國立臺灣大學電機資訊學院電子工程學研究所

碩士論文

Graduate Institute of Electronics Engineering

College of Electrical Engineering & Computer Science

National Taiwan University

Master thesis

一個預測性服務質量機制控管的

雙向通道晶片網路架構設計

Design of an Anticipative QoS Control Bi-directional Network-on-Chip Architecture

林孝恩

Lin, Hsiao-An

指導教授：陳少傑 博士

Advisor: Chen, Sao-Jie, Ph.D.

中華民國 99 年 6 月

June, 2010

授權書

中文簽名頁

英文簽名頁

誌　謝

論文能夠及時完成並順利地取得碩士學位，最重要的必須感謝陳少傑老師在論文撰寫過程中每一個細節所給予的指導和建議，以及口試委員們在學位考試中對於此研究題目所提出的寶貴意見，並使我能夠從更多元的角度去看待問題，為本篇論文增添了更詳備的內容。

在研究所就讀期間受到了博理407實驗室同學及學長姐們的關懷與照顧，其中籃英誠學長的指導以及給予研究方向的提點更是讓我在對研究內容感到困頓時有如在茫茫大海中恰逢明燈。感謝同為97級的幾位夥伴們，在每個需要挑燈夜戰或是倍加奮進時的關頭，彼此的慰問或閒談都是讓生活更增樂趣的良劑。即使將來四散各地，這些有趣回憶也必定會讓人在想起時會心一笑吧。當然也要謝謝學弟妹在口試時周邊細節的妥善處理，讓流程能順利的推進。

最後也是最為重要的，感謝父母以及家人在心靈與物質上都給予我強大的後盾而能無後顧之憂地專心以研究為生活重心，不管是感到挫折時或是身體虛弱病痛時，那最後的避風港總是無條件地讓我這遠方的遊子有個永不熄燈的歸處。

謹以本文獻給所有關心我的人，願與你們分享這份榮耀。

中文部分

摘　要

本文提出一個支援服務質量機制的特性並以預測方式增加效能的雙向通道晶片網路架構，它同時支援了不同服務品質的資料傳輸，使得晶片內部的傳輸效能有所改善。此雙向晶片網路架構允許特定服務品質的資料傳輸時能依其可預期的部份特性以達成傳輸方向預先轉變以及穿透路由器之設計。對於每一個晶片網路的路由器而言，資料通訊的延遲時間以及傳輸吞吐量都受到這個附加的通道靈活性影響而得到更好的效能。這篇論文呈現出一個創新的路由器架構設計以及一個改進控制雙向通道的機制。透過分析可以證明此架構的額外硬體成本是可忽略的。本文利用一個精準時脈週期的測試環境進行模擬，對於在假想的交通型態的傳輸情況下，此雙向通道晶片網路相對於傳統的單向通道架構都能展現出可觀的效能優勢。

關鍵字：晶片網路、路由器、雙向通道、虛擬通道、服務質量、預測性。

貳

目　錄

肆

第一章

簡介

在現今高度進步的積體電路技術下，一個單晶片系統已足夠容納多個矽智財同時運作。隨著電路設計的製程演進，一個晶片的效能瓶頸漸漸從運算時間轉移至不同運作單元之間傳輸的效率。傳統的匯流排架構在功率消耗，傳輸時間，以及系統的擴充性等方面已難以適用於目前多核心的系統。為了能夠達到晶片內部各模組的溝通需求，一種利用網路封包交換技術實現而成的晶片網路傳輸架構(Network-on-Chip)在近年來被提出。

本論文主要提出一個基於可動態調整通道方向的晶片網路架構而更進一步輔以質量服務機制並且擁有更佳效能表現的架構，相對於傳統使用單向通道的做法，雙向通道可以提供額外的彈性使之能夠針對不同的傳輸狀況再加以調整通道方向，因此可以達到更佳的頻寬利用率進而改善整體的傳輸效能。在第二章中，我們將會介紹晶片網路之路由器硬體架構，以及網路服務質量(Quality-of-Service)和快速虛擬通道(Express-Virtual-Channel)的背景知識，第三章描述本論文所提出之預測性支援服務品質之雙向通道晶片網路(Anticipative QoS control BiNoC)的概念。並於第四章中針對此路由器硬體架構做出更詳細的說明，實驗結果及討論將會在第五章呈現，並於第六章作本論文之總結。

第二章

背景知識

一個晶片網路的硬體骨幹是由分布於各個節點上的路由器所組成,封包在實體網路層內透過路由器的傳輸過程中,根據不同的交換方式會產生不一樣的效能特性。本章針對基本路由器的硬體架構以及一種廣泛使用的虛擬通道流量管控(virtual channel flow control)機制做出簡單的介紹，並概觀現今用於網路服務質量控制所提出的各種技術,另外再提供一種可繞過路由器運算管道層的快速傳輸架構簡介，此快速虛擬通道的設計可以使的封包傳輸時的運算有效減少，藉此提升整體效能。

第三章

設計動機

本章描述此篇論文所提出的預測性服務質量控制雙向晶片網路架構，此先前技術利用可動態調整方向的通道進行封包傳輸，可增大網路內頻寬的使用率進而提升整體的傳輸效能。在此架構中我們要試著解決路由器間的方向仲裁機制問題，以及路由器內的冗贅管線運作延遲的問題。

捌

第四章
路由器架構實現

本章針對一個用於雙向晶片網路的虛擬通道路由器硬體架構做出詳細的介紹。利用預測封包特性的技術，路由器利用服務質量的機制來提升網路資源的使用效率及傳輸效能。在改進雙向晶片網路的架構之下，此論文提出了一個可增強服務質量效果的改良機制。

第五章

實驗結果及討論

本章呈現出傳統的單向通道晶片網路以及本論文所提出的路由器內、路由器間之改進在傳輸效能上的比較。實驗使用假想交通型態封包傳送。透過傳輸延遲時間，路由器吞吐量，以及頻寬使用率的數據分析，可以發現封包傳輸的效能在預測性雙向晶片網路的平台上可以獲得更好的結果。最後經由額外硬體成本消耗的評估，證明本架構在可忽略的低成本下可以達到較好的效能表現。

第六章

結論

本論文提出一個基於動態調整通道技術的雙向晶片網路傳輸架構進而輔以整體仲裁機制更有效率的新架構。透過控制通道方向的路由器協定，此架構不需額外的邏輯控制硬體外加在路由器之間並且同時減去傳輸延遲的耗費時間。因此適用於各種型態的非間接網路系統。基於雙向通道網路的概念下，並針對網路服務質量機制做了更進一步的改良,使得高重要性封包的傳輸效能獲得增進並且減少整體運算管線的浪費.

實驗結果證實此晶片網路架構在假想交通型態情況下,表現出相對於傳統架構更高的頻寬利用效率，進而縮短了封包傳送的延遲時間。此優勢特別是在一些通道負載較不平衡的封包分布情形下更為明顯。最後根據硬體面積的分析，本篇提出的雙向通道晶片網路架構能夠在節省電路成本的效益下達成更佳的效能表現。

英文部分

ABSTRACT

A Bidirectional channel Network-on-Chip (BiNoC) architecture with previous direction request and pipeline bypass mechanism is proposed to enhance the performance of on-chip communication while supporting prioritized traffics in the network. The Anticipative QoS controlled BiNoC not only allows each communication channel to be dynamically self-configured to transmit flits in either direction in order to better utilize on-chip hardware resources but also enhances the latency performance by using penetration and observing previous direction request. This added flexibility promises better bandwidth utilization, lower packet delivery latency, and makes high priority packet be served with better guaranteed performance. In this Thesis, an improved bi-directional on-chip router architecture supporting the hybrid bypass mechanism is presented. It is shown that the associated hardware overhead is negligible. Cycle-accurate simulations run on this AQ-BiNoC network under synthetic traffics demonstrate consistent and significant performance advantage over the conventional mesh-grid BiNoC architecture.

Keywords: Network-on-Chip, Router, Bidirectional Channel, Virtual Channel, Quality-of-Service, Anticipative.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

Increasing transistor density, higher and higher operating frequency, and shorter product life cycle characterize present semiconductor industry trend. Under these conditions, designers are developing ICs that integrate complex heterogeneous functional elements into a single chip, known as a System-on-Chip (SoC). SoC design is based on intellectual property (IP) cores reuse. It defines a core as predesigned, pre-verified silicon circuit block that can be used to build a larger or more complex application on a semiconductor chip. An SoC must include an interconnection architecture and interfaces to connect peripheral devices. The interconnection architecture includes physical interfaces and communication mechanisms, to allow communication between SoC components. The performance of most digital systems today is limited by the speed of their communication or interconnection, not by their logic or memory. As technology improves, memory and processors become smaller, faster, and less expensive. The pin density and wiring density that govern interconnections between components are scaling at a slower rate than the components themselves. Also, the frequency of communication between components is lagging far beyond the clock rates of modern processors. To meet the need of new generation system design, Network-on-Chip (NoC) has been announced in the past decade.

1.1 Current Trend of Network-on-Chip

In a high-end system, most of the power is used to drive wires and most of the clock cycle is spent on wire delay instead of circuit delay. As system complexity grows more and more rapidly, the interconnection performance between cores become much important than the traditional single core performance. Multi-core chip has much more cycle propagation across chip. The dedicated wiring communication may cause effective transmission but also bring area cost. The bus system has successfully been used to a less cost and more efficient connection between cores and peripheral devices like I/O or memory. But there still exists some problem when system scale grows up. When the communication between cores is not only within the distance of a few devices in local area but also on the whole system containing tens, or hundreds of cores, the wiring using a bus or switch architecture cannot satisfy the target at all. That is why network-on-chip is proposed in past few years and continued to be the focus of all people who work on SoC [1,2].

1.2 Concept of Network-on-Chip

Network-on-Chip is a platform-based interconnection design. The concepts of Network-on-Chip were derived from many different fields, but the most important one is to provide a unique solution to the specific problem of on-chip communication. The topic of NoC can range from the bottom level of physical wire interconnection to the highest level of software application. The major work of this Thesis is focused on the level of router architecture.

1.2.1 Communication Layers in a Network-on-Chip Design

The Network-on-Chip design includes four layers. The physical layer determines the number and characteristic of wires which connect resource and switches [3].

The data-link layer defines the protocol to transmit a cell between a resource and a switch or between two switches. Both the data link and physical layer are dependent on the technology.

The network layer defines how a packet is transmitted over the network from an arbitrary sender to an arbitrary receiver directed by the receiver’s network address. It consists of a routing algorithm and a flow-control mechanism. The routing algorithm determines the strategy where a packet should head for by its source, destination and the routing elements. The flow-control mechanism decides whether the connection between nodes is successful or not. The research of this Thesis is focused on this layer, which is also dependent on technology.

The transport layer is technology independent. The message size can be variable. The interface has to pack transport layer messages into network layer packets.

1.2.2 Network-on-Chip Architecture

The network architecture specifies the network topology and physical organization of an interconnection network. Like all the digital system designs, we should determine the constraints and build different solutions to the problems that we want to solve [4,5,6] .

First we have to decide the topology of a network by considering the roadmap and path that we want to implement and to satisfy the whole system transmission condition. It is always better to use a general-purpose topology than to design a unique topology matched to a specific problem. Tree, Butterfly, Mesh, and Torus networks are examples of network topologies used in interconnect backbone. Two-dimensional mesh is the most popular topology for current NoC design because of its simplicity and regularity.

After the topology is selected, the interconnection network can be built up by deploying all network elements. The designer must choose what to put in an NoC as resources, how to map functionality into those resources, and how to validate the decisions. Just like building a city block, we have to choose which block is downtown, which block is the residential area, and where has light traffic or how to set one-way street, *etc*.

Except IP placement, a typical NoC consists of routers, network interfaces, and physical channels that comprise the communication architecture. The channels are sets of wires which connect each pair of neighbor routers. Channels enable packet to be delivered through routers. Network Interface is the interface between transport and network layers, it can pack the core message or data to packets. The packet is an actual transport data type in NoC. Every IP is attached to an interface which connects the IP to the router where the IP belongs to. Fig. 1-1 shows a typical NoC architecture in a 2-D mesh 3×3 mesh technology. When a message was sent from a source IP, it will be packed to a packet. The packet is delivered through routers until it reaches the destination. And the destination interface will unpack it into message for the local IP at destination. For every packet entering a router, the packet data will be written into the input buffer of the port where it entered. And the control logic will detect its routing path and the channel which it should head for. After winning arbitration, the packet can leave the router from the output port which direction is decided from the routing information. Otherwise, the packet will be stored in the buffer until it is granted to leave by later arbitration.

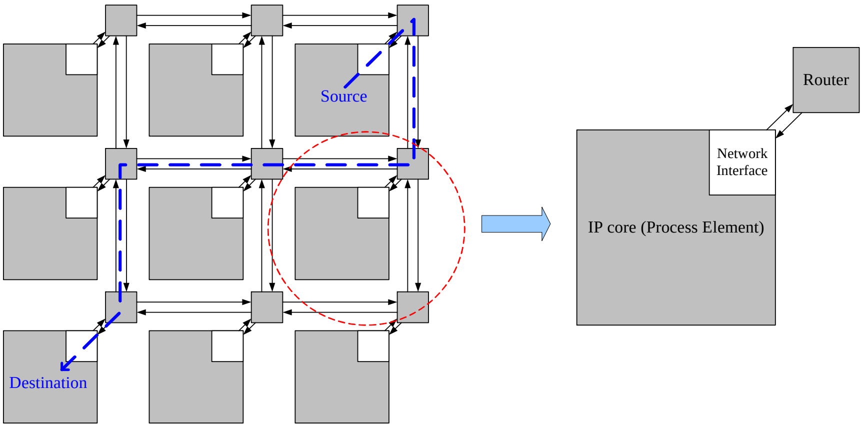


Fig. 1-1. Typical NoC Architecture with a Mesh Topology.

1.3 Network Basic

To satisfy the performance specification of a particular application, the designer working within technology constraints has to implement a routing algorithm and a flow control based on the topology of the desired network.

1.3.1 Routing

When the topology has been chosen, there are several possible paths from a source terminal node to a destination terminal node. The routing method employed by a network determines the path taken by a packet according to the routing information of the packet. The routing information varies with different routing algorithms. Some routing algorithms choose paths based on the as-fast-as-possible way such that they just route message to the destination using the shortest paths. Other ones try to balance the traffic loading in a network such that they can gather more information like buffer usage amount or the traffic history to route packets. However, to keep a routing algorithm simple and fast while considering every details of the network is too hard to optimize, because a routing logic is just a logic circuit in chip, not a program.

1.3.2 Flow Control

Flow control manages the allocation of resources to packets as they progress along their route. The most important resources in interconnection networks are channels and buffers. A channel is used for transporting packets between routers. Buffers are storage implemented within the routers, such as registers or memories, and they allow packets to be held temporarily at the nodes. The flow control strategy must avoid resource conflict that can hold a channel idle. A good flow-control strategy is fair and deadlock-free. An unfair flow-control strategy may cause a packet to wait indefinitely. Deadlock is a situation that occurs when a cycle of packets are waiting for one another to release resources, and hence are blocked indefinitely.

Buffer flow control is the main stream in present networks. Every router is implemented with a fixed amount of buffers to store packets temporarily. The way of buffer flow control can be categorized into two different types by the storage data size: One is packet buffer flow control, and another is flits (flow control digits) buffer control. If we allocate bandwidth and buffers in units of packets, we have a packet flow control.

Packet flow control can be defined as store-and-forward flow control and cut-through flow control. With store-and-forward flow control, each node along a route path waits until a packet has been completely received and then forwards the packet to next node. The packet must be allocated a packet size buffer on the far side of the channel and exclusively use the channel. Cut-through flow control forwards a packet as soon as the header of packet is received and resources are acquired without waiting for the entire packet to be received. At this point, cut-through flow control looks like an ideal method. It gives very high channel utilization by using buffers to decouple channel allocation. It also achieves low latency by forwarding packets as soon as possible. However, there are still weak points. First, allocating buffers in units of packets is very inefficient in buffer management. Second, allocation channels in units of packets also increases the contention latency. If a packet is allocated with a channel and the channel has contention, other packets without allocating the channel have to wait for the whole packet transmitting. The solution is allocating resources in smaller units. That is why the flit-buffer flow control is used by most of network-on-chip architectures.

In order to avoid excess of buffer area counting, flit-buffer flow control is used to segment a packet into flits. The wormhole flow control is the most representative of flit-buffer flow control. It operates like cut-through, but with channels and buffers allocated to flits rather than packets. When the head flit of a packet arrives at node, it must request and acquire three channel resources for the packet, one flit buffer and one flit of channel bandwidth before it can be forwarded to the next node along a route. Body flits of a packet use the channel acquired by the head flit and hence need only acquire a flit buffer and a flit of channel bandwidth to advance. The tail flit of a packet is handled like a body flit but also releases the channel after its passing. This kind of flit-buffer flow control is more and more common in on-chip networks because of its area efficiency in buffer memory usage.

1.3.3 Performance Evaluation

To distinguish whether a system is good or not, performance evaluation is important to decide the specification of an NoC architecture to implement. The first to consider about is throughput which determines the data transmission rate of a network. The second performance variable is latency which represents the time consumption of packet.

1.3.3.1 Throughput

Throughput is the rate at which packets are delivered by the network for a particular traffic pattern. It is measured by counting the packets that arrive at destinations over a time interval for each flow (source-destination pair) in a given traffic pattern and computing from these flows the fraction of the traffic pattern delivered. It is contrasted with demand, or offered traffic, which is the rate at which packets are generated by the packet source. As shown in Fig. 1-2, at traffic levels less than saturation, the throughput equals the demand and the curve is a straight line. Continuing to increase the offered traffic, we eventually reach saturation, the highest level of demand for which throughput equals demand. We typically present throughput as a fraction of network capacity. This gives a more intuitive understanding of the performance of the network and allows direct comparison between networks of different specifications.

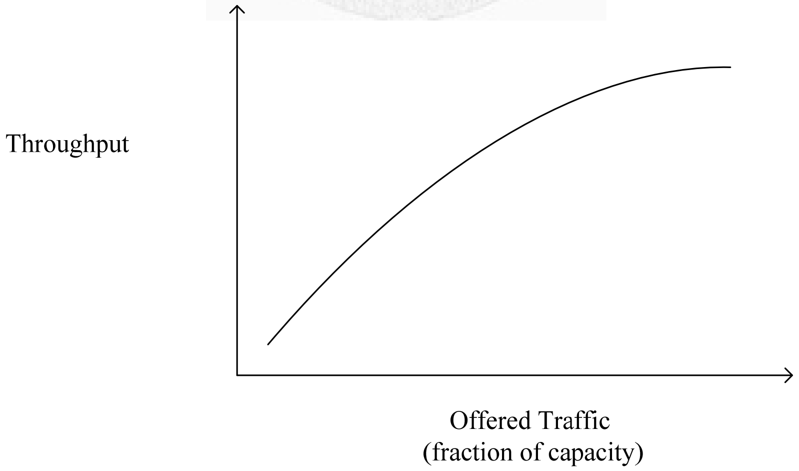


Fig. 1-2. Throughput vs. Offered Traffic in a Network.

1.3.3.2 Latency

Performance of a network can be generally described by a curve that depicts the relationship between average latency and offered traffic as shown in Fig. 1-3. Latency is the time required for a packet to traverse the network from source to destination. Latency vs. offered traffic graph shares a distinctive shape that starts at the horizontal asymptote of zero-load latency and slopes upward to the vertical asymptote of saturation throughput. At low offered traffic, latency approaches zero-load latency. Zero-load latency gives a latency lower bound. The assumption is that a packet does not have to contend for network resources with other packets. The zero-load latency T_0 can be separated into two components.

$$T_0 = H_{ave}\, t_r + \frac{L}{b} \qquad (1.1)$$

The first term is the average hops count H_{ave} and the router delay t_r. The second term represents the serialization latency, which is the time for one packet of length L to across a channel of bandwidth b.

Latency grows to infinity at the saturation throughput λs which is affected by network topology, routing algorithm, and flow control. The goal of network architecture usually is to extend the packet injection rate saturation point as much as possible.

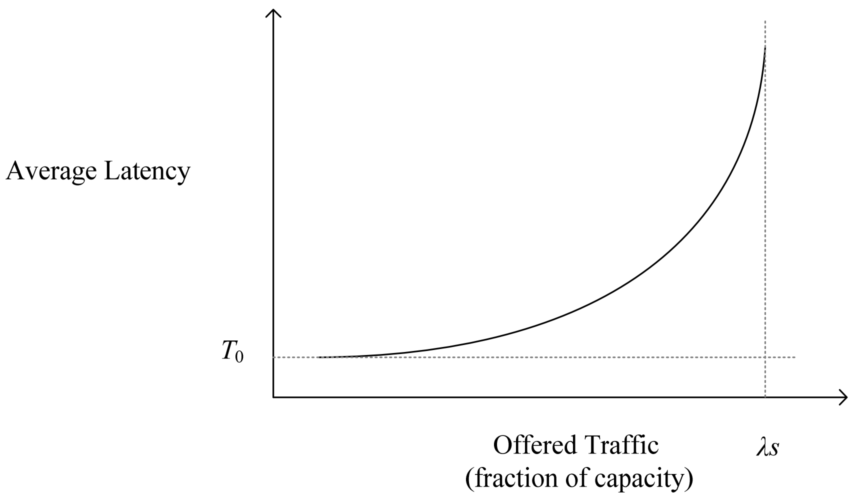


Fig. 1-3. Latency vs. Offered Traffic in a Network.

1.4 Quality-of-Service

In some applications of interconnect networks, it is useful to divide network traffic into a number of classes to efficiently manage the allocation of resources to packets. Allocating resources based on different classes allow us to prioritize services, more important classes get a higher level of services. With prioritized services, we may give packets of higher class a strict priority in allocation of buffers and channels over packets of lower class. Traffic classes fall into two broad categories: GS (guaranteed service) and BE (best effort). GS classes are guaranteed with a certain level of performance. The performance can be achieved as long as the clients comply with the service contrast between network and client. In contrast of GS, the network makes no strong promise about BE packets. Depending on the network, BE packets may have arbitrary delay or even be dropped. The network will simply make the best effort to deliver the packet to its destination.

In general computer networks, there are various algorithms targeting on the performance between GS and BE traffics. But not every algorithm can fit in the NoC architecture because there are many differences between computer network and network-on-chip. The state-of-the-art QoS mechanism for NoC can be categorized into connection-oriented and connection-less. In the connection-oriented scheme, the connection path between source and destination is built and preset before the packets are actually injected into network. It is reliable to satisfy QoS requirement. But it comes with great hardware cost because the resource reservation and allocation take huge control and storage elements.

1.5 Thesis Organization

The rest of this Thesis is organized as follows. In Chapter 2, we take a brief description about each functional component in a typical router and the characteristic and architecture of BiNoC with QoS. Chapter 3 covers the main concept of out proposed Anticipative QoS controlled Bi-directional NoC (AQ-BiNoC) architecture that adopts speculative channel direction switch functionality and high priority packet penetration ability to enhance the communication performance. In Chapter 4, the implementation of AQ-BiNoC router architecture is presented. Based on the concept of the bidirectional network, we propose a whole-new direction switch strategy which is similar to the bypass pipeline to improve the QoS guarantee on packet latency. Experimental results and discussion are presented in Chapter 5. Finally, Chapter 6 concludes the Thesis.

CHAPTER 2

BACKGROUND KNOWLEDGE

In this chapter, we will briefly review some concepts on the design of NoC and BiNoC router architectures, a popular flow control method using virtual channels, and the QoS mechanisms implemented in BiNoC which provide both guarantee service (GS) and best effort (BE) traffic classes. Several distinguished works in these research topics will be noted and introduced.

2.1 Design of Router Architecture

When a packet travels from source to destination, routers are the intermediate nodes on the path where the packet must pass through. The datapath of the router handles the storage and movement of a packet payload, and it consists of buffers, switches, function units, and control logic to implement the routing and flow control functions. Fig. 2-1 shows the block diagram of a typical virtual channel NoC router. It is essential to understand the network function and its performance by observing the router design. Modern routers are pipelined at the flit level. Head flits proceed through pipeline stages that perform routing and virtual channel allocation and all flits pass through switch allocation and switch traversal stages. Sometimes stalls occur in the pipeline stage, which means that the flit routing cannot be completed in current cycle.

Each flit of a packet arrives at the input unit of a router. The input unit contains a set of flit buffers to hold arriving flits until they can be forwarded and also maintains the state of each virtual channel associated with that input link. To begin transporting a packet, path routing must first be handled to decide the output direction to which the packet can be forwarded. With a given output port, the packet requests the output virtual channel from the virtual-channel allocation stage. Once a route has been decided and a virtual channel allocated, each flit of the packet is forwarded over the virtual channel by allocating a time slot on the switch and the output channel using the switch allocator and forwarding the flit to the appropriate output unit during this time slot. Finally the output unit forwards the flit to the next router in the packet's routing path.

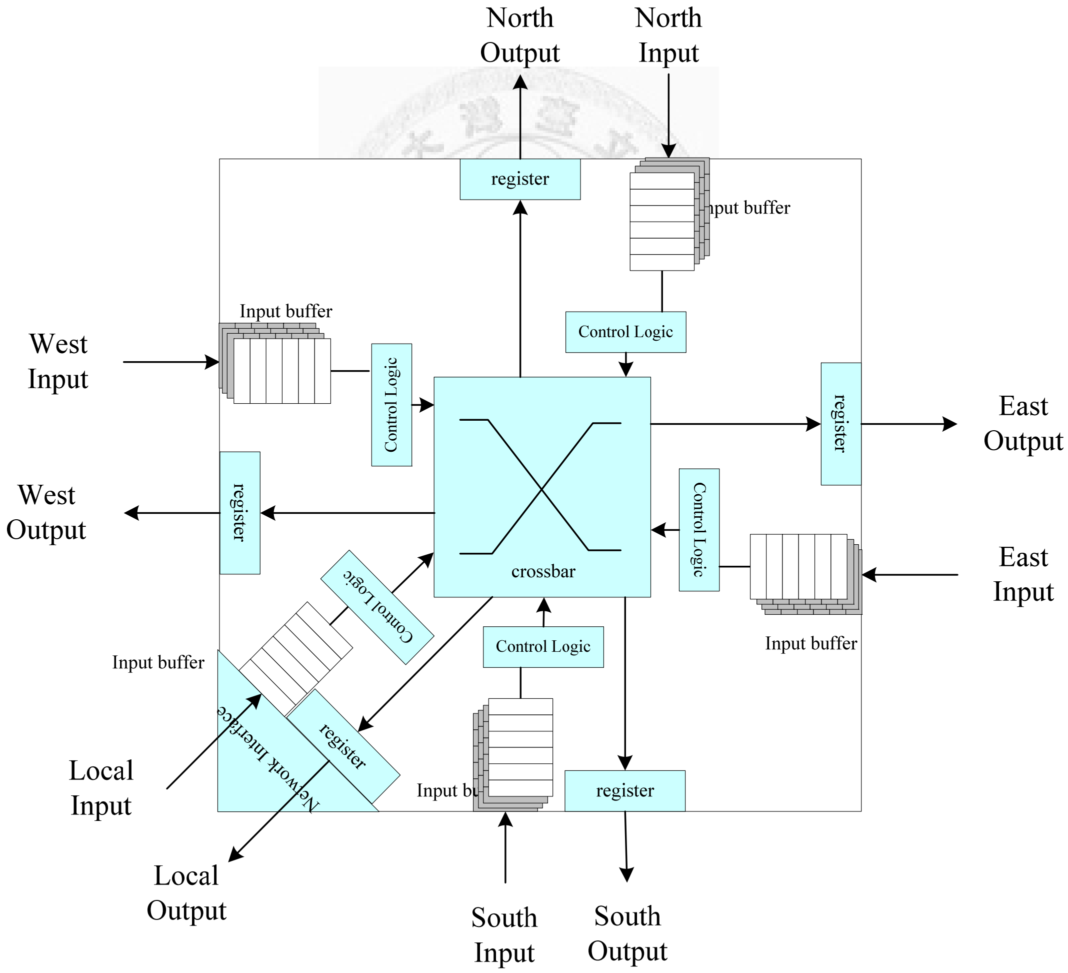
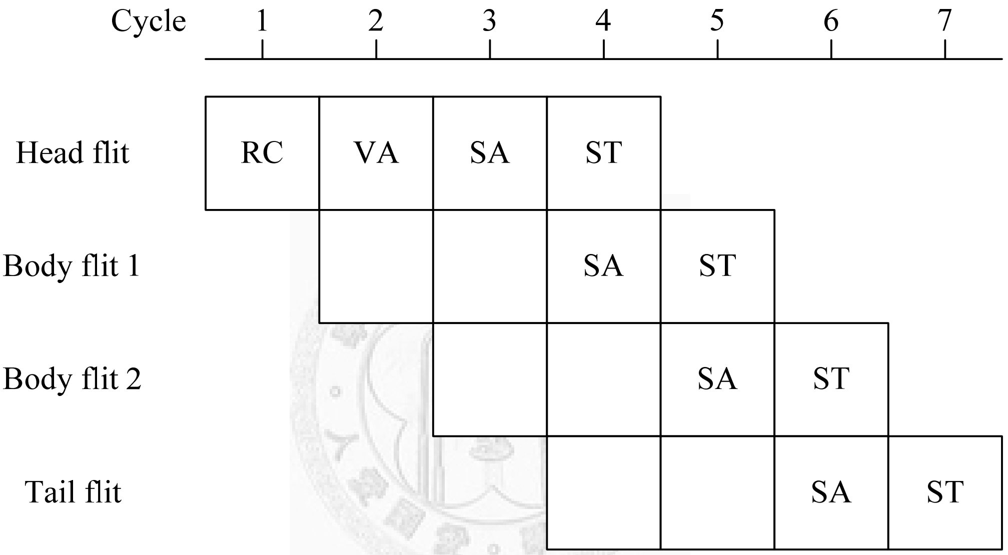


Fig. 2-1. Typical Virtual-Channels Router used in a Mesh Network.

2.1.1 Pipeline of Router Stages

Figure 2-2 shows the pipelined stages for routing a packet. Each flit of a packet proceeds through the stages of routing computation (RC), virtual channel allocation (VA), switch allocation (SA), and switch traversal (ST) [7]. The RC and VA stages are only performed for the head flit. Body flits passing through these control stages do not need the RC and VA stages. The SA and ST stages operate on every flit of the packet.



Cycle	1	2	3	4	5	6	7
Head flit	RC	VA	SA	ST			
Body flit 1				SA	ST		
Body flit 2					SA	ST	
Tail flit						SA	ST

Fig. 2-2. Pipelined Package Traversal Stages.

2.1.2 Virtual-Channel Flow Control

The performance of interconnection network is limited to only part of the network capacity because of coupling resource allocation. The resources include both buffer and channel. In wormhole flow control, a single buffer queue is associated with one channel in the router. This coupling type makes control logic simple and small, but also causes bandwidth utilization problem by making some useable channel idle.

As illustrated in Fig. 2-3, the path of packet A uses channel x and channel y but it fails to grant channel x. The path of packet B uses channel x and channel z. When packet B enters router 2 and tries to use channel z to leave router 2, the channel z is blocked. And the idle channel y is wasting its bandwidth. Packet A is unable to use channel y because it cannot attain channel x held by packet B. This is called the head-of-line (HOL) problem.

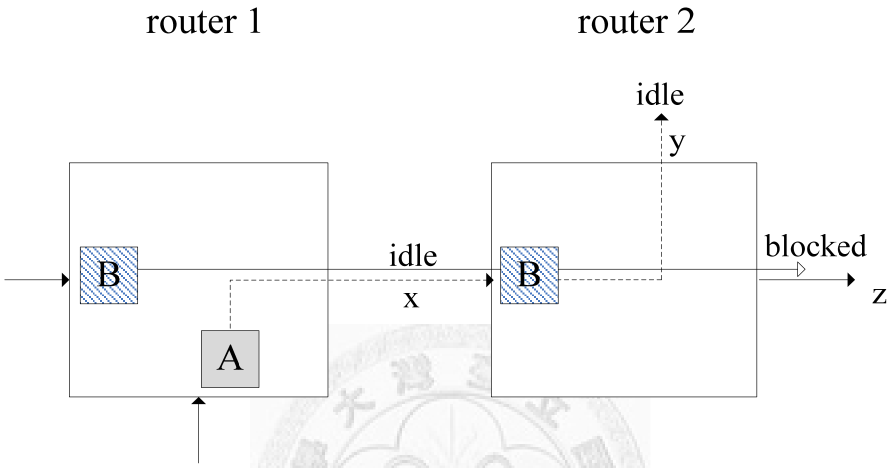


Fig. 2-3. Blocking Problem in Wormhole Routers.

Virtual-Channel flow control [8,9] associates several virtual channels to a single physical channel, thus it overcomes the blocking problem of wormhole flow control by allowing another packet to use the channel bandwidth that would otherwise be idle when the current packet is blocked. As in wormhole, an arriving head flit must allocate a virtual channel, a downstream buffer, and a channel bandwidth to advance. Subsequent body flits from the packet use the virtual channel allocated by the header and still must allocate a flit buffer and a channel bandwidth. Virtual channels allow packets to pass blocked packets, making use of idle channel bandwidth. As illustrated in Fig. 2-4, the same case in Fig. 2-3, the difference is the input of routers is associated with two virtual channels to store packets A and B at the same time. It allows one physical channel to be shared by several VCs. This can decouple the allocation of buffers from channels. When channel z is blocked, channel x still can be allocated to packet A and it can bypass the blocked packet B buffer, just like an extra lane in street. Thus virtual channels can enhance bandwidth utilization [10].

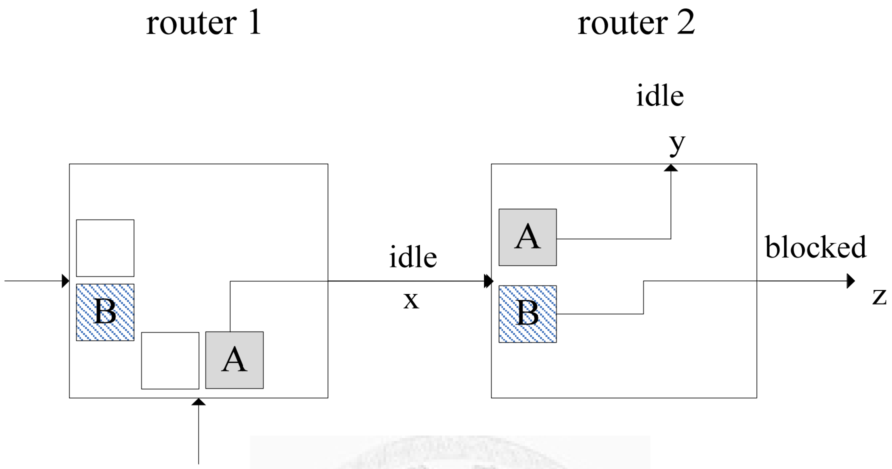


Fig. 2-4. Blocking Problem solved by Virtual Channels.

To implement virtual channels on a typical wormhole router, we need to partition the single-queue buffer into multi-queues at each input port and use an additional VC allocator. Virtual Channels share one physical channel by operating time-multiplexing transmission at each cycle.

2.2 Quality-of-Service in Network-on-Chip

Unlike computer networks which are built for on-going expansion, future growth and standards compatibility, on-chip networks can be designed and customized for a priority-known set of computing resources, given pre-characterized traffic patterns among them. These imply that various components of the network architecture including addressing fields and QoS classification can be modified between implementations. Moreover, placement of the computing resources can be made simultaneously within the network. Dynamic changes of links (link upgrades or failures) are not expected on-chip. But highly reliable link operation has been assumed in the early generations of NoCs [11].

The principal goal of an on-chip interconnection network is to satisfy all communication demands of heterogeneous modules within the chip. In order to manage on-chip network resources adequately, traffic flow can be categorized into guaranteed service (GS) and best effort (BE) classes [12,13]. GS traffics are often used in some timing critical signals or data streams such as the interrupt signal of processors and multimedia application. In contrast to GS traffics, BE traffics, which are generally applied in non-critical traffic flows, ensure transmission correctness and could only get the grant to use the bandwidth that GS traffic does not need. Various algorithms facilitating the network performance among GS and BE traffics have been proposed for general computer network. However, they are not appropriate for on-chip communication since the hardware complexity and computation period must be considered as critical constraints [14]. There are some different types of guaranteed services implemented on a network. Referring to the state-of-the-art QoS mechanisms for NoCs, they can be categorized into two types of schemes: connection-oriented (circuit-switching) and connection-less (packet-switching).

In connection-oriented schemes, guaranteed service packets traverse on some particular channels or buffers that were previously reserved for them. Specifically, the connection path between the source and destination pair of GS packets is built at the time before they are injected onto the network. The control flow of a GS traffic in a connection-oriented scheme is similar to that in conventional circuit switching techniques. That is the origin of its name. There are several reservation approaches to be used. The reservation can be implemented on time, space [15], or both time and space [14,16]. Connection-oriented scheme is a reliable method to achieve GS communications in which the QoS guarantee can be up to 100%. However, it comes with the greater hardware overhead in the control and storage of resource reservations. Moreover, the setup phase of guaranteed traffic presents a timing overhead, which causes this scheme not efficient in a non-deterministic application.

Connection-less scheme is an alternative way to implement QoS mechanism in NoCs. Different from the connection-oriented schemes, connection-less schemes do not execute any resource reservation. It could not provide 100% guarantees because the maximum packet latency is still not predictable. The GS traffic is guaranteed in a relative fashion in a connection-less scheme by prioritizing each type of traffic flow. When two flits in different types are presenting on the same channel simultaneously, the higher prioritized flit can interrupt the lower one and traverse this channel antecedently [16]. This approach gets the benefits of saving hardware cost and control overhead. Also, connection-less schemes can achieve better bandwidth utilization because all traffics are allocated with network resources dynamically. With the consideration of performance requirements for each service level, a network designer can select an appropriate bandwidth implemented in an NoC to both meet the QoS constraints and save the wiring cost [17, 18].

As for the BE service classes, they could get the grant to use the bandwidth only at the time that guaranteed service does not use. An important task of that is to improve the channel utilization in the network. In order to let the communication smoother, resources allocation between BE packets should be as fair as possible.

Although connection-oriented communication guarantees tight bounds for several traffic parameters, an erroneous decision of resource reservation might cause an unexpected performance penalty. While in a connection-less network, a non-optimal priority assignment has less degradation of throughput though it provides coarse QoS support. There are some quantitative modeling and comparison of these two schemes by running simulations on a multimedia processing platform [19]. The results show that under a variable-bit-rate application, connection-less technique provides a better performance in terms of the end-to-end packet delay. As pointed out in [18], guaranteed services require resource reservation for the worst-case in a connection-oriented network even when the average is much lower, which causes a lot of wasted resources in the network. These comparisons can help us to design an application-specific NoC using a suitable QoS scheme.

2.3 QoS-aware BiNoC

Typical NoC owns channels with fixed direction. It is observed that under various traffics, one channel may be full of traffic and another is empty and idle. The basic concept of BiNoC [20] is switching channel direction to achieve better bandwidth utilization. However, BiNoC architecture only provides good latency results for BE traffic because of its channel utilization flexibility, but it is incapable to support critical communication guarantees that are much more important for real world applications. Fig. 2-5 shows the basic architecture of a QoS-aware BiNoC router [21]. There are four virtual channels in each input port. The main concepts of the QoS-aware BiNoC are described in the following sub-sections.

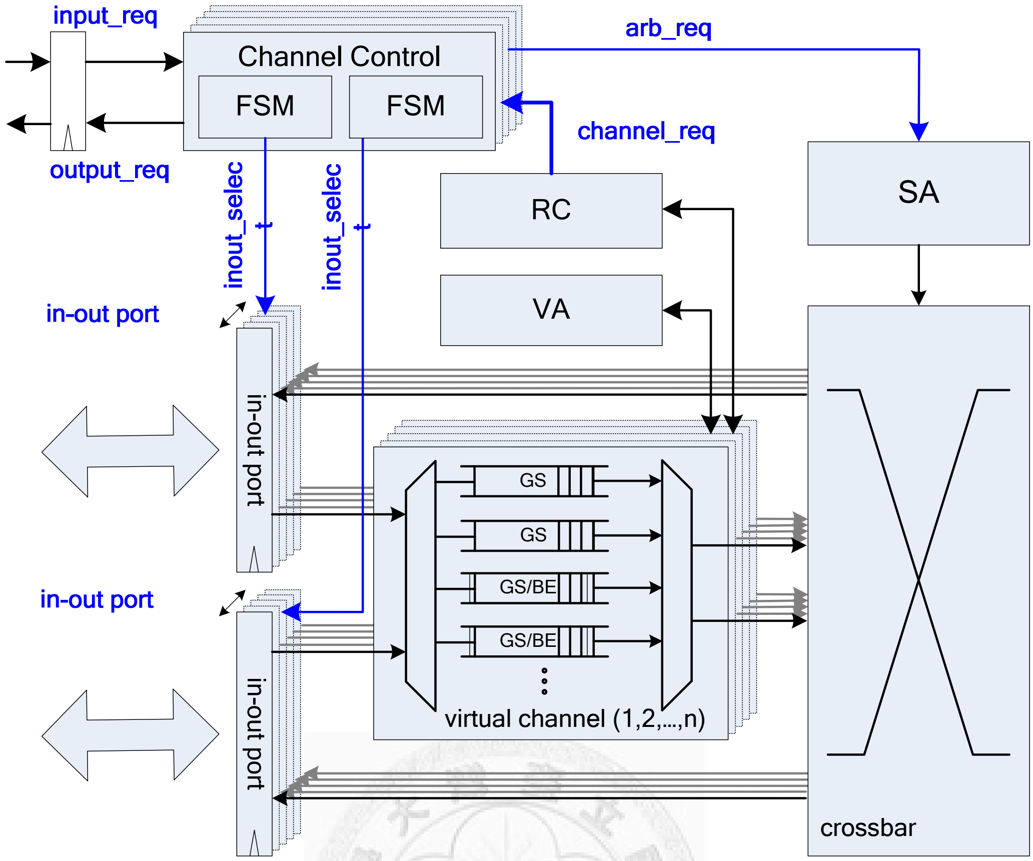


Fig. 2-5. QoS-aware BiNoC Architecture.

2.3.1 Prioritized Virtual Channel Management and Inter-router Arbitration

A flexible VC management mechanism is applied to enhance the authority of the GS packets in this design as shown in Fig 2-5. In each input port of the router, a four-entry prioritized virtual channel module is implemented. Two of the virtual channels are specifically designed for GS packets but the other two virtual channels can be utilized by both GS and BE packets. The two specific virtual channels for GS packet can reduce the blocking probability of GS packets in the VA stage. The arbitration is applied to the prioritized virtual channels at the SA stage according to the QoS requirements of traffic flows.

2.3.2 Inter-Router Transmission Scheme

As Fig. 2-6 shows, the inter-router transmission scheme includes a finite state machine and two input/output signals each side. The FSMs are driven by the SA stage needing and another side's requests. The *output_req_BE/GS* signal rises when a BE/GS packet at the SA stage requests the direction. Another side will receive the corresponding *input_req_BE/GS* signal to drive FSM.

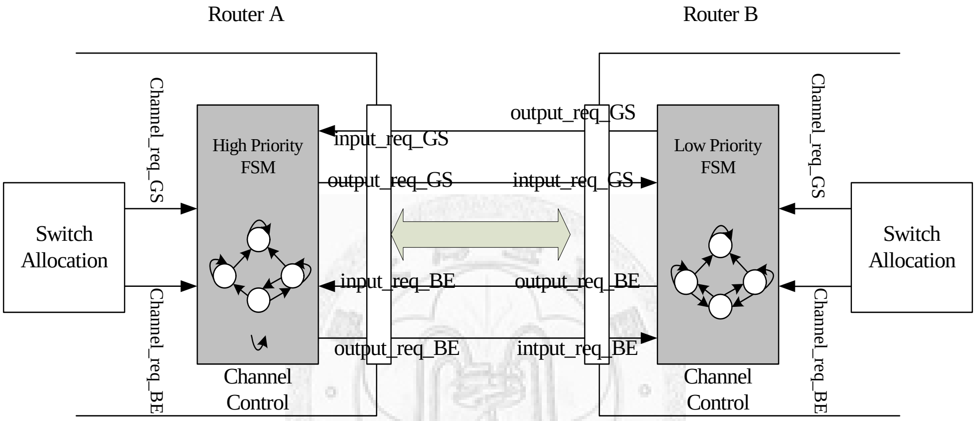


Fig. 2-6. Inter-Router Transmission Scheme.

2.3.3 Bi-directional Channel Routing Direction Control

The states and control signals of the High-priority and Low-priority channel control FSMs are shown in Fig. 2-7. The FSM includes four states: *Free, GS_wait, BE_wait* and *idle* which are defined as follows:

- *Free* State: The channel is ready to output data to another connection router.
- *Idle* State: The channel is available to input data from another connection router.
- *GS_wait* State: An intermediate state for the GS packet to transform from the *idle* state to a *free* state. It is also the temporary state from input direction to output direction.
- *BE_wait* State: An intermediate state for BE packet to transform from the *idle*

state to a *free* state. It is also the temporary state from input direction to output direction.

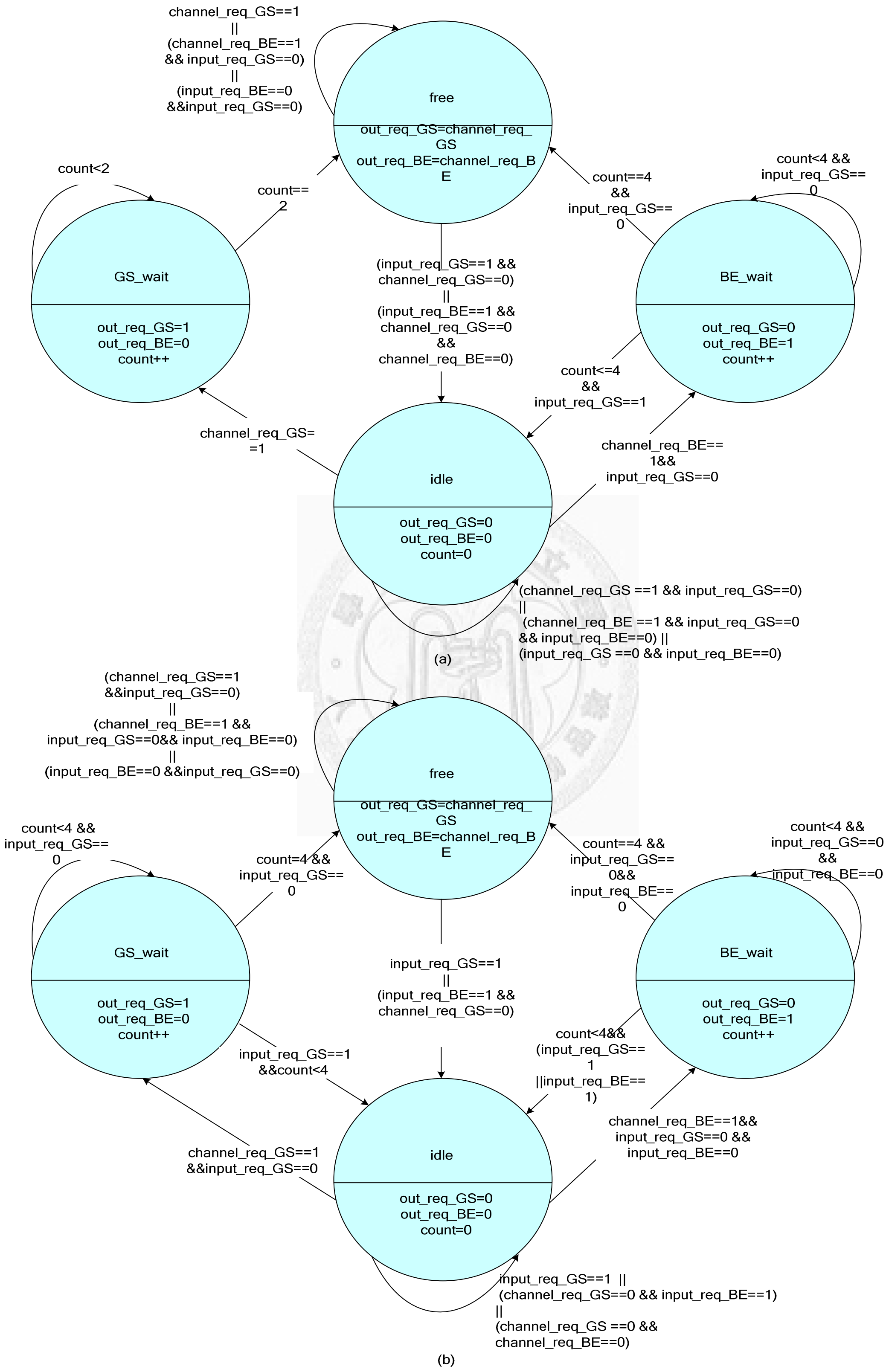


Fig. 2-7. Finite State Machine used in QoS-aware BiNoC.

When the state of FSM is *idle*, it will remain at this state only if there is no any data needed to transmit outbound, or the input request from another connection router has a higher priority than the local channel output request. Moreover, when the state of FSM is *idle*, as soon as a GS channel request from the input unit is received, the state will be triggered into a *GS_wait* state. And if a BE channel request is received without any GS request triggering, the state will change into a *BE_wait* state.

At each wait state, the output request will be pulled up and a counter will be counted up. For the GS request from the port of a High-priority FSM as shown in Fig. 2-7(a), it will definitely take the channel authority after two-cycle waiting for the *GS_wait* state. When the counter reaches two, the High-priority FSM returns to *free* state and starts data transmission. At the *BE_wait* state, the waiting process is four cycles because there may be higher priority request from the neighbor router to interrupt it, and change it into a *free* state.

The Low-priority FSM as shown Fig. 2-7(b) will be initialized at an *idle* state with output requests disabled. If the connection router releases the channel direction, the Low-priority FSM will have chance to enter *GS_wait* or *BE_wait* stage depending on the request priority of traffic. Being a low priority FSM, it needs also four cycles in the wait stage because the high priority FSM can interrupt the wait stage of a low priority FSM by any channel request. After experiencing four cycles without any input request, the low priority FSM will enter the free state and begin the output transmission.

2.4 Express Virtual Channel NoC

In the router pipeline bypass discussion, EVC (express virtual channel) is the first well-known architecture to show the performance of bypass [22]. It stores the bypass packets with an additional buffer which is called express virtual channel. The packets pipeline works as the pipeline baseline router where packets can bypass router pipeline under some special condition. EVC-based flow controls the delay and energy of a dedicated link by allowing packets to virtually bypass intermediate routers along pre-defined virtual express paths between pairs of nodes. Thus, EVCs allow packets to skip the entire router pipeline at intermediate nodes and approach the energy and delay of a dedicated wire interconnect. Intuitively, this is achieved by statically designating a set of EVCs at each router that always connect nodes A and B which are k hops away, and prioritizing EVCs over normal virtual channels (NVCs).

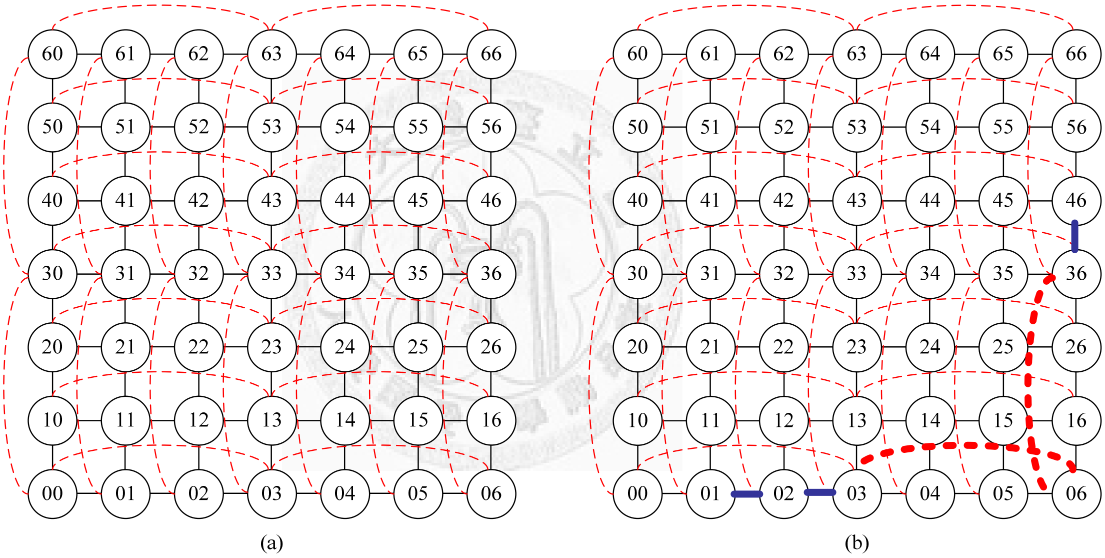


Fig. 2-8. EVC Network and its Packet Routing.

Figure 2-8(a) shows an EVC network, the red dotted lines represent EVCs available among the two routers. The traversal example is illustrated in Fig. 2-8(b), the data is first transmitted from 01 to 03 in a normal way. And the 03 to 06 and 06 to 36 paths are both transmitted through an express virtual channel. The pipeline stages in the router, (04, 05) and (16, 26), are bypassed. Thus the latency is obviously decreased.

The implementation of an EVC needs some modifications of buffers, routing, and allocation. The bypass is simple because when a packet is available to bypass a router, the data will not be written into a buffer but only be stored into a latch and straight forwarded. Moreover, the routing algorithm will be more complex because the router has to tell which path packet is available to use the bypass function. The EVC allocator is responsible for allocating finite resources like destination virtual channel buffers and link transmission authority to those packets which request EVC services.

CHAPTER 3
MOTIVATION

In the QoS NoC with a connection-less scheme, the GS packets are always served with higher priority when they meet congestion and competition. However the GS packets still have an upper bound of performance in most of NoC architectures. Even in the QoS-aware BiNoC that we are concerned about, the performance of GS packets still have space to improve. Reducing latency of GS packets means not only increasing the throughput of GS packets but also supporting more traffic under the same injection rate. Under a reasonable GS traffic utilization, the GS packets meet competition with only one GS packet; therefore, it can be recognized as an absolutely granted request at most of cases. This useful characteristic can be used to solve many problems in a QoS-aware BiNoC architecture.

3.1 Problem Description

BiNoC is a link direction self-configurable network on-chip design. By the link flexibility provided, it takes less overhead and achieves better performance and utilization. Using this concept, BiNoC explores similar idea as a mechanism to decrease the intermittent traffic congestion in an NoC communication backbone, and hence to enhance its overall performance. But there is still a problem in direction-switched BiNoC. As shown in Fig 3-1, the wait stage causes a channel idle and additional time consuming for direction switching. The pipeline stage executes Buffer Writing at T=0, Routing Computation at T=1, Virtual Allocation at T=2, and Switch Allocation at T=3. But when the direction is detected reverse at the SA stage, additional cycles are needed to wait the channel switch to the side that the packet requests. The wait state of FSM is set because there is propagation delay between two routers. For example, as shown in Fig. 3-1 at T=4, there may be data at another side being sent to the local router. If we just turn the channel direction reversely, the data will be lost, since the try-state buffer acts on one direction at one time. This waiting behavior results in a stall in the pipeline and inefficient transmission, because the channel will be idle if there is no other data needed to transmit at T=4 from the neighbor router.

Another phenomenon is illustrated in Fig. 3-2. In every router, there is BW, RC, VA, SA, ST and OP stages to proceed. For example, there are three hops to transmit, then it causes $3 \times (5 + 1) = 18$ cycles to propagate the data for three hops. That is the reason why most of pipeline router architectures usually have zero-load latency defined by:

$$\text{Zero load latency} = \text{Hops count} \times \text{pipeline stages} \tag{3.1}$$

However, the actual pipeline stages count usually can be more than the pipeline stages because of stall. In a QoS scheme, there is a unique characteristic in the GS packet traversal, that is, the GS packet usually grants the allocation competition in a routing path. It represents that most of the allocation pipeline stages like Switch Allocation or Virtual Channel Allocation are waste in time because the GS packets take a must-winning competition. As a result, we try to reduce these ignorable stage cycles to decrease packet latency.

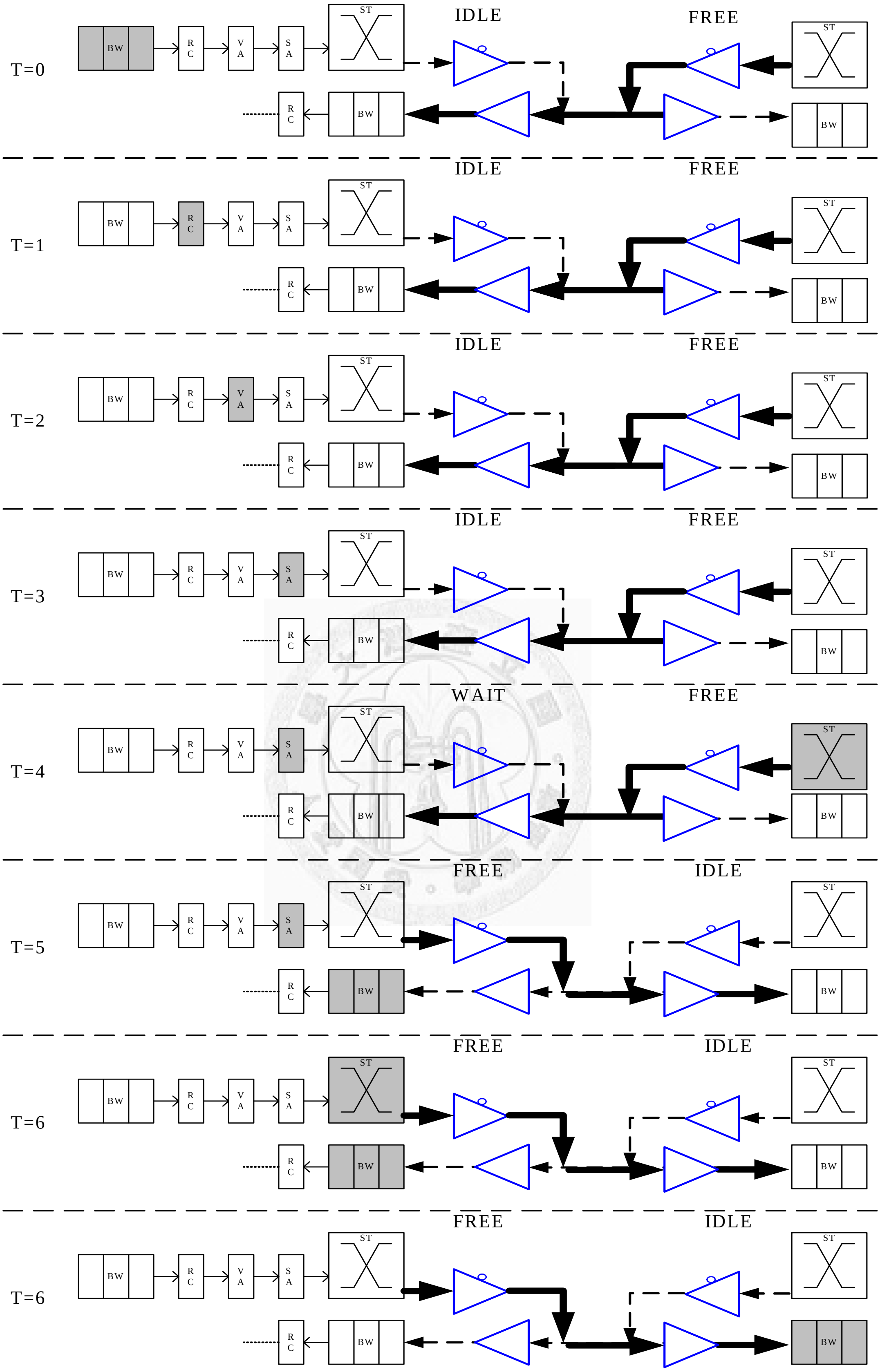


Fig. 3-1. Data Flow Diagram of BiNoC Direction Switching.

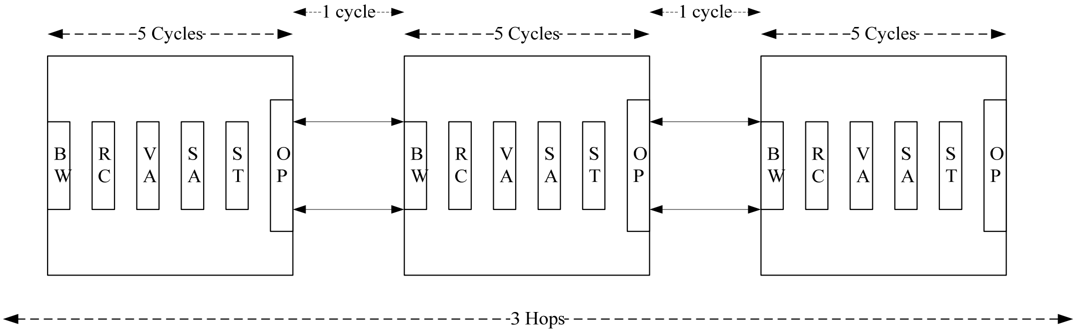


Fig. 3-2. Three-hops Pipelined Stages.

3.2 Anticipative Bidirectional Channel Control

To eliminate the time in waiting for direction switch, we will move the direction request mechanism forward to the routing stage, because the RC stage is the first stage that provides the output direction information of packets. As illustrated in Fig. 3-3, the original direction request based on the VA and SA allocations is successful because the channel authority is granted based on the buffer (VC) and link (SA). The anticipative direction switch can be used to request direction authority even when the VA/SA stage status is unknown. And the request sending is speculating that the VA/SA request will be granted and successful passed. This work will be mounted with GS packets only since the GS packets have more possibility to grant VA/SA both.

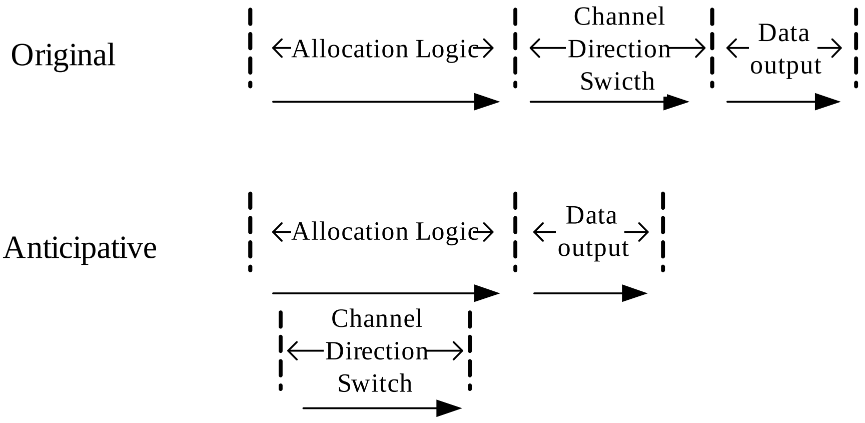


Fig. 3-3. Anticipative Bidirectional Channel Control.

With the statistic data table as listed in Fig. 3-4, we can observe that the most of output competition at every output port of a QoS-aware BiNoC router has only one GS packet involved in. Therefore, we decide to serve this anticipative mechanism for GS packets only.

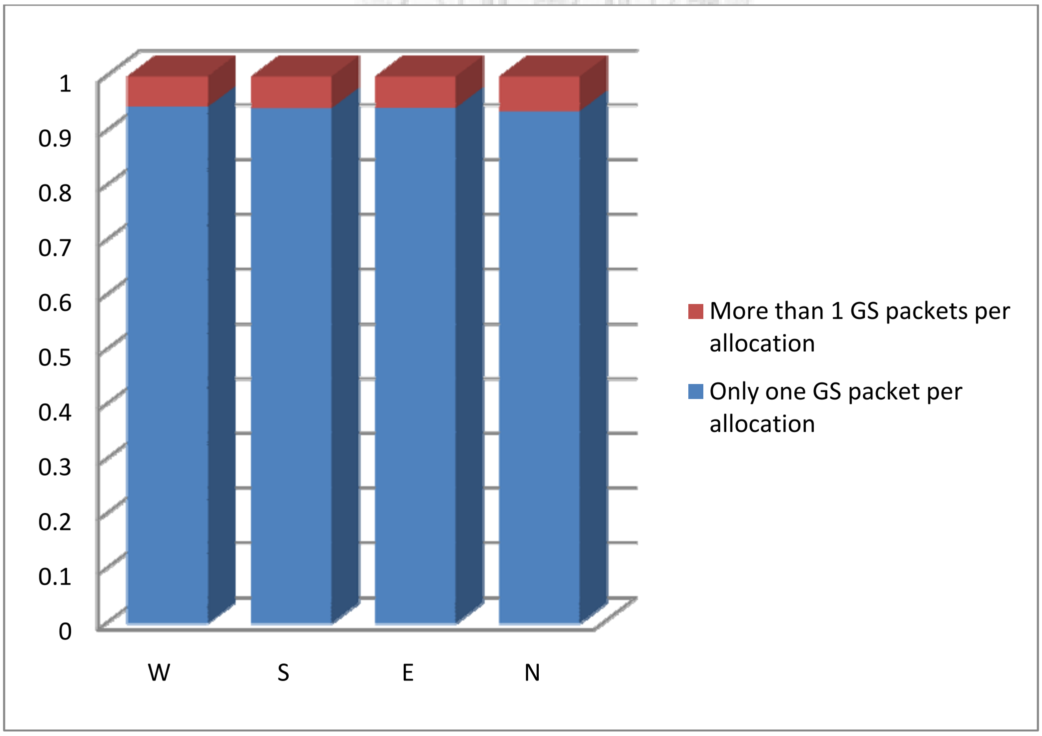


Fig. 3-4. GS Packet Allocation under Uniform Traffic, with GS percentage=0.3 and Injection Rate=0.32.

There is a similar example in real world: an ambulance carrying patient(s) usually has a higher priority in street. It can ignore traffic lights or vehicles in other direction. After all, the congestion still happen when there is a car or truck occupying the crossroad. The ambulance still has to wait until the road is free to go. If police officers are guarding on the crossroad which is far away from the ambulance to ensure that the direction and pass authority are guaranteed for the ambulance, the ambulance will get more opportunity to pass the intersection of road.

3.3 Packet Penetration

The penetration idea is similar to the penetration pressure in Chemistry Theory. The liquid from the higher pressure side of a semi-permeable membrane can pass through the membrane to another side. In the packet penetration design, we map the membrane into a router, particle availability into priority level, and the pressure into the request counts. The GS packets with a high priority can pass through the router when they do not excess the GS output requests. We can choose some specific packets routing according to the situation of the penetration mechanism. If the priority of a packet that can successfully pass a router is higher than the others, we can make the packet pass through the router when the data flow is not totally broken by the judge. There is an example as shown in Fig. 3-5. Fig. 3-5(a) shows a GS packet traveling the pipelined stages of two routers. The priority of this GS packet is high enough to compete against other BE packets; we call this phenomenon as oblivious winning competition. The penetration can reduce the pipelined stages and resources which the GS packet is occupying, in Fig. 3-5(a) the second router has to provide additional buffer space and virtual channel/switch allocation. But in Fig. 3-5(b), the second hop only provides a simple bypass and configuration to let the GS packet pass through the router. The penetration mechanism not only serves the GS packet with shorter latency and less ignorable allocation, but also makes resources not be occupied occasionally.

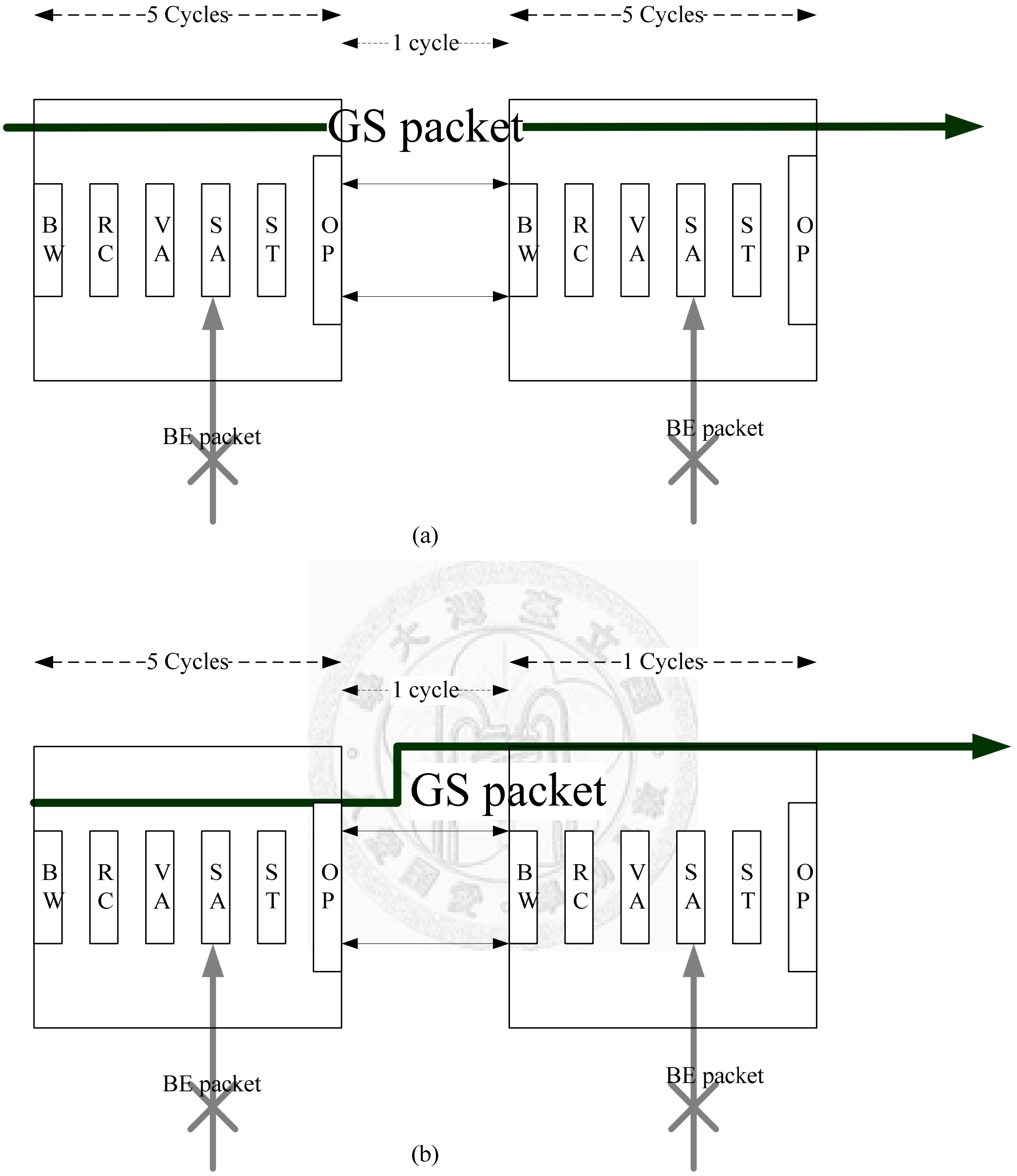


Fig. 3-5. Example of Packet Penetration.

There is a similar example in real world: the most of avenues in a big city usually are carriageway with more than 3 lanes in each direction. The lanes are separated into fast and slow lanes. The fast lanes are guaranteed higher speed and less intersection with other roads. The slow lanes are available to pass but almost cross with every road and have much more traffic lights than fast ones. We try to build a much fast transmission scheme than the original pipeline and the authority is prioritized instead of every available packet to use.

3.4 Pressure Balance

The balance will be required because the fast path build is set based on priority and the speculation that the path is not used by any other high priority packet. But in real world, there still is opportunity that an ambulance is meeting with another one on the road with only one lane. When a direction is monopolized by only one packet for a long time, it will make other packet stalled. Moreover, the other packets which need transmission quality will be stalled or delayed. When we sacrifice performance of some packets, we may break the original quality bottleneck for the specific packet. With sacrificing more and more packets, the improvement will reach the gap and become saturated. That is why we have to set a balance threshold to keep the trade-off not too over and to maintain the overall performance of network.

3.5 The Proposed Router

We propose an Anticipative QoS controlled BiNoC router architecture to improve the performance of GS packets. The main idea is reducing the transmission latency and inefficient resource occupation. Better bandwidth utilization can be obtained from the bi-direction channel switch and virtual channel authority prioritized. Moreover, the balance control between the improvement and scarification of packets is established by a simple calculation scheme.

CHAPTER 4

ROUTER IMPLEMENTATION

Based on the BiNoC architecture introduced in Chapter 2 and combing the idea discussed in last chapter, we adopt the bi-direction channel transmission scheme with QoS support and present a practical router architecture to be implemented with virtual channel in a pipeline fashion. Fig. 4-1 shows the Anticipative QoS controlled BiNoC router architecture.

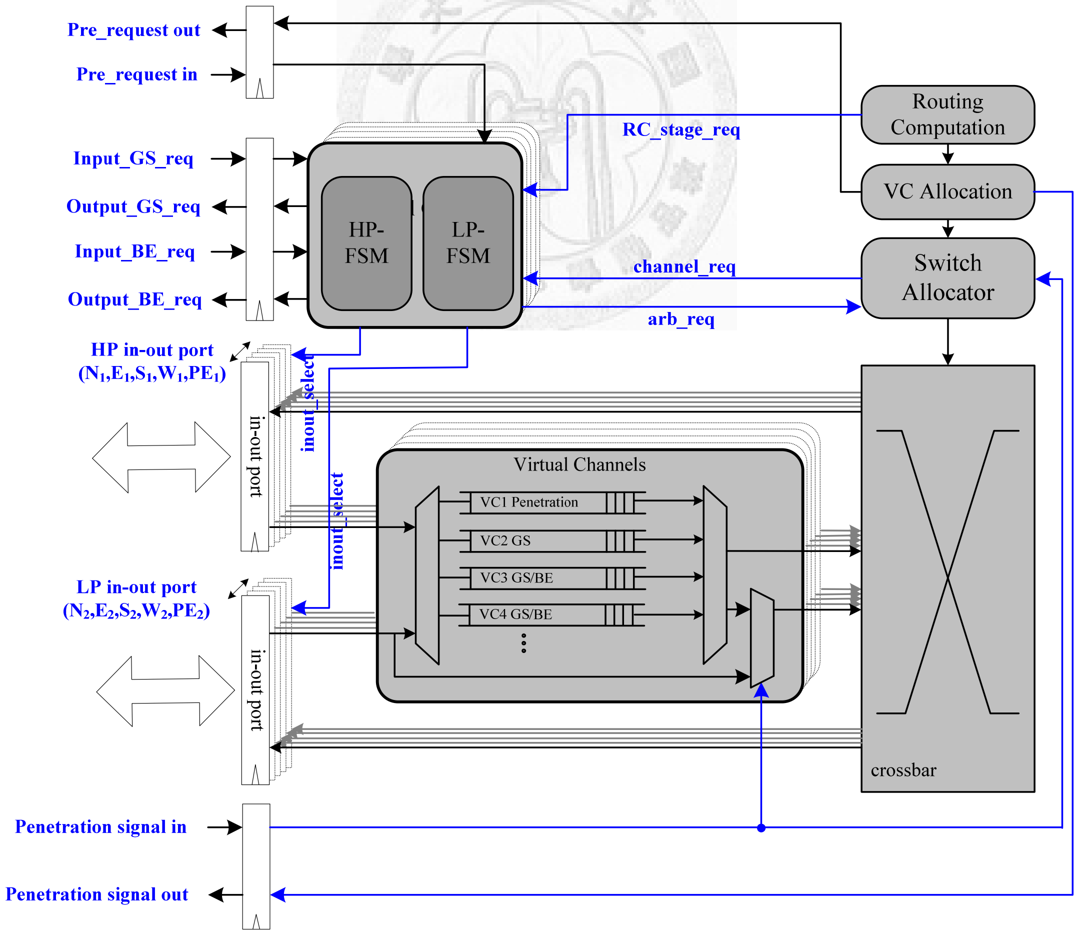
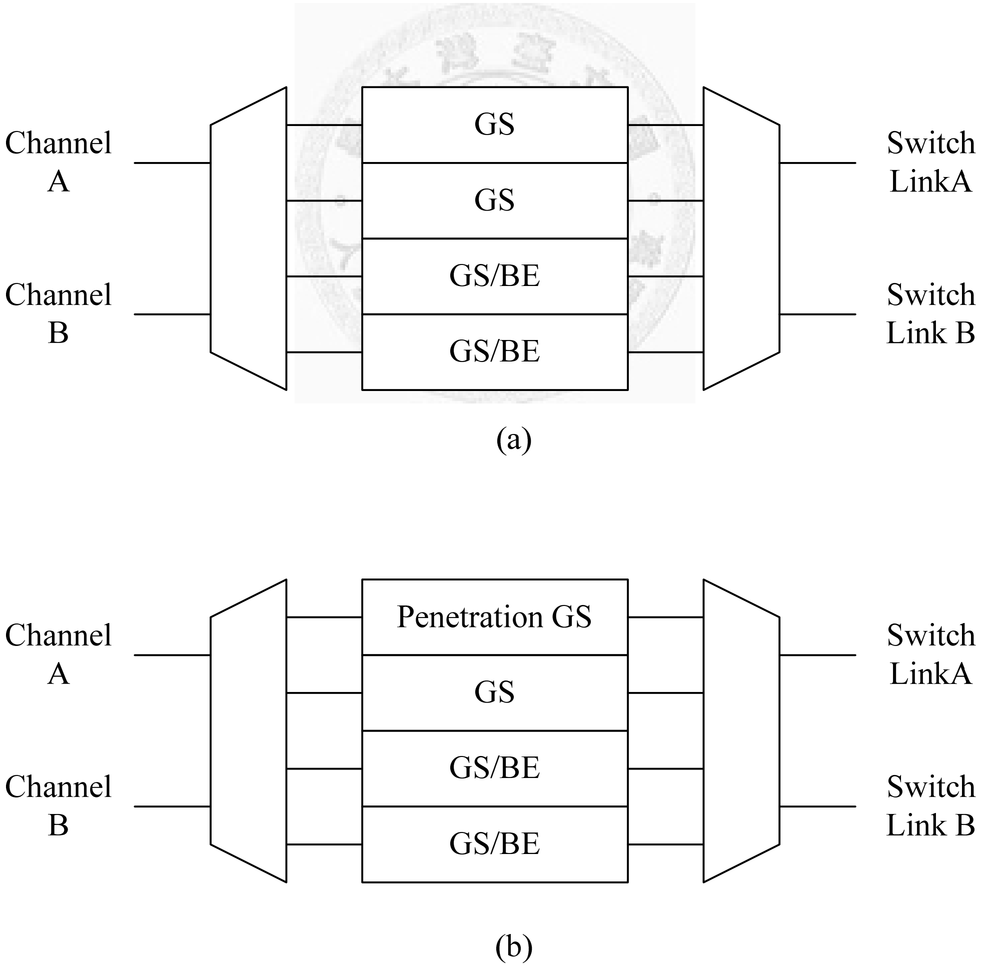


Fig. 4-1. Anticipative QoS Controlled BiNoC Router Architecture.

4.1 Basic Router Design

The basic router architecture as shown in Fig. 4-1 is simple and inherits all the characteristics of the QoS-aware BiNoC. The virtual channel architecture is developed from the prioritized virtual channel, but the priority level is different from the original one. As Fig. 4-2(a) shows, the original virtual channel priority is divided into GS and GS/BE levels. In our design as shown in Fig. 4-2(b), we separate the four virtual channels into one Penetration GS channel, one GS channel, and two GS/BE channels. The additional bypass function will be mentioned in Section 4.3. The Penetration GS channel receives the packets sending from the router, which is at a distance of two hops away instead of the neighbor routers.



(a)

(b)

Fig. 4-2. Virtual Channel Architecture Comparison.

4.2 Anticipative Bidirectional Channel Control

In Chapter 3, we mentioned about the previous request direction with speculation. In Fig. 4-1, we can see that the request from the Routing Computation stage to the Channel Controller. This implies that the request will be directly sent to the channel controller when the routing computation detects that the packet has a high priority on getting granted in other stages. However, we may not grant all packets' request directions that they are heading for. In the pure pre-request experiment, discovering every request of direction by a packet at the routing stage causes worse performance, because most packets may change direction successfully but fail in switch allocation or virtual channel allocation. So we need to limit the number of packets which are available to use speculative switch.

The anticipative bidirectional channel control is suitable only for GS packets. As Fig. 4-3 shows, the previous request at the RC stage brings effective time reduction. However, not all of packets can get authority at the VA and SA stages. Without passing the VA and SA stages at T=2 and T=3, the direction changes for nothing. GS packets provide higher priority to guarantee more possibility to grant the authority at both VA/SA stages.

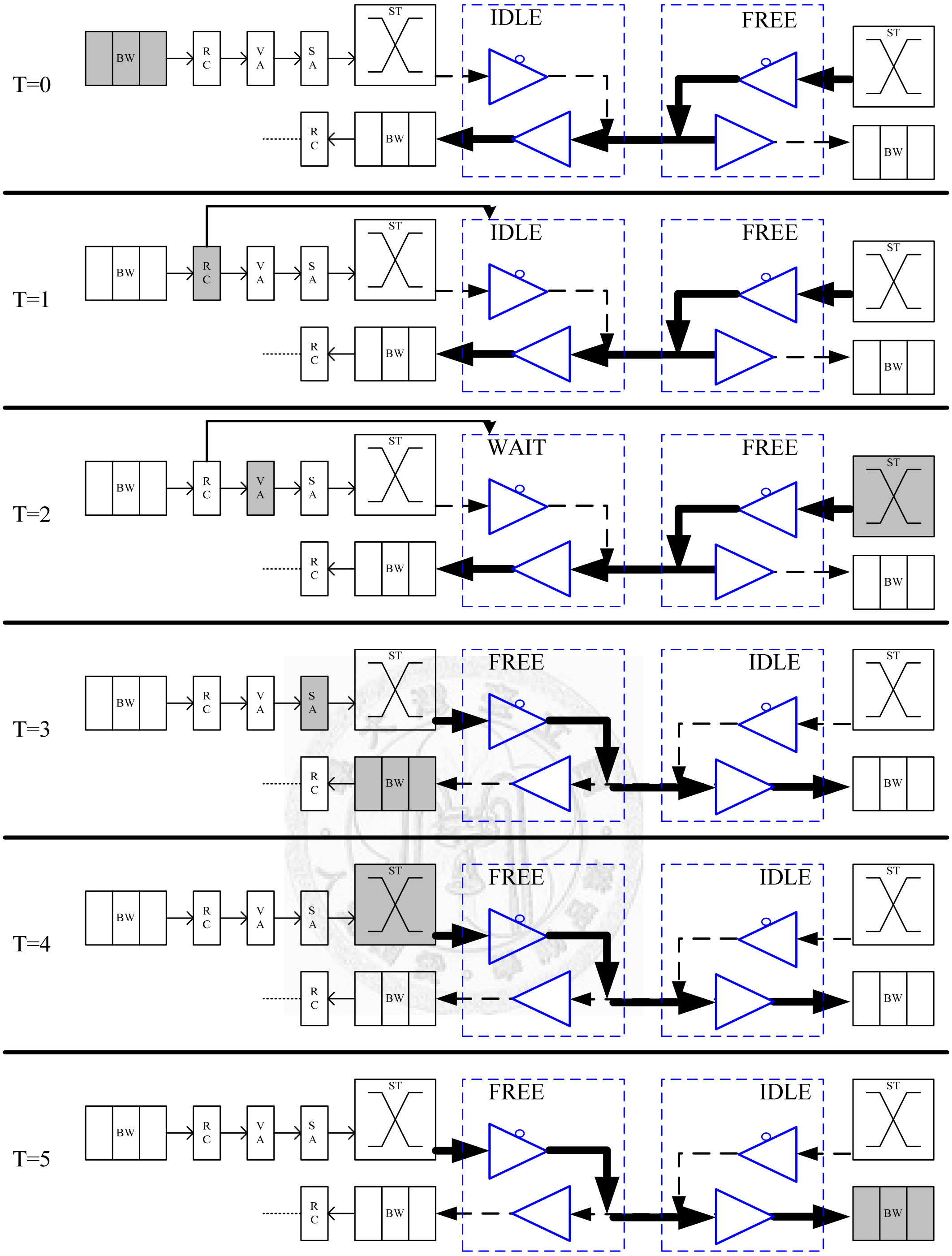


Fig. 4-3. Data Flow Diagram of Anticipative Bidirectional Channel Control.

4.3 Penetration Ability

As introduced in Chapter 3, we want to make GS packets able to bypass the router pipelines conditionally. Fig. 4-4 shows the condition whether a packet can bypass the router or not as follows. A GS packet in Router A is bypassing Router B and then transmitted into Router C. The packet first gets penetration availability information from the routing computation at the RC stage of Router A. And it requests the destination virtual channel buffer of Router C at the VA stage of Router A. Finally after switch allocation, it can transmit the output data from Router A with an additional penetration signal. This signal can stall the original SA process to the same direction of Router B and set the crossbar to a specified output direction. The packet can directly traverse the ST stage without additional allocation. After departing Router B, the packet arrives Router C and will follow the same steps of normal router pipelining to complete its routing.

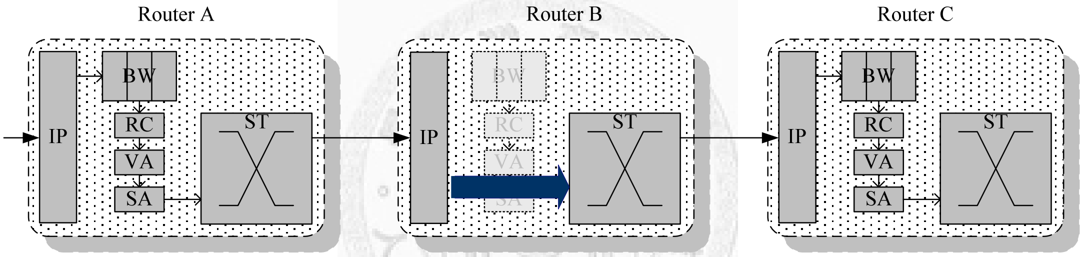


Fig. 4-4. Penetration Ability.

To further introduce the sequence of penetration mechanism, we propose a flow diagram in Fig. 4-5. If there is an available info detected in RC stage, packet having penetration conditions will compete for the destination virtual channel at VA, and the others will do normal virtual channel allocation. After having the penetration virtual channel granted, the packet will have a higher priority for competition at the Switch Allocation stage. The packet which actually has an authority to penetrate the router will raise the signal to stall the router at next hop.

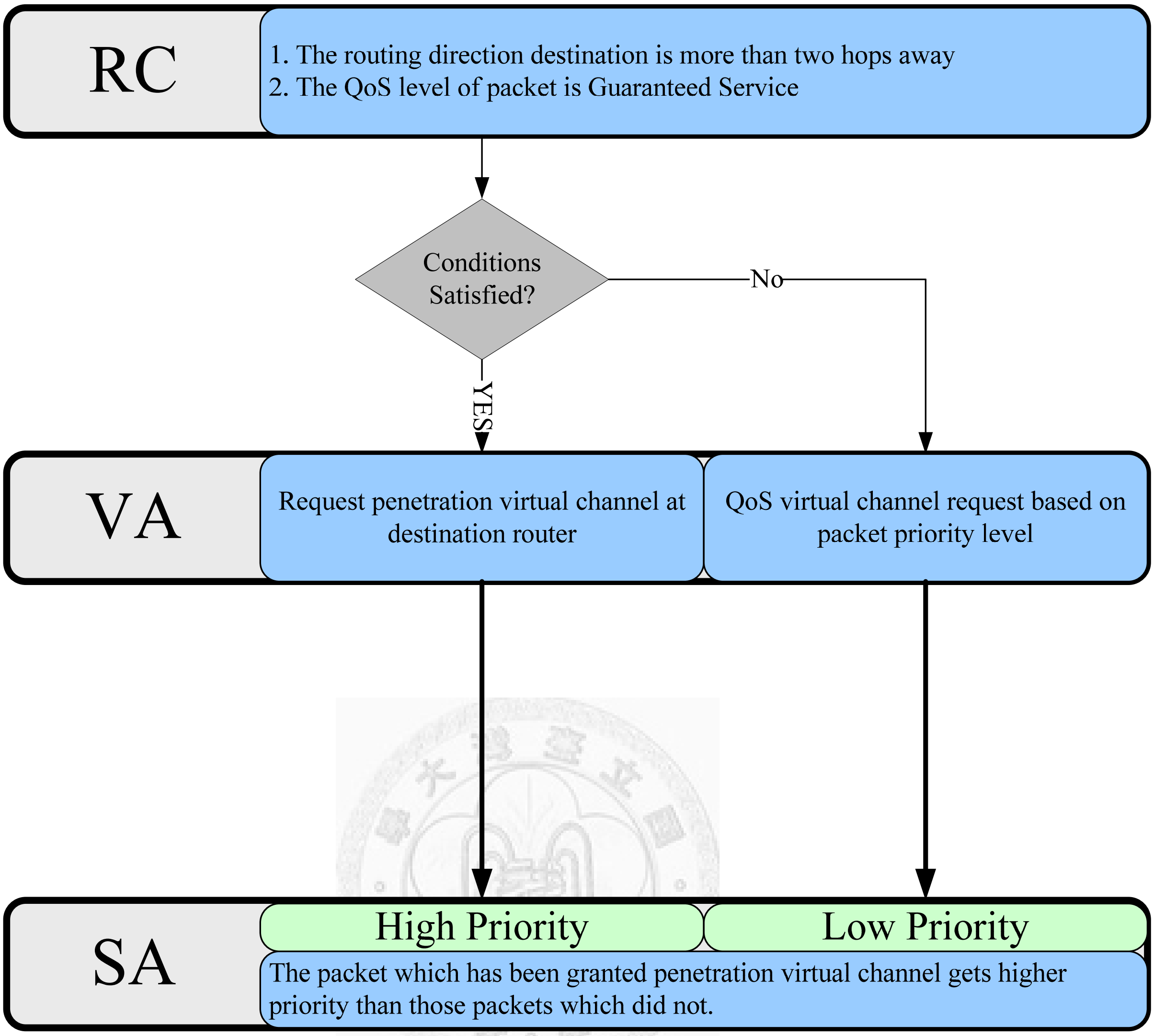

RC
1. The routing direction destination is more than two hops away
2. The QoS level of packet is Guaranteed Service
Conditions Satisfied?
No
YES
VA
Request penetration virtual channel at destination router
QoS virtual channel request based on packet priority level
SA
High Priority
Low Priority
The packet which has been granted penetration virtual channel gets higher priority than those packets which did not.

Fig. 4-5. Generation of Penetration Packet.

4.4 Penetration Balance

The penetration behavior not only reduces the latency of a specific packet but also breaks the other GS packets which are transmitting. To improve the case, we propose a balance mechanism that uses a flow control to control the penetration process by the specified direction in the output channel that was already requested by the excessive GS packets. The original flow control for penetration buffer is illustrated in Fig. 4-6(a). Since the distance between Routers A and C is too far, the buffer-full control signal will propagate from Router C through Router B to Router A. The balance mechanism is simple to implement as shown in Fig. 4-6(b), we block the flow control signal propagation from Router B to Router A, when the GS packet count from Router B to the direction is higher than the threshold value.

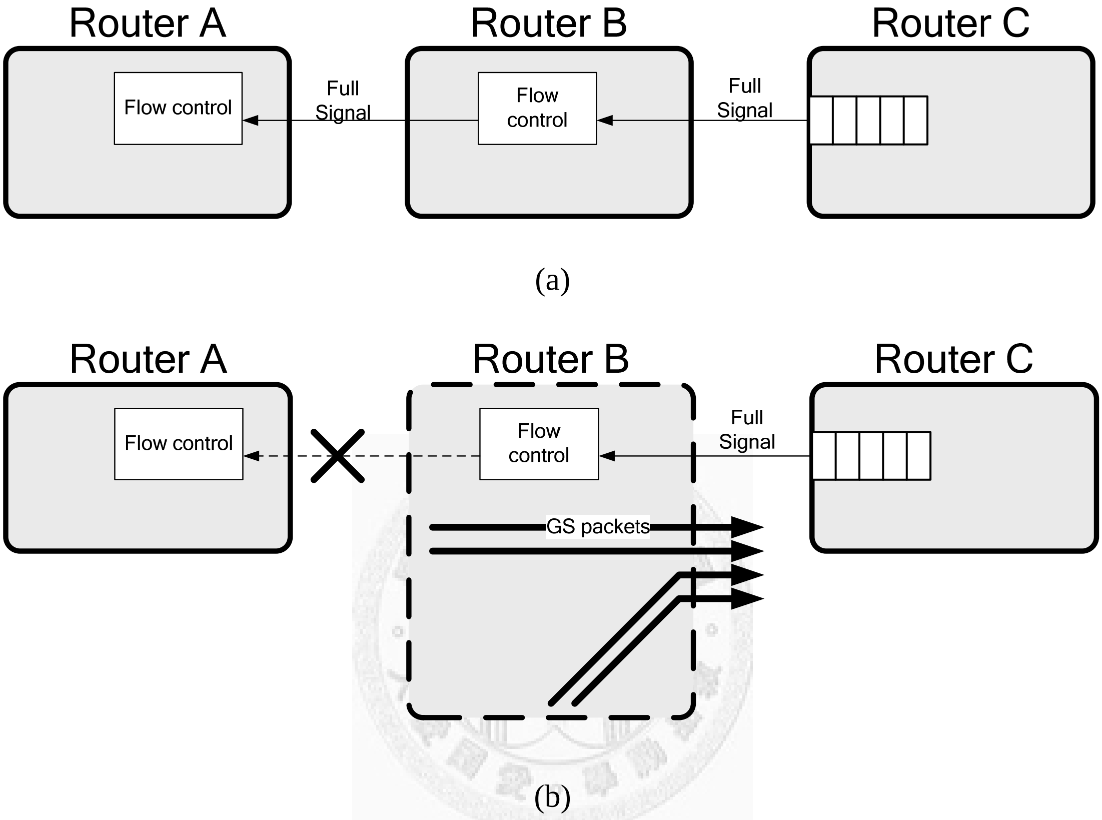


Fig. 4-6. Penetration-Buffer Flow Control.

4.5 Proposed Anticipative QoS Controlled BiNoC

To integrate the above two specific abilities with BiNoC, we propose an improved QoS-aware BiNoC with anticipative bidirectional channel control and penetration ability. The channel direction control becomes much complex because the output direction of the bypassed router has to be set to the designed direction. Fig. 4-7 illustrates the active behavior of penetration in BiNoC. At T=2, when the virtual channel allocation completely allocates the packet with penetration to a virtual channel, the direction request of next hop from Router B will be sent. At T=4, the channel direction between Routers B and C is from B to C. At T=5, the packet will bypass the router pipeline and directly use the crossbar with configuration to traverse through Router B.

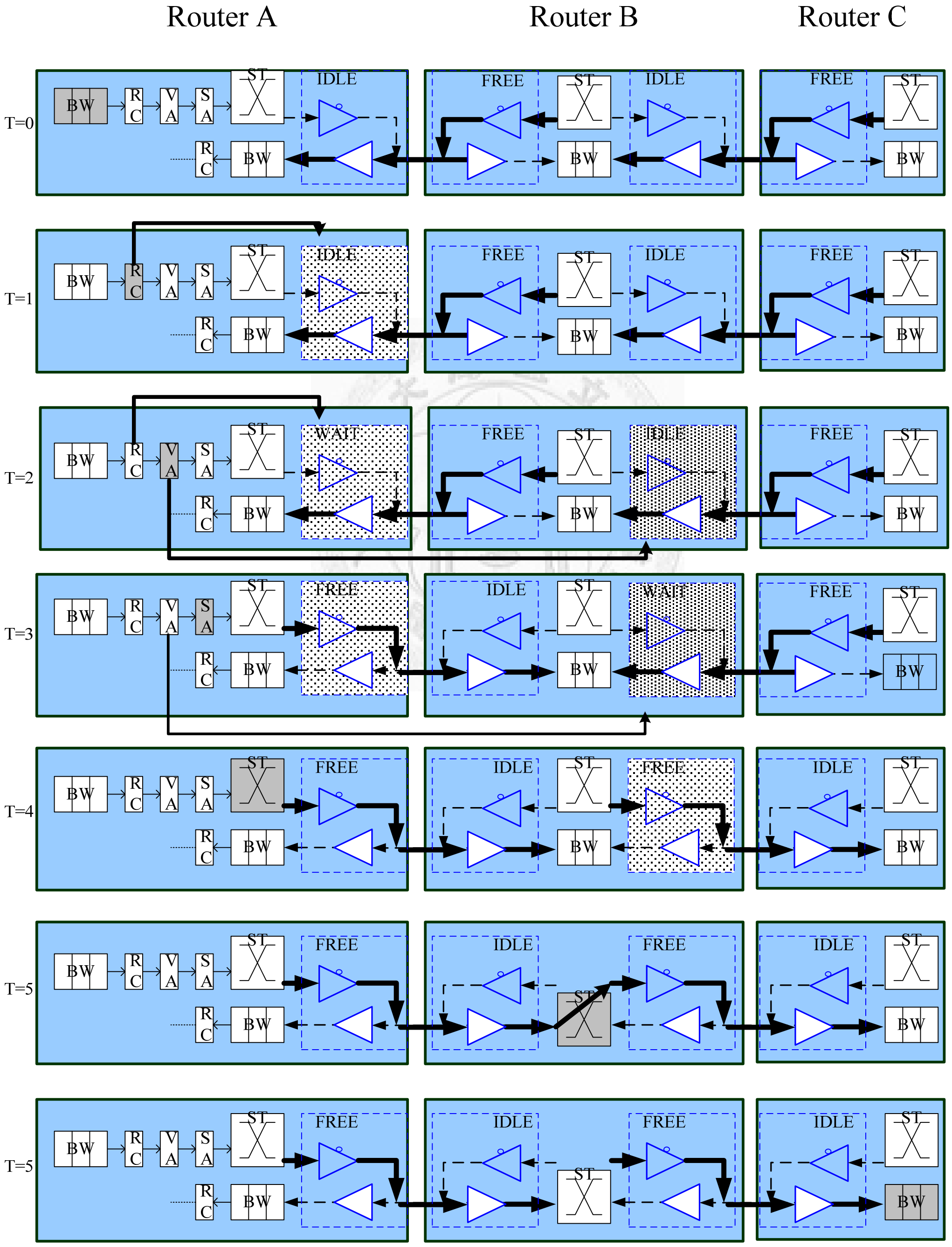


Fig. 4-7. Behavior of the Proposed Anticipative QoS Controlled BiNoC.

CHAPTER 5

EXPERIMENT RESULT AND DISSCUSION

In order to make a performance comparison and hardware cost estimation between the QoS aware BiNoC and our proposed Anticipative QoS Controlled BiNoC architectures. A cycle-accurate simulation environment was implemented with different router architecture designs in HDL. Every design is comprised of multiple functional blocks including routing, channel control, arbitration, and switch fabric. The traffic patterns in our simulation environment are synthetic traffics [23] with varying QoS utilization from 0.1 to 0.5. The environment simulates wormhole switching with a packet separated into head, body, tail flits. The routing algorithm used is XY routing which routes packets X direction first and then Y direction. And the arbitration for solving packets conflicts with the same priority level is round-robin principle.

5.1 Performance Evaluation on Virtual Channel Routers

The characteristics of Typical-NoC, BiNoC, Pre-request BiNoC (Pre-Req BiNoC) and Anticipative QoS BiNoC (AQ-BiNoC) used in our experiments are listed in Table 5.1. To make fair comparison, all the four routers use 4VC BUF32, it means that there are total 4 parallel 8-flit-deep-buffers in one input unit with a total of 32 flit buffers. The Pre-Req BiNoC is implemented as the BiNoC with channel pre-request functionality.

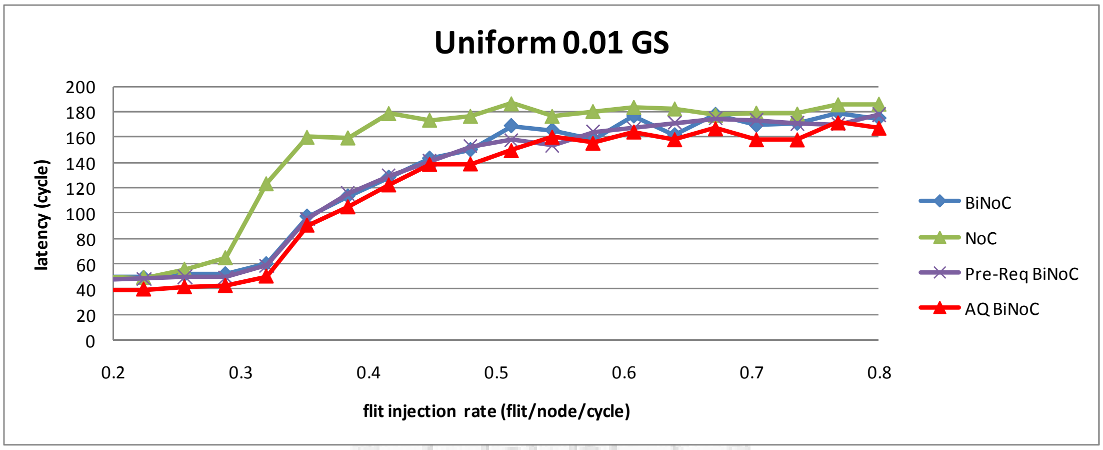
Table 5-1. NoC Architectures used in our Experiments.

Architecture	Typical-NoC	BiNoC	Pre-request BiNoC (Pre-Req BiNoC)	AQ-BiNoC (proposed architecture)
Total # VC	4	4	4	4
Each VC Size	8	8	8	8
Total Channels	5-in 5-out	10-inout	10-inout	10-inout
Channels / Dir.	1-in 1-out	2-inout	2-inout	2-inout
Each Buf. Size	32 flits	32 flits	32 flits	32 flits
Total Buf. Size	160 flits	160 flits	160 flits	160 flits
Crossbar	5 X 5	10 X 10	10 X 10	10 X 10

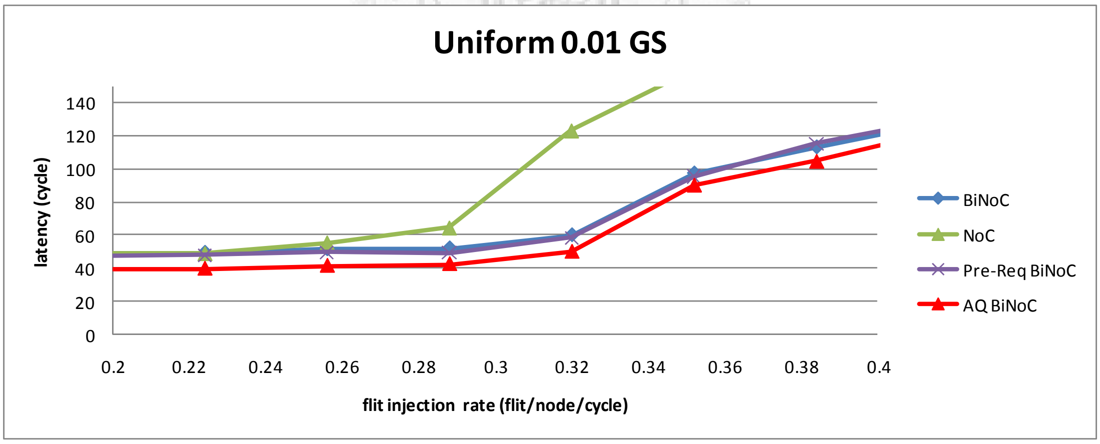
5.2 Synthetic Traffic Analysis

In synthetic traffic analysis, the physical layer of our simulation environment comprises of 8×8 nodes connected as a mesh array. Each packet has a constant length of 16 flits. The tests performed sent packets at varying injection rates and varying GS packets percentages under different traffics for 25000 cycles. Four types of synthetic traffic patterns were run: uniform, regional, transpose, and HotSpot. In uniform traffic, a node will be the destination of every other nodes with equal probability based on the injection rate. In regional, 90% of the packets are sent to the destination at distance less than 3 hops, and most transmission occur among two neighbor routers. In transpose traffic, a node at coordinate (*i*,*j*) will sent a packet to the destination which coordinate is (*j*,*i*). In HotSpot traffic, most of packets are send to the same destination at once. The traffic is similar with real-case traffic because in SoC most IP have communication with the main CPU. The GS percentage defines the GS packets count versus total packet count, this value not only represents the GS packets utilization but also shows that the architectures can get fit in different QoS requirements.

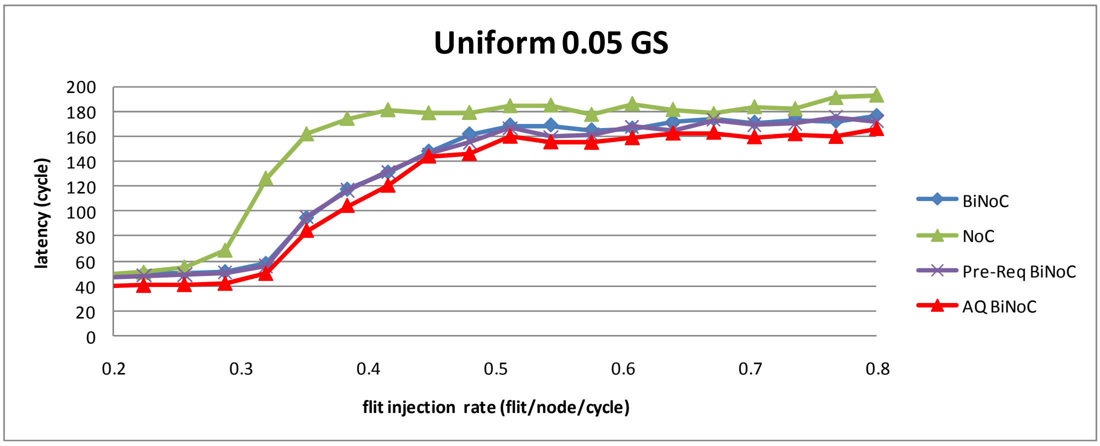
We analyze the simulation results obtained from low GS percentage first under the uniform traffic. Figs. 5-1(a) and (c) represent the GS packet latency with GS percentages of 0.01 and 0.05 respectively. Figs. 5-1(b) and (d) show the respective focused-parts of Figs. 5-1(a) and (c). We can discover that the latency of AQ-BiNoC is lower than BiNoC, and Pre-Req BiNoC has also latency improvement compared with the traditional BiNoC.



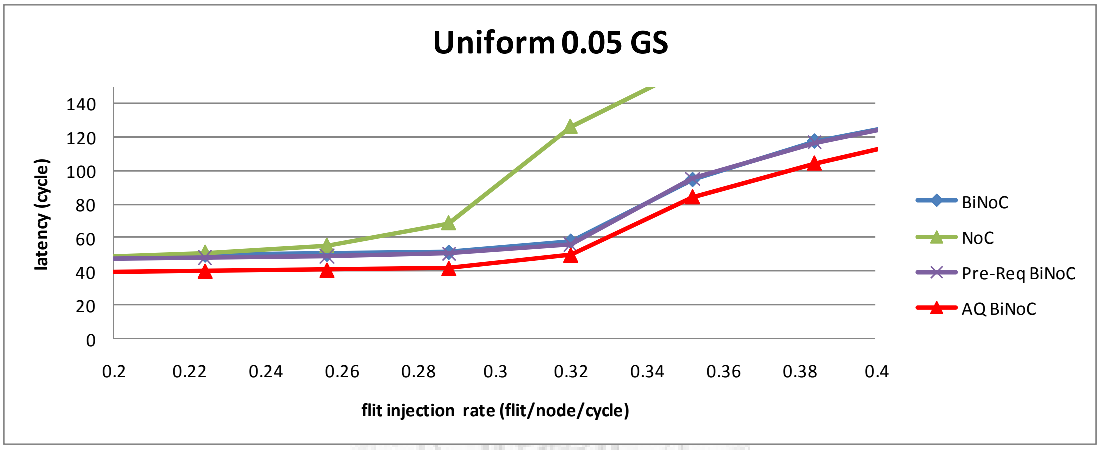
(a)



(b)



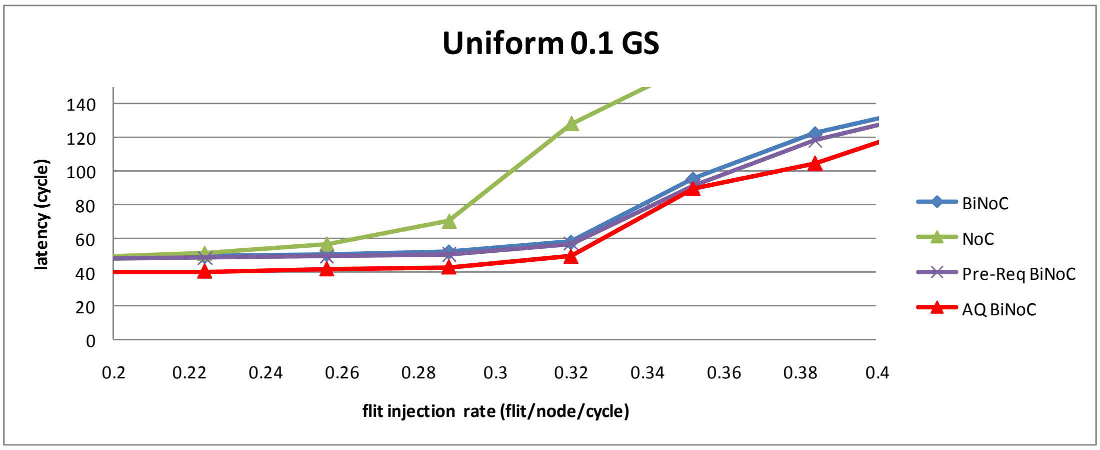
(c)

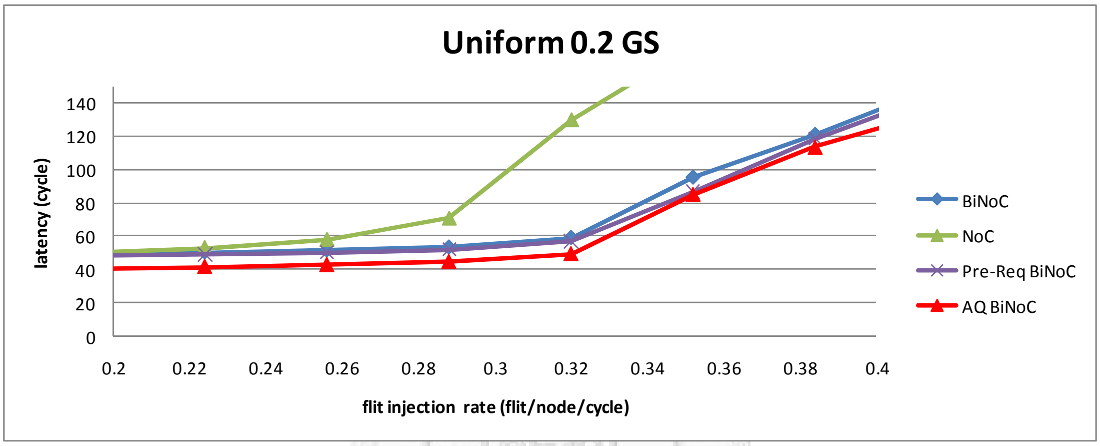


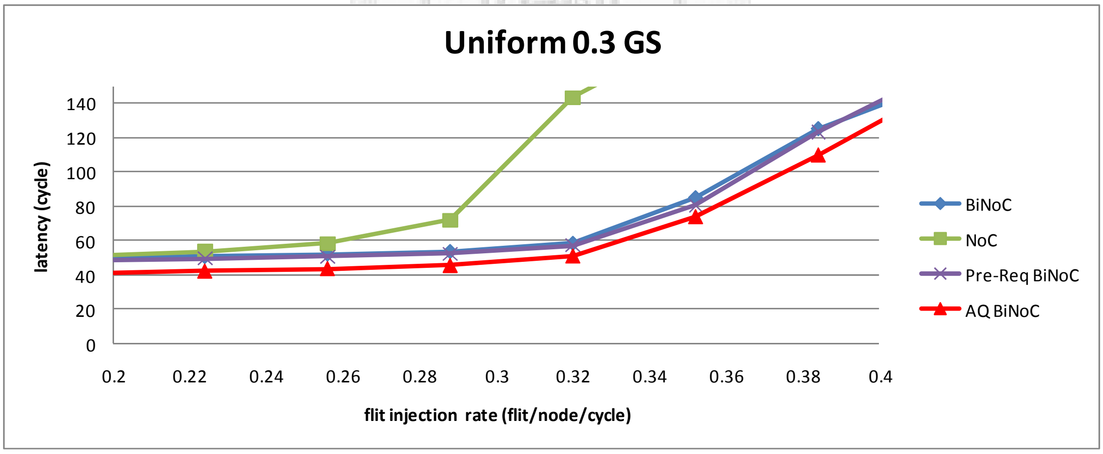
(d)

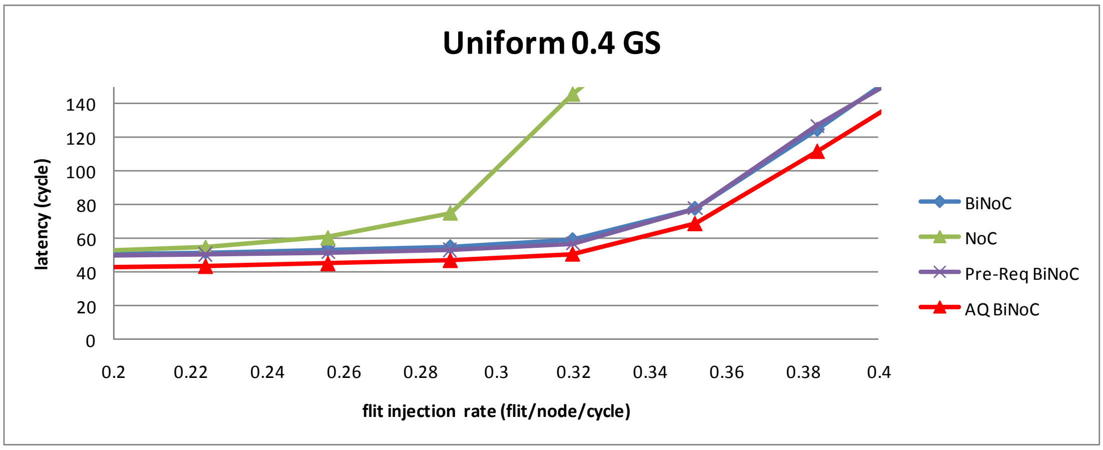
Fig. 5-1. GS Packet Latency at Low GS Percentage under Uniform Traffic.

Figure 5-2 shows the latency of uniform traffic under higher GS percentage from 0.1 to 0.5. We only focus on the latencies from zero-load to saturation which is double of the zero-load latency. The latency of AQ-BiNoC is 15% less than BiNoC even in GS with a percentage of 50%. The simple Pre-Req BiNoC still has similar latency curve comparing with BiNoC. The minor improvement comes from the inter-router arbitration mechanism.


Uniform 0.1 GS
latency (cycle)
140
120
100
80
60
40
20
0
0.2
0.22
0.24
0.26
0.28
0.3
0.32
0.34
0.36
0.38
0.4
flit injection rate (flit/node/cycle)
BiNoC
NoC
Pre-Req BiNoC
AQ BiNoC


Uniform 0.2 GS
latency (cycle)
140
120
100
80
60
40
20
0
0.2
0.22
0.24
0.26
0.28
0.3
0.32
0.34
0.36
0.38
0.4
flit injection rate (flit/node/cycle)
BiNoC
NoC
Pre-Req BiNoC
AQ BiNoC


Uniform 0.3 GS
latency (cycle)
140
120
100
80
60
40
20
0
0.2
0.22
0.24
0.26
0.28
0.3
0.32
0.34
0.36
0.38
0.4
flit injection rate (flit/node/cycle)
BiNoC
NoC
Pre-Req BiNoC
AQ BiNoC


Uniform 0.4 GS
latency (cycle)
140
120
100
80
60
40
20
0
0.2
0.22
0.24
0.26
0.28
0.3
0.32
0.34
0.36
0.38
0.4
flit injection rate (flit/node/cycle)
BiNoC
NoC
Pre-Req BiNoC
AQ BiNoC

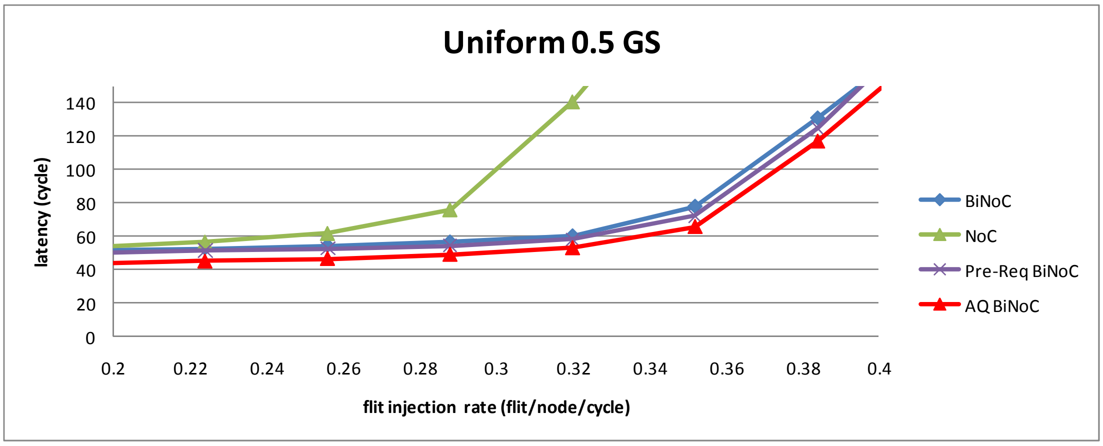
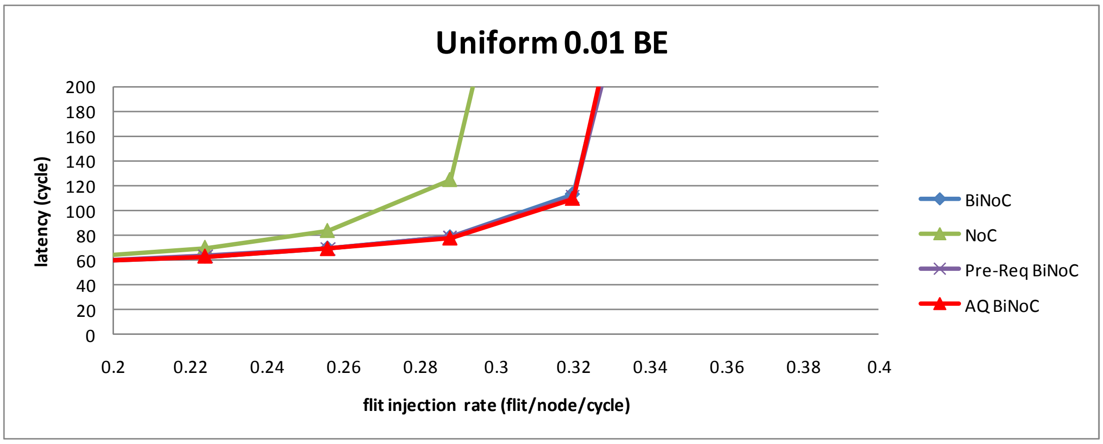
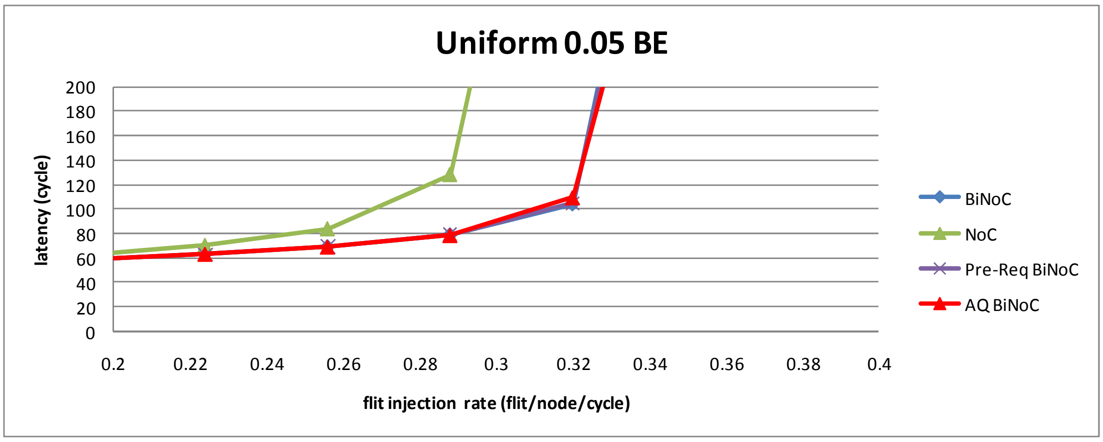
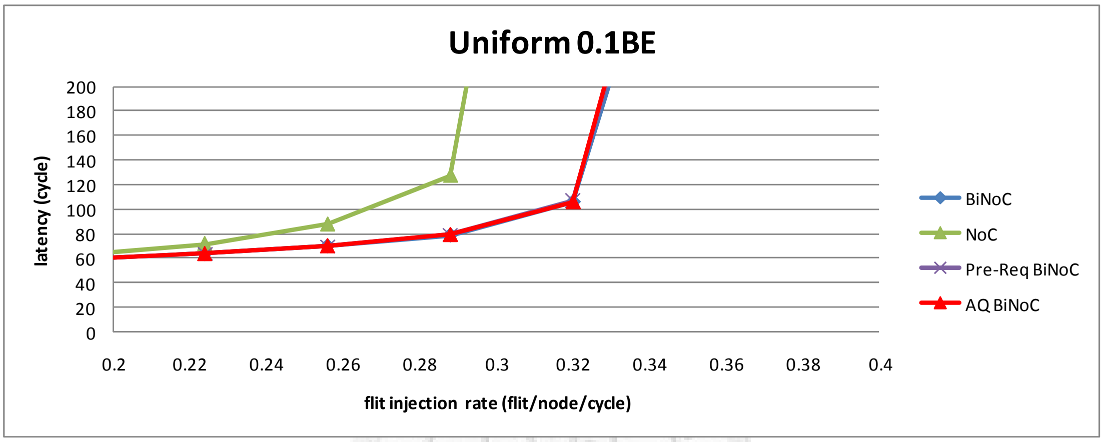


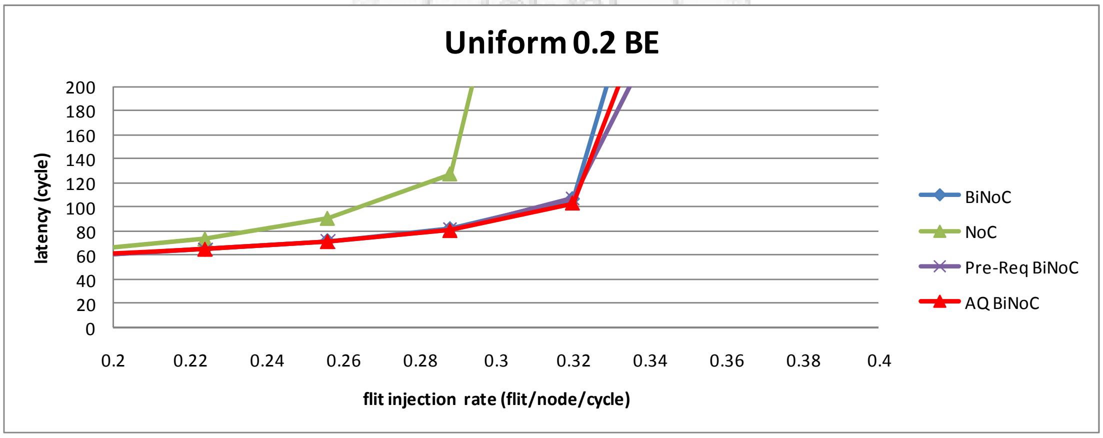
Fig. 5-2. GS Packet Latency at High GS Percentage under Uniform Traffic.

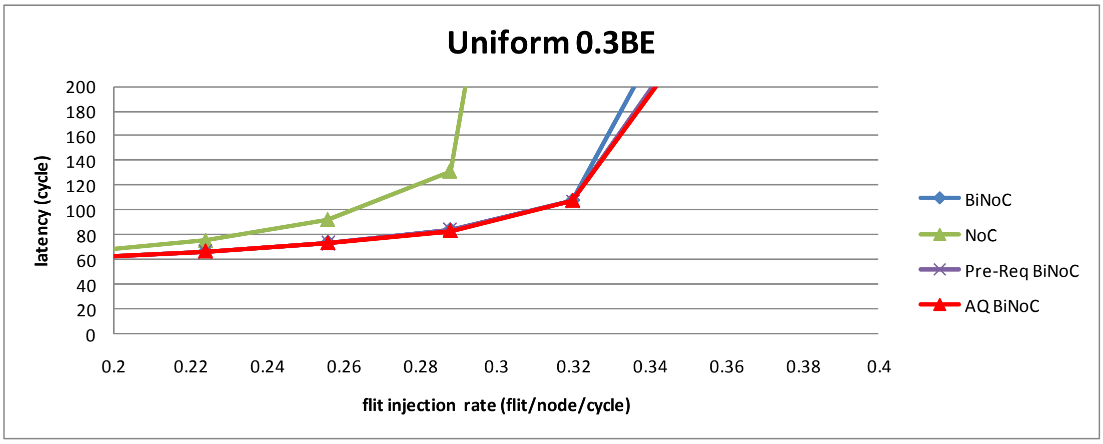
The following graphs show the BE packet latencies in different QoS percentages over different injection rates. We can notice that the latencies of BE packet in BiNoC, Pre-Req BiNoC and AQ-BiNoC almost have no difference. The reason is that most functions implemented are used to serve GS packets only. When the GS packets travel much fast in the network, the resources occupied by GS packets could also be released as soon as possible. That is why in some cases the BE latency in AQ-BiNoC is better than the one in BiNoC.




Uniform 0.05 BE
latency (cycle)
200
180
160
140
120
100
80
60
40
20
0
0.2
0.22
0.24
0.26
0.28
0.3
0.32
0.34
0.36
0.38
0.4
flit injection rate (flit/node/cycle)
BiNoC
NoC
Pre-Req BiNoC
AQ BiNoC


Uniform 0.1BE
latency (cycle)
200
180
160
140
120
100
80
60
40
20
0
0.2
0.22
0.24
0.26
0.28
0.3
0.32
0.34
0.36
0.38
0.4
flit injection rate (flit/node/cycle)
BiNoC
NoC
Pre-Req BiNoC
AQ BiNoC


Uniform 0.2 BE
latency (cycle)
200
180
160
140
120
100
80
60
40
20
0
0.2
0.22
0.24
0.26
0.28
0.3
0.32
0.34
0.36
0.38
0.4
flit injection rate (flit/node/cycle)
BiNoC
NoC
Pre-Req BiNoC
AQ BiNoC


Uniform 0.3BE
latency (cycle)
200
180
160
140
120
100
80
60
40
20
0
0.2
0.22
0.24
0.26
0.28
0.3
0.32
0.34
0.36
0.38
0.4
flit injection rate (flit/node/cycle)
BiNoC
NoC
Pre-Req BiNoC
AQ BiNoC

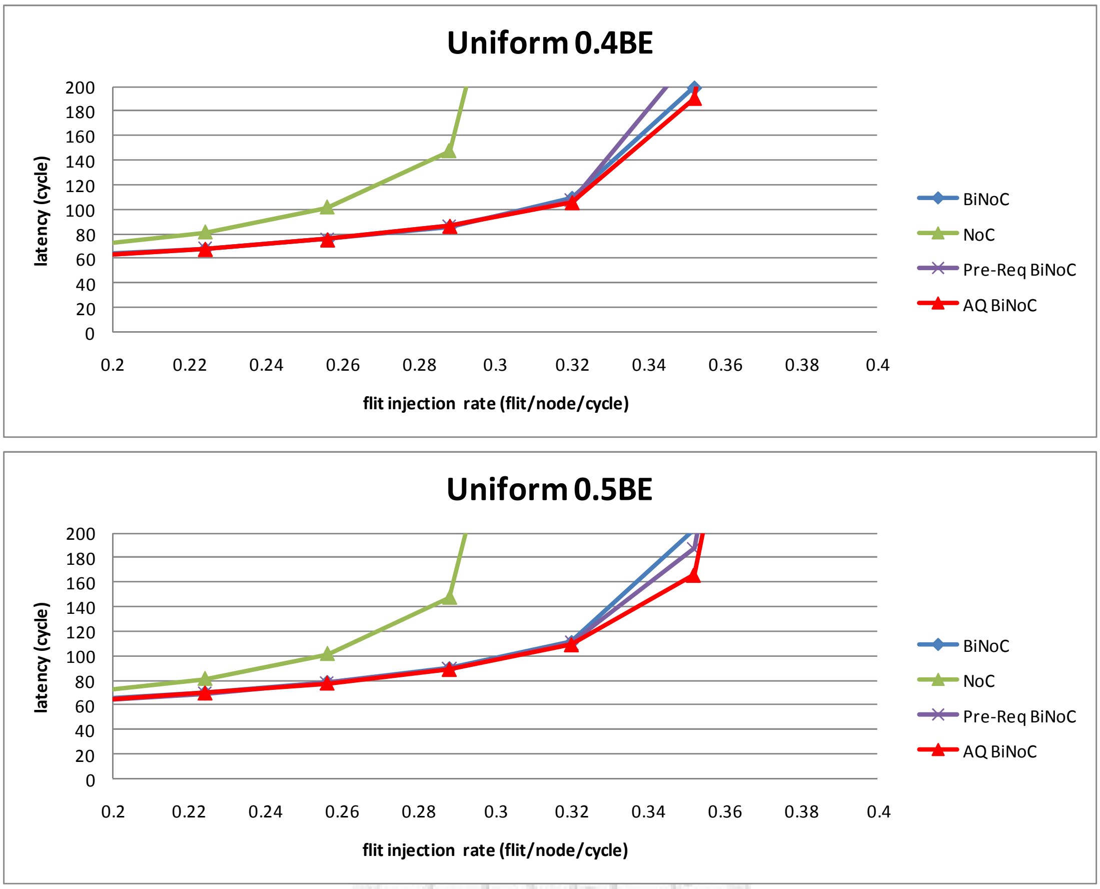
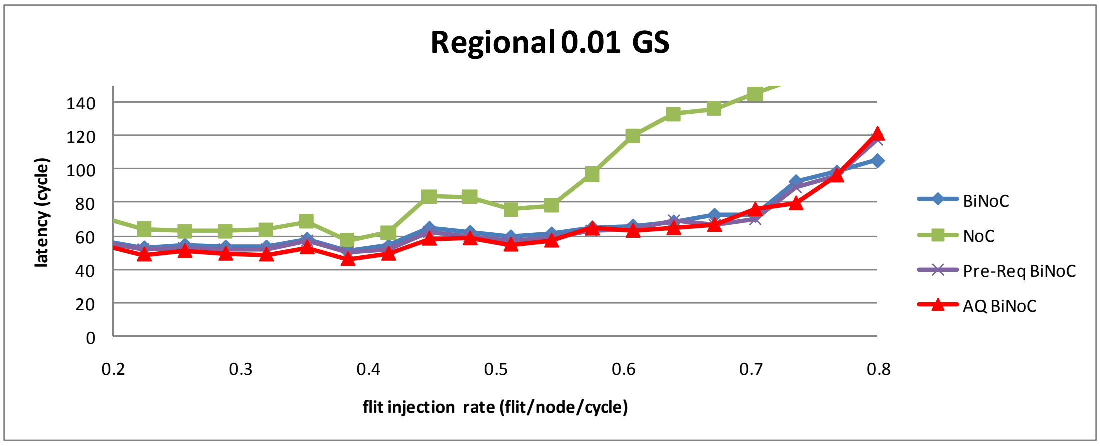
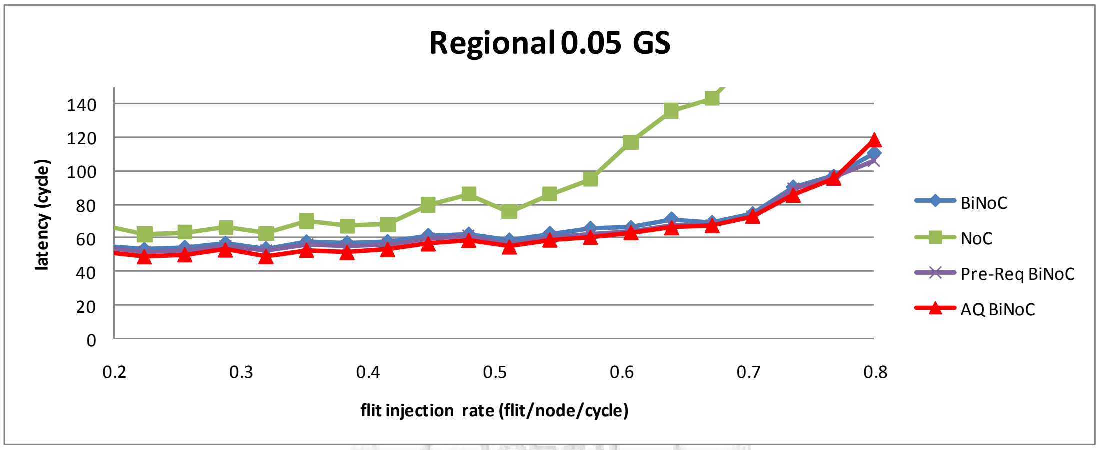
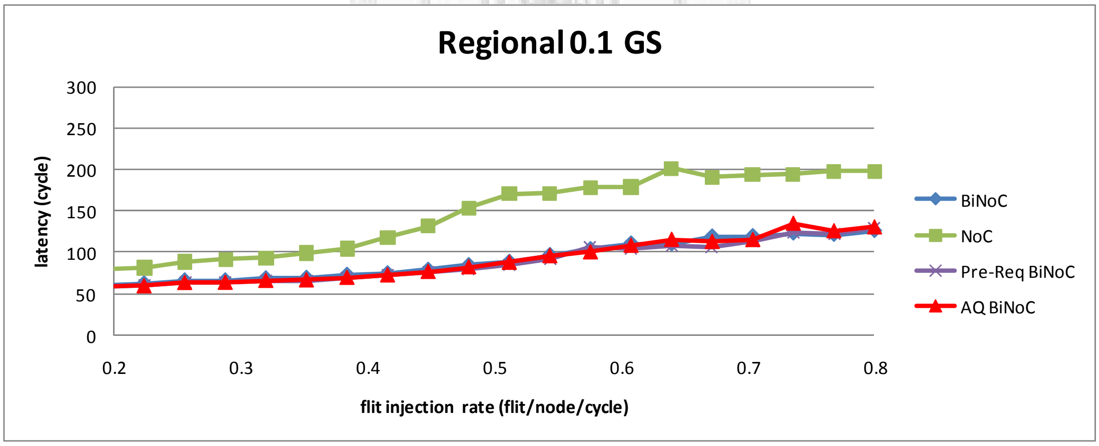


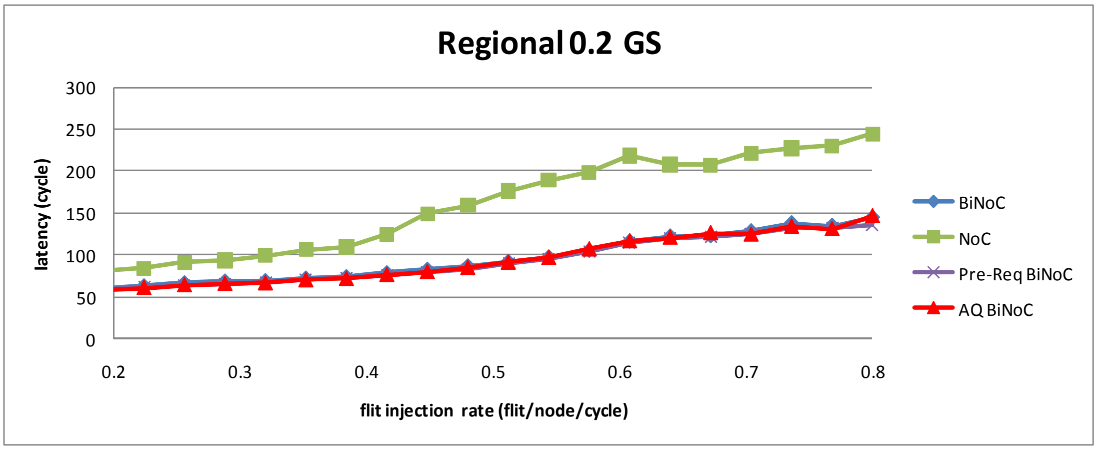
Fig. 5-3. BE Latency vs. Injection Rate in GS Percentages from 0.01 to 0.5.

From Figs. 5-4 and 5-5, we can observe that the improvement of packet is not oblivious under regional traffic, the most important reason for the latency of each architecture being such close is that the regional traffic is transmitting data to close destination mostly. The latency decrease is from the anticipative bi-direction channel control only in case where there is no long enough distance of packet transmitting. It causes the Pre-Req BiNoC and AQ-BiNoC have the same performance curves. And they both improve the neighborhood transmission by the inter-router channel control.


Regional 0.01 GS
latency (cycle)
0
20
40
60
80
100
120
140
0.2
0.3
0.4
0.5
0.6
0.7
0.8
flit injection rate (flit/node/cycle)
BiNoC
NoC
Pre-Req BiNoC
AQ BiNoC


Regional 0.05 GS
latency (cycle)
0
20
40
60
80
100
120
140
0.2
0.3
0.4
0.5
0.6
0.7
0.8
flit injection rate (flit/node/cycle)
BiNoC
NoC
Pre-Req BiNoC
AQ BiNoC


Regional 0.1 GS
latency (cycle)
0
50
100
150
200
250
300
0.2
0.3
0.4
0.5
0.6
0.7
0.8
flit injection rate (flit/node/cycle)
BiNoC
NoC
Pre-Req BiNoC
AQ BiNoC


Regional 0.2 GS
latency (cycle)
0
50
100
150
200
250
300
0.2
0.3
0.4
0.5
0.6
0.7
0.8
flit injection rate (flit/node/cycle)
BiNoC
NoC
Pre-Req BiNoC
AQ BiNoC

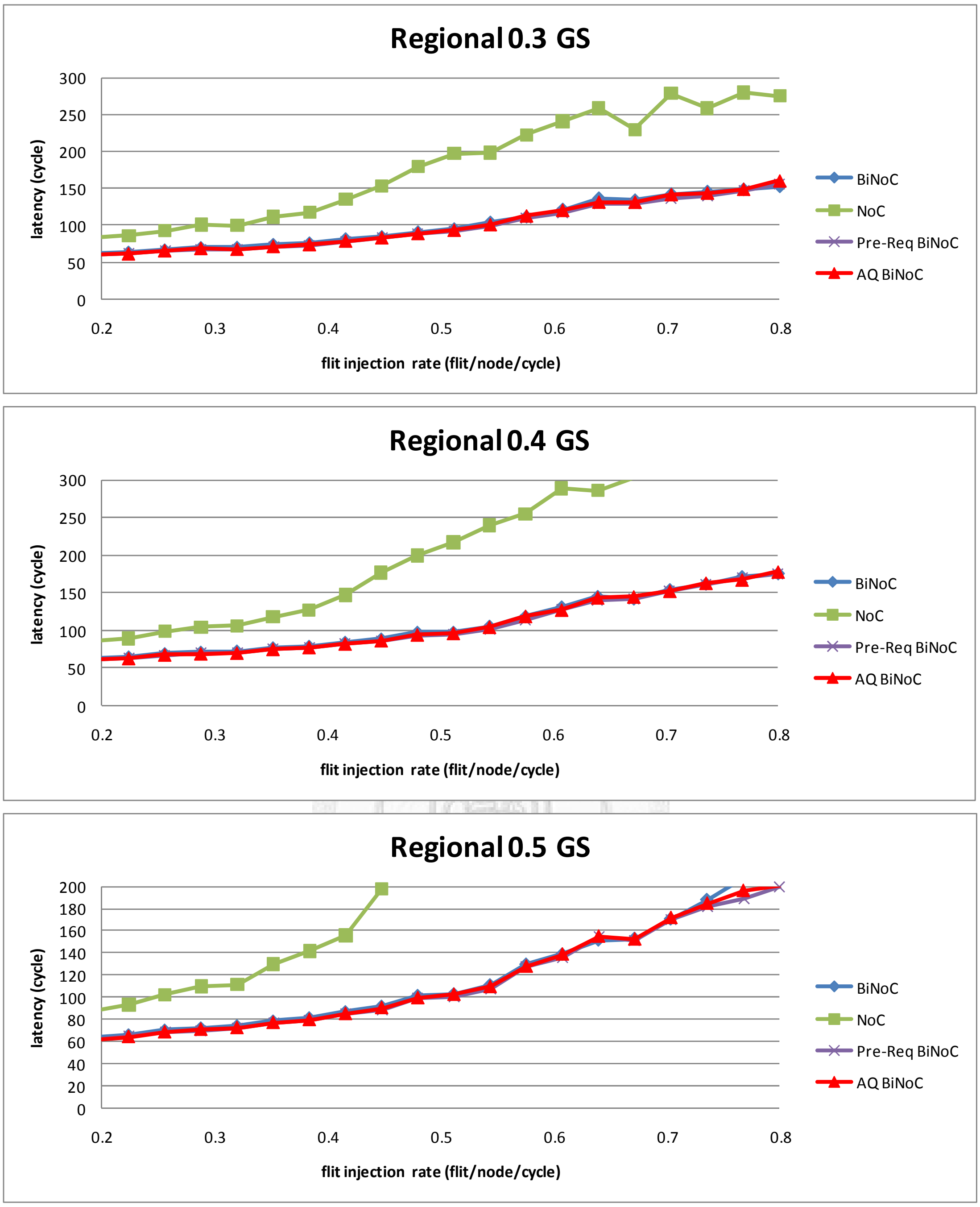
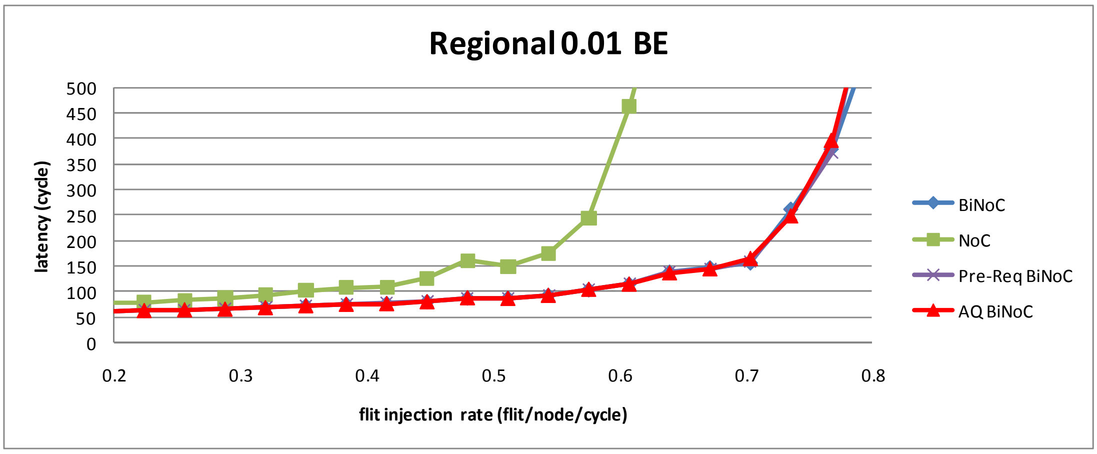
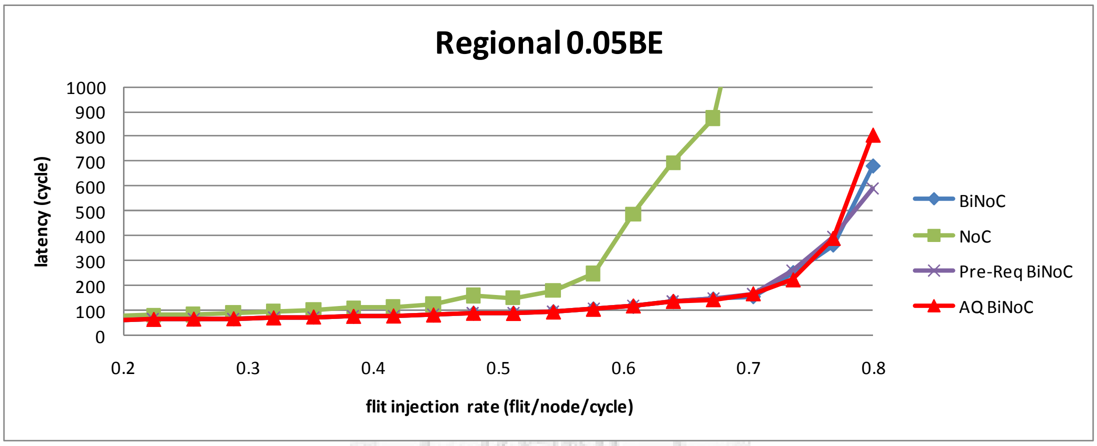
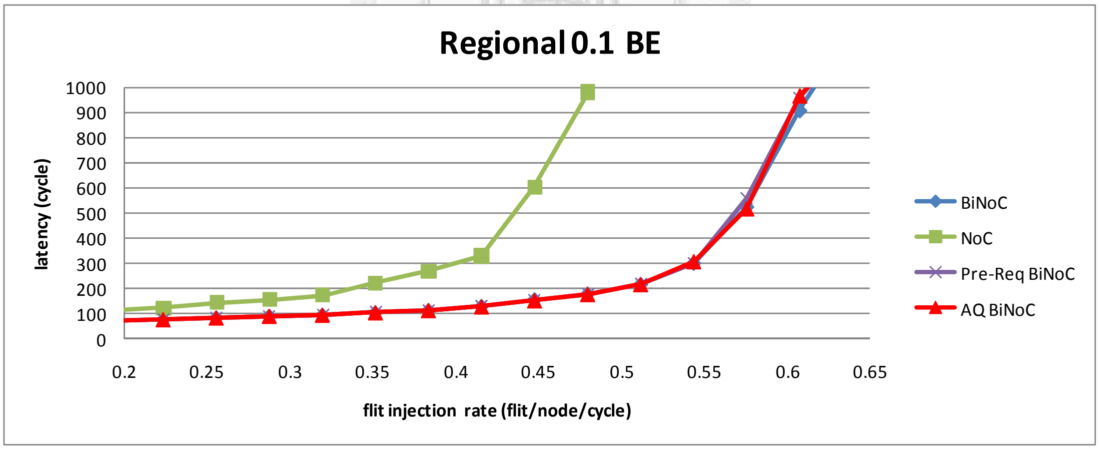
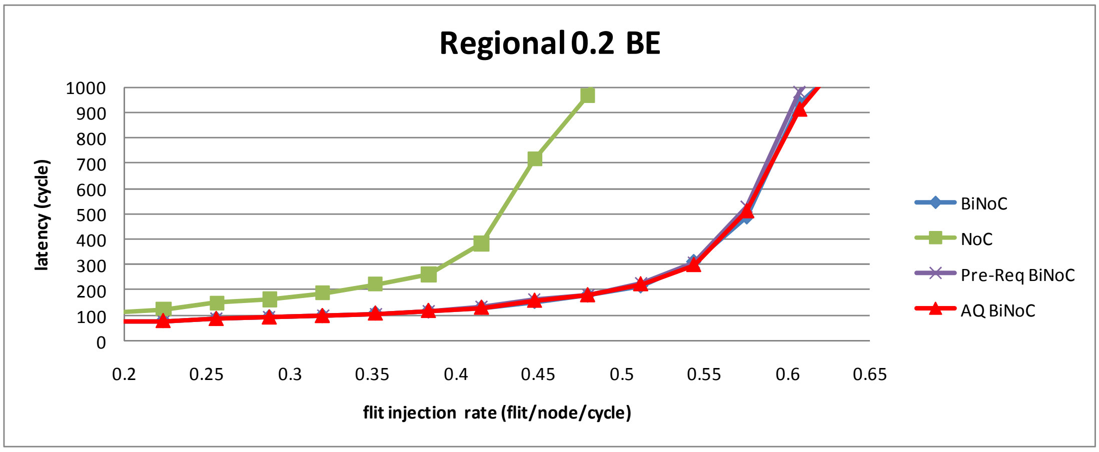


Fig. 5-4. Latency of GS Packet in Regional Traffic.


Regional 0.01 BE
latency (cycle)
500
450
400
350
300
250
200
150
100
50
0
0.2
0.3
0.4
0.5
0.6
0.7
0.8
flit injection rate (flit/node/cycle)
BiNoC
NoC
Pre-Req BiNoC
AQ BiNoC


Regional 0.05BE
latency (cycle)
1000
900
800
700
600
500
400
300
200
100
0
0.2
0.3
0.4
0.5
0.6
0.7
0.8
flit injection rate (flit/node/cycle)
BiNoC
NoC
Pre-Req BiNoC
AQ BiNoC


Regional 0.1 BE
latency (cycle)
1000
900
800
700
600
500
400
300
200
100
0
0.2
0.25
0.3
0.35
0.4
0.45
0.5
0.55
0.6
0.65
flit injection rate (flit/node/cycle)
BiNoC
NoC
Pre-Req BiNoC
AQ BiNoC


Regional 0.2 BE
latency (cycle)
1000
900
800
700
600
500
400
300
200
100
0
0.2
0.25
0.3
0.35
0.4
0.45
0.5
0.55
0.6
0.65
flit injection rate (flit/node/cycle)
BiNoC
NoC
Pre-Req BiNoC
AQ BiNoC

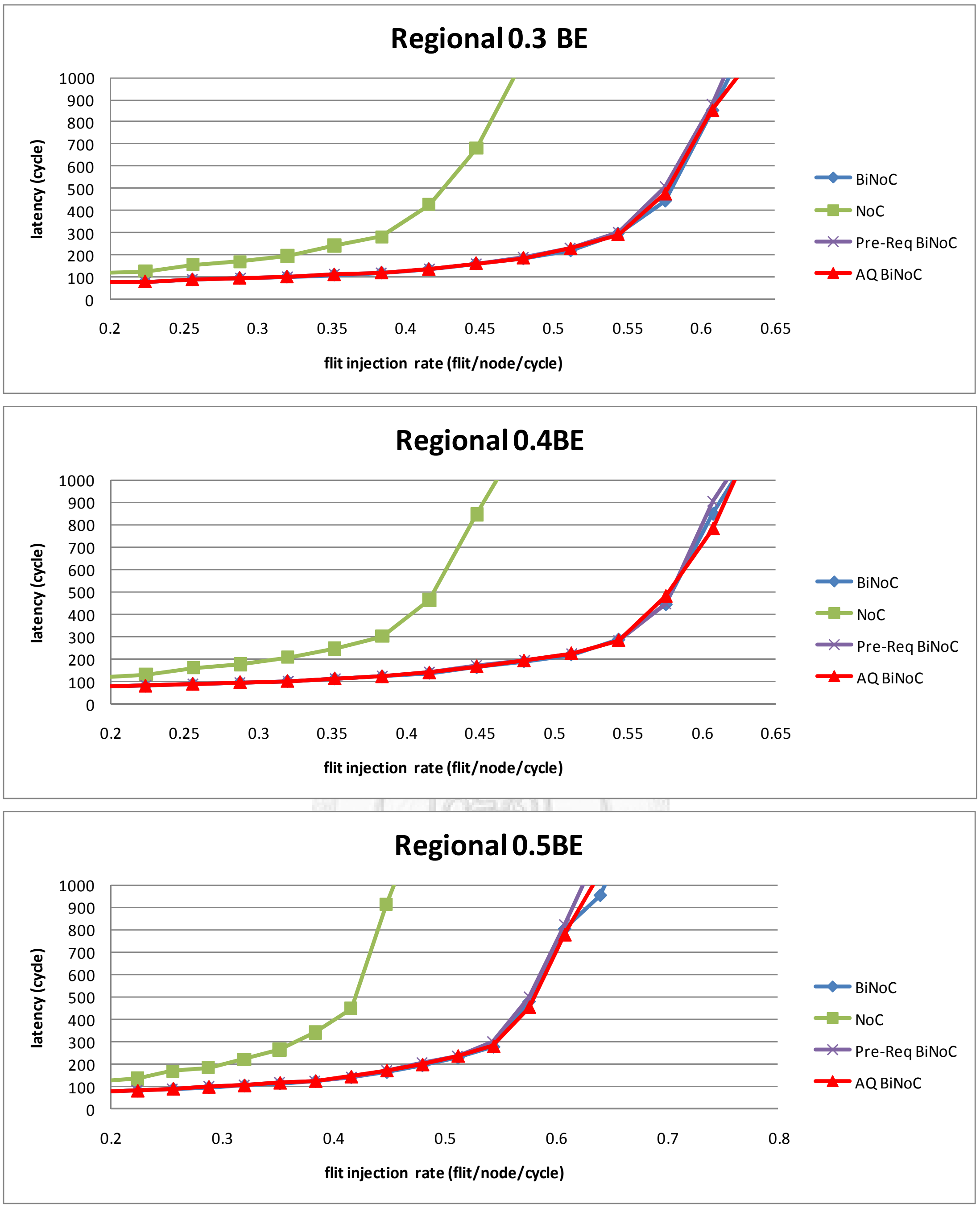
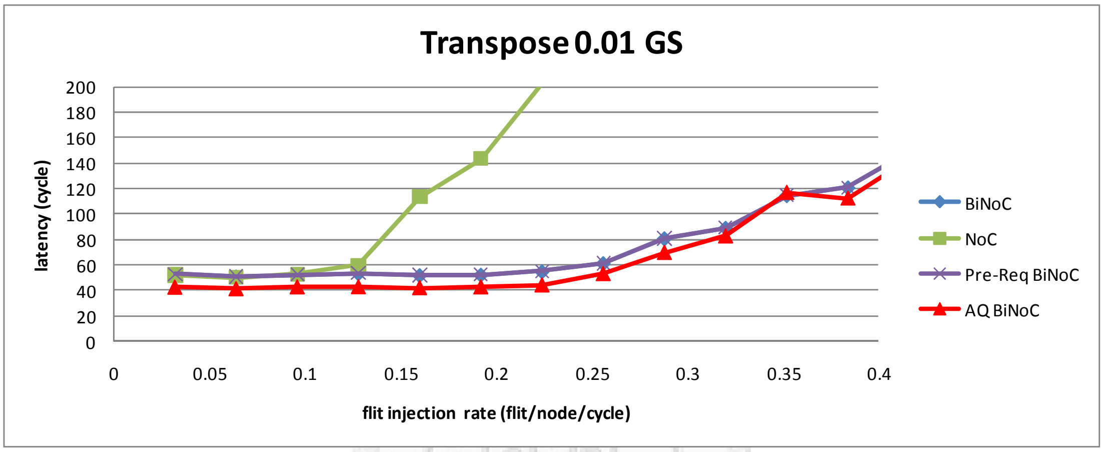
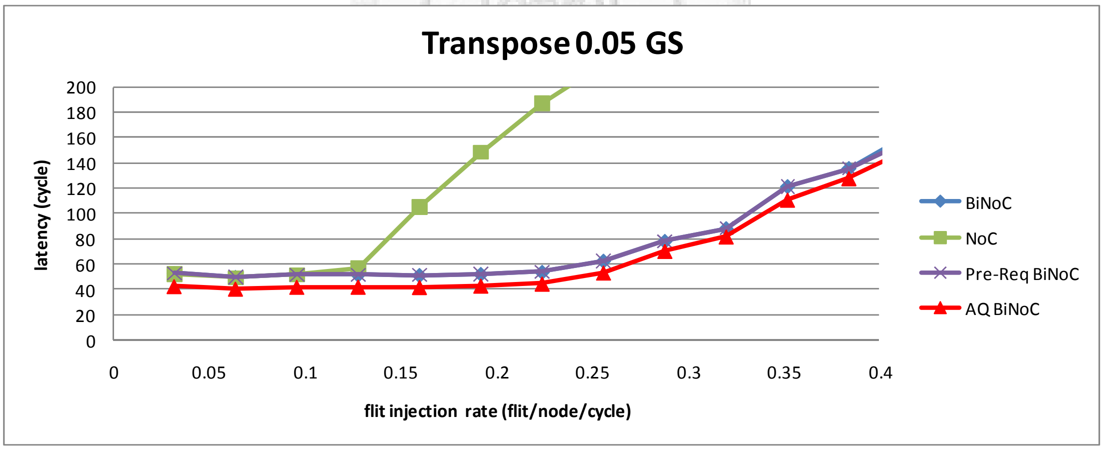
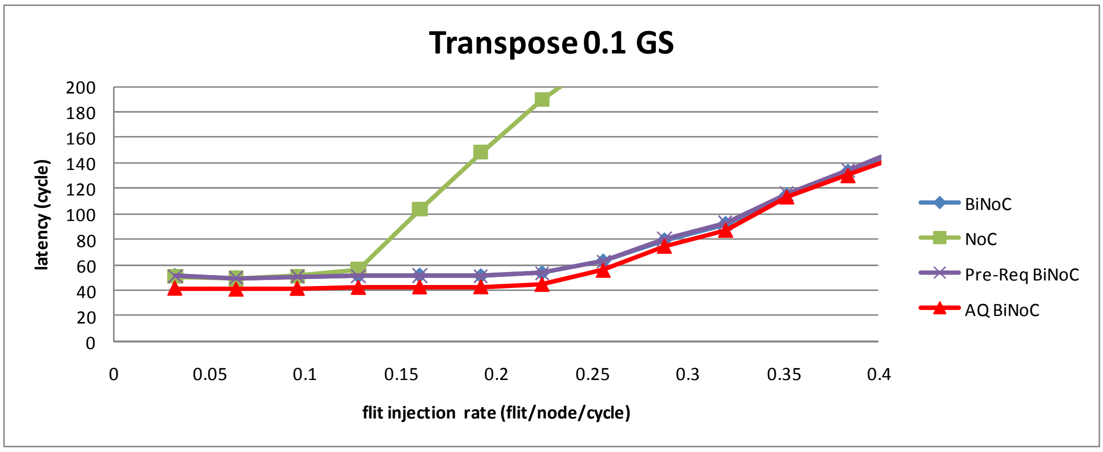


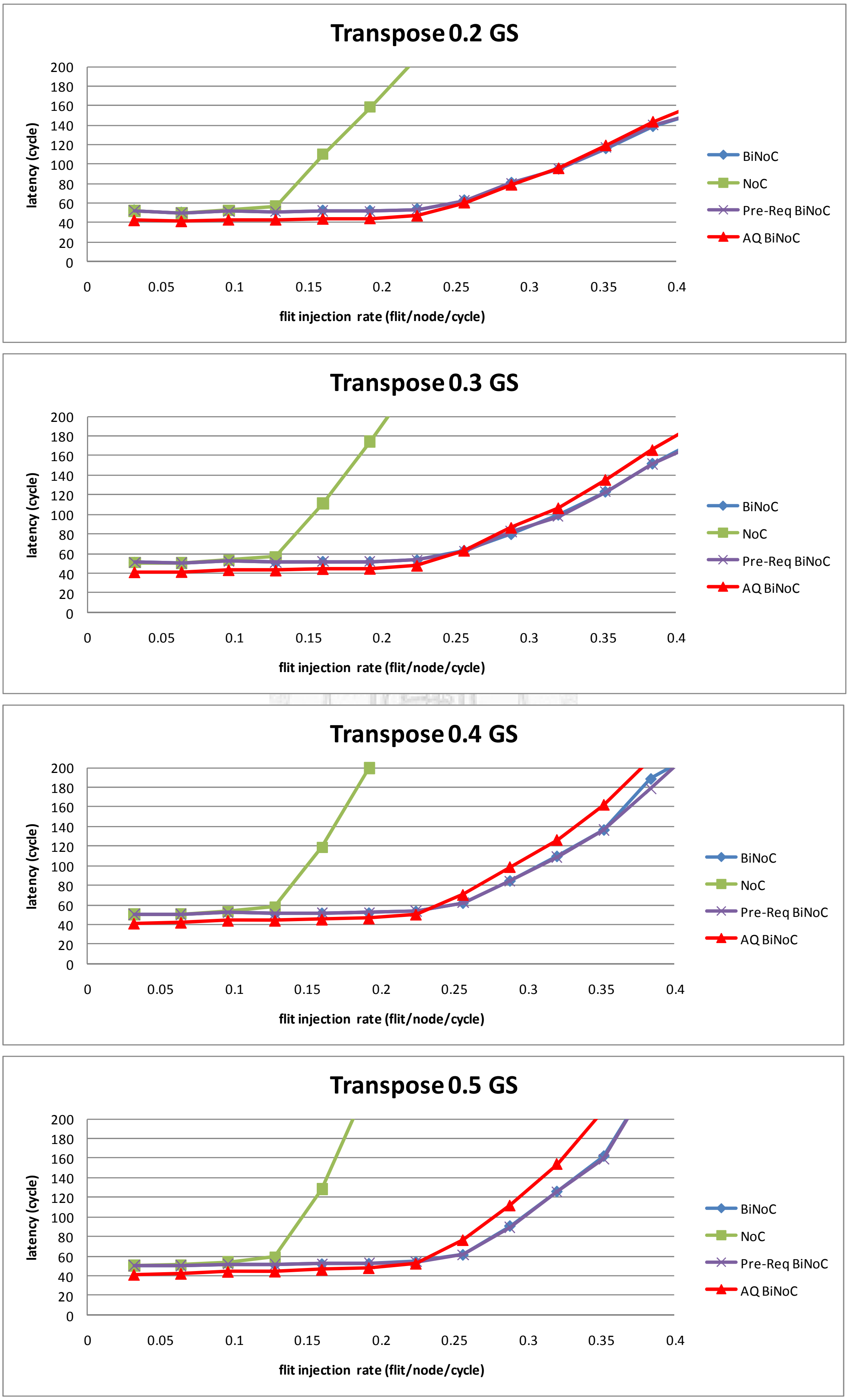
Fig. 5-5. Latency of BE Packet in Regional Traffic.

The results of latency of GS packets in transpose traffic are illustrated in Fig. 5-6. The latency improvement is decreased as the GS percentage grows. The latency of AQ-BiNoC is worse than the original BiNoC as the latency grows up to twice of the zero-load latency. The reason of this effect may be that since the transpose traffic causes congestion at router coordinate (*i,i*), the penetration ability is limited by half of the network. And the anticipative bi-direction channel control is also useless because the transpose traffic data has only one direction to flow. The latency of BE packets are shown in Fig. 5-7. The Pre-Req BiNoC has similar behavior with BiNoC. Even in high GS percentage, it also can achieve lower latency than AQ BiNoC without losing one virtual channel buffer transform into penetration buffer.







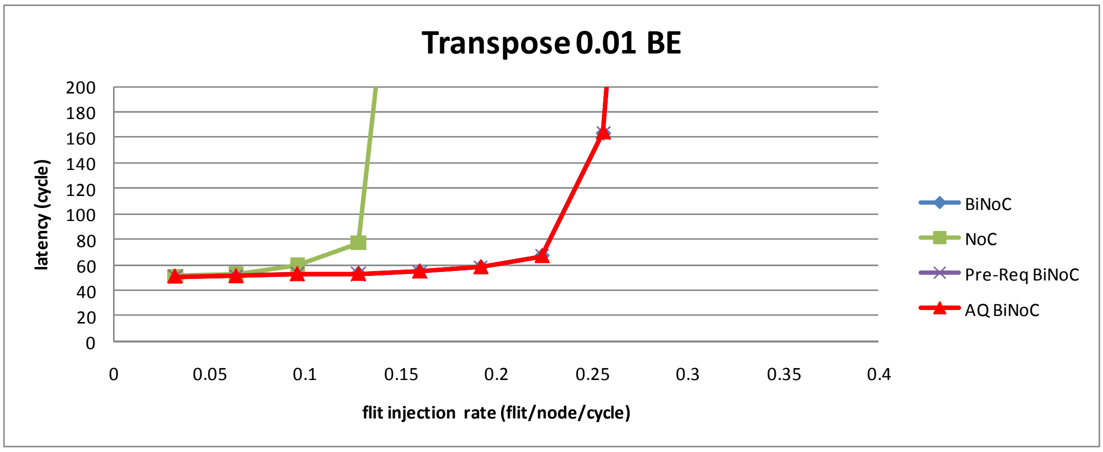

Transpose 0.2 GS

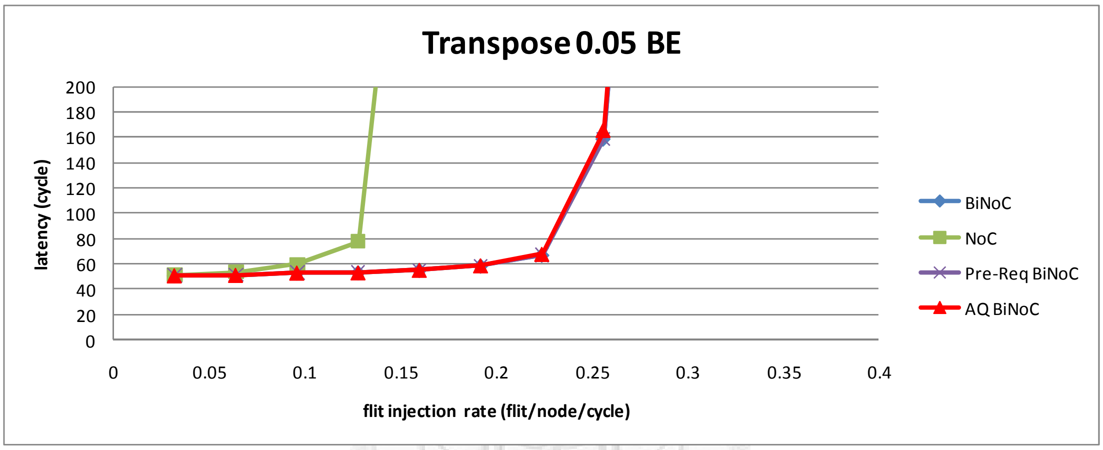
Transpose 0.3 GS

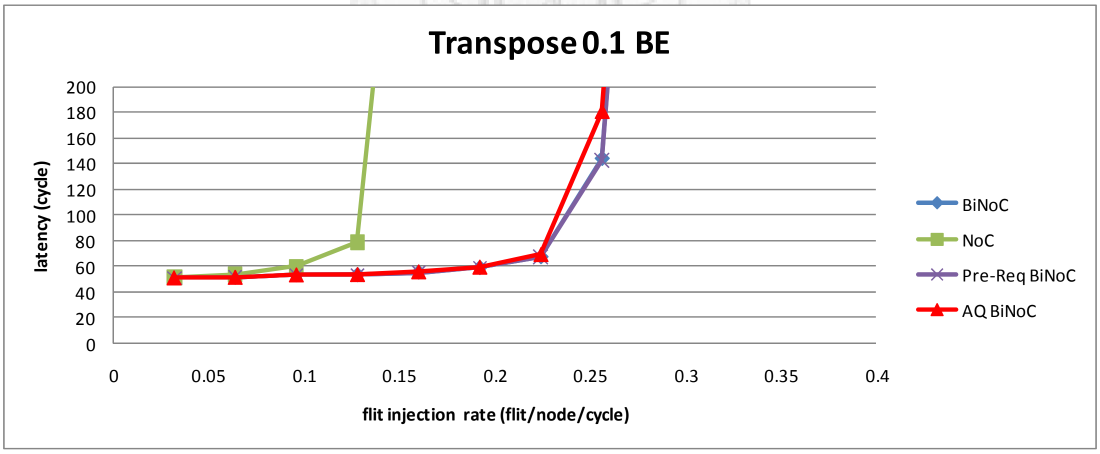
Transpose 0.4 GS

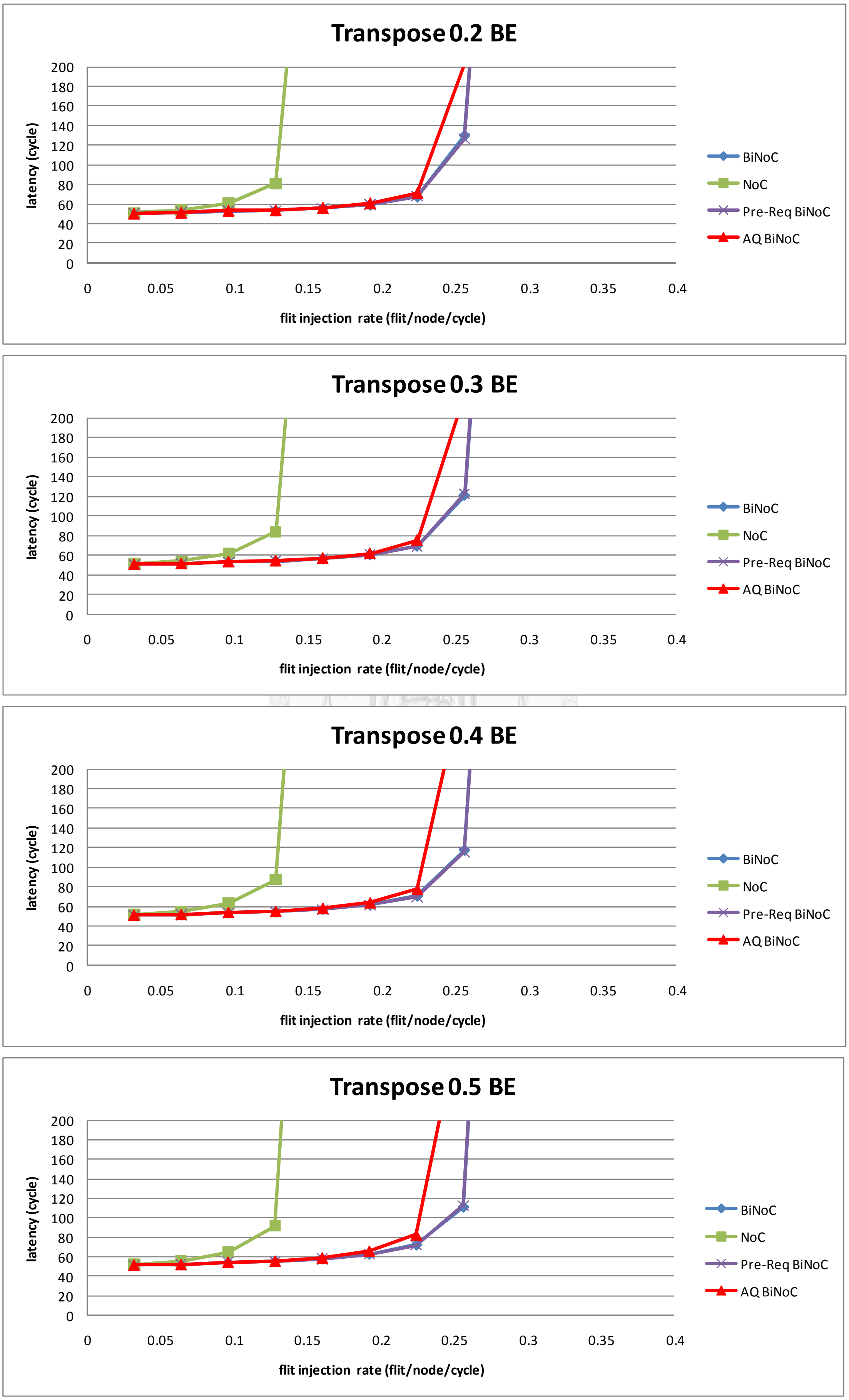
Transpose 0.5 GS

Fig. 5-6. Latency of GS Packet in Transpose Traffic.


Transpose 0.01 BE
latency (cycle)
200
180
160
140
120
100
80
60
40
20
0
0
0.05
0.1
0.15
0.2
0.25
0.3
0.35
0.4
flit injection rate (flit/node/cycle)
BiNoC
NoC
Pre-Req BiNoC
AQ BiNoC


Transpose 0.05 BE
latency (cycle)
200
180
160
140
120
100
80
60
40
20
0
0
0.05
0.1
0.15
0.2
0.25
0.3
0.35
0.4
flit injection rate (flit/node/cycle)
BiNoC
NoC
Pre-Req BiNoC
AQ BiNoC


Transpose 0.1 BE
latency (cycle)
200
180
160
140
120
100
80
60
40
20
0
0
0.05
0.1
0.15
0.2
0.25
0.3
0.35
0.4
flit injection rate (flit/node/cycle)
BiNoC
NoC
Pre-Req BiNoC
AQ BiNoC



Transpose 0.2 BE

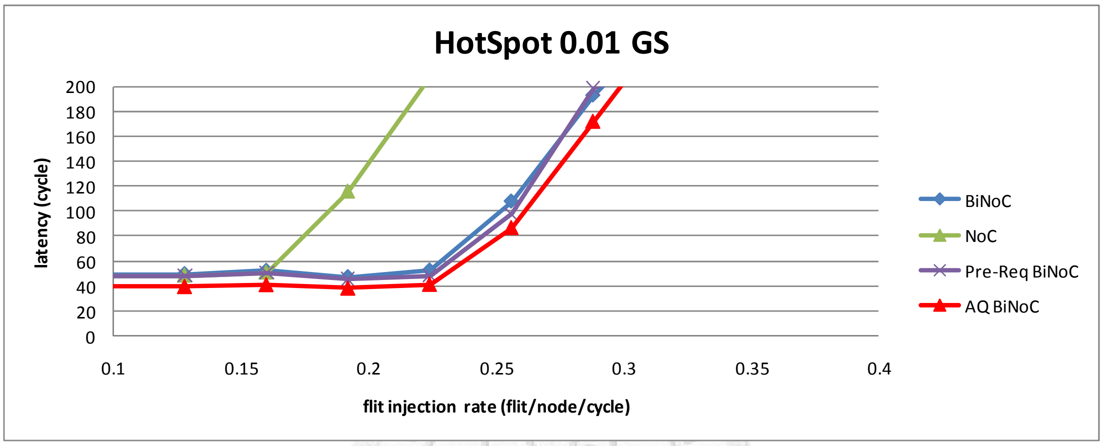
Transpose 0.3 BE

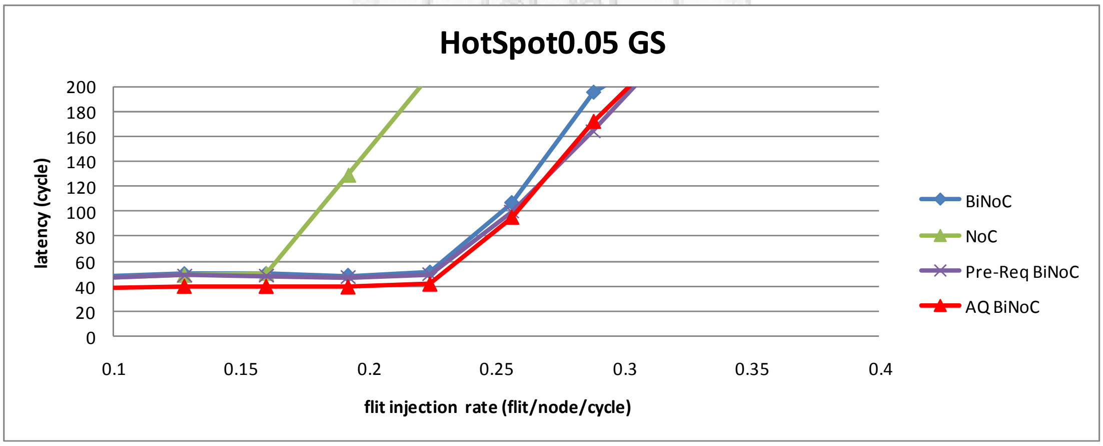
Transpose 0.4 BE

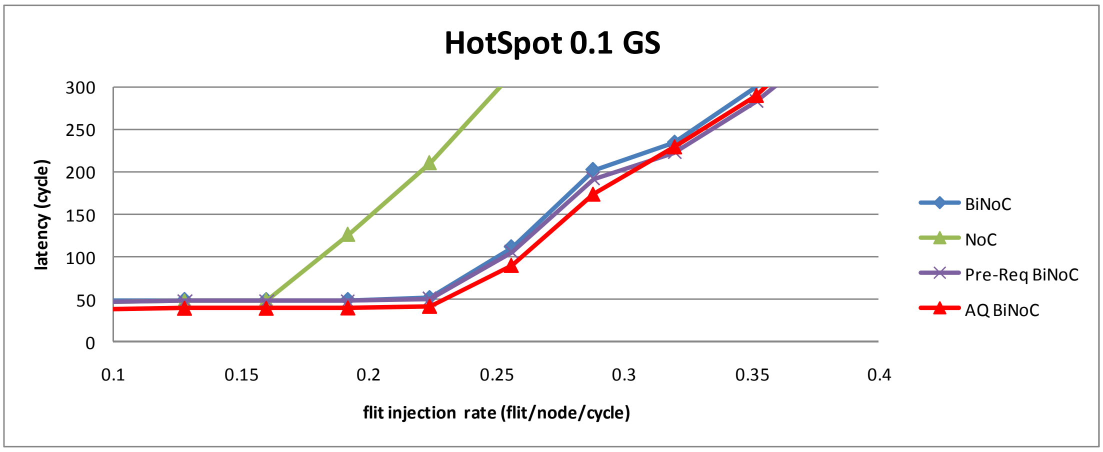
Transpose 0.5 BE

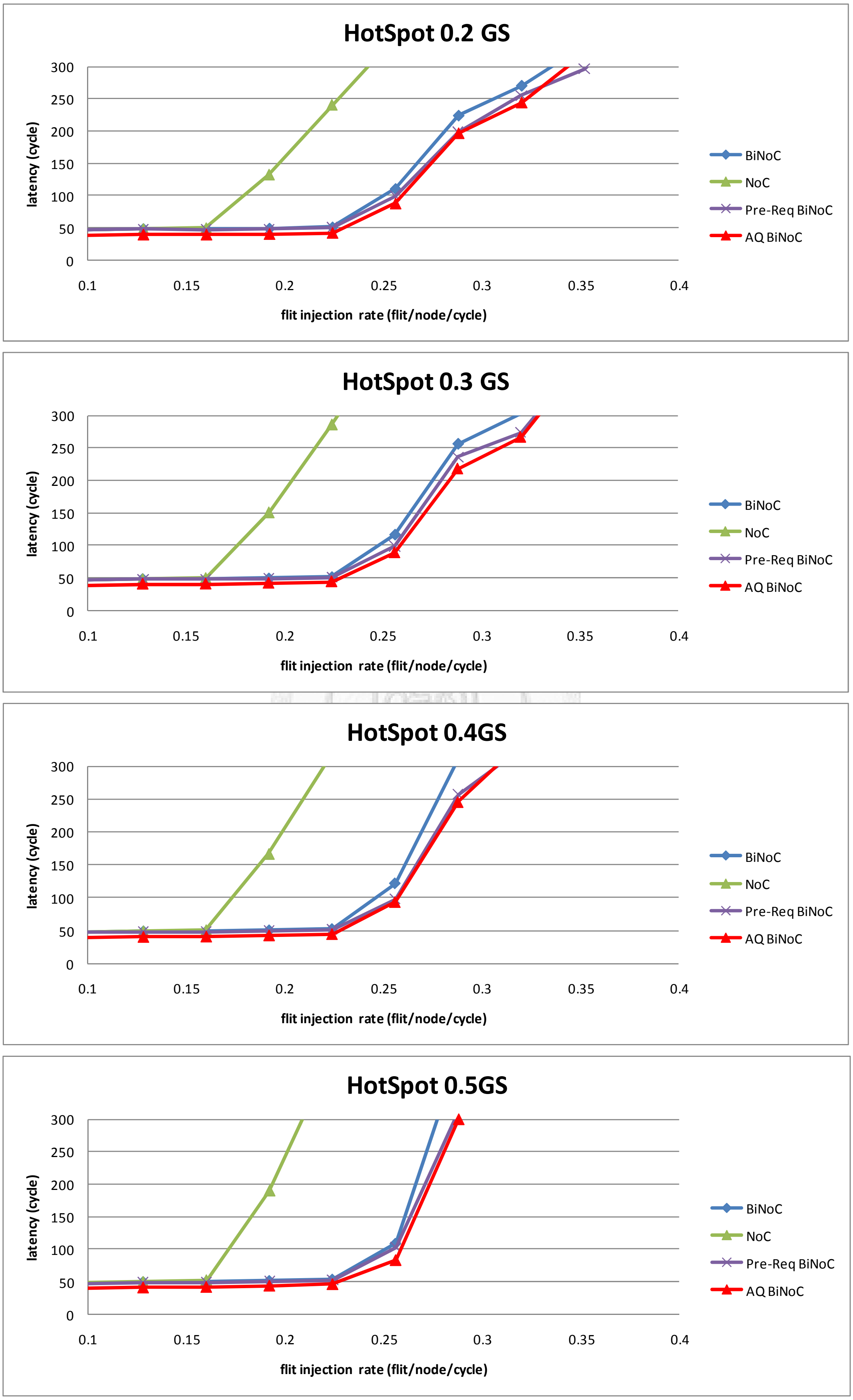
Fig. 5-7. Latency of BE Packet in Transpose Traffic.

The GS and BE latencies in HoSpot traffic are illustrated in Fig. 5-8 and Fig. 5-9. The saturation behavior in HotSpot traffic is worse than other traffics because under high utilization at small area, a resource of single router will be consumed out at low injection rate. We can discover under the burst traffic AQ-BiNoC still got better latency even in high GS percentage.









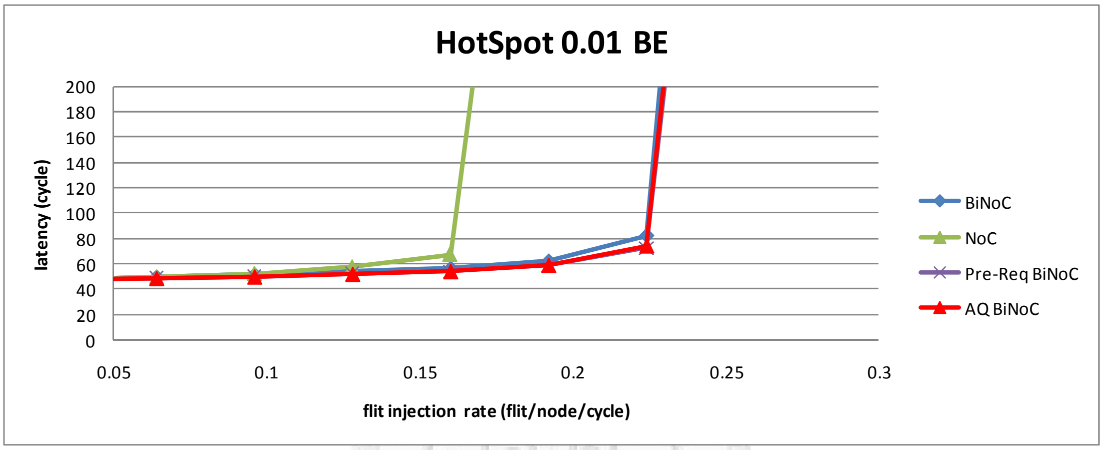
HotSpot 0.2 GS

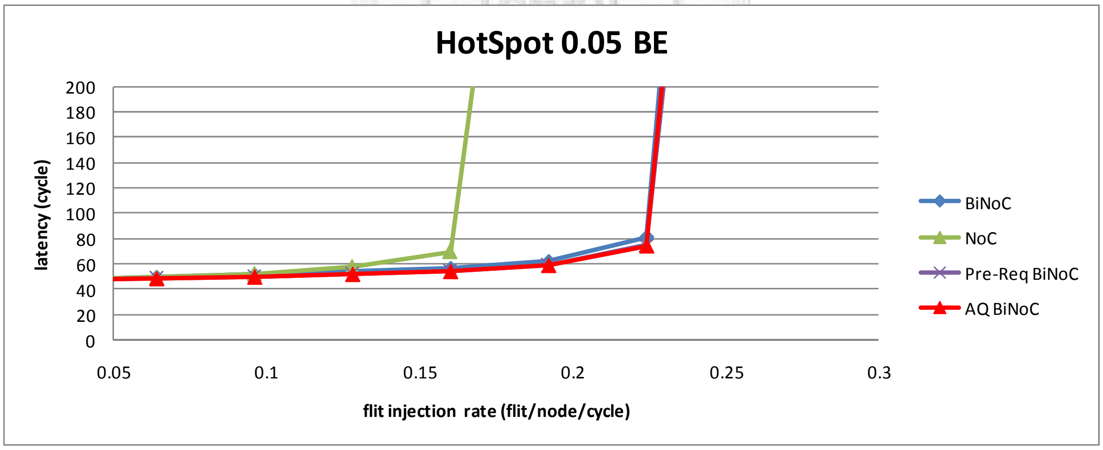
HotSpot 0.3 GS

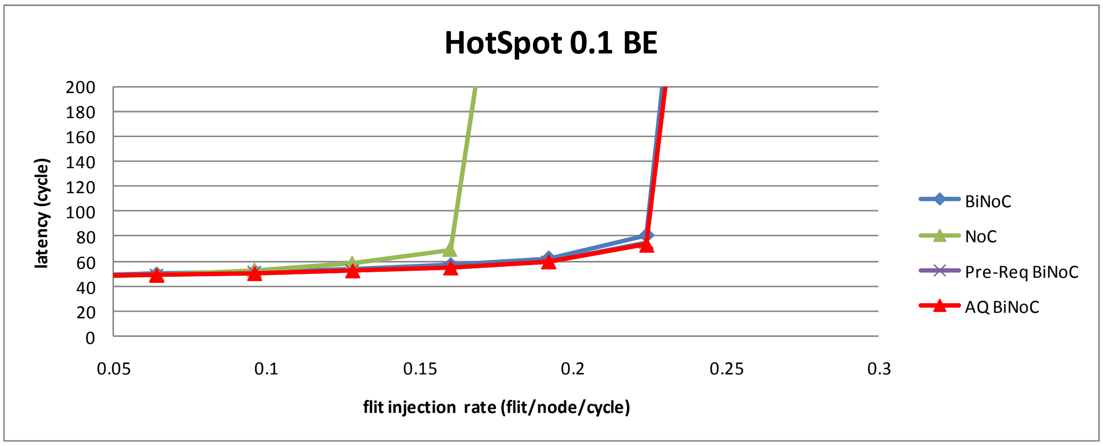
HotSpot 0.4GS

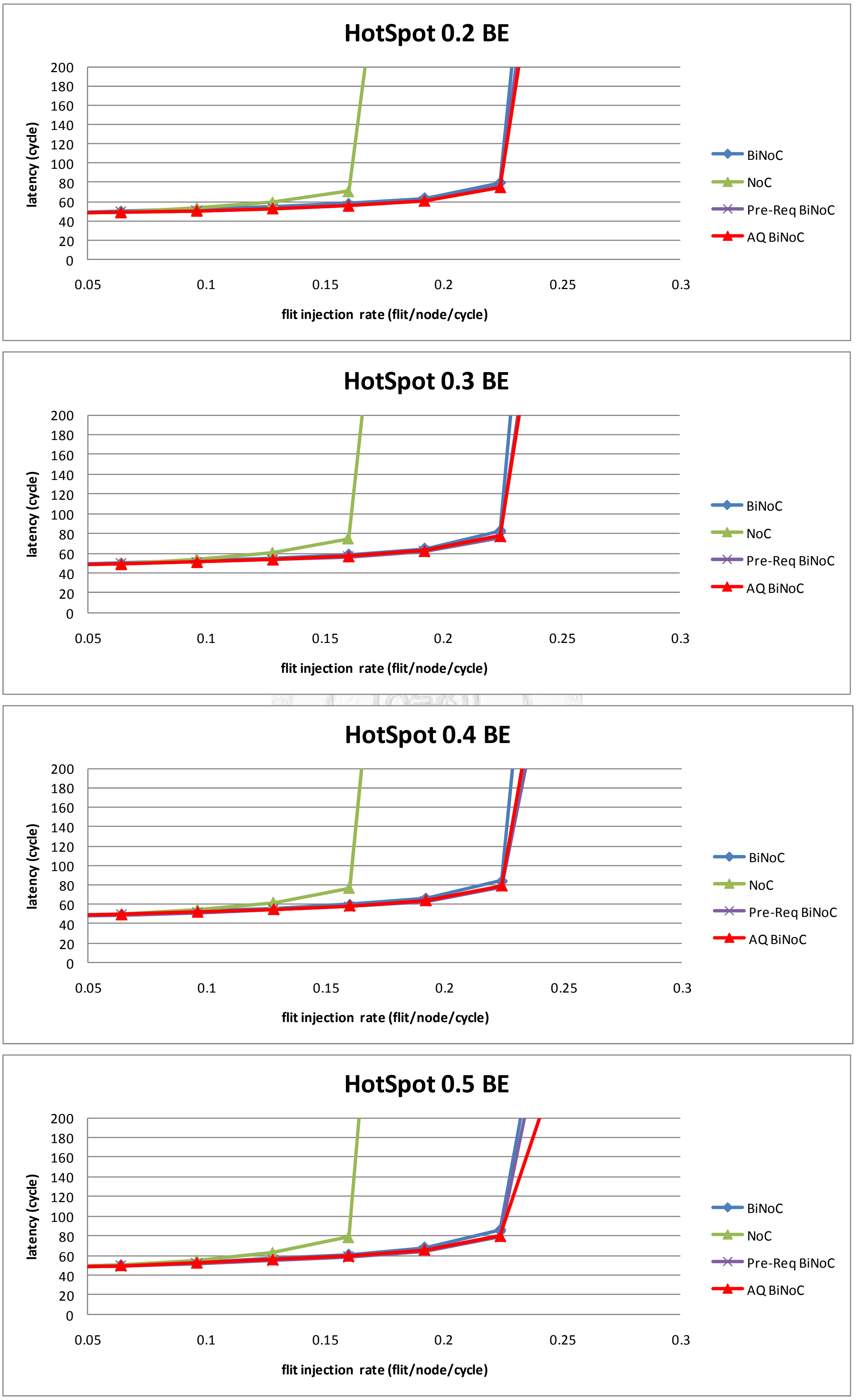
HotSpot 0.5GS

Fig. 5-8. Latency of GS Packet in HotSpot Traffic.


HotSpot 0.01 BE
latency (cycle)
200
180
160
140
120
100
80
60
40
20
0
0.05
0.1
0.15
0.2
0.25
0.3
flit injection rate (flit/node/cycle)
BiNoC
NoC
Pre-Req BiNoC
AQ BiNoC


HotSpot 0.05 BE
latency (cycle)
200
180
160
140
120
100
80
60
40
20
0
0.05
0.1
0.15
0.2
0.25
0.3
flit injection rate (flit/node/cycle)
BiNoC
NoC
Pre-Req BiNoC
AQ BiNoC


HotSpot 0.1 BE
latency (cycle)
200
180
160
140
120
100
80
60
40
20
0
0.05
0.1
0.15
0.2
0.25
0.3
flit injection rate (flit/node/cycle)
BiNoC
NoC
Pre-Req BiNoC
AQ BiNoC



HotSpot 0.2 BE

HotSpot 0.3 BE

HotSpot 0.4 BE

HotSpot 0.5 BE

Fig. 5-9. Latency of BE Packet in HotSpot Traffic.

Figures 5-10 and 5-11 illustrate the latencies of AQ-BiNoC vs. BiNoC and NoC, respectively. Even under the condition where the GS percentage and the injection rate are high, we can observe that the AQ-BiNoC has better latency performance than BiNoC and NoC architectures under uniform traffic.

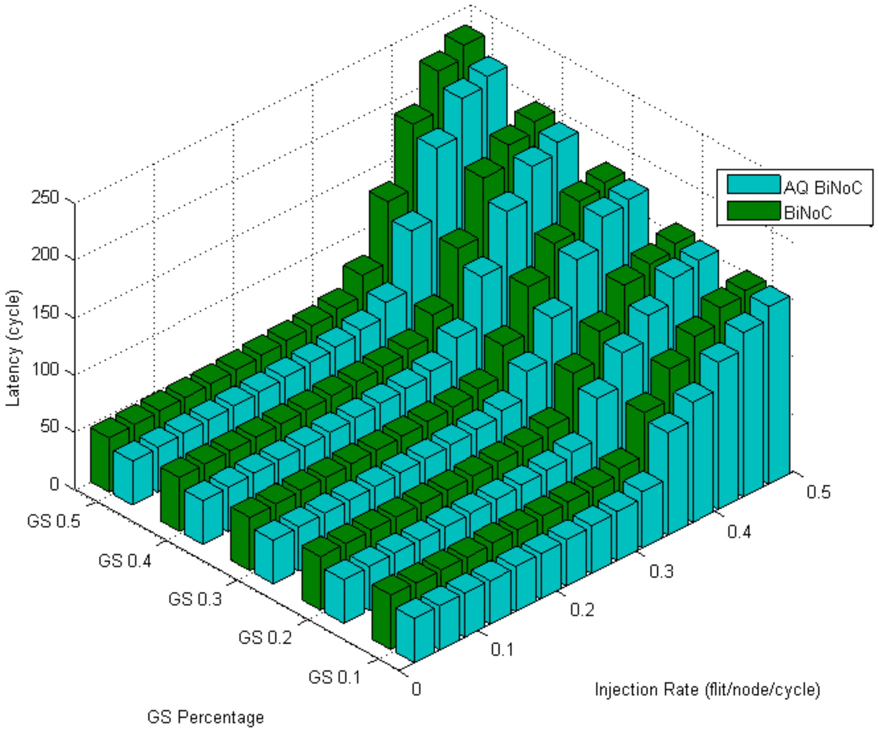


Fig. 5-10. Latency of AQ-BiNoC vs. BiNoC in Uniform Traffic.

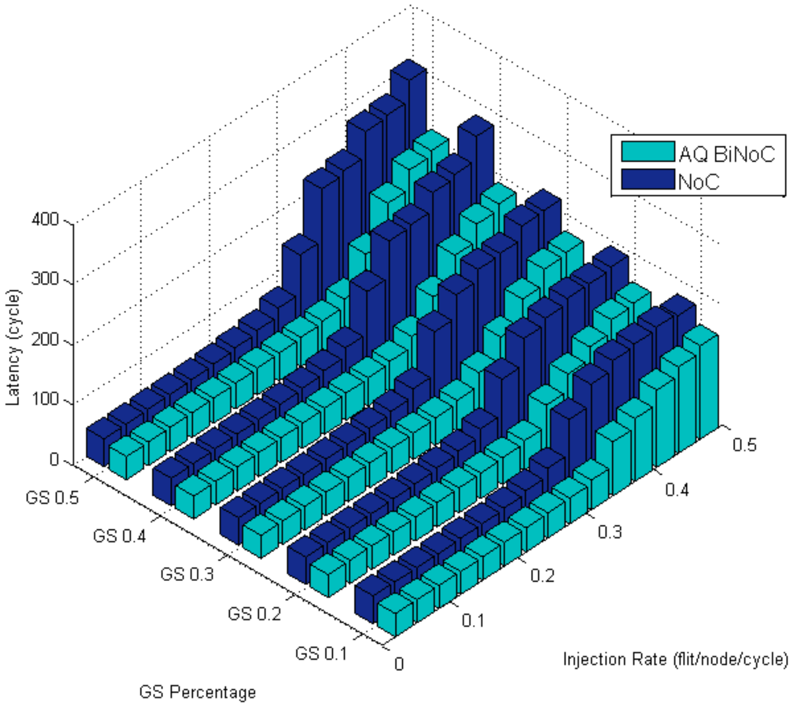


Fig. 5-11. Latency of AQ-BiNoC vs. NoC in Uniform Traffic.

Analyzing the overall performance improvement of uniform traffic as shown in Fig. 5-12, we can find that under higher GS packet percentage, the improvement scale becomes smaller. It is reasonable because performance will become worse as same resources like penetration buffers are shared by more requesters. But the improvement is still effective under higher GS percentage in uniform traffic. This phenomenon can be realized by the characteristic of uniform traffic where the packet flows are separated into whole network equally. Thus, the resources can be shared averagely to whole GS packets which need high transmission quality.

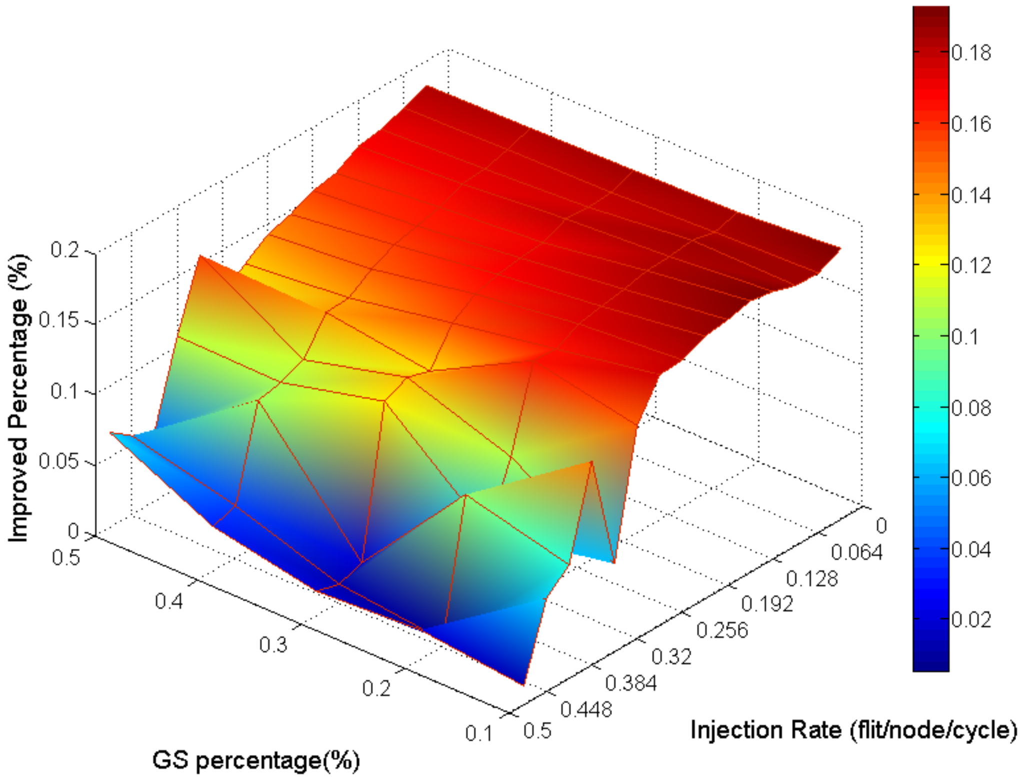


Fig. 5-12. Improved Percentage of AQ-BiNoC vs. BiNoC in Uniform Traffic.

Figures 5-13 and 5-14 illustrate the latencies in regional traffic of AQ-BiNoC vs. BiNoC and NoC, respectively. Even under the condition where the GS percentage and the injection rate are high, we can observe that the AQ-BiNoC has better latency performance than BiNoC and NoC architectures. The regional traffic sends packets to the destinations which are close to the source coordinates. The improvement is mainly provided by the inter router channel arbitration functionality.

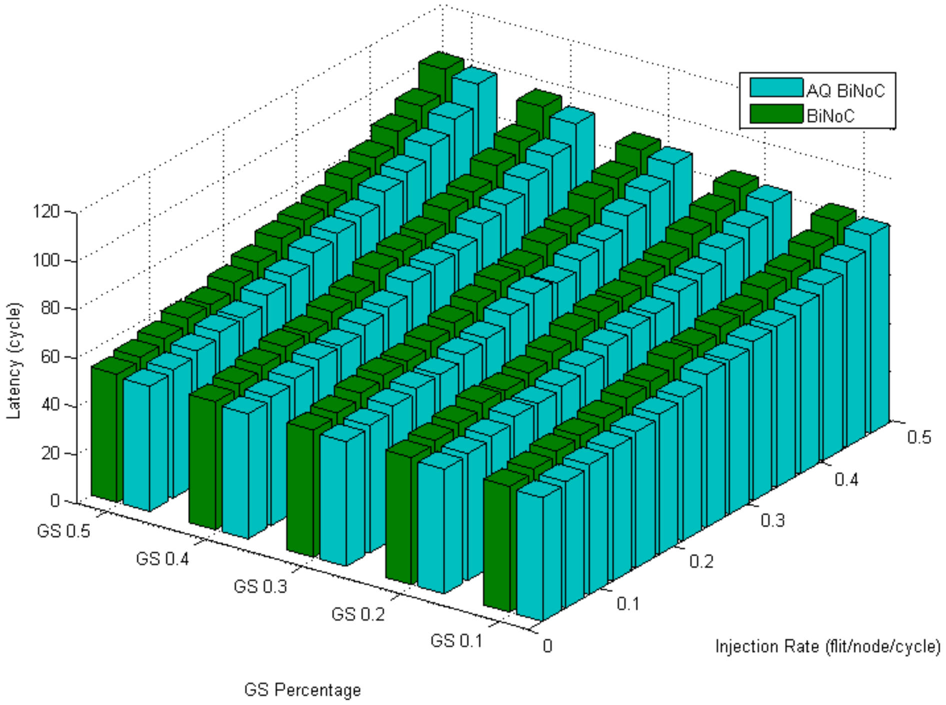


Fig. 5-13. Latency of AQ-BiNoC vs. BiNoC in Regional Traffic.

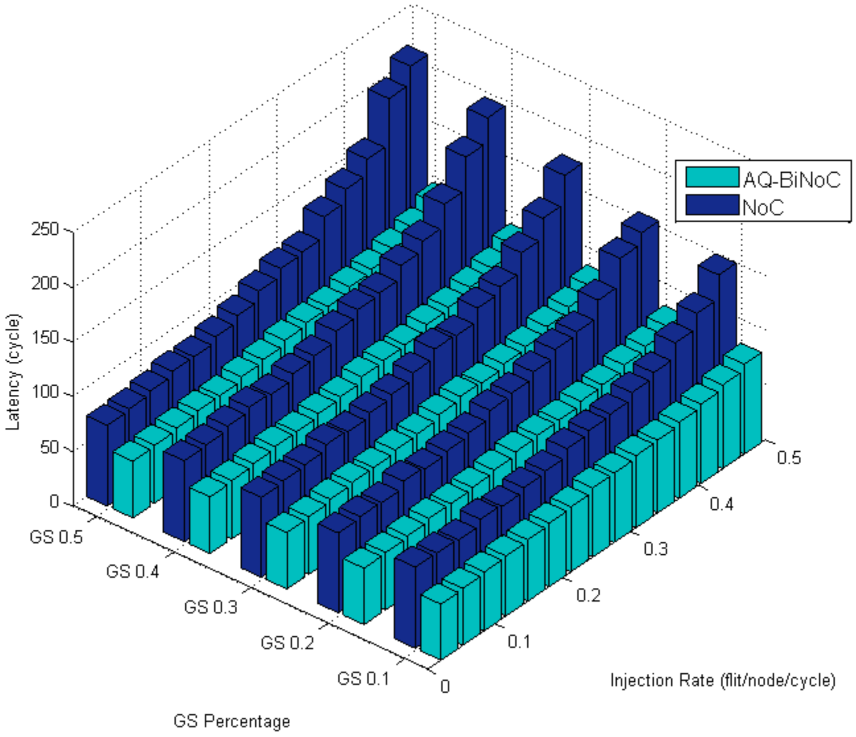


Fig. 5-14. Latency of AQ-BiNoC vs. NoC in Regional Traffic.

Analyzing the overall performance improvement of regional traffic as shown in Fig. 5-15, we found that the improved percentage is generally minor than it in uniform traffic. But we can observe that under higher GS packet percentage, the improvement still does not decay critically.

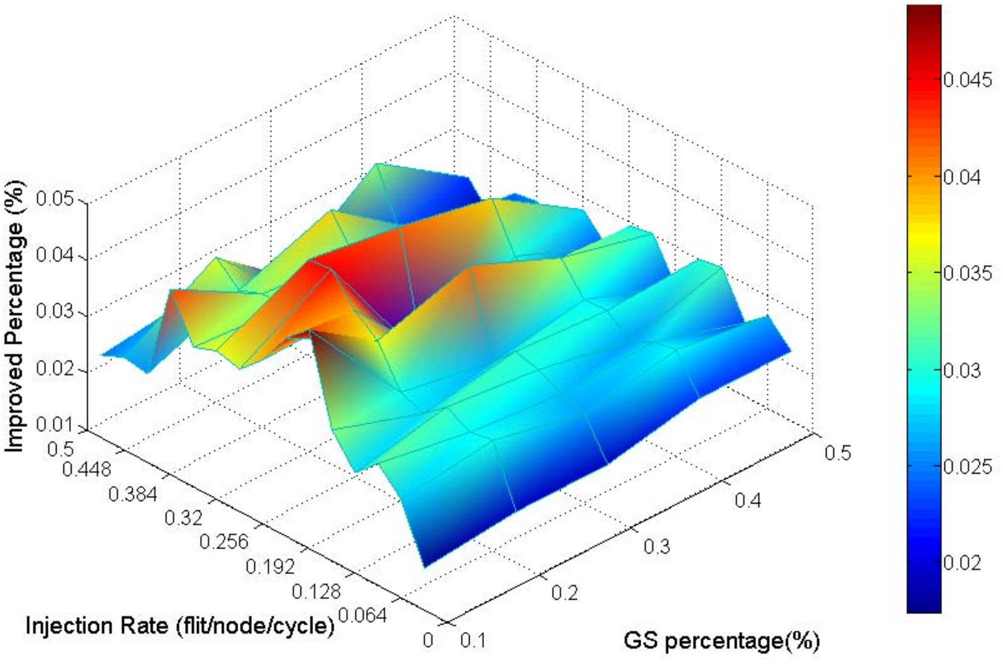


Fig. 5-15. Improved Percentage of AQ-BiNoC vs. BiNoC in Regional Traffic.

Figures 5-16 and 5-17 show the latencies in transpose traffic of AQ-BiNoC vs. BiNoC and NoC, respectively. The performance becomes worse when the GS percentage and the injection rate are high, but AQ-BiNoC has better latency performance than BiNoC and NoC architectures under low injection rate. Since transpose traffic sends packets to the destinations which coordinate is the transpose value of the sources, most of the packet flow will stuck and against each other at the routers which coordinate is (i,i) . Comparing with BiNoC, the AQ-BiNoC router gets less sources; thus, the congestion will be much serious than the competition in BiNoC.

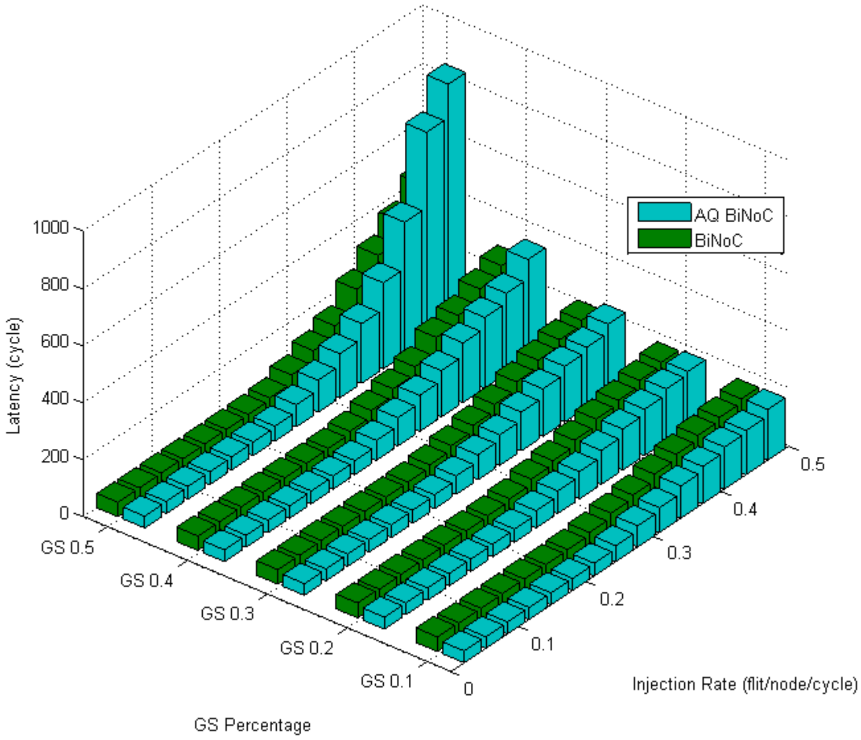


Fig. 5-16. Latency of AQ-BiNoC vs. BiNoC in Transpose Traffic.

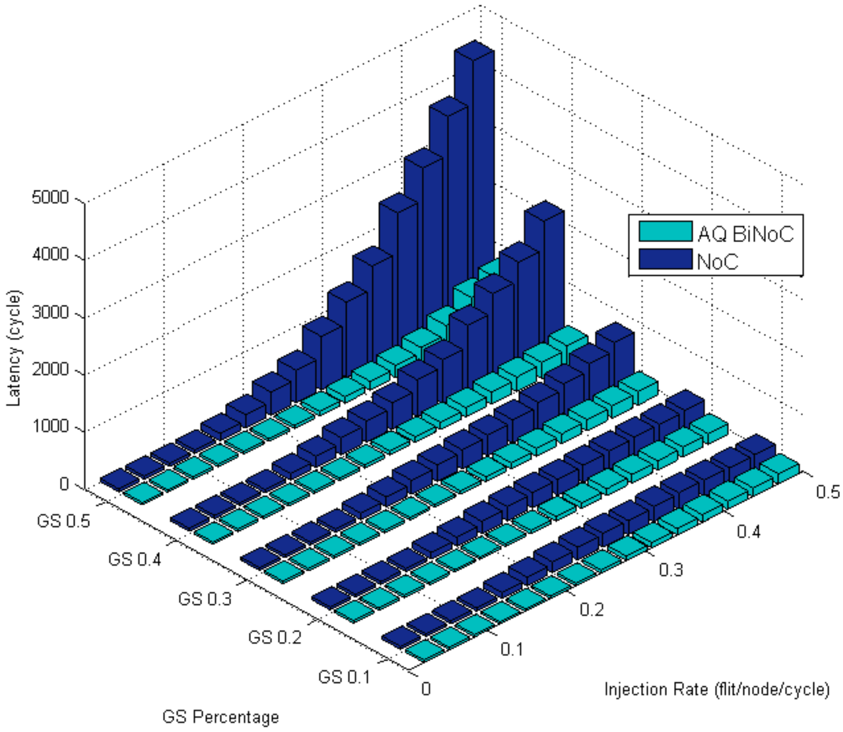


Fig. 5-17. Latency of AQ-BiNoC vs. NoC in Transpose Traffic.

Figure 5-18 emphasizes this phenomenon by the worse latency under high GS percentage and high injection rate.

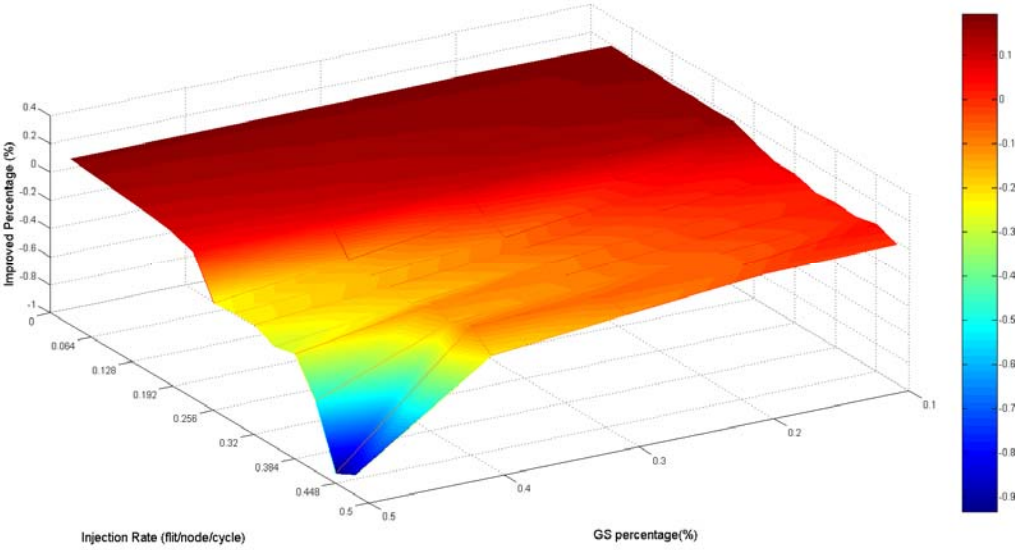


Fig. 5-18. Improved Percentage of AQ-BiNoC vs. BiNoC in Transpose Traffic.

Figures 5-19 and 5-20 show the latencies in hotspot traffic of AQ-BiNoC vs. BiNoC and NoC, respectively. The overall performance shows that our proposed AQ-BiNoC works well under such similar to real traffic case.

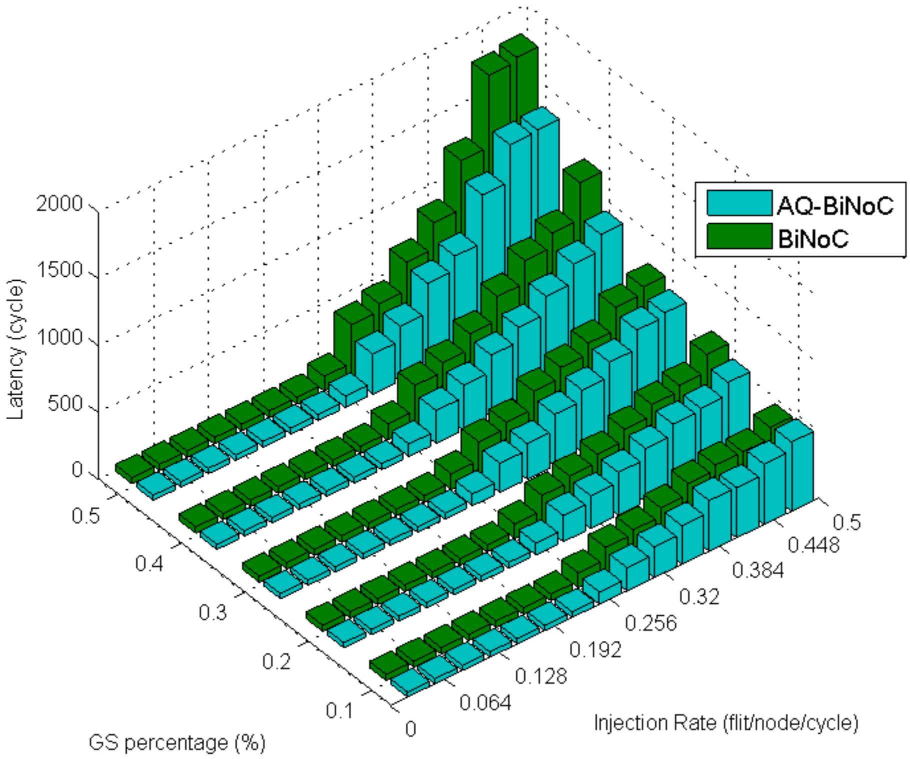


Fig. 5-19. Latency of AQ-BiNoC vs. BiNoC in HotSpot Traffic.

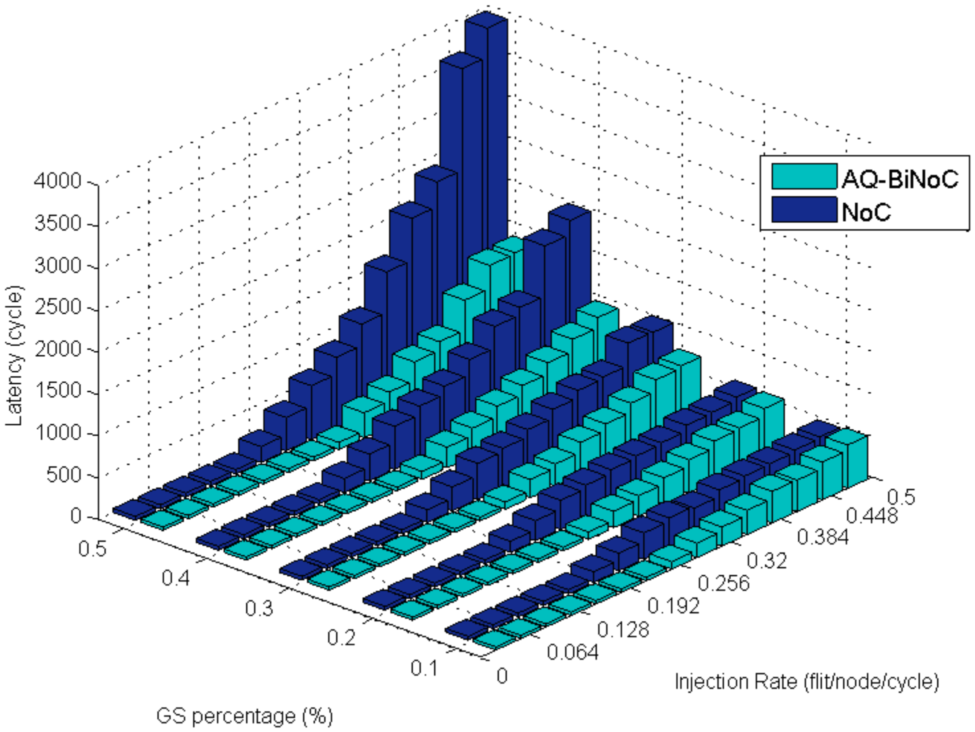


Fig. 5-20. Latency of AQ-BiNoC vs. NoC in HotSpot Traffic.

Figure 5-21 shows the improved percentage of AQ-BiNoC vs. BiNoC. The performance decayed when injection rate goes high. But in overall situation, AQ-BiNoC gets better latency.

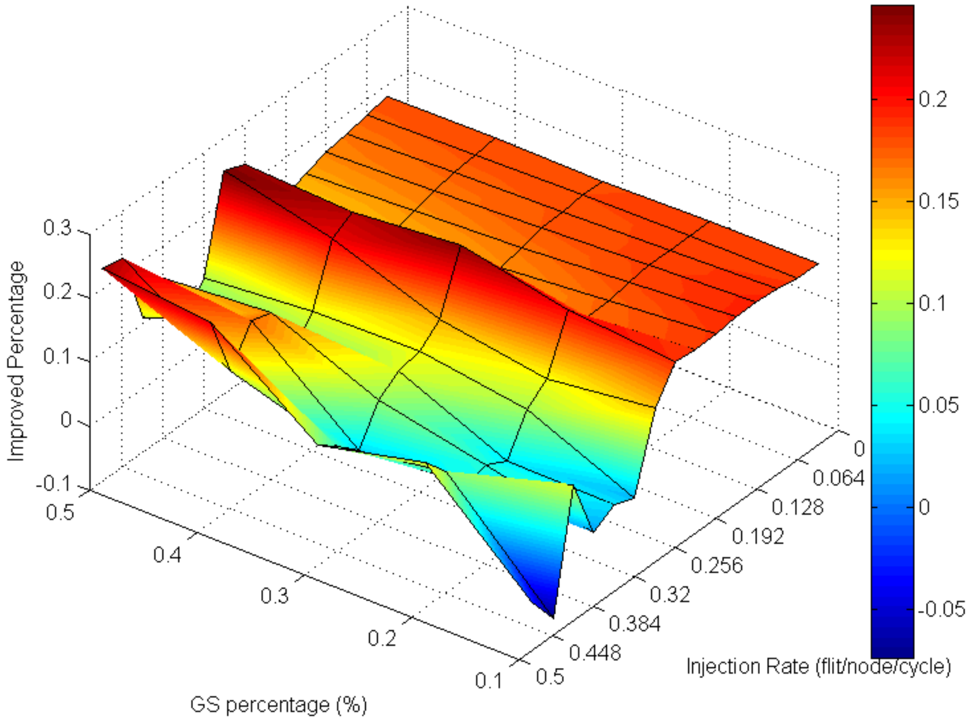


Fig. 5-21. Improved Percentage of AQ-BiNoC vs. NoC in HotSpot Traffic.

5.3 Estimation on Implementation Overhead

Table 5-2 lists the synthesis results of the four architectures compiled in Verilog code with Design Compiler. These router architectures are implemented in UMC 90 nm technology. The timing conditions of all four architectures are fixed at 400Mhz to compare their area and power. And we can notice that the timing, hardware, and power overhead on AQ-NoC vs BiNoC are all less than 5%.

Table 5-2. Implementation Information.

Architecture	Typical-NoC	BiNoC	Pre-Req BiNoC	AQ-BiNoC (proposed architecture)
Cell Area	369743	448010	457994	470490
Power (mw)	42.5285	42.9385	43.0414	43.5554
Normalized Area	82.5%	100%	102.2%	105.1%

Table 5-3. Critical Timing Implementation Information.

Architecture	Typical-NoC	BiNoC	Pre-Req BiNoC	AQ-BiNoC (proposed architecture)
Cycle (ns)	1.75	2.36	2.36	2.42
Normalized Cycle	0.741	1	1	1.02

CHAPTER 6
CONCLUSION

In this Thesis, we proposed a novel BiNoC backbone architecture AQ-BiNoC which features the use of specification in QoS to improve the overall network performance. An inter-router communication scheme achieving effective channel direction switch and less transmission time was designed without huge hardware overhead. As a result, the router architecture and QoS framework can be directly applied to any network structure.

Experimental results using three different featuring traffic patterns verified the performance under different data flow types. The varying GS percentages also provide a flexibility to verify the architecture under different QoS requirements. The results showed that under these different conditions, the network can get overall better latency performance. Finally, the implementation overhead of AQ-BiNoC is negligible compared with BiNoC. Under almost the same hardware cost and specification, AQ-BiNoC provides better performance.

REFERENCE

[1] W.J. Dally and B. Towles, *Principles and Practices of Interconnection Networks*, Morgan Kaufmann, 2004.

[2] S. Kumar, A. Jantsch, J. P. Soininen, M. Forsell, M. Millberg, J. Oberg, K. Tiensyrja and A. Hemani, "A Network on Chip Architecture and Design Methodology," *Proceedings of Computer Society Annual Symposium on VLSI,* pp. 105-112, Apr. 2002.

[3] G. De Micheli and L. Benini, *Networks on Chips*, Morgan Kaufmann, 2006.

[4] A. Jantsch and H. Tenhunen (Eds.), *Networks on Chip*, Kluwer Academic Publishers, 2003.

[5] W.J. Dally and B. Towles, "Route Packets, Not Wires: On-Chip Interconnection Networks," *Proceedings of Design Automation Conference*, pp. 684-689, Jun. 2001.

[6] R. Ho, K.W. Mai and M.A. Horowitz, "The Future of Wires," *Proceedings of the IEEE*, vol. 89, no. 4, pp. 490-504, Apr. 2001.

[7] L. S. Peh and W. J. Dally, "A Delay Model and Speculative Architecture for Pipelined Routers," *Proceedings of International Symposium on High-Performance Computer Architecture,* pp. 255-266, Jan. 2001.

[8] W. J. Dally, "Virtual-Channel Flow Control," *IEEE Transactions on Parallel and Distributed systems,* vol. 3, no. 2, pp. 194-205, Mar. 1992.

[9] M. Lai, Z. Wang, L. Gao, H. Lu and K. Dai, "A Dynamically-Allocated Virtual Channel Architecture with Congestion Awareness for On-Chip Routers," *Proceedings of Design Automation Conference,* pp. 630-633, June 2008.

[10] L. S. Peh, "Flow Control and Micro-Architectural Mechanisms for Extending the Performance of Interconnection Networks," *Ph. D. Thesis,* Stanford University, Aug. 2001.

[11] R. Guerin and V. Peris, "Quality-of-Service in Packet Networks: Basic

Mechanisms and Directions,” *Computer Networks*, vol. 31, no. 3, pp. 169-179, Feb. 1999.

[12] E. Rijpkema, K. Goossens, A. Radulescu, J. Dielissen, J. V. Meerbergen, P. Wielage and E. Waterlander, “Trade-offs in the Design of a Router with both Guaranteed and Best-effort Services for Networks on Chip,” *IEE Proceedings of Computers and Digital Techniques,* vol. 150, no.5, pp. 294- 302, Sept. 2003.

[13] N. Kavaldjiev, G. J. M. Smit, P. G. Jansen and P. T. Wolkotte, “A Virtual Channel Network-on-Chip for GT and BE Traffic,” *Proceedings of IEEE Computer Society Annual Symposium on Emerging VLSI Technologies and Architectures,* pp. 211-216, Mar. 2006.

[14] C. A. Nicopoulos, D. Park, J. Kim, N. Vijaykrishnan, M. S. Yousif and C. R. Das, “ViChaR: A Dynamic Virtual Channel Regulator for Network-on-Chip Routers,” *Proceedings of International Symposium on Microarchitecture,* pp. 333-346, Dec. 2006.

[15] M. Millberg, E. Nilsson, R. Thid and A. Jantsch, “Guaranteed Bandwidth using Looped Containers in Temporally Disjoint Networks within the Nostrum Network on Chip,” *Proceedings of Design Automation and Test in Europe,* vol. 2, pp. 890-895, Feb. 2004.

[16] Z. Guz, E. Bolotin, I. Cidon, R. Ginosar and A. Kolodny, “Efficient Link Capacity and QoS Design for Network-on-Chip,” *Proceedings of Design Automation and Test in Europe,* vol. 1, pp. 1- 6, Mar. 2006.

[17] M. D. Harmanci, N. P. Escudero, Y. Leblebici and P. Ienne, “Providing QoS to Connection-less Packet-Switched NoC by implementing DiffServ Functionalities,” *Proceedings of International Symposium on System-on-Chip,* pp. 37-40, Nov. 2004.

[18] E. Bolotin, I. Cidon, R. Ginosar and A. Kolodny, “QNoC: QoS architecture and Design Process for Network on Chip,” *Elsevier Journal of Systems Architecture,* vol. 50, no. 2-3, pp. 105- 128, Feb. 2004.

[19] M. D. Harmanci, N. P. Escudero, Y. Leblebici and P. Ienne, “Quantitative Modeling and Comparison of Communication Schemes to guarantee Quality-of-Service in Networks-on-Chip,” *IEEE International Symposium on Circuits and Systems,* vol. 2, pp. 1782- 1785, May 2005.

[20] Y.C. Lan, S.H. Lo, Y.C. Lin, Y.H. Hu, and S.J. Chen, “BiNoC: A Bi-directional NoC Architecture with Dynamic Self-Reconfigurable Channel,” *Proceedings of the 3rd ACM/IEEE International Symposium on Networks-on-Chip*, pp. 266-275, May 2009.

[21] S.H. Lo, Y.C. Lan, H.H. Yeh, W.C. Tsai, Y.H. Hu, S.J. Chen, "QoS Aware BiNoC Architecture" *Proceedings of the 24th IEEE International Parallel & Distributed Processing Symposium*, pp. 1-10, May 2010.

[22] A. Kumar, L.-S. Peh, P. Kundu, and N. K. Jha. “Express Virtual Channels: Towards the Ideal Interconnection Fabric,” *Proceedings International Symposium on Computer Architecture*, pp. 150-161, June 2007.

[23] D.C. Gazis, *Traffic Science,* John Wiley & Sons, 1974.