Towards the Practical Design of Performance-Aware Resilient Wireless NoC Architectures

Michael Opoku Agyeman[1], Wen Zong[2], Triantafyllos Kanakis[1], Kin-Fai Tong[3], Terrence Mak[4]
[1]Department of Computer and Immersive Technologies, University of Northampton, Email: Michael.OpokuAgyeman@northampton.ac.uk
[2]Department of Computer Science and Engineering, The Chinese University of Hong Kong, HK SAR
[3]Department of Electrical and Electronic Engineering, UCL, London, UK
[4]ECS, Faculty of Physical Sciences and Engineering, University of Southampton, UK

***Abstract*—Recently, an improved surface wave-enabled communication fabric has been proposed to solve the reliability issues of emerging hybrid wired-wireless Network-on-Chip (WiNoC) architectures. Thus, providing a promising solution to the performance and scalability demands of the fast-paced technological growth towards exascale and Big-Data processing on future System-on-Chip (SoC) design. However, WiNoCs trade-off optimized performance for cost by restricting the number of area and power hungry wireless nodes. Consequently, in this paper, we propose a low-latency adaptive router with a low-complexity single-cycle bypassing mechanism to alleviate the performance degradation due to the slow wired routers in such emerging hyhbrid NoCs. The proposed router is able to redistribute traffic in the network to alleviate average packet latency at both low and high traffic conditions. As a second contribution the paper presents an experimental evaluation of a practically implemented surface wave communication fabric. By reducing the latency between the wired nodes and wireless nodes the proposed router can improve performance efficiency in terms of average packet delay by an average of 50% in WiNoCs.**

***Index Terms*—Router Architecture; Hybrid Wired-Wireless Network-on-Chip; Surface Wave; mm-Wave; WiNoC; Waveguide**

I. Introduction

Recent advances in Cyber Physical System (CPS) that seamlessly integrates the autonomous automobile systems, advanced distributed robotics, medical monitoring (complex biological sensing, computation and actuation) transforms engineering and life sciences into a quantitative, data-rich scientific domain. The large amount of heterogeneous data of high variability sensed from biological and/or non-biological entities of different forms paired with novel data types introduces several challenges to High-Performance Computing (HPC) systems at multiple dimensions. Networks-on-chip (NoC) overcomes these challenges by exploiting the massive fine-grained parallelism and sustaining the inherent communication requirements of big data applications and exascale collective communications.

Hybrid wired-wireless Networks-on-Chip (WiNoCs) have been proposed to combine the low power and area benefits of the traditional wireline as well as single hop traversal of CMOS compatible wireless layer in NoCs. Specifically, conventional wireline based NoCs, are more efficient for localised communication 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). Two emerging wireless communication fabrics for WiNoCs are 1) the scalable millimeter wave (mm-Wave) which relies on free space signal radiation and 2) the reliable 2D Waveguide where the signal is propagated in the form of Zenneck surface wave (SW) on a specially designed sheet which is an inhomogeneous plane that supports electromagnetic wave transmission [1]. WiNoCs in the form of millimeter wave (mm-Wave) relies on free space signal radiation which has high power dissipation with high degradation rate in the signal strength per transmission distance. Moreover, over the lossy wireless medium, combining wireless and wireline channels drastically reduces the total reliability of the communication fabric. On the other hand, Surface wave has been proposed as an alternative wireless technology for low power on-chip communication with improved reliability. Although surface wave-enabled WiNoCs promises to resolve the poor scalability and performance issues of conventional wireline NoC design, the multi-hop among the long wired routers is still a performance bottleneck. Our goal is to mitigate the performance reduction of such communication fabric by replacing the slow wired routers with an efficient router architecture that accounts of the manufacturing cost in terms of area and power consumption.

With regards to packets contending for resources in the wireline layer of WiNoCs, the following observations can be made: 1) On paths without contention, a packet traverses through routers' pipeline without stall and experiences solely the zero-load delay. Contrarily, on congested paths, a packet needs to compete for NoC resources to proceed. 2) A packet that fails to acquire desired resources is stalled, adding a non-deterministic queuing delay to its packet latency. To improve the performance of WiNoCa, both router pipeline and queuing delay should be minimized within the wireline layer to efficiently reduce the communication delay of multi-core workload. We exploit the uneven utilization of resources under different traffic intensities. Consequently, in this paper, we propose to replace the wired routers in WiNoCs with an efficient router architecture (SlideAcross) that employs bypassing and adaptive routing to significantly reduce the average packet delays. SlideAcross is a 3-stage adaptive VC compatible router with single-cycle bypassing mechanism to

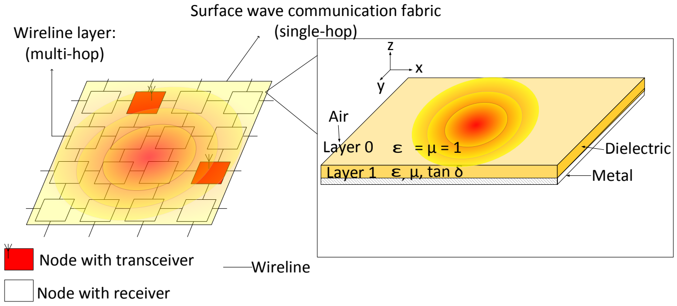


Fig. 1. Hybrid wireline-surface wave NoC

meet the communication needs of emerging communication fabrics for modern CMPs. The proposed router integrates adaptive routing with low-latency bypassing in a cost-effective way to overcome the drawbacks of existing adaptive routing and low-latency architectures. The rest of the paper is organized as follows. Section II discusses the state-of-the-art high performance communication fabrics for CMPs and formulates the problem of improving their performance. Section III shows the overview of the proposed SlideAcross router. Here, the proposed bypass datapath and the practical bypassing mechanism are presented. Section IV presents the adaptive routing pipeline and the proposed VA and deadlock avoidance technique. Section V, presents the performance evaluation of the proposed router in WiNoCs. We also include area and power overhead estimation as well as a practical evaluation of surface wave communication fabric. In Section VI, we conclude this work.

II. Emerging High Performance Communication Fabric CMPs

A. Surface Wave Enabled WiNoCs Architectures

RF interconnects are CMOS compatible, and hence are power and area efficient. On the other hand, they rely on long transmission lines for transmitting guided waves which requires alignment between transmission pairs. Conversely, mm-Wave has been proposed as a more feasible CMOS compatible on-chip wireless solution. However, the on-chip antennas and transceivers have large area and power overheads. Traditional wireline based NoCs, are highly efficient for localised data transmission. 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. However the wireless communication fabric is lossy with reduced reliability [2], [3]. Surface wave communication has been recently demonstrated as a feasible wireless NoC solution with improved global data transmission, low-power and high bandwidth [1], [4]. In surface wave based-WiNoCs, the wireless layer is implemented with a dielectric coated metal layer as waveguide medium (Fig. 1). Compared to mm-Wave based WiNoCs, surface wave has lower power and lower latency with reasonably high performance to area ratio. [4].

In WiNoCs, the routers at the wireless nodes are equipped with a wireless transmission interface which separates the wireline layer from the wireless layer. 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. Consequently, WiNoCs have extra delays due to 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.

B. Problem Formulation

Transmitting packets over multi-hops along the long horizontal links may result in significant latencies due to the buffering, hop-by-hop traversal and limited number of wireless nodes. To this end, the *M/M/1/B* queueing model is employed as a closed-form expression or the average packet latency. Here the number of nodes in a transmission queue can be derived as [5]:

$$\zeta^h_{(i_k,i_{k+1})} = \frac{\rho^h_{i_k,i_{k+1}} + (\beta_{h\rho_{i_k,i_{k+1}}} - \beta_h - 1)(\rho^h_{i_k,i_{k+1}})^{\beta_h+1}}{(\rho^h_{i_k,i_{k+1}} - 1)((\rho^h_{i_k,i_{k+1}})^{\beta_h+1} - 1)}, \tag{1}$$

Adopting Little's results [6], the average time spent over any path q_{ij} can be given by:

$$T^h_{q_{ij}} = \sum_{i_k,i_{k+1} \in q_{ij}} \left(\frac{\zeta^h_{(i_k,i_{k+1})}}{\lambda^h_{i_k,i_{k+1}}(1 - P^{block}_{(i_k,ik+1),h})}\right) \tag{2}$$

where $P^{block}_{(i_k,ik+1),h}$ is the blocking probability:

$$P^{block}_{(i_k,ik+1),h} = \frac{((\rho^h_{i_k,i_{k+1}})^{\beta_h})(\rho^h_{i_k,i_{k+1}} - 1)}{(\rho^h_{i_k,i_{k+1}})^{\beta_h+1} - 1}, \tag{3}$$

β_h is the relative buffer length of the router with respect to application with β_h= $\frac{\beta}{L_h}$. L_h and β are the packet length [flits] of application h and buffer size, respectively. $\rho^h_{i_k,i_{k+1}}$ is the intensity of the traffic at link (i_k, i_{k+1}) which is given by:

$$\rho^h_{i_k,i_{k+1}} = \frac{\lambda^h_{i_k,i_{k+1}}}{\mu^h_{i_k,i_{k+1}}} \tag{4}$$

where $\lambda_{i_k,i_{k+1}}$ is the aggregated incoming traffic of application h [packets/s] traversing link (i_k, i_{k+1}) including the traffic from previous nodes that are either directly or indirectly connected to the node. $\mu^h_{i_k,i_{k+1}}$ [packets/s] is the service rate, which is expressed as:

$$\mu^h_{i_k,i_{k+1}} = \frac{\log(1 + \gamma_{k,k+1})}{8L_h}. \tag{5}$$

Here, W is the available bandwidth at node i_k.

Hence to solve the problem of improving the performance efficiency of WiNoCs, our objective is to design a router

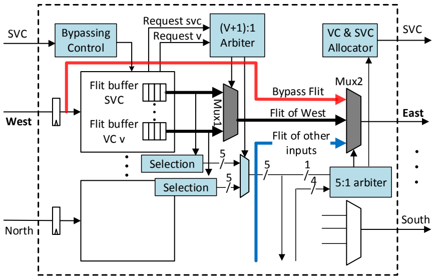


Fig. 2. Router micro-architecture

micro-architecture that is able to reduce the average time T packets spend along the slow wireline layer such that:

$$\underset{\forall(i_k,i_{k+1},h)}{\mathtt{min}}(T^h_{q_{ij}}) \tag{6}$$

subject to:

$$\psi = (A_{new_R} - A_{old_R}) + (P_{new_R} - P_{old_R}) \tag{7}$$

where

$$\psi \leq min \tag{8}$$

where new_R and old_R are the proposed new router and conventional wired routers micro-architecture, respectively. A_x and P_x represents the area and power consumption of router x. The most efficient design has a ψ = 0.

III. Proposed Router Architecture

We propose to replace the multi-hop routers in the wired layer of WiNoCs with SlideAcross, an adaptive virtual-channel router with some single-cycle bypass datapaths. SlideAcross contains two types of datapaths, one optimized for low latency, the other optimized for adaptivity. Fig. 2 shows the micro-architecture of proposed router. Input buffers are connected to output ports through the crossbar which forms the adaptive routing pipeline. For adaptive routing, each input port has a dedicated VC for bypassing, Slide Virtual Channel (SVC) buffer reserved for fast packet traversal. The crossbar is composed of input multiplexers and output multiplexers to be cost effective [7], [8]. The red bold arrow in the figure is a bypass datapath that connects *West* input link directly to the *East* output multiplexer. *Mux2* connects the red arrow with *East* output port when there's no request for *East* output port, forming the one of the pre-setup intra-dimension bypass datapaths.

Control modules are colored with blue in this figure, including the bypassing control, input multiplexer arbiter, output multiplexer arbiter, VC allocator and SVC allocator. VC and SVC allocator absorb the arbitration result of the 5:1 arbiter (SA-II) and allocate VC and SVC tag to the winning packet accordingly. Selection units automatically select the less congested for buffered packets by masking the congested output port in the output port request vector.

The bypass datapath is developed from the single-cycle-per-hop router DSR [9]. 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 [10] 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.

A. Wired Layer Intra-dimension Bypassing

At very low-loads, a packet can reach its destination through any of the minimal paths with similar latency. Inspired by this, we can pre-setup some crossbar connections that are potentially useful for some packets. We add a set of bypass paths on top of a VC router to achieve single-cycle intra-dimension traversal to provide shorter paths between wired routers and wireless routers. 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 [11] 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.

Our goal is to design an adaptive VC router with reduced low-load latency. 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. Also, we assume the flit retains its VC ID when bypassing (VA details will be covered in Section IV-B). 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 [9] 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 [12].

IV. 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.

A. 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 [10]. 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. 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).

B. 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 [13], 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 [13] 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 [12].

C. 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 III-A. 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.

V. Evaluation

A. Impact of Proposed Router on Surface Wave-Enabled WiNoCs

To validate the performance benefits of the proposed router in emerging WiNoCs, M5 simulator [15] is employed to acquire memory access traces from a full system running PARSEC v2.1 benchmarks [16] 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 [17], 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. 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):

$$p_p = 1 - (1 - p_e)^{|P|} \tag{9}$$

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. 9 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^{-13} and 10^{-14} are used for surface wave and wireline layer, respectively.

Fig. 3 show the normalized packet delays of various WiNoCs. As shown in Fig. 3, while SmallWorld can improve the performance of SW WiNoCs, SlideAcross significantly outperforms SmallWorld in all workloads. 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. Also, in benchmarks with heavy traffic such as swaptions, SlideAcross achieves over 50% performance improvement on the average in both SW enabled WiNoCs compared to SmallWorld.

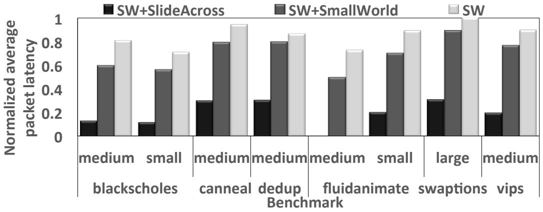


Fig. 3. Normalized average packet latency under PARSEC benchmark

B. Evaluation of Surface Wave Communication Fabric

To further investigate the practicality of the performance benefit of SW communication fabirc, an experiment was conducted as follows. A Keysight N5250A-017 millimeter-wave network analyzer was used for the transmission loss (S_{21}) of the surface wave platform. The network analyzer was first calibrated and normalized to eliminate the cable and SW transducers effects, i.e. S_{21} equals 0dB when the two transducers were directly connected (Fig. 4(a)). Then the two transducers are placed on a piece of Taconic TLY-5 microwave substrate (ε_r = 2.2, thickness = 0.38mm, loss tangent at 10GHz = 0.0009, 300 × 300mm^2). The separation between the two transducers was then fixed at 150mm for the measurement (Fig. 4(b)). Here, the bottom side of the substrate was copper-cladded (electrical conductivity = 5.96 × 10^7S/m) with surface impedance at 60GHz is 0.063 + j98.2ohm. Fig. 5 shows the S_{21} transmission measurement results for an exaggerated separation between the two transducers of about 10cm. The figure shows that the 3dB bandwidth is from 37.5 to 80GHz which is 42.5GHz. Though the conducted experiments is at the macro-level the achieved bandwidth is significantly wide and promising for multi-core application. This serves as a very good foundation towards the development of a practical SW fabric for on-chip communication.

VI. Conclusion And Future Work

In this paper, an efficient router with reduced low-load latency is proposed to improve the performance of surface wave-enabled WiNoC. 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. Particularly, the proposed router employes a fast bypass datapath to reduce the long packet delays due to multi-hops along the long horizontal wires. Furthermore, a deadlock-free adaptive routing algorithm is proposed to avoid congested paths when the NoC is heavily loaded with traffic. Cycle-accurate simulation is conducted to evaluate the performance effect of replacing conventional wired routers with the proposed router architecture in WiNoCs. The simulation results reveal significant improvement in terms of average packet delay compared to existing surface wave-enabled WiNoC even when efficient adaptive routing is used. Future work includes a practical implementation of SW communication fabric at the on-chip (nanotechnology) level.

References

[1] M. O. Agyeman, J. X. Wan, Q. T. Vien, W. Zong, A. Yakovlev, K. Tong, and T. Mak, "On the design of reliable hybrid wired-wireless network-on-chip architectures," in *IEEE Embedded Multicore/Many-core Systems-on-Chip (MCSoC)*, 2015, pp. 251–258.

[2] 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 *Proceedings of MWSCAS*, 2011, pp. 1–4.

[3] C. Xiao, Z. Huang, and D. Li, "A tutorial for key problems in the design of hybrid hierarchical noc architectures with wireless/rf," *Smart CR*, no. 6, pp. 425–436.

[4] M. O. Agyeman, Q. T. Vien, A. Ahmadnia, A. Yakovlev, K. F. Tong, and T. Mak, "A resilient 2-d waveguide communication fabric for hybrid wired-wireless noc design," *IEEE Transactions on Parallel and Distributed Systems*, vol. PP, no. 99, pp. 1–1, 2016.

[5] J. MacGregor Smith, "Properties and performance modelling of finite buffer m/g/1/k networks," *Comput. Oper. Res.*, vol. 38, no. 4, pp. 740–754, 2011.

[6] S. K. Bose, *An Introduction to Queuing Systems*. Springer Press, 2001.

[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] R. Mullins, A. West, and S. Moore, "Low-latency virtual-channel routers for on-chip networks," in *Proceedings of ISCA*, vol. 32. IEEE, 2004, p. 188.

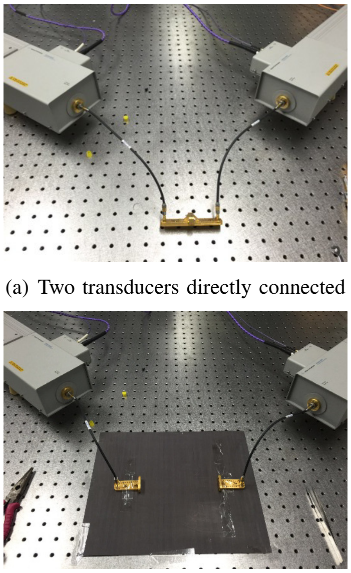
[9] J. Kim, "Low-cost router microarchitecture for on-chip networks," in *Proceedings of Micro*. ACM, 2009, pp. 255–266.

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

[11] 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.

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

[13] 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.



(a) Two transducers directly connected

(b) Two transducers separated by a distance

Fig. 4. Practical Implementation of surface wave communication fabric

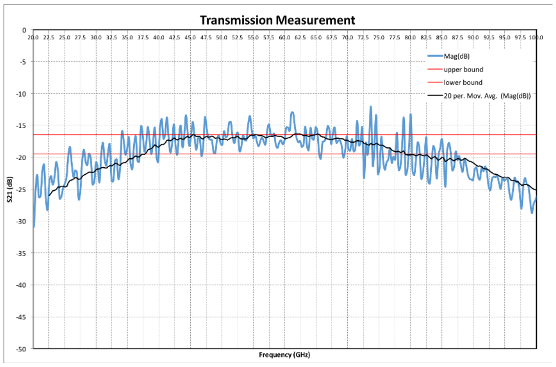


Fig. 5. S_{21} (transmission) measurement results

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

[15] 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.

[16] 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.

[17] J. Hestness, B. Grot, and S. W. Keckler, "Netrace: Dependency-driven trace-based network-on-chip simulation," in *Proceedings of NoCArc*, 2010, pp. 31–36.