On Improving the Performance of Hybrid Wired-Wireless Network-on-Chip Architectures

Wen Zong
Department of Computer Science and Eng.
The Chinese University of Hong Kong, HK SAR

Michael Opoku Agyeman
Computing and Immersive Technologies
University of Northampton, UK

ABSTRACT

Recently, hybrid wired-wireless Network-on-Chip (WiNoC) have been proposed to meet the performance and scalability demands of modern System-on-Chip (SoC) design. However, due to the presence of wirelines with multi-hop nodes in the hybrid architecture, WiNoCs have reduced performance efficiency. In this paper, we propose a low-complexity single-cycle bypassing mechanism to alleviate the performance degradation in such emerging hybrid NoCs. The proposed router employs both dimension-ordered routing (DoR) and a deadlock free adaptive routing to transmit flits at low-loads and high traffic loads, respectively, to efficiently balance traffic in WiNoCs. By reducing the latency between the wired nodes and the wireless nodes, the proposed router can improve performance efficiency in terms of average packet delay by an average of 45% (or 50%) in WiNoCs.

Keywords

Router Architecture; Hybrid Wired-Wireless Network-on-Chip; mm-Wave; WiNoC

1. INTRODUCTION

Hybrid wired-wireless Networks-on-Chip (WiNoCs) have emerged to combine the global performance benefits of CMOS compatible wireless layer as well as the short range low power and area benefits of the wireline communication fabric in NoCs. Specifically, conventional wireline based NoCs, are highly efficient for short distances while the wireless layer overcomes their limitations of long distance and scalability. Moreover, the hybrid architecture in WiNoCs reduces the number of on-chip expensive antennas and transceivers (which have non-negligible area and power overheads). An emerging wireless communication fabrics for WiNoCs is the scalable millimeter wave (mm-Wave) which relies on free space signal radiation. While mm-Wave promises to resolve the poor scalability and performance issues of conventional wireline NoC design, the multi-hop traversal in the wireline layer of the hybrid architecture is still a performance bottleneck. Our goal is to mitigate the performance reduction of such communication fabric by proposing an efficient routers architecture that accounts of the manufacturing cost in terms of area and power consumption.

Adaptive routing in the wireline layer can reduce queuing delay at high loads. However, adaptive routing needs to perform per-hop output port selection to avoid congestion, and also requires complex virtual channel allocation scheme to avoid deadlocks [1]. These operations lead to the non-ideal zero-load delays. Recently proposed networks [2] and [3] develop simpler pipelines to reduce zero-load latency. But these designs perform poorly under high-load traffic. Moreover, these routers do not support virtual channel (VC) to reduce complexity and pipeline latency. A low-complexity adaptive router that is able to avoid congestion yet maintain low zero-load latency is desirable for the wireline layer in WiNoCs. WiNoCs must provide reliable data transmission with both low queuing and pipeline delays under the non-uniform and dynamic traffic conditions. Moreover, it is desirable that routers have low power and area overhead with VC support can isolate messages of different classes. Moreover, inhomogeneous 3D NoCs (or WiNoCs) require the multi-hop 2D routers to deliver packets to the 3D routers (or wireless routers) with high efficiency in order to fully exploit the benefit of the short inter layer wires (or single hop wireless layer).

In this paper, we propose a 3-stage adaptive VC compatible router with single-cycle bypassing mechanism to improve the performance efficiency of WiNoCs. The proposed router integrates adaptive routing with low-latency bypassing to overcome the performance bottleneck in the wireline layer of emerging WiNoC architectures. Our main contributions are as follows:

- We propose a low-complexity single-cycle bypassing mechanism for adaptive routers without using side-banded lookahead signals. The bypass datapath applies DoR at low traffic loads to escape the long adaptive routing pipeline and effectively reduces packet delay at low-loads.
- We present a 3-stage non-speculative high-throughput adaptive router that supports the proposed bypassing mechanism. By employing a tagging mechanism the proposed router is able to either avoid congested nodes under high traffic conditions or employ single stage bypassing technique under low traffic conditions.
- We extend the performance of mm-Wave WiNoCs by replacing the slow multi-hop routers in the wireline

MichaelOpokuAgyeman Michael.OpokuAgyeman@northampton.ac.uk

© 2016 ACM. ISBN 123-4567-24-567/08/06. . . $15.00
DOI: 10.475/123_4

layer with the proposed router to provide fast transfer between remote nodes and high performance nodes.

2. RELATED WORK

Latency is one of the key challenges of designing practical on-chip networks [4]. Adaptive routing is a possible solution that helps reduce packet queuing delay effectively [5,6]. However, at low-load conditions, adaptive routing has negligible performance benefit. Moreover, adaptive routers have more complex router architecture. An alternative solution to on-chip latency issue is to perform some of the operations in parallel. In [7] virtual-channel allocation (VA) and switch allocation (SA) are performed in parallel speculatively. Here non-speculative packets are prioritized in SA to increase resource utilization. [8] exploits the abundant bandwidth in routers and multicast flits to output ports speculatively rather than wait for SA. Parallel processing of a packet can also be implemented on different routers with the help of control flits which goes ahead of data flits [9]. SA for a flit is done based on the control flit while the data flit is traversing the link on previous router. When the data flit arrives, it can bypass SA stages and goes directly to switch traversal (ST). However, the sideband network for control flits introduces extra wiring and power overhead.

Low-swing signaling [10] and asynchronous link [11, 12] have been adopted in NoC to allow multiple-hop traversal in one cycle. Low-swing signaling has poor bandwidth density, and asynchronous link can have signal skew issues due to interference [13]. In chips operating at high frequency the signal traversal length can be limited due to the small clock cycle. Structural simplification such as the ring topology allows router to have simple and low-latency micro-architecture [3]. In 2D mesh topology, dimension-sliced router (DSR) is proposed to reduce router cost and latency [2]. DSR abandons the input buffers of routers and also decouples datapath of the two dimensions to reduce cost.

Emerging hybrid NoC architectures will require low-latency adaptive routers to reduce communication latency and also requires VC support in NoC to achieve message isolation. In existing work, latency of NoC routers is reduced by either enhancing the classic router [14] or developing simpler micro-architectures. Approaches like lookaheads [15] add wiring and logic complexity to routers, and increase NoC's area overhead and power consumption. Speculation [7] does not reduce the worst case pipeline delay. Simple NoC micro-architectures like [2, 3] are not adaptive and have no VCs. High radix routers [16, 17] usually have higher serialization delay, and do not work well under adversarial traffic [12]. NoC with multi-hop traversal in single cycle capability such as SMART [12] shows significant latency reduction. However, such feature may not sustain in chips operating at high frequency or with long links (*e.g.* hierarchical topology), or in the combination of two. In contrast, single-cycle-per-hop routers are still good candidates for such scenarios. In this paper, we extend the performance efficiency of WiNoCs by propose a 3-stage non-speculative adaptive VC router to replace the slow wireline routers and develop a low-complexity single-cycle bypassing mechanism to reduce low-load latency without using sidebanded lookahead signals.

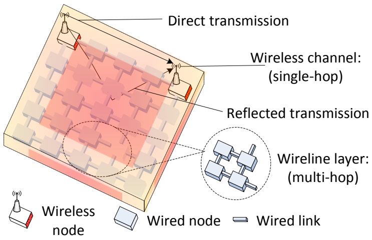


Figure 1: Hybrid wireline-mm-Wave NoC

3. MILLIMETER-WAVE WINOC ARCHITECTURES

Mm-Wave has emerged as a more feasible wireless solution to the on-chip global communication delay issues with promising CMOS components that can scale with transistor technology (Fig. 1). However, the on-chip antennas and transceivers have non-negligible area and power overheads. Conventional wireline based NoCs on the other hand, are highly efficient for short distances despite their limitations over long distance. Consequently, WiNoCs have been proposed to exploit both the global performance benefits of mm-Wave as well as the short range low power and area benefits of the wireline communication fabric in NoCs [18].

In mm-Wave WiNoCs, the routers at the wireless nodes in WiNoCs are equipped with a wireless transmission interface which serves as a bridge between the wireless and the wireline communication layers. The wireless transmission interface, responsible for transmitting and receiving wireless signals, works closely with the routing logic, virtual channel allocator, arbiter and crossbar switch for efficient wireless signal transmission. Routers without the wireless transmission interfaces have to forward packets to the nearest wireless nodes in a multi-hop manner before they can finally exploit the single-hop wireless links to remote destinations. Moreover, if the destination node is not a wireless node, the packet is transmitted to the nearest wireless node and then transmitted through the multi-hop wireless layer. Therefore, the performance of WiNoCs is reduced due to the extra timing overhead and multi-hop transmission of packets in the network. Hence, novel router architectures that offer long range minimal-hop communication with low area and power overheads are required at the non-wireless to exploit the full potential of emerging WiNoCs.

Hence, our objective is to design a router micro-architecture that is able to reduce the average time packets spend along the slow multi-hop wirelines layer.

4. EFFICIENT BYPASS ROUTER ARCHITECTURE FOR WINOCS

We proposed an adaptive virtual-channel router with some single-cycle bypass datapaths for the wireline layer of WiNoCs. The proposed router contains two types of datapaths, one optimized for low latency, the other optimized for adaptivity. Fig. 2 shows the overview of proposed adaptive VC router. The red arrows in the figure represent the low-latency datapath that connect input channel to output channel directly. Input buffers are connected to output ports through the crossbar which forms the adaptive routing pipeline. The bypass datapath is developed from the single-cycle-per-hop

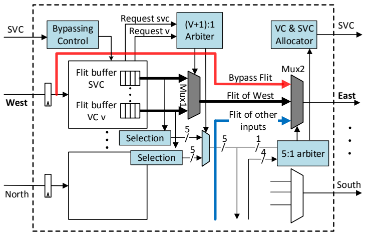


Figure 2: Router micro-architecture

router DSR [2]. Packets traversing through the bypass datapath maintains its progress on current dimension and incurs a single-cycle delay. The adaptive datapath is similar to existing adaptive routers [5] but with a simplified VA scheme. We modify the VA constrain a packet retain its original VC. Moreover, VA is performed after SA in the same cycle non-speculatively. There is a single-bit tag in each flit to notify a downstream router if this flit can utilize the bypass datapath. If the tag bit is set, upon receiving the flit, a router will try to use bypass datapath to transmit the flit, otherwise the router lets it follow the adaptive routing datapath. Packets from all VCs have chances to utilize the bypass datapath using the tagging mechanism proposed in this paper.

5. WIRELINE LAYER INTRA-DIMENSION BYPASSING

At very low-loads, a packet can reach its destination through any of the minimal paths with similar latency. Consequently, we can pre-setup some crossbar connections to provide shorter datapaths in the wireline layer. Packets taking advantage of these paths can avoid crossbar setup and go directly to switch traversal. First, we present the idea of bypassing and its bypass datapath, then we present how adding a dedicated VC for bypassing makes bypassing scalable and practical in a VC router.

DSR [2] elegantly combines routing algorithm with router micro-architecture optimization. We add a set of bypass paths on top of a VC router to achieve single-cycle intra-dimension traversal like that of DSR [2]. Fig. 2 shows the overview of this adaptive router. During SA if an output port receives no requests (indicating that the output port will be idle in next cycle), the output port is connected directly to the input channel of the opposite side in a router. For example, *East* output is connected to *West* input if it receives no requests from the buffered packets. In this case, an incoming packet of *West* input can go directly to the *East* output without waiting for switch allocation. We assume a 128-bit 1.5mm long bypass datapath (including crossbar and link). DSENT [19] reports that the bypass datapath can satisfy a delay constraint of **0.2ns** with proper repeater insertion. Traversing through a bypass path skips the buffering procedure as well as multi-stage allocation procedures and incurs a single-cycle delay. The bypass datapath applies DoR on packets so to utilizes the pre-setup intra-dimension crossbar connections. Utilizing these bypass paths skips the long adaptive routing pipeline and effectively reduces packet delay at low-loads.

Bypassing should be well designed to provide VC compatibility meanwhile sustain the efficiency of intra-dimension bypassing in DSR. An incoming flit may belong to an arbitrary VC. Deciding whether a flit can bypass current router, firstly, the VC must be decoded and then the availability of corresponding credits for downstream router must be checked. We assume the *flit* to be a head flit for illustration purpose, other flits can be processed in a slightly different manner using a small finite-state machine. Aliso, we assume the flit retains its VC ID when bypassing (VA details will be covered in Section 6.2). Suppose the VC ID of a received flit is vc, and the output port of DoR is o. If the following two conditions are met, the received flit can bypass current router in one cycle. Firstly, bypassing must not cause overshooting to the destination (minimal routing). Secondly, the vc at output o must be idle (ensuring a successful VA). Implementing this bypassing logic requires using the VC ID as the input to index corresponding information. This control logic will inevitably increase the critical path length of bypassing logic compared to the one in [2] due to VC decoding. For example, the implementation on X dimension is as follows:

```
bypassing <= (dst.x != current.x) &
             vc_idle[o][vc]
```

Preliminary synthesis result shows that the path delay for this decision making on 16 VCs is **0.1ns** on 45nm standard cell library. In this implementation, the decision making speed slows down as the number of VC increases.

To speedup this process, we introduce a dedicated VC for bypassing. Suppose the special VC introduced is called *slide virtual channel* (SVC). We now only perform bypassing for flits belonging to SVC. To check if a SVC flit can bypass current router, a router only needs to check if SVC of output o is idle. Bypass decision making is faster because we do not need to use VC ID as index to absorb credit information or other information. The processing speed is **invariant** to the number of VCs. If we use SVC for bypassing. The decision making on X dimension is as follows:

```
if (svc) begin
  bypassing <= (dst.x != current.x) &
               svc_idle[o]
end
```

Path delay for this logic is reduced to **0.05ns** using the same 45nm standard cell library.

Only SVC packets are considered for bypassing, and there is also dedicated buffer space reserved for SVC in each router. This design reduces the complexity of bypass decision making. Bypassing with SVC is faster, and more importantly, invariant to the number of VCs. Adding an extra VC does not necessarily increase buffer space in router because most NoC routers use shared buffer between VCs [20].

5.1 SVC Tagging Mechanism

Packets of SVC can enjoy bypassing. Now the problem is: what packet should be tagged with SVC? In this work, all VCs have the chance to be tagged with SVC to reduce overall packet delay and increase link utilization. SVC can be allocated to any packet that wins the output port. All packets are injected to network with the SVC tag being zero. A router updates the SVC tag of a packet after it wins the output port. A packet has the first chance to be tagged with SVC when leaving the its source router. Each output port (excluding ejection port) has a tagging unit. The principle

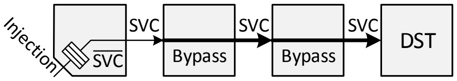


Figure 3: An example of SVC tagging. The packet is injected to network with SVC being zero ($\overline{SVC}$), and is tagged with SVC when leaving its source node. Packets from other input buffers can be tagged with SVC as well

to tag a head flit with SVC is simple, meeting the following two conditions: 1) The SVC tag of the output port is not assigned to any packet. 2)The SVC buffer at corresponding downstream is empty. Otherwise, the SVC tag bit is set to zero. The two rules work together as a lightweight SVC allocator which assigns the SVC tag to packets. A body flit of a packet follows the SVC tag of its head flit, and the tail flit releases the possession of the SVC tag of that output port.

Fig. 3 shows an example of SVC tagging. The SVC flag of this packet initialize to zero ($\overline{SVC}$) when it is injected to network. Suppose the SVC tag of *East* output is idle and the downstream SVC buffer is empty. When this packet wins the *East* output port, according to the SVC tagging rule above, its SVC flag will be set after SA. The downstream router that receive this packet will try to bypass this packet if possible. In this figure, the packet bypass the second router because it is tagged with SVC in the first router. This example shows SVC tagging for the packet in injection port. The SVC tagging works the same for all packets buffered in other input ports. As long as a packet can win an output port, it can be tagged with SVC if the two conditions for SVC tagging are met.

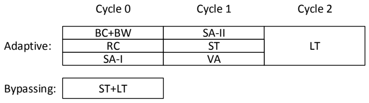
Incoming packet to the bypass path belongs to different VCs. Making bypassing decision is slow due to looking up per VC information. Proposed SVC and its tagging mechanism is a fast and scalable solution for single-cycle bypassing in virtual-channel adaptive routers. In this work, the SVC tagging is **transparent** to CPUs or upper level applications. Any packet that wins switch allocation on current router has a chance to to be tagged with SVC. The packet tagged SVC can enjoy bypassing in the next hop.

6. ADAPTIVE ROUTING

Packet that cannot utilize bypass datapath are routed through the adaptive routing datapath in SlideAcross. We propose a cost-effective adaptive routing pipeline in SlideAcross which is compatible with intra-dimension bypassing. We also propose a simple VA scheme to allow VA be performed efficiently after SA in the same cycle to reduce adaptive routing pipeline. The network is also guaranteed to be deadlock-free based on the proposed VA scheme.

6.1 Router Pipeline

The adaptive router is mainly composed of input buffer, crossbar and allocators. If a received packet cannot bypass current router, it is written to input buffer (BW) and meanwhile route computation (RC) is performed. Adaptive selection is done automatically by masking the congested output port similar to [5]. The crossbar in this router is implemented using two sets of multiplexers like those in [7] to be cost-effective. The SA process thus contains the arbitration for multiplexer of input buffer (SA-I) and that of the output port (SA-II). The winner of SA-II will then transmit a flit to the output link (LT). An idle VC of the output port is also assigned to the the SA-II winner which forms VA procedure. Fig. 4 shows the pipeline of this adaptive routing process. To support bypassing, upon receiving a packet, we need to perform bypassing control (BC) to determine if the packet should be written to buffer, so there is a BC procedure before BW operation in pipeline. If the packet can bypass current router, it follows the single stage bypassing traversal (ST+LT).



	Cycle 0	Cycle 1	Cycle 2
Adaptive:	BC+BW RC SA-I	SA-II ST VA	LT
Bypassing:	ST+LT		

Figure 4: Bypassing and adaptive routing pipeline stages

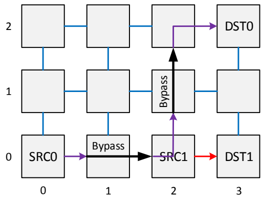


Figure 5: Example of packet paths. The bold arrows are for the single-cycle bypass pipeline, the other arrows refer to the 3-stage pipeline traversal

Fig. 5 uses an example to demonstrate how adaptive routing and bypassing determine the path of a packet. There are two communication pairs in the figure, $SRC0$ to $DST0$ and $SRC1$ to $DST1$. When the packet leaves $SRC0$ it is tagged with SVC and bypassed Router (1,0). But on Router (2,0), it has to be buffered because the *East* port required by bypass is occupied by communication between $SRC1$ and $DST1$. Router (2,0) selects $North$ for this packet to avoid congestion and also tag it with SVC. The packet then bypassed Router (2,1). It is buffered at Router (2,2) due to dimension switching and finally reach the destination $DST0$. This example demonstrates how proposed NoC can avoid congestion and exploit low-load regions to reduce communication delay.

6.2 Virtual Channel Allocation

In SlideAcross, VA is performed non-speculatively after SA in the same cycle according to the pipeline design. VA is hence required to be very lightweight so to prevent increasing the critical path delay of the router dramatically.

NoC uses VCs to implement virtual network (VN) for CMP to isolate different type of messages. Each VN can also contain multiple VCs. In this work, we require at least two VCs (*VC0* and *VC1*) in each VN to prevent routing deadlock. To make VA simple, we require a packet to **retain** its original VC inside its VN. For example, packet of

VC0 will still be *VC0* after successful VA. So the VC of a packet is determined at injection and is not changed during its lifetime in network. Such simple VA rule can be appended to SA process, the winner of an output port also owns the corresponding VC of the output port. The SVC tag (if idle) is also assigned to the winner of SA and performed in parallel with VA. This VA procedure is simpler than what has been done in [21], where VA picks up a VC from the idle VC pool and is done after SA in the same cycle. The high-performance router in [21] demonstrates the efficiency of such pipeline design.

Proposed VA scheme allows VA to be performed efficiently after SA in the same cycle, reducing router pipeline without speculation. Due to this deterministic VC assignment scheme, a head flit requests for SA only when its VC at the output port is idle. So when a head flit wins SA it will surely obtain a VC increasing crossbar utilization. A potential drawback for this simple VA scheme is that the buffer utilization of different VCs can be imbalanced in asymmetric traffic patterns. But this problem can also be solved by sharing buffer between VCs [20].

6.3 Deadlock Avoidance

Routing in this router is minimal and fully adaptive and is hence prone to be deadlock. To break the cycles in resource dependency graph [14], we require at least two VCs (*VC0* and *VC1*) in each VN. A packet is assigned to a VC during injection according to the position of its destination. Packets with destination locating at the left and right side of its source node are assigned to *VC0* and *VC1* respectively. If a packet's destination is on the same column with the source node, the packet can be assigned to either VC randomly or according to congestion status. As the routing is minimal, turns in neither VC form a circle. So both *VC0* and *VC1* are deadlock-free.

Packets from all VCs have chances to use the SVC buffer, so SVC can potentially be a shared media that chains the turns of *VC0* and *VC1* to form a circle. To prevent this deadlock configuration, we only allow one packet to stay in SVC buffer. This is achieved by controlling SVC tagging, a head flit will be tagged SVC only when the downstream SVC buffer is empty as imposed by the second rule in section 5.1. Because SVC contains at most one packet, it will not chain up the turns of different VCs. The rules above all together guarantee a deadlock-free network. Sharing the SVC is also protocol-level deadlock-free. Suppose all SVCs are occupied by a certain class of message, a message of other classes can still reach their destination through the normal VCs, which are guaranteed to drain. So there won't be dependency between different classes of messages making the network protocol-level deadlock-free.

7. EVALUATION

7.1 Impact of Proposed Router on WiNoCs

To further validate the performance benefits of the proposed router in emerging WiNoCs, M5 simulator [22] is employed to acquire memory access traces from a full system running PARSEC v2.1 benchmarks [23] which is used to drive our extended version of Noxim (a cycle-accurate network simulator). In the setup, 64 two-wide superscalar out-of-order cores with private 32KB L1 instruction and data caches as well as a shared 16MB L2 cache are employed. Following the methodology presented in Netrace [24], the memory traces are post-processed to encode the dependencies between transactions. Consequently, the communication dependencies are enforced during the simulation. Memory accesses are interleaved at 4KB page granularity among 4 on-chip memory controllers. A summary of the benchmarks is presented in Table 1. Thus we apply a wide range of benchmarks with varied of granularity and parallelism to study the effects of the proposed bypassing technique on the state-of-the-art wireless communication fabrics on WiNoCs. For each trace, we simulate at least 100 million cycles of the PARSEC-defined region of interest (ROI) where we schedule 2 threads per core. 5 evenly distributed nodes in the WiNoC are equipped with transceivers. All other nodes have receivers. For WiNoCs with bypass techniques, the receiving nodes are enhanced with SlideAcross routers. Similarly, for WiNoCs with SmallWorld, the receiving nodes are enhanced with the 7-port small world routers which have long links with repeaters that connect directly to wireless nodes. Thus packets can exploit both the bypass links and adaptive routing (Buff NVH) within the wireline layer to access the wireless and destination nodes. To model the effect of different BER of the wireline and wireless layer on the network performance in terms of packet latency, we employ packet error ratio (which dictates the probability of packet retransmission):

$$p_p = 1 - (1 - p_e)^{|P|} \qquad (1)$$

where $|P|$ is the packet length in bits and p_e is the bit error probability which is the expectation value of the BER for the communication fabric. Thus, Eq. 1 is modeled and imported into the NoC simulator to assign the probability of retransmission of different communication fabrics at different packet injection rates. Alternating bit protocol is used for transmitting and receiving data, and credit flit (ACK/-NACK). While wormhole flow control is used for the wireline layer, FDMA media access control is adopted to give more than one node the right to transmit over the shared wireless medium at a data rate of 256Gbps in one clock cycle over 128 carrier frequencies. A fixed BER 10^{-7} and 10^{-14} are used for mm-Wave and wireline layer , respectively.

Table 1: Simulated PARSEC traces

Benchmark	Input Set	Cycles	Total Packets
blackscholes	small	255M	5.2M
blackscholes	medium	133M	7.2M
channeal	medium	140M	8.6M
dedup	medium	146M	2.6M
fluidanimate	small	127M	2.1M
fluidanimate	medium	144M	4.6M
swaptions	large	204M	8.8M
vips	medium	147M	0.9M

Figs. 6 show the normalized packet delays of various WiNoCs. Particularly, it can be deduced from Fig. 6 that SlideAcross increases the performance of mm-Wave significantly, compared to SmallWorld and conventional mm-Wave WiNoCs. Besides having a larger crossbar with 7-ports router and longer input buffer waiting time, SmallWorld routing involves intermediate buffering which increases the router pipeline and hence contention in the network. Consequently, packets in SlideAcross experience shorter delays in the reduced pipelined routers which allow bypassing of the input buffers and crossbar.

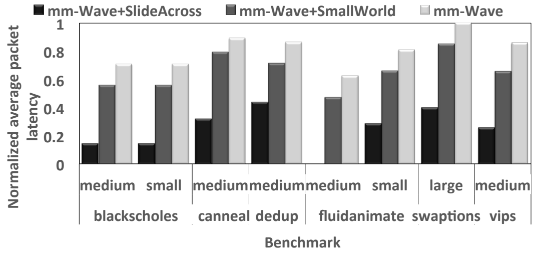


Figure 6: Normalized average packet latency under PARSEC benchmark

7.2 Area and Power Analysis

We implement the proposed router in RTL and synthesized it using Synopsys Design Compiler in TSMC 45nm technology. This 5-port router has 16 buffer slots in each input port and the flit width is 128 bits. The operating frequency is set to be 2GHz. We present the area and power of the adaptive router with and without bypass datapath in Table 2. Bypassing increases the router's area by 3.66% and power by 1.40% mainly due to the larger crossbar used.

Table 2: Area and power estimation

	Area (μm^2)	Power (mW)
Adaptive Router	56896.4	95.6
SlideAcross	59059.8	96.9
Bypass Overhead	3.66%	1.40%

We also examine the power overhead of sidebanded lookahead signals using DSENT [19]. In SWIFT, the flit width is 64-bit, and it requires a 14-bit lookahead flit for each data flit. Table 3 shows the link power differences with and without lookaheads at injection rate of 0.1. The lookahead signals increases SWIFT's link power consumption by 21.9%. SlideAcross can avoid such power overhead introduced by lookahead signals.

Table 3: Lookahead signal overhead

	Dynamic (mW)	Leakage (mW)
Data Flit	1.025	4.847e-02
Data+Lookahead Flit	1.249	5.907e-02
Overhead	21.85%	21.87%

8. CONCLUSION

In this paper, an efficient 3-stage pipelined adaptive VC router with reduced low-load latency is proposed to improve the performance of emerging WiNoC architecture. The proposed router architecture has a cost-effective dual datapath design that is able to minimize packet delay under both low-loads and high loads. A fast bypass datapath is proposed to alleviate the performance degradation due to multi-hops along the long horizontal wires of the state-of-the art hybrid NoC architectures under low-load conditions. Furthermore, a deadlock-free adaptive routing algorithm is proposed to avoid congested paths when the NoC is heavily loaded with traffic. Hybrid wired-wireless NoCs surfer from reduced performance due to the heterogeneity in hop-count and wire-delays between the present communication fabric. As a result, the proposed dual datapath router is designed to extend the performance of the 2D routers in such hybrid NoC. The performance effect of replacing conventional 2D routers with the proposed router architecture in mm-Wave WiNoCs is evaluated by cycle-accurate simulations. The experimental results show significant reductions in the average packet delay compared to existing WiNoCs even when efficient adaptive routing is used.

9. REFERENCES

[1] J. Duato, "A new theory of deadlock-free adaptive routing in wormhole networks," *TPDS*, vol. 4, no. 12, pp. 1320–1331, 1993.

[2] J. Kim, "Low-cost router microarchitecture for on-chip networks," in *Proceedings of Micro*. ACM, 2009, pp. 255–266.

[3] R. Ausavarungnirun *et al.*, "Design and evaluation of hierarchical rings with deflection routing," in *Proceedings of SBAC-PAD*. IEEE, 2014, pp. 230–237.

[4] J. D. Owens *et al.*, "Research challenges for on-chip interconnection networks," *IEEE Micro*, vol. 27, no. 5, pp. 96–108, 2007.

[5] J. Kim *et al.*, "A low latency router supporting adaptivity for on-chip interconnects," in *Proceedings of DAC*. ACM, 2005, pp. 559–564.

[6] P. Gratz, B. Grot, and S. W. Keckler, "Regional congestion awareness for load balance in networks-on-chip," in *Proceedings of HPCA*. IEEE, 2008, pp. 203–214.

[7] L.-S. Peh and W. J. Dally, "A delay model and speculative architecture for pipelined routers," in *Proceedings of HPCA*. IEEE, 2001, pp. 255–266.

[8] Y. He *et al.*, "Mcrouter: Multicast within a router for high performance network-on-chips," in *Proceedings of PACT*. IEEE, 2013, pp. 319–330.

[9] T. Krishna *et al.*, "Swift: A swing-reduced interconnect for a token-based network-on-chip in 90nm cmos," in *Proceedings of ICCD*. IEEE, 2010, pp. 439–446.

[10] C.-H. O. Chen *et al.*, "Smart: a single-cycle reconfigurable noc for soc applications," in *Proceedings of DATE*. EDA Consortium, 2013, pp. 338–343.

[11] T. N. Jain, P. V. Gratz, A. Sprintson, and G. Choi, "Asynchronous bypass channels: Improving performance for multi-synchronous nocs," in *Proceedings of NOCS*. IEEE, 2010, pp. 51–58.

[12] T. Krishna *et al.*, "Breaking the on-chip latency barrier using smart," in *Proceedings of HPCA*. IEEE, 2013, pp. 378–389.

[13] R. Kumar, Y. S. Yang, and G. Choi, "Intra-flit skew reduction for asynchronous bypass channel in nocs," in *Proceedings of VLSI Design*. IEEE, 2011, pp. 238–243.

[14] W. Dally and B. Towles, *Principles and Practices of Interconnection Networks*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2003.

[15] S. Park, T. Krishna, C.-H. Chen, B. Daya, A. Chandrakasan, and L.-S. Peh, "Approaching the theoretical limits of a mesh noc with a 16-node chip prototype in 45nm soi," in *Proceedings of the 49th Annual Design Automation Conference*. ACM, 2012, pp. 398–405.

[16] J. Kim, J. Balfour, and W. Dally, “Flattened butterfly topology for on-chip networks,” in *Proceedings of the 40th Micro*. IEEE Computer Society, 2007, pp. 172–182.

[17] B. Grot, J. Hestness, S. W. Keckler, and O. Mutlu, “Express cube topologies for on-chip interconnects,” in *HPCA 2009*. IEEE, 2009, pp. 163–174.

[18] X. Yu, S. Sah, S. Deb, P. Pande, B. Belzer, and D. Heo, “A wideband body-enabled millimeter-wave transceiver for wireless network-on-chip,” in *International Midwest Symposium on Circuits and Systems (MWSCAS)*, 2011, pp. 1–4.

[19] C. Sun *et al.*, “Dsent-a tool connecting emerging photonics with electronics for opto-electronic networks-on-chip modeling,” in *Proceedings of NOCS*. IEEE, 2012, pp. 201–210.

[20] D. U. Becker, “Efficient microarchitecture for network-on-chip routers,” Ph.D. dissertation, Stanford University, 2012.

[21] A. Kumar *et al.*, “A 4.6 tbits/s 3.6 ghz single-cycle noc router with a novel switch allocator in 65nm cmos,” in *Proceedings of ICCD*. IEEE, 2007, pp. 63–70.

[22] N. Binkert, R. Dreslinski, L. Hsu, K. Lim, A. Saidi, and S. Reinhardt, “The m5 simulator: Modeling networked systems,” *IEEE Micro*, vol. 26, no. 4, pp. 52–60, 2006.

[23] C. Bienia, S. Kumar, J. P. Singh, and K. Li, “The parsec benchmark suite: Characterization and architectural implications,” in *Parallel Architectures and Compilation Techniques*, 2008, pp. 72–81.

[24] J. Hestness, B. Grot, and S. W. Keckler, “Netrace: Dependency-driven trace-based network-on-chip simulation,” in *International Workshop on Network on Chip Architectures (NoCArc)*, 2010, pp. 31–36.