

Efficient of Multicast Wireless NOC Architecture by using Hammer Protocol

S. WASIM AKRAM¹, G. JYOSHNA²

¹PG Scholar, Dept of ECE, Sri Krishnadevaraya University College of Engineering and Technology, Anantapuramu, AP, India.

²Lecturer, Dept of ECE, Sri Krishnadevaraya University College of Engineering and Technology, Anantapuramu, AP, India.

Abstract: Today's multiprocessor platforms employ the network-on-chip (NOC) architecture as the preferable communication backbone. Conventional NOCs are designed predominantly for uni cast data exchanges. In such NOCs, the multicast traffic is generally handled by converting each multicast message to multiple unicast transmissions. Hence, applications dominated by multicast traffic experience high queuing latencies and significant performance penalties when running on systems designed with unicast-based NOC architectures. Various multicast mechanisms such as XY-tree multicast and path multicast have already been proposed to enhance the performance of the traditional wire line mesh NOC incorporating multicast traffic. However, even with such added features, the multi hop nature of the wire line mesh NOC leads to high network latencies and thus limits the achievable system performance and to sustain the high-bandwidth and high-throughput requirements of emerging applications, we propose the design of a wireless NOC (Wi NOC) architecture incorporating necessary multicast support. By integrating congestion-aware multicast routing with network coding, the Wi NOC is able to efficiently handle heavy multicast injections. For applications running with a broad cast heavy Hammer cache coherence protocol, the proposed multi cast aware Wi NOC achieves an average of 47% reduction in message latency compared with the XY-tree-based multicast-aware mesh NOC. This network level improvement translates into a 26% saving in full-system energy delay product.

Keywords: Multicast Communication, Network Coding (NC), Networks On Chip (NoCs), System-On-Chip, Wireless NoCs (WiNoC).

I. INTRODUCTION

Collective communication i.e. multicast/broadcast, lies on the critical path of many real-world applications with intensive data exchange requirements on many-core platforms. Traditional networks on chip (NOCs) treat the collective communication as repeated unicast, which easily leads to the following problems:

- Local congestion on the source node.
- Poorly guaranteed QOS.
- Global congestion due to competition for the same network resource among repeated unicast packets.
- Power consumption overhead due to repeated injections of the same packets.

Therefore, without properly supported by the underlying NOC architectures and routing protocols, the overall network performance deteriorates sharply even as the percentage of collective communication increases by a small amount. This prevents NOCs to exploit massive-scale parallelism in many critical large-scale applications, which are inherently embedded with considerable collective traffic requirements. Neural-network based computing real-time object recognition processing genetic-algorithm-based protein folding analysis and neuromorphic computing are a few of the SOC applications that exhibit significant multicast traffic patterns. Moreover, many SOC control functions such as passing global states, managing, and configuring the on-chip networks and implementing: 1) operand exchange networks; 2) non uniform cache architectures; and 3) cache coherence protocols also require efficient multicast data exchange. To elaborate on this, as examples, we consider the traffic patterns associated with two different cache coherence protocols, directory and Hammer protocols. Both directory and Hammer protocols can give rise to traffic hotspots and generate significant number of long-range traffic patterns warranting low-latency NOC links, even among the physically far apart processing cores.

However, unlike the directory protocol that involves only sparse multicasts, the Hammer protocol involves dense broadcast injections and hence stresses the NOC affecting the overall performance when implemented on systems predominantly designed for unicast-based transmissions. Moreover, the multicast traffic arising due to the cache coherence protocols is inherent to the many-core platform and can be observed irrespective of the executed applications. On the other hand, the on-chip area overhead associated with implementing the Hammer protocol is very low, and hence, it is highly preferable for large many-core chips than the traditional directory protocol. Hence, designing a high-performance multicast-aware NOC that enables an efficient implementation of Hammer cache coherence protocol is a highly relevant research topic. As we show later, cache coherence protocols also exhibit multicast bursts, leading to high multicast traffic volumes at a particular instance of time and thus necessitating a multicast and congestion-aware NOC. Wireless NOC (WiNoC) is an emerging paradigm to design high-bandwidth and energy-efficient communication backbone for many-core chips.

II. MULTICAST-AWARE WiNoC NETWORK DESIGN

As discussed earlier, in this paper, we strive to design a congestion-aware low-hop-count multicast-enabled NoC. In this context, we present the design of an SW-network-based multicast-enabled WiNoC architecture.

A. WiNoC Design Constraints

In our WiNoC, each processing core is connected to a router and the routers are interconnected following a power law-based SW connectivity[1]. Essentially, SW networks incorporate multiple short-range links and a few long-range shortcuts so as the overall inter-router hop count (H_{avg}) is minimized. However, while adding links in SW NoCs we need to follow certain restrictions. First, we should restrict the average number of ports (K_{avg}) per router to four so that the WiNoC does not introduce any additional router port overhead compared with a conventional mesh. Next, we should restrict the maximum number of ports in a router (K_{max}) so that no particular router becomes unnecessarily large. For a system size of 64 cores, the SW-based networks achieve highest throughput with lowest energy dissipation when the maximum port count in a router is restricted to seven [1]. Finally, as we explain for link lengths exceeding 6.25 mm in 28 nm CMOS technology node, wireless links are more efficient than the traditional metal wires in terms of energy delay product (EDP). Hence, WiNoCs should prefer using wireless links instead of wireline links for data exchanges exceeding a communication distance of 6.25 mm. However, depending on the available wireless resources and allowed area overhead, we can make only a limited number of the longest links wireless, while the others need to remain wireline. Considering these facts, for our WiNoC, we follow a region based wireless node placement strategy. In this strategy, the whole system is divided into multiple equal-sized non overlapping regions and each region is provided with a set of WIs. Principally, in this design, the short-range communication (within a region) is handled through the wireline links, while the long-range communication (inter-region data exchange) is handled using wireless links. In each considered region, we place the WIs such that the communication distance from all the non wireless nodes to their nearest WIs is minimized. The number of WIs allowed in a region is equal to the number of available non overlapping wireless channels. So far, it has been demonstrated that using on-chip mm-wave wireless links, it is possible to create three non overlapping channels. For a system of 64 nodes, it is already shown that placing the WIs at the center of each region minimizes the communication distance from any node to its nearest WI, and hence it ensures that the utilization of wireless medium is maximized. Fig. 3.9(a) shows an example where a system of 64 cores is divided into four regions (quadrants), with each quadrant having three WIs in its center.

B. Orthogonal Paths

In order to facilitate the congestion-aware multicast routing we propose that each non wireless node in the system is connected to all the WIs in its own region via two orthogonal wireline paths, where none of the links in these two paths are

being the same. This can be achieved by creating two orthogonal spanning trees for each region from the existing wireline connectivity.

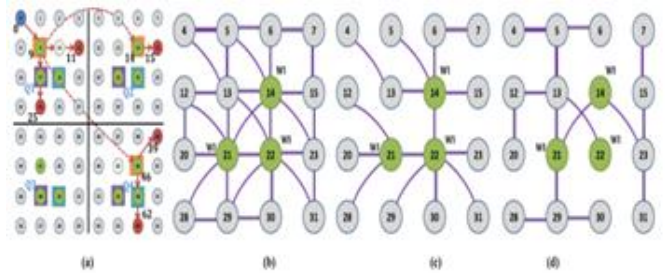


Fig1. Sample WiNoC multicast scenario where a source node(0) in quadrant Q1 is transmitting a multicast message to destinations across three different quadrants.

In order to more clearly explain the orthogonal paths concept, Fig1 shows the entire wireline connectivity of a region and the two orthogonal spanning trees associated with it. Fig1(b) illustrates the entire wireline connectivity associated with quadrant Q₂.

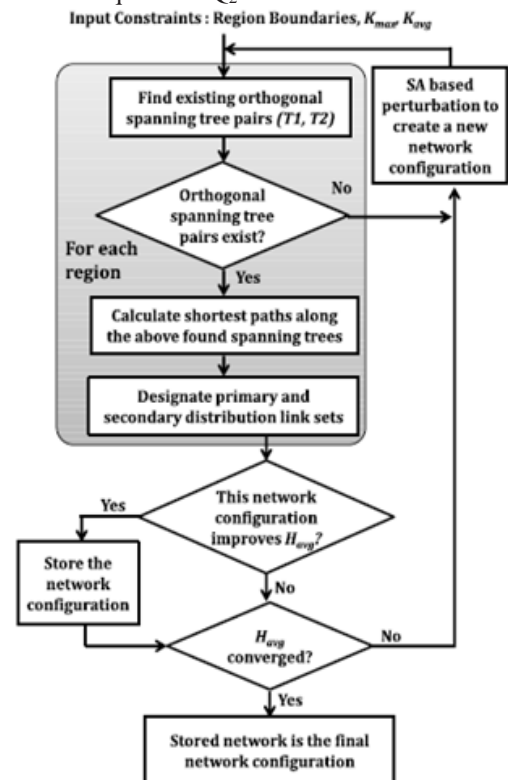


Fig2.Flow diagram for establishing wireline connectivity.

Fig1(c) and (d) shows the two orthogonal spanning trees created from the wireline connectivity shown in Fig. 1(b). The links constituting the connectivity shown in Fig. 1(c) and (d) are called the primary distribution links and the secondary distribution links, respectively. Among these two sets of links, the set of primary distribution links [Fig. 1(c)] provide minimal hop count paths between all non wireless nodes and their nearest WI. Hence these links are usually preferred for distributing messages from a WI to its nearest non wireless

Efficient of Multicast Wireless NOC Architecture by using Hammer Protocol

destinations and thus have the name primary distribution links. The other set of links, the secondary distribution links will be used for this distribution in case the primary distribution links are not available due to congestion. Moreover, as it can be observed from Fig. 1(b)–(d), the primary and secondary distribution links are merely subsets of the existing intra quadrant wireline connectivity, and hence, employing orthogonal wireline paths does not introduce any additional wireline overheads. Finally, following the above-discussed design principles and constraints, we employ a simulated annealing (SA)-based mechanism [30] to place the wireline links so that the average inter-router hop count (H_{avg}) is minimized (as shown in Fig. 2). In this SA-based mechanism, at the start of each cycle, a perturbation is created in the current network connectivity to create a new network configuration. In this perturbation, it is also ensured that the input constraints such as K_{max} and K_{avg} are preserved. Then, the wireline connectivity of each region is divided into pairs of orthogonal spanning trees (e.g., each pair has two spanning trees named T1 and T2). For each region, the links of the spanning tree (say T1) that provide the minimal hop count paths between all non wireless nodes and their nearest WI constitute the set of primary distribution links. The links in the spanning tree, which is its orthogonal pair (T2), constitute the set of secondary distribution links. It should be noted that this wireline link addition mechanism will work for a WiNoC with any number of regions and also for any region size. The only constraint that should be met is that for a region with N nodes, there should be at least $2(N - 1)$ intra region links in order to establish the two orthogonal spanning trees (this is because each spanning tree is constituted of $N - 1$ links).

C. Components of the Wireless Interface

The two principal components of a WI are the antenna and the transceiver. The WiNoC uses a metal zigzag antenna that has been demonstrated to provide the best power gain with the smallest area overhead. A detailed description of the transceiver circuit is out of the scope of this paper. However, the transceiver was designed following. The WI is completely CMOS compatible and no new technology is needed for its implementation. We use the three non overlapping on-chip wireless channels already demonstrated in frequency ranges of 30, 60, and 90 GHz [3]. Each WI is tuned to one of these three frequencies.

III. ROUTING AND MAC PROTOCOLS

The WiNoC has an irregular network topology and hence requires a topology agnostic routing method. Thus, principally, we follow adaptive layered shortest path (ALASH) for data routing in WiNoC [31]. ALASH is built upon the layered shortest path (LASH) algorithm [36]. The LASH algorithm takes advantage of the multiple virtual channels in each port of the NoC routers in order to route messages along the shortest physical paths. In order to achieve a deadlock-free operation, the network is divided into a set of virtual layers, which are created by dedicating the virtual channels from each router port into these layers. The shortest physical path between each source–destination pair is then assigned to a layer such that the

layer's channel dependency graph remains free from cycles and this ensures the deadlock freedom. The ALASH protocol improves on the LASH routing scheme, by enabling an adaptive layering function that considers the expected traffic patterns. We follow the priority layering function explained. Priority layering allocates as many layers as possible to source–destination pairs with high traffic intensities. This improves the adaptability of messages under high traffic intensities by providing greater routing flexibility. In this thesis, we propose to expand this basic ALASH protocol to perform efficient multicast routing (called MALASH protocol). To start, in our WiNoC, for all source–destination pairs residing in the same region, we perform the above discussed layering for shortest paths along both the primary and the secondary distribution links. The entire WiNoC design flow to achieve a layered-routing-enabled WiNoC is shown in Fig. 3. In Fig3 F_{ij} denotes the matrix containing frequency of traffic interactions between all the routers and $N_V L$ denotes the number of available virtual layers.

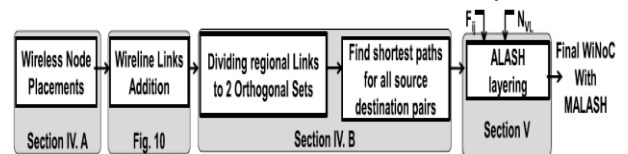


Fig3.Overall WiNoC design flow to enable MALASH routing.

A. Transmission of Messages

In MALASH routing, the multicast messages are distributed using a region-based approach [as shown in Fig3]. To distribute the message to destinations that are present in the same region as that of the source, MALASH uses wireline links. To distribute the message to the destinations located in the non source regions, MALASH uses a combination of wireless and wireline links. First, the multicast message is forwarded from the source node to the nearest WI. Then, from this WI, the message is broadcast using the wireless channel. Finally, WIs residing in the non source regions receive this broadcast message and distribute it to the intended destinations using wireline links. This wireline distribution is carried out using the tree multicast mechanism where the message is first forwarded to a set of intermediate nodes. The message is then replicated at these intermediate nodes before being forwarded to the intended final destinations [2]. Following the congestion aware routing mechanism either the primary or the secondary distribution links are used to reach a final destination node from the destination WI. 1) Example: As an illustrative example, let us consider the situation shown in Fig3 where the whole system is divided into four quadrants (regions). We illustrated this example with the help of one particular wireless channel. The same method will be repeated using other wireless channels depending on the locations of the source–destination pairs.

B. Congestion-Aware Routing

As explained earlier in Section III-C, for the Hammer cache coherence- protocol-induced traffic patterns, multiple broadcast injections can occur within short windows. Such an

injection can specifically lead to heavy wireline traffic congestion caused by forwarding of multicast packets to their final destinations from the destination WIs. MALASH addresses this issue by employing a congestion aware routing. The WiNoC for all source– destination pairs within a region, two different paths are available (along the two orthogonal spanning trees). For a source (i) and destination (j) pair, the shortest among these two paths is referred as path P_{ij1} , while the other path is referred as P_{ij2} . A multicast message has a higher impact on the overall network performance than a unicast message. Hence, in MALASH, the multicast messages are treated with a higher priority over unicast messages. With this in mind, the congestion-aware routing for forwarding messages from WIs is implemented using the following rules.

1) For a message M with priority p, the path P_{ij1} is selected if P_{ij1} is free or occupied with a lower priority message. If P_{ij1} is occupied with a higher or equal priority message, then P_{ij2} is selected.

C. Network Coding

To efficiently distribute multiple multi cast messages (say two messages M1 and M2) that are received at the same time in an intermediate node, MALASH combines orthogonal paths with NC technique. In such a case, instead of forwarding both messages to the final destinations, the intermediate node creates a coded message M3 and forwards it to the final destination. The message M3 is created by XOR-ing individual bits of M1 and M2. At the final destinations, the desired message is recovered by XOR-ing one of the original messages with M3. NC is needed only when two messages M1 and M2 start their wireline distribution from different destination WIs in the region. This is a necessary condition to employ NC. In the case of M1 and M2 starting their wireline distribution from the same destination WI, the congestion can be avoided simply by forwarding M1 strictly along primary distribution links and forwarding M2 strictly along secondary distribution links.

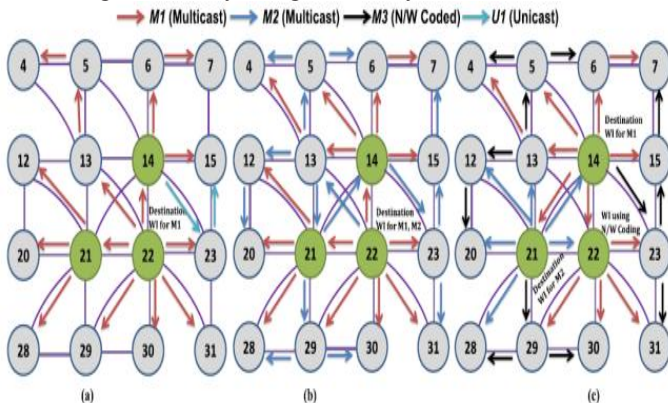


Fig4. Sample WiNoC multicast distribution scenarios in quadrant Q2.

In such a case, network congestion and intermediate queuing is avoided by employing NC. The coded message is forwarded along secondary distribution links from these intermediate nodes to the final destinations. At the same time, one of the original messages (both M1 and M2 are valid), is forwarded along the primary distribution links.

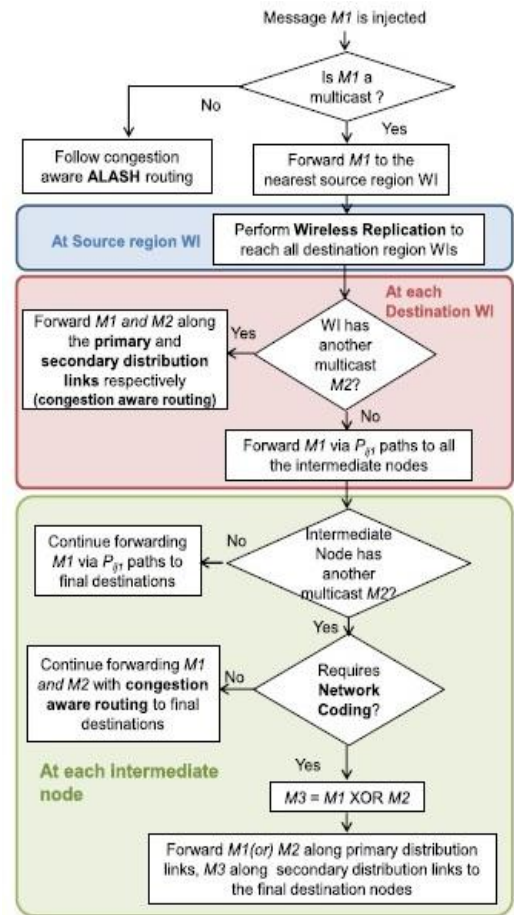


Fig5. Flowchart explaining MALASH routing.

D. Aggregation of Acknowledgements in WiNoC

Similar to the XY-tree-based multicast NoC (mentioned in Section IV-A), in WiNoC, the ACKs are aggregated at the intermediate nodes. Initially, each destination forwards its ACK message to the same intermediate node that forwarded the original multicast message. The ACKs are then aggregated at each region WI and are finally forwarded toward the source node. The use of ACK aggregation greatly reduces the overall traffic. For a 64-core system with unicast mesh, 63 ACK messages are returned to the source for each broadcast. In WiNoC, only four ACKs are forwarded to the source (one aggregated ACK for each quadrant).

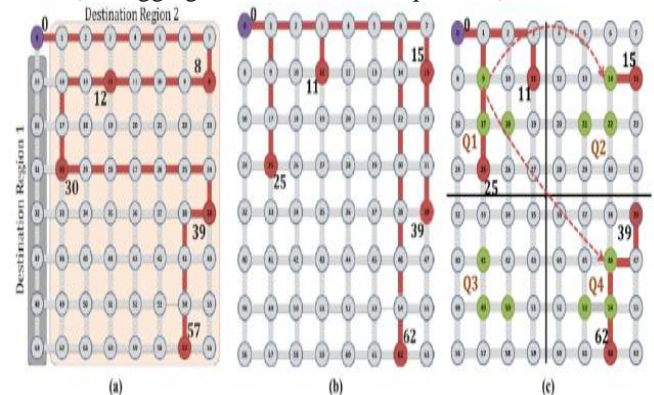


Fig6. Sample multicast scenarios.

Efficient of Multicast Wireless NOC Architecture by using Hammer Protocol

IV. SIMULATION RESULTS

We use the same five applications (CNL, FLD, FFT, LU, and RAD) discussed in Section III for the following analyses. We consider a 64-core ALPHA core system operating with a frequency of 2.5 GHz and having a die size of 20×20 mm². The memory system is composed of private 64-kbyte L1 instruction and data caches and one shared 16-Mbyte L2 cache (256-kbyte distributed L2 per core). All the processing cores and memory system are interconnected using an NoC fabric with 64 routers. For all the NoC architectures considered, we employ a generic three-stage router architecture modifying [38]. This architecture has three functional stages, namely, input arbitration, routing/switch traversal, and output arbitration with link traversal. Following routing, a multicast flit is replicated to all necessary output Virtual Channel buffers (VCs) simultaneously. The flits at output VCs are then handled by the output arbiter. The additional cycles required for congestion-aware routing and MAC protocol processing are accounted for while determining the network latency. Energy dissipation of the network routers, including the routing, MAC, and NC blocks, was obtained from the synthesized netlist using a 28-nm commercial Fully Depleted Silicon On Insulator technology by running Synopsys Prime Power. The total area overhead required in a WiNoC router to incorporate the WI, routing logic, NC, and MAC protocol units is 0.2514 mm² (4.02% area overhead for an NoC tile sized 2.5 mm× 2.5 mm). The router ports are provided with a buffer depth of two flits. The width of all wireline links is the same as the considered flit width (32 bits).¹ Each wireline is designed with the optimum number of uniformly placed and sized repeaters in the 28-nm technology node. The energy dissipation of the wireline links was obtained through Cadence SPECTRE simulations.

Project File:	Online_NOC.xise	Parser Errors:	No Errors
Module Name:	Top_module	Implementation State:	Placed and Routed
Target Device:	xc3e100e-5vq100	Errors:	No Errors
Product Version:	ISE 14.6	Warnings:	21 Warnings (1 new)
Design Goal:	Balanced	Routing Results:	All Signals Completely Routed
Design Strategy:	Xilinx Default (unlocked)	Timing Constraints:	All Constraints Met
Environment:	System Settings	Final Timing Score:	0 (Timing Report)

Device Utilization Summary				
Logic Utilization	Used	Available	Utilization	Note(s)
Number of Slice Latches	24	1,920	1%	
Number of 4 input LUTs	38	1,920	1%	
Number of occupied Slices	26	960	2%	
Number of Slices containing only related logic	26	26	100%	
Number of Slices containing unrelated logic	0	26	0%	
Total Number of 4 input LUTs	38	1,920	1%	
Number of bonded IOBs	25	66	37%	
IOB Latches	1			
Number of BUFMUXs	3	24	12%	
Average Fanout of Non-Clock Nets	1.85			

Fig7. Device utilization diagram.

We use GEM5, a full-system simulator, to obtain processor and network-level information[9]. We keep track of each message injected through the network interface module of GEM5 to extract the multicast traffic traces associated with real applications[4]. We use a modified GARNET interconnection network along with GEM5 to model the multicast supported NoCs in full-system simulations performed

to obtain execution times. The processor-level statistics generated by the GEM5 simulations are incorporated into multicore power, area, and timing to determine processor power values.

A	B	C	D	E	F	G	H	I	J	K	L	M	N
Device		On-Chip	Power (W)	Used	Available	Utilization (%)			Supply	Summary	Total	Dynamic	Quiescent
Family	Spartan3e	Clocks	0.000	7	1920	0.000			Source	Voltage	Current (A)	Current (A)	Current (A)
Part	xc3e100e	Logic	0.000	38	1920	2			Vccint	1.200	0.008	0.000	0.008
Packaging	vq100	Signals	0.000	65					Vccaux	2.500	0.008	0.000	0.008
Temp Grade	Commercial	I/Os	0.000	25	66	38			Vcc0v25	2.500	0.002	0.000	0.002
Process	Typical	Leakage	0.034										
Speed Grade	5	Total	0.034										
Environment		Thermal Properties	Effective TjA	Max Ambient	Junction Temp								
Ambient Temp (C)	25.0	(C/W)	49.0	(C)	83.4	25.9							
Use custom TjA?	No												
Custom TjA (C/W)	NA												
Airflow (LFM)	0												
Characterization													
PRODUCTION	v12.06-23-09												

Fig8. Power Report Diagram.

V. CONCLUSION

With an ever-expanding application pool, today's many core architectures are expected to handle a high diversity of on-chip traffic patterns. The Hammer cache coherence protocol is one such system-on-chip application that stresses on chip networks with heavy multicast injections. In this thesis, we presented a multicast-aware WiNoC architecture that can efficiently handle multicast-heavy cache coherence communications. Incorporated with a congestion-aware multicast routing and NC, WiNoC eliminates the initial and intermediate queuing latencies seen in conventional wireline mesh NOCs. Moreover, using wireless shortcuts, the WiNoC achieves significant reductions in network latencies leading to improved system performances. Compared with a many core system using the multicast-aware XY-tree mesh NoC, the WiNoC incorporated many-core platform achieves an average of 26% full system EDP improvement in the presence of Hammer protocol.

VI. REFERENCES

- [1] A. Karkar, N. Dahir, R. Al-Dujaily, K. Tong, T. Mak, and A. Yakovlev, "Hybrid wire-surface wave architecture for one-to-many communication in networks-on-chip," in Proc. IEEE Design Autom. Test Eur. Conf. Exhibit. (DATE), Mar. 2014, pp. 1–4.
- [2] D. Vainbrand and R. Ginosar, "Network-on-chip architectures for neural networks," in Proc. 4th ACM/IEEE Int. Symp. Netw.-Chip (NOCS), May 2010, pp. 135–144.
- [3] J.-Y. Kim, J. Park, S. Lee, M. Kim, J. Oh, and H.-J. Yoo, "A 118.4 GB/s multi-casting network-on-chip with hierarchical star-ring combined topology for real-time object recognition," IEEE J. Solid-State Circuits, vol. 45, no. 7, pp. 1399–1409, Jul. 2010.
- [4] Y. Xue and P. Bogdan, "User cooperation network coding approach for NoC performance improvement," in Proc. 9th ACM Int. Symp. Netw.-Chips, 2015, Art. no. 17.
- [5] Neuromorphic Computing: From Materials to Systems Architecture, Report of a Roundtable Convened to Consider Neuromorphic Computing Basic Research Needs, U.S. Dept. Energy, Gaithersburg, MD, USA, Oct. 2015.

- [6] N. E. Jerger, L.-S. Peh, and M. Lipasti, "Virtual circuit tree multicasting: A case for on-chip hardware multicast support," in Proc. 35th Int. Symp. Comput. Archit. (ISCA), Jun. 2008, pp. 229–240.
- [7] A. Ros, M. E. Acacio, and J. M. García, "Dealing with traffic-area tradeoff in direct coherence protocols for many-core CMPs," in Proc. Int. Workshop Adv. Parallel Process. Technol., Berlin, Germany: Springer, Aug. 2009, pp. 11–27.
- [8] P. Conway and B. Hughes, "The AMD opteron north bridge architecture," IEEE Micro, vol. 27, no. 2, pp. 10–21, Mar. 2007.
- [9] M. Lodde, J. Flich, and M. E. Acacio, "Heterogeneous NoC design for efficient broadcast-based coherence protocol support," in Proc. NOCS, May 2012, pp. 59–66.
- [10] S. Deb, A. Ganguly, P. P. Pande, B. Belzer, and D. Heo, "Wireless NoC as interconnection backbone for multicore chips: Promises and challenges," IEEE J. Emerg. Sel. Topics Circuits Syst., vol. 2, no. 2, pp. 228–239, Jun. 2012.