

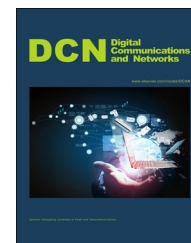
HOSTED BY



ELSEVIER

Available online at www.sciencedirect.com

ScienceDirect

journal homepage: www.elsevier.com/locate/dcan

$I(Re)^2$ -WiNoC: Exploring scalable wireless on-chip micronetworks for heterogeneous embedded many-core SoCs

Dan Zhao^{a,*}, Yi Wang^{b,1}, Hongyi Wu^a, Takamaro Kikkawa^c^aThe Center for Advanced Computer Studies, University of Louisiana at Lafayette, USA^bXingtera US, USA^cResearch Center for Nanodevices and Systems, Hiroshima University, Japan

Received 25 November 2014; accepted 24 January 2015

Available online 19 March 2015

KEYWORDS

Network-on-Chip;
RF interconnect;
Topology design;
Routing algorithm;
Microarchitecture design

Abstract

Modern embedded SoC design uses a rapidly increasing number of processing units for ubiquitous computing, forming the so-called embedded many-core SoCs (McSoC). Such McSoC devices allow superior performance gains while side-stepping the power and heat dissipation limitations of clock frequency scaling. The main advantage lies in the exploitation of parallelism, distributively and massively. Consequently, the on-chip communication fabric becomes the performance determinant. To bridge the widening gap between computation requirements and communication efficiency faced by gigascale McSoCs in the upcoming billion-transistor era, a new on-chip communication system, dubbed Wireless Network-on-Chip (WiNoC), has been proposed by using the recently developed RF interconnect technology. With the high data-rate, low power and ultra-short range interconnection provided by UWB technology, the WiNoC design paradigm calls for effective solutions to overhaul the on-chip communication infrastructure of gigascale McSoCs.

In this work, an irregular and reconfigurable WiNoC platform is proposed to tackle ever increasing complexity, density and heterogeneity challenges. A flexible RF infrastructure is established where RF nodes are properly distributed and IP cores are clustered. Consequently, a performance-cost effective topology is formed. A region-aided routing scheme is further designed and implemented to realize loop-free, minimum path cost and high scalability for irregular WiNoC infrastructure. To implement the data transmission protocol, the RF micro-architecture of WiNoC is developed where the RF nodes are designed to fulfill the functions of

*Corresponding author.

E-mail addresses: dzhao@cacs.louisiana.edu (D. Zhao), yi@xingtera.com (Y. Wang), wu@cacs.louisiana.edu (H. Wu), kikkawat@hiroshima-u.ac.jp (T. Kikkawa).¹Yi conducted this work during his Ph.D. study at University of Louisiana at Lafayette.

Peer review under responsibility of Chongqing University of Posts and Telecommunications.

<http://dx.doi.org/10.1016/j.dcan.2015.01.003>2352-8648/© 2015 Chongqing University of Posts and Telecommunications. Production and Hosting by Elsevier B.V. This is an open access article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>).

distributed table routing, multi-channel arbitration, virtual output queuing, and distributed flow control. Our simulation studies based on synthetic traffics demonstrate the network efficiency and scalability of WiNoC.

© 2015 Chongqing University of Posts and Telecommunications. Production and Hosting by Elsevier B.V.

This is an open access article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>).

1. Introduction

Ubiquitous embedded computing is playing a key role to drive the technological evolution in the near future. System-on-chip (SoC) design has become a cost-effective approach for embedded systems to integrate a number of heterogeneous IP cores, such as processors, DSPs, GPUs, FPGA modules and embedded memories. With continued scaling of microelectronics, future SoC devices may utilize several hundreds or even a thousand of processors yielding the so-called many-core SoCs (McSoC). Such nanoscale integrated many-core platforms pave the way to the ubiquitous era to accomplish such applications as mobile image/video processing, advanced driver assistance systems and other software intensive embedded systems e.g., cyber-physical systems. On one hand, orders of magnitude gains in price/performance, power and form factor efficiency can be realized using this emerging technology. On the other hand, ever increasing complexity, heterogeneity, performance, and productivity, put a heavy burden on the next-generation McSoC designs. The performance of McSoCs will be limited by the ability to efficiently interconnect heterogeneous IP cores to accommodate their communication requirements.

A scalable, cost-efficient, flexible and reusable on-chip communication infrastructure is becoming an enabling technology for this McSoC paradigm. The state-of-the-art shared-bus and point-to-point connections have been shown unable to supply nanoscale SoCs with both sufficient bandwidth and low latency under a stringent power consumption limitation [1]. Network-on-Chips (NoC) are emerging as an alternative communication platform for complex McSoCs to provide plug-n-play interconnection of heterogeneous IP cores. In the meantime, the speed improvements of silicon and SiGe bipolar transistors and MOS transistors have made the implementation of integrated circuits operating at ultrahigh frequency feasible. Consequently, it becomes possible to build RF circuits operating at 90 GHz with a cutoff frequency of 280 GHz at the 50 nm CMOS node [2]. In fact, a 900 μm monopole antenna on a high resistivity substrate has been proposed for 100 GHz intra-chip applications [3]. As CMOS and BiCMOS technologies improve, the cost of building on-chip antenna and radio frequency (RF) circuits will decrease dramatically, providing a greater freedom to use on-chip radios. As a result, a new RF/wireless interconnect technology has been investigated for future intra-/inter-chip communication, such as free-space transmission [4], guided-wave transmission [5,6], ultra wideband (UWB) [7,8] and direct near-field coupling [9].

Among them, the UWB interconnect (UWB-I) technology is introduced for high-data rate, low-power and short-range communication. Given its ultra-short transmission range and

the isolated communication environment, an extremely wide spectrum is available, leading to great potential of achieving supereminent data rate, ranging from 150 Gbps to 1.5 Tbps [10]. Based on UWB-I, the chip-based wireless radios can be deployed to replace the wires for increasing accessibility, improving bandwidth utilization, and eliminating delay and cross-talk noise in conventional wired interconnects. A revolutionary on-chip communication infrastructure, namely wireless network-on-chip (WiNoC) is thereby established for the communication among highly integrated heterogeneous IP cores with diverse functionalities, sizes and communication requirements in the nanoscale McSoCs [11]. WiNoC will provide higher flexibility, higher bandwidth, reconfigurable integration, and freed-up wiring when compared to NoC. With the uniqueness of UWB-I, the system architecture and data transmission protocol of WiNoC must adapt to the critical challenges posed by both large scale integration and small device geometries. The major contributions of this paper center on:

- Provide an insightful discussion of current state-of-the-art UWB-I technology and its recent technical advances and design impacts on many-core on-chip communication (Section 2).
- Present the design of application-specific communication infrastructure of WiNoC, specifically the RF node placement and core clustering to establish an irregular and reconfigurable wireless on-chip micronetwork (Section 4).
- Propose a hardware efficient region-aided routing scheme for loop-free shortest path multiple hop routing aiming at developing simple and compact protocol architecture for micro-scale communications (Section 5).
- Develop a RF node microarchitecture integrated with various mechanisms of region-aided routing, QoS enabled multi-channel arbitration, deadlock avoidant virtual output-queuing and contention-aware flow control to facilitate system implementation and performance demonstration (Section 6).

2. On-chip ultra wideband interconnect

RF/wireless interconnect technology has emerged over the last few years to address future global routing needs and provide performance improvement [6,7,12]. Among them, the introduction of Ultra-Wideband Interconnect (UWB-I) brings in new opportunity for low-power, short-range communication, thanks to its low-cost on-chip implementation, ultra-wide frequency band, and ultra-high data rate.

The UWB-I is based on transverse electromagnetic wave propagation with the use of on-chip antennas. Since its signal has very short pulse duration, high data rate (C) with constant signal-to-noise ratio (S/N) is achieved by increasing the bandwidth (B), following Shannon's capacity equation ($C = B \log_2(1 + S/N)$). It consumes low power in the range of a few milliwatts due to its very low duty cycle (typically $<0.1\%$). RF circuits can also be simpler since UWB signals are carrier free. Furthermore, pulse position modulation is typically used to modulate a sequence of very sharp Gaussian monocycle pulses. Each individual pulse is delayed in time depending on the data signal and the pseudo-random code (time-hopping sequence) assigned to the transmitter. Therefore, N "channels" are separated to support N users, which simultaneously communicate with their corresponding receivers on the *same frequency band*. The received signal and the template signal are automatically correlated during the demodulation of multiple access such as time hopping. Meanwhile, CMOS-integrated antennas such as various dipole antennas have been proposed for on-chip implementation.

One of the recent implementations of on-chip UWB-I as in Table 1 has achieved 1.16 Gbps data rate for single band at central frequency of 3.6 GHz in 0.18 μm CMOS technology [13,14]. A Si-integrated meander type dipole antenna has been implemented for 1 mm range data transmission at antenna area of $2.98 \times 0.45 \text{ mm}^2$. The area overhead for the transceiver design is 0.64 mm^2 . A 56 GHz architecture is described in [12] to enable inter-chip communications at greater than 10 Gbps. The system employs a self-locking receive section which attains 6.4 pJ/bit efficiencies in 40 nm CMOS.

Tab. 1 UWB interconnect in 0.18 μm CMOS.

| Technology node | 0.18 μm CMOS | | |
|-----------------------|--------------------------------------|----------|--------------------|
| Transmitter | Power 21.6 mW | Area | 0.1 mm^2 |
| Receiver | Power 40 mW | Area | 0.54 mm^2 |
| Modulation | On-off keying (OOK) | | |
| Data rate | 1.16 Gbps (single channel) | | |
| GMP central frequency | 3.6 GHz | | |
| Supply voltage | 1.8 V | | |
| Antenna type | Meander type dipole | | |
| Antenna | Area 1.8 mm^2 | Distance | 1 mm |
| Wireless channel | Additive white Gaussian noise (AWGN) | | |

According to ITRS [2], the cut-off frequency (f_T) and unity maximum power gain frequency (f_{max}) are projected to be at 400 GHz and 580 GHz respectively. As the maximum operating frequency of RF CMOS circuits increases with technology scaling, it is possible to implement RF circuits operating at ~ 20 GHz, achieving a data rate of ~ 20 Gbps (with the typical 1 bps/Hz bandwidth efficiency) in 32 nm technology. With multiple bands, the aggregate data rate can be further improved to above 1 Tbps. The characteristic antenna dimensions are proportional to the operation wavelength. Migration of short-range wireless communication to higher frequency allows smaller antennae and thus facilitates their on-chip integration. With such scaling, the required antenna and circuit areas will scale down. For example, at 400 GHz, the meander type dipole antennae will be only about 0.19 mm^2 in silicon, dramatically reducing the cost and increasing the flexibility of on-chip wireless interconnects. Moreover, the energy consumption per bit is expected to scale down too. As the technology node scales, UWB-I shows prominent scaling capability and its scaling trend is summarized in Table 2.

3. Architectural overview of WiNoC

Based on the UWB-I technology, a WiNoC platform is established for the communication among highly integrated heterogeneous IP cores with diverse functionalities, sizes and communication requirements in the nanoscale McSoCs. A WiNoC system architecture consists of two basic components, transparent network interfaces (TNI) and radio frequency (RF) nodes [11] as shown in Fig. 1. TNI serves as the interface between the IP cores and the WiNoC. Each TNI

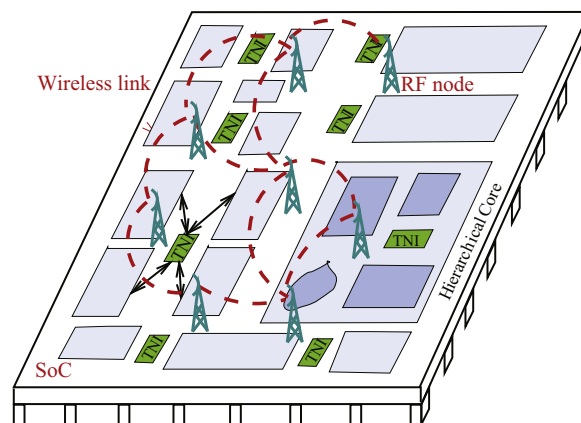


Fig. 1 A WiNoC-based McSoC paradigm.

Tab. 2 UWB interconnect scaling.

| Technology (nm) | 90 | 65 | 45 | 32 | 22 |
|---|-------|-------|-------|-------|------|
| Cut-off frequency (GHz) | 105 | 170 | 280 | 400 | 550 |
| Data rate per band (Gbps) | 5.25 | 8.5 | 14 | 20 | 27.5 |
| Dipole antenna length (mm) | 8.28 | 5.12 | 3.11 | 2.17 | 1.58 |
| Meander type antenna area (mm^2) | 0.738 | 0.459 | 0.279 | 0.194 | 0.14 |
| Power (mW) | 33 | 40 | 44 | 54 | 58 |
| Energy per bit (pJ) | 6 | 4.7 | 3.1 | 2.7 | 2.1 |

may be shared by a group of cores or dedicated to a core. It diminishes the heterogeneity of the cores and interacts with the network for packet assembly, delivery, and disassembly. TNI can efficiently decouple the designs of IP cores and WiNoC, thus supporting not only component reuse but also WiNoC plug-n-play. Through TNI along with the so-called point-to-point transfer protocol, each core may deem itself directly connected to other heterogeneous cores, as if the WiNoC is completely transparent. In this work, we use Virtual Component (VCI)-compatible interface for the implementation of TNI, where the VCI embedded split protocol can ease the round-trip latency constraints between a request-response pair.

The RF node is equipped with a radio-frequency interface (i.e., tiny low-power and low-cost UWB transceivers and antenna) for broadcast communication, and thus any nearby nodes within the transmission range may receive the signal. Each node implements the data transmission protocol stack including routing, channel arbitration, buffering and flow control for high-speed cost-efficient on-chip communication. In a WiNoC, a number of RF nodes are properly distributed according to various core functionality, non-uniform core size, and different traffic requirements between the cores, forming a multi-hop wireless micronetwork for packetized data communication. Each RF node communicates with each other through one or multiple “hops” across the network. With the multi-hop scheme, some nodes operate not only as a host but also as a router, forwarding signals to other nodes in the network. The collection of RF nodes connected by wireless links form the RF infrastructure which is a critical design issue in determining the feasibility and applicability of WiNoC communication architecture such as wireless routability, communication capacity, power and area cost.

4. WiNoC RF infrastructure design

RF infrastructure is a key design issue to establish an application-specific WiNoC topology. In contrast to many NoCs that rely on regular topology, we develop a flexible RF micronetwork infrastructure aiming to build any domain-specific topology for nanoscale McSoCs without any “wiring” concern. In such a flexible network, several IP cores may share one RF node and thus are grouped into a cluster or an IP core has its dedicated RF node. The cores in a cluster are hardwired to an RF node via TNIs and share it for data/control communication. Given the short distance between IP cores and their associated RF nodes, the hard-wired connection results in minimum routing cost and area overhead. In this work, we propose an efficient RF nodes distribution scheme that results in proper RF nodes placement and core clustering so as to form any irregular wireless micronetwork topology that meets the application needs by tuning the wireless accessibility. We name such an irregular and reconfigurable WiNoC, $I(Re)^2$ -WiNoC. The tunable accessibility of RF nodes and the reconfigurable integration capability render the $I(Re)^2$ -WiNoC a vital solution to on-chip communication.

4.1. RF node distribution

Problem statement: Given a McSoC model with a tentative floor plan, each core is abstracted as a vertex located at its

central coordinates (and a large complex core may be virtually partitioned into several subcores and each subcore is abstracted as a vertex). Given the maximum clustering range of R_c , within which a RF node can be hard-wired to its neighboring cores. The *problem of RF node distribution* is to find the minimum number of RF nodes that are optimally placed to ensure that all cores are within the maximum clustering range of at least one RF node. The cores are thus properly clustered, and the wireless transmission range is properly tuned to provide full wireless coverage on chip. Without loss of generality, this problem can be formulated into minimum geometric disk covering [15] in a way that a clustered wireless micronetwork is abstracted as a set of N_d disks $D = \{D_i | i \in N_d\}$. Each disk D_i is centered at a RF node with a radius of R_c that covers a set of N_c IP cores $C = \{C_j | j \in N_c\}$ embedded on the Euclidean chip plane. The objective is to minimize the cardinality of the disk cover N_d while ensuring full connectivity.

We construct a graph $G_c(V_c, E_c)$ to represent the SoC model. V_c is a set of vertices ($|V_c| = N_c$), each denoting an embedded core (without loss of generality, we assume that all cores require global communication via WiNoC). E_c is a set of edges, each connecting two vertices within a distance of $2R_c$. Assuming an optimally placed RF node can assist at least two IP cores (it becomes trivial if one RF node can assist only one core), i.e., a disk covers at least two vertices, we can always move the RF node such that there are two vertices on the circumference of the disk, while the disk covers the same set of vertices. For an edge connecting vertices a and b in Fig. 2 with Euclidean distance $l_{ab} < 2R_c$, there are at most two RF node placements such that a and b are on the circumference of the disks while the disks uniquely covers two sets of vertices. One disk cover is said to be unique only if there is at least one vertex different from any other disks. For two vertices c and d where $l_{cd} = 2R_c$, there is only one disk covering. For $|V_c| = N_c$ vertices on the plane to be covered by disks with radius R_c , there are at most $2|E_c|$ possible disks placements to be considered ($|E_c| \leq \binom{N_c}{2}$). The position of each disk is specified by its center, i.e., the coordinates of the RF node denoted by $D_i(X_i, Y_i)$ ($1 \leq i \leq 2|E_c|$). By formulating into minimum disk covering, this problem can be solved by a simple greedy set covering heuristic that

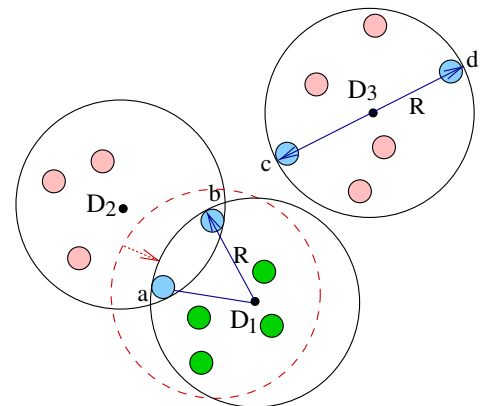


Fig. 2 The illustration of disk covering.

selects at each iteration the disk that covers the largest number of vertices [16].

4.2. Accessibility tunable topology formation

Once the RF nodes are properly placed, the nodes which fall within each other's transmission range are connected by wireless links, thus forming the WiNoC topology. Two factors affect the topology construction for a given SoC floor plan, namely, the core clustering range (i.e., R_c) and the wireless transmission range (i.e., R_t), which work interdependently to determine the number of RF nodes needed and their corresponding placement to cover the data communication of all IP cores. As the RF infrastructure counts for the major hardware overhead of WiNoC, minimizing the number of RF nodes can effectively reduce the area cost and power consumption. In the meantime, large clustering of cores may overload the associated RF node as a RF node can only assist one core at a time. Such performance-cost tradeoffs must be studied in the formation of topology. Our proposed topology formation scheme seeks the minimal number of RF nodes while ensuring workload balancing.

More specifically, a reference clustering range $R_{c,ref}$ bounded by the maximum number of RF nodes is set up first. Such a reference range is found to be the maximum possible disk radius which ensures the maximum coverage of non-overlapped disks on chip. To balance the traffic workloads, a 2-D IP core communication relationship matrix is constructed where each entry represents the communication requirement of a source-destination pair in a unit time period. For example, if the McSoC application targets at uniformly distributed traffic, all matrix entries are the same as the node injection rate divided by the number of IP cores. Once the clusters are formed, a 2-D RF communication relationship matrix is constructed accordingly by summing up the traffic demands of all IP cores within the same cluster. The overall traffic workloads of a RF node TL_{rf} are thus obtained, which is the sum of all incoming and outgoing traffic via the RF node. A high performance and cost effective topology is determined by iteratively running the RF node distribution algorithm by gradually increasing the clustering range R_c from $R_{c,ref}$ until TL_{rf} where each RF node hits the maximum traffic load threshold. Note that, as a RF node handles both local and global network traffic, we set the threshold below 0.5 where the network works mostly under the saturation point. Once the RF nodes are placed on-chip, the WiNoC topology is formed by searching for the least wireless transmission range which ensures full connectivity. Let S be the set with different disk covering and D be the disk set returned by the greedy algorithm, Algorithm 1 delivers a topology design to cover all $|C| = N_c$ IP cores on the chip plane. Fig. 3 illustrates a hypothetical WiNoC topology with 15 RF nodes formed for an example McSoC of 30 IP cores.

Algorithm 1. RF node distribution and topology formation algorithm.

```

1   $C \leftarrow \{C_i(x_i, y_i) | i \in |C|\};$ 
2   $S \leftarrow \emptyset;$ 
3   $D \leftarrow \emptyset;$ 

```

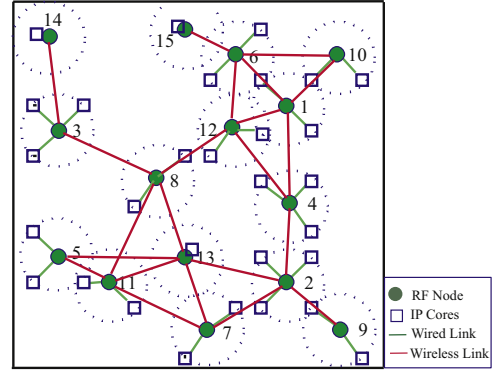


Fig. 3 Topology formation with enhanced disk covering.

```

4  for ( $i = 1 : |C| - 1$ ) do
5  | for ( $j = 2 : |C|$ ) do
6  | | if ( $dist(C_i, C_j) \leq 2R$ ) then
7  | | | obtain at most two different disk covering,
8  | | | |  $S \cup S_{XSX+1} \cup S_{XSX+2};$ 
9  | | | end
10 | | end
11 | end
12 while ( $C \neq \emptyset$ ) do
13 | select set  $S_i \in S$  that maximizes  $|S_i \cap S|;$ 
14 | balance workload;
15 |  $S = S - S_i;$ 
16 |  $D = D \cup S_i;$ 
17 | remove the vertices covered by  $S_i$  from  $S;$ 
18 | end
19 return  $D;$ 

```

5. Design of region aided routing for irregular WiNoC

Data transmission protocol is an integral part of WiNoC. To facilitate packetized data transmission in irregular WiNoC, an efficient routing scheme is further devised to address its unique characteristics that distinguish itself from all other networks developed so far. First, WiNoC has extremely limited resources (e.g., circuit area and power) available on chip for the implementation of routing protocol; second, it has fixed network topology with medium scale and density that is known prior to implementation; third, it has very low delay tolerance for supporting the real time applications. In this work, we develop a new distributed table routing scheme, namely *region-aided routing* to find the shortest multi-hop routing path while minimizing the hardware complexity of RF node protocol architecture. The overhead involved in routing mainly stems from the implementation of routing logic (i.e., the routing algorithm) and the maintenance of routing information (e.g., the neighbor list, network topology, and routing table). Aiming at hardware efficient design while achieving the quality of service required by a wide range of on-chip applications, we envision that this research will push the frontier of wireless technology to an extreme of simple and compact design for micro-scale communications.

5.1. Distributed region table routing

To ensure successful data delivery and low-cost implementation of routing decision, we propose a region-aided routing (RAR) scheme as illustrated in Fig. 4. The basic idea is that each RF node divides the entire chip plane, from its own perspective, into several rectangle zones, we name them as *regions*. Each region has a *region header* which is the neighbor of the decision node. The *region border* is defined by the minimum and maximum XY-coordinates among all nodes within the region. Thus the four border coordinates of $[X_{min}, X_{max}, Y_{min}, Y_{max}]$ of a region will be stored at the comparator array (as explained in Section 5.3) to uniquely identify this region. If N_r regions are delimited for each RF node, an array of $N_r \times 4$ border coordinates is maintained at the RF node. The region borders are efficiently utilized to identify the region where the destination node is located. The regions therefore should represent the compressed topology information to make globally optimized routing decision. A region can be quickly identified by comparing the destination address $[D_x, D_y]$ contained in the packet header to the region borders. If $X_{min}^i \leq D_x \leq X_{max}^i$ and $Y_{min}^i \leq D_y \leq Y_{max}^i$, destination D is found to be located in region R_i and the packet will be forwarded to the header of region R_i . For example, an RF node A wants to send packet m to its destination node D , it forwards m to its neighbor C which delimits the unique region where D is located. We repeat this process until D is eventually reached.

5.2. Region delimitation

If the regions can be properly delimited in a region-aided routing scheme, the routing decision can be tightly controlled within the optimal solution. In this section, we design an efficient region delimitation (RD) algorithm to define the regions. Since the network topology is determined a priori, this algorithm runs off-line and the regions as indicated by region borders are recorded for each node. Some critical concerns should be taken to orient the delimitation of regions. First of all, a region should reflect the compressed topology information that gives us the ability to calculate the shortest path to a destination. Second, a destination node can only appear in one region and there is no mutual overlapping between regions. Last but not least, the number of regions at each decision node should be minimized to reduce the hardware cost. The RD algorithm consists of three major steps, namely, region pre-

identification, mutual overlapping elimination, and region border extension.

Region pre-identification: The best routing path between a source-destination pair should be the one with the minimum path cost which is evaluated by various metrics, such as hop count, expected transmission time, and path interference. For WiNoC such a static network, these metrics can be combined together to estimate the cost. In this paper, we only consider hop count as the routing metric, however the algorithm can be easily extended to incorporate various metrics.

We consider a WiNoC topology $I = (N, L)$ consisting of a finite set of RF nodes N and a finite set of wireless links L . Each wireless link is associated with a link cost. The minimum cost path between any source-destination pair is found by running the shortest path algorithm (e.g., Dijkstra's algorithm). We obtain a routing decision tree for any node to any other destinations as shown in Fig. 4(b). An $N \times N$ path cost matrix (PCM) is constructed to record the minimum cost (hop count) for all source-destination pairs in a WiNoC. For each decision node S , we may derive a dedicated $N_{neb} \times N_{dst}$ path cost matrix (namely $N2D$ matrix) including all path entries from any of its neighbor to any destination node (excluding S and the N_{neb} neighbors). Each entry $N2D(i, j)$ represents the minimum path cost from the i th neighbor to destination node D_j . For a column j , we can always find an entry with the minimum path cost within the column which indicates that destination D_j can be minimum-cost reached by S 's i th neighbor. Thus a region is created that contains D_j while neighbor N_i is defined as the header of this region. In this way, we go through every column to identify the region header for each destination, and determine accordingly either to create a new region when encountering a new header or to expand an existing region for the same header by adding more nodes into it. The region border is updated by the minimal and maximal XY coordinates among all nodes in the current region. Note that, unless a column contains exactly one minimum cost entry, additional effort is required to ensure that one destination is contained only in one region.

For efficient region delimitation, two types of region headers are defined, irreplaceable or replaceable region headers. If a column in an $N2D$ matrix contains a single minimum entry, i.e., a destination can only be minimum-cost reached by one neighbor, such a neighbor is named as an *irreplaceable region header* while this destination is defined as the *identifier of the region*, and this region must exist. Otherwise, the region is replaceable and may not be

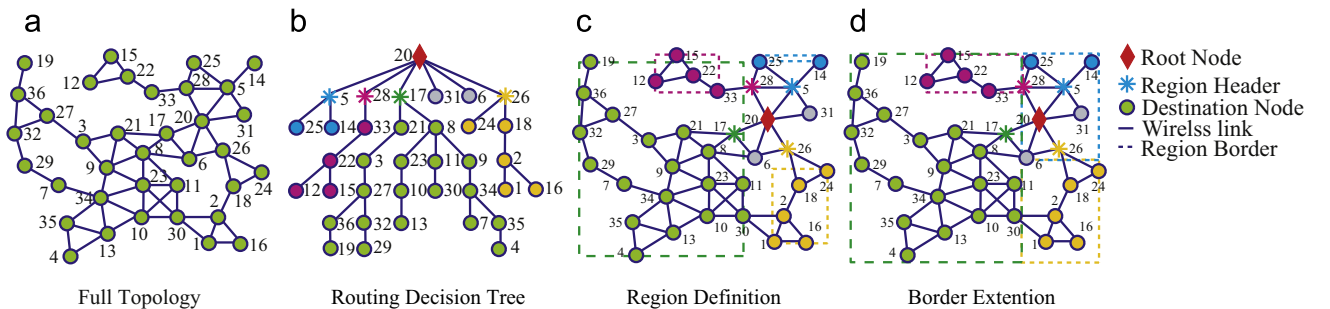


Fig. 4 The illustration of region aided routing.

necessary to be created. In order to minimize the number of regions while keeping the region area as small as possible for accurate region delimitation, an efficient way is to first find all region identifiers to determine the corresponding irreplaceable regions. Then we define the regions for the remaining unidentified destination nodes. There are three cases.

- If a node can be minimum-cost reached by either an irreplaceable region header or one or several replaceable region headers, the node is always allocated in the irreplaceable region.
- If a node can be minimum-cost reached by more than one irreplaceable regions headers, the node is allocated in the irreplaceable region that results in minimum region area expansion. If the increased areas are the same, the node will belong to the closest region header from S.
- If a node can be minimum-cost reached by more than one replaceable regions headers (the replaceable regions are empty initially), the furthest neighbor will be chosen with the potential to cover more such unidentified destinations. If those regions are not empty, the one that covers more destinations will be chosen.

Mutual overlapping elimination: It becomes essential to tackle the region overlapping problem, as the rectangle border may represent an enlarged area than the actual geographic occupation of the nodes in the region. Overlapping of different regions causes confusion for correct routing decision. There are two types of overlapping, simple overlapping and mutual overlapping.

- If the nodes in the overlapping area belong to only one of the overlapped regions, *simple overlapping* occurs.
- If the nodes in the overlapping area belong to different overlapped regions, *mutual overlapping* occurs.

Region overlapping can be easily detected by introducing an $N_{ov} \times N_{ov}$ region overlapping matrix (OVM), where N_{ov} is the number of overlapped regions. If a node belonging to region R_j also falls within the border of region R_i , $OVM(i, j)=1$, otherwise $OVM(i, j)=0$. A simple overlapping is found if $OVM(i, j)=1$ while $OVM(j, i)=0$. A mutual overlapping is detected if $OVM(i, j)=OVM(j, i)=1$ as shown in Fig. 5(a).

The problem of simple overlapping can be solved by priority decoding the decision making of overlapped regions. For example, several consecutive simple overlapping form a loop as shown in Fig. 5(b). We may derive a decision truth table (DTT) from the OVM simply by converting (i) a '1' in OVM to a '0' in DTT, (ii) a 'do not care' in OVM to '1' in DTT, and (iii) a '0' in OVM to a 'do not care' in DTT. The region thus can be uniquely defined based on the truth table. It is more complicated in mutual overlapping. The first measurement that can be taken is to check whether the nodes in the overlapping area can also be minimum-cost reached by other region headers. If so, these nodes can be reassigned to other regions. By proper reassigning, all the nodes in a mutual overlapping will belong to the same region or the regions can be redelimited to eliminate overlapping. In the worst case, new regions have to be created for unsolvable overlapping, each containing one node from the overlapping area with its original region header as the region header of the newly created region. The original regions will be redelimited. For the example WiNoC, we get a total of 4 regions as shown in Fig. 4(c) with mutual overlapping being eliminated and a simple overlapping being handled by the region decoder.

Border extension: The region border can be extended to further reduce the hardware cost without hurting the routing decision. For example, in order to identify the region where a destination node is located, we need $4N_r$ comparators. Even with the number of regions as small as 4, we still need 16 comparators. We observe that if the regions defined for an RF node share the same X or Y border coordinate, they may share the comparators as well. A refinement step is pursued to extend the original borders to reach the chip edge or meet the other borders without introducing any mutual overlapping problem. Clearly, the right vertical border or bottom horizontal border might be extended through a decreasing (in terms of coordinate) extension and the left vertical border or top horizontal border can be extended through an increasing extension. We restrict the extension rule as follows. If the borders can be extended to the chip edge, the border coordinates are set to the same as the chip edge; otherwise an extension is performed in a way that the region borders of two opposite regions expand towards each other by 1 at a time. With border extension, the required comparators are reduced from 16 to 4 as illustrated in Fig. 4(d), which leads to a 75% hardware cost reduction.

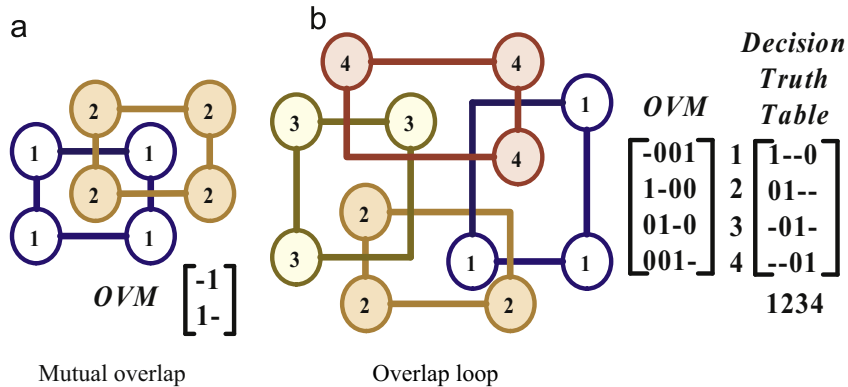


Fig. 5 Region overlap: (a) mutual overlap, (b) overlap loop.

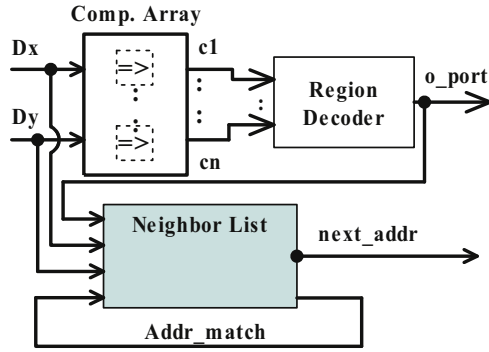


Fig. 6 Hardware architecture of RAR routing decision logic.

5.3. Routing decision logic implementation

The hardware implementation of the routing decision logic is illustrated in Fig. 6, consisting of three major functional blocks of comparator array, region decoder, and neighbor list.

The neighbor list of a node contains a sequence of addresses of all its immediate neighbors. Upon receiving a data packet, the RF node reads in the destination address (DA) contained in the packet header. If the destination address matches with an entry in the neighbor list, i.e., the destination is a neighbor of the node, the next-hop address is thus found that is the same as the DA. If the destination address is not in the neighbor list, the RF node searches for the next-hop address through the region decoder. Taking the destination address as inputs, the comparator array compares DA with the pre-stored region borders to identify the region. The number of comparators needed is determined by the number of different coordinates of region borders except for those at the chip edge. If the destination address falls within the border ($X_{min} \leq D_x \leq X_{max}$ and $Y_{min} \leq D_y \leq Y_{max}$) of a region, the comparison result for the corresponding region is '1', otherwise '0'. The comparison results are then fed into the region decoder that implements DTT using combinational logic and the neighbor ID (o_port) is obtained. The neighbor ID is used to access the neighbor list to obtain the neighbor address so that the address of the next hop can be inserted into the outgoing packets. Based on the border definition and DTT, the VHDL code for each node can be automatically generated from the proposed RAR algorithm. After RTL synthesis, the maximum area required for the array of comparators is 40 gates, and for the neighbor list is 161 gates.

6. RF node microarchitecture design

To facilitate WiNoC system implementation and performance demonstration, the RF node microarchitecture is developed which implements the mechanisms for routing, channel arbitration, buffering and flow-control. As shown in Fig. 7, the routing decision logic (RDL) implements the proposed region aided routing scheme to define the path a packet routes from a sender to a receiver. The channel arbiter implements a multi-channel synchronous and distributed arbitration scheme. The flow control logic implements a distributed prioritized flow control strategy to deal

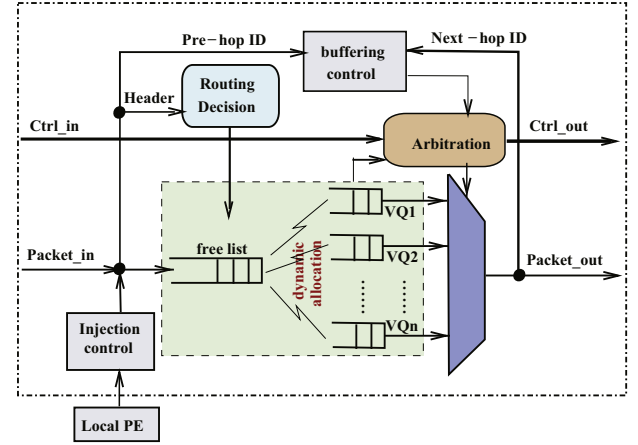


Fig. 7 Illustration of RF node microarchitecture.

with the data traffic on a channel and inside a RF node. The buffers are used to store packets when a channel is busy to ensure lossless data transmission. To minimize buffer sizing, a virtual output-queuing scheme is applied to efficiently manage buffering. Due to limited processing resources, a RF node can only receive or send a packet at a time which greatly simplifies the intra-node routing and buffering.

6.1. Multi-channel arbitration

A synchronous and distributed arbitration scheme (i.e., SD-MAC) [11] has been proposed to resolve single channel access contention. Similar to SD-MAC, we propose a multi-channel counterpart to ensure collision-free transmission on a multi-channel WiNoC, dubbed multichannel SD-MAC (*mSD-MAC*). Again, *mSD-MAC* is based on synchronized data frames, where each frame consists of two intervals: the contention interval and the data interval. The right to access the data channel is granted by the negotiation in the control channel. Such negotiation is carried out by a binary contention mechanism in replace of binary countdown handshaking in SD-MAC. As the channels are coded in a way of collision avoidance, *mSD-MAC* is much simplified.

More specifically, in the contention interval, a node (say, X) that has data to send generates a random contention number with k bits. To reduce the contention delay, multiple contention bits can be sent simultaneously. The contention numbers received from multiple senders will be compared once at the receiver and the sender with the largest contention number will be the winner and be granted the channel access to the receiver. Such contention procedure is simple, but paid at the cost of wired controlling overhead. The *mSD-MAC* ensures 100% collision free while retaining the same features of high-efficiency, simplicity, robustness, fairness, and QoS capability as SD-MAC. Moreover, *mSD-MAC* greatly improves the network capacity and enhances end-to-end performance by ensuring more parallel transmissions than SD-MAC.

6.2. Virtual output queuing

In order to differentiate packet flows, we use an output queuing architecture. More specifically, the buffers are

organized in a virtual output queuing strategy with a dynamically shared linked buffer, where each virtual queue corresponding to one neighboring node temporarily stores the packets going to the neighbor. In addition, a shared buffer stores the incoming packet before its next-hop is determined by the RDL and consequently the packet is dynamically linked to the corresponding virtual output queue. To minimize the buffering cost, we determine the smallest buffer size for ensuring the transmissions free of deadlock. With the N_c credit-based backpressure scheme, only N_c packets can be delivered from a node to any one of its downstream neighbors. The worst case scenario happens when multiple flows crossing at a node contend for the same next-hop, requiring $(N_{nei} - 1) \times N_c$ buffer units, where N_{nei} denotes the number of neighbors. Similarly, the node itself may also serve as the downstream node of its neighbors, calling for again a total of $(N_{nei} - 1) \times N_c$ buffer units. Besides, N_c buffer units are needed for the self-injected packets, summing up to a total of $(2N_{nei} - 1) \times N_c$ buffer units required to guarantee deadlock freedom.

6.3. Distributed flow control

An efficient flow control and congestion alleviation strategy is essential to improve WiNoC end-to-end performance due to the severe congestion resulted from the close interaction between wireless channel contention and traffic overflow. A customized distributed flow control and congestion control strategy as in [17] is seamlessly integrated into *mSD-MAC* that involves multiple mechanisms, such as fast forwarding by prioritized channel arbitration and credit-based backpressure congestion control. The fast forwarding scheme grants higher priority to a downstream node which just receives a packet to increase its winning probability for the new round of channel arbitration. To resolve receiver-based channel contention, a receiver-prioritized forwarding scheme is developed and incorporated into *mSD-MAC* scheme by assigning prioritized contention numbers in binary countdown. The N_c -credit based backpressure congestion control blocks the up-stream node from continuously forwarding packets unless the previous upstream packets have been released to or consumed by the downstream node, thus resolving intra-flow contention.

In a nutshell, when a packet arrives at a RF node, the packet header which contains the destination address is sent to RDL to determine the next-hop and to generate a new header while the data payload is sent to the dedicated virtual queue combined with the new header. Upon the round-robin scheduling, one of the packets waiting at the head of the virtual queues will be sent the channel access request, i.e., the *mSD-MAC* contention number to its corresponding down-stream node. In the meantime, a node will receive multiple requests from its neighbors and the node's channel arbiter will grant authorization to one of requesters. As a result, a channel is established between the winning sender and the receiver and the chosen packet will be transmitted to its next-hop immediately. Note that, limited number of channels are distributed beforehand to enhance transmission concurrency while providing sufficient channel bandwidth. In addition, the packet queuing process for the next packet including intra-node routing, buffering

and output scheduling can be launched in parallel with the channel competition of the current packet.

7. Simulation study and performance evaluation

A WiNoC simulator with identical and omnidirectional radio range is built up to cover the communication among the IP cores within an McSoC model. Specifically, the IP core placements are randomly generated on the chip plane. The RF nodes are optimally distributed and consequently, the wireless topology is formed by running the optimization algorithm as discussed in Section 4. The RF node micro-architecture developed in Section 6 is implemented at each node that runs our proposed data transmission protocol for successful multi-hop data/control communication in irregular WiNoC.

7.1. Routing efficiency and routing cost evaluation

In our simulation model, 100 randomly generated network topologies with the number of RF nodes ranging from 16 to 116 are used to evaluate the performance of *RAR* algorithm. The routing decision on all these topologies is 100% loop-free. The total hop count for all source-destination pairs on any topology is exactly the same as the hop count obtained from the shortest path algorithm. The performance of *RAR* is guaranteed by properly grouping the destination nodes into regions based on the minimum path cost.

Proposition 1. *RAR based routing decision achieves minimum path cost.*

Proof. It is clear that any one-hop decision is correct. Let C_{SD} be the minimum path cost for an n -hop path ($n > 1$) from node S to node D based on the predefined routing metric. Let C_{SD_R} be the path cost by running *RAR* algorithm, where $C_{SD_R} = C_{SD} + c$ ($c > 0$). There is at least one intermediate node I along the path that satisfies $C_{SI_R} = C_{SI}$, and $C_{ID_R} = C_{ID} + c$. When node I makes routing decision according to the routing decision tree obtained from the shortest path algorithm, it must ensure that $C_{ID_R} = C_{ID}$, thus $c = 0$, and $C_{SD_R} = C_{SD}$. \square

Proposition 2. *RAR based routing decision guarantees loop-free.*

Proof. Since the minimum cost from the node to itself is infinite, based on Proposition 1, if there exists a path loop from S to S , $C_{SS_R} = C_{SS} = \infty$. For any intermediate node I along this path loop, $C_{SI_R} + C_{IS_R} = C_{SI} + C_{IS} = \infty$. Either C_{SI} or C_{IS} should be infinite, which is impossible for a connected network. We may simply conclude that a path built by *RAR* is loop-free. \square

The performance of *RAR* routing scheme is compared with table driven routing (*TDR*) [18] and location based routing (*LBR*) [19]. *TDR* relies on a central node to maintain a routing table and to make the routing decision. This approach involves high overhead stems from a centralized routing algorithm, large packet size (the entire path information should be contained in packet header), and

large routing table, thus leads to low scalability. *LBR* reduces routing algorithm complexity by making routing decision at the intermediate node based on the geographic distance to the destination. Neither the routing table nor the network topology needs to be maintained. However, *LBR* results in much higher hop count than the ideal case due to localized decision. Further, *LBR* may fail at concave nodes, thus partial flooding should be performed to ensure 100% delivery. Flooding will limit the network efficiency since it occupies the capacity of the whole network. Moreover, due to the uncertainty accompanied with flooding, the implementation complexity is high. The three types of routers are implemented and their performance are compared in terms of delay and area cost as illustrated in Fig. 8. In the experiments, all topologies use 10 bits address to represent the node location. The designs are synthesized through Mentor Graphics LeonardoSpectrum level 3 based on the SCL05u technology. For fair comparison, the timing constraint for synthesis is set to ensure that the decision logics meet their minimum delay requirements. From the comparison we can see that *RAR* achieves the best performance while the scalability of the network is ensured.

7.2. $I(Re)^2$ -WiNoC network performance evaluation

The experiments are carried out to evaluate the network performance of the proposed $I(Re)^2$ -WiNoC under three

synthetic traffics: uniform, one-hot spot and two-hot spot. Under uniform traffic pattern, each RF node has the same chance to receive a packet. Under one-hot (or two-hot) spot traffic pattern, one randomly selected node (or two nodes) will accept 50% (or 1/3 per node) of the total generated traffic while the remaining traffic is uniformly distributed among all other nodes. During the simulation, the packets are injected in frame times where the unit of injection rate is determined by the number of packets per node per frame time. When a packet is generated at a node, its destination is randomly assigned. Each RF node is associated with a random traffic generator that generates the above three different traffic patterns. After a WiNoC system is warmed up by running 1000 frame times, 20,000 packets are generated under certain traffic patterns and injection rates. Once all packets reach their destinations, the WiNoC system performance is evaluated in terms of throughput and delay under a 10 Gbps network bandwidth.

The performance improvement is apparent when shifting from a single channel WiNoC to a multi-channel network. As shown in Fig. 9a, the concurrency level of multi-channeling is more than doubled when compared with the single channel alternative. As a result, the network capacity doubles under uniform traffic. Under hot-spot traffic, the performance bottleneck is the processing speed of the hot spots. Thus, the throughput difference between the multi-channel and single channel configurations tends to shrink. As observed, the throughput is comparable under both one-hot and two-hot spot traffics. The comparison of latency in

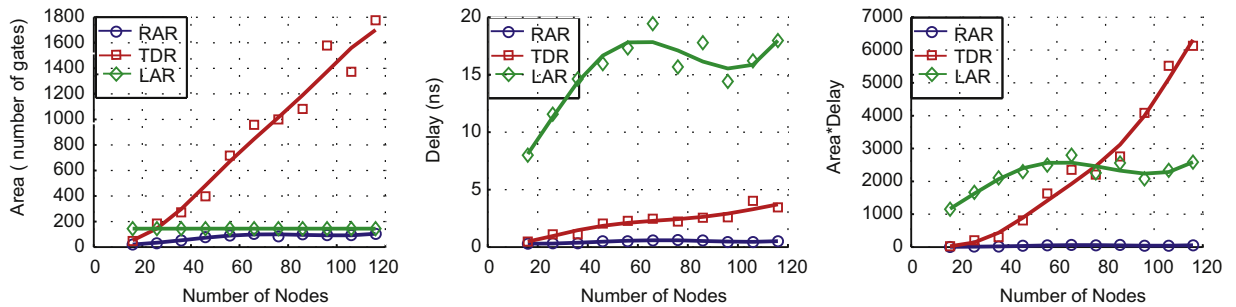


Fig. 8 Performance comparison of *RAR*, *TDR* and *LBR*.

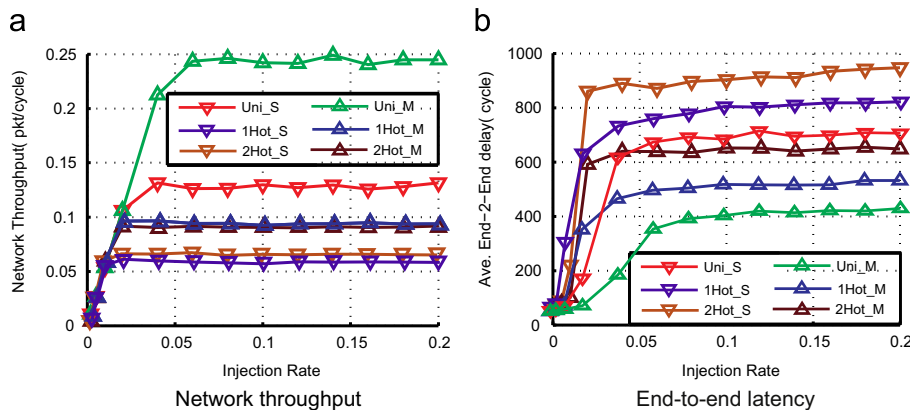


Fig. 9 Single vs. multi-channel network performance comparison under different traffics: (a) network throughput; (b) end-to-end latency.

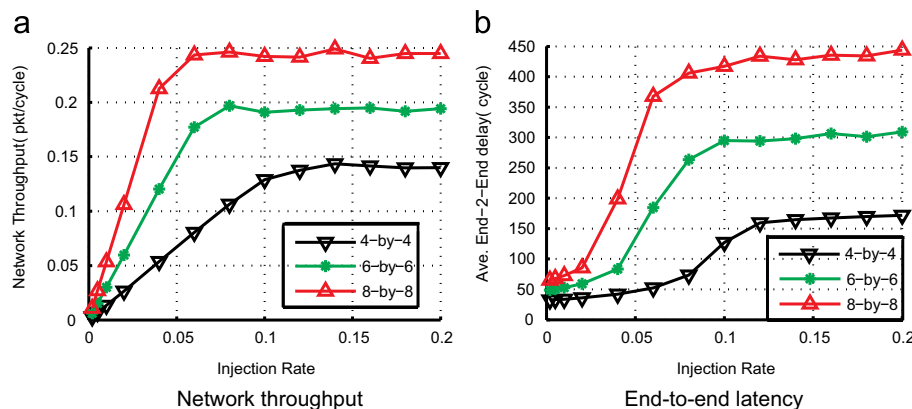


Fig. 10 Network performance comparison under different network scales: (a) network throughput; (b) end-to-end latency.

Fig. 9b again demonstrates that the multi-channel configuration brings in performance improvement over the single channel counterpart. The frame-based channel arbitration interleaves contention handshaking and data transmission and thus ensures high utilization of wireless bandwidth, leading to great end-to-end performance even for large irregular WiNoCs. We evaluate the scalability of WiNoC under 3 different network sizes: 4×4 , 6×6 and 8×8 . As we can see from Fig. 10, the network throughput scales up while the latency scales down when the network size increases.

8. Conclusion

The complexity and heterogeneity nature of nanoscale McSoCs promote the idea of cost-effective and power-efficient wireless NoC for the on-chip communication among diverse, mixed technology IP cores. This paper centers on the design of WiNoC RF infrastructure with the features of reconfigurable integration and wireless tunable accessibility. This paper has further presented the design and hardware implementation of a loop-free, minimum cost guaranteed distributed table routing scheme for irregular WiNoC. An efficient and low-cost RF node microarchitecture was implemented, aiming to devise simple and compact RF nodes for establishing WiNoC under extremely limited resources. The RF nodes are properly distributed and wireless topology is optimally built up by striving for balance between performance and cost. The simulation study demonstrated that both the performance and scalability of the network are ensured and low-cost and high-efficiency design of WiNoC RF infrastructure is achievable. We have demonstrated that the heterogeneous nature of on-chip cores and energy efficiency requirements of high performance embedded ubiquitous computing call for WiNoC paradigm.

References

- [1] R.H. Havemann, J.A. Hutchby, High-performance interconnects: An integration overview, *Proc. IEEE* 89 (5) (2001) 586-601.
- [2] International technology roadmap for semiconductors—interconnect, (<http://www.itrs.net/Links/2011itrs/2011Chapters/2011Interconnect.pdf>), 2011.
- [3] K.T. Chan et al., Integrated antennas on Si with over 100 GHz performance, fabricated using an optimized proton implantation process, *Microw. Wirel. Compon. Lett.* 13 (2003) 487-489.
- [4] B.A. Floyd, C.-M. Hung, K.K. O, Intra-chip wireless interconnect for clock distribution implemented with integrated antennas, receivers, and transmitters, *IEEE J. Solid-State Circuits* 37 (5) (2002) 543-552.
- [5] M.F. Chang et al, RF/wireless interconnect for inter- and intra-chip communications, *Proc. IEEE* 89 (4) (2001) 456-466.
- [6] M.F. Chang, E. Socher, S.-W. Tam, J. Cong, G. Reinman, Rf interconnects for communications on-chip, in: *Proceedings of International Symposium on Physical Design*, 2008, pp. 78-83.
- [7] S. Watanabe, K. Kimoto, T. Kikkawa, Transient characteristics of integrated dipole antennas on silicon for ultra wideband wireless interconnects, in: *Proceedings of Antennas and Propagation Society International Symposium*, 2004, pp. 2277-2280.
- [8] M. Sun, Y.P. Zhang, G.X. Zheng, W.-Y. Yin, Performance of intrachip wireless interconnect using on-chip antennas and uwb radios, *IEEE Trans. Antenna Propag.* 57 (9) (2009) 2756-2762.
- [9] N. Miura, D. Mizoguchi, T. Sakurai, Analysis and design of inductive coupling and transceiver circuit for inductive inter-chip wireless superconnect, *IEEE J. Solid-State Circuits* 40 (4) (2005) 829-837.
- [10] K.K. O, et al., On-chip antennas in silicon ICs and their application, in: *Proceedings of International Conference on Computer Aided Design*, 2005, pp. 979-984.
- [11] D. Zhao, Y. Wang, Design and synthesis of a hardware-efficient collision-free QoS-aware MAC protocol for wireless network-on-chip, *IEEE Trans. Comput.* 57 (9) (2008) 230-245.
- [12] K. Kawasaki, et al., A millimeter-wave intra-connect solution, in: *Digest of International Solid-State Circuits Conference*, 2010, pp. 413-415.
- [13] T. Kikkawa, P.K. Saha, N. Sasaki, K. Kimoto, Gaussian mono-cycle pulse transmitter using $0.18\mu\text{m}$ cmos technology with on-chip integrated antennas for inter-chip uwb communication, *IEEE J. Solid-State Circuits* 43 (5) (2008) 1303-1312.
- [14] W.M.N. Sasaki, K. Kimoto, T. Kikkawa, A single-chip ultra-wideband receiver with silicon integrated antennas for inter-chip wireless interconnection, *IEEE J. Solid-State Circuits* 44 (2) (2009) 382-393.
- [15] D.S. Hochbaum, W. Maass, Approximation schemes for covering and packing problems in image processing and VLSI, *J. ACM* 32 (1) (1985) 130-136.

-
- [16] M.R. Garey, D.S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, Macmillan Higher Education Publisher, 1979.
- [17] Y. Wang, D. Zhao, Distributed flow control and buffer management for wireless network-on-chip, in: Proceedings of International Symposium on Circuits and Systems, 2009, pp. 1353-1356.
- [18] D. Zhao, S. Upadhyaya, M. Margala, Design of a wireless test control network with radio-on-chip technology for nanometer systems-on-chip, *IEEE Trans. Comput-Aided Des. Integr. Circuits Syst.* 25(6).
- [19] D. Zhao, Y. Wang, MTNET: design and optimization of a wireless SoC test framework, in: Proceedings of IEEE SoCC, 2006.