MULTIPATH ROUTER ARCHITECTURES TO REDUCE LATENCY IN NETWORK-ON-CHIPS

A Thesis

by

HRISHIKESH NANDKISHOR DESHPANDE

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

May 2012

Major Subject: Computer Engineering

MULTIPATH ROUTER ARCHITECTURES TO REDUCE LATENCY IN NETWORK-ON-CHIPS

A Thesis

by

HRISHIKESH NANDKISHOR DESHPANDE

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Approved by:

Chair of Committee,	Seong Gwan Choi
Committee Members,	Paul Gratz
	Eun Jung Kim
Head of Department,	Costas N. Georghiades

May 2012

Major Subject: Computer Engineering

ABSTRACT

Multipath Router Architectures to Reduce Latency in Network-on-Chips. (May 2012)

Hrishikesh Nandkishor Deshpande, B.E., Birla Institute of Technology & Science, India

Chair of Advisory Committee: Dr. Seong Gwan Choi

The low latency is a prime concern for large Network-on-Chips (NoCs) typically used in chip-multiprocessors (CMPs) and multiprocessor system-on-chips (MPSoCs). A significant component of overall latency is the serialization delay for applications which have long packets such as typical video stream traffic. To address the serialization latency, we propose to exploit the inherent path diversity available in a typical 2-D Mesh with our two novel router architectures, Dual-path router and Dandelion router. We observe that, in a 2-D mesh, for any source-destination pair, there are two minimal paths along the edges of the bounding box. We call it XY Dimension Order Routing (DOR) and YX DOR. There are also two non-minimal paths which are non-coinciding and out of the bounding box created by XY and YX DOR paths. Dual-path Router implements two injection and two ejection ports for parallel packet injection through two minimal paths. Packets are split into two halves and injected simultaneously into the network. Dandelion router implements four injection and ejection ports for parallel packet injection. Packets are split into smaller sub-packets and are injected simultaneously in all possible directions which typically include two minimal paths and two non-minimal paths. When all the sub-packets reach the destination, they are eventually recombined. We find that our technique significantly increases the throughput and reduces the serialization latency and hence overall latency of long packets. We explore the impact of Dual-path and Dandelion on various

packet lengths in order to prove the advantage of our routers over the baseline. We further implement different deadlock free disjoint path models for Dandelion and develop a switching mechanism between Dual-path and Dandelion based on the traffic congestion.

To My Parents

ACKNOWLEDGMENTS

I take this opportunity to thank those people who made this thesis possible.

I owe my deepest gratitude to my advisor, Dr. Seong Gwan Choi, for his support and direction over the last two years of my master's study. Without his continued encouragement and sound advice, this work would not have been possible.

I would like to thank Dr. Paul Gratz for providing us the baseline platform and invaluable guidance as a committee member.

I would also like to thank Dr. Eun Jung Kim for serving on my committee and for being an inspiring teacher.

I am indebted to many of my student colleagues for providing a stimulating and fun environment in which I had the opportunity to learn and grow. I am especially grateful to Yoonseok Yang for guiding me and working with me as a team to develop and analyse the design. I really enjoyed working with him.

A number of colleagues and friends have made my stay in College Station memorable. I would like to thank Reeshav, Dibakar, Akshay, Prasanna, Sapan, Prabal and Rohan for their constant support and affection.

Lastly, and most importantly, I wouldn't have been able to accomplish this work without the unconditional love and support of my family.

TABLE OF CONTENTS

CHAPTER Page

LIST OF FIGURES

FIGURE Page

LIST OF TABLES

CHAPTER I

INTRODUCTION

Advances in technology have nurtured growth in the parallel on-chip computing units in the form of chip-multiprocessors (CMPs) and multiprocessor systems-on-a-chip (MPSoCs). These designs typically involve high performance machines such as servers as well as multimedia portable devices such as smart-phones. CMPs involve very large number of processors along with multi-level caches and memory arrays. MPSoCs typically consists of increasingly large number of integrated components such as processors, memory arrays, application specific IPs such as baseband processors and video processing units. In these systems, the massive internal data handling among the computation elements oblige the use of scalable, high bandwidth interconnect fabric in order to provide high system-level throughput and minimal latency. Traditional bus architectures do not scale sufficiently to meet the bandwidth demands of such highly parallel systems. Network-on-chips (NoCs) [1] offer large communication bandwidth with low message transfer latency and are therefore outperforms the traditional bus based on-chip interconnect implementations. NoCs are comprised of several nodes interconnected in many possible topologies such as mesh, torus, cube etc. with regular point to point links. Each node has a router which employs a particular routing policy to route the packets from the local Processing Element (PE) as well as the incoming packets from all the directions. PE connects to the router using Network Adapter which takes the message from the PE to be sent to a distant node, breaks it into several packets, applies the destination information in the header and sends it over to the local injection port. We can compare a typical NoC architecture

---

The journal model is *IEEE Transactions on Automatic Control*.

with the road infrastructure in the city, where the roads are links, block across the square is the PE and the square is a router. When a car wants to go to a building several blocks away, it gets on to the road and wait at the square for traffic signal. The traffic signal can be thought of as a routing policy.

NoC design efforts to-date have largely been aimed at reducing the latency and relieving congestion. Peh and Dally first introduced router delay models for the pipelined routers and proposed a micro-architecture for a speculative virtual-channel router to reduce latency [2]. Since then, there have been many attempts to improve the routing policy in order to reduce the load imbalance and thus, the latency. It is important to note that a significant portion of network traffic in modern MPSoC is lengthy multimedia streaming data, e.g. the applications explored by Lee et al. contain 64-flit, video block packets [1]. Another example is the study of NoC designs for MPEG-4/H.264 parallel coding [3]. Multiple video streams are coded simultaneously in parallel while video stream data are subsequently distributed to processing elements (PEs). These studies point the need for an efficient NoC architecture for massive data streaming. One solution for such applications is to increase the link-widths. While this may reduce the total message transfer time, it comes at a high cost of increased power consumption. Alternatively, we can increase the packet length. This approach may not increase power and area, but it is severely constrained by network congestion and throughput. Network congestion can be reduced and thus, throughput can be increased by employing multipaths from source to destination.

Multipath routing has widely been explored in the general networking community. It has been recognized to yield reduced network congestion and traffic hotspots. Heuristic approaches have been extensively explored [4]. Banner et al. extend the discussion to feasibility of non-minimal paths as well [5]. Implementing these complex algorithms in an NoC design constrained by power, latency and hardware complexity

overheads, however, is not readily feasible. Daeho Seo et al. propose O1turn routing which routes packets in orthogonal direction (XY and YX) with equal probability [6]. O1turn routing achieves better path diversity than standard DOR router [2] with minimal hardware overhead. Michelogiannakis et al. introduced multi-dimension routing concept for bufferless flow control [7] in which flits can travel in any productive directions. All these approaches do not overcome the serialization latency of a long packet incurred due to single injection port.

Our research focuses on that same objectives but with observation that we can prescribe specific multiple paths and inject packets *simultaneously* through multiple I/O ports (and eject at the receiving node) to achieve parallelism. This is in contrast with the multi-dimension routing works or O1turn routing which selects one of alternative paths among available multi-paths and sending flits through the path sequentially. C. Izu et al. have studied effects of multiple injection ports on highly congested network and concluded that injecting multiple complete packets into network might not help in performance [8]. However it is assumed that injection rate will increase in the multiple of injection ports which makes the network resource too scarce to accept all the packets, and saturating the network early. We design a Network Adapter which takes a complete packet, splits it into different vestiges and send them uniformly to the queues for injection ports. This makes the injection rate in each injection port low as compared to the single injection port router and hence does not exert the unnecessary injection pressure on network throughput. To the best of our knowledge, this is the first work to explore leveraging path diversity via packet splitting and simultaneous packet injection comprehensively.

For most source destination pairs in a mesh network there can be at most 4 productive output links from the source node in the direction of the destination, two of them being minimal length paths and the rest being non-minimal path. However,

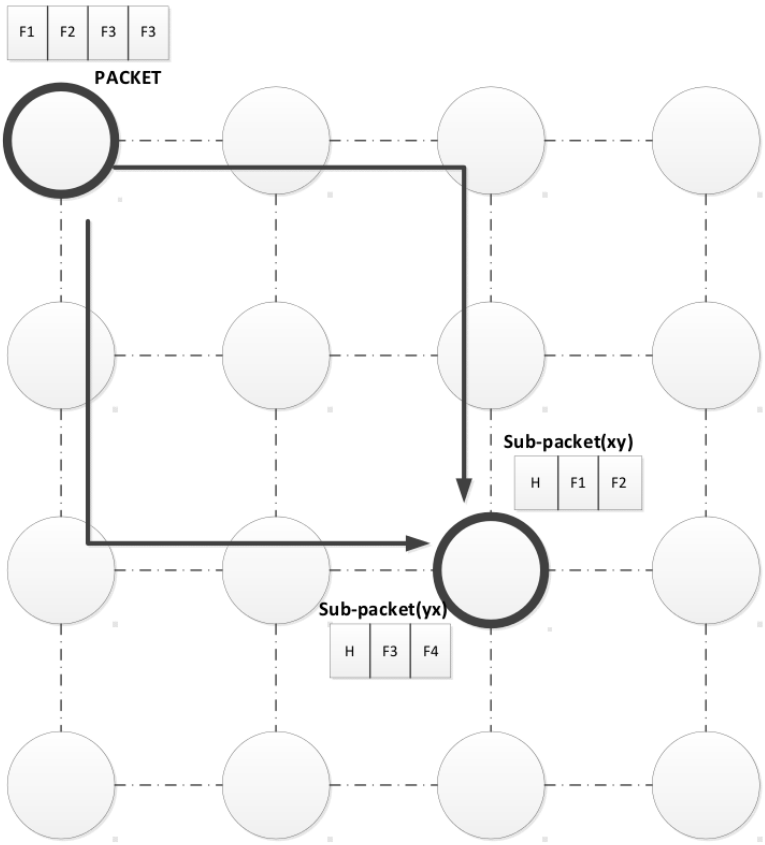

Fig. 1. Minimal Paths in Dual-path

only one of them is used at a time in traditional networks because of a single injection port. We propose two router architecture namely, Dual-path router and Dandelion router. In Dual-path router, we provide injection and ejection ports in each minimal path while in Dandelion router, injection and ejection ports are implemented in every direction. By dividing the packet into smaller sub-packets which can be injected in every direction, we significantly reduce the serialization latency and achieve higher throughput without having to increase the link-width. As per the schemes shown in the fig. 1 and fig. 2, the packet is split into two and four smaller sub-packets respectively and simultaneously injected through separate injection ports. They are

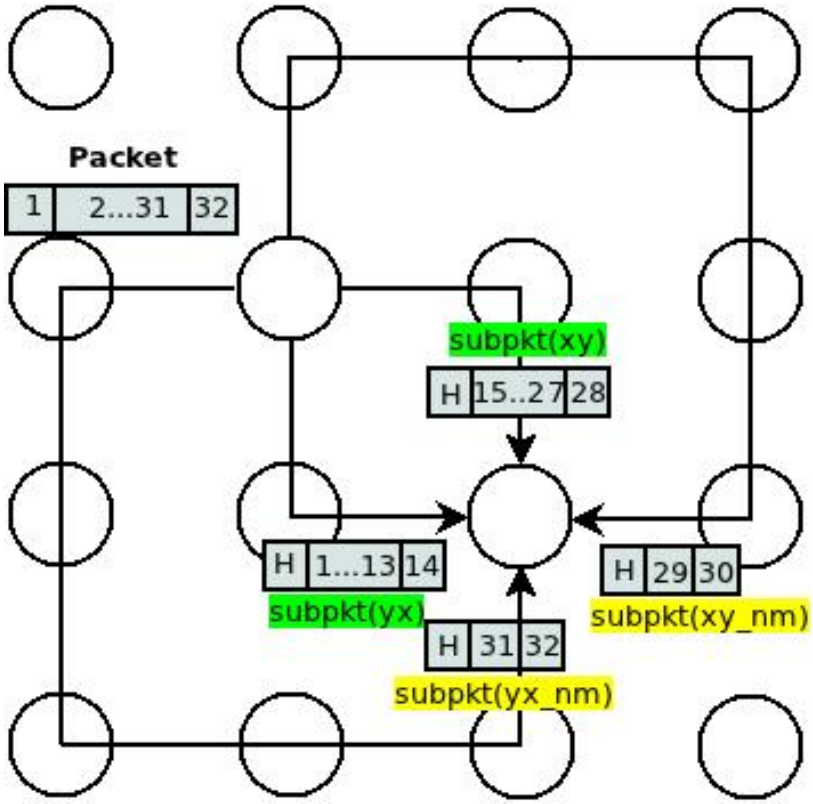

Fig. 2. Minimal and Non-minimal Paths in Dandelion

ejected at the destination in parallel as well. The reconstruction of packets can be done using the order information stored in the header flit. We will focus on DMA data transfers for this work, and therefore providing a reconstruction order number in the header is sufficient to allow for reconstruction within the DMA buffer at the destination, eliding the need for a dedicated packet reconstruction buffer. We note that these assumptions also hold for data cache line transfers, but we don't plan to explore this in our work. We note that, the Dandelion router shows improvement for long packets where traversal time through extra hops involved in non-minimal paths and the overhead due to additional headers is not significant.

Dual-path router has 2 different vestiges while Dandelion router has 4 different vestiges of the packet traveling to the destination. When the traffic congestion is high,

the difference between fastest vestige and the slowest vestige increase exponentially making the actual latency of the packet increase. This entails different latency profile in Dual-path and Dandelion for different packet length. In order to deal with this, we devise a mechanism to switch between Dual-path and Dandelion based on the flow control information such as available buffer space at the next node. This technique helps to achieve the best latency results over varying packet length and congestion levels.

Furthermore, we look into the aspect of bi-directional links on NoCs. In a typical unidirectional physical interconnect, there are two separate unidirectional channels to handle the traffic between two nodes, one for upstream and the other for downstream traffic. The link is acquired by only one node at a time to transfer the data. It is commonly observed that, one of the channel might be congested with heavy traffic while other being idle due to very less traffic in other direction. This leads to performance loss and inefficient use of resources. Throughput of unidirectional link can be increased by a) adding pipeline stages on the link, and b) increasing the link-width. Both these techniques have their own limitations. Adding pipeline stages to the link increases the interconnect latency and the power consumption. Increasing the link-width increases the power and on-chip area. Akl and Bayoumi talks about point-to-point inter-block link design by inserting accelerating repeaters and a mid-way latch to achieve bi-directional transmission [9]. Y. Lan et al. propose flexible channels which are self-reconfigurable in both directions [10]. K. Bolapalli et al. [11] explore the bidirectional point-to-point link for heavily pipelined systems. We model bi-directional bus on NoC and analyse the gain in terms of latency and saturation bandwidth. This design achieves double the throughput using the same link resource and the power. Moreover it considerably reduces the performance hit in unidirectional due to increased latency in higher congestion.

CHAPTER II

DUAL-PATH ROUTER

A. Serialization Latency and Optimal Packet Length

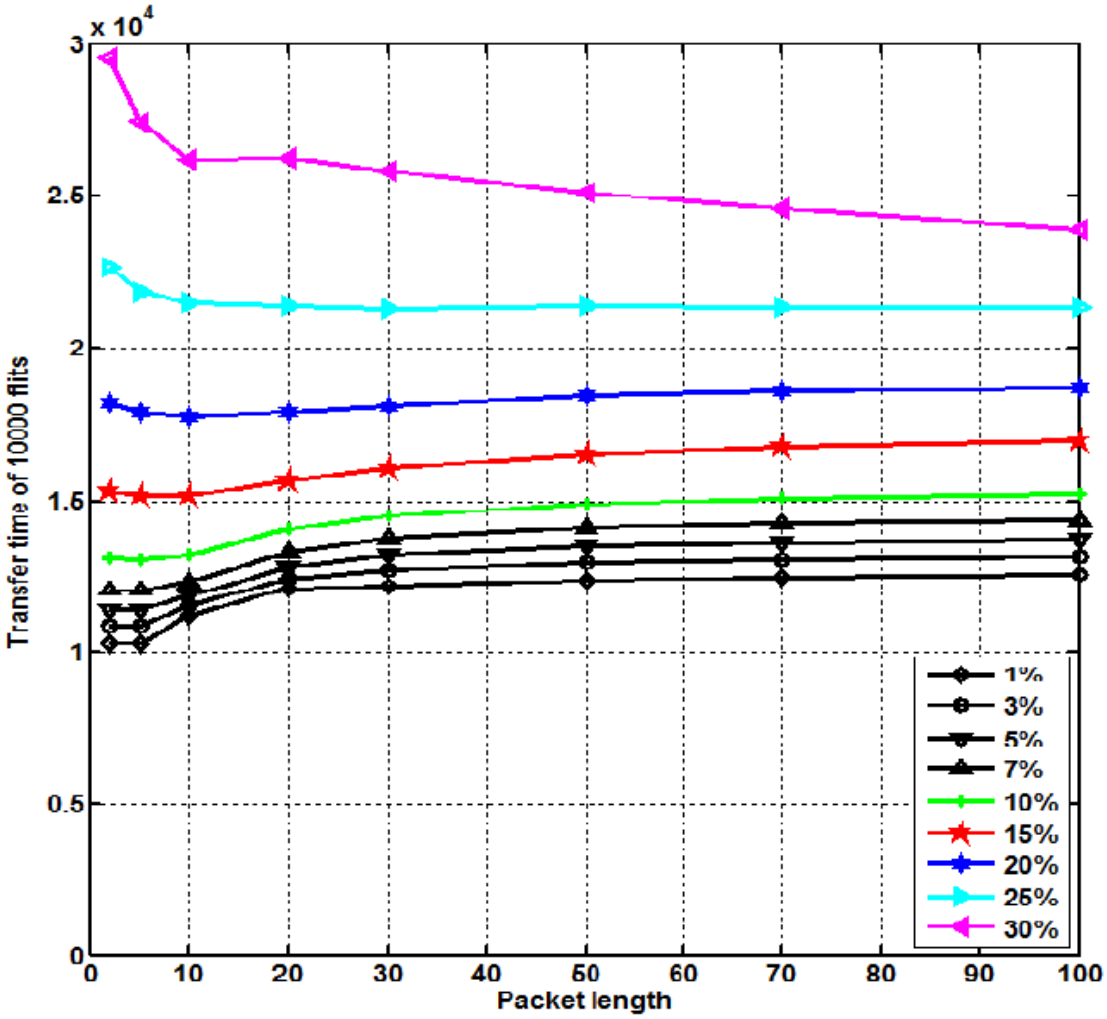

Fig. 3. Total Transfer Time for 10000 Flits for Varying Packet Length in Lower Congestion

In modern MPSoCs, multimedia streaming data constitutes significant portion of network traffic. Multimedia traffic typically consists of lengthy packets. The latency

of such long packets can be reduced either by increasing link-width or packet length. Widening link-width increases area and power, thus may not be a good choice. Thus, we plan to find the optimal length for the packet.

Fig. 3 shows the total time taken for transferring 10000 flits. Packet length is changed from 0 to 100. Experiment is performed for different congestion levels from 1% to 30%. As we see, the minimum transfer time for congestion lower than 25% is for packet length from 5 to 10.

Fig. 4 shows higher congestion results with longer packet lengths. For congestion higher than 30%, minimal transfer time is achieved by longer packets, while in lower congestions, packet lengths ranging from 5 to 10 require minimum transfer time. In a realistic traffic, congestion is generally lower and hence we can say that the packet length should be less than 10 flits for minimum latency. Serialization latency plays a major role in overall latency. Expression for latency is given by

Average Latency (in cycles) = (Ave. Hop Count X Hop Traversal Time) + Packet Length

For 7x7 NoC with 2 stage pipeline,

Ave. Hop count = 6

Hop Traversal Time = 2 (pipeline stages) + 1 (Link Latency) = 3 cycles

Thus,

Average Latency (in cycles) = 18 + Packet Length

This demonstrates that packet length which, in other words, is serialization latency does play a major role in case of realistic low congestion traffic.

For most source-destination pairs in a mesh network there are two productive output links from source node in the direction of destination (i.e. XY and YX dimension order routes), but only one can be used in traditional networks because of a single injection port. By doubling the width of the injection port and ejection port

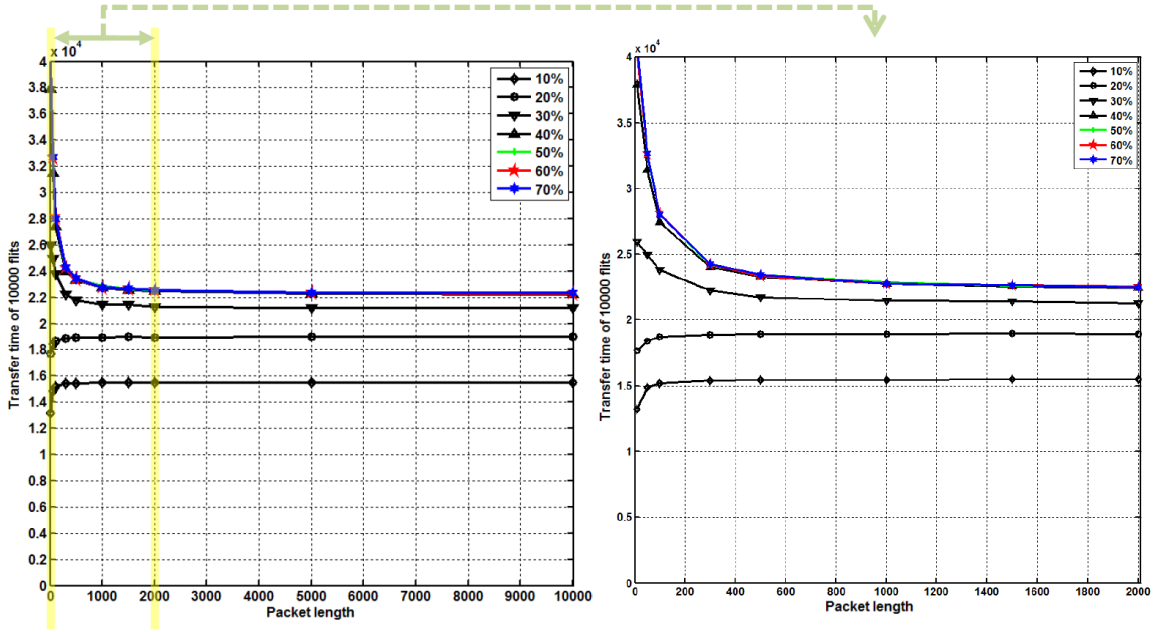

Fig. 4. Total Transfer Time for 10000 Flits for Varying Packet Length in Higher Congestion

and splitting packets into two halves we can leverage this available path diversity and cheaply approximate a higher bandwidth network. Particularly, the two split halves are simultaneously injected through the two independent injection ports, traversing XY and YX dimension order routes (DORs) in parallel. They are retrieved at the destination node through two dedicated ejection ports in parallel as well. The reconstruction of packets can be done using the order information stored in the header flit. Given this design, the Dual-path router can ideally reduce serialization latency by 50% under no loads. We note, however, that the Dual-path router provides greater benefit for large or medium size packets such as video streams because segmenting a short packet into two halves incurs overhead in header flit generation. Daeho Seo et al. reviewed the path diversity in O1turn routing [6] by injecting the packets in XY DOR and YX DOR paths with equal probability. However it is done with a single injection port which does not really reduce the serialization latency of packet

injection. The motivation behind the Dual-path router is to employ complete parallelism in packet traversal. The shortened two halves traverse two independent paths in parallel, leading to 50% reduction in serialization latency ideally under no load. The network adapter splits a message into two halves and these packets are injected to travel XY DOR and YX DOR respectively, leveraging the two available, productive, output ports toward the destination to increase the effective network bandwidth. Each Dual-path router, thus, consists of two injection ports and two ejection ports along with minimum hardware overhead. These packets approach the destination from different directions and get absorbed by different ejectors in parallel. If there exists only one minimal path between source and destination like traversing only on X or Y axis, the packet is sent intact without using Dual-path routing. DOR is inherently deadlock free. In Dual-path routing, separate channels are allocated for XY and YX DOR paths to prevent deadlock. VCs are provided in these separate channels to avoid the head-of-line blocking in XY and YX DOR paths. The baseline and Dual-path router details are described in this chapter.

B. Baseline Router

The baseline is a standard 2-D mesh NoC with a pipelined wormhole router [2]. Wormhole routing carries a message along network by dividing packets into fixed sized flow control digits or flits. The router contains different logic blocks such as input unit for each port, VC allocator, Switch allocator, Crossbar and output unit for each port. Input unit has buffers for every virtual channel. The header of every packet has the information of virtual channel ID (VCID), source-destination coordinates, Current Node Output Port (CNOP). Input unit stores the flit of a particular packet into the VC denoted by the flit itself, while it requests the arbitration in round robin

fashion. Once a packet gets the VC for next node, input unit updates the header with new VCID and CNOP and requests for Switch Allocator. Switch allocator receives requests from every port and process them on flit by flit basis. The flits granted by switch allocator, then go to crossbar where they are sent to the next node. Output unit at every port keeps track of the credits left at the downstream node which are then used by the VC allocator to allocate the available VC at the next node.

The pipeline of the baseline router consists of 2 stages: route computation and arbitration. The router traversal takes at least 2 clock cycles in ideal case. It may increase due to waiting for the arbitration. The physical link latency between the nodes is assumed to be one cycle.

C. Network Adapter

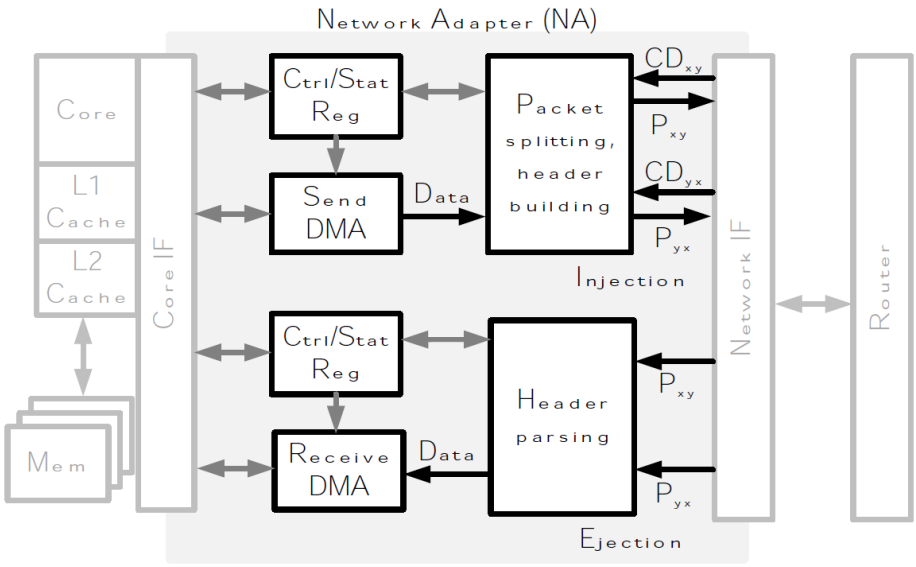

Fig. 5. Network Adapter for Dual-path Router

Network adapter (NA) is an interface between the Processing Element (PE) and router. It divides messages into network compatible packets. Every packet is made

of flits whose width is generally equal to the physical link. Packet has a header flit which contains all the routing information followed by the payload and the tail flit which is the end of packet. NA takes in the credit information and injects the packet flit-by-flit through separate injection ports into the separate paths as shown in fig. 5. When the DMA has a message (i.e. packet) to send, it is forwarded from the cache to NA for packet splitting which is then re-packetized with modified header and tail flits. An initial message with n flits is divided into two packets of *n/2 + 1(header)* flits each. New header flits for the half packets (P_{xy} and P_{yx}) are built and injected to the Dual-path router through network interface. At the destination, two ejected packets (P_{xy} and P_{yx}) are delivered to NA for header parsing and transferred by DMA to cache memory. When DMA finishes sending two ejected packets to the cache, PE reconstructs the split packets.

D. Dual-path Router

1. Micro-architecture

Unlike the baseline router, Dual-path router has an extra local injection and ejection port to make the Dual-path routing beneficial. In Fig. 6, the Dual-path router consists of four directional input/output ports, two local injection ports, credit signals, and two ejection ports. Each directional input port contains two channels, one for XY DOR and the other for YX DOR, supporting two completely disjoint paths from source to destination. The XY or YX path in a packet is determined by 1-bit DOR information embedded in the packet. If there are multiple virtual channels (VCs) for each path, the VCID indicates which virtual channel is used by this packet. In the figure, the Dual-path router uses only one virtual channel, hence the VCID should be 0. In case of using two VCs for each path, the VCID can be 0 or 1 to denote

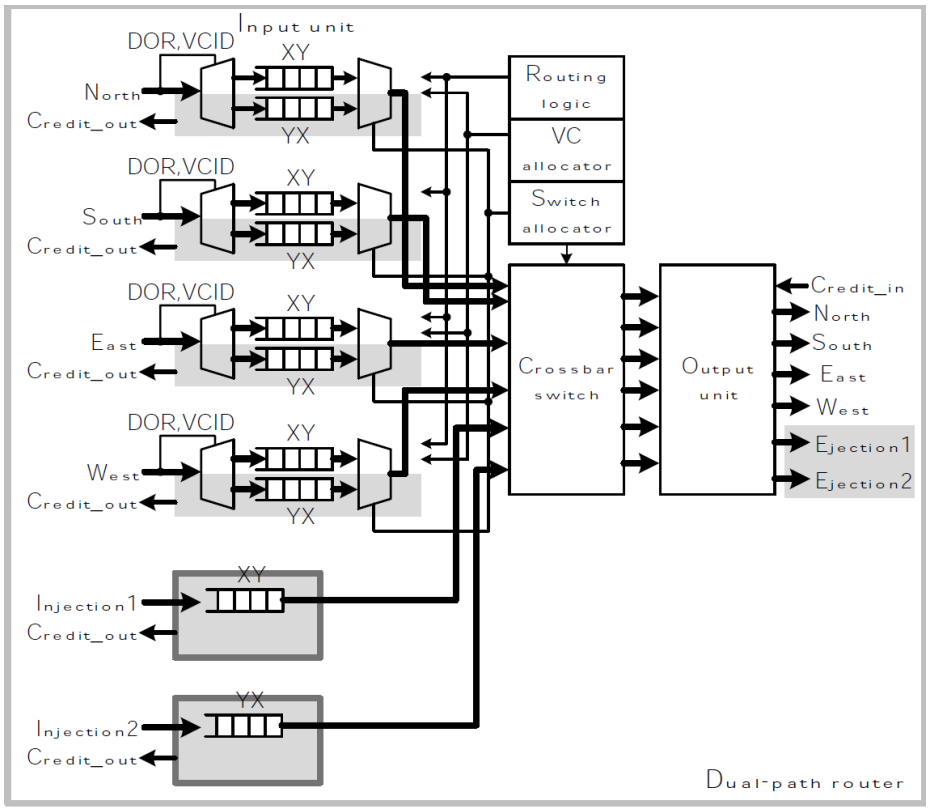

Fig. 6. Dual-path Router Micro-architecture

which one is occupied. The credit based flow-control system is used for buffer management. As highlighted in grey on Fig. 6, the hardware overhead of the Dual-path router is minimal as the only additions are an extra injection, ejection port and hence wider crossbar with slightly modified control logic. No additional buffers are required compared with the baseline router, as same number of total VCs per port are used.

2. Packet Structure

Fig. 7 shows the packet format. The size of a flit is 64 bits. 2 MSBs indicate the flit type. The DOR bit determines XY DOR or YX DOR, and the VCID bit is the same as that in baseline. We note that 1-bit VCID was used to provision maximum 2 VCs per port in our experiment, but the number of VCs can be extensible. CNOP (3

bits) indicates the output port for the packet at the current node and is required to request the output port in our design. Once the output port is acquired, the CNOP is updated for the next node.

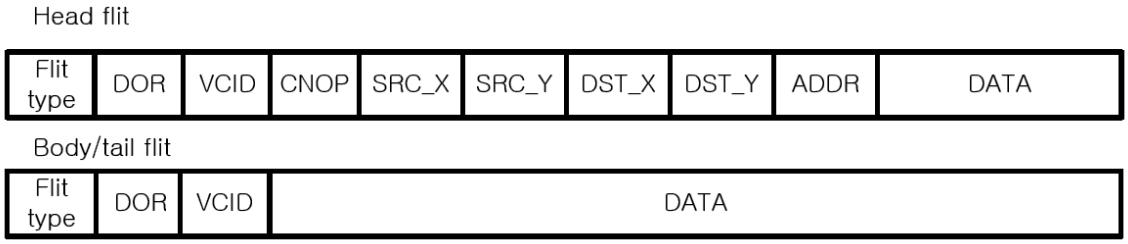

Fig. 7. Packet Structure for Dual-path Router

Table I. Description of Fields in the Flit for Dual-path

Flit type	DOR	VCID	CNOP	SRC	DST
00 : head 01 : body 10 : tail 11 : single	0: XY 1: YX	0: 1 VC 0,1: 2 VCs	000: local1 001: north 010: south 011: east 100: west 101: local2	Coord. for src.	Coord. for dst.

The source and destination addresses are used to calculate the CNOP for the next node (i.e. lookahead routing). 10-bit ADDR followed by payload denotes the order information of split packets for reconstruction.

For the body and tail flits, it only has the flit type, DOR, and VCID fields, and the rest is payload. Table I explains the information stored in the flit and their meaning.

3. Packet Processing at Source and Destination

Each source node decides to use Dual-path routing based on the destination address of packets. If the destination is "inline", there are no two paths to the destination

and hence packet is sent intact without splitting. When there are two minimal paths to the destination, the network adapter splits the packet into two halves and assigns the header information to both packets at the source.

The re-packetized packets are injected simultaneously through dual injection ports. When the split packets reach the destination, they exit through the dedicated ejection ports. Packet splitting incur overhead of extra header and tail. Thus, for short packets (2-5 flits), Dual-path cannot give much benefit. However we target video stream traffic where the packet lengths are longer and serialization latency is dominant. Splitting the long packets and thus injecting them simultaneously provide double network bandwidth and cause at most 50% reduction in serialization latency with a negligible overhead.

We have observed that the reconstruction of the separated halves could be done using small amount of cache memory in the experiment for realistic video benchmarks. This is because the network load of realistic video traffic was low, leading to 2-3 cycles difference of ejection time of two split packets on average at the destination. Furthermore, we note that in many applications such as cache line or DMA data transfers, memory has already been set aside for the reception of the data, thus even for the worst case ejection time difference, a reconstruction buffer is unnecessary. Therefore, in cache line transfers or DMA data transfers, the reconstruction order number (i.e. ADDR field in the header) can sufficiently support to reconstruct the separate packets within the cache or DMA buffer at the destination.

E. Experiments and Evaluation

In this section, we evaluate the Dual-path router experimentally to analyze the performance under different types of synthetic and realistic workloads. We compare the

performance of Dual-path router against that of baseline single-path (SP) router.

1. Methodology

In these experiments, we evaluated a fully synthesizable network consisting of Dual-path routers connected in the 7x7 2-D mesh topology, to obtain the performance numbers. In the SP router, we used 2 VCs and 4 VCs (SP-VC2 and SP-VC4) each with buffer depth of 4. For a reasonable comparison, we compared the SP routers with the Dual-path routers using 1 VC and 2 VCs per path (DP-VC1 and DP-VC2) since SP-VC2 and SP-VC4 contain the same FIFO buffering and resources as DP-VC1 and DP-VC2, respectively.

The performance analysis of the proposed router uses uniform random, transpose and bit-complement synthetic workloads and H.264 video benchmarks for realistic workloads. In the synthetic workload simulation, packet length was varied randomly between two to 100 flits. For the evaluation on the H.264 video streams, we simulated four different QCIF (176x144) video sources (foreman, akiyo, mother, and mobile each 150 frames) generally used for a video test. The packet size of video streams was 11 on average where the video streams suitably provided a realistic benchmark for evaluating our design on small to medium packet sizes in low network loads. The latency of the packet is calculated as time difference between injection of the first half packet at source and ejection of the last half packet at destination.

2. Results and Discussion

Fig. 8 shows the results for synthetic traffic loads and H.264 video benchmarks. The comparison results shown in Fig. 8 depict the average packet latency versus traffic loads on a 7x7 NoC across the synthetic traffic patterns and video test streams. The results show DP-VC2 shows a significant improvement over the single-path designs

for all traffic loads due to its better utilization of path diversity.

Under transpose traffic (Fig. 8(b)), the latency and saturation throughput results of the Dual-path router are much better than that of the SP router because the transpose pattern exploits the diagonal network bisection which increases the probability of packet splitting and parallel traversal through the Dual-path routes. On the other hand, the bit-complement traffic mainly uses horizontal and vertical network bisections and provides less chance of parallel traversal than the transpose traffic, leading to the less performance improvement than the transpose and random uniform loads. However, it still achieves up to 28% latency reduction in low traffic rates. Fig. 8(a) and (c) show that the saturation throughput of DP-VC1 is less than SP-VC2. This is because DP-VC1 contains only one channel for each DOR path, leading to head-of-line blocking which the SP-VC2 design does not experience. DP-VC1 nevertheless achieve significantly lower no-load latencies than the baseline SP-VC2.

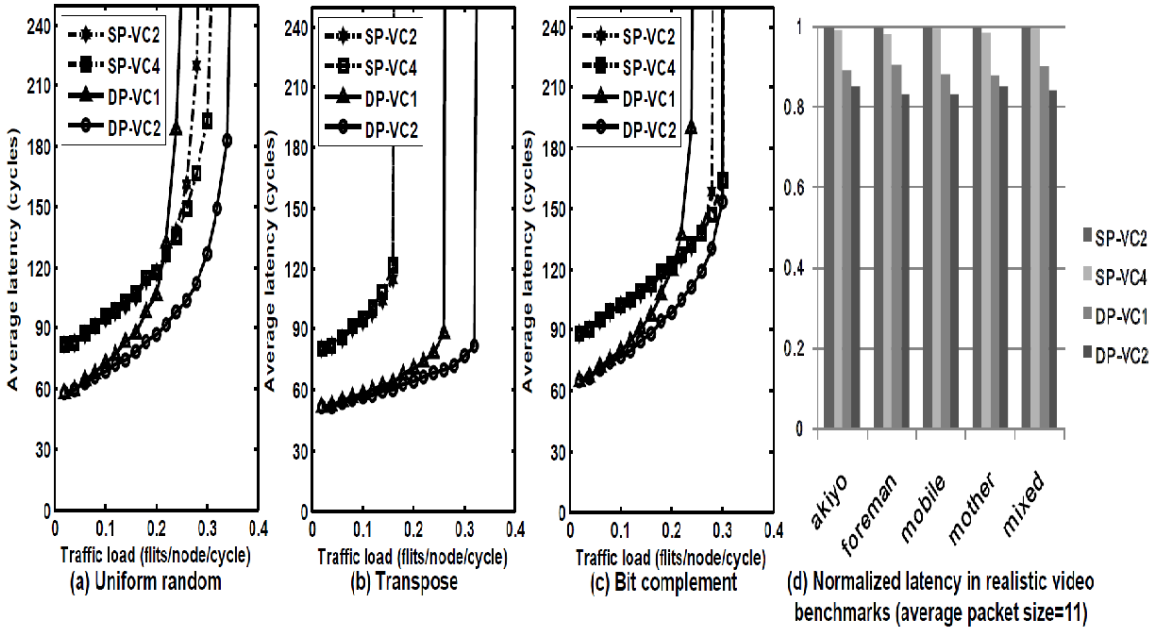

Fig. 8. Synthetic and Realistic Benchmarks for Dual-path Router

In the result of video benchmarks as shown in Fig. 8(d), the latency gain of DP-VC2 is 17% on average, compared with SP-VC4. We find that the performance gain in the video test is lower than the gain of the synthetic traffic test since the video packets vary in packet size from small to medium (11 on average). Therefore the benefit of packet splitting in the video workloads is less than that of the synthetic workloads with an average packet size of 50. Generally, the Dual-path routing scheme is most effective, when streaming packets of medium to long sizes.

Table II profiles and compares the synthesis results of fully synchronous single-port router and dual-port router. We synthesized those two routers using Synopsys Design Compiler on TSMC 45nm technology with default switching activities. Dual-path denotes higher power/area overhead than SP due to the additional path and its buffer.

Table II. Synthesis Results of SP-4VC and DP-2VC with 4-depth FIFOs at 1GHz (IN:input unit with FIFOs, VA:VC allocator, SA: switch allocator, CS: cross-bar switch, OUT: output unit, TO: router total)

	Router	IN	VA	SA	CS	OUT	TO
Area (μm^2)	SP-VC4	57869	20166	2287	4242	1022	86842
	DP-VC2	59568	24193	3143	6226	1220	91156
Power (mW)	SP-VC4	25.40	4.01	1.03	1.98	0.51	29.75
	DP-VC2	25.71	4.66	1.23	2.31	0.59	30.89

CHAPTER III

NOC MODELING FOR BI-DIRECTIONAL LINK

With increasing scale of Network-on-Chip (NoC), a key challenge is to achieve the higher bandwidth and low latency. In a typical uni-directional physical interconnect, there are two separate uni-directional channels to handle the traffic between two nodes, one for upstream and the other for downstream traffic. The link is acquired by only one node at a time to transfer the data. It is commonly observed that, one of the channel might be congested with heavy traffic while other being idle due to very less traffic in other direction. This leads to performance loss and inefficient use of resources. Throughput of uni-directional link can be increased by,

a) Adding pipeline stages on the link, and

b) Increasing the link-width

Both these techniques have their own limitations. Adding pipeline stages to the link increases the interconnect latency and the power consumption. Increasing the link-width increases the power and on-chip area. Akl and Bayoumi talks about point to point inter-block link design by inserting accelerating repeaters and a midway latch to achieve bi-directional transmission [9]. Y. Lan et al. propose flexible channels which are self-reconfigurable in both directions [10]. K. Bolapalli et al. explore the bi-directional point to point link for heavily pipelined systems.

A. Bi-directional Links

We propose a novel low power bi-directional interconnect design which eliminates the key limitations of uni-directional bus. Bi-directional interconnect allows the traffic from both the ends at the same time by employing a meta-state detector (MSD) circuit to detect if the two ends of the wire are being driven by opposite logic levels. As

compared to uni-directional bus, this design achieves double the throughput using the same link resource and the power. Moreover it considerably reduces the performance hit in uni-directional due to increased latency in higher congestion. We only focus on evaluation of the technique instead of going into design details of bi-directional links. To evaluate the performance, we have designed a 7x7 fully synchronous NoC in verilog, modeling the bi-directional bus operation. Extensive analysis with synthetic and realistic SPLASH-2 benchmarks demonstrates that bi-directional NoC performs exceedingly better in terms of latency than the uni-directional NoC at higher traffic rates. This is mainly due to delayed saturation of bi-directional NoC. The basic advantages of the design are :

- Double throughput between the nodes using same link resources.
- Better latency results as compared to baseline interconnect at higher congestion.
- Moderate logic overhead in terms of complexity and power.

B. Evaluation

Baseline is a standard 2-D mesh, pipelined router with virtual channels. The interconnect in the baseline is uni-directional. There are two independent links of same width (64-bits) between 2 nodes for downstream and upstream traffic. Each link is non-pipelined and has a latency of one clock cycle. In bi-directional bus NoC, each link supports bi-directional traffic and hence there is no need to have 2 separate channels for upstream and downstream traffic. Bi-directional channel with same link resources as baseline can be modeled as twice the channel width (128 bits). Thus, in order to transfer the same amount of information across the network, number of flits in a bi-directional packet is half as that of baseline packet. The delay incurred

due to bi-directional logic is modeled as 3 pipeline stages across the link though it might be further decreased by optimizing the timing for bi-directional link logic. The extra pipeline stages on the link add a steady value to the latency which render the bi-directional interconnect NoC less useful than the baseline at lower injection rates. However at higher injection rates, the latency incurred by bi-directional is lower than the latency in the baseline. This is because bi-directional NoC has almost double the bandwidth and does not saturate at the injection when baseline does and hence has lesser congestion as compared to baseline.

Fig. 9 shows the comparison results that depicts average packet latency versus traffic load on a 7x7 NoC across synthetic traffic which is generated by uniform random injection process. Packet length is varied between 4-10 flits randomly for baseline while it is 2-5 flits for bi-directional network. The simulator was run for 100,000 cycles including 1000 warm up cycles. Saturation bandwidth for baseline and bi-directional is measured as a point at which the average packet latency is 3.4 times the zero load latency and 6.5 times the zero load latency respectively. As expected, bi-directional latency is more than the baseline for low injection rates. This is mainly because of the additional pipeline stages required by the bi-directional bus which adds a fixed value to the link latency. As we reach the saturation bandwidth of baseline, the latency incurred in bi-directional bus is lower than that of baseline. As the flit width in bi-directional has doubled, the number of flits in a packet to transfer the message from source to destination is halved. Thus, the injection level and the latency in bi-directional is lower than the baseline. Saturation bandwidth of bi-directional NoC is 90% higher than the baseline.

Fig. 9 also shows the average packet latency results across different combinations of SPLASH-2 benchmarks. We use 500000 cycles with 1000 warm-up cycles in realistic simulation. The standalone SPLASH-2 benchmarks inherently have lesser congestion

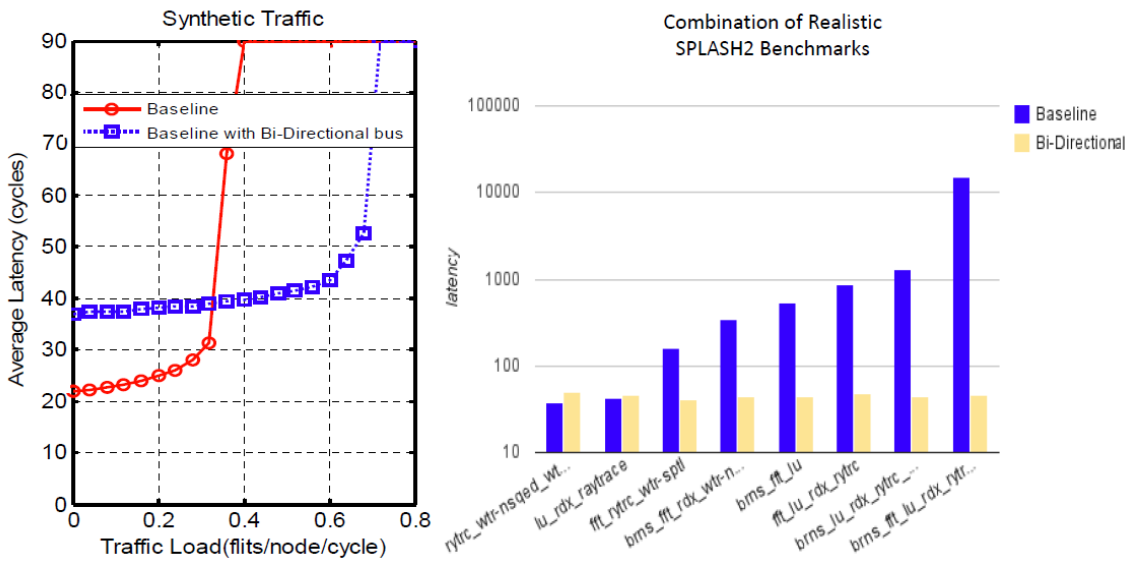

Fig. 9. Synthetic and Realistic Benchmarks for Baseline Router with Bi-directional Links

and cannot highlight the merit of bi-directional bus over the uni-directional bus. Hence we have run multiple SPLASH-2 benchmarks simultaneously in order to simulate the higher congestion in realistic traffic. As we see, in some of the combinations which have injection rates lower than the saturation bandwidth of baseline network, the bi-directional NoC does not do better. However in all the combinations where the injection rates are higher than the saturation level of baseline, the bi-directional NoC does exceedingly better than baseline results.

CHAPTER IV

DANDELION ROUTER

The motivation behind the Dandelion router is to exploit the maximum possible parallelism in packet traversal. If parts of a packet is sent in all the directions, we believe that the serialization latency of the whole packet will be reduced. As every router has 4 directional output ports, we implement four injection ports at the source and four ejection ports at destination. The packet gets injected in all the direction at the same time, ideally reducing the serialization latency by 75% under no load. The splitting and processing of the packets is done by the network adapter and the smaller sub-packets are injected simultaneously in all the directions. These sub-packets reach destination from different directions and are absorbed by dedicated ejection ports to be recombined at the processing element. Number of directions might be different depending on the source destination orientation. In such cases, packet is split into the sub-packets equal to the number of available paths. In order to get a clear picture of non-minimal paths we present the following section.

A. Non-minimal Paths

In a typical 2-D mesh topology, source-destination pair has non-minimal paths apart from the minimal paths along the bounding box created by the pair. As show in fig. 10, there can be different kinds of non-minimal paths based on the number of extra hops incurred and the number of turns. If the destination is 'in-line' with the source, we have just one minimal path and two non-minimal path with two extra hops with two turns and eight extra hops with four turns. In general, there are two minimal paths and two non-minimal paths with 4 extra hops with three turns as discussed earlier in chapter I. Non-minimal paths with different number of extra

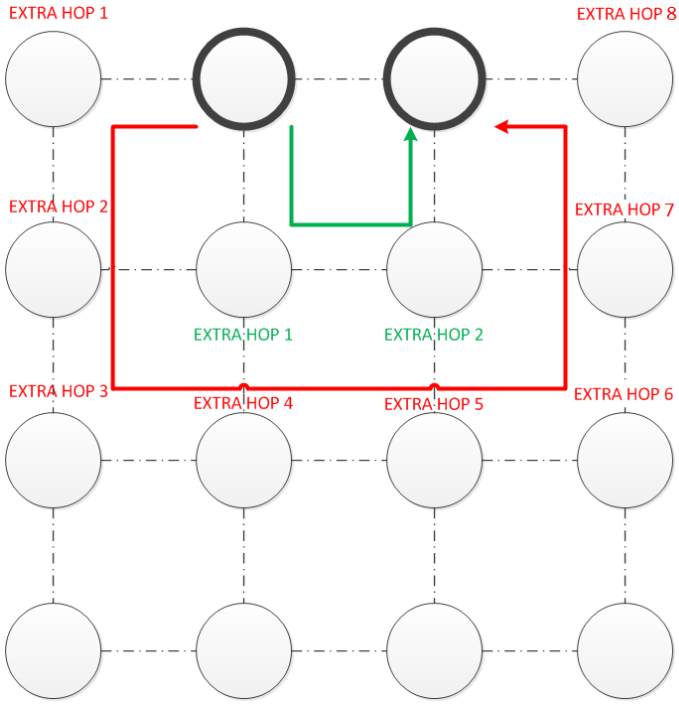

Fig. 10. Non-minimal Paths in 2-D Mesh Topology

hops require different routing logic. Non-minimal paths with 4 extra hops are more frequent and as we increase the size of NoC, the percentage of non-minimal paths with 4 extra hops increases. Thus we only consider such type of non-minimal paths for Dandelion router in order to keep our routing logic as simple as possible.

B. Network Adapter

Network adapter (NA) is essential for packet splitting and processing. As shown in the fig. 11, it splits messages and injects them through separate injection ports into the available separate paths. When the DMA has a message (i.e. packet) to send, it is forwarded from the cache to the packet processing block, and then split into sub-packets, and re-packetized with modified header and tail flits. We consider the only non-minimal sub-packets that travel 4 extra hops. For an ideal router, flit traversal

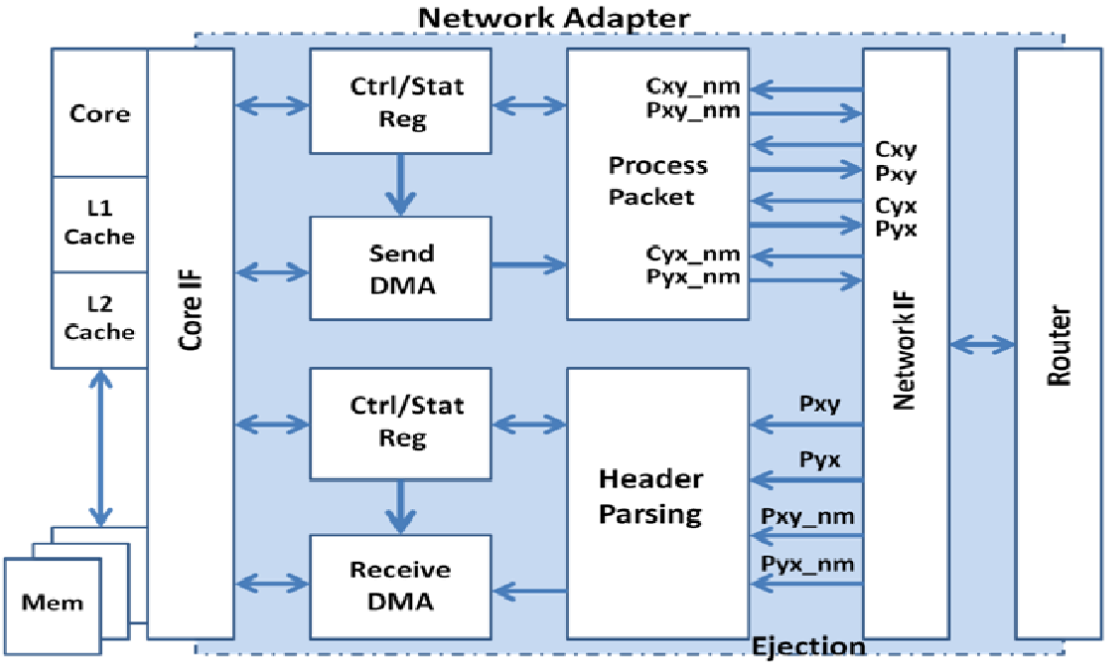

Fig. 11. Network Adapter for Dandelion Router

through a router takes at least 2 cycles and 1 cycle as link delay. Thus, for a non-minimal path, the sub-packet travels at least 12 extra cycles. Therefore, in order to mitigate the extra hop latency, an initial message with n flits is divided into four with sub-packets. Sub-packet on the minimal paths typically have length of *12 + (n-24)/4 +1(header)* flits and the sub-packets for non-minimal paths have the length of *(n-24)/4 + 1(header)* flits each. Here we note that for a source-destination pair with 2 non-minimal paths, the packet length has to be at least 24 flits to use all the injection ports. In the cases where 2 minimal or non-minimal paths are not available, the sub-packet lengths are determined accordingly. As shown in fig. 11, new header flits for the sub-packets (P_{xy}, P_{yx}, P_{xy_nm} and P_{yx_nm}) are built and injected to the Dandelion router through network interface. At the destination, ejected sub-packets are delivered to NA for header parsing and DMA transfers them to cache memory.

When DMA finishes sending ejected sub-packets to the cache, PE reconstructs the split packets.

C. Baseline Router

The baseline is a standard 2-D mesh 10x10 NoC, with a pipelined router with 8 virtual channels (VC) [2]. The pipeline consists of 2 stages: route computation and arbitration at output port. When a flit enters a router, it is sent to the particular VC FIFO depending on the VC identification (VCID) carried by the flit. If the flit is a header carrying the current node output port (CNOP) information, arbitration is requested for acquiring the VC at next node as conventional wormhole routers.

We also compare Dandelion router with Dual-path router. Dual-path router is laid out in 10x10 2-D mesh NoC. Dual-path router has two injection and ejection ports. There are total 8 VCs in Dual-path router, with each minimal path having 4 VCs.

D. Dandelion Router

1. Micro-architecture

Unlike the baseline router, the Dandelion router has 4 injection and ejection ports. As shown in fig. 12, Dandelion router is comprised of four directional ports, four injection ports, arbitration logic and four ejection ports. Every directional port has eight VCs. The distribution of VCs is selectively done among minimal and non-minimal paths to avoid the cyclic dependency. We have developed and analysed two different schemes of VC distribution, namely 4 VC classes model and 6 VC classes model. Fig. 12 shows 4 VC classes model. We explain both the models in details in the following subsection. Every injection port has 2 VCs. As we see in fig. 12, 2

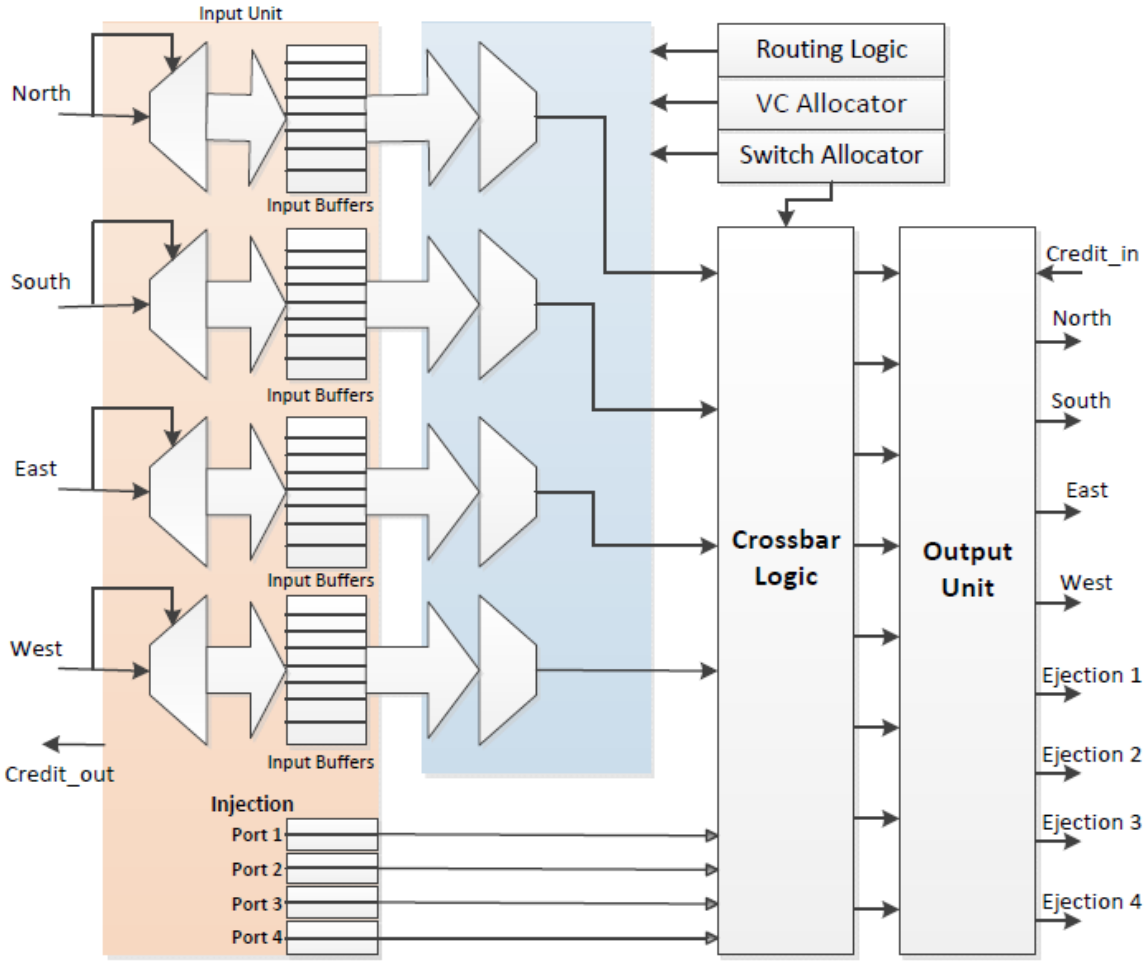

Fig. 12. Dandelion Router Micro-architecture

bit information is used to decode the XY/YX route and minimal/non-minimal paths. VCID bits are used to decide the virtual channel for a particular path. The credit based flow control is used for buffer management. Available VCs for downstream node are maintained at every output port which is used by VC allocator to arbitrate for available channel. Credits are also maintained at the output port which denote the available buffer space at subsequent node. Credit information is used by switch allocator to send the contending flits. Crossbar facilitates the parallel traversal of flits across the router. The hardware overhead of the Dandelion as opposed to baseline

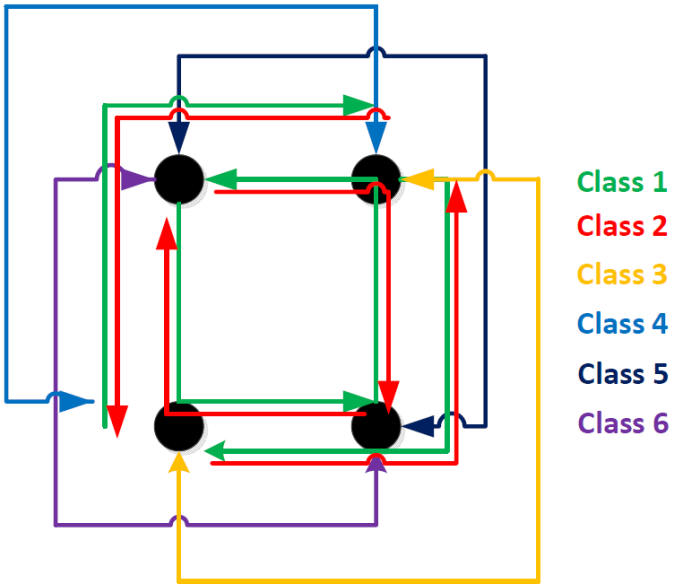

Fig. 13. 6 VC Classes Model for Dandelion

router is extra injection and ejection ports, logic overhead for non-minimal routing, increased size of crossbar due to additional ports.

2. VC Classes

In Dandelion router, the deadlock free paths can be created by distributing VCs in two different ways. We have designed two separate schemes, namely 4 VC classes model and 6 VC classes model, and analyzed their efficiency. We divide 8 available VCs in 4 and 6 classes respectively. Each class of VCs is deadlock free. Packets are assigned a particular class depending on the source destination orientation and are not allowed to change it's class till it ejects in the destination. In 6 VC classes model, minimal and non-minimal paths are kept separate as shown in fig. 13. The XY and YX DOR minimal paths are inherently deadlock free and constitute 2 classes with 2 VCs each. Non-minimal paths have 3 turns and hence they can create a deadlock if VCs are shared among them. In order to cancel the dependency cycles, we allow

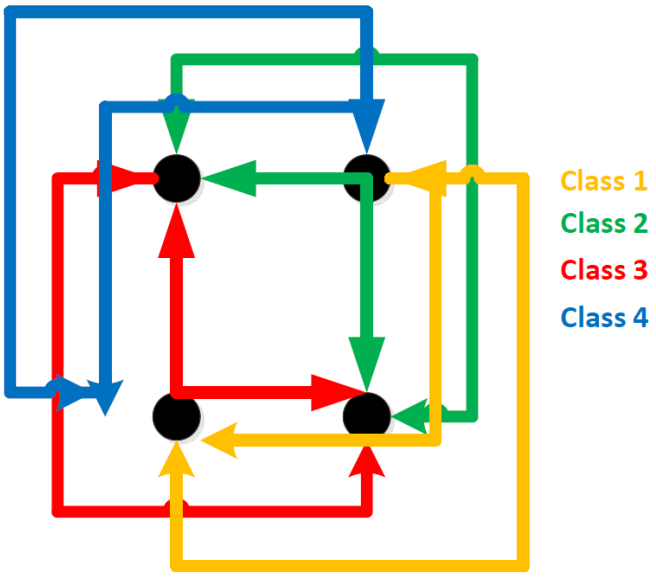

Fig. 14. 4 VC Classes Model for Dandelion

only 3 turns in every channel. There are 4 different orientations of XY non-minimal or YX non-minimal paths. All of the XY non-minimal paths cannot be grouped together as they involve all four turns and can cause deadlock. Thus, every channel has only 2 XY or YX non-minimal paths instead of all 4 paths of different orientations. We note that, in this model, all the injections are done in completely disjoint paths and serialization latency gain due to splitting is maximally exploited. However as there is only one VC available for XY non-minimal and YX non-minimal each for a particular source-destination pair, head-of-line blocking for these non-minimal paths is not eliminated and moreover, at least 2 channels are always unused for a particular source destination pair.

In 4 VC classes model, minimal and non-minimal paths are not disjoint as shown in fig. 14. We borrow concept from the traditional turn model here and divide the 8 available VCs uniformly in 4 different classes. Each class has one turn prohibited in order to break the cyclic dependency. We note that, all the VCs are always used

for a particular source and destination, eliminating the head-of-line blocking effect. However, since minimal and non-minimal paths share the same VCs, the serialization latency gain might be compromised.

3. Packet Structure

The fig. 15 shows the packet flit format. The size of a flit is 64 bits. Table III illustrates the meaning of various fields in the flit. 2 MSBs indicate the flit type. The MIN bit determines whether it is an minimal or a non-minimal path. The DIR bit denotes XY or YX route, and the VCID bits are the same as that in baseline and can be scaled depending on total VCs. CNOP (3 bits) indicates the output port for the packet at the current node and is required to request the output port in our design. Once the output port is acquired, the CNOP is updated for the next node. The source and destination addresses are used to calculate the CNOP for the next node (i.e. lookahead routing). Source and destination coordinates are 4 bits each to support 10x10 NoC. 10-bit ADDR followed by payload denotes the order information of split packets which is used to reconstruct the packet at the destination. The body and tail flits only have the flit type, MIN, DIR, and VCID fields, and the rest is payload.

Table III. Description of Fields in the Flit for Dandelion

Flit Type	***MIN***	***DIR***	***VCID***	***CNOP***	***SRC***	***DST***
00:Head 01:Body 10:Tail 11:single	0:min 1:non -min	0:XY 1:YX	0/1:VC	000:local1 001:north 010:south 011:east 100:west 101:local2 110:local3 111:local4	Coord for SRC	Coord for DST

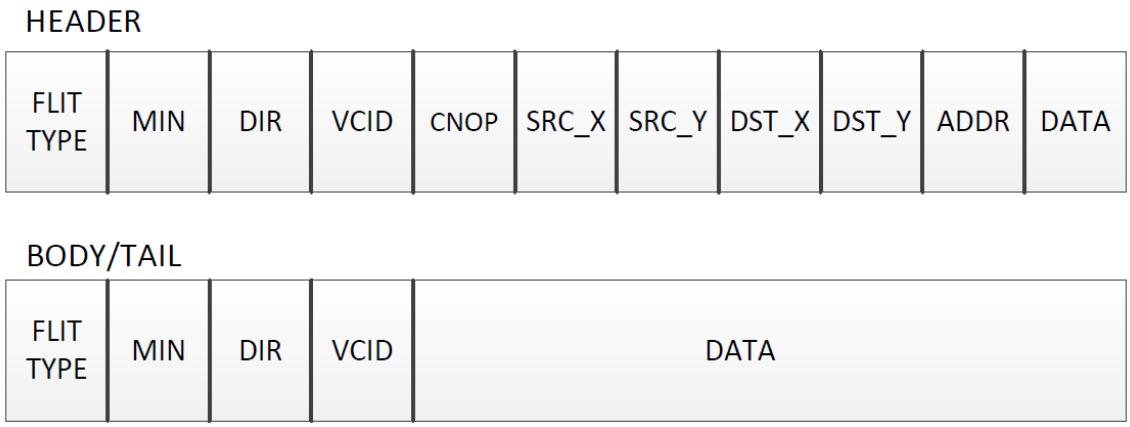

Fig. 15. Packet Format for Dandelion

4. Packet Processing

The packets are split and processed depending on the number of directional paths available from source to destination. If the destination is 'inline', there exists only one minimal path. In all other cases, there are two minimal paths between source and destination. Number of non-minimal paths also depend on the orientation of source-destination pair. We keep the routing policy simple by only considering the minimal paths and non-minimal paths with 4 extra hops and ignore the non- minimal paths with 2 or 8 extra hops. In the 2-D mesh network, the non-minimal paths incurs extra latency due to additional hops. So it's necessary to split the packets considering the extra time the non-minimal paths will take. It takes 2 cycles to traverse through a router and one cycle in a link traversal. Hence for 4 extra hops, it will take at least 12 cycles more for a packet to reach the destination under no load. Thus, the sub-packets in the non-minimal paths have the length shorter by 12 than those in the minimal paths so that the extra latency involved due to more number of hops can be mitigated. All the sub-packets are assigned with the header at the source and are injected simultaneously through 4 dedicated injection ports. When the sub-packets

reach the destination, they exit through the dedicated ejection ports.

The latency of the whole packet is calculated as the time difference between the injection and latest ejection of the slowest sub-packet. At the destination, the time difference between ejection time of fastest sub-packet and slowest sub-packet increases rapidly with traffic, potentially requiring substantial buffering for recombination. However, we note that, for a standard video stream traffic, the applications such as cache line transfers or DMA data transfers where memory has already been set aside for reception of the data, the reconstruction order number (i.e. ADDR field in the header) can sufficiently support to reconstruct the separate sub-packets within the cache or DMA buffer at the destination.

E. Experiments and Evaluation

In this section, we evaluate the Dandelion router experimentally to analyze the performance under different types of synthetic and realistic workloads. We compare the performance of Dandelion router against that of baseline single-path (SP) router and Dual-path router.

1. Methodology

In these experiments, we evaluated a fully synthesizable network consisting of Dandelion routers connected in the 10x10 2-D mesh topology, to obtain the performance numbers. We compare our results with single injection port DOR router as well as Dual-path router. For a reasonable comparison, we maintain 8 channels per port with a depth of 4 for every router. The performance analysis of the proposed router uses uniform random and transpose synthetic workloads and medium data-rate H.264 video stream. The latency of the packet is calculated as the time difference between

the injection of the sub-packets and the ejection of the last sub-packet at the destination.

2. Results and Discussion

Fig. 16 shows the results for synthetic and realistic traffic loads. The comparison results shown in fig. 16 depict the average packet latency on a 10x10 NoC. For synthetic benchmarks, packet length is kept fixed at 50 flits in order to have the specific insights into long packets. H.264 video sources with medium bit-rate is used for realistic benchmarks which have average packet length as 52 flits. The results show that Dandelion router exhibits improvement over baseline and Dual-path designs for uniform random traffic. Dandelion cannot perform better than other routers in higher congestion in case of skewed traffic patterns such as synthetic transpose and bit complement benchmarks. For the realistic traffic, Dandelion does better for lower congestion prone traffic patterns such as mobile, mother and all while it cannot outperform Dual-path as well as baseline for higher congestion traffic patterns such as akiyo and foreman.

Power analysis and comparison is done for Dandelion in table IV. Synthesis is done on TSMC 45nm technology with default activity using Design Compiler tool. The power and area overhead is mainly due to extra injection and ejection ports, wider crossbar and improved routing and control logic. Number of input buffers and VCs are kept constant.

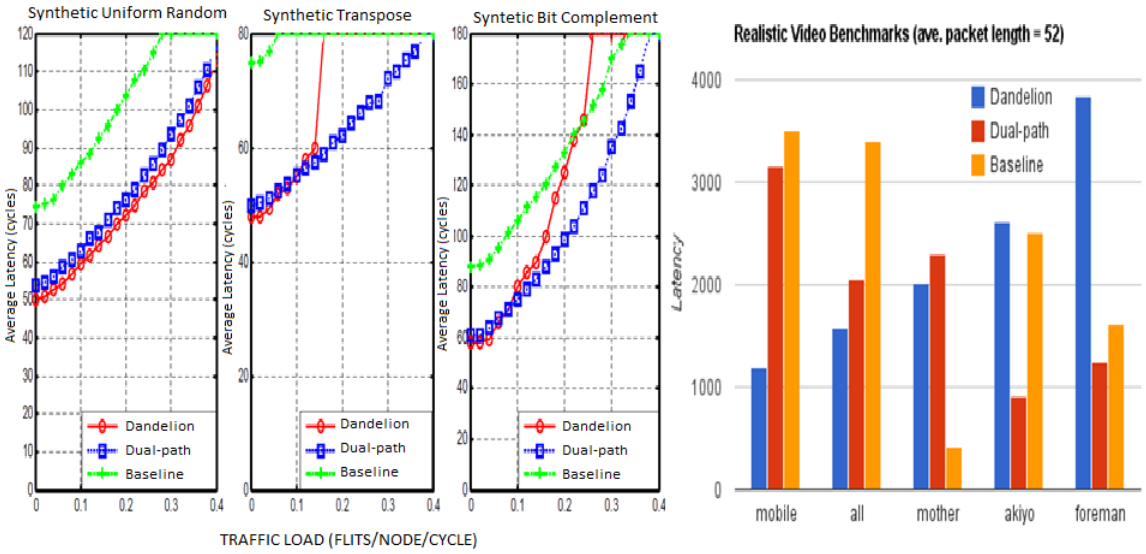

Fig. 16. Synthetic and Realistic Benchmarks for Dandelion Router

Table IV. Synthesis Results of Dandelion compared with Dual-path and Baseline

	Router	Value
Area (μm^2)	Baseline	86842
	Dual-path	+5% (91156)
	Dandelion	+21.7% (105740)
Power (mW)	Baseline	29.75
	Dual-path	+3.8% (30.89)
	Dandelion	+20.6% (35.9)

CHAPTER V

EXPLORATION OF DANDELION FOR VERY LONG PACKETS

As we have already discussed in chapter II, section A, serialization latency can play increasingly major role in overall latency if the packets are longer. Dandelion router promises to reduce the serialization latency to as much as one-fourth. This motivates us to analyze the performance of Dandelion for very long packets.

A. Dandelion for Different Packet Lengths

In this section, we analyze and compare the performance of Dandelion with Dual-path and baseline for packets of lengths ranging from 40 flits to 400 flits as shown in fig. 17. For fixed packet lengths of 40 and 50, Dandelion does better than Dual-path and baseline in all congestion levels. As we increase the packet length further, even though Dandelion has lower latency than Dual-path and baseline for lower congestion, it has higher latency in higher congestion. The main reason behind this is the difference of arrival times of sub-packets at the destination. The latency of the whole packet is the latency of the slowest sub-packet. In other words, the latency of a packet is the worst latency of 4 sub-packets.

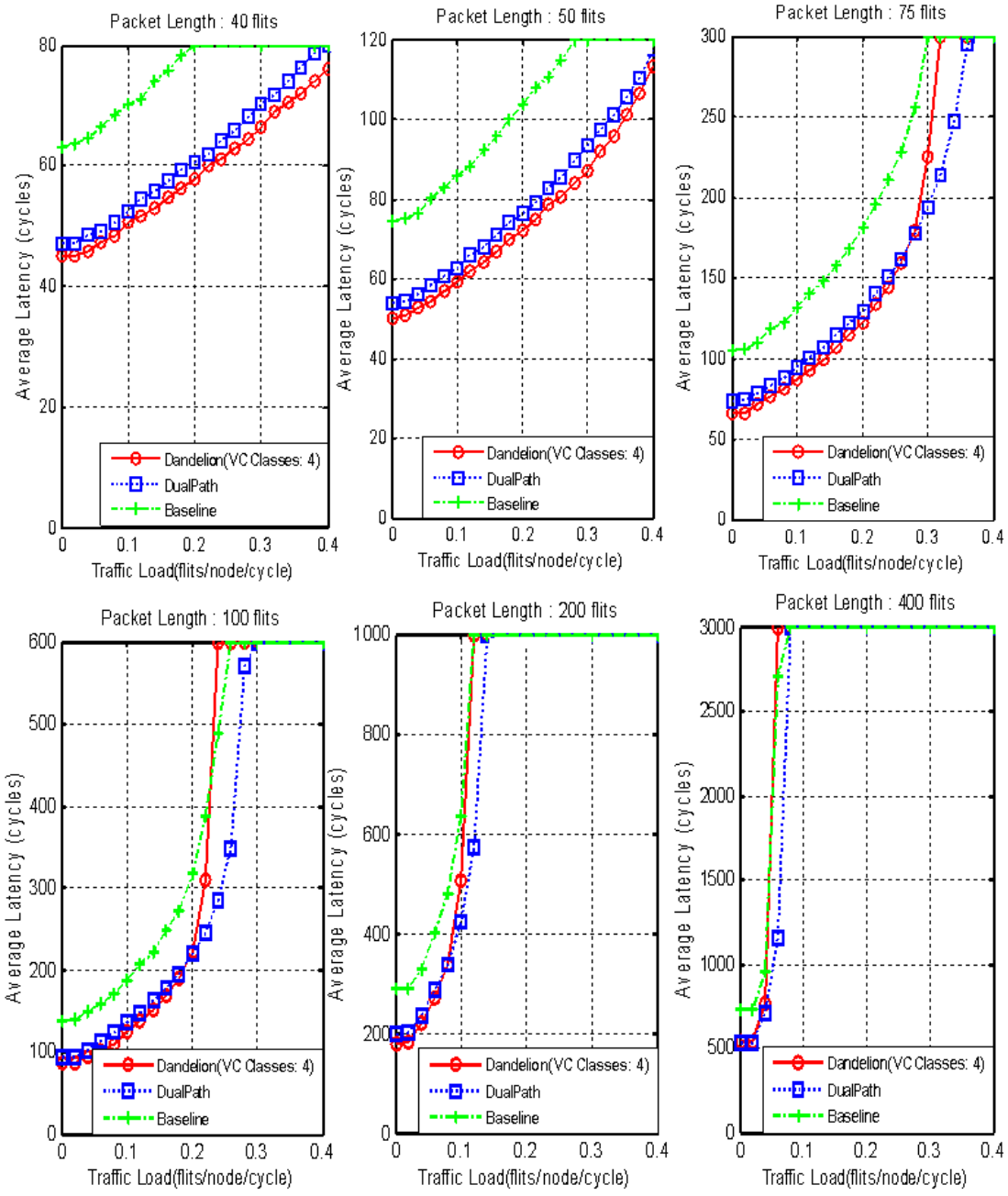

Fig. 17. Average Latency for Different Packet Lengths

In order to delve deeper into the arrival time difference, we plot average arrival time difference of sub-packets against the congestion level in fig. 18. As we notice, the average arrival time difference between sub-packets increases exponentially with congestion. The difference is always larger for Dandelion than Dual-path. The vertical bars denote the standard deviation for average arrival time difference and is also higher for Dandelion than Dual-path. For lower congestion, difference in arrival time is lesser and hence, Dandelion does better than Dual-path in terms of latency. As we increase the congestion, arrival time difference between sub-packets overwhelms the serialization latency gain achieved by splitting the packets. The arrival time difference and it's standard deviation increase with packet length as well, thus making longer packets a wrong choice for Dandelion in higher congestion.

Dandelion saves the serialization latency but is more sensitive to congestion due to exponential increase in the average arrival time difference. In order to support this, we do a single packet test where we inject only a single packet of 100 flits length and analyze it's latency for all the routers. As we see in table V, Dandelion packet latency is less than Dual-path and baseline as the serialization latency of Dandelion is one-fourth of baseline and half as that of Dual-path. However as we increase the injection rate slightly, the latency for Dandelion degrades much faster than that of Dual-path and baseline.

Table V. Single Packet Test

	Latency (in cycles)	
	Single Packet	1% injection
Baseline	153.05	+2.18 (155.23)
Dual-path	93.2	+9.9 (103.1)
Dandelion	70.3	+17.83 (88.13)

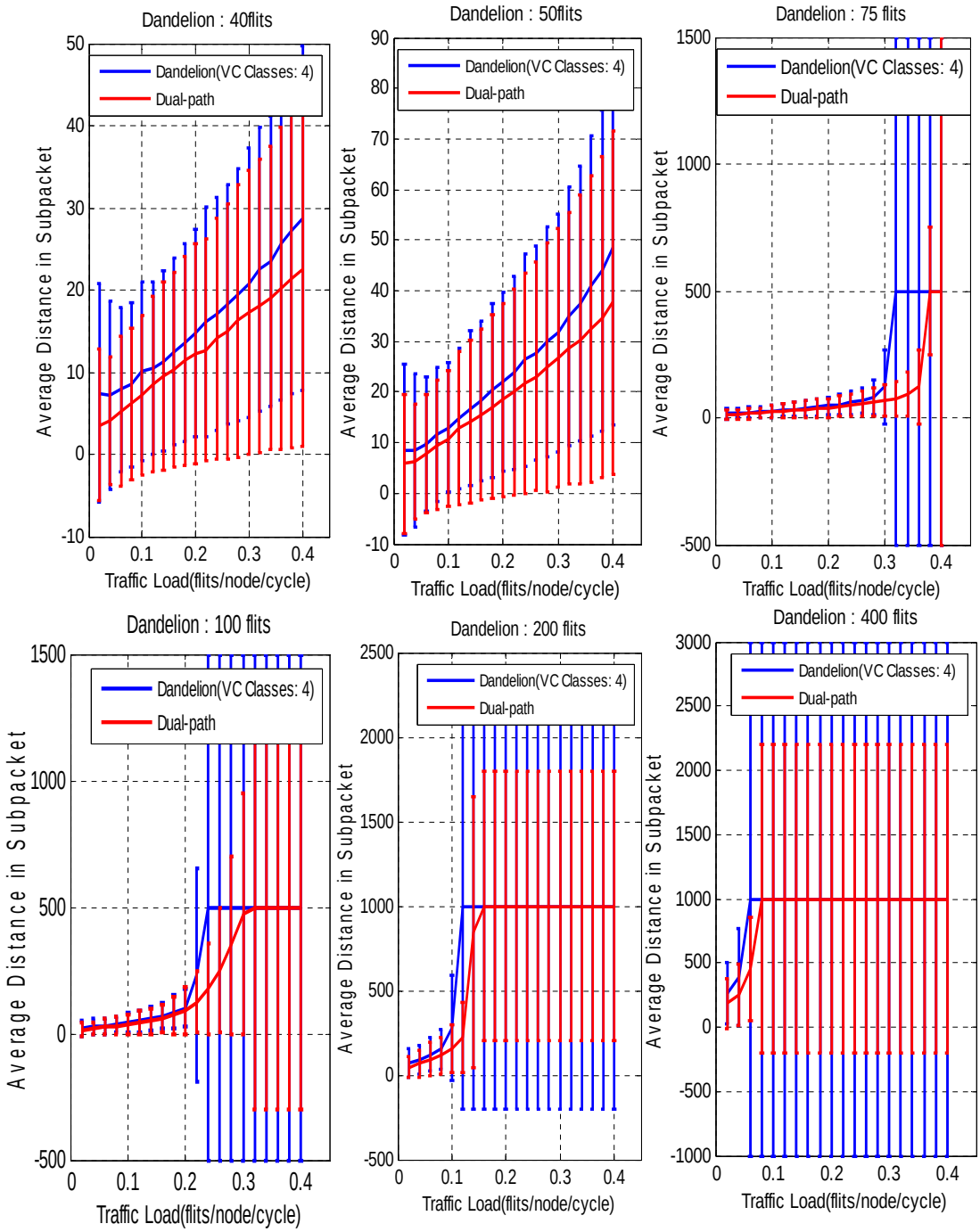

Fig. 18. Average Arrival Time Difference for Different Packet Lengths

B. Dynamic Switching among Dual-path and Dandelion

We know that Dandelion has lower latency than Dual-path for lower congestion, but it doesn't do better for higher congestion. Thus, to achieve similar performance as Dual-path in higher congestion, we devise a mechanism to switch between Dandelion and Dual-path mode depending on the congestion level. Congestion level can be determined by many local factors such as free buffer, free VC, crossbar availability or combinations of these [12, 13] and regional congestion aware techniques [14, 15, 16]. We only consider free buffer status for simplicity. A threshold is determined empirically for individual node based on free buffer status. NoC mode is switched dynamically among Dandelion and Dual-path based on this threshold. The deadlock is automatically avoided while switching back and forth between Dandelion and Dual-path as VCs are distributed in VC classes which break the cyclic dependency.

In fig. 19, we analyze the performance of various routers such as Dandelion with 4 VC classes and Dandelion with 6 VC classes with and without switching mechanism for a packet size of 100 flits and compare with Dual-path and baseline. As we observe, Dandelion with both the VC classes models does better with the switching mechanism as compared to non-switching Dandelion. Switching mechanism is most effective with Dandelion 4 VC classes model because all the VCs are used when the mode is switched to Dual-path. In case of Dandelion 6 VC classes model, minimal and non-minimal paths are separated from each other. Thus when the mode is switched to Dual-path, non-minimal VCs are not used. Dual-path still does better than Dandelion in higher congestion.

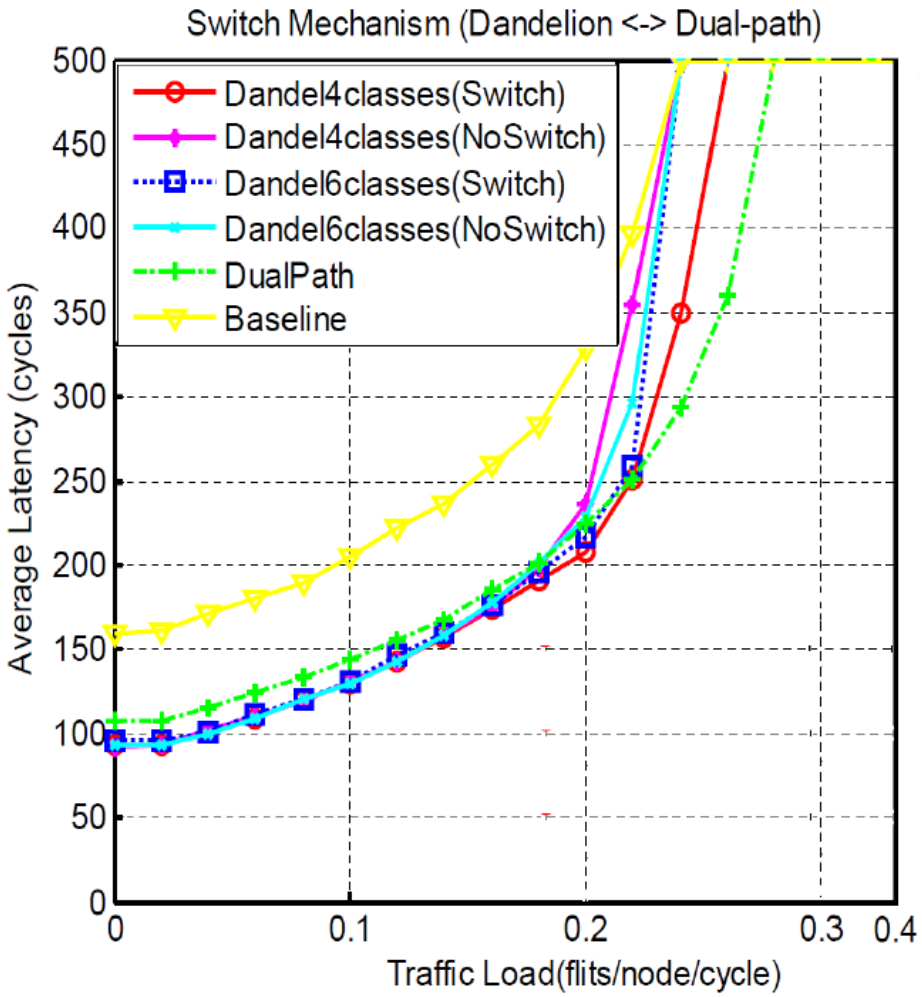

Fig. 19. Comparison of Performance for Dynamic Switching (Packet Length= 100 Flits)

C. Recombination of Sub-packets

Latency is calculated as the time difference between the injection of packet and ejection of the slowest sub-packet. We need a buffer at the destination to store all the sub-packets of a packet until the last sub-packet arrives. Time lag between the slowest and fastest sub-packet increase with the congestion exponentially, requiring a large buffer for recombination. As the technology advances, the size as well as the cost of large buffer decreases making it feasible. Moreover, in typical cache line or DMA data

transfers, memory is already allocated for reception of the data. And for a realistic traffic such as multimedia streaming, the congestion level is generally low, eliding the need for large recombination buffer.

CHAPTER VI

CONCLUSION

This thesis presents multipath router architectures which focus on reducing the serialization latency and exploiting the path diversity by sending split sub-packets in different directions simultaneously. Dual-path router employs two injection and ejection ports and routes the sub-packets in XY and YX DOR minimal paths. Dandelion router employs four injection and ejection ports and exploits minimal as well as non-minimal paths.

Chapter II presents the Dual-path router architecture for reducing the serialization latency and exploiting path diversity. In Chapter III, the NoC has been analyzed for bi-directional interconnects which provide higher saturation bandwidth. Chapter IV introduces Dandelion router which exploits minimal as well as non-minimal paths by implementing injection and ejection ports in all the directions. Chapter V further analyzes the effect of longer packets on Dandelion and establishes that time lag in arrival of different sub-packets at destination overwhelms the serialization gain achieved due to packet splitting. Dynamic switching mechanism is also proposed to improve the performance of Dandelion in higher congestion.

To summarize, this work demonstrates the following:

- Serialization latency is crucial in lengthy multimedia traffic.
- 5-10 flits is the optimal packet length for congestion less than 25% and longer for higher congestion.
- Dual-path router saves serialization latency by packet splitting and reduces congestion by exploiting path diversity.
- Dandelion employs injection/ejection ports in all the directions and uses mini-

mal as well as non-minimal paths.

- Dandelion does 10% better than Dual-path and 33% better than baseline for uniform random traffic.
- Gain due to Dandelion is restricted by arrival time difference of sub-packets at destination.
- Switching mechanism improves performance of Dandelion.

REFERENCES

[1] Hyung Gyu Lee, U. Y. Ogras, R. Marculescu, and N. Chang, "Design space exploration and prototyping for on-chip multimedia applications," in *Proc. 43rd ACM/IEEE Design Automation Conf.*, 2006, pp. 137-142

[2] Li-Shiuan Peh and W. J. Dally, "A delay model for router microarchitectures," *Micro, IEEE* , vol.21, no.1, pp.26-34, Jan/Feb 2001

[3] T. C. Xu, A. W. Yin, P. Liljeberg, and H. Tenhunen, "A study of 3D Network-on-Chip design for data parallel H.264 coding," *NORCHIP*, vol., no., pp.1-6, 16-17 Nov. 2009

[4] Yufei Wang and Zheng Wang, "Explicit routing algorithms for Internet traffic engineering," in *Proc. 8th Computer Communications and Networks conf.*, 1999, vol., no., pp.582-588

[5] R. Banner and A. Orda, "Multipath Routing Algorithms for Congestion Minimization," *Networking, IEEE/ACM Transactions on* , vol.15, no.2, pp.413-424, April 2007

[6] Daeho Seo, Akif Ali, Won-Taek Lim, and N. Rafique, "Near-optimal worst-case throughput routing for two-dimensional mesh networks," in *Proc. 32nd Computer Architecture (ISCA '05)* , vol., no., pp. 432- 443, 4-8 June 2005

[7] G. Michelogiannakis, D. Sanchez, W. J. Dally, and C. Kozyrakis, "Evaluating Bufferless Flow Control for On-chip Networks," *Networks-on-Chip (NoCS 2010), Fourth ACM/IEEE International Symposium on* , vol., no., pp.9-16, 3-6 May 2010

[8] C. Izu, J. Miguel-Alonso, and J. A. Gregorio, "Effects of injection pressure on network throughput," in *Proc. 14th Euromicro Parallel, Distributed, and Network-Based Processing conf. (PDP 2006)* , vol., no., pp. 8 pp., 15-17 Feb. 2006

[9] C. J. Akl and M. A. Bayoumi, "Wiring-Area Efficient Simultaneous Bidirectional Point-to-Point Link for Inter-Block On-Chip Signaling," in *Proc. 21st VLSI Design conf. (VLSID 2008)* , vol., no., pp.195-200, 4-8 Jan. 2008

[10] Ying-Cherng Lan, Shih-Hsin Lo, Yueh-Chi Lin, Yu-Hen Hu, and Sao-Jie Chen, "BiNoC: A bidirectional NoC architecture with dynamic self-reconfigurable channel," *Networks-on-Chip (NoCS 2009), 3rd ACM/IEEE International Symposium on* , vol., no., pp.266-275, 10-13 May 2009

[11] K. C. Bollapalli, R. Garg, K. Gulati, and S. P. Khatri, "On-chip bidirectional wiring for heavily pipelined systems using network coding," in *Proc. IEEE Computer Design conf. (ICCD 2009)* , vol., no., pp.131-136, 4-7 Oct. 2009

[12] W. J. Dally and H. Aoki, "Deadlock-free adaptive routing in multicomputer networks using virtual channels," *Parallel and Distributed Systems, IEEE Transactions on* , vol.4, no.4, pp.466-475, Apr 1993

[13] Jongman Kim, Dongkook Park, T. Theocharides, N. Vijaykrishnan, and C. R. Das, "A low latency router supporting adaptivity for on-chip interconnects," in *Proc. 42nd Design Automation Conf., 2005* , vol., no., pp. 559- 564, 13-17 June 2005

[14] P. Gratz, B. Grot, and S. W. Keckler, "Regional congestion awareness for load balance in networks-on-chip," *High Performance Computer Architecture (HPCA 2008), IEEE 14th International Symposium on* , vol., no., pp.203-214, 16-20 Feb. 2008

[15] R. S. Ramanujam and B. Lin, "Destination-based adaptive routing on 2D mesh networks," *Architectures for Networking and Communications Systems (ANCS 2012), ACM/IEEE Symposium on* , vol., no., pp.1-12, 25-26 Oct. 2010

[16] Sheng Ma, Natalie Enright Jerger, and Zhiying Wang., "DBAR: an efficient routing algorithm to support multiple concurrent applications in networks-on-chip," in *Proc. of the 38th ACM annual international symposium on Computer architecture (ISCA '11)*, New York, NY, USA, 413-424

VITA

Name:	Hrishikesh Nandkishor Deshpande
Address:	C/O Dr. Seong Gwan Choi
	Computer Engineering Group
	Department of Electrical and Computer Engineering
	WERC 333C, MS 3259,
	Texas A&M University,
	College Station, TX 77843-3259.
Email Address:	hrishh4me@tamu.edu
Education:	B.E., Electrical & Electronics Engineering,
	Birla Institute of Technology and Science, Pilani, India, 2008.
	M.S., Computer Engineering,
	Texas A&M University, 2012.

.

The typist for this thesis was Hrishikesh Nandkishor Deshpande.