS 超宽带 (UWB) 技术的纯无线 NoC 设计。如图 2.4 所示, 处理器通过具有数据收发能力的射频 (RF) 节点传输数据。与 64 核基于网格 (mesh) 结构的有线 NoC 相比, WiNoC 平均执行时间加快了 23.3%, 平均端到端延迟降低了 65.3%。然而, 多个天线部件的集成将带来面积与功耗问题。

D. DiTomaso^[7] 提出采用 iWise 架构, 在毫米波频率范围内通过构建路由器间无线可扩展快速通道来降低功耗并提高性能。在这项工作中, 节点以网格方式排列, 每个路由器都有自己的发送器和接收器。与集中式集线器不同, 这种设计通过在每个路由器上分配收发器来减少跳数。这种带有分布式无线集线器的混合网络允许单跳 64 核网络和三跳 256 核网络。

A.Ganguly^[8] 等人提出了一个混合有线传输与无线传输 WiNoC 架构, 如图 2.5 所示, 整个网络被分成多个子网络, 子网络内部通过有线链路连接, 而子网之间通过有线链路或无线链路相连, 形成一个具有“小世界”结构的混合网络。

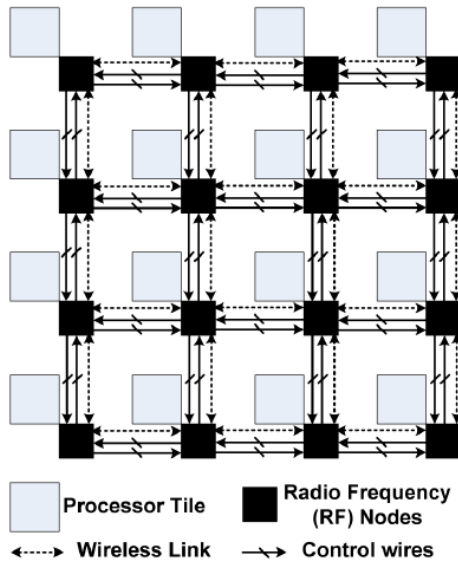


图 2.4: UWB 4*4 2D 网络 WiNoC 架构

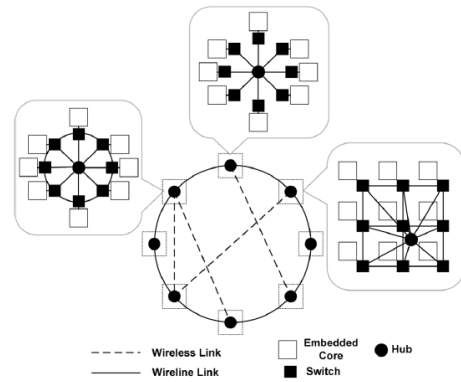


图 2.5: 小世界不规则 WiNoC 架构

2.2.2 路由算法

路由算法决定了网络节点之间通信时选择经过的路径，对片上网络传输的延迟时间有着重要影响。如上文所述，WiNoC 基本沿用了 NoC 的路由算法。

D.Zhao^[6] 提出了一种基于超宽带 (UWB) 的 WiNoC 确定性路由算法，通过限制特定位置的转弯来避免死锁 (dead lock)。该算法确保了分布式、无周期、最短路径的路由，从而有利于低延迟、低功耗和低区域开销的片上通信。

对于基于小世界的 WiNoC，无线互联的引入限制了可采用的路由算法。U. Y. Ogras^[9] 提出，如果整个网络是网格结构，则可以采用 XY 路由算法和 south-east 路由算法的组合，即对于无线路由器的通信采用 south-east 路由算法，有线路由器的通信采用 XY 路由算法。这种结构同样可以避免死锁。

2.2.3 无线通信技术

WiNoC 的通信效果极大程度上由物理层的无线路由设计所决定，即天线 (antenna) 设计与收发器 (transceiver) 结构设计。而天线传输特性与通信频率有很大的关系。根据频率范围，WiNoC 的研究可分为四类：超宽带 (UWB, 1-10GHz), 毫米波段 (mm-wave, 10-100GHz), 次太赫兹波段 (Sub-THz, 100-1000GHz), 太赫兹波段 (THz, 1000-10000GHz), 如图 2.6 所示。

D.Zhao^[6] 采用了无载波脉冲无线电 UWB 收发器。该发射机设计来产生具有

适当驱动强度的脉冲，从而使信号能够从片上天线有效辐射。接收器 (transceiver) 由同步电路、模数转换器 (ADC) 和宽带低噪声放大器 (LNA) 组成。使用的天线是一个 2.98 毫米长的弯曲型偶极子天线，数据传输范围为 1 毫米。该 UWB 收发器可以在中心频率为 3.6GHz 的情况下，为单个信道维持 1.16GB/s 的数据速率。

对于毫米波 WiNoC, 片上天线需要在最小区域开销的前提下达到最佳功率增益。金属锯齿形天线 (Zig-zag antenna) 已经被证明具有如上特性^[10]。这种天线可以忽略旋转（发射天线和接收天线之间的相对角度）对接收信号强度的影响。类似的，为确保高吞吐量和高能效，收发器电路必须提供高带宽和低功耗。X. Yu^[11] 提出了基于非相干的开-关键控 (OOK) 的收发器设计。这种电路设计较为简单且功耗低，如图 2.7 所示。

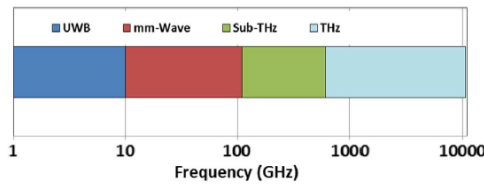


图 2.6: WiNoC 通信频段

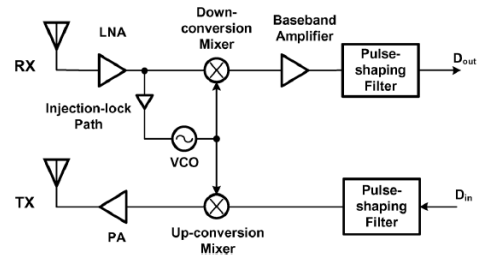


图 2.7: 毫米波 WiNoC OOK 收发器

2.2.4 无线介质访问协议

在 WiNoC 中，多对无线节点可能同时有数据传入，从而导致对无线信道的争用，有可能导致信号碰撞。为避免出现这一情况，需要对无线信道进行合理的分配，即选择合适的无线介质访问控制协议 (MAC protocol)。

从收发器设计的角度来看，设计多个频率上非重叠信道以实现频分多址 (Frequency Division Media Access, FDMA) 的难度较大，并且扩展性低。因此，大多数 WiNoC 中多个无线接口共享单频无线信道。文献^[27] 中提出了基于码分多址 (Code Division Media Access, CDMA) 的 MAC 机制，但该机制需要保持发射机中的代码信道之间的正交性。

D.Zhao^[6] 提出了一个分布式同步访问 (SD-MAC) 协议。其中，信道分配决策分成四步：(1) 初始化（每个参与竞争的无线节点随机生成二进制伪随机码）；(2) 信道竞争（无线节点将生成的伪随机码送到仲裁单元）；(3) 信道授权（仲裁单元将无线信道授权给伪随机码最大的无线节点）；(4) 数据传输（获得授权的无线

节点开始进行数据传输)。文献^[12]提出了一种基于媒体访问机制的码分复用技术(CDMA),然而它需要高能耗的收发器来保持同步,以维持发射器中的代码信道之间的正交。

考虑到片上有限的资源和相对简单的无线信道状况,基于时分多址技术(TDMA)的令牌介质访问机制被广泛研究^[13]。该算法在无线路由器节点之间传输令牌(Token),只有持有令牌的无线节点才能使用无线信道进行通信。在数据传输完成后,令牌会被传递到下一个无线路由节点。

第 3 章 WiNoC MAC 机制研究

访问控制 (Access Control) 指的是调解各个组成结构对系统资源和数据的请求, 决定是应该授予或是拒绝请求的过程。对于无线片上网络而言, 数据在无线信道中以广播的方式传播, 一个信道中同一时刻只能有一对无线节点进行数据的传输, 即所有无线节点共享同一信道。因此, 要保证无线信道的通信效率必须为 WiNoC 执行严格的介质访问控制协议 (Media Access Control protocol)。

常见的无线介质访问控制协议可以分为两大类, 即基于竞争的和基于时分多址 (TDMA, Time Division Media Access) 的协议^{[14][15][16]}。基于竞争的 MAC 协议允许节点独立访问共享无线介质^{[14][16]}。在基于 TDMA 的 MAC 协议中, 系统时间被分成多个时隙, 每个无线集电器节点 (Radio Hub) 都分配有自己的时隙, 并且仅可以在该时隙中访问共享的无线介质。该种协议可以避免节点对无线介质的访问冲突, 并可以控制收发器的休眠时间, 而无需额外的开销。

3.1 基于时分复用的 MAC 机制

考虑到片上资源的有限性, WiNoC 无法支撑十分复杂的 MAC 协议。通常, 在 WiNoC 架构中使用的无线介质访问控制协议 (MAC protocol) 基于时分复用 (TDMA)^{[8][17][18]}, 其中, 每个集电器可以传输数据的时隙长短通过令牌 (token) 进行调节。“令牌”是一个特殊数据段, 它在各个 radio hub 之间循环^[19]。

3.1.1 单无线信道 MAC 机制

无线电介质一次只可以由单个无线电集线器访问。因此, 必须使用无线电介质访问控制协议来避免多个无线电集线器同时访问无线电介质。首先, 可以将网络中所有的无线电集线器连接成一个虚拟的环形网络拓扑 (Token Ring), 如图 3.1 所示。令牌在这个环中按照一定方向进行循环, 通过每个无线电集线器, 只有持有令牌的无线电集电器才能够通过无线信道进行数据传输。

该机制可以通过简单的有限状态机来实现, 如算法 1 所示。令牌环的每个节点 (即无线电集电器) 具有两个输入和两个输出, 即令牌输入 (*token_in*), 请求输

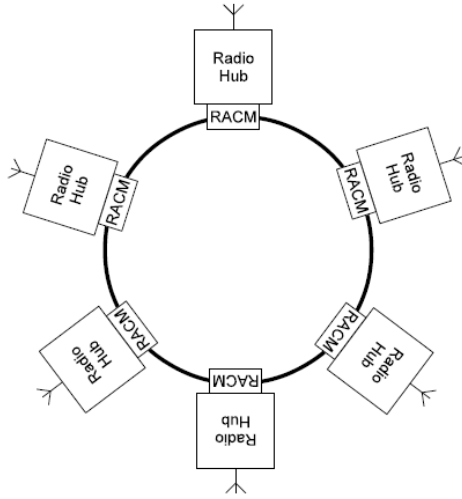


图 3.1: 令牌环示意图

入 (*request*), 令牌输出 (*token_out*) 和授权输出 (*grant*)。此外, 每个节点还有一个计数器 (*hold_count*) 来统计该节点持有令牌的时隙长短, 以及一个布尔类型的变量 *owns_token* 用来记录该节点是否目前持有令牌。

无线电集线器使用输入请求 (*request*) 来请求访问无线电信道。如果环节点拥有令牌 (*owns_token* 为真), 则给予输出授权 (即将 *grant* 值设置为真), 否则没有输出授权 (即将 *grant* 值设置为假)。此外, 在节点持有令牌时, *hold_count* 变量进行累加计数, 当数值超过某个预先设定的阈值 (*Hold_Cycle*), 该节点就将令牌传递给下一个节点。因此, 如果用 HC_i 表示环的第 i 个节点的可以持有令牌的时间, 则第 j 个无线电集线器每隔 $\sum_{i \neq j} HC_i$ 个时隙才会有无线电介质的接入。

此外, 每个无线电集线器的 *hold_count* 和 *owns_token* 寄存器值在重置期间都会被初始化: *hold_count* 寄存器初始化为零, 而 ID 为 0 的 *radio_hub* 的 *owns_token* 值初始化为 **true** (假定令牌从编号为 0 的节点开始循环), 而其他无线电集线器的 *owns_token* 值初始化为 **false**。

Algorithm 1 传统基于时分复用的 MAC 机制

Require: $request, token_in$

Ensure: $grant, token_out$

```
if  $request$  and  $owns\_token$  then
     $grant \leftarrow true$ 
else
     $grant \leftarrow false$ 
end if
if  $owns\_token$  then
     $hold\_count \leftarrow hold\_count + 1$ 
    if  $hold\_count > Hold\_Cycle$  then
         $token\_out \leftarrow token\_in$ 
         $owns\_token \leftarrow false$ 
    end if
else
    if  $token\_in$  then
         $owns\_token \leftarrow true$ 
         $hold\_count \leftarrow 0$ 
    end if
end if
```

设定 $Hold_Cycle$ 的原因是避免某个节点持有令牌的时间过长，造成其余数据包在其他节点的缓冲区排队，增加网络延迟甚至造成拥塞。这种方法的主要缺点是在网络负载低，无线使用率低的网络中，由于令牌循环所带来的开销使得无线网络通信延迟增加与无线通信效率降低。

另一种传统的基于时分复用的 MAC 机制不使用保持计数器（hold count），仅在已经发送了该节点目前排队的所有数据包后才释放令牌，被称为 $token_packet$ MAC 机制。该方法在网络负载低的情况下延迟较低，但是在网络负载较高时，由于某个节点对无线介质的使用时间长度没有限制，会导致比上述采用保持计数器的 MAC 机制更高的延迟。

3.1.2 多无线信道 MAC 机制

在设计中，WiNoC 架构可以选择无线电信道的数量。尽管目前的 WiNoC 实现仅限于使用单个无线电信道，因为天线的面积开销较大（多个信道需要多个天

线)。然而,随着工艺水平的发展,将来在同一芯片中可以集成多个无线信道时,仍然可以采用基于时分复用的 MAC 机制。

事实上,多个信道导致多个环,可以在每个环中使用该 MAC 机制。目前,一些研究分析了 WiNoC 的系统性能和能量指标随着无线信道的数量增多的变化。例如,在 [20] 中,研究者提出了一种基于 UWB 收发器的多通道 WiNoC 系统。[21] 中的作者提出了一种用于设计毫米波 WiNoC 架构的 mm 金属锯齿形天线,然后将这种单通道 mWiNoC 设计扩展到多通道 mWiNoC。

研究人员提出了一种采用亚太赫兹天线的 WiNoC^[22],它采用 32 纳米 CMOS 工艺沉积在聚酰亚胺层中。这种结构可以在 100-500GHz 的频率范围内提供不同的无线信道。此外,可以使用碳纳米管 (Carbon Nano-tube) 天线以实现太赫兹频率范围的无线信道^[23]。CNT 天线由于尺寸小,可以提供更高的通信数据速率,从而导致更低的面积开销。使用这种技术,可以实现频带上非重叠的无线信道,每个信道具有 10 Gb / s 带宽和 0.33 pJ / 位能量^[8]。

3.2 动态 MAC 机制

然而,在片上网络中,通过路由器的流量在时间和空间上不是稳定不变的,与 NoC 的应用场景有关^{[24][25]}。事实上,多核芯片中的流量模式常常被表征为突发性^[26],这将导致无线接口 (wireless interfaces) 上的流量需求的产生突然而剧烈的变化。

如图 3.2 统计了在 4 种不同流量模式下,给定相同注入率 (0.003 flit/cycle/IP), 每个无线电集电器在仿真时间内传输数据包总数的平均值。显然,在不同的流量模式下,不同 radio-hub 的利用率有很大不同:在 Random 模式下四个 hub 的利用率基本相同,而在转置流量 (transpose) 模式下,radio-hub 的利用率之间差别很大:在转置流量 1 模式下 hub2, hub3 没有传输数据。即使是在均匀 (random) 流量模式下,4 个 radio hub 传输数据量基本相同,但每个 radio hub 传输数据量随时间还是会有很大的波动。如图 3.3 统计了在 random 流量模式下某个特定无线电集电器在一段仿真周期内传输数据微片的数量,图中一个周期 (epoch) 代表 20 个系统时钟周期 (cycle)。可以看出传输数据量会有较明显的变化。因此,各个无线电集电器对无线介质 (无线信道) 的需求是非均匀的。然而,传统的基于 TDMA 的 MAC

机制无法感知通过 radio hub 的这种流量变化，并且不能动态地将传输时隙分配给 radio hub 以响应流量的突然变化。因此，为了充分发挥毫米波段下无线互连技术的潜力，MAC 机制应根据 WiNoC 中的流量变化动态分配每个无线电集电器可以占有无线介质并进行无线传输的时隙长短。

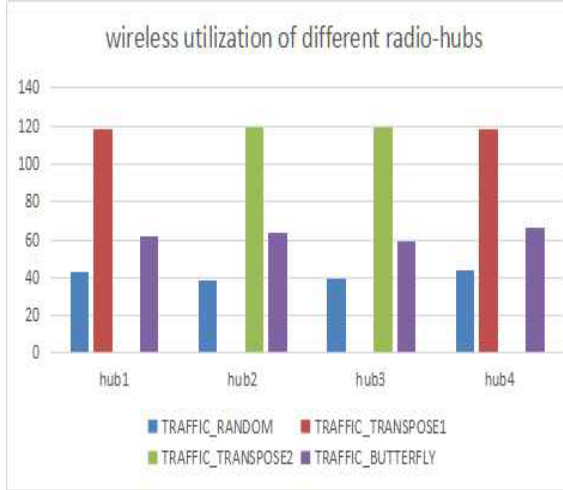


图 3.2: 不同流量模式下无线电集电器传输数据量对比

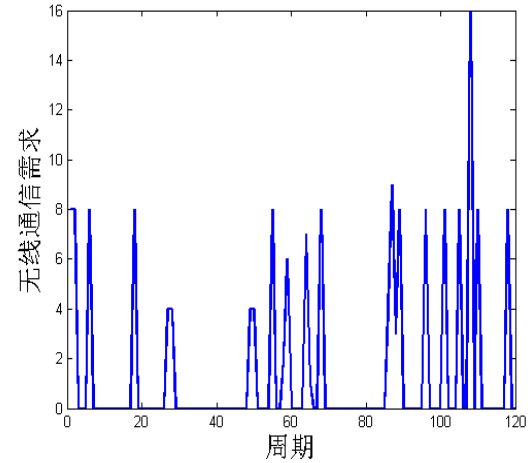


图 3.3: 均匀流量模型下某无线电集电器一段时间内传输数据量

3.2.1 无线信道的需求感知

为了动态分配每个无线电集电器可以持有令牌的时隙长短，首先要对“无线电集电器对无线信道的需求量”进行量化。如图 2.3 所示，数据在进入无线信道进行传输前，要先从 hub-tile 的缓冲区传输至天线收发器的缓冲区，即图中的 tx_{CH0} 。因此，可以采用单位时间内传输到天线发送端缓冲区的数据量（微片数量）多少来代表下一个时隙内该无线电集电器对无线信道的需求。

3.2.2 无线信道的需求预测

统计得到的传输数据量是一个时间序列。为了对持有令牌的时隙进行“按需分配”，需要对下一个时隙内每个 radio hub 的传输需求进行预测。指数平滑法 (Exponential smoothing, ES) 于 1950 年被提出^[31]，为时间序列预测提供了新思路。使用指数平滑法计算的预测值是过去和现在观测值的加权平均值。随着观测时间的推移，过去观测值的权重会以指数倍数衰减。换句话说，越近的观察值在预测

时的对应权值越高。该框架可以快速生成可靠的预测结果，适用于许多特性的时间序列，在工业应用中具有重要意义。

一次指数平滑法

如果时间序列值没有显著的趋势性，通常选择采用一次指数平滑法进行预测，下一时刻的预测值即为该时刻的一次平滑值，即

$$\widehat{y^{t+1}} = \alpha y^t + (1 - \alpha) \widehat{y^t}$$

式中， $\widehat{y^{t+1}}$ 为 $t + 1$ 时刻的预测值， y^t 为 t 时刻的实际值， $\widehat{y^t}$ 为 t 时刻的预测值。该预测值又可以递归展开，由 $t - 1$ 时刻的实际值， $t - 2$ 时刻的实际值.... 进行表示。可以观察到，由于 α 是一个小于 1 的常数，离 t 时刻越远的数据对 $t + 1$ 时刻的影响力越小，即权重呈指数次衰减。这也是“指数平滑法”的内涵。

二次指数平滑法

一次指数平滑法直接将 t 时期的一次指数平滑值作为 $t + 1$ 期的预测值^[32]。当序列值有一定的线性变化趋势，则预测结果将会呈现显著的滞后性。此时，可以采用二次指数平滑模型。二次指数平滑解决了一次平滑只能预测下一时刻值的不足，即只能预测 $t + 1$ 时刻而不能预测 $t + 2$ 时刻。二次指数平滑公式如下：

$$\begin{aligned} S_t^{(1)} &= \alpha y^t + (1 - \alpha) S_{t-1}^{(1)} \\ S_t^{(2)} &= \alpha S_t^{(1)} + (1 - \alpha) S_{t-1}^{(2)} \end{aligned}$$

y^t 为 t 时刻的实际值， $S_t^{(1)}$ 表示 t 时刻的一次指数平滑值， $S_t^{(2)}$ 和 $S_{t-1}^{(2)}$ 分别为 t 时刻和 $t - 1$ 时刻的二次指数平滑值； α 为平滑系数，介于 0 和 1 之间。在 $S_t^{(2)}$ 和 $S_{t-1}^{(2)}$ 已知的情况下，二次 ES 法的预测模型为：

$$\begin{aligned} \widehat{y^{t+l}} &= a_t + b_t l \\ a_t &= 2S_t^{(1)} + (1 - \alpha) S_{t-1}^{(2)} \\ b_t &= \frac{\alpha}{1 - \alpha} (S_t^{(1)} - S_{t-1}^{(2)}) \end{aligned}$$

式中， $\widehat{y^{t+l}}$ 表示二次指数平滑法在 $t + l$ 时刻的预测值； l 是预测向后延伸的时期数； a_t 对应该线性预测函数的截距， b_t 则对应斜率。

三次指数平滑法

二次指数平滑法适用于有线性趋势的时间序列。然而，当序列值呈非线性趋势改变时，二次指数平滑的预测效果将有一定误差。因此，研究者引入了三次指数平滑法。顾名思义，三次平滑法是在二次平滑法的基础上的再一次平滑。

三次指数平滑法的计算公式为：

$$\begin{aligned} S_t^{(1)} &= \alpha y^t + (1 - \alpha) S_{t-1}^{(1)} \\ S_t^{(2)} &= \alpha S_t^{(1)} + (1 - \alpha) S_{t-1}^{(2)} \\ S_t^{(3)} &= \alpha S_t^{(2)} + (1 - \alpha) S_{t-1}^{(3)} \end{aligned}$$

式中， $S_t^{(1)}$ ， $S_t^{(2)}$ ， $S_t^{(3)}$ 分别表示 t 时刻的一次、二次、三次指数平滑值。在这三个值已知的情况下，三次 ES 模型为：

$$\widehat{y^{t+l}} = a_t + b_t l + c_t l^2$$

其中，

$$\begin{aligned} a_t &= 3S_t^{(1)} - 3S_t^{(2)} + S_t^{(3)} \\ b_t &= \frac{a}{2(1-a)^2} [(6 - 5a)S_t^{(1)} - 2(5 - 4a)S_t^{(2)} + (4 - 3a)S_t^{(3)}] \\ c_t &= \frac{a^2}{2(1-a)^2} (S_t^{(1)} - 2S_t^{(2)} + S_t^{(3)}) \end{aligned}$$

统计某个无线电集电器在一段时间周期内传输的数据微片数目，并对该序列分别进行一次、二次、三次指数平滑，结果如下表和下图所示。此处平滑系数 α 取值为 0.3。一个周期 (epoch) 代表 80 个系统时间周期 (cycle)。

表 3.1: 指数平滑法预测结果

	1	2	3	4	5	6	7	8	9
实际值	8	8	0	0	0	8	0	0	0
一次平滑预测值	5.33	6.13	6.69	4.68	3.27	2.29	4.01	2.80	1.96
二次平滑预测值	5.33	6.93	7.81	3.46	1.02	-0.27	3.92	1.5	0.24
三次平滑预测值	5.33	7.73	8.69	1.73	-1.23	-2.15	5.08	1.18	-0.47

可以看到，由于网络中流量的不均匀性，在某个仿真周期 (epoch) 内传输的数据量可能会突然增加。该时间序列的变化趋势性不明显，且没有线性趋势。因此，采用三次指数平滑法进行预测的结果较好，从图中也可以看出。

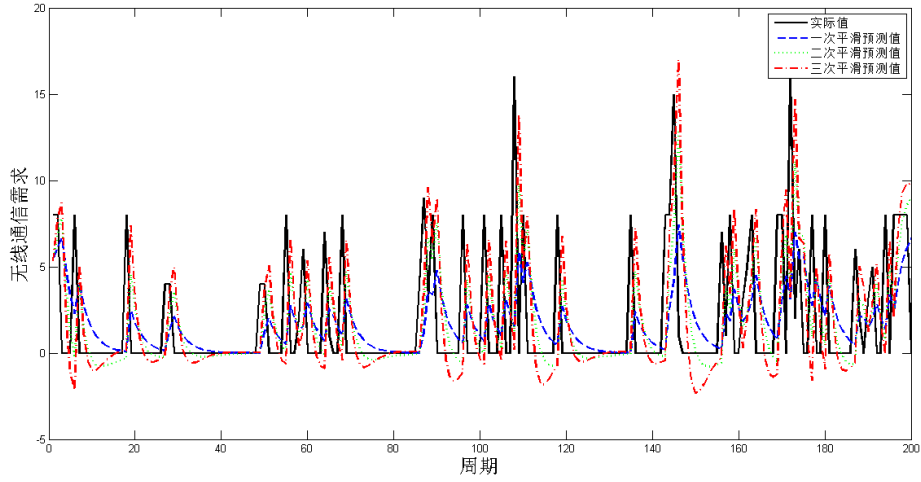


图 3.4: 指数平滑结果示意图

3.2.3 基于指数平滑法的动态 MAC 机制

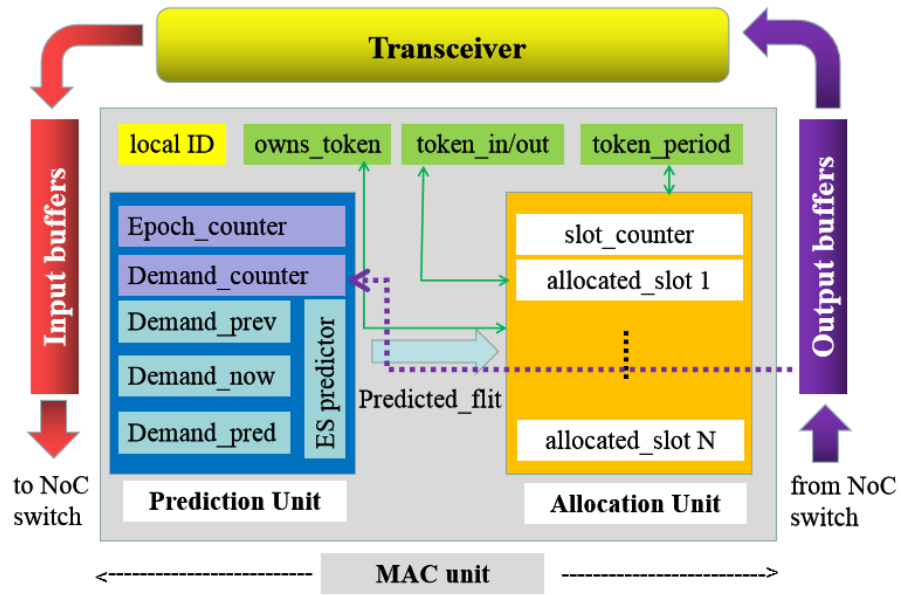


图 3.5: 动态 MAC 机制示意图

综上所述，可以采用基于指数平滑法的预测算法对无线电集电器持有令牌的时隙进行动态分配，如图 3.5 所示。

首先，定义令牌在环中循环一周的时间为 $token_period(TP)$ 。对于 N 个 radio hub 的 WiNoC, $token_period = N * HC$ ，其中 HC 为外界设定的令牌持有时间，单位是仿真时钟周期数；

其次，根据每个 radio hub 在目前和之前 $token_period$ 中传输数据的多少来预

测下一 TP 中 hub 的传输需求，用来代表其对无线信道 (wireless channel) 的需求。
预测算法采用指数平滑法。

$$predicted_flit = ES_predictor(Demand_prev, Demand_now)$$

最后，根据预测值，在下一个 TP 内对每个 radio hub 持有令牌的时隙按比例分配：

$$allocated_slot[i] = \frac{predicted_flit[i]}{\sum_{i=1}^N predicted_flit[i]} * token_period$$

其中, i 表示某个特定的 radio hub 的序号， N 为网络中 radio hub 的总数。

Algorithm 2 基于无线通信需求的动态 MAC 机制

Require: $request, token_in, token_period, predicted_flit$

Ensure: $grant, token_out$

```

if  $request$  and  $owns\_token$  then
     $grant \leftarrow true$ 
else
     $grant \leftarrow false$ 
end if
if  $owns\_token$  then
     $hold\_count \leftarrow hold\_count + 1$ 
    if  $hold\_count > Hold\_Cycle$  then
         $token\_out \leftarrow token\_in$ 
         $owns\_token \leftarrow false$ 
    end if
else
    if  $token\_in$  then
         $owns\_token \leftarrow true$ 
         $hold\_count \leftarrow 0$ 
         $Hold\_Cycle \leftarrow \frac{predicted\_flit[i]}{\sum_{i=1}^N predicted\_flit[i]} * token\_period$ 
    end if
end if

```

因此，动态 MAC 机制为每个 radio hub 配备了一个计数模块与一个预测模块，即图 3.5 中的 prediction unit。预测结果（即图中的 $predicted_flit$ ）传送到令牌环的决策部分（即图中的 allocation unit）。 $predicted_flit$ 的值代表下一个 $token_period$

中该 radio hub 预测将传输的数据微片数。算法 2 总结了该动态 MAC 机制的基本思路。与传统基于时分复用的 MAC 机制相比，动态 MAC 机制的优点在于可以感知并预测网络流量的变化趋势并通过设定 *Hold_Cycle* 值，使得 radio hub 能够更有效地利用无线信道的进行数据传输。

3.2.4 动态切换 MAC 机制

如前所述，*predicted_flit* 的值代表下一个 *Token_period* 中该 radio hub 预测将传输的数据微片数。当该预测值较小时，代表网络处于低负载状态。此时，多余的令牌持有时间对每个 radio hub 来说是一种冗余与浪费。此时，采用前文所述的 *Token_packet* MAC 机制进行数据传输更为有效，因为该算法中不需要持有令牌达到固定的时间，只要数据传输完成后即可释放。在低网络负载情况下，每个 radio hub 进行数据传输的时间很短。因此，可以插入一个人为设定的阈值 *threshold*，当 *predicted_flit* 值低于该阈值时，可以判断网络处于低负载状态，此时将 MAC 机制切换为 *Token_packet*。否则，网络处于高负载状态，采用前文提出的动态 MAC 机制 *Dynamic_token_hold*，如算法 3 所示。

Algorithm 3 动态切换 MAC 机制算法

Require: *predicted_flit*, *N*

Ensure: *MAC_policy*

if $\sum_{i=1}^N predicted_flit[i] < threshold$ **then**
 $MAC_policy \leftarrow Token_packet$

else

$MAC_policy \leftarrow Dynamic_token_hold$

end if

第 4 章 WiNoC 性能仿真

4.1 仿真平台介绍

Noxim 仿真器是使用 SystemC 开发的一款开源 NoC 仿真平台^[39]。SystemC 是用 C++ 编写的系统描述库。采用这种语言进行建模符合 NoC 的内在特性：可扩展性高，即可以在不大范围改动源代码的情况下将新的设计方法加入到原有的仿真平台，同时，该仿真平台支持周期精确的模拟。片上网络架构的配置实例化了两个主要的元素：多个瓦片节点 (tile nodes) 和通信基础设施。每个区块节点实现一些计算/存储任务，通过通信基础设施与其他节点异步地交换数据。该体系结构的实例完全由 NoC 配置决定。研究者可以定制 NoC 的若干参数，这些参数会影响 NoC 的性能并且决定执行仿真所需的时间。

具体来说。Noxim 仿真器支持人工配置多个网络参数，这些参数可以分为以下类：拓扑结构、互联方式、工作负载、动态行为和仿真要求。

- 拓扑结构：即实例化节点的总数以及它们之间的互连类型。Noxim 将节点分为两类，即 tile 节点和 hub 节点。tile 节点中，存在 NoC 的计算/存储节点，而在 hub 节点中，存在用于对用于对区块进行分组和缩短网络的远程区域的一种“网关”进行建模的节点。

需要注意的是，Noxim 仿真中支持的网络拓扑结构为传统的网格 (mesh) 型。

- 互联方式：可以实例化三种不同类型的互连：(a)Tile-Tile：两个瓦片节点之间的有线点对点连接；(b)Tile-Hub：瓦片和集线器元件之间的有线点对点连接；(c)Hub-Hub：两个集线器之间的连接。需要注意的是，在 Hub-Hub 互连中包含无线通信机制，并使用信道概念进行建模。
- 工作负载：工作负载类参数指的是会影响 NoC 输入的计算方式的参数。这包括输入网络的数据将以怎样的方式注入到网络中，即所谓的“数据流量模型” (traffic pattern)。Noxim 提供了几种常用的数据流量模型，这些模型抽象了典型的通信模式^[33]。此外，Noxim 可以仿真将任务通信图映射到基于表 (traffic table) 的自定义流量的实际应用程序。外加参数可以确定 NoC 组件在

仿真期间执行决策的方式。这可能包括不同网络路径之间的数据包路由相关的决策，无线或有线通信之间的选择等。

- 动态行为：动态行为类参数指的是影响仿真时决策的参数，例如路由算法。常用的路由算法 [34][35][36] 已经集成在 Noxim 源代码中。研究者还可以采用类似插件的方式增加尝试任何一种新的路由算法。
- 仿真要求：可以指定仿真总时间 (simulation time), 预热时间 (warm up time), 重置时间 (reset time)。其中，设定预热时间的意义在于得到更加稳定的网络性能数据。因为在数据刚开始注入时，网络中没有多余负载，数据传输非常顺畅，同时延迟很低，因为前面没有多余的数据在排队。这可以类比与清晨的城市交通状态。然而，随着时间的推移，网络的平均延时和吞吐量将会降低，可以类比于进入“早高峰”时期的城市交通。因此，过度关注处于良好状态下的网络数据意义不大，因为这不是网络的正常工作状态。于是，可以将这部分时间预留为“预热时间” (warm up time), 在进行最终的数据统计 (延时，吞吐量，网络收发数据包总量) 时去除这部分数据，从而可以减小统计误差，得到更具有一般性的网络性能数据。

综上所述，研究者可以配置的网络参数如表 4.1 所示。

表 4.1: Noxim 可配置的网络参数

有线网络拓扑	mesh_dim_x,mesh_dim_y,buffer_depth,flit_size,link_length
无线网络拓扑	toTileBufferSize,fromTileBufferSize,rxBufferSize,txBufferSize
互联方式	Hub_attached_Nodes
路由算法	XY,West_first,North_last,Negative_first Odd_even,Table_based
流控制策略	Random,Buffer_level,NOP
无线信道访问控制协议	[Token_packet]、[Token_hold,num_hold_cycles]
仿真参数	clock_period、reset_time、simulation_time、warm_up_time max_volume_to_be_drained、show_buffer_stats
工作负载	min_packet_size、max_packet_size packet_injection_rate、probability_of_retransmission
流量模式	Random、Transpose1、Transpose2、Hotspot Table_based、Bit_Reversal、Shuffle、Butterfly

4.2 仿真参数说明

在 Noxim 运行时, 仿真器 (Noxim Runtime Engine,NRE) 对用户确定的 NoC 配置实例进行仿真模拟, 如图 4.1 所示。需要注意的是, Noxim 需要在 Linux 环境下运行, 研究者可以通过命令行调用不同的 NoC 配置文件, 如图中的 256_8h 文件即表示该网络结构为 16*16 的网格, 且放入了 8 个无线电集电器。在导入这个配置文件后, 用户依然可以通过命令行的方式重新指定一些配置参数, 例如数据包注入率等。通过命令行输入的参数值会覆盖掉原有配置文件中的参数值。在仿真结束后, 用户可以得到多个执行统计信息, 包括性能(延迟, 吞吐量)和能耗相关数据。其中, 延迟(delay)分为 average_delay 与 max_delay, 吞吐量为 throughput。可

```
Loading configuration from file "../config_examples/256_8h.yaml"... Done
Loading power configurations from file "power.yaml"... Done
Reset for 1000 cycles... done!
Now running for 10000 cycles...
Noxim simulation completed. (11000 cycles executed)

% Total received packets: 1176
% Total received flits: 14112
% Received/Ideal flits Ratio: 0.945216
% Average wireless utilization: 0.0382653
% Global average delay (cycles): 27.1616
% Max delay (cycles): 300
% Network throughput (flits/cycle): 1.568
% Average IP throughput (flits/cycle/IP): 0.006125
% Total energy (J): 3.87464e-05
%     Dynamic energy (J): 1.07958e-06
%     Static energy (J): 3.76668e-05
```

图 4.1: Noxim 仿真结果示意图

以通过三个参数来评估片上网络的性能: 带宽, 吞吐量和延迟。

- 带宽是指网络中的最大数据传播速率。带宽的度量单位是比特/秒, 它通常考虑整个数据包, 包括报头, 有效负载和尾部的比特。
- 吞吐量 (throughput) 由 Duato 等人定义^[28] 作为网络接受的最大流量, 即每个时间单位传递的最大信息量。吞吐量的度量是每秒消息数或每个时钟周期的消息数。通过将其除以消息的大小和网络的大小, 可以得到标准化的吞吐量。归一化吞吐量与网络的具体大小无关。归一化吞吐量的单位是每个时钟周期 (或每秒) 每个节点传输的比特数/微片数。

$$\text{归一化吞吐量} = \frac{\text{所有网络节点接收到的微片总数}}{\text{IP 节点的个数} * \text{总时钟周期数}}$$

- 延迟 (delay) 是指数据从传输开始到其在目标节点处被完全接收之间的时间间隔。延迟以时间单位测量，如若干个仿真时钟周期。通常，研究者使用平均延迟 (average delay) 而不是单个数据包的延迟来评估网络性能 [28]。

$$\text{平均延时} = \frac{\sum_{i=1}^N \text{数据包延时}}{N}$$

其中 N 为接收到的数据包总数。

- 无线利用率 (Average wireless utilization) 是指通过无线传输的数据量占整个网络传输的数据量的比例。加入无线节点后，数据从源节点到目的节点之间的路径有了更多选择，而在何时选择走无线信道也是路由层面的研究热点。文献 [30] 通过对比有线传输和无线传输的跳跃数来进行选择，还同时考虑了无线节点的缓冲区占用率。

$$\text{无线利用率} = \frac{\text{通过无线传输的数据包总数}}{N}$$

其中 N 为接收到的数据包总数。

通常，研究人员通过改变网络中的数据包注入率，收集不同注入率情况下的网络平均延时/吞吐率等信息，来衡量片上网络的性能。

数据包注入率指单个节点在一个时钟周期内注入到 NoC 中的数据包数量，即

$$\text{数据包注入率} = \frac{1}{\text{两次发送数据包的时钟周期间隔}}$$

当数据包注入率较低时，注入的数据包可以在低拥塞的情况下传输到接收端，网络平均延时较低且保持稳定。当注入率逐渐增大时，网络接收数据包的速率逐渐小于发送数据包的速率，此时会出现网络拥塞，平均延时也会增加。当平均延时急剧增大时，我们称网络处于饱和点。在网络饱和后，再增加注入率将无法使网络吞吐量提高，因为大量的数据包拥塞在网络中无法顺畅传输，接收到的数据包总数与发送的数据包总数的比例将急剧降低。

4.3 仿真结果

4.3.1 不同流量模型下仿真结果

NoC 流量模式可以描述网络中消息的空间分布。片上网络中消息的空间分布可以用流量矩阵 (traffic matrix) λ 表示，其中每个矩阵元素 λ_{sd} 给出从节点 s 发送

到节点 d 的流量的比例。下表列出了用于评估互连网络性能的一些常见静态流量模式。这些流量模式的产生基于一些历史上曾出现的特定应用。例如，矩阵转置 (transpose) 或转角 (corner-turn) 操作对应转置流量模式 (traffic transpose)，而快速傅里叶变换 (FFT) 或分类应用可能会导致混洗流量模式 (shuffle traffic)，流体动力学中通常会表现出相邻模式 (neighbor traffic)。

NoC 流量模式

名称	流量模式的数学建模
<i>Random</i>	$\lambda_{sd} = 1/N$
<i>Bitcomplement</i>	$d_i = -s_i$
<i>Bitreverse</i>	$d_i = s_{b-i-1}$
<i>Bitrotation</i>	$d_i = s_{i+1} \bmod b$
<i>Shuffle</i>	$d_i = s_{i-1} \bmod b$
<i>Transpose</i>	$d_i = s_{i+b/2} \bmod b$
<i>Tornado</i>	$d_x = (s_x + (\lceil \frac{k}{2} \rceil - 1)) \bmod k$
<i>Neighbor</i>	$d_x = (s_x + 1) \bmod k$

随机 (Random, 均匀) 流量模式中，每个源节点发送数据到每个目的节点的可能性完全相同。这是网络评估中最常用的流量模式。随机流量对于网络来说非常友好的，因为通过使流量均匀分布，即使拓扑和路由算法通常具有非常差的负载平衡，网络也可以达到负载平衡。因此，当仅使用随机流量进行网络性能评估时，一些非常糟糕的拓扑和路由算法看起来也会有非常好的结果。

为了强调拓扑或路由算法对网络性能的影响，我们通常使用置换流量 (permutation traffic) 模型对网络性能进行评估。在置换流量模型中，每个源将其所有流量发送到单个目的地。目的地址与源地址通过特定数学模型进行计算得出。置换流量模型的流量矩阵是置换矩阵 (permutation matrix)。因为它们将负载集中在各个源-目标对上，置换流量模型会强调拓扑和路由算法的负载平衡。比特排列 (Bit permutations) 通过置换和选择性地补充源地址的比特来计算目的地地址。例如，如果四位源地址是 (s_3, s_2, s_1, s_0) ，则位反转流量模式的目标是 (s_0, s_1, s_2, s_3) ，对于位补码流量模式，目标是 $(-s_3, -s_2, -s_1, -s_0)$ ，对于混洗流量模式，目的地是 (s_2, s_1, s_0, s_3) 。转置流量模型 (traffic transpose) 分为两种，即 transpose1 与 transpose2。

- transpose1: 节点 (i, j) 的数据发送到节点 (j, i) ，源节点与目的节点关于直线 $y=x$ 对称。

- transpose2: 节点 (i,j) 的数据发送到节点 $(\max(j)-j,\max(i)-i)$, 源节点与目的节点关于直线 $y=-x$ 对称

在不同流量模式下对传统基于令牌传递法的 MAC 机制与改进后的动态 MAC 机制进行仿真, 网络其余配置如表 4.2 所示, 仿真实验结果如图 4.2 - 4.5 所示。

表 4.2: 仿真实验参数

拓扑结构	mesh(16*16),8 hubs
路由算法	XY
流控制策略	RANDOM
无线信道访问控制协议	Traditional_token_hold,Dynamic_token_hold(hold_cycles=10)
仿真周期/预热周期	10000/1000
数据包大小	12 (微片)
微片大小	64 (比特)
端口缓存空间大小	4 (微片)
流量模式	Random、Transpose1、Transpose2、Butterfly
包注入的时间分布	泊松分布

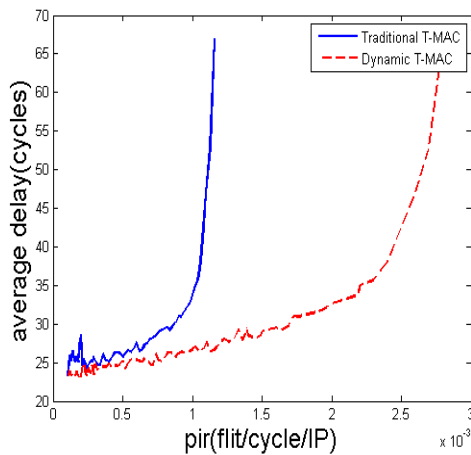


图 4.2: 均匀流量下的平均延时-注入率曲线

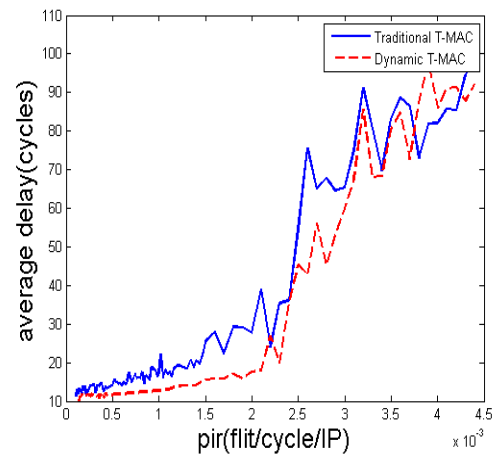


图 4.3: 蝶形流量下的平均延时-注入率曲线

由图可以看出, 采用改进后的动态 MAC 机制后, 网络进入饱和点的注入率增大。例如, 在转置流量 1 情况下, 传统 MAC 机制在数据包注入率为 1.3 flit/cycle/IP 时即进入饱和状态, 但改进后的 MAC 机制在数据包注入率为 2.7 flit/cycle/IP 时才进入饱和状态。此外, 在未进入饱和状态之前, 动态 MAC 机制的网络平均延迟也更低, 这是由于在网络负载较低的情况下, 每个无线电集电器不需要花费所有的令牌时隙 (在本仿真情况下, hold_cycle=10 个时钟周期) 才能传输完毕数据。在动态 MAC 的情况下, 一旦传输完成, 集电器就会释放令牌给下一个集电器, 而不

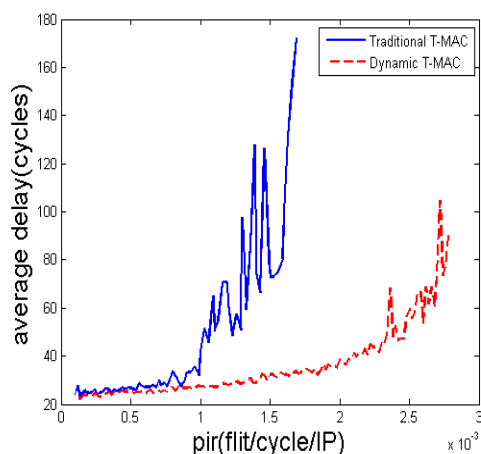


图 4.4: 转置流量 1 下的平均延时-注入率曲线

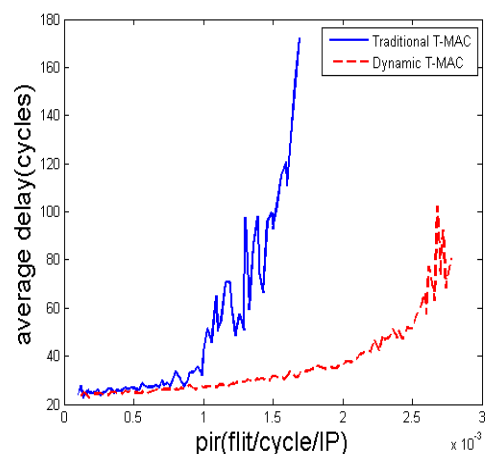


图 4.5: 转置流量 2 下的平均延时-注入率曲线

必等待固定的时间。这有利于减少网络延迟，提高无线信道利用率。

4.3.2 不同无线拓扑下仿真结果

改变无线网络拓扑结构，即在 16*16 的网格结构中放置 4 个，8 个或 16 个无线电集电器，对传统基于令牌传递法的 MAC 机制与改进后的动态 MAC 机制进行仿真，网络其余配置如表 4.3 所示。

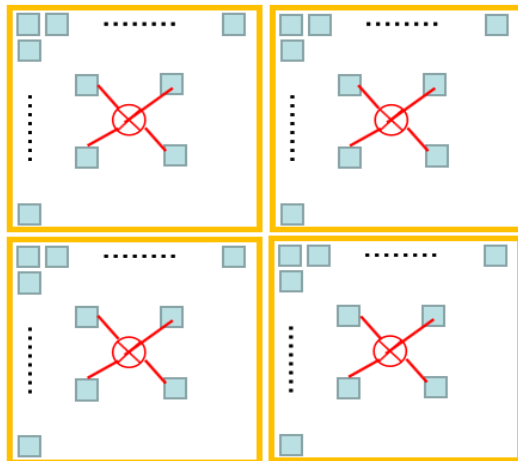


图 4.6: 256_4 hubs 拓扑示意图

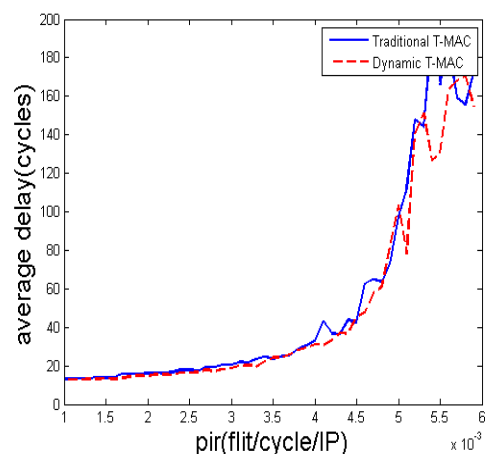


图 4.7: 256_4 hubs 下的平均延时-注入率曲线

表 4.3: 仿真实验参数

拓扑结构	mesh(16*16),4 hubs,8 hubs 16 hubs
路由算法	XY
流控制策略	RANDOM
无线信道访问控制协议	Traditional_TOKEN_HOLD,Dynamic_TOKEN_HOLD(hold_cycles=10)
仿真周期/预热周期	10000/1000
数据包大小	12 (微片)

微片大小	64 (比特)
端口缓存空间大小	4 (微片)
流量模式	BUTTERFLY
包注入的时间分布	泊松分布

图 4.6 展示了本实验中如何在网格中放置无线电集电器。对于 256_4 hubs 的情况，首先将网络分割成 4 个小网络，即图中的 4 个黄色网络。其次，在每个小网络的正中央 4 个节点的中间放置一个无线电集电器，即集电器通过图中所示的红色线条（代表网络中的有线连接）连接到 4 个瓦片节点 (tile node)。每个瓦片节点都有固定的编号，即 0-255。通过指定编号即可完成连接。4 个无线节点之间通过无线信道进行数据传输，可以指定无线信道的数据传输速率，即带宽。在本实验中，带宽被设定为 16Gbps。同理，对于 256_8 hubs 与 256_16 hubs 的情况，也是首先将网络分成 8 个或 16 个小网络，再在小网络的中间位置放置无线电集电器。当然，通过改变无线电集电器连接的瓦片节点的编号，也可以实现 hubs 的不规则放置。

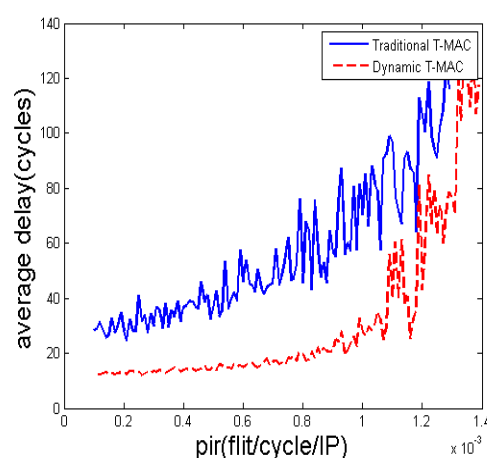
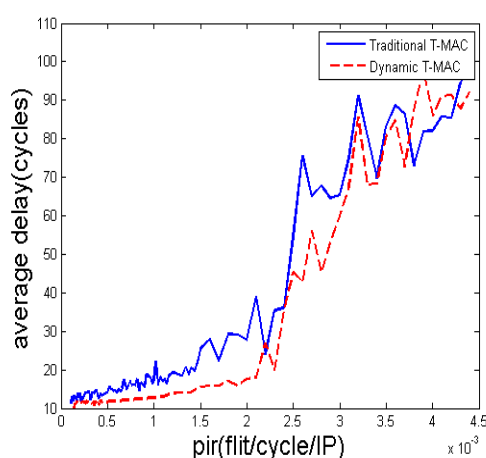


图 4.8: 256_8 hubs 下的平均延时-注入率曲线 图 4.9: 256_16 hubs 下的平均延时-注入率曲线

由图可以看出，随着放置无线节点数量的增多，改进后的动态 MAC 机制对网络性能的影响更加明显。这是由于 Noxim 中默认使用无线传输的情况当且仅当目标节点与无线电集电器直接相连。因此，在网络中放置更多无线电集电器的情况下，使用无线信道的比例会随之增加，而对无线 MAC 协议所做的改进对网络整体性能的提升也会更加明显。在只放置 4 个 radio hubs 的情况下，无线利用率 (wireless utilization) 的值约为 3% - 5%；在放置 8 个 radio hubs 的情况下，无线利用

率约为 8% - 15%; 在放置 16 个 radio hubs 的情况下, 无线利用率约为 15% - 25% 。

4.3.3 不同路由算法下仿真结果

改变路由算法, 对传统基于令牌传递法的 MAC 机制与改进后的动态 MAC 机制进行仿真, 网络其余配置如表 4.3 所示。本实验采用了两种常用的路由算法: XY 路由与 WF 路由。

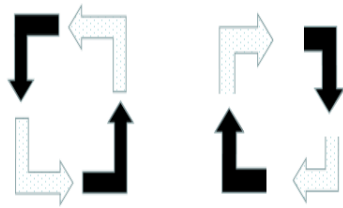


图 4.10: XY 路由转弯可能性

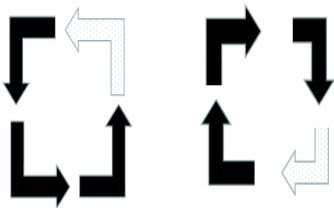


图 4.11: WF(West First) 路由转弯可能性

- XY 路由: 数据包先从 X 方向传递到达目的节点的横坐标所在位置, 再从 Y 方向传递到达目的节点。即给定源节点与目的节点的坐标, 每次运行过程中数据包选择经过的路径不变。如图 4.10 所示, XY 路由中 4 个方向的转弯是不允许的, 即图中的浅色部分。
- WF 路由: West First(WF) 路由算法不允许 south-west 方向的转弯, 如图 4.11 浅色箭头所示。因此, 在传输开始时, 数据包就要考虑是否向西传递, 这也是 west first 的来源。根据目的节点的横坐标在源节点的西边还是东边, 数据包可能进行确定性或者适应性的传输。

表 4.4: 仿真实验参数

拓扑结构	mesh(16*16),16 hubs
路由算法	XY,West First
流控制策略	RANDOM
无线信道访问控制协议	Traditional_TOKEN_HOLD,Dynamic_TOKEN_HOLD(hold_cycles=10)
仿真周期/预热周期	10000/1000
数据包大小	12 (微片)
微片大小	64 (比特)
端口缓存空间大小	4 (微片)
流量模式	BUTTERFLY
包注入的时间分布	泊松分布

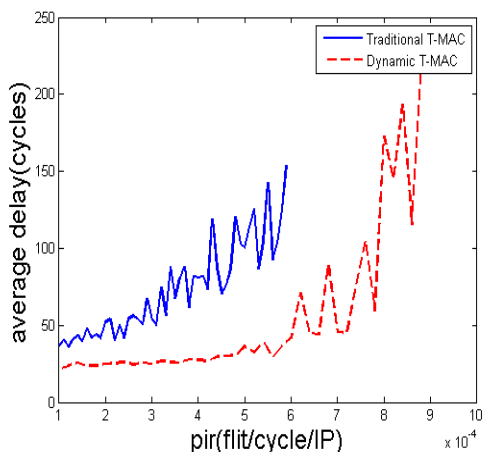


图 4.12: XY 路由下的平均延时-注入率曲线

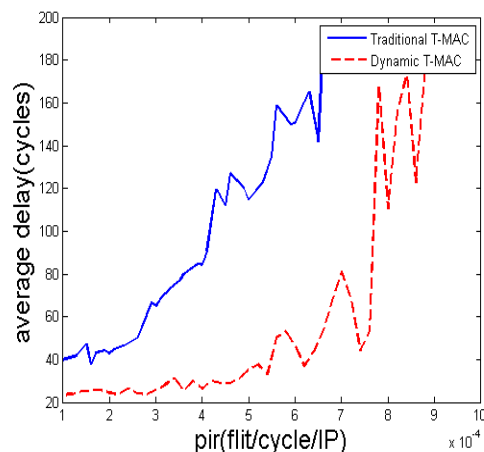


图 4.13: WF 路由下的平均延时-注入率曲线

由图可以看出，在不同的路由算法下，采用改进后的动态 MAC 机制后，网络进入饱和点的注入率增大，在进入饱和区之前的平均延迟更低。

4.3.4 不同令牌持有时间下仿真结果

根据第三章所述的基于令牌传递法的 MAC 机制，每个无线电集电器持有令牌的时隙长短需要人为设定。在本实验中，采用令牌持有时间为 10 个时钟周期与 20 个时钟周期分别进行实验。网络其余参数配置如表 4.5 所示。

表 4.5: 仿真实验参数

拓扑结构	mesh(16*16), 16 hubs
路由算法	XY
流控制策略	RANDOM
无线信道访问控制协议	Traditional_TOKEN_HOLD,Dynamic_TOKEN_HOLD(hold_cycles=10 或 20)
仿真周期/预热周期	10000/1000
数据包大小	12 (微片)
微片大小	64 (比特)
端口缓存空间大小	4 (微片)
流量模式	BUTTERFLY
包注入的时间分布	泊松分布

由图可以看出，随着令牌持有时间的增长，改进后的动态 MAC 机制对网络性能的影响更加明显。这是由于在较低网络负载时，外界给定的 hold_cycle 越长，无线电集电器等待令牌的时间越长，网络延迟也就更大。例如，在传统 MAC 机制下，hold_cycle=20 时，对于 16 个无线电集电器，某个特定的集电器需要等待 $20 * 15 = 300$ 个时钟周期才能再次拿到令牌进行数据传输。而改进后的 MAC 进制

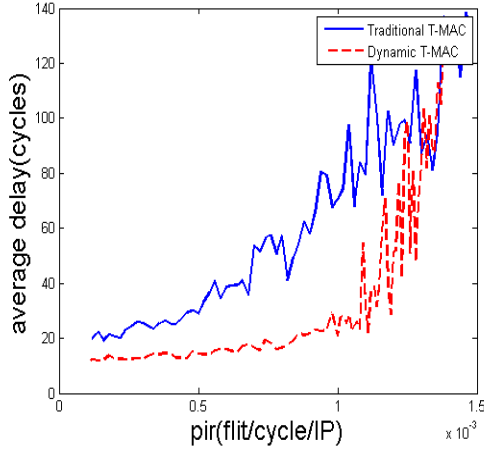


图 4.14: Hold10 时的平均延时-注入率曲线

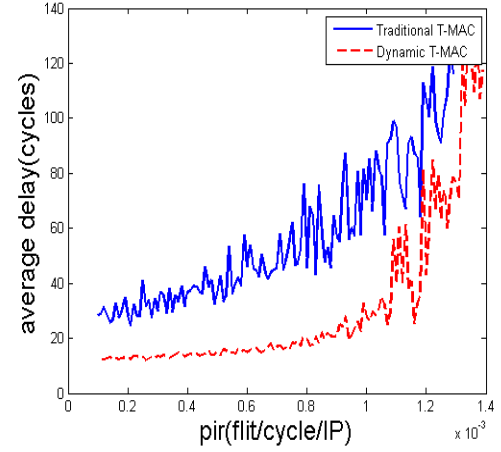


图 4.15: Hold20 时的平均延时-注入率曲线

减少了等待时间，因此网络性能的提升也就更加明显。在较高网络负载时，改进后 MAC 进制通过对令牌持有时隙的按需分配，也能够提高无线信道利用率，从而减小网络平均延迟。

4.3.5 模拟实际应用仿真结果

在仿真中，研究者可以人为设定数据包注入率的值，在整个仿真期间这个值都是固定的；然而，在实际应用中，数据包注入率不是稳恒不变的，可以类比于城市中的街道，在不同时间段的车辆状况不同，从而导致道路拥塞情况的变化。因此，为了模拟实际应用，可以对数据包注入率增加一个随机变量，在这种情况下再次衡量 NoC 的性能。

Algorithm 4 模拟实际应用的数据包注入算法

Require: $packe_injection_rate, fluct_pir$

Ensure: $threshold, shot$

if $fluct_pir$ **then**

$threshold \leftarrow packet_injection_rate + A * ((double)rand() / RAND_MAX) - B$

else

$threshold \leftarrow packet_injection_rate$

end if

$shot \leftarrow (((double)rand()) / RAND_MAX < threshold)$

if $shot$ **then**

$start\ transmission\ based\ on\ traffic\ pattern$

end if

如算法 3 所示,除了外界给定的数据包注入率 (*packe_injection_rate*),在网络配置中还集成了一个另外的开关 *fluct_pir* 用来控制是否产生随机流量。*fluct_pir* 是一个布尔类型的值, **true** 表示产生随机流量, **false** 表示不产生。如果设定为真,则阈值 *threshold* 为数据包注入率再加上一个随机数。*shot* 也是一个布尔类型变量,它代表一个位于 (0,1) 之间随机数是否小于该阈值,一旦小于,就开始数据注入与传输。因此,当数据包注入率越大,阈值越大,随机数小于该阈值的可能性越高,数据开始传输的可能性也就越大。这符合人们对“数据包注入率”的感性认知。

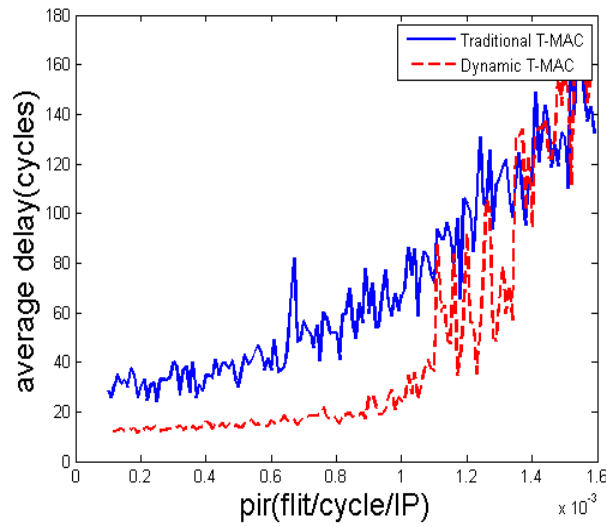


图 4.16: 注入率波动情况下的平均延时-注入率曲线

由图可以看出,在产生随机流量的情况下,改进后的动态 MAC 机制仍然可以提高网络的性能指标。

4.4 实验结论

在不同拓扑结构、流量模型、路由算法等情况下对提出的动态 MAC 机制进行仿真,发现它对网络整体性能的提升有积极影响:一方面,动态 MAC 机制推迟了网络进入饱和点的时间;另一方面,它也减少了网络的平均延迟。这得益于在每个 radio hub 中加入了一个计数模块与预测模块用来感知并预测其对无线通信的动态需求。根据预测结果,研究者可以通过人为设定阈值判断网络处于低负载状态还是高负载状态。在低负载状态下,采用 *token_packet* MAC 机制减少网络延迟;在高负载情况下,通过 *Dynamic_token_hold* MAC 机制提高无线信道利用率。

第 5 章 总结与展望

5.1 总结

无线互连是新兴的互连模式之一，可以解决传统 NoC 中因金属线互联而带来的高延迟与高能耗问题。无线片上网络的设计需要合适的无线介质访问协议。然而，由于任务映射，任务迁移，工作负载和异构组件的集成，无线电集电器的无线通信需求在空间和时间上都会有较大的变化。传统基于时分复用的 MAC 机制无法感知这种动态变化。本文提出了一种改进后的动态 MAC 机制，它能够根据每个无线电集电器现在和过去的通信情况对将来的通信需求进行预测，并根据预测结果动态切换 MAC 机制或动态分配令牌持有的时隙长短。在不同流量模式，不同网络拓扑，不同路由算法等情况下的仿真结果证明了该动态 MAC 机制对提升整个 WiNoC 网络性能的显著效果。

5.2 有待解决的问题及展望

经过算法的改进，动态 MAC 机制相比传统 MAC 机制已经有了更好的效果。然而，还存在一些待解决的问题。

5.2.1 指数平滑法中参数的选择

3.2.2 节中介绍了用于拟合和预测的指数平滑法。该算法的优点在于流程简单，并且没有复杂的数学运算，适合片上系统，不会带来太多的面积开销。然而，指数平滑法的预测效果与其中的一个参数 α 有关。本实验中，经过数据拟合，确定在 $\alpha = 0.3$ 左右时预测值与实际值的误差较小。但是，对于多种不同参数配置的 NoC 网络， α 的选择与 *token_period* 的长短，也即人为设置的 *hold_cycle* 与网络中总共的 radio hub 数量有关。因此，在不同实验环境下，需要首先通过拟合确定 α 的值为多少时误差较小，才能够进行下一步实验。

5.2.2 无线利用率问题

4.3.2 节中指出, 在无线利用率较高的情况下, 采用动态 MAC 机制对整体网络性能的提高才更加明显。仿真中, 数据采用无线信道传输的情况设定非常单一: 仅当目的节点与无线集电器直接相连时。而事实上, 如果目的节点与无线集电器虽然不直接相连但是距离很近, 也可以选择无线传输, 减小跳数与延迟。因此, 可以进一步改进选择有线或无线的仲裁策略, 例如对比二者需要的总跳数, 选择总跳数更少的路径; 或者考虑网络拥塞情况。由此, 可以进一步提高无线利用率, 从而体现出本文提出的动态 MAC 机制的优点。

5.2.3 令牌循环总时间的问题

3.2.3 节中描述了基于指数平滑法的动态 MAC 机制。该算法的不足在于, *token_period* 值是外界设定的参数, 在实验过程中只是对其进行比例分配, 但无法改变这个值的大小。这是由于每个无线集电器配备的计数器是用来统计 *token_period* 长短的时隙内其传输的数据微片数目, 并对下一个 *token_period* 内的传输情况进行预测。一旦 *token_period* 不是固定值, 预测结果也就相应没有了意义, 因为在更长的时隙中显然传输的数据微片数目也会变多。事实上, 鲜有针对此类“不等时间间隔序列”的预测算法。然而, *token_period* 是定值的缺点在于, 一旦外界设定的参数不合适, 例如在 256_16 hubs 的情况下设置 *hold_count* = 30, 即 $token_period = 30 * 16 = 480$, 显然令牌循环一周的时间过长。又例如设置 *hold_count* 的时间过短, 无论如何动态分配也无法满足高负载状况下的传输需求。因此, *token_period*, 即令牌循环总时间的动态调整是一个问题。

综上所述, 本实验仍然存在一些待完善的部分。受时间条件制约, 本研究尚未能够在这些领域进行深入研究。望今后的研究中或他人的研究中在这些领域取得进展。

参考文献

- [1] L. Benini and G. D. Micheli, Networks on chips: A new SoC paradigm, *IEEE Computer*, vol. 35, no. 1, pp. 70–78, Jan. 2002.
- [2] V. F. Pavlidis and E. G. Friedman, 3-D topologies for Network-on-Chip, *IEEE Trans. Very Large Scale (VLSI) Syst.*, vol. 15, no. 10, pp. 1081–1090, Oct. 2007.
- [3] A. Shacham et al., Photonic Network-on-Chip for future generations of chip multi-processors, *IEEE Trans. Computers*, vol. 57, no. 9, pp. 1246–1260, Sep. 2008.
- [4] M. F. Chang et al., CMP Network-on-Chip overlaid with multi-band RF-interconnect, in *Proc. IEEE Int. Symp. High-Performance Comput. Architecture (HPCA)*, Feb. 16–20, 2008.
- [5] K. Kempa et al., Carbon nanotubes as optical antennae, *Adv. Mater.*, vol. 19, pp. 421–426, 2007.
- [6] D. Zhao and Y. Wang, SD-MAC: Design and synthesis of a hardware-efficient collision-free QoS-aware MAC protocol for wireless Network-on-Chip, *IEEE Trans. Computers*, vol. 57, no. 9, pp. 1230–1245, Sep. 2008.
- [7] D. DiTomaso et al., iWise: Inter-router wireless scalable express channels for Network-on-Chips (NoCs) architecture, in *Proc. Annu. Symp. High Performance Interconnects*, 2011, pp. 11–18.
- [8] A. Ganguly et al., Complex network inspired fault-tolerant NoC architectures with wireless links, in *Proc. 5th ACM/IEEE Int. Symp. Networks-on-Chip*, 2011, pp. 1485–1502.
- [9] U. Y. Ogras and R. Marculescu, It's a small world after all: NoC performance optimization via long-range link insertion, *IEEE Trans. Very Large Scale (VLSI) Syst.*, vol. 14, no. 7, pp. 693–706, Jul. 2006.

- [10] J. Lin et al., Communication using antennas fabricated in silicon integrated circuits, *IEEE J. Solid-State Circuits*, vol. 42, no. 8, pp. 1678-1687, Aug. 2007.
- [11] X. Yu et al., A wideband body-enabled millimeter-wave transceiver for wireless Network-on-Chip, in *Proc. 54th IEEE Midwest Symp. Circuits Syst.*, Aug. 2011, pp. 1-4.
- [12] Vijayakumaran, V. et. al. CDMA Enabled Wireless Network-on-Chip. *ACM Journal on Emerging Tech and Comp Sys.*, 10(4). Article 28, 2014.
- [13] Chang, K., et. al. Performance evaluation and design trade-offs for wireless network-on-chip architectures. *ACM Journal of Emerg. Tech. in Comp. System*, 8 (3), Article 23, 2012.
- [14] Nguyen, C.K.; Kumar, A. An energy-aware medium-access-control protocol with frequent sleeps for wireless sensor networks. In *Proceedings of the IEEE Symposium on Computers and Communications*, Murcia, Spain, 27-30 June 2005; pp. 386-391.
- [15] Chiras, T.; Paterakis, M.; Koutsakis, P. Improved medium access control for wireless sensor networks-a study on the smac protocol. In *Proceedings of the IEEE Workshop on Local and Metropolitan Area Networks*, Crete, 18, September, 2005; pp. 1-5.
- [16] Liu, A.; Yu, H.; Li, L. An energy-efficiency and collision-free mac protocol for wireless sensor networks. In *Proceedings of the IEEE Vehicular Technology Conference*, Stockholm, Sweden, 30 May-1 June 2005; pp. 1317-1322.
- [17] Carloni, L.; Pande, P.; Xie, Y. Networks-on-chip in emerging interconnect paradigms: Advantages and challenges. In *Proceedings of the ACM/IEEE International Symposium on Networks-on-Chip*, San Diego, CA, USA, 10-13 May 2009; pp. 93-102.
- [18] Wang, C.; Hu, W.-H.; Bagherzadeh, N. A wireless network-on-chip design for multicore platforms. In *Proceedings of the 19th Euromicro International Conference on Parallel, Distributed and Network-Based Processing (PDP)*, Ayia Napa, Cyprus, Greece, 9-11 February 2011; pp. 409-416.

- [19] Kumar, A.; Peh, L.-S.; Jha, N.K. Token flow control. In Proceedings of the International Symposium on Microarchitecture, Lake Como, Italy, 8–12 November 2008, pp. 342–353.
- [20] Zhao, D.; Wang, Y.; Li, J.; Kikkawa, T. Design of multi-channel wireless NoC to improve on-chip communication capacity. In Proceedings of the IEEE/ACM International Symposium on Networks on Chip (NoCS), Pittsburgh, PA, 1–4 May 2011; pp. 177–184.
- [21] Deb, S.; Chang, K.; Yu, X.; Sah, S.P.; Cosic, M.; Ganguly, A.; Pande, P.P.; Belzer, B.; Heo, D. Design of an energy-efficient CMOS-compatible NoC architecture with millimeter-wave wireless interconnects. *IEEE Trans. Comput.* 2013, 62, 2382–2396.
- [22] Lee, S.-B.; Tam, S.-W.; Pefkianakis, I.; Lu, S.; Chang, M.F.; Guo, C.; Reinman, G.; Peng, C.; Naik, M.; Zhang, L.; et al. A scalable micro wireless interconnect structure for CMPs. In Proceedings of the 15th Annual International Conference on Mobile Computing and Networking, MobiCom '09, New York, NY, USA, 20–25 September 2009; pp. 217–228.
- [23] Kempa, K.; Rybczynski, J.; Huang, Z.; Gregorczyk, K.; Vidan, A.; Kimball, B.; Carlson, J.; Benham, G.; Wang, Y.; Herczynski, A.; et al. Carbon nanotubes as optical antennae. *Adv. Mater.* 2007, 19, 421–426, .
- [24] A. K. Mishra, N. Vijaykrishnan and C. R. Das, "A case for heterogeneous on-chip interconnects for CMPs," in *ACM 38th Annual International Symposium on Computer Architecture (ISCA '11)*, San Jose, CA, pp. 389–399, 2011.
- [25] M. Badr and N. E. Jerger, "SynFull: Synthetic traffic models capturing cache coherent behaviour," in *ACM/IEEE 41st International Symposium on Computer Architecture (ISCA '14)*, Minneapolis, MN, pp. 109–120, 2014.
- [26] G. V. Varatkar and R. Marculescu, "On-chip traffic modeling and synthesis for MPEG-2 video applications," in *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 12, no. 1, pp. 108–119, Jan. 2004.

- [27] Vijayakumaran, V. et. al. “CDMA Enabled Wireless Network-on-Chip” . ACM Journal on Emerging Tech and Comp Sys., 10(4). Article 28, 2014.
- [28] Duato J, Yalamanchili S, Ni LM (2003) Interconnection networks: an engineering approach. Morgan Kaufmann, Burlington
- [29] A Wireless Network-on-Chip Design for Multicore Platforms Chifeng Wang, Wen-Hsiang Hu, Nader Bagherzadeh Dept. of Electrical Engineering and Computer Science University of California, Irvine, 19th International Euromicro Conference on Parallel, Distributed and Network-Based Processing, 2011.
- [30] Brown, R. G. (1959). Statistical forecasting for inventory control. McGraw/Hill.
- [31] 负荷预测指数平滑法“厚近薄远”规律研究夏家盛，吉培荣 2019 电力学报第 34 卷第 1 期
- [32] W. J. Dally and B. Towles, Principles and Practices of Interconnection Networks. San Francisco, CA: Morgan Kaufmann, 2004.
- [33] J. Hu and R. Marculescu, “DyAD - smart routing for networks-on-chip,” in ACM/IEEE Design Automation Conference, San Diego, CA, USA, Jun. 7–11 2004, pp. 260–263.
- [34] G.-M. Chiu, “The odd-even turn model for adaptive routing,” IEEE Transactions on Parallel Distributed Systems, vol. 11, no. 7, pp. 729–738, 2000.
- [35] M. Palesi, R. Holsmark, S. Kumar, and V. Catania, “Application specific routing algorithms for networks on chip,” IEEE Transactions on Parallel and Distributed Systems, vol. 20, no. 3, pp. 316–330, Mar. 2009.
- [36] Christopher J. Glass and Lionel M. Ni. “The turn model for adaptive routing.” In Proc. of the International Symposium on Computer Architecture (ISCA), pages 278–287, 1992.

- [37] Chelcea T, Nowick SM (2001) Robust interfaces for mixed-timing systems with application to latency-insensitive protocols. In: Proceedings of the 38th design automation conference (DAC), Las Vegas, pp 21–26.
- [38] Catania, V., Mineo, A., Monteleone, S., Palesi, M., Patti, D. Improving energy efficiency in wireless network-on-chip architectures (2017) ACM Journal on Emerging Technologies in Computing Systems, 14 (1).

致谢

首先对我的指导老师李丽教授表示诚挚的感谢。在我毕业设计选题、研究、论文撰写的全过程中，李老师都给予我关心与指导。通过在每周组会上的报告，我能够及时汇报我的研究进展并得到反馈，一些疑难问题也可以与同学们进行讨论。

其次，我要向傅玉祥老师，宋文清博士表示感谢，他们拓宽了我在相关领域的视野，在每周都会抽出时间与我讨论实验中遇到的问题，并为我提供具体可行的解决方案。

同时，我要向高珺学姐、周禹辰学长、张永刚同学表示感谢，他们与我同属于片上网络研究的科研小组，我们常常会一起对上一阶段的研究进展情况进行讨论。相互之间就他人的情况提出问题和见解，相互启发，共同进步。

最后，感谢所有在 NoC 相关研究领域贡献智慧成果的研究者们，我的实验与研究成果立足于前人已有的基础。