Using Proportional-Integral-Differential approach for Dynamic Traffic Prediction in Wireless Network-on-Chip

Abhishek Vashist
av8911@rit.edu
Using Proportional-Integral-Differential approach for Dynamic Traffic Prediction in Wireless Network-on-Chip

Abhishek Vashist
Using Proportional-Integral-Differential approach for Dynamic Traffic Prediction in Wireless Network-on-Chip

Abhishek Vashist
August, 2017

A Thesis Submitted in Partial Fulfillment of the Requirements for the Degree of Master of Science in Electrical Engineering

Supervised by
Dr. Amlan Ganguly
Department of Computer Engineering
Kate Gleason College of Engineering
Rochester Institute of Technology
Rochester, New York
August 2017
Using Proportional-Integral-Differential approach for Dynamic Traffic Prediction in Wireless Network-on-Chip

Abhishek Vashist

Committee Approval:

Dr. Amlan Ganguly *Advisor*  
Associate Professor - R.I.T. Dept. of Computer Engineering

Dr. Jayanti Venkataraman  
Professor - R.I.T. Dept. of Electrical and Microelectronic Engineering

Dr. Panos P. Markopoulos  
Assistant Professor - R.I.T. Dept. of Electrical and Microelectronic Engineering

Dr. Sohail Dianat  
Department Head - Professor - R.I.T. Dept. of Electrical and Microelectronic Engineering
Acknowledgments

I would like to express my sincere appreciation and gratitude for my advisor Dr. Amlan Ganguly for his constructive remarks, optimal guidance, devoted time, and contributions throughout the duration my research and for the completion of this work. I would also like to thank Dr. Jayanti Venkataraman and Dr. Panos P. Markopoulos for being as thesis committee members and for their time, effort in supporting my research, and providing pointers to further improve my work.

I would like to thank my friends who are always by my side and being constant source of support towards achieving my goals.

Finally, I am grateful to the members of the multi-core systems lab at Rochester Institute of Technology especially Naseef Mansoor for his encouragement and support during the work on this thesis.
To my beloved parents and my sister without whom, my dreams of obtaining my Master’s degree would not have came into a reality.
Abstract

The massive integration of cores in multi-core system has enabled chip designer to design systems while meeting the power-performance demands of the applications. Wireless interconnection has emerged as an energy efficient solution to the challenges of multi-hop communication over the wireline paths in conventional Networks-on-Chips (NoCs). However, to ensure the full benefits of this novel interconnect technology, design of simple, fair and efficient Medium Access Control (MAC) mechanism to grant access to the on-chip wireless communication channel is needed. Moreover, to adapt to the varying traffic demands from the applications running on a multicore environment, MAC mechanisms should dynamically adjust the transmission slots of the wireless interfaces (WIs). To ensure an efficient utilization of the wireless medium in a Wireless NoC (WiNoC), in this work we present the design of prediction model that is used by two dynamic MAC mechanism to predict the traffic demand of the WIs and respond accordingly by adjusting transmission slots of the WIs. Through system level simulations, we show that the traffic aware MAC mechanisms are more energy efficient as well as capable of sustaining higher data bandwidth in WiNoCs.
# Contents

Signature Sheet ........................................... i
Acknowledgments ........................................... ii
Dedication .................................................. iii
Abstract .................................................... iv
Table of Contents .......................................... v
List of Figures .............................................. vii
List of Tables .............................................. ix

1 Introduction .............................................. 1
   1.1 Motivation .......................................... 1
   1.2 Thesis Contribution ................................ 2
   1.3 Organization of The Thesis ......................... 3

2 Background Information and Related Works .......... 4
   2.1 Multi-core System On-Chip (SoC) ................... 4
      2.1.1 Network-on-Chip (NoC) ........................ 6
      2.1.2 Network-on-Chip (NoC) Topology ............. 8
      2.1.3 Emerging Interconnects ........................ 9
      2.1.4 Wireless Network-on-Chip (WiNoC) .......... 11
   2.2 Related Works ....................................... 12

3 The Proposed PID Prediction Model ................... 15
   3.1 Basic Assumptions and Notations .................. 15
   3.2 PID Prediction Model ................................ 16
      3.2.1 Analytical Model .............................. 17
      3.2.2 Cost Function ................................ 17
      3.2.3 LMS Algorithm ............................... 17
   3.3 Evaluation of PID Prediction Model ............... 18
      3.3.1 Optimization of PID weights ................. 19
List of Figures

2.1 Transistors counts against dates of introduction in accordance with Moore’s law [1]. The number of transistors in an integrated circuit is doubling every eighteen months. .......................... 5
2.2 Typical Network-on-Chip NoC architecture with Mesh topology. . . . 7
2.3 Overview of Network-on-Chip NoC switch architecture. ................. 8
2.4 Network on-Chip ( NoC ) topology architectures [54] where (a) and (d) are examples of indirect topologies while (b), (c), (d) and (e) are examples of direct topologies. ................................. 9
2.5 Network-on-Chip NoC 3D Mesh based architecture [34]. .............. 11
3.1 Typical prediction model. .............................................. 16
3.2 Cost function $J(K)$ with respect to $K_I$ and $K_P$ .................. 19
3.3 Cost function $J(K)$ with respect to $K_D$ .......................... 20
4.1 Architecture of the WI with Dynamic MAC unit ..................... 22
4.2 64 core WiMesh architecture with 8 WIs .......................... 30
4.3 Peak Achievable Bandwidth per core for baseline WiMesh with varying subnet size ........................................... 31
4.4 Peak achievable bandwidth per core and packet energy for uniform random traffic .................................................... 33
4.5 Average packet latency with varying injection load for uniform random traffic ..................................................... 33
4.6 Peak achievable bandwidth per core and packet energy for hotspot traffic ..................................................... 35
4.7 Peak achievable bandwidth per core and packet energy for bit complement traffic .................................................... 35
4.8 Average packet latency with varying injection load for hotspot traffic ..................................................... 36
4.9 Average packet latency with varying injection load for bit complement traffic ..................................................... 37
4.10 Percentage gain in peak bandwidth per core and reduction in average packet energy with broadcast traffic ................... 39
4.11 Percentage reduction in average packet latency and average packet energy with application specific traffic ................... 41
4.12 Relative performance gain with varying flit size for uniform random traffic ..................................................... 43
LIST OF FIGURES

4.13 Relative performance with varying system size . . . . . . . . . . . . . . 44
4.14 Percentage gain in performance for HiWiMesh . . . . . . . . . . . . . . 46
List of Tables

4.1 Characteristics of P-SAM mechanism ............................................. 24
4.2 Characteristics of D-SAM mechanism ............................................. 25
Chapter 1

Introduction

1.1 Motivation

Network-on-Chip (NoC) has emerged as the enabling technology for catering to the communication needs of high performance applications in modern multi-core System-on-Chips (SoCs). As per Moore’s law, the number of transistors on a chip will roughly double in every technology generation [1], as a result of which, we now have entered the billion transistor era. With the scaling of CMOS technology, multi-core systems have become the de facto design choice to meet the power-performance demands of the applications. Although, these systems now include as many as eight cores (e.g. Xeon Processors) to hundreds of cores [2], this number is predicted to scale up to 1000 of cores in near future [3].

The conventional approach to tackle data communication within multi-core system is the global crossbars or the shared bus interconnection system. This approach scales unsteadily with the increasing number of cores in multi-core system. Therefore, the interconnection between the increased cores drives the need to have a modular-efficient interconnection network-on-Chip (NoC) [4]. NoC architectures can be used to provide the communication requirement for the future increasing of cores in which data have routed across these networks by using switches. Although a bus based interconnect is traditionally used for small scale multi-cores, due to the scalability
issues, Network-on-Chip (NoC) architecture are used to provide the communication infrastructure in large multi-core systems [5]. Wireless Network-on-Chip has emerged as an efficient interconnection network for large multi-core chips. With the use of wireless transmission and to avoid any conflict of resources an efficient and fair medium access control (MAC) mechanism is required. In this work we propose a prediction model that is used by traffic aware dynamic MAC mechanism to efficiently allocate transmission slots for wireless interfaces. An overview of the proposed Proportional, Integral, and Differential (PID) prediction model is shown in Figure.

1.2 Thesis Contribution

The following points summarize contributions made during the work of this thesis:

- Design and implementation of Proportional, Integral, and Differential (PID) based prediction model for predicting the traffic demands of WIs in the WiNoC interconnection system

- Integrating the proposed PID prediction mechanism with the traffic aware dynamic MAC units

- Discussing the optimization process for evaluating the weights of the prediction model

- Performing system-level simulations for evaluating the WiNoC interconnection system with this integrated prediction mechanism

- Present a system performance comparison between the Token passing MAC and PID prediction model enabled dynamic MACs
CHAPTER 1. INTRODUCTION

1.3 Organization of The Thesis

This thesis is organized in five chapters. A brief information about each chapter is mentioned below:

- **Chapter 1. Introduction**: introduces the motivation behind the work that has been performed toward completing this thesis. Then, it states the contribution of this thesis and thesis organization.

- **Chapter 2. Background and Related Work**: explains the trend from single chip to multiple cores systems and evolution of Network on-Chip as an interconnection fabric. Then it describes the need of Wireless Network-on-Chip (WiNoC) interconnection system and how it can further improve the performance of interconnection system.

- **Chapter 3. The Proposed Prediction Model**: describes the different components involved in the proposed prediction model, and how this model is suitable for our dynamic MAC schemes. It also discusses the optimization process involved while setting the weights of the prediction model.

- **Chapter 4. Traffic Aware Dynamic MAC Mechanism**: discusses the impact of the prediction model and how the dynamic MACs improves the system performance with respect to the baseline NoC and token-based MAC scheme.

- **Chapter 5. Conclusion and Future Works**: mentions a brief summary about the work of this thesis and the obtained results. It also point out the future work.
Chapter 2

Background Information and Related Works

2.1 Multi-core System On-Chip (SoC).

With the technology’s scalability, the number of transistors fabricated on a single chip is going to increase. This is in accordance with Moore’s law as it states that the number of transistors in integrated circuits have been doubling every eighteen months [6]. Figure 2.1 shows the increase in the transistors counts with time per Moore’s law. Traditionally, a better performance of the processor can be achieved by increasing the operating clock frequency. Based on power formula $Power = V_s^2 FC$ where $V_s$ is the voltage supply, $C$ is the effective switched capacitance, and $F$ is the operation frequency, it is certainly obvious that clock speed is directly proportional to the power consumption. Thus, any increase in the clock frequency rate will lead to a massive increasing in power dissipation. Therefore, the increase in operation clock frequency speed to achieve better performance has reached to its boundary limit. As a result, chip designers have examined a promising alternative method that can be used to get better performance with a agreeable power dissipation. These have approved to be obtained by using multi-core chips instead of single core system. The reason for that is because the capabilities of the parallel computing which make multi-core faster while operating at the same clock speed. Furthermore, comparatively, achieving better performance at moderate clock speed make multi-core chips ideal to use with
Figure 2.1: Transistors counts against dates of introduction in accordance with Moore's law [1]. The number of transistors in an integrated circuit is doubling every eighteen months.

lower power dissipation.

The Traditional architecture to interconnect multiple cores on-chip and achieve better performance is the shared medium arbitrated bus [7]. There is a couple of bus interconnect architectures that can be used to integrate IP blocks such as ARM AMBA [8], Wishbone [9], and IBM CoreConnect [10]. Unfortunately, these shared bus interconnects do not scale as the system size increases due to the increase in the parasitic resistance and capacitance. Furthermore, adding more IP blocks to the bus can cause in increasing the parasitic capacitance thereby increasing the propagation delay. Therefore, using shared bus in the interconnection between IP blocks is eventually limiting the system scalability [11]. With the high demand for
the modern systems which integrate hundreds of IP blocks while maintaining the delay of the global communication, chips designer have explored an alternative way for global interconnection across a chip. Network-on-Chip interconnects paradigm was found to be the promising architecture for global communication within IP blocks.

2.1.1 Network-on-Chip (NoC)

The influential disadvantage of using multi-core chips is the not scalability of the global wire delays. Global wires are used to carry the signals in multi-core chips. These wires does not scale in length as the technology scale, and their delay increases significantly [12]. Their delay can exceed multiple clock cycles even after insertion repeaters. It clearly demonstrated that, in ultra-deep submicron processes, 80 percent of delays of the critical paths are because of interconnections [13], [14].

As the future generation of multi-core system will contain hundreds or even thousands of IP cores integrated on a single chip, the length of the global wire to connects those cores will significantly increase. This has resulted in massive increasing interconnect communication delay and thereby limits the system scalability. Therefore, there is a need to have a scalable interconnection network that can be utilized to integrate more and more IP block cores without limiting system scalability. This can be achieved by using on-chip interconnection network. With this interconnection, the IP block cores could separate from the communication network which provides a plug-and-play system. The new paradigm of designing a scalable interconnection network is called Network-on-Chip (NoC) [4].

Figure 2.2 shows a simple NoC architecture in which it can be seen that it consists of the following components: IP cores, switches, and inter-switch links. With NoC system, global wires have replaced with a logic network, and the data is routing across the network through switches and links. For routing thses data, there are different types of switching, namely Circuit Switching, Packet Switching, and Wormhole
Figure 2.2: Typical Network-on-Chip NoC architecture with Mesh topology.

Among these switching technique, Wormhole switching has been adapted in this thesis due to the smaller network area and more efficient of network utilization. In wormhole switching, the packet has broken into smaller units called flits which are header flit and body flit. The size of each flit is determined so that they can traverse between any two adjacent switches within certain clock cycle. The first flit of the packet which is the header flit contains routing information used to establish the path of the entire packet from the source node to destination node. The rest of the flits follows the header flit to their final destination in a pipeline fashion [15]. The path setting up by the header flit may block by other communication of other nodes e.g. the path could be reserved by a particular packet till it is completely transmitted.
To overcome this problem, virtual channel buffers have introduced. NoC switches have virtual channel buffers to store the flits until the path is available to send them. Figure 2.3 shows an overview of the structure of NoC switch with input and output virtual channels.

2.1.2 Network-on-Chip (NoC) Topology

Network topology determines how many hops i.e. routers in the network that the packet must traverse to reach the final destination as well as the interconnect lengths within these routers. Thus, Selecting NoC topology plays a significant role in determining the system performance. In last few decades, many NoC topology has been explored. They can be classified as direct or indirect topologies. In direct topology, every router or switch is directly connected to end node which is the source or destination of packets. In indirect topology, nodes could serve as a direct IP core with an associated router or intermediate node help transferring the packet across terminal nodes. Figure 2.4 shows direct and indirect NoC topology.

Mesh topology has been proposed in [16] as NoC architecture in which this design consists of m x n switches interconnecting IP blocks cores. The number of switches in this topology is equal to the number of IP blocks. Every switch is connecting to four neighboring switches, and one connects local IP block except the ones at the edge.
CHAPTER 2. BACKGROUND INFORMATION AND RELATED WORKS

Figure 2.4: Network on-Chip (NoC) topology architectures [54] where (a) and (d) are examples of indirect topologies while (b), (c), (d) and (e) are examples of direct topologies.

Furthermore, the links between switches or between switch and IP block consist of two unidirectional interconnects. Folded Torus topology [17] is similar to the Mesh topology except that it has a wrap around links in the switches at the edge. This has resulted in having five ports for every switch, four to connect with the neighboring switches and one to the local IP blocks. Mesh and Folded Torus both are example of NoC regular topology.

2.1.3 Emerging Interconnects

As the technology continues to scale down, the interconnects wire are getting thinner and thinner resulting in increasing its resistance. The increase in resistance can lead to significant increase in power dissipation and latency. According to the International Technology Roadmap for Semiconductor (ITRS), interconnects are the major bottlenecks to overcome the power-performance barrier in the future genera-
Increasing in power consumption can lead to a high on-chip temperature which in turn compromises the performance and reliability of the chip [18]. Hence, it is clear that the challenges facing future chips are the scalability of on-chip interconnects.

Because of that, different interconnect technologies has been proposed to improve the performance of the traditional NoC interconnects such as three-dimensional integration, photonic interconnects, and multi-band RF (wireless) interconnects [19], [20]. These new approaches are considered the promising paradigms that are capable of improving the power dissipation and the performance of the NoC design.

In three-dimensional integration interconnects, multiple active layers have integrated onto a single chip. The main advantage of this interconnects is that it reduced the hop counts due to the reduction in length and number of global interconnects. Another benefit of 3D integration of chips is that two different technologies can be connecting with each other. However, due to the small footprint, the power density of 3-D interconnects would be high thereby high-temperature dissipation [21] which in turn requires cooling mechanism [22]. Furthermore, fabricating 3D chips has sets of issues with inter-layer alignment, bonding, and inter-layer patterning [21] resulting in high risk of manufacturing defects.

In photonic interconnect technology, optical interconnects are used instead of traditional global interconnect to transmit data within switches [23]. The transmission of data is carrying at the speed of light. Therefore, photonic interconnects has an advantage of having low latency with a high bandwidth. Also, due to the extremely low data loss, optical interconnects is considered a reliable data transmission. The disadvantage of this interconnects is that it is hard to integrate photonic interconnects with silicon devices.

In wireless interconnects, switches are equipping with a wireless link that contains antenna and transceiver [24]. Using this interconnect, different NoC architectures has been proposed [25], [26]. This has lead to transmit the traditional multi-hop intercon-
nects with a single-hop long distance shortcut. Several recent studies have found that wireless links can significantly improve chips performance and better power consumption [27], [28].

2.1.4 Wireless Network-on-Chip (WiNoC)

Among, these alternatives, wireless interconnect operating in millimeter wave (mm-wave) band is nearer term solution due to its CMOS compatible integration of the underlying enabling technology of miniature antennas and transceivers [29]. However, utilizing the full potential of the novel mm-wave interconnect technology in a wireless NoC (WiNoC) requires overcoming two critical design challenges: i) design of efficient, simple and fair medium access control (MAC) mechanism and ii) managing the wireless bandwidth effectively. In any wireless network, a MAC mechanism is responsible for ensuring contention free communication among the wireless nodes over the shared wireless channel. However, unlike macro networks, the MAC for WiNoC requires to be simple to minimize area and power overheads [30]. In mm-wave WiNoC architecture, designing multiple non-overlapping channels for Frequency Division Multiple Access (FDMA) is non-trivial from the perspective of transceiver...
design and is not easily scalable into more than a few concurrent channels. Hence, a single wireless channel is shared among the wireless interfaces (WIs). To divide this shared channel into multiple orthogonal code channels, enabling concurrent communication, a Code Division Multiple Access (CDMA) mechanism has been proposed in [31]. However, such scheme requires power hungry coherent transceiver circuits. On the other hand, due to the distributed and low-overhead implementation, and fairness in channel access, Time Division Multiple Access (TDMA) is used in many WiNoC architectures [32], [33].

In this work, we use two TDMA based dynamic MAC mechanisms; Proportionate Slot Allocation Mechanism (P-SAM) MAC and Demanded Slot Allocation Mechanism (D-SAM) MAC that is able to dynamically allocate the WI slot durations based on a prediction of the traffic demands. The proposed slot allocation mechanism relies on a prediction of the traffic demands rather than being based on current utilization of links or switches to minimize reaction times to changes in traffic. Furthermore, we design a proportional, integral, and differential (PID) based prediction mechanism to predict the traffic demands of the WIs with high accuracy and low implementation complexity. Based on these predicted demands, the MAC dynamically allocates transmission slot durations to the WIs.

2.2 Related Works

A comprehensive survey regarding various WiNoC architectures and their design principles is presented in [34]. A WiNoC architecture augmented with directional on-chip planner log periodic antennas is explored in [35] for simultaneous multi-channel communications. However, placing multiple directional antennas without interference among them is not trivial. Hence, in many WiNoC architectures [30], [32], [33] antennas with omni directional radiation pattern has been proposed where the wireless channel is shared among the WIs. However, in order to communicate via this shared
wireless channel without interference and contention, a MAC mechanism is required. Due to energy, area and memory constraints in the on-chip environment, complex MAC mechanisms used in conventional networks are not suitable for WiNoCs [36]. Therefore design of efficient, low overhead, and fair MAC mechanisms are considered as one of the critical challenges for WiNoCs [37]. A synchronous and distributed MAC mechanism (SD-MAC) is proposed in [38] for the Ultra-Wide-Band (UWB) WiNoCs where impulse based transceivers are used. However, the communication range for the WIs in such WiNoC is limited to a millimeter. Furthermore, to access the wireless medium, local arbitration between the WIs using wired links is required. Thus, such MAC mechanism cannot be adopted for WiNoCs where the WIs is more than a millimeter apart. A WiNoC architecture where the WIs are distributed across the chip is proposed in [39]. The WIs are equipped with Carbon Nanotube (CNT) antennas to enable multiple concurrent channel among the WIs. A hybrid MAC mechanism combining both TDMA and FDMA is reported for such CNT based WiNoC architecture. However, the CNT based wireless technology is difficult to integrate in current CMOS process. On the other hand, miniature antennas operating in the mm-wave frequencies are CMOS compatible and are nearer term solution [29]. Authors in [40], [41] has proposed a WiNoC architecture with multiple non-overlapping channels to enable FDMA based medium access. However, such FDMA based approach is non-trivial from the perspective of transceiver design and the number of concurrent channels are not easily scalable. A CDMA based MAC mechanism is proposed in [31] for mm-wave WiNoCs to efficiently utilize the wireless bandwidth. However, CDMA requires coherent Binary Phase Shift Keying (BPSK) receiver along with Analog-to-Digital Converters (ADC) in the transceivers making the design significantly challenging. Similar to the CDMA MAC mechanism, a distributed MAC mechanism is proposed in [36] for mm-wave WiNoCs that uses simple orthogonal request packets. These request packets are processed at each WI and permission to the wireless channel is
granted by a priority based mechanism. However, maintaining orthogonality among these channels is difficult to achieve. Moreover, this method has an overhead of maintaining the state of current transmission at each transceiver. In [42], authors proposed TDMA based CSMA MAC mechanism for WiNoC architectures. However, due to contention based retransmission, such MAC suffers from performance issues as demonstrated in [43]. Therefore, token passing based MAC (T-MAC) mechanism is adopted for most TDMA based WiNoC architectures [32], [33]. In the T-MAC mechanism, the access to wireless medium is granted to a WI by the possession of a token, circulating among the WIs, organized in a virtual ring. As the traffic demands through the WIs vary both spatially and temporally a fixed transmission duration or slot for each WI will result in underutilization of the wireless channel [44]. In order to improve the performance of this simple, and distributed mechanism, a dynamic radio access control mechanism (RACM) is proposed in [45] where the unused slot in an epoch is redistributed among the WIs with higher slot usage in the next epoch. However, this mechanism relies only on the current utilization of the WIs. Similarly, in [46] a mechanism to monitor the link utilization and allocate time slots to the wireless transceivers was proposed. However, this scheme also reacts to changes in the utilization instead of predicting it, making the mechanism slower in a system level response. In this work, we propose the design of a PID based prediction model for dynamic MAC mechanism that allocates the transmission slots based on a prediction of the traffic demand of the WIs and improves the performance of the WiNoC architecture.
The proposed PID prediction model is based on the PID controller widely used in control systems. The PID design used in control system is a feedback design, but our PID prediction model is not a feedback based learning. In this chapter we will discuss the design, working, and optimization process involved in PID prediction model.

3.1 Basic Assumptions and Notations

To establish the notation for future use, we will use \( x^{(i)} \) to denote the "input" variables, also called as features, and \( y^{(i)} \) to denote the "output" or target variable that we are trying to predict. A pair of \( (x^{(i)}, y^{(i)}) \) is called a training example, and the dataset that we will be using to learn is called a training set, i.e. a list of \( m \) training examples. The superscript "(i)" in the notation is simply an index into the training set. Let \( X \) denote the space of input features, and \( Y \) the space of output values. Our goal is, given a training set, to learn a function \( h: X \leftrightarrow Y \), where function \( h \) is called as hypothesis function. The typical top-level prediction process is shown in the figure 3.1.

The type of learning that we have targeted falls in the domain of supervised learning. In supervised learning, training example is the pair of input value and the corresponding output value and here the learning algorithm uses this information to train the prediction model. Contrary, if the learning algorithm training stage
involves only the input features it is called **unsupervised** learning. When the target variable that we are trying to predict is continuous then this type of learning is called **regression**. When \( y \) can take only a small number of discrete values it is called **classification**. Our prediction is a regression problem, where we are trying to predict the traffic demand for a WI in the next epoch.

### 3.2 PID Prediction Model

In our proposed prediction model we have three units, proportional (P), integral (I), and differential (D). The proportional term captures the instantaneous traffic demand of a WI for the present epoch. The integral part captures the average traffic demand till the present epoch, and the differential term captures the difference between the traffic demand of present epoch and previous epoch. So, the proportional and integral terms captures the current and steady state demand of the WI, whereas the differential captures the temporal variation in the demand. Hence, the PID based prediction mechanism capture the change in WI traffic demand due to both the short term and long term traffic variations, suitable for bursty traffic in NoCs.
3.2.1 Analytical Model

Let \( D_{j}^{\text{pred}} \) be the predicted traffic demand for wireless interface (WI) \( i \) for an epoch \( j \), the proportional part for the WI \( i \) be \( D_{j-1} \), integral part will be average demand \( D_{\text{avg}} \), and differential part will be the difference of \( D_{j-1} \) and \( D_{j-2} \). Then the equation 3.1 will represent our prediction model and it will also be our hypothesis \( h_K(D) \)

\[
D_{j}^{\text{pred}} = K_P \times D_{j-1} + K_I \times D_{\text{avg}} + K_D \times (D_{j-1} - D_{j-2}) \tag{3.1}
\]

3.2.2 Cost Function

Now given the training set, the next step is to decide the values of PID weights \( K_I, K_P, \) and \( K_D \). To achieve this we will try to make \( h_K(D) \) close to \( D_{j}^{\text{pred}} \), for our training examples. To formalize this, we will construct a function that will measure, for each value of PID parameters, how close \( h_K(D^i) \)'s are to the corresponding \( D_{j}^{\text{pred},i} \). This function is called cost function:

\[
J(K) = \frac{1}{2} \sum_{i=1}^{m} (h_K(D^i) - D_{j}^{\text{pred},i}) \tag{3.2}
\]

3.2.3 LMS Algorithm

The next step after we have formulated the cost function is to determine the values of PID weights for which \( J(K) \) is minimized. To do so we will use a search algorithm that starts with an initial guess for \( K \), and that repeatedly changes \( K \) to make \( J(K) \) smaller. For this we will use gradient descent algorithm, which starts with some initial \( K \), and repeatedly performs the update.

\[
K_j := K_j - \alpha \frac{\partial J(K)}{\partial K_j} \tag{3.3}
\]

This update is simultaneously performed for all values of \( j \), and \( \alpha \) is called learn-
**CHAPTER 3. THE PROPOSED PID PREDICTION MODEL**

**ing rate.** This is a very natural algorithm that repeatedly takes a step in the direction of steepest decrease of $J(K)$. If we take partial derivative of equation 3.3 for a single training example, we will get:

$$ K_j := K_j + \alpha(D_{j}^{pred,i} - h_K(D^i))x^i_j $$  \hspace{1cm} (3.4)

This rule is called the LMS update rule (LMS stands for "least mean squares"), and is also known as **Widrow-Hoff** learning rule. We have derived the LMS rule for only single training example. For training set of more than one training example the rule can be modified as:

**repeat until convergence;**

$$ K_j := K_j + \alpha \sum_{i=1}^{m} (D_{j}^{pred,i} - h_K(D^i))x^i_j \quad (for \ every \ j) $$  \hspace{1cm} (3.5)

The equation 3.5 looks at every example in the entire training set on every step, and it is called as **batch gradient descent**. The gradient descent can be susceptible to local minima in general, but our cost function $J(K)$ is a convex quadratic function, so the gradient descent on our cost function will give only on minima i.e. global minima.

### 3.3 Evaluation of PID Prediction Model

The training dataset that we have used for training our prediction model is generated by running the system level simulation of our multi-core WiNoC simulator, which is modeled in MATLAB. So to generate the training set of actual traffic demand of the WI, we monitor the number of incoming flits at the WI per epoch for the baseline WiMesh architecture for uniform random traffic at full injection load of 1 flit per core per cycle. The training set consists of 5000 traffic demand values per WI.
3.3.1 Optimization of PID weights

The PID weights are optimized by applying a two-step optimization process, first we apply the gradient descent algorithm on the constructed cost function for getting the values of $K_P$ and $K_I$. Then we use the obtained $K_P$ and $K_I$ values in the cost function to get the $K_D$ gain value. This type of optimization process is commonly used for control system based PID controller.

The values of the cost function with different $K_P$ and $K_I$ values at different iteration of the gradient descent algorithm is shown in Figure 3.2. It can be seen from the figure that the value of the cost function is minimum when $K_P = 0.66$ and $K_I = 0.13$. Then, at the second step of the optimization process, the optimal KP and KI values are used to determine the value of KD using the same cost function and gradient descent algorithm used in the first step. The values of the cost function for different values of $K_D$ is shown in Figure 3.3. From figure, it is observable that the cost function is minimized when $K_D$ is 0.2041.
In the next chapter we will discuss how we integrate the proposed prediction mechanism with dynamic MACs, and later we will analyze the impact of the traffic aware dynamic MAC mechanisms on the WiNoC system.
Chapter 4
Traffic Aware Dynamic MAC Mechanism

In this chapter, we will first discuss the integration of the PID prediction mechanism with dynamic MAC. Then we will evaluate the performance and energy efficiency of the proposed dynamic MAC based WiNoC architectures. Although, many different WiNoC architectures has been proposed in literature, for this work we have considered a Mesh based hybrid WiNoC architecture (i.e. WiMesh) with both wired and wireless links as a test case. The WiMesh architecture is discussed in details in the next subsection. The performance of the WiNoC architecture is measured as the peak achievable bandwidth per core or bandwidth and packet latency. The bandwidth of a WiNoC is determined as the average number of bits successfully routed to the destination cores per second from each source core. The packet latency is the average number of clock cycles required to successfully transmit a packet to the destination core. The energy efficiency of the WiNoC is measured as the packet energy which is the average energy (both dynamic and static) required to route a whole packet from source to destination through the NoC components (i.e. switches and the links). In the next subsection we describe the baseline WiMesh architecture used in this work for evaluations.
4.1 Integrating PID prediction with dynamic MAC

The dynamic MAC unit consists of a prediction unit, allocation unit, and a sleep/wake unit. The PID predictor will reside in the prediction unit of the dynamic MAC. The output of the prediction unit will be the predicted traffic demand of a WI, this output is fed to the allocation unit for further slot allocation based on prediction output. We observe that due to the accurate traffic demand prediction and efficient allocation of the transmission slots, the number of wasted slots (no flits are transmitted) reduces by 1.3% and 13.5% respectively for the WiMesh+P-SAM and WiMesh+D-SAM architecture when compared to the baseline WiMesh for uniform-random traffic.

Figure 4.1 shows the architecture of dynamic MAC unit, in this work we have used two dynamic MAC mechanism, the slot allocation is based on the predictive traffic demand of the WI. Consequently, the slot durations are adjusted to cope up with the traffic variation. Two slot allocation mechanisms, proportional slot allocation mecha-
nism (P-SAM) and demanded slot allocation mecha-nism (D-SAM) for determining the slot duration of the WIs based on the predictive demand are used. The WIs are equipped with an allocation unit that implements one of the two slot allocation mechanisms.

4.1.0.1 P-SAM

In the P-SAM scheme, the slot duration of a WI is dynamically adapted based on the proportion of the predicted traffic demand of the WI compared to other WIs. The duration of a slot for a WI in an epoch is allocated dynamically at the start of each epoch to cope with the varying demand of the WIs. However, the allocation of slot durations in a P-SAM scheme is constrained in such a way that the duration of the epoch remains constant. The allocated slot duration of a WI $I$, at epoch $j+1$, $S_{i}^{j+1}$, is given by equation 4.1

$$S_{i}^{j+1} = \frac{D_{i}^{pred}}{\sum_{i=1}^{n} D_{i}^{pred}} E_F$$ (4.1)

Here, $D_{i}^{pred}$ is the predicted traffic demand for WI $i$ for epoch $j$ calculated using equation 3.1, $E_F$ is a constant value that represents the number of data flits that can be transmitted over the wireless medium in an epoch and $N$ is the number of WIs in the system. Hence, in the P-SAM scheme the duration of epoch remains constant. However, each individual transmission slot for a WI within an epoch changes among epochs based on the predicted demand on the WIs.

To enable this P-SAM scheme, the Allocation Unit (AU) contains a register file, $REG_{demand}$ that is used to store the predicted demand of other WIs received from the slot information packet. In an epoch, when all the values of the $REG_{demand}$ is updated and $Demand_{self}$ is shared with other WIs, the slot duration for the WIs are calculated using the values in $Demand_{self}$ and $REG_{demand}$. This value is then used to update the Slot_counter at the beginning of the next epoch to indicate the number
Table 4.1: Characteristics of P-SAM mechanism

<table>
<thead>
<tr>
<th>Metric</th>
<th>Prediction Unit</th>
<th>Allocation Unit</th>
<th>Sleep/Wake Unit</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Power (mW)</td>
<td>0.1482</td>
<td>0.163</td>
<td>0.062</td>
<td>0.373</td>
</tr>
<tr>
<td>Area (µm²)</td>
<td>958.67</td>
<td>1264.32</td>
<td>406.05</td>
<td>2629.02</td>
</tr>
<tr>
<td>Delay (ns)</td>
<td>0.12</td>
<td>0.12</td>
<td>0.14</td>
<td>0.14</td>
</tr>
</tbody>
</table>

of flits the WI can transmit over the next epoch. The Epoch_counter is set to the fixed epoch duration at the end of each epoch. This update process and calculation of slots is performed in parallel with data transmission and hence it has no effect on the data transmission speed. Table 4.1 tabulates the area, power, and delay for implementing P-SAM scheme.

4.1.0.2 D-SAM

In a D-SAM scheme the slot duration is allocated dynamically based on the predicted traffic demand of the WI. However, unlike the P-SAM scheme, the number of flits that can be transmitted in a slot is equal to the predicted demand of a WI. Hence, the total number of flits that can be transmitted in an epoch can vary among epochs for the D-SAM scheme. The allocated slot duration for WI, i at epoch, j+1, \( S_{j+1}^i \) is given by equation 4.2

\[
S_{j+1}^i = \min(D_{j}^{pred}, M)
\]

(4.2)

Where, \( D_{j}^{pred} \) is the predicted traffic demand for WI i at epoch j and M is a maximum number of flits any WI can transmit within a slot. This maximum number of flits in a slot ensure no WI is completely denied access to the wireless medium. Similar to the P-SAM scheme, in the D-SAM scheme, the demand information shared using the slot information packet is used to calculate the number of flits that can be transmitted in an epoch. The flits that can be transmitted in epoch \( j + 1 \), \( E_{F}^{j+1} \), is
Table 4.2: Characteristics of D-SAM mechanism

<table>
<thead>
<tr>
<th>Metric</th>
<th>Prediction Unit</th>
<th>Allocation Unit</th>
<th>Sleep/wake unit</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Power ((mW))</td>
<td>0.1482</td>
<td>0.0759</td>
<td>0.062</td>
<td>0.286</td>
</tr>
<tr>
<td>Area ((\mu m^2))</td>
<td>958.67</td>
<td>373.319</td>
<td>406.05</td>
<td>1738.048</td>
</tr>
<tr>
<td>Delay ((ns))</td>
<td>0.12</td>
<td>0.13</td>
<td>0.14</td>
<td>0.14</td>
</tr>
</tbody>
</table>

given by equation 4.3

\[ E_{F}^{j+1} = \sum_{i=1}^{N} S_{j+1}^{i} \quad (4.3) \]

Here, \(S_{j+1}^{i}\) is the allocated slot duration in the epoch \(j + 1\) for WI \(i\). The duration of the next epoch is calculated after all the demand information of the WIs are shared in an epoch. The allocation unit in a D-SAM contains the register file \(REG_{demand}\) as in the P-SAM MAC. However, unlike the P-SAM scheme, the Epoch_counter for the D-SAM scheme is updated with the sum of \(REG_{demand}\) and \(Demand_{self}\) at the end of each epoch. Table 4.2 tabulates the area, power, and delay for implementing D-SAM scheme.

4.2 WiMesh Architecture

In the WiMesh architecture, each core is connected to a NoC switch using a wireline link. The switches are then connected with other switches in its cardinal directions (i.e. NSEW) using wireline interconnects to form a regular Mesh. The Mesh is chosen as it is a conventional NoC topology used in several multicore based products and is relatively easy to design, verify, and manufacture. To provide single-hop shortcuts among the distant NoC switches to reduce the data transfer over multi-hop wireline paths, the wireless interconnects are over-laid on top of this Mesh topology by deploying the WIs at some of the NoC switches. To deploy the WIs for best performance gains, we adopt the optimization method outlined in subsection 4.3.

To realize the wireless interconnect, each WI contains on-chip antenna, transceiver circuit and serializer/deserializer buffers. As the WIs can be potentially at different
angles with respect to each other’s axes, the radiation pattern for the antenna should be non-directional. The antenna should also provide the best power gain for the smallest area overhead. A CMOS compatible metal mm-wave zigzag antenna operating at 60GHz mm-wave bands and bandwidth of 16GHz has been demonstrated to possess these characteristics. Hence, we equip the WIs with such miniature on-chip zig-zag antenna to enable the long-range shortcuts among the WIs located at different part of the chip. On the other hand, to ensure high performance and energy-efficiency, the transceiver circuit has to provide a very wide bandwidth and consume low power. We adopt the transceiver design from [47], [48] where low power design considerations are taken into account at the architecture level. Non-coherent on-off keying (OOK) modulation is chosen, as it allows relatively simple and low-power circuit implementation. The power and bandwidth of the OOK transceivers are adopted from fabricated prototypes demonstrated in 65nm technology [47], [48]. The wireless transceiver is shown to dissipate 2.06pJ/bit sustaining a data rate of 16Gbps with a bit-error rate (BER) of less than 10-12 while occupying an area of 0.17mm2 in post-layout design using TSMC 65nm CMOS process. The serializer/ deserializer buffers realized through shift registers work as a data interface between the transceiver and NoC switch.

The WiMesh is an irregular network as it contains both short (e.g. wired interconnects) and long (e.g. wireless) links. In [49], a shortest path based routing is used to optimize the performance of such irregular networks. Hence, we adopt such shortest path based routing scheme for the WiMesh architecture. We use a forwarding table based routing over precomputed shortest paths. The shortest path between any two pairs of nodes in the network is determined using a minimum spanning tree formed by Dijkstra’s algorithm. The minimum spanning tree formed by the Dijkstra’s algorithm depends on the chosen start node but the length of paths between any particular pair is independent of the start node. Hence, the minimum spanning tree is selected randomly. Furthermore, deadlock is avoided by transferring flits along the shortest path.
routing tree extracted by Dijkstra’s algorithm, as it is inherently free of cyclic dependencies. Hence, each switch only has local forwarding information eliminating the need for maintaining non-scalable global routing information resulting in a scalable routing mechanism.

4.3 Simulation Platform

The NoC architectures (i.e. topology, switch architecture, flow control and routing mechanism, MAC scheme, link delay and bandwidth etc.) are characterized using a cycle accurate NoC simulator. The simulator accurately models the progression of the flits over the switches and links per cycle accounting for those flits that reach the destination as well as those that are stalled. The post-synthesis delay and the energy dissipation of the NoC components considering both dynamic and static power consumption are annotated into the simulator for evaluating the performance and energy efficiency of the NoC architectures. We consider a system size of 64 cores that represents the current trends in multi-core chip design in the industry [50]. We also demonstrate the scalability of the proposed MAC by evaluating it for higher system size of 256 cores. In each experiment with synthetic traffic, ten thousand iterations were performed eliminating transients in the first thousand iterations. We also evaluate the proposed MAC mechanism for application specific traffic scenarios. For the wired switches, we adopted a wormhole based flow control mechanism, where the packets are broken in smaller flow control units or flits. The NoC switch is adopted from a three-stage pipelined design [51]. Each switch is considered to have 4 VCs with a buffer depth of 2. However, as the WIs handle a large volume of traffic, an increased number of VC of 8 with buffer depth of 16 is used. A moderate packet size of 64 flits is considered for all our experiments. The width of all wired links is considered to be same as the flit size, which is considered to be 32 bits. For the comparative study among the MAC mechanisms, we consider a wireless token passing based MAC
mechanism (e.g. T-MAC) as the baseline MAC mechanism. In the baseline T-MAC mechanism, a token in the form of a wireless flit is circulated among the WIs in a round-robin fashion. A WI can only transmit one entire packet when it possesses the token to maintain the integrity of the wormhole routing mechanism over the wireless channel. Therefore the fixed duration of transmission slot in the baseline T-MAC is considered to be the size of one complete packet. This necessitates increased buffer depth to accommodate complete packets in the VCs of the WIs with this baseline MAC.

The packet energy is estimated by adding the energy consumption in link (both wired and wireless) and switch traversals by the packet. The energy dissipation and delay of the wired link is obtained through Cadence simulations taking into account the specific lengths of each link based on the established topology in the 20mmx20mm die. For the wireless interconnect, we adopted the antenna [32] and the transceiver design from [47], [48] as mentioned in the previous subsection. The NoC switches and proposed dynamic MAC units are synthesized from a RTL level design using 65nm standard cell libraries from TSMC using Synopsys. A 2.5GHz clock and 1V Vdd representing the nominal frequency and voltage for the 65nm technology node is used for synthesis. The synthesis result of the proposed dynamic MAC unit is shown in Table 4.1, 4.2. The power consumption and the delay of the MAC unit is considered in our simulations. However, as is evident from Fig. 2 the MAC unit is parallel to the datapath of flit transmission and reception. Therefore, its latency does not impact the overall data latency. However, the energy and delay overheads of circulating the slot information packet is considered in our simulations. The delay overhead due to the slot information packet depends on its size. In the WiMesh architecture, the maximum size of the slot information packet turns out to be 4 flits for . Next, we use this simulation platform to optimize the WiMesh architecture for best performance before performing the evaluations of the proposed MAC mechanism.
4.4 Optimizing the Baseline WiMesh

To optimize the baseline WiMesh architecture for best performance, we divide the Mesh in small logical subnets of equal size where the size denotes the number of cores in a subnet. The WIs are then deployed in a center switch of the subnet. A conceptual view of the WiMesh architecture is depicted in Figure 4.2. The performance of the baseline WiMesh architecture (i.e. WiMesh with T-MAC) varies with different subnet sizes as the number of WIs in the system varies due to the WI deployment strategy. In the WiMesh architecture when the subnet size is large (i.e. the number of WIs is low), each WI is shared by many cores. This results in a high traffic load through the WIs as they provide long-range shortcuts between distant switches. On the other hand, when the subnet size is small (i.e. number of WIs is high), the interval between two consecutive channel accesses by a WI is large. Due to this long interval in channel access, the flits at the WIs waits longer. Hence, to achieve the best performance in the baseline WiMesh, the subnet size or number of WIs should be optimized.

Figure 4.3 depicts the peak achievable bandwidth per core for the baseline WiMesh with different subnet sizes for uniform-random traffic pattern. We follow the same subnet based WI deployment strategy while generating the WiMesh configurations with different number of subnets. Uniform-random traffic pattern is used for this optimization to capture the effect of both short and long distance communication. In the uniform-random traffic pattern, a packet is destined to any other core with equal probability. From figure 4.3, it is observable that the performance of the WiMesh is maximum when the subnet size is 8 (i.e. number of WIs = 8). When the subnet size is lower (e.g. 4) than the optimal value, due to an increased number of WIs in the system, the interval in channel access by a WI increases. This increases the packet waiting time at the WIs and results in a lower performance than the WiMesh with subnet size of 8. Alternatively, when the subnet size is larger (e.g. 16) than this
optimal value, each WI is shared by more number of cores and the traffic load at the WIs is increased. This results in a congestion at the WIs and reduces the performance of the WiMesh. As the performance for the WiMesh is maximum for a subnet size of 8 (i.e. 8 WIs), we consider this as the baseline WiMesh configuration. This configuration is also used for evaluating WiMesh with other MAC mechanisms so that consistency across all the architectures is maintained. To differentiate these WiMesh architectures with different MAC mechanisms, we use the notation WiMesh+MAC to denote the WiMesh with a specific MAC mechanism.

4.5 Performance Evaluation with Synthetic Traffic

In this section, we evaluate the performance and energy efficiency of the WiMesh architecture with different MAC mechanisms for synthetic traffic patterns. To vary the traffic load passing through the WIs, uniform random, hotspot, and bit complement synthetic traffic patterns are used in this evaluation. We also evaluate the
WiMesh architecture for broadcast traffic pattern to capture a wide range of cache coherency traffic in a multi-core system.

4.5.1 Evaluation with Uniform Random Traffic Pattern

The peak achievable bandwidth per core and packet energy at network saturation for both the wireline Mesh and the WiMesh architectures with different MAC mechanism for uniform-random traffic pattern is shown in figure 4.4. Due to the presence of long-range wire-less shortcuts in the baseline WiMesh architecture, the peak bandwidth per core and the energy efficiency improves compared to the wireline Mesh architecture. However, in the baseline WiMesh, the transmission slots for the WIs are equal and not adjusted dynamically based on the varying traffic demands of the WIs. Due to such demand agnostic allocation, transmission slots are wasted for WIs with lower traffic demand. This limits the performance gain of the baseline WiMesh. Hence, the performance of the baseline WiMesh can be improved, by dynamically ad-
justing the transmission slots of the WIs. The dynamic WiMesh architectures studied in this paper (i.e. WiMesh+P-SAM, WiMesh+RACM, WiMesh+D-SAM), adjusts the transmission slots of the WIs at each epoch. Such allocation strategy of transmission slots enables WIs with higher traffic demand to access the wireless channel for a longer duration. This results in improved performance and energy efficiency for the dynamic WiMesh architectures compared to the baseline WiMesh as shown in figure 4.4. However, for both WiMesh+P-SAM and WiMesh+RACM architectures, the transmission slots are adjusted keeping the duration of the epoch constant. Moreover, in the RACM, the adjustment is based on current utilization instead of predicted ones. On the other hand, in the WiMesh+D-SAM architecture the duration of the epoch is determined based on the predicted traffic demands of the WIs further reducing wasted slots. Due to the efficient allocation of the transmission slots and the adjustment in epoch, the number of slots wasted for the WiMesh+D-SAM architecture reduces by 12.4% and 10.7% when compared to the WiMesh+P-SAM and WiMesh+RACM architectures respectively. This results in improving the performance of the WiMesh+D-SAM architecture.

The benefits of the WiMesh+D-SAM architecture is more evident in figure 4.5, where the average packet latency at different injection load is shown for the wireline Mesh and WiMesh architectures for uniform random traffic. Even at low load when the WI handles less volume of traffic, the average packet latency for the WiMesh+D-SAM is lower than all other architectures considered in this work. The gain in average packet latency for the WiMesh+D-SAM significantly increases at higher injection loads due to the increase in temporal and spatial variation in traffic demand. This is due to the efficient allocation of transmission slots based on the predicted traffic demand of the WIs.
Figure 4.4: Peak achievable bandwidth per core and packet energy for uniform random traffic

Figure 4.5: Average packet latency with varying injection load for uniform random traffic
CHAPTER 4. TRAFFIC AWARE DYNAMIC MAC MECHANISM

4.5.2 Performance Evaluation with Non-Uniform Traffic Pattern

In this section, we evaluate the performance and energy efficiency of the WiMesh architectures for non-uniform synthetic traffic. Two non-uniform synthetic traffic patterns, hotspot and bit-complement are used for this evaluation. For the hotspot traffic pattern a certain volume of traffic generated from all cores is destined towards a hotspot core. All other packets are sent to other cores following a uniform distribution. This type of traffic pattern is common for directory-based cache-coherent shared memory multiprocessor system where communication among the cores and directory is more frequent [52]. In our experiment, 10% of the total traffic is destined to the hotspot core which is chosen to be a core with a WI. In the bit-complement traffic pattern, packets from each core is always destined to cores whose ID is a complement of the source core. For example, packets generated from core i is always destined to core with ID \((N-i+1)\), where N is the number of cores in the system. Figure 4.6 and Figure 4.7 shows the peak achievable bandwidth per core and packet energy for the wireline Mesh and the WiMesh architectures with different MAC mechanisms at network saturation for hotspot and bit complement traffic pattern respectively. For both non-uniform traffic patterns, the baseline WiMesh architecture outperforms the wireline counter-part due to the long-range wireless shortcuts. The performance of the WiMesh architectures with dynamic MAC is better compared to the baseline WiMesh architecture. Similar to the uniform random traffic, the peak achievable bandwidth per core is highest and the packet energy is lowest for the WiMesh+D-SAM architecture among all the dynamic WiMesh architectures due to the reduction in number of wasted slots.

We also evaluate the average packet latency for different WiMesh architectures with these non-uniform traffic patterns as shown in Figure 4.8 (for hotspot traffic) and Figure 4.9 (for bit complement traffic). It can be seen from the figure that the average packet latency for the dynamic WiMesh architectures are lower than the
Figure 4.6: Peak achievable bandwidth per core and packet energy for hotspot traffic.

Figure 4.7: Peak achievable bandwidth per core and packet energy for bit compliment traffic.
Figure 4.8: Average packet latency with varying injection load for hotspot traffic

baseline WiMesh at all injection load due to the reduction in number of wasted slots. The packet latency is lowest for the WiMesh+D-SAM architecture, as less number of slots are wasted due to the efficient demand based allocation. This is also consistent with our observation on average packet latency for uniform-random traffic where the packet latency is lowest for the WiMesh+D-SAM architecture.

From these evaluations of the WiMesh architectures, we see that the D-SAM MAC mechanism provides the best performance and energy efficiency due to the efficient slot allocation. Hence, for further investigation, we consider only this dynamic MAC mechanism.
4.5.3 Evaluation with Broadcast Traffic Pattern

Maintenance of cache coherency is a common requirement in a multi-core environment [53]. As mentioned in section 4.3, a hotspot traffic pattern best represents the traffic for directory based cache coherency. However, to capture the communication pattern of broadcast based cache coherency protocols, we evaluate the proposed D-SAM MAC for broadcast traffic patterns. For this purpose, we consider a certain percentage of the traffic generated by the cores to be of broadcast nature. The rest of the traffic is unicast and the destinations are generated following the same uniform-random strategy described in subsection 4.4.1. Broadcast packets are duplicated during routing only when shortest paths to receiving cores diverge.

Figure 4.10 shows the relative gain in peak bandwidth per core and reduction in average packet energy for the WiMesh+D-SAM and baseline WiMesh architectures with respect to the wireline Mesh for varying percentage of broadcast traffic. It can
be observed from the figure that for different percentage of broadcast traffic the relative gain in performance and reduction in packet energy for the WiMesh+D-SAM architecture is higher than the baseline WiMesh architecture due to the efficient allocation of the transmission slots based on the predicted traffic demands. However, the relative gain in performance and packet energy displays a convex pattern with the increase in broadcast traffic percentage for both architectures. This is because the performance is also dependent on the traffic load of the underlying Mesh architecture. Due to the inherent broadcast capability of the wireless interconnect, a single transmission via the wireless medium is sufficient for transmitting the broadcast packets. This results in the initial improvement in relative performance in both wireless architectures. However, the receiving WIs routes these broadcast packets downstream using the underlying wireline Mesh network. Increasing the percentage of broadcast packets congest the underlying Mesh and increases the waiting time for packets. This eventually reduces the bandwidth and energy gain per packet with the increase in the percent of broadcast traffic.

4.6 Performance Evaluation with Application Specific Traffic

In this section, we evaluate the performance of the WiMesh+D-SAM architecture with application specific traffic patterns from PARSEC and SPLASH2 benchmark suites. To generate the application specific traffic patterns, we consider a multicore chip with 16 memory cores and 16 out-of-order (OoO) cores. Each core consists of a 32KB of L1 and 512KB of L2 cache running a Directory-Based MOESI cache coherency protocol. This core configurations are then used to extract the core-to-memory and memory-to-memory cache coherency traffic for the PARSEC and SPLASH2 benchmark applications when they are executed till completion using SynFull [54]. To map these traffic patterns to the 64-core environment, we consider 16 equal sized clusters where each cluster contains one memory core and 3 OoO cores.
Figure 4.10: Percentage gain in peak bandwidth per core and reduction in average packet energy with broadcast traffic.
Three threads of the same application are then executed on the multicore chip so that each core in a cluster runs certain portion of a thread and the memory cores in the clusters are shared among the threads.

The percentage reduction in average packet latency and average packet energy for the baseline WiMesh and WiMesh+D-SAM architecture with respect to the Wireline Mesh for different application specific traffic patterns is shown in Figure 4.11. The latency best represents the performance in these cases as the interconnection network is not saturated in the steady-state. The reduction in average packet latency and average packet energy for both wireless architecture varies between applications due to the variation in traffic patterns resulting in different traffic demands at the WIs. The long-range wireless shortcuts present in the baseline WiMesh architecture reduces the hop-count and provides efficient paths be-tween core-to-memory and memory-to-memory. Consequently, the energy efficiency and packet latency for the baseline WiMesh improves by 27.5% and 8.22% on an average over the wireline Mesh. On the other hand, the WiMesh+D-SAM architecture, not only provides the benefits of the long-range wireless shortcut but also en-ables efficient allocation of transmission slots to WIs based on their traffic demands. This dynamic adjustment of the transmission slots and the epoch in the WiMesh+D-SAM architecture enables further improment in energy efficiency and latency for the application specific compared to the baseline WiMesh. The average reduction in packet latency and packet energy for the WiMesh+D-SAM architecture compared to the Wireline Mesh is 33.29% and 47.73%. Hence, like synthetic traffic, for the application specific traffic patterns, the dynamic MAC enables improvement in energy efficiency and performance over the baseline T-MAC mechanism.
Figure 4.11: Percentage reduction in average packet latency and average packet energy with application specific traffic
4.7 Performance Evaluation with Varying Flit Size

We discuss the performance of the WiMesh+D-SAM architecture with varying flit size and compare it with the baseline WiMesh architecture. The increase in flit size is accommodated in the wireline network by the cost of extra hardware and silicon real estate (e.g., increasing wire width, buffer size). However, the bandwidth of the wireless interconnect is not easily scalable. Consequently, with increasing flit size, the cycles required to transmit a flit also increases. Thus, it is important to explore the effectiveness of dynamic MAC mechanism with varying flit size. Here, we investigate the performance of the WiNoC with the proposed MAC for flit sizes of 32, 64 and 128 bits. This is because as noted in [55], higher flit widths beyond 128 are shown to provide marginal gains in performance of a NoC based system. For all the cases the packet size is considered to be 64 flits.

The relative gain in peak bandwidth and reduction in packet energy for the baseline WiMesh and WiMesh+D-SAM architecture compared to the wireline Mesh with varying flit size is shown in Figure 4.12. In this evaluation, we have used the uniform-random traffic pattern. From the figure, it is observable that the performance gain of the baseline WiMesh compared to the wireline Mesh diminishes with increasing flit size. This is because, in the baseline WiMesh, the increasing in flit size results in an increase in the number of wasted slots. On the other hand, in the WiMesh+D-SAM architecture, the WI transmission slots in each epoch are dynamically adjusted based on the predicted traffic demand resulting in a lower number of wasted slots even with an increase in flit size. Due to this, the degradation in relative performance gain with increasing flit size for the WiMesh+D-SAM architecture is less steep than the baseline WiMesh. With a flit size of 128 bits the performance and packet energy of the baseline WiMesh is similar to the wireline Mesh. On the other hand, the performance of the WiMesh+D-SAM is higher than both the baseline WiMesh and wireline Mesh.
Figure 4.12: Relative performance gain with varying flit size for uniform random traffic for all flit sizes. The same trend is observed for reduction in packet energy.

### 4.8 Performance Evaluation with System Size

In this section, we investigate the performance and energy benefits of the dynamic MAC mechanism for larger system sizes. We considered two architectures with system size of 256 cores. In the first architecture, we considered the same number of WIs as in the 64 core system (i.e. 8 WIs). Hence, the subnet size will be much larger in this case and the WIs handle a high volume of traffic. In the second architecture, we considered the subnet size to be constant, increasing the total number of WIs in the system (i.e. 16 WIs). We adopted the same WI placement strategy mentioned in subsection 4.3 for these architecture.
CHAPTER 4. TRAFFIC AWARE DYNAMIC MAC MECHANISM

Figure 4.13: Relative performance with varying system size

The relative gain in peak achievable bandwidth per core and packet energy for the 256 core WiMesh architectures compared to the baseline 256 core WiMesh architecture with uniform random traffic is shown in Figure 4.13. It is observable from the Figure that the dynamic allocation based D-SAM MAC outperforms the baseline MAC in terms of performance and energy efficiency even for a larger system size of 256 cores irrespective of the scaling methodology. However, the improvement in performance is lower when the number of WIs are increased along with the system size (i.e. constant subnet size) compared to the case when the number of WIs remains constant. This is because, with increase in the number of WIs, the traffic is also distributed among the WIs and there is less spatial variation among the WIs. Therefore, for larger system size also the D-SAM MAC improves the performance of the WiMesh architectures compared to the baseline MAC mechanism.
4.9 Alternative WiNoC Architecture

To justify the benefits of the dynamic MAC mechanism it is important to investigate the performance of other WiNoC architectures with the proposed dynamic MAC mechanism. In this section we study the performance and energy efficiency of an alternative WiNoC architecture equipped with the proposed dynamic MAC mechanism. For the alternative WiNoC architecture we considered a hierarchical architecture as many of the WiNoC architectures proposed in literature adopt a hierarchical approach [39], [29], [33]. Similar to the WiMesh, the underlying wireline topology of the hierarchical WiNoC is a Mesh. The Mesh is then divided into equal sized subnets and each subnet has a hub. The hub is a NoC switch that is connected to all other switches in a subnet using wireline links. These hubs are placed at the center of each subnet. The hubs are in turn, connected to other hubs in a Mesh fashion using wireline links. We considered the same number of subnets for the hierarchical architecture as in the WiMesh. We refer to this wireline architecture as HiMesh. To overlay the wireless links on top of this HiMesh architecture the hubs are then equipped with WIs. System-level simulations are used to find the optimal number and location of the WIs among the hubs to maximize the bandwidth of the HiWiMesh. The optimum bandwidth is observed for the configuration with 3 WIs. The HiWiMesh with the T-MAC and D-SAM MAC mechanism is referred to as baseline HiWiMesh and HiWiMesh+D-SAM architecture.

Figure 4.14 shows the percentage gain in peak achievable bandwidth per core and average packet energy for the baseline HiWiMesh and the HiWiMesh+D-SAM architecture with respect to the wireline HiMesh architecture for uniform-random traffic pattern. The single-hop wireless shortcuts in the baseline HiWiMesh enables efficient data transfer between the subnets and improves the performance and energy efficiency by 3.6% and 18% compared to the wireline counterpart. On the other hand,
the gain in performance and packet energy for the HiWiMesh+D-SAM architecture is higher than the baseline HiWiMesh architecture, as the transmission slots are dynamically adjusted based on the high traffic demand of the WIs. This is consistent with our earlier evaluation with WiMesh architecture. Hence, the improvement in performance and energy efficiency for the D-SAM MAC is independent of the underlying architecture.

4.10 Overhead Analysis

From post-synthesis analysis shown in table 4.2 the area of the D-SAM MAC unit is 0.0017mm². Each transceiver occupies an area of 0.17mm² [47] [48]. Therefore the overhead of the dynamic MAC unit is 1% of the transceiver. The longest dimension of the zig-zag antennas is 0.3mm and they occupy a passive area using top-layer
metal traces [32]. For the WiMesh architecture with 64 cores and 8 WIs the total area overhead of the MAC units is only 0.0038% of the 400mm$^2$ chip.
Chapter 5
Conclusion and Future Work

5.1 Conclusion

Wireless interconnection is envisioned as an energy efficient communication backbone for future multicore systems. One of the key aspect for the adoption of such novel interconnect paradigm is the MAC mechanism that ensures the efficient utilization of the wireless channel based on the varying demand of the applications. In this thesis, we propose a proportional, integral, and differential (PID) based prediction model for predicting traffic demands for wireless interfaces, and based on the prediction the dynamic MAC mechanism are able to efficiently adjust the transmission slots to the WIs with spatial and temporal variation of traffic demand through the WIs. Using cycle accurate simulations, we show that the PID prediction based dynamic MAC mechanism improves the performance of a WiNoC architecture compared to a baseline token based MAC for a wide range of synthetic and application specific traffic patterns.

5.2 Future Work

- In this thesis, we have discussed single chip framework, we can investigate the wireless multichip system where wireless can also be used for inter-chip communication.

- In our work only wireless adapt dynamically to variations in traffic, this work
can be further extended to study the dynamic behavior not only in wireless links but also in wireline links.

- Also with WiNoC interconnection further analysis can be done on extending the idea of reusing the WiNoC resources for transporting test data for post-manufacturing testing.
Bibliography


dynamic medium access mechanism,” in Proceedings of the 9th International

[26] V. Vijayakumaran, M. P. Yuvaraj, N. Mansoor, N. Nerurkar, A. Ganguly,
and A. Kwasinski, “Cdma enabled wireless network-on-chip,” ACM Journal on

[27] N. Mansoor, P. J. S. Iruthayaraj, and A. Ganguly, “Design methodology for
a robust and energy-efficient millimeter-wave wireless network-on-chip,” IEEE

and D. Heo, “Design of an energy-efficient cmos-compatible noc architecture
with millimeter-wave wireless interconnects,” IEEE Transactions on Computers,

[29] ———, “Design of an energy-efficient cmos-compatible noc architecture with
millimeter-wave wireless interconnects,” IEEE Transactions on Computers,

and A. Cabellos-Aparicio, “Scalability of broadcast performance in wireless network-
on-chip,” IEEE Transactions on Parallel and Distributed Systems, vol. 27, no. 12,

[31] V. Vijayakumaran, M. P. Yuvaraj, N. Mansoor, N. Nerurkar, A. Ganguly,
http://doi.acm.org/10.1145/2536778

and D. Heo, “Performance evaluation and design trade-offs
http://doi.acm.org/10.1145/2287696.2287706

wireless scalable express channels for network-on-chips (noc) architecture,” in
Proceedings of the 2011 IEEE 19th Annual Symposium on High Performance
Interconnects, ser. HOTI ’11. Washington, DC, USA: IEEE Computer Society,

[34] P. P. Pande, A. Nojeh, and A. Ivanov, “T1b: Wireless noc as interconnection
backbone for multicore chips: Promises and challenges,” in 2014 27th IEEE In-
ternational System-on-Chip Conference (SOCC), Sept 2014, pp. xxxvii–xxxviii.


