Survey on Real-Time Networks-on-Chip

Salma Hesham, Jens Rettkowski, Diana Goehringer, and Mohamed A. Abd El Ghany

Abstract—Multi-Processor Systems-on-Chip (MPSoCs) have emerged as an evolution trend to meet the growing complexity of embedded applications with increasing computation parallelism. Particularly, real-time applications make out a significant portion of the embedded field. Networks-on-Chip (NoCs) are the backbone of communications in an MPSoC platform. However, the use of NoCs in real-time systems imposes complex constraints on the overall design. This paper discusses the challenges faced, when designing NoCs for real-time applications. Contributions in this area are surveyed on the level of guaranteed Quality-of-Service (QoS) support, adaptivity, and energy efficient techniques. Furthermore, the evaluation methodologies and experimental performance measurements of real-time NoCs are examined. This survey provides a comprehensive overview of existing endeavors in real-time NoCs and gives an insight towards future promising research points in this field.

Index Terms—Multi-core/single-chip multiprocessors, on-chip interconnection networks, power management, real-time and embedded systems

1 Introduction

MPSoCs have proved their merit as the typical platform for embedded and cyber physical systems for their computational power and unique energy efficiency provided in a compact design. Today's MPSoCs such as the Knights Core with 50 cores from Intel [1] hold tens of sophisticated PEs or up to hundreds of simple cores. Future MPSoCs are expected to embrace thousands of processing cores within the next few years [2]. Networks-on-Chip (NoCs) [3] have been proposed as the solution for the increasingly complicated communication requirements of MPSoCs and future Many-Cores. NoCs offer a scalable and flexible communication infrastructure with highly supported modularity and powerful performance [4], [5]. Fig. 1 shows an example for a NoC-based heterogeneous MPSoC architecture, connecting IP cores, DSPs, CPUs, and memory blocks, suitable for real-time multimedia processing applications. Real-time applications (RTAs) make out a significant portion of the embedded field. However, the use of NoCs in real-time systems (RTSs) imposes complex constraints. These constraints are implied by the fact that real-time communication performance relies on both the logical result as well as the completion time bound. A data packet received by a destination too late is useless or even causes severe consequences. Accordingly, both computation and communication between the components must complete within certain deadlines.

- *S. Hesham is with the German University in Cairo, Cairo 11835, Egypt and the Ruhr-University Bochum, Universitaetstr. 150, Bochum 44780, Germany. E-mail: salma.hesham@guc.edu.eg.*
- *M. A. Abd El Ghany is with the German University in Cairo, Cairo 11835, Egypt and the Technical University of Darmstadt, Merckstr. 25, Darmstadt 64283, Germany. E-mail: mohamed.abdel-ghany@guc.edu.eg.*
- *J. Rettkowski and D. Goehringer are with the Ruhr-University Bochum Universitaetstr. 150, Bochum 44780, Germany. E-mail: {jens.rettkowski, diana.goehringer}@rub.de.*

Manuscript received 10 Dec. 2015; revised 25 Oct. 2016; accepted 26 Oct. 2016. Date of publication 1 Nov. 2016; date of current version 12 Apr. 2017. Recommended for acceptance by C. Cumplido.
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier no. 10.1109/TPDS.2016.2623619

This paper discusses the implications of RTAs on NoC design. Existing contributions covering guaranteed Quality-of-Service (QoS) support, adaptivity, and energy efficiency aspects within real-time NoCs (RT-NoCs) are reviewed. The paper further examines the evaluation methodologies and experimental performance measures for RT-NoCs. In addition, this paper gives an insight into open research points in this field. Several surveys about NoCs were previously conducted providing an introduction to the NoC field [6]; an advanced introduction to the area of NoC-based MPSoC design [7]; or focusing on specific challenges in the design [8], [9]. As an extension to the previous work [10], this is the first survey which targets RT-NoCs to the best of the authors' knowledge. Compared to [10], this paper provides a deeper classified architectural overview and expands to cover the real-time energy efficiency techniques. The work is further fastened by a performance comparison section.

The paper is organized as follows. Section 2 discusses design challenges of RT-NoCs. Section 3 surveys the state-of-the-art techniques for RT-support in NoC architectures. Existing adaptivity and energy efficiency approaches targeting RT-NoCs are presented in Section 4. Section 5 provides an overall comparison of the surveyed approaches, analyzes the evaluation methodologies and experimental performance measures and discusses future insights. Finally, Section 6 concludes the work.

2 Overview and Preliminaries

2.1 Real-Time Challenges in NoC design

Extensive research is found in the literature for efficient customization of NoC design features to allow for applications requirements [11]. However, the use of NoC in RTSs imposes complex constraints. This relates to the fact that performance metrics in an RT-NoC are interlocked with an extra paramount constraint, namely predictability. Messages belonging to a real-time flow have a strict deadline and missed deadlines are as faulty as lost packets. In hard real-time systems (HRTSs), no deadline is allowed to be missed even under worst-case communication scenarios, as it would cause a catastrophic consequence. On the other hand, soft real-time

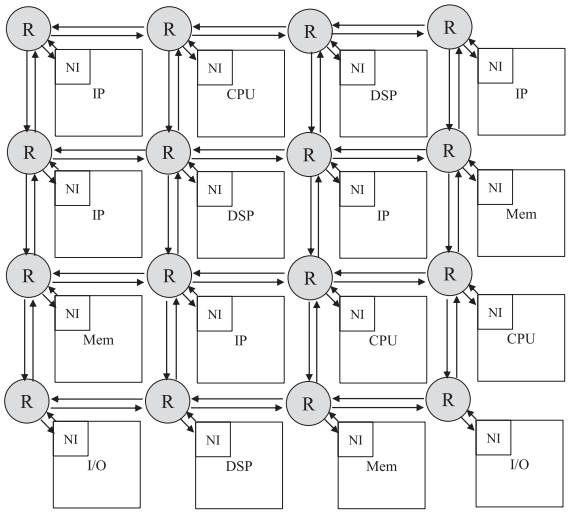

Fig. 1. Real-time NoC-based heterogeneous MPSoC architecture.

systems (SRTSs) can tolerate a small number of missed deadlines. Accordingly, transmission delays between the NoC routers must be predictable in order to be able to guarantee an end-to-end latency. However, predictability is a challenging constraint in a NoC as a shared medium between several concurrent applications and basically in congestion situations.

In that context, a NoC supporting RTAs must provide guaranteed services in terms of bandwidth and end-to-end latency [12]. As a branch of embedded systems, RTAs are also restricted by cost and energy efficiency requirements. Several approaches were presented in the literature to minimize NoC power consumption. However, they affect the timing behavior of the NoC, and thus, they are not suitable for RT-NoCs, highlighting the need for techniques minimizing energy consumption while committing to timing constraints [13]. Additionally, real-time MPSoCs are subject to varying workloads and application-specific demands during runtime which makes adaptive techniques highly justified. This survey covers state-of-the-art RT-NoCs architectural solutions as well as adaptivity and energy efficient techniques as outstanding challenges for RT-NoCs in future NoC-based platforms.

Before discussing the contributions within each design goal, the following section provides the preliminary definitions and parameters framing a real-time NoC.

2.2 Preliminaries

The basic structure of a NoC-based MPSoC platform, as shown in Fig. 1, consists of a set of cores $\{C_0, C_1, \ldots, C_n\}$, connected using a set of routers $\{R_0, R_1, \ldots R_n\}$. The platform is used to run a certain application which consists of a set of tasks $\{T_0, T_1, \ldots, T_m\}$, each assigned to a core. Tasks communicate data between each other generating a set of traffic flows $\{F_0, F_1, \ldots\}$ traversing over the NoC. Flows are structured in a set of packets $\{p_0, p_1, \ldots, p_k\}$ and packets are structured in a set of flits $\{f_0, f_1, \ldots, f_z\}$ which is the basic unit transferred over the NoC. An RTA, mapped onto a NoC-based MPSoC, is basically concerned with performance guarantees with respect to time constraints and commitment to deadlines. From the NoC perspective, this refers to the communication performance particularly under the worst case scenario. This section presents the basic terms for timing and contention analysis within NoCs in relation to the real-time performance and the predictability challenge. For more detailed analysis, the reader is referred to Sections 3 and 4 in [14] and [15].

- *NoC performance guarantees*: refers to a guaranteed maximum latency or a guaranteed minimum throughput for a given NoC architecture under certain input traffic constraints.
- *Throughput*: refers to the rate at which a given NoC architecture produces flits to a destination core.
- *End-to-end latency*: refers to the total computation and communication latencies associated with a pair of source and destination cores. It covers the time interval from the start of execution of an assigned task on the source core till delivering all data produced by that task to the destination core through the NoC.
- *Packet latency*: refers to the delay of the NoC traversal for a complete packet p_i from a source S to a destination D core. It consists of two main components, as given by (1), the basic packet latency and the network jitter.

$$L_{i:S\rightarrow D} = L_{i_{basic}:\,S\rightarrow D} + J_{i:S\rightarrow D} \tag{1}$$

- *Basic packet latency*: refers to the latency of a packet p_i transferred over the NoC from a source S to a destination D under zero network load condition. It covers the delay of the first flit to traverse the full path, with hop length $h_{S\rightarrow D}$, followed by the consecutive arrival of the rest of the flits. Assuming a one clock cycle link delay, a router pipeline latency L_R and a packet size of z_i flits, the basic latency is calculated using (2).

$$L_{i_{basic}:S\rightarrow D} = h_{S\rightarrow D}\,.L_R + z_i \tag{2}$$

- *Network jitter*: refers to the waiting time experienced by a packet p_i due to direct and indirect interference with packets belonging to other flows. Interference is the blocking time of the packet traversal due to either direct contentions with other packets sharing parts of its path, or indirect contentions between packets sharing and packets not sharing any parts of the path that causes blocking of the former which in turn blocks p_i.
- *Worst case packet latency*: refers to the longest possible delay that a packet p_i may experience during its path traversal from S to D through the NoC. It reflects the worst case network jitter value, since the basic (zero-load) latency term of a packet is fixed for a specific NoC topology and router architecture [4], as provided by (3). The jitter is the elusory parameter in NoC communication challenging the RT-constraints. It depends on several NoC parameters such as the switching technique, the flow control, the arbitration policy, the scheduling policy and above all the instantaneous traffic flow characteristics which originate from the task mapping and the modeled application. For real-time flows, Eq. (3) must commit to the flow deadline d_i as described by (4). SRTSs can tolerate a number of missed deadlines, while for

HRTSs, Eq. (4) is not allowed to fail. When (4) holds for all flows in an RTS, the system meets all its timing constraints and is termed schedulable.

$$L_{i_{worst}:S\rightarrow D} = L_{i_{basic}:S\rightarrow D} + J_{i_{worst}:S\rightarrow D} \quad (3)$$

$$L_{i_{worst}:S\rightarrow D} \leq d_i \quad (4)$$

- *Packet deadline*: refers to the worst case accepted time delay d_i, for the arrival of a packet p_i. It further defines the slack as the maximum jitter allowed for p_i to commit to d_i.

Research efforts concerned with RT-predictable NoCs can be classified into architectural contributions; and formal analytical models. Formal methods for NoC-based designs in general, such as Foroutan et al. [16], estimate performance metrics using high level models at an early design phase allowing a faster exploration for the NoC design space. This preserves the place for these models alongside the implementation and simulation path however, with an abstract exposure to many technological aspects. For the RT-NoCs field, which focuses on the worst-case latency, these models are the means to estimate the worst-case among all scenarios that are hard to cover only by simulation. Kiasari et al. [17] review the popular NoC mathematical formalisms and specify the applicable models for the RT-NoC field. This work focuses on the architectural contributions for real-time NoCs and how they are validated by means of analytical models and simulated performance analysis, as it will be presented in the following sections.

3 NoC Architectures for QoS Guarantees

Quality of service (QoS) denotes the ability of the NoC to provide the communication requirements specified by the application. Several parameters define these requirements such as packet loss, latency, jitter and throughput. From a QoS perspective, NoC designs are classified into two main categories: best effort services (BE) and guaranteed services (GS) [18]. A BE NoC guarantees only the correctness and completion of transmission. In contrast, a GS NoC guarantees commitments to performance bounds. Accordingly, in real-time domain, where predictability is the basic endeavor, GS NoCs are the main concern.

According to Mello et al. [19], QoS is supported in NoC designs using one of the following approaches: (1) network dimensioning, to fit the bandwidth needs of all applications in the system; (2) circuit-switched networks; (3) packet-switched networks with priorities; (4) time-division-multiplexing (TDM) and (5) hybrid combinations of previous techniques. The first approach is used in Xpipes NoC [20]. The NoC is sized at the design phase with adjustment of channels bandwidths to fit all the IP communication requirements. However, relying solely on this approach does not guarantee the aimed NoC solution for two main concerns. First, flows fairness does not offer a congestion-free design even with enlarged bandwidth; and second, it is not the practical solution to fit for different applications with today's complex requirements [19]. The other four approaches have been extensively evolving in the literature. They are categorized based on their followed switching technique; whether circuit-switched, packet-switched, table-switched or hybrid methods. This categorization first, determines a different frame for the means to provide the QoS guarantees within the NoC. Second, it influences the supported class of RTAs. In the following four sections, the basic router and the different contributions to provide QoS guarantees within each approach is presented. Further examination and comparison of performance are provided in Section 5 to show the weaknesses and strengths of each approach and its range of suitability in the field of RTAs.

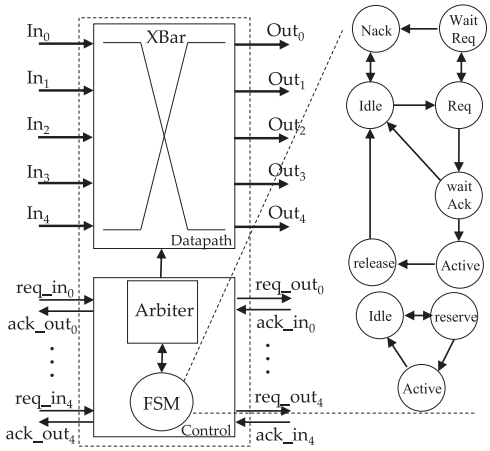

Fig. 2. Conventional baseline architecture of circuit-switched router.

3.1 Circuit-Switched NoCs

The circuit-switching (CS) approach is based on resource reservations and is also referred to as the connection-oriented approach. The communication links from the source to the destination are reserved [21]. A connection path is established for the packets before being injected into the network. Communication is achieved over three basic steps: (1) path setup phase to check or reserve a free path; (2) actual data transmission; (3) cancellation phase to release the path [22]. The basic architecture of a circuit-switched router differentiates between the data and the control circuits, as illustrated in Fig. 2. The control part is responsible for establishing the communication path from the source to the destination [22]. It consists of an arbiter that arbitrates between the different requests from the input links and an FSM for each input and output port. The output FSM keeps track of the state of each output link, while the input FSM keeps track of the processing state of an input request as illustrated by Fig. 2. An active output link refers to a configured connection ready for data transmission from its corresponding active input port. The data-path consists of a crossbar that continuously multiplexes the data from the input to the output ports based on the selection lines configured by the control part. An established route is an ownership for the traffic source, thus, data is transferred at full rate of the links, $Linkwidth.(1/T_{data})$, with no contention. The transmission latency of a message flow becomes a factor of the route length and the message size, while the dependency between traffics is confined to the route setup phase [21]. The total latency from the time the request is raised by the source till the complete data message reaches the destination covers both the setup and transmission time as given by

$$\begin{aligned} L_{i_{Transmission}:\,S\rightarrow D} &= (h_{S\rightarrow D} + z_i - 1)\, T_{data} \\ L_{i_{Setup}:\,S\rightarrow D} &= (FWD+1)h_{S\rightarrow D}\, T_{ctrl} + L_{i_{setup\ blocking}} \\ L_{i_{Total}:S\rightarrow D} &= L_{i_{Setup}:\,S\rightarrow D} + L_{i_{Transmission}:\,S\rightarrow D} \end{aligned} \quad (5)$$

where $h_{S\rightarrow D}$ is the hop distance h from the source S to the destination D; $z_i = Message\ size/Linkwidth$ is the total number of consecutive data flits for a constant bit rate flow F_i; $(FWD+1)h_{S\rightarrow D}T_{ctrl}$ is the basic latency for a successful setup when no contention blocking occurs. It covers the forward request path to the destination with FWD cycles per hop and the backward acknowledgement path to the source with 1 cycle per hop; and $L_{i\ setup\ blocking}$ is the setup blocking delay representing the traffic dependency term. This term highly depends on the routing algorithm of the setup packets, the arbitration scheme as well as the blocking and retry decisions. SoCBUS [21] is the first circuit-switched NoC. It implements a similar structure to Fig. 2. It uses mesochronous clocking and follows a distributed dynamic minimum path routing for path establishment. A blocked request at any node releases all temporary reserved links through the path back to the source where it initiates a retry phase. The backtracking probing is a recently presented technique by Pham et al. [22] to address the setup blocking situation in SoCBUS. Instead of waiting for busy channels to become available, the setup headers are allowed to flexibly backtrack and seek minimum paths alternatives. Though the backtracking probing NoC [22] enhances the setup blocking delay compared to SoCBUS [21], however both NoCs do not provide setup latency bounds.

From the real-time perspective, upon path setup, the CS approach guarantees hard throughput and tight transmission latency bounds. This comes at the cost of non-scalability, path setup delay and inefficient resource utilization due to wasted resources to cover worst case behavior [19]. Therefore, contributions in connection-oriented NoCs are mainly concerned with efficient path setup schemes and improved resource utilization. An efficient setup scheme work on reducing the setup blocking time given in (5) and increasing the percentage of successful setups. From an SRT application perception, this reflects on a reduced number of missed deadlines. However, an HRT application necessitates either a bounded latency setup scheme or a design-time setup phase. SoCBUS [21] relies on a pre-runtime static scheduling phase in order to fit for HRTSs. Similarly, Wolkotte et al. [23] follow a pre-runtime scheduling and configuration however using an external central coordination node (CCN) to configure the path setup and NoC routing instead of a distributed scheme. They further introduce the concept of spatial-division-multiplexing (SDM) in synchronous CS-NoC. SDM addresses both the setup blocking and the resource utilization issues by dividing the links into lanes and allowing physical separation between data streams. Based on bandwidth requirements, a connection uses one or more lanes. The proposed router provides flexibility, however, the use of a CCN limits the scalability.

From another perspective, Liu et al. [24] propose the parallel probing technique as a guaranteed bounded-delay setup scheme. Upon a raised request from a source to a destination, each node sends out two probes to the neighboring nodes which by turn split the received probe into two new probes to reach the destination throughout all possible minimum paths and confirm one established path. Contention between probes from different requests is monitored and handled using an aging priority strategy. The basic input FSM in Fig. 2 is modified to handle three setup failure types and apply an appropriate retry policy accordingly. The proposed search scheme is bounded in the worst case by $(3h_{S\rightarrow D}+6)$ cycles, and a request is guaranteed a successful setup after a worst case delay of $\alpha \times n \times m(3h_{S\rightarrow D}+6)$, where α is the percentage of nodes sending requests in an $n \times m$ NoC.

On the level of resource utilization, it is addressed in the literature by applying the concept of virtual circuits using time-division-multiplexing (TDM) which will be detailed in Section 3.3. In the next section, RT-NoC architectures using packet-switching are presented.

3.2 Packet-Switching NoCs with Priorities

Packet-switched (PS) NoCs, known as connection-less NoCs, are promoted for the efficient bandwidth and network resources usage compared to CS-NoCs, however, with a more complex router and longer transmission latencies [19], [25]. Data is divided into packets, which find their way through the NoC according to a flow control, a routing algorithm, an arbitration rule and the current traffic load. Wormhole-switching with credit-based flow-control is typically used in PS-NoCs for its high throughput and low buffering requirements [4]. However, from a RT-view, wormhole NoCs are very hard to predict as they include both direct and indirect interferences between all flows equally which complicate the analysis of the worst-case latency [26]. This highlights the need for priority-based PS-NoCs. Traffic flows are clustered into different groups, according to their required level of service guarantees, each assigned a priority value based on a scheduling policy. In contention situations, higher priority flows are serviced first based on a priority-aware arbitration-policy. The priority-arbitration policy together with the scheduling policy confine the influence of interference which simplifies the worst-case analysis and allows guarantees to each service level [26]. This approach is promoted by the typical use of NoC-based MPSoCs running diverse applications in parallel bidding for different service requirements, hard and soft real-time, or non-real-time. The scheduling policies are categorized into either static or dynamic priority assignment; while the priority-based arbitration policies are categorized into either packet-level (non-preemptive) or flit-level (preemptive). A preemptive policy allows a higher priority packet to preempt an already progressing lower-priority packet necessitating the use of virtual channels (VCs) capable of storing a packet blocked during its progression. This feature gives VCs-based PS-NoCs [4] the advantage of a mitigated interference influence between traffic flows. In addition, a VC-based router is highly applicable with the prioritized traffic flows, where each VC may belong to a particular priority of one of the traffic clusters. Fig. 3 illustrates the structure of a baseline VC-PS-router. A priority-based VC-PS-router has a similar structure with the priority arbitration policy integrated in the VC and switch allocation blocks in addition to a defined condition for VC acquirement.

According to the used policies for prioritization and arbitration, the router serves different categories of RTAs and may need to undergo validation tests to fit the target RTS

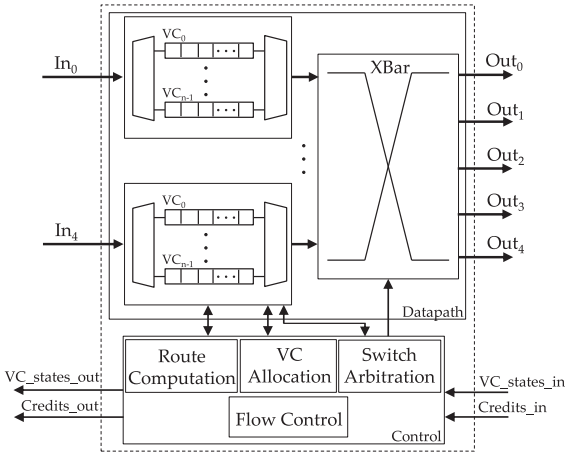

Fig. 3. Conventional baseline architecture of packet-switched router with n virtual channels.

constraints. In the case of HRTAs, flit-level preemption is recommended with fixed priority scheme to restrict the interference for each flow to only flows of higher priorities thus allow tighter guarantees. Furthermore, the priority-based PS-NoC should undergo a design-time schedulability test to guarantee full commitment for all flows under all possible traffic scenarios. Indrusiak [15] presents an analytical method to evaluate the schedulability of such particular NoC architecture. In fact, the schedulability of a RTS highly correlates from the top application model and tasks mapping, the prioritization strategies down to the deep NoC specifications. This highlights the need for efficient RT-mapping schemes alongside with the RT-NoCs architectures, such as the work of Shi and Burns in [27] combining task allocation and priority assignment for HRTAs by a recursive greedy algorithm; the pioneer work of Racu et al. [28], and further Bonilha et al. [29], Sayuti et al. [30] who make use of genetic algorithms to develop static allocations strategies for HRTAs on NoC-based platforms.

State-of-the-art contributions in this area propose and evaluate different scheduling policies, priority arbitration techniques, and efficient architectural solutions targeting different categories of RTSs. QNoC [31] is a primary priority-based PS-NoC architecture supporting four different classes of communication services with prioritized VCs. Each service class has a fixed priority and additional service levels can be further defined by adhering priority ranking. QNoC uses a preemptive arbitration and round robin arbitration is applied for contending packets within same service level. Shi and Burns [27] tackle the hardware complexity problem by proposing a shared priority policy. They derived the corresponding communication analysis and evaluated their proposal in terms of schedulability targeting HRTS with reduced hardware cost.

On the other hand, Diemer et al. [32] and Cheshmi et al. [33] focus on platforms running real-time and non-real time (BE) traffic. They use fixed priority values, however, they propose new arbitration policies targeting an enhanced fairness between flows while keeping guarantees for the high priority real-time flow. In [32], this is based on distributed traffic shapers on each output port allowing low QoS (BE) traffic to access NoC resources as long as enough resources are available for the RT guarantees to be met; otherwise the RT traffic is prioritized. In [33], this is based on a quota setting methodology allowing lower priority flows to use the channel even with the existence of high priority ones however; in a limited access frame based on an assigned quota table. The required performance guarantees are achieved by adjusting the values in the quota table. The QoS aware BiNoC [34] supports real-time traffic with a reduced end-to-end latency. It uses a fixed scheme for prioritized VCs and a new prioritized routing policy for inter-router bandwidths allowing dynamically self-reconfigurable bidirectional communication channels. Idle channels can temporarily change their direction of operation based on the real-time traffic need to resolve congestions on the opposite direction and provide better bandwidth utilization. Nikolić et al. [14] propose the earliest deadline first (EDF) as a dynamic priority-arbitration policy for wormhole priority-preemptive NoCs. They evaluated their proposal with respect to fixed priority schemes in terms of schedulability performance and complexity of computation.

The work in [35], [36], and [37] propose different dynamic priority schemes and target SRTA platforms. In [35], two new mechanisms are proposed, namely aging and discarding mechanisms. The first tackles the starvation problem of blocked data by incrementing their priority after certain number of cycles. The latter, improves the bandwidth utilization by dropping packets of lower priority data if their deadlines are missed and if the number of dropped packets can be tolerated within the SRTA. PRNoC is proposed in [36] with a new round-based priority arbitration scheme. A round refers to the set of packets arriving simultaneously when a new arbitration is initiated, where packets are served based on their priority order. A new arriving packet in an already running round cannot be served except at the end of the round even if it has a higher priority than all in-round packets. Based on a proposed latency prediction model, PRNoC is softly predictable with provided guarantees for higher priority packets while avoiding starvation for lower priority ones. Sudev et al. [37] present the DHARA priority arbitration approach as an enhanced version of their PFS (Priority Forwarding packet Splitting) NoC [38]. The PFS NoC addresses the problems of head of blocking (HoL) and tail-backing of high priority flows due to low priority flows to improve the higher priority latency. HoL is tackled by a priority forwarding technique to temporarily increase the priority of blocked low priority flows which are blocking high priority ones. Tail-backing is solved by a packet splitting technique to end the currently serviced low priority flow and resume its transmission as soon as the requesting high priority flow is serviced. The DHARA approach [37] further improves the overall latency of the RT-NoC by dynamically augmenting the instantaneous priority of the packets based on given packet timeliness information (slack). Late packets are serviced by trading the expendable time-slack of higher priority flows. The latency and number of late packets for low priority flows are improved at the cost of a slightly increased latency for high priority flows but without an increased number of late packets. In the next section, RT-NoC architectures based on TDM are surveyed.

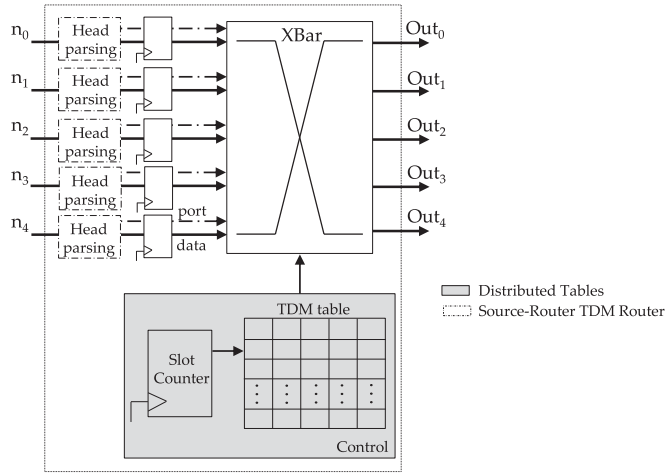

Fig. 4. Basic architecture for distributed/source-routed TDM Router.

3.3 Time-Division-Multiplexing

Time-Division-Multiplexing (TDM) is an alternative to pure circuit-switching to achieve hard QoS guarantees with higher resource utilization efficiency [19]. Resources are shared in time according to associated time tables defining the periodic sequence of use for a given number of consecutive time slots; where each slot is exclusively reserved for a specific connection. Tables are harmonized according to a global TDM schedule creating virtual circuits on top of the NoC architecture in a connection-oriented communication scheme [39]. Once data are injected into the network in their assigned time slot, the global schedule guarantees their traversal through the network in a contention free scheme without intermediate stalls. Thus, the need for in-router arbitration, flow control and buffering is eliminated resulting in a reduced cost and network traversal latency, in addition to flows isolation and predictable performance [40]. The throughput TP_{i} of a flow $\mathrm{F_i}$ becomes function in the ratio of the number of reserved slots S_i to the total table size S_T as given by (6). The latency to transfer the complete data n_i of a flow $\mathrm{F_i}$ depends on the amount of data n_{slot} that can be sent in one slot, the path length $h_{S\rightarrow D}$, and the ratio of reserved slots S_i to the total table size S_T as given by (7).

$$TP_i = TP_{max} \cdot S_i/S_T \tag{6}$$

$$L_{i:\ S\rightarrow D}(cycles) = \frac{n_i}{n_{slot}} \cdot \frac{S_T}{S_i} + h_{S\rightarrow D} \tag{7}$$

However, TDM NoCs are challenged by the computation of their schedules as a cost- and compute-intensive operation in addition to the global synchronicity among all TDM slot tables. TDM schedules are either statically [39] or dynamically [41] computed and may be placed locally at each router for distributed routing [41] or globally in the network interfaces (NIs) only for source routing [40]. Fig. 4 illustrates the basic router architecture for both schemes of distributed and source routing.

Nostrum [42] and Æthereal [40] are two pioneering NoC examples based on TDM. In Nostrum [42], TDM virtual circuits are set up semi-statically. Routes are fixed at design time however with variable bandwidth in runtime using the concept of looping containers. A special type of packets called container are loaded with information and are looped between the source and destination as much bandwidth reservation is required. However, to fit for HRTAs, even the bandwidth must be fixed at design-time. Æthereal TDM NoC has been evolving through [40], [43], and [44]. Æthereal [40] uses static distributed TDM slot tables based on global synchronicity where all routers must always be in the same time slot. Æthereal is followed by its light version Ælite [43] which supports TDM-based composability and predictability in a source routing manner [45]. Ælite further introduces the distributed global notion of time in a TDM-based NoC through mesochronous and asynchronous links allowing scalability on the architectural, physical and functional levels. dÆlite [44] is the final version of Æthereal, using distributed TDM tables on a synchronous platform and supporting multicast connections. It uses a new configuration technique based on a dedicated broadcast network to increase the speed of setting up and release of connections in the distributed tables. On the other hand, the Argo [46] asynchronous TDM NoC is proposed tackling the clock distribution problem in synchronous designs. It uses mesoch ronous NIs to keep common notion of time however with asynchronous routers. Explicit need for synchronization between the TDM routers is revoked by exploiting the fact that pipelined asynchronous circuits behave as FIFOs.

In contrast to the above contributions, SurfNoC [47] and PhaseNoC [48] apply the TDM approach on top of a virtual-channels-based (VCs) PS-NoC. The VCs within every input port are divided into different groups each called a domain inferring different categories of applications. The operation of the VC-PS-router is organized into a series of pipelined steps, where each is further organized into phases based on the different set of domains. At any point in time, each router step is at a different phase, i.e., it is associated with a different single domain of VCs according to a static but programmable schedule. Thus, no contention arises between different domains but only between VCs within the same domain, ensuring a guaranteed non-interfering performance. The router follows an explicit pipelining, where all router steps are exploited however they are all out of phase. The number of different phases, i.e., the number of time slots in the TDM schedule refer to the number of supported application domains. An efficient schedule should allow a smooth wave-like propagation of the packets through the network routers without stalls caused by waiting for their domain at the intermediate routers. SurfNoC [46] proposes a dimension order wave-scheduling analogy. Packets only face stalls when change of dimensions is needed for a dimension order routing, otherwise they are forwarded with no interruption. PhaseNoC [48] further reduces the traversal delay by proposing a zero-latency-overhead scheduling scheme capable of avoiding the stalls at the routing turning points where the dimension changes. It further makes use of the one-domain-per-phase guarantee to reduce the size of the allocators. In the next section, hybrid real-time NoC architectures are reviewed.

3.4 Hybrid Techniques

Through Sections 3.1 to 3.3, the adaptation of each of the three switching techniques: CS, PS and TDM for RT-NoCs was presented. Each either belongs to connection-oriented

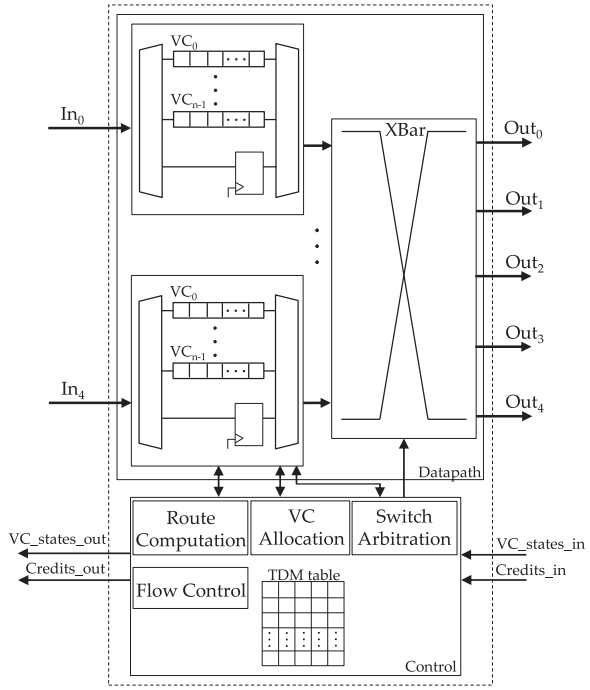

Fig. 5. Aggregate architecture for a hybrid switched router.

(CS and TDM) or connectionless (PS) approach and differently provides hard or soft RT-guarantees. In fact, RTAs, mainly targeting MPSoCs, invoke a mixture of traffic types with different predictability, criticality and QoS requirements. In that context, hybrid NoC architectures are promoted to provide the right mix of soft and hard RT-guarantees and to efficiently handle both streaming and BE traffics generated by RTAs. Fig. 5 illustrates an aggregate architecture for a hybrid router with combined switching techniques which may not all co-exist.

The MANGO NoC [49] is one typical asynchronous hybrid architecture where the routers internally consist of a BE router and a GS router implemented separately. The BE router supports simple source-routing for connection-less data packets. While, the GS router uses the virtual circuits concept to establish hard GS connections by reserving a sequence of independently buffered VCs. The GS router provides non-blocking switching, thus, link arbitration becomes the basis for guaranteed service. A comparable approach is presented by Heisswolf et al. [50] to support both hard GS and BE traffic however through dynamic time slots reservation per VC. For GS traffic, the header surfs first through the network reserving the requested time slots in each router till the destination, while for BE traffic, the packet is injected into the network without prior reservation. Allocation flexibility is supported through on demand VCs and time slots reservation in a self-governed scheme for each router with no VCs specifically reserved for BE or GS traffic. Weighted round robin (WRR) arbitration further allows reserving different number of time slots therefore supporting different levels for hard guarantees.

In [51], [52], and [53], hybrid NoC architectures are proposed, integrating both TDM-based CS-routers with PS-routers. In [51], the circuit switching sub-network further combines the SDM and TDM techniques in an attempt to provide a guaranteed service with increased path diversity and improved resource utilization. Pakdaman et al. [52] implement the circuit-switching based on an online hierarchically-centralized TDM allocation. A root node keeps the state of all NoC links and guarantees to efficiently find a path with the required number of slots if such a free path exists. The proposed scheme supports communication prediction to hide the circuit setup time. The PS and CS parts of the router share the same inter-router links and crossbar, where the CS-data is prioritized if allocated. Yin et al. [53] integrate a time-slot stealing technique which allows packet-switching data to snatch a circuit-switching allocated slot if no belonging flits exist. The slots allocation is implemented in a distributed manner following an adaptive routing. CS-data is allowed to be sent through packet-switched links until the requested path is setup. The NoC is further outfitted with CS-path-sharing techniques which improves the slot table utilization in case a source is communicating with two adjacent destinations, or two sources communicate with two destinations if the new needed path is a subset of the already reserved path. The next section discusses the adaptivity and energy efficiency in RT-NoCs.

4 Outstanding Aspects for Future RT-NoCs

This section focuses on adaptive and energy-efficient techniques as outstanding aspects for RT-NoCs in future chips.

Adaptive schemes allow the system to flexibly support a wide range of RTAs with varying demands during runtime. System characteristics are dynamically changed to fulfill the applications requirements with the optimum exploit of resources. In an RT-NoC, adaptivity can be applied on the level of switching techniques [54], [55], priority values [55], resource allocation [56] and routing schemes [57]. Goehringer et al. [54] and Ruaro et al. [55] monitor the system runtime performance to dynamically change between CS and PS connections. The Star-Wheels NoC by Goehringer et al. [54] avoids long waiting times of blocked CS-channels; thus, it implicitly, as hinted by the authors, guarantees bounded deterministic WCCL, suitable for RTAs, however not computed. Ruaro et al. [55], based on received latency events or deadline violations, they establish or release CS-connections and further adapt the priority level of the flows. This improves deadline violations for SRTAs; while for HRTAs, adaptation must be disabled to avoid deadline miss. Diguet [56] monitors the status of NI FIFOs to allow maximum exploit of time slots in self-adaptive TDM NoC. A received empty or full FIFO signal triggers the sizing of the FIFO and the switching between compliant statically-computed TDM tables. Heisswolf et al. [57] presented the connection-oriented rerouting NoC. Established connections, if blocking other requests, are rerouted to alternative paths if the latter exist. Route switching is only performed if the new route is reserved. Requests success is increased but not guaranteed, and reservation latency is not bounded. However, within a design-time scheduling, rerouting would allow a wider set of established HRT connections with bounded latency and better exploit of resources.

In RT-NoCs, energy is optimized at different levels, however, the applied technique must not interfere the timing

behavior of the NoC to keep the guarantees. Techniques start from mapping algorithms, voltage frequency scaling (VFS) down to deep architectural power-gating (PG) [7]. They are either applied on top of an RT-NoC architecture such as the PG to turn off unused parts of the NoC [53], or applied at an early design stage where energy constraints are integrated together with the RT-constraints into an algorithmic solution [58], [59], [60], [61], [62]. The hybrid-switched TDM NoC [53] from Section 3.4, applies PG eminently on the level of the TDM tables and the VC buffering. The table size as well as the number of activated VCs are dynamically determined according to the traffic need. For the VFS, Zhan et al. [58] statically assign heterogeneous VF values for HRTS based on heuristic search algorithm to meet time constraints and minimized energy. They developed a WCCL model based on network calculus used to determine the time slack. A feedback-control system is further integrated to refine the VF values in runtime based on the NoC RT-performance. Grammatikakis et al. [59] propose a dynamic frequency scaling strategy for SRTAs based on a linear prediction scheme. It predicts the completion time of traffic initiators based on monitored cumulative output packet rate. The predicted time is compared with the prescribed deadline and scaling decisions are taken accordingly. For an application-specific TDM NoC, He et al. [60] propose two algorithmic solutions: a path-based integer linear programming (ILP) and a negotiation based heuristic. The algorithms determine the routing, time slots allocation, links and buffers capacity such that the QoS requirements are met and the total power is minimized. Sayuti et al. [61], [62] focus on energy minimization through task allocation as one of the most effective ways to address energy efficiency in RT-NoCs. They propose a multi-objective genetic optimization algorithm (MOGA) capable of guaranteeing a static task allocation with both: (1) schedulability for HRTSs and (2) minimized energy consumption. They developed energy macromodels and RT-schedulability analysis which are both integrated into a multi-objective fitness function to meet timing constraints and minimized energy. Other techniques such as low power codes (LPC) are applied to minimize NoC energy [63], [64] [65], [66]. LPC encode data before their transmission to minimize signal transition activity, thus reducing dynamic power. This provides an attractive alternative to be applied in RT-NoCs without affecting timing constraints. Encoding delay can be considered in the end-to-end latency model without affecting the prediction accuracy of the NoC worst-case behavior. However, no LPC solution is found yet to specifically tackle energy efficiency in RT-NoCs.

5 Performance Comparison of RT-NoCs

In this section, a broad perspective is provided in terms of the evaluation metrics, methodologies and tools followed to validate an RT-NoC solution. This is based on a detailed analysis and discussion of the contributions surveyed in Sections 3 and 4 mainly tabulated in Tables 1 and 2. Table 1 provides a comprehensive summary for all the approaches covered in this work; while Table 2 summarizes several evaluation metrics over both the hardware and the performance levels. It further includes a listing of the utilized tools and the evaluated traffic and/or real applications. Though the evaluation conditions are not similar for all references covered in this work, Tables 1 and 2 provide a means for compare and contrast of the different contributions.

5.1 Approaches Comparison

5.1.1 Classification

In Table 1, the references are tabulated in the same order they are discussed in Sections 3 and 4 with their publication year listed in the first column in the form of xx referring to 20xx. The references are first classified in terms of contribution level either architectural, adaptiveness or energy. The architectural level is further classified based on the switching technique adopted by each reference NoC.

5.1.2 Base-NoC and Proposed Technique

Each of the surveyed references in this work either proposes a distinguished new NoC architecture or bases its contribution on a referenced NoC from the literature, or on the conventional baseline (CB) architecture for CS or PS-NoC illustrated previously in Figs. 3 and 4. This is illustrated by the Base-NoC column in Table 1, such as Bolotin et al. [31] who proposed QNoC, Diemer et al. [32] who build their proposal on top of the conventional PS-NoC and Cheshmi et al. [33] who build their proposal on top of QNoC [31] and the ASPIN NoC [67] from the literature. The next column lists the keywords of the proposed techniques discussed in Sections 3 and 4.

5.1.3 Supported NoC Features

The following columns analyze the references in terms of the supported NoC features.

For the traffic types, the references support either RT-GS traffic, classified RT traffic or a mix of RT-GS and BE traffic. Most of the references support both GS and BE traffics, to fit the typical platforms serving a mixture of both traffic types and to further allow the exploit of the suspended resources reserved but not used by the GS traffic [68].

For the routing schemes, the most widely used is the distributed static XY routing. Some references provide different schemes for BE and GS as denoted by the first and second sub-columns respectively for references [42], [40], [49], [50], [51], [52], [53]. Very few references support adaptive routing [34], [36], [42], [57].

For the clocking scheme, less than 30 percent of the architectural proposals support the GALS approach using either mesochronous [43], asynchronous [46], [49], [50] or source synchronous clocking [22], [24], [31]. GALS architectures highly fit the growing complexity of future many-cores by getting over the clock distribution problem and eliminating dynamic power consumption in the idle state.

For the design flexibility, references are classified into run-time or design-time approaches according to the handling of the proposed design features. For CS-NoCs, this refers to the routing decisions in the setup phase; for the PS-NoCs, it refers to the classification of the traffic flows in terms of priority levels, quota values or traffic shapers. For TDM, it denotes the computation of the TDM schedules; and for the hybrid NoCs it denotes both the computation of the (virtual) circuits reservation and TDM schedules. For the adaptive and energy

TABLE 1
Comparative Summary of State of the Art Real-Time NoCs Contributions

Ref.	Year	Contribution	Switching	Base-NoC	Proposed technique	Traffic Types	Routing	Clocking	Design flexibility	Real-time guarantees	Target platform/ apps	Real-time evaluation metrics	Static analytical approach
[21]	03	RT-support Architectures	CS	SOCBUS	PCC	GS	M	M	D	H	He	Design-time setup	[73]
[23]	05			CB-CS	SDM-CCN		-	S	D		He-WCS		
[24]	12			CB-CS	Parallel probe		M	SS	R		LL	Bounded latency	✓
[31]	04		PS	QNoC	Service levels	Classes	XY		D		He, Ho		✓
[27]	10			CB-PS	Priority share policy			-	D		Ho	Schedulability	✓
[32]	10			CB-PS	Traffic shapers	GS, BE		-	R		He-GPMP	Bounded performance	[74]
[33]	13			QNoC [31]	Quota setting	Classes		A	D		He	Bounded latency	✓
[34]	10			BiNoC	Bidirectional	GS, BE	XY, a-OE	S	R		Ho		-
[14]	14			CB-PS	EDF arbitration	Classes	XY	-			Ho	Schedulable threshold	✓
[35]	07			SoCIN [75]	Aging/discarding	GS, BE		-		S	He, Ho	# LP	×
[36]	09			PRNoC	Round-based		XY, a-OE	-			Ho	Proposed parameters[1]	✓
[37]	15			Hermes [76]	DHARA priority		XY	-			He	# LP	×
[42]	04		TDM	Nostrum	Loop containers		D-a Static	S	S-D	H		Design-time allocation	-
[40]	05			Æthereal	Contention-free		S D		D		-		[77]
[43]	09			Ælite	Flit-synchronous TDM	GS	S	M,A			He-Ct		[78]
[44]	14			dÆlite	Setup tree/multicast		D	S			He-MC		[79]
[46]	15			Ælite [43]	Argo asynchronous TDM		S	A	D		Ho		[80]
[47]	13			CB-VC-PS	SurfNoC TDM @ VC	Domains	XY	S			He-Isol	Isolated performance using static scheduling	✓
[48]	15			CB-VC-PS	PhaseNoC TDM @ VC		XY LA						✓
[49]	05		H	MANGO	Clockless VCs NoC	GS,BE	XY Static	A			He, Ho	Bounded performance	[81]
[51]	12			Subnetworks	SDM-TDM CS / PS		D	S	R	S	Ho	# SR	×
[50]	13			CB-VC-PS	WRR VC-based TDM			A	R D	H	He, Ho	Bounded latency & Schedulability	✓
[52]	14			CB-PS	Hierarch. TDM & predict.			S	R		Ho-CMP		-
[53]	14	RT, E		Garnet [82]	Slot stealing/ VC PG		D-a-M	S		S	He	# SR	×
[55]	15	A	H	CB-PS/CS	Adaptive priority & PS/CS	GS, BE	H	S	R	S	He, Ho	# LP	×
[56]	14		TDM	Self-adaptive	Adaptive TDM & FIFOs	GS	M	S	R	H	He	Design-time allocation	-
[57]	13		H	CB-VC-PS [50]	Rerouting of flits	GS,BE	a	S	R	H	He, Ho	Schedulability analysis & Bounded latency	[50]
[54]	11		H	Star-Wheels	Adapting NoC size	GS, BE	Specific	FS	R	H	He		-
[58]	14	E	H	MANGO [49]	Time slack-hetero-VFS	GS,BE	XY	FS	D	H	Ho, He	Bounded latency	✓
[59]	13		-	-	Prediction-based-DFS		M		R	S	He	# LP	×
[61]	13		PS	CB-PS	MOGA		XY	-	D	H	Ho	Schedulability	[83]
[60]	14		TDM	-	ILP, heuristics		D S-M	S			He	Design-time allocation	✓

Table abbreviations key for:

Contribution:
A: Adaptivity
E: Energy level
RT: architectural level for RT-Support

Traffic type:
BE: Best Effort
C: Common deadline set of applications
GS: Guaranteed Services

Clocking:
A: Asynchronous
FS: Frequency Scaling scheme
M: Mesochronous
S: Synchronous
SS: Source-Synchronous

Switching:
CS: Circuit-Switching
H: Hybrid-switching
PS-v: Packet-Switching (Virtual-cut-through)
PS-w: Packet-Switching (Wormhole)

Base-NoC:
CB-CS: Conventional Baseline Circuit-Switched
CB-PS: Conventional Baseline Packet-Switched
CB-VC-PS: CB-PS NoC with Virtual Channels

Target platform/applications:
Ct: Concentrated topology
D-RT: pre-Defined Real-Time applications
Isol: Isolated applications
CMPs: Chip-Multi-Processors
GPMPs: General Purpose Multi-Processors

Routing:
a: adaptive
D: Distributed
H: Hamiltonian routing
LA: Look-Ahead

QoS guarantees:
H: Hard guarantees
S: Soft guarantees
NC: No Contention case

He: Heterogeneous
Ho: Homogeneous
MC: Multicast
WCS: Wireless Comm. Standard
LL: Long Lifetime applications

M: Minimal
OE: Odd-Even
S: Source routing
(w)XY: distributed (weighted) XY
2 columns: BE routing (L) GS routing (R)

Workload management:
D: Design-time
R: Run-time
S-D: Semi-Design time

Real-time metrics:
#LP: number of Late Packets
#SR: number of successful requests

[1] Proposed evaluation parameters: temporal fairness (between GS and BE), Quality of QoS support for high priority and Latency prediction accuracy.

management contributions, it refers to whether the proposed technique is to be applied during run-time or only at design-time.

Finally, the column of real-time guarantees specifies the category of supported RTSs whether hard or soft.

5.1.4 Supported Platform and Target Applications

A NoC-based MPSoC platform is classified into homogeneous (Ho) or heterogeneous (He) in terms of the connected IP types as listed in the target platform column. This refers to uniform or variant traffic patterns and loads flowing throughout the network. Around 65 percent of the surveyed works support heterogeneous architectures, while only 30 percent of the surveyed works explicitly mention their exclusive support for homogeneous platforms. Some references mention specific target platform or a set of applications which benefit more from the proposed approach such as the long transmission-path lifetime (LL) targeted by [22] and [24] and specifically the wireless communication standards (WCS) supported by [23]. Furthermore, GPMPs and homogeneous CMPs are specifically concerned in [32] and [52] respectively. On the other hand, the work in [47] and [48] target isolated application tasks (Isol) where the system is divided into a set of different application domains that should be handled in an isolated manner without affecting the performance of each other. Special features are further supported by some proposals such as multicasting communication in [44] and concentrated (Ct) topologies in [43].

TABLE 2
Comprehensive Results for the State of the Art Real-Time NoCs

Ref.	NoC Design parameters					Hardware evaluation					Simulated performance evaluation								Evaluation tools		
														Tested traffic							
														Synthetic							
	# of nodes	Data width (bits)	Buffer size (flits)	# VC/SDM/ TDM slots	Hop latency (cycles)	Technology (nm) /FPGA	Area (10^{-3} mm²)	Frequency (GHz)	BW/link (MB/s)	Energy (pJ/bit)	Latency (cycles)	Throughput (flits/node/cycle)	Packet size (flits)	UR	TO	TR	HS	Real Apps	Simulator	HW	Energy
[21]	64	8	n/a	n/a	1 4	-	-	1	1	-	687	-	400	x				PSTN/VoIP	C++		
[23]	1	16	n/a	4	-	130	50.6	1.075	2.17	0.329	-	-	-					HL,UMTS,DRM		SDC	SPC
[24]	64	64	n/a	n/a	1 2	90	18733[2]	1.8	14.4	-	360	-	400	x					-	SDC	
[31]	16	16	2	4	-	-	-	1	2	-	11[3]	-	2	x				Signaling,DMA,RT	OPNET		
[32]	64	32	16	4	4	V4	2712[4]	-					8	x	x			PARSECSuite	SysC	Xil	
[33]	49	-	-	n/a	-	90	1285	-					128					VOPD	SysVer		Spice
[34]	64	-	8	4	4	-					50	0.25	16	x		x	x		HDL		
[35]	16	16	8	n/a	3	-					9.81[4]	-	2	x				Signaling,DMA,RT	C++		
[36]	16	8	-	n/a	-						116	-	-	x		x			SysC		
[37]	16	-	2	n/a	-	V7A	2763[5] 1176	-					-	x					SysVer	Viv	
[42]	16	-	n/a	-	-	-	13896[6]	-			8	-	1	x				Ericsson DSP	-	SDC	
[40]	4	32	8	256	2	130	130	0.5	2	-	246	1	-						Æ [77]		
[43]	4	32	4	n/a	3	90	32	0.875	3.5	-	246	1	-	x							
[44]	4	32	n/a	32	2	65[7]	5.774	0.925	3.7	-	60	1	-							SDC	
[46]	16	32	n/a	23	3	65	7.715	1.13	4.5	8.24	-	1	-							SDC	SPT
[47]	64	32	8	16 2	4	45	inc.	-	-	inc.	35.9	0.43	1	x					BS 2	SDC	
[48]	64	64	min	2 4	1	45	80.66	1.25	10	-	31.25	-	1-5	x		x			SysC	Cad.	
[49]	1	32	1	8	-	130	277	0.646	2.58		-		V							STM	
[51]	49	64	n/a	4	4 -	65	86.49	2	16		-		-	x					SysC	STM	
[50]	100	128	8	4	8 2 1	45	139.09	1.15	17.6	inc.	9.3	0.55	5	x				VOPD, MPEG4, PIP, MWD	SysC	SDC	
[52]	49,16	64	8	2	10 ×5	-	45	-			app.specific results		-	2-8				Splash-2, Apache, Oracle, TPC-H	BS 2		OR3
[53]	36	128	5	4	128	-	45	188	1.5	24	rel.	128	-	5	x	x	x		SPEC OMP 2001 Rodinia & CUDA	Garnet	NG	OR2
[55]	20	16	-	-	-	-	-	-	-	-	-	-	-					MJPEG, DTW	VHDL/SysC		
[56]	-	-	n	-	-	V5	-	-	-	-	-	-	-					Object tracking			
[57]	64	256	-	8	2	45	515.19	0.8	25.6	-	-	-	-	x		x	x	MWD, MPEG4, VOPD, PIP	SysVer		
[54]	-	32	-	-	-	V5	233-1105[8]	0.1	-	-	-	-	-					Z-Align		Xil	
[13]	16	128	4	3	5	45	-	2-S	32-S	rel.	-	-	-					MJPEG, PIP	BS 2		OR2
[58]	16	128	4	3	5	45	-	2-S	32-S	rel.	-	-	-					MJPEG, PIP	BS 2		DSENT
[59]	16	128	-	-	0-4	-	-	1-S	16-S	rel.	-	-	-					MPEG4	SysC		[94]
[61]	16	16	2	-						rel.	-	-	-					Autonom. vehicle			[65]
[60]	-	-	-	-	-	65	-	-	-	rel.	-	-	1-3	x				Multimedia SoC			OR3

Table abbreviations key for:
General entry:
"-": not specified, "n/a": not applicable
"inc.": incomplete results setup
"rel.": relative improvement value
Tools-Hardware (HW):
Cad.: Cadence models
NG: NanGate models
SDC: Synopsys Design Compiler
STM: STMircoelectronics models
Viv: Vivado Design Suite
Xil: Xilinx tools

Packet size: V: Variable size
#VC/SDM(TDM): 2 columns: VCs(L), Table size (R)
Node latency: 2 columns: data (L), setup (R)
Technology: V2,4,5; V7A: Xilinx Virtex-2,4,5; Artix FPGA
Frequency: S: frequency Scaling scheme
Tools-Energy:
Or2, Or3: Orion 2.0 [95], 3.0 [96]
SPC: Synopsys Power Compiler
SPT: Synopsys Prime Time
Traffic-Synthetic: UR: Uniform Random, TO: Tornado, TR: Transpose, HS: Hotspot.

Tools-Simulator:
Æ: Æthereal tools [77]
BS 2: BookSim2.0 [97]
C++: developed using C++
HDL: developed using HDL
SysC: developed using SystemC
SysVer: developed using System Verilog
Traffic-Applications:
DRM: Digital Radio Mondiale
HL: HiperLAN
MWD: Multi-Windows-Display
PIP: Picture-in-Picture
PSTN: Public Switched Telephone Network
UMTS: Universal Mobile telecom. System
VoIP: Voice over IP
VOPD: Video Object Plan Decoding
CS-approach
PS approach
TDM approach
Hybrid approach
Energy efficient
Adaptive

[2] Area reported in NAND gates
[3] Results for highest priority flow representing 12.5% traffic portion
[4] Area of only arbiter reported in equivalent gates
[5] FPGA area usage: 2763 LUT and 1176 register
[6] Area reported in gates with no specified technology
[7] Design also implemented on Virtex 6 FPGA
[8] Heterogeneous router sizes range from 233 to 1105 slices

5.1.5 Real-Time Evaluation and Analytical Approaches

The last two columns in Table 1 examine the validation of a NoC solution for real-time guarantees. The first column indicates the parameter used by each reference to evalluate the level of their real-time support. This falls under one of the following criteria:

- Implicit guarantees by static allocation/setup schemes.
- Schedulability tests performed statically at design-time. References either derive their own subjective analysis, use state-of-the-art analysis, or directly assume schedulability study applied at design-time and build their proposal on top of this assumption.
- Bounded performance (max latency, min throughput). Bounded throughput is demonstrated in the references by either derived analysis or simulated test cases while bounded latency relies on derived mathematical latency bounds either within the reference itself or based on a state-of-the-art analysis.

- Number of late packets (#LP) reflecting the number of missed deadlines or the number of schedulable tasks.
- Number of successfully established connection requests in a circuit-/ hybrid-switched NoC.
- Isolated performance between different flows in terms of constant throughput/latency following static scheduling.

In fact, the NoC evaluation process, in general, may either adopt a simulation-based flow or an analytical-model flow. Simulation-based flows offer the designer higher flexibility to select the detailed design parameters and provide accurate results. However, due to the increased complexity and details of the NoC-based systems, simulation-based flows turn to be very time consuming and limited to only a finite set of test scenarios [17]. Each design iteration may take hours to simulate few seconds of a particular system-configuration, and evaluate a single point in the huge NoC-design space, to provide exact figures for each and every communication packet traversing the NoC. Furthermore, the non-linear behavior of the NoC-based systems produces dispersed results, which hardly lead to decisive conclusions. Alternatively, for fast and aggregate exploration of the NoC design space, abstract analytical models for NoCs are promoted [15], [17], [69]. An appropriate model should allow for fast and accurate performance estimations early at the design phase. Trading accuracy for abstraction speed-ups has been studied in several models [69], [70], [71], [72]. Specifically analytical and simulation models were very well characterized by Indrusiak et al. in [69].

From a RT-NoC evaluation perspective, the main concern is the worst-case communication latency (WCCL) prediction. This infers considering all possible scenarios for the system, which is hard to cover only by simulation. Thus, for HRTS, design-time formal analysis for WCCL estimation is compulsory to provide hard guarantees. Simulations can further be used to illustrate the NoC performance at specific traffic scenarios. While for SRTS, simulation models are used to evaluate NoCs using parameters such as number of late packets and successful requests.

Accordingly, for the surveyed references in this work, the defined evaluation parameter for each contribution is directly related to the supported category of RT-guarantees specified previously in the table, as well as to the adopted analytical approach specified by the last column. HRT-NoCs must involve a static analysis validation, which can be a design-time setup/allocation scheme, a bounded WCCL analysis or a schedulability test at design-time to guarantee no single deadline miss during operation.

This analysis may either be based on a state-of-the-art schedulability or WCCL estimation model, or based on a subjectively developed model as depicted in the last column in Table 1 by either the reference number of the state-of-the-art model or a "✓" respectively. Other references, assume the application of such models and build their proposal on top of this assumption, as depicted in this column by "-", such as the TDM references which are based on design-time allocation of TDM slots. References which do not include static analysis are depicted by "✘".

Further analytical formalisms for RT-NoCs are found in the literature [84], [85], [86], [87]. They are either based on real-time network calculus, schedulability (response-time) analysis or data-flow analysis, which can all estimate the WCCL using different modeling for the NoC parameters and resources [17]. Bekooij et al. [84] use synchronous data-flow models to derive the end-to-end temporal behavior of a TDM-NoC-based multiprocessor system. Qian et al. [85] used network-calculus to compute the WCCL for each flow in wormhole NoCs with credit-based flow control. They assume prior knowledge of the arrival curve of each flow and derive an equivalent service curve for the routers on each flow, based on developed components models and analyzed interference patterns. Shi and Burns [26] compute the WCCL for priority preemptive wormhole NoCs with VCs using response-time (scheduling) analysis. Based on the work of Shi and Burns, Indrusiak [15] developed an end-to-end schedulability analysis test for the priority preemptive wormhole NoC-based platforms. Nikolić et al. [86] derived WCCL estimates for wormhole NoC-based platforms with limited migration model (LMM) for application mapping as a promising approach in the RT-field towards scalable and predictable multicores. They propose techniques to handle the traffic non-determinism caused by the LMM to analyze the WCCL. Rambo and Ernst [87] also used scheduling analysis to derive the WCCL in wormhole NoCs with equally treated VCs however, with each VC allowed to be shared among more than one traffic flow.

On the other hand, simulation models are further used for the RT-NoCs to either provide illustrative results for particular test application scenarios, or to validate improvements of specific NoC performance parameters at standard traffic patterns such as better fairness, increased saturation throughput, reduced average latency. These parameters allow cross reference comparison and provide a frame for the supported category of applications.

5.2 Evaluation Comparison

The main evaluation concern of an RT-NoC is its validation with respect to the promised guarantees, as discussed in Section 5.1.5. Alongside, an RT-NoC is further evaluated in terms of its hardware cost and simulated performance parameters to allow cross reference comparison and provide a frame for the supported category of applications. This is given by Table 2 through a deeper look into the surveyed references. The table presents a set of reported evaluation parameters and lists the used experimental setup for each contribution. NoC evaluation parameters are not restricted to those presented in Table 2. However, at an attempt for a reliable overview, the most commonly evaluated parameters are selected with common definitions to fit the maximum possible number of references. Table 2 is divided into four main parts: the NoC design parameters, the hardware evaluation, the simulated performance and the used evaluation tools.

5.2.1 NoC Design Parameters

The first part of Table 2 lists the set of design parameters for the evaluated NoCs. All reported results are for $n \times n$ mesh structures, except for the star-wheels [54] NoC topology. The first column lists the total number of nodes for the evaluated NoC. The following columns define the design parameters of the router architecture in terms of the router

datawidth, the buffer depth per input port (per VC if applicable) and the number of VCs and/or the number of slots per TDM table. The last column provides the router hop-latency for data traversal of one flit, referring to the pipeline-depth of the router. For router architectures involving a setup phase, the second sub-column denotes the router hop-latency of the setup control path.

5.2.2 Hardware Evaluation

The second part provides the hardware results for the proposed design of a single router in terms of the area, the data-path operating frequency, maximum theoretical bandwidth per link and energy evaluated for a certain technology. Most of the reported results are for an ASIC design. In case the results are reported for FPGA implementation, the technology refers to the type of the FPGA board and area is reported in terms of FPGA resources.

Among the surveyed references, only 50 percent report hardware results, all belonging to the architectural level category. None of the contributions on the energy levels provide any real hardware verification. The maximum reported operating frequency for a RT-NoC router is 2 GHz [51]. The bandwidth per link is computed as the operating frequency $\times$ the datawidth referring to the maximum aggregate bisection bandwidth. A few references report results regarding energy consumption. The presented values here express the normalized energy consumption per bit for a uniform traffic with 50 percent switching activity computed using the following equation to allow a reliable cross reference comparison:

$$\text{Energy } (\text{J/bit}) = \frac{\text{P}}{\text{T}_{\text{simulation}} \times \text{N}_{\text{bits}}}, \tag{8}$$

where P is simulated power consumption, T is total simulation time and N is total number of transferred bits in this time. In the energy column, the "-", "inc." and "rel." entries designate non-simulated energy, incomplete results that miss one or more of the above described parameters or only given relative improvement percentage results respectively.

5.2.3 Simulated Performance

The third part covers the system level performance evaluation in terms of latency and throughput, and the traffic types and application benchmarks tested by each reference. RT-NoCs are simulated to provide illustrative results for specific test scenarios, or to validate enhancements for specific NoC performance parameters such as better fairness, increased saturation throughput, reduced average latency; at standard traffic patterns.

Tested Traffic and Applications: The most commonly tested pattern is uniform random (UR) traffic. Other patterns such as tornado (TO), transpose (TR) and hotspot (HS) traffics are also used in some references. Several references settle the results validation based only on synthetic traffic, while others extend the reliability of the results through real-life applications as provided in the last column in Table 2. Video processing applications such as MPEG-4, MJPEG, Video Object Plan Decoding (VOPD), Picture-in-Picture (PiP), Multi-Windows-Display (MWD) are repeatedly used based on task graphs representation [13], [33], [50], [58], [59]. Other (hard) RTAs applications are also found in the literature [21], [23] such as the public switched telephone network (PSTN), the voice over IP (VoIP), HiperLAN (HL), the UMTS and the Digital Radio Mondiale (DRM) wireless communication standards. Ericsson Radio Systems application, autonomous vehicle application, in addition to open-source and commercial benchmarks such as PARSECSuite [88], Rodinia [89], CUDA [90], SPEC OMP 2001 [91], Splash-2 [92], Apache, Oracle and TCP-H [93] are also used for validation.

Latency and Throughput Parameters: The average latency and throughput comes on top of the list of common NoC evaluation parameters. Both latency and throughput values listed in Table 2 belong to simulations performed under uniform random traffic as the most repeatedly used in all references. The tested packet size is listed respectively in flits.

The reported latency values in cycles are the average latency at an offered traffic load $\approx$ 0.2 flits/node/cycle extracted from presented graphs or tables in each reference. For HS-NoCs or PS-NoCs, the listed values refer to the average transmission latency of RT-GS traffic. For CS-NoCs and TDM-NoCs, the listed values refer to the average latency of path-setup and tables-configuration, respectively. The values vary largely between the references since, in addition to the switching technique and the NoC architecture, latency further depends on the size of the NoC and the size of transmitted data packets. In Section 5.3, normalized latency values are compared and discussed.

On the other hand, the reported throughput values correspond to the maximum saturation throughput for RT-GS traffic in flits/node/cycle. As listed in the table, design-time allocated TDM-NoCs guarantee to achieve a full load operation with the same transmission latency as zero load.

5.2.4 Evaluation Tools

The last part lists the tools used for the NoC evaluations. The listed evaluation tools used by each reference are categorized into system level simulator, hardware evaluation and energy modeling and simulation tools.

Researchers may use open source simulators such as BookSim 2.0 [97] used in [47], [52], [58], [59], Garnet [82] used in [53] and OPNET [98] used in [31]. Others build their own simulators using C++, SystemC, SystemVerilog or an HDL language [21], [34], [35], [36], [37], [40], [43], [44], [48], [50], [51], and [59].

On the hardware level, most reported results target ASIC implementation synthesized using Synopsys Design Compiler (SDC). Others use models from Cadence, STMicroelectronics or NanGate libraries such as [48], [51], and [53] respectively. For the FPGA implementations in [32], [37], [44], Xilinx tools are used. Some references report synthesis results on both ASIC and FPGA such as [44] and [46] using Virtex-6 and Altera Cyclone II respectively.

For energy modeling and simulation, Spice models, Synopsys Power Compiler (SPC), Synopsys Prime Time (SPT) are used to simulate the power consumption on the hardware level. Orion 2.0, 3.0 (OR2, OR3) [95], [96] and DSENT [99] as well as other open source NoC energy macro-models [65] and [94] are used for system level.

5.3 Normalized Results Comparison

For a more conclusive comparison, normalized results for area, bandwidth per link and latency are provided in Table 3. The best result in each column is underlined. The

TABLE 3
Normalized Comparison

Ref.	Area 10^{-3}mm^2/bit		Bandwidth/ Link	ε_H	normalized Latency (cycles)	ε_P
	Router/port	NI				
[23]	0.076	-	2.17	5.71	0.511	11.17
[24]	0.082	-	14.40	35.12	0.36	97.56
[31]	-	-	2.00	-	2.06	-
[34]	-	-	-	-	0.58	-
[35]	-	-	-	-	1.83	-
[40]	0.132	0.508	2.00	3.03	0.54	5.61
[43]	0.050	-	3.50	14.00	0.54	25.93
[44]	**0.018**	0.505	3.70	**41.11**	0.29	141.76
[46]	0.024	0.455	4.50	37.5	**0.22**	**170.46**
[47]	-	-	-	-	6.73	-
[48]	0.252	-	10.00	7.94	3.26	2.43
[49]	0.208	0.385	2.58	2.48	-	-
[51]	0.130	-	16.00	24.60	-	-
[50]	0.218	0.544	17.60	16.14	0.93	17.35
[53]	0.294	-	24.00	16.32	6.40	2.55
[57]	0.402	-	**25.60**	12.74	-	-

table includes the references reporting hardware results and/or latency results.

5.3.1 Normalized Area

The reported area values for the routers from Table 2 are all scaled to the 45 nm technology following the ITRS [100] cross generation scaling using the square of the linear-dimension cycle-reduction ≈ 0.7. The area is further normalized into a per bit area depending on the router datawidth. All results are reported for mesh-NoC routers, thus all reported area values correspond to 5-port routers. However, for a generalized normalization, the area is further divided by the number of ports. The area values are presented in 10^{-3} mm^2/bit/port. The dÆlite TDM-router of Stefan et al. [44] which does not include any buffering or arbitration is the smallest router with area of 18 μm^2/bit/port. While the adaptive HS-router of Heisswolf et al. [57] is the largest router applying the TDM concept on top of an 8-VCs PS-NoC structure resulting in an area of 402 μm^2/bit/port. Table 3 further includes area results for network interfaces (NIs) normalized to per-bit values at 45 nm. NIs implement the interfacing to the connected IPs. For PS-NoCs, this involves data packetization and transmission, and typically the complexity of NIs is not comparable to that of the routers [101]. However, for TDM NoCs, the NI overhead dominates the area cost of the NoC, which compromises the routers simplicity as they further involves a scheduling and configuration units [44], [50], [102]. This is notable in Table 3 for the TDM references [40], [44], [46], [49], [50], while for other routers NI area is not reported.

5.3.2 Normalized Total Latency

The reported average latencies in Table 2 are normalized with respect to the tested NoC size and the size of transmitted data packets.

For PS-NoCs and HS-NoCs, latency normalization is computed following (9), where L_{avg} is the NoC average latency listed in Table 2, h_{avg} is the average path length in an $n \times n$ NoC given by $h_{avg} = 2n/3$, and z is the tested packet size in flits. This provides an average normalized latency value liberated from the influence of the transmitted packet length and the NoC size.

$$\hat{L} = \frac{L_{avg}}{h_{avg} \cdot z} \tag{9}$$

For the CS-NoCs the reported values refer only to the setup delays. Therefore, based on the analyzed architectures in Section 3.1, we compute the total latencies including both setup and transmission phases, then normalize the resulting values using (9). For the CS-NoCs, average total latency is given by (10), where average transmission latency is total number z of transmitted flits added to the average path length h_{avg} as given by (11).

$$L_{avg_{Total}} = L_{avg_{Setup}} + L_{avg_{Transmission}} \tag{10}$$

$$L_{avg_{Transmission}} = h_{avg} + z - 1 \tag{11}$$

For the TDM NoCs Æthereal [40], Ælite [43] and dÆlite [44], the reported latencies refer to the online configuration of the TDM tables based on a design-time allocation scheme. The dÆlite improves the configuration time compared to Ælite and Ælthereal by using a new centralized tree-based scheme. The latencies for the three NoCs are extracted from [44]. However, the concerned total latency should include both the configuration and the transmission. The transmission latency for TDM NoCs as described by (7) in Section 3.3, depends on the reserved slots. Assuming 0.2 traffic load, i.e., 0.8 of the TDM slots can be used. Therefore the total latency can be computed for a 128-flit transmitted data using (12).

$$\begin{aligned} L_{avg_{Total}} &= L_{avg_{Config.}} + L_{avg_{Transmission}} \\ &= L_{avg_{Config.}} + 128 \times 1.25 + h_{avg} \end{aligned} \tag{12}$$

For the Argo TDM NoC [46], the TDM schedules are allocated and configured offline, therefore the total latency refers to the transmission latency only.

Some references report their latencies for uniform traffic, however constrained under specific path lengths such as 4 hops in [36], 6 hops in [40], [43], [44], [46] and 2 hops in [50]. For these references, h_{avg} is the used path length. Table 3 lists the normalized latency values showing the shortest value for [46] and the longest value for [47].

5.3.3 Normalized Efficiency

Furthermore, for a more equitable comparison where the cost, the speed and the system level performance of the router are considered, two new metrics are defined, namely hardware efficiency (ε_H) and performance efficiency (ε_P). Eq. (13), for ε_H, combines the tradeoff parameters, area (cost) and bandwidth (speed) in one metric serving as an overall per-router evaluation for the hardware design. The bit area is considered for a 5-ported mesh router. The hardware efficiency is further combined with the latency results as given by (14) providing an aggregate evaluation of the router performance efficiency enclosed in one parameter ε_P.

$$\varepsilon_H = \frac{\text{link bandwidth } (MB/s)}{\text{bit area } \left(10^{-3} mm^2\right)} . \tag{13}$$

$$\varepsilon_P = \frac{\text{link bandwidth } (MB/s)}{\text{bit area} \left(10^{-3} mm^2\right) . \hat{L}(cycles)} . \tag{14}$$

5.4 Discussion

In this section, we discuss the reported results inferring the advantages and disadvantages of the different approaches.

5.4.1 Hardware Evaluation

Area Cost: Based on the normalized results in Table 3, the area of CS-routers and TDM-routers are comparable, whereas the area of the HS- routers is two orders of magnitude higher. This is due to the integration of either a CS-PS-router or a TDM-PS-router in the same router. On the other hand, all the PS-router contributions do not report their area cost. Yet, the work of Shi and Burns [27], concerned with decreasing the area overhead of priority-based PS-NoCs, reports improvement of area only in terms of the reduced number of needed VCs due to sharing of priority values without real hardware implementation. However, according to our ART-NoCs framework [103], a PS-Router cost is comparable to an HS-router, since the main expensive cost is that of the VCs. However, it is worth mentioning that TDM and hybrid TDM NoCs need considerably bulky NI cost.

The smallest router bit-area is guaranteed by the dÆlite router [44] which consists of a crossbar, a counter, a TDM table using a simple configuration FSM without the need for arbitration or buffering. dÆlite configures the TDM tables by a central configuration module based on a predetermined sequence of configuration packets that limits the runtime flexibility and the NoC scalability. In addition, the NoC is based on a synchronous platform which further limits the scalability for many-cores. The directly following router in area is the asynchronous Argo router [46] addressing the problem of TDM synchronism.

Bisection Bandwidth per Link: The listed values for the bandwidth per link in Table 3, vary from 1.85 up to 25.6 MB/s. By examining the corresponding frequency values in Table 2, it can be inferred that the vari-ance in the bandwidth per link is not only due to the variance in operating frequency which ranges for the same references between 0.5 to 2 MHz. Rather, all references with bandwidth per link ≥ 10 MB/s have a large bisection datawidth $\geq$ 64-bit. In fact, this is highly aligning with the findings of ARTNoCs [103] stating that increasing the datawidth has negligible impact on the operating frequency of a CS or a TDM router. Thus, increasing the datawidth directly increases the bisection bandwidth since it has no contradicting effects on speed. However, for TDM connections, the maximum used bandwidth per flow is always a fraction of the maximum link bisection bandwidth. This fraction is always < 1 and is based on the number of reserved slots as given by (6) in Section 3.3; otherwise, if $TP_i = TP_{max}$ it will turn into a CS-router. This is also applicable on the SurfNoC [47] and PhaseNoC [48], where the fraction of bandwidth offered for each domain equals the total bandwidth divided by the number of supported domains. In case of only one supported domain, the architecture will turn down to a baseline VC-PS-NoC. The maximum reported bisection bandwidth per link of 25.6 MB/s, is provided by Heisswolf et al. [57]. The architecture uses the adaptive rerouting technique on top of their previously published HS-router [50] and has a datawidth of 256-bit.

Hardware Efficiency (ε_H): Although the adaptive HS-router of Heisswolf et al. [57] provide the highest bandwidth per link, it consumes the largest bit area of $0.294 \times 10^{-3}\mathrm{mm}^2$ as it uses the TDM concept together with VCs. This results in a reduced overall router hardware efficiency by 69 percent compared to the dÆlite router. Yin et al. [53] provide an equivalent bandwidth of 24 MB/s to Heisswolf et al. [57], however they dispense quarter the area in a TDM hybrid-switched router. The dÆlite [44] preserves its position as the top hardware efficiency router, followed by the Argo router [46]. Their small area with relatively reasonable bandwidth keep them on top of the other contributions. In fact, among the routers which support the GALS approach; Argo NoC provides the highest hardware efficiency. Argo almost achieves the same efficiency level of dÆlite with only 8% difference while providing an asynchronous platform of routers. However, taking into account the complexity of the NIs in the bit-area consideration would reduce the per-node hardware efficiency of the dÆlite and Argo to 6.2 and 7.83 ($\mathrm{MB/s/10^{-3}mm^2}$). Nevertheless, such per-node efficiency is not comparable across the references since most of the contributions do not report results about NI area.

5.4.2 Performance Evaluation

Average Latency: The values of the normalized average latency in Table 3 for the TDM approach range from 0.2 to 0.5 except for the SurfNoC [47] and PhaseNoC [48]. This is due to the fact that the two NoCs apply the TDM approach on the phase (domain) of the router stage and not the ports connections. Therefore the domain which is granted access to a certain router stage still applies arbitration between the different flows belonging to the same domain from other input ports. In case a request of a flow loses arbitration at a certain router step, it will be suspended to the next TDM period and not the next cycle; i.e., when the phase of the same stage turns again to the same domain. This increases the average latency in addition to the internal VC-router 4-stage pipeline. The latency is worse for SurfNoC since it does not apply a full explicit wave-like synchronism of TDM phases as discussed in Section 3.3.

The HS-router of Yin et al. [53] uses TDM schedules for each VC and for the switch traversal, with offered online TDM slots allocation. This highly increases the average latency as illustrated in Table 3.

It is further worth to highlight that the range of normalized latencies for the PS-NoCs, CS-NoCs and TDM NoCs is unexpectedly proximate. This comes from the fact that PS-NoCs provide better latencies than CS-NOCs and TDM NoCs with online configuration for short transmitted data and vice versa. This is because for long transmitted-data the dominating term in (10) and (12) will be the transmission latencies while the setup/ configuration time can be neglected. On the other hand, for short transmissions, the setup latency will be the dominating part and may even be higher than the whole PS transmission. Therefore, with tested transmission data of 400 and 128 flits for CS [23], [24] and TDM NoCs [40], [43], [44], [46], and with a packet length of 2-16 flits for PS-NoCs and HS-NoCs, the computed normalized latency values, based on the tested traffic, turn to be in the same range. This is exactly the opposite to the case of the tested traffic of 5-flits packets in the HS-router of Yin et al. [53] with online allocation, which resulted in large average latency values. In addition, localized traffic with short path lengths further enhances the latency for CS-NoCs and TDM-NoCs with online configurations, as can be deduced from (9), (10), and (12). The best normalized latency of 0.22

cycles is achieved by the Argo NoC which uses static TDM tables allocated and configured at design-time.

Performance Efficiency (ε_P): The maximum overall router performance efficiency is provided by the Argo NoC [46]. This highly achievable efficiency comes from the adequate link-bandwidth and the minimum average latency governed by the statically allocated and configured TDM tables. The static TDM tables necessitate well predefined set of applications at design-time.

5.4.3 Real-Time Support

Based on the surveyed contributions in this work, as listed in Table 1 in the "design flexibility" column as well as the last two columns, a RT-NoC must involve a pre-runtime phase to fit for HRTSs.

For CS-NoCs, this involves a pre-runtime scheduling phase to ensure a bounded setup configuration phase. The CS-NoC of Liu et al. [24] is listed as a runtime bounded latency scheme, however, this is constrained by the load of the running traffic. The guaranteed bounded latency is for the path search, while for the setup success, it is bounded until a cross-over point of the traffic load reported to be 0.3 for a 400 cycle data transmission. Therefore, it also requires a pre-runtime scheduling test phase though the path search and setup can be left to runtime. Improving the path setup strategy such as the backtracking [22], the parallel probing [24], even if not bounded, it gives higher chances for wider range of applications schedulability at design time.

For priority-based PS-NoCs, this involves a pre-runtime schedulability analysis. Fixed priority preemptive arbitration allow bounded latency such as in QNoC [31], however, at design-time it must be ensured that this latency meets the deadline for each flow, based on the different flows characteristics and the applications mapping. Traffic shaping [32] and Quota setting [33] are fixed-priority non-preemptive PS-NoCs with runtime handling of low priority load however, according to a pre-runtime defined shaping conditions and quota tables respectively which are in turn dependent on the task mappings and flow characteristics. BiNoC [34] is another fixed-priority PS-NoC with runtime changing architecture based on prioritized routing. It follows preemptive arbitration with fixed priorities, thus can provide bounded latencies. This allows BiNoC to fit for HRTSs upon a pre-runtime schedulability test not hinted by the authors.

TDM-NoCs involve 3 different steps, allocation, configuration and transmission. All reported hard real-time TDM NOCs perform allocation at design-time. Æthereal [40], Ælite [43] and dÆlite [44] perform online configuration with bounded configuration times, while Argo [46], PhaseNoC [48]and SurfNoC use [47] completely static TDM tables. PhaseNoC and SurfNoC guarantee isolation and bounded performance for each domain, however, for contentions within each domain, no hard guarantees can be provided. Thus the different VCs within each domain must be prioritized to provide tighter HRTS guarantees. HS-NoCs would integrate the respective conditions according to the integrated techniques.

Non-restricted dynamic run-time techniques can only fit in SRTSs, such as the dynamic priority schemes in [35], [36], [37], and the online slots allocation in [53].

All switching techniques can provide hard-guarantees. However, based on the target application, schedulability-as

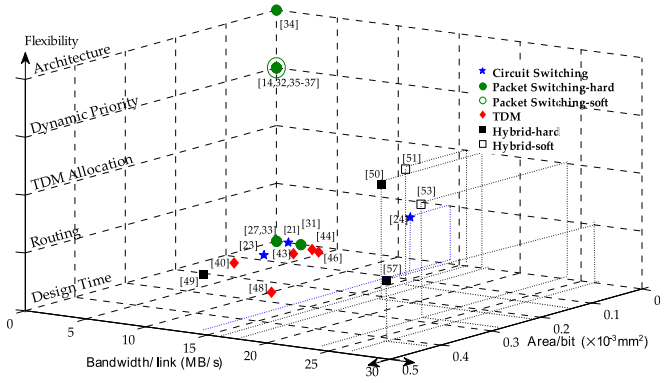

Fig. 6. Architectural comparison in terms of area, bandwidth per link and flexibility support.

the central provision-will define for each approach a minimum cost and performance limitations which will promote one approach over the others. Correspondingly, the quantitative comparison in [25] shows that connection-less approach proves better performance over connection-oriented approach for variable bit-rate applications, in terms of guaranteed end-to-end delay for individual flows and prediction of transmission latency.

Finally, Fig. 6 provides a compact overview for the discussed contributions. It gives an illustrative architectural comparison in terms of flexibility, real-time support, normalized bit area and bandwidth per link. A solid filled marker is a hard-real-time reference, while an empty marker is a soft real-time one. A zero value for the area or the bandwidth indicates a non-specified value.

5.4.4 Future Insights

The surveyed work and discussed performance comparison reveal challenges and future insights for the RT-NoCs field.

As mentioned earlier, energy minimization comes on top of the outstanding challenges in RT-NoCs especially in the existence of the Dark Silicon Era [104] which is tipping the scales of the multicore technology. The effects of the Dark-silicon and how to leverage its existence in RT-NoC based platforms is yet an untackled problem. Furthermore, based on the reviewed results for the RT-energy contributions, we would like to hint that a complete hardware implementation for a reliable, energy efficient RT-NoC is still missing.

Moreover, we highlighted the significance of runtime adaptivity and scalability for future chips. Especially, mixed critical applications can gain a lot from runtime adaptivity. However, providing these features while maintaining H RT-guarantees is still an open research problem.

On the level of the analytical approaches, a mathematical analysis for WCCL using dynamic adaptive routing for of HRT-guarantees is missing in the literature. Furthermore, Nikolić et al. [14] introduced the schedulability of dynamic priorities by using the EDF. However, further studies for mathematical analysis and WCCL using dynamic priorities can lead to better schedulability thresholds in RT-NoCs.

Finally, this review has presented and discussed the wide design space of RT-NoCs. In dealing with this design space, a framework that takes the application requirement and finds a minimum cost RT-NoC solution for each of the switching techniques would simplify the design flow.

Nevertheless, such framework is still missing in the literature. Shi and Burns [27] presented a similar approach using a greedy algorithm, however it only reduces the cost of priority-PS-VC-NoCs in terms of number of VCs and does not necessarily find optimal solution, but only a reduced cost.

6 Conclusion

A comprehensive survey was presented for existing real-time NoC architectures covering QoS support, adaptivity, energy efficient techniques. On the level of architectures, TDM is widely used mainly with global synchronicity for HRT guarantees in CS, PS as well as HS-NoCs. Asynchronous TDM NoCs is a recent proposal that attracts interest for further exploration with an ultimate bandwidth per link approaching the 40 Gb/s. The surveyed work showed that HRT-NoC solutions must involve design-time phase with static characterization and lack of adaptivity or flexibility during runtime. The work further hinted some future insights in this field. This includes techniques to provide high link bandwidth in the range of hundreds of Gb/s in real-time NoCs to cope with the increasing performance of real-time applications. For future research, integrating run-time adaptivity and system reconfiguration with fault tolerance and energy-efficient techniques would make significant contributions in this area. Finally, it is worth to mention that a complete hardware implementation for a reliable, energy efficient real-time NoC is still missing in the literature.

References

[1] Intel® many integrated core architecture–advanced. (2011). [Online]. Available: http://www.intel.com, Accessed on: 2015.

[2] S. Borkar, "Thousand core chips: A technology perspective," in *Proc. 44th Annu. Des. Automat. Conf.*, 2007, pp. 746–749.

[3] W. J. Dally and B. Towles, "Route packets not wires: on-chip interconnection networks," in *Proc. 38th Annu. Design Autom. Conf.*, 2001, pp. 684–689.

[4] W. J. Dally and B. Towles, Principles and practices of interconnection networks, Burlington, MA, USA: Morgan Kaufmann, 2003.

[5] M. A. Abd El Ghany, M. A. El-Moursy and M. Ismail, "High throughput architecture for high performance NoC," in *Proc. Circuits Syst.*, 2010, pp. 2241–2244.

[6] T. Bjerregaard and S. Mahadevan, "A survey of research and practices of Network-on-chip," *ACM Comput. Surveys*, vol. 38, no. 1, 2006, Art. no. 1.

[7] U. Y. Ogras and R. Marculescu, "Literature survey," in *Modeling, Analysis and Optimization of Network-on-Chip Communication Architectures*, vol. 184. Dordrecht, Netherlands: Springer, 2013, pp. 9–32.

[8] A. Abbas, et al., "A survey on energy-efficient methodologies and architectures of network-on-chip," *Comput. Electr. Eng.*, vol. 40, no. 8, pp. 333–347, 2014.

[9] G. Haiyun, "Survey of dynamically reconfigurable network-on-chip," in *Proc. IEEE Int. Conf. Future Comput. Sci. Appl.*, 2011, pp. 200–203.

[10] S. Hesham, J. Rettkowski, D. Goehringer, and M. A. Abd El Ghany, "Survey on real-time network-on-chip architectures," in *Proc. 11th Int. Symp. Appl. Reconfigurable Comput.*, 2015, pp. 191–202.

[11] P. P. Panda, C. Grecu, M. Jones, A. Ivanov and R. Saleh, "Performance evaluation and design trade-offs for network-on-chip interconnect architectures," *IEEE Trans. Comput.*, vol. 54, no. 8, pp. 1025–1040, Aug. 2005.

[12] J. Sparsoe, "Design of networks-on-chip for real-time multi-processor systems-on-chip," in *Proc. 12th Int. Conf. Appl. Concurrency Syst. Des.*, 2012, pp. 1–5.

[13] J. Zhan, N. Stoimenov, J. Ouyang, L. Thiele, V. Narayanan, and Y. Xie, "Designing energy-efficient NoC for real-time embedded systems through slack optimization," in *Proc. 50th ACM/EDAC/IEEE Des. Automat. Conf.*, 2013, pp. 1–6.

[14] B. Nikolić and S. M. Petters, "EDF as an arbitration policy for wormhole-switched priority-preemptive NoCs - myth or fact?," in *Proc. 14th Int. Conf. Embedded Softw.*, 2014, Art. no. 28.

[15] L. S. Indrusiak, "End-to-end schedulability tests for multiprocessor embedded systems based on networks-on-chip with priority-preemptive arbitration," *J. Syst. Archit.*, vol. 60, no. 7, pp. 553–561, 2014.

[16] S. Foroutan, Y. Thonnart, and F. Petrot, "An iterative computational technique for performance evaluation of networks-on-chip," *IEEE Trans. Comput.*, vol. 62, no. 8, pp. 1641–1655, Aug. 2013.

[17] A. E. Kiasari, A. Jatsch and Z. Lu, "Mathematical formalisms for performance evaluation of networks-on-chip," *ACM Comput. Surveys*, vol. 45, no. 3, pp. 38:1–38:40, 2013.

[18] K. Goossens, et al., "Guaranteeing the quality of services in networks on chip," in *Networks on Chip*, A. Jantsch and H. Tenhunen, Eds. New York, NY, USA: Springer, 2003, pp. 61–82.

[19] A. Mello, N. Calazans, and F. Moraes, "QoS in Networks-on-Chip - beyond priority and circuit-switching techniques," in *VLSI-SoC: Advanced Topics on Systems on a Chip*. Berlin, Germany: Springer, 2007.

[20] D. Bertozzi and L. Benini, "Xpipes: A network-on-chip architecture for gigascale systems-on-chip," *IEEE Circuits Syst. Mag.*, vol. 4, no. 2, pp. 18–31, Sep. 2004.

[21] D. Wiklund and D. Liu, "SoCBUS: Switched network on chip for hard real time embedded systems," in *Proc. Int. Parallel Distrib. Process. Symp.*, 2003, pp. 78–85.

[22] P.-H. Pham, J. Park, P. Mau, and C. Kim, "Design and implementation of backtracking wave-pipeline switch to support guaranteed throughput in network-on-chip," *IEEE Trans. VLSI Sys.*, vol. 20, no. 2, pp. 270–283, Feb. 2012.

[23] P. T. Wolkotte, G. J. M. Smit, G. K. Rauwerda, and L. T. Smit, "An energy-efficient reconfigurable circuit-switched network-on-chip," in *Proc. 19th IEEE Int. Parallel Distrib. Process. Symp.*, 2005, pp. 155a–155a.

[24] S. Liu, A. Jantsch, and Z. Lu, "Parallel probing: Dynamic and constant time setup procedure in circuit-switching NoC," in *Proc. Des. Automat. Test Europe Conf. Exhib.*, Mar. 2012, pp. 1289–1294.

[25] M. D. Harmanci, N. P. Escudero, Y. Leblebici, and P. Ienne, "Quantitative modelling and comparison of communication schemes to guarantee quality-of-service in networks-on-chip," in *Proc. IEEE Int. Symp. Circuits Syst.*, 2005, pp. 1782–1785.

[26] Z. Shi and A. Burns, "Real-time communication analysis for on-chip networks with wormhole switching," in *Proc. 2nd ACM/IEEE Int. Symp. Netw.-on-Chip* , 2008, pp. 161–170.

[27] Z. Shi and A. Burns, "Schedulability analysis and task mapping for real-time on-chip communication," *Real-Time Syst.*, vol. 46, no. 3, pp. 360–385, 2010.

[28] A. Racu and L. S. Indrusiak, "Using genetic algorithm to map hard real-time on NoC-based systems," in *Proc. 7th Int. Workshop Reconfigurable Commun. Centric Syst.-on-Chip*, 2012, pp. 1–8.

[29] I. S. Bonilha, O. M. D. Santos, and L. Indrusiak, "Heuristics for mapping real-time applications to NoC-based architectures using genetic algorithms," in *Proc. Brazilian Symp. Comput. Syst. Eng.*, 2014, pp. 144–149.

[30] N. M. S. M. Sayuti and L. S. Indrusiak, "A function for hard real-time system search-based task mapping optimization," in *Proc. IEEE 18th Int. Symp. Real-Time Distrib. Comput.*, 2015, pp. 66–73.

[31] E. Bolotin, I. Cidon, R. Ginosar, and A. Kolodny, "QNoC: QoS architecture and design process for network on chip," *J. Syst. Archit.*, vol. 50, no. 2/3, pp. 105–128, 2004.

[32] J. Diemer, R. Ernst, and M. Kauschke, "Efficient throughput-guarantees for latency-sensitive networks-on-chip," in *Proc. 15th Asia South Pacific Des. Automat. Conf.*, 2010, pp. 529–534.

[33] K. Cheshmi, J. Trajkovic, M. Soltaniyeh, and S. Mohammadi, "Quota setting router architecture for quality of service in GALS NoC," in *Proc. Int. Symp. Rapid Syst. Prototyping.*, 2013, pp. 44–50.

[34] S.-H. Lo, et al., "QoS aware BiNoC architecture," in *Proc. IEEE Int. Symp. Parallel Distrib. Process.*, 2010, pp. 1–10.

[35] E. d. F. Corrêa, L. A. d. P. e. Silva, F. R. Wagner, and L. Carro, "Fitting the router characteristics in NoCs to meet QoS requirements," in *Proc. 20th Annu. Conf. Integr. Circuits Syst. Des.*, 2007.

[36] C.-H. Lu, K.-C. Chiang, and P.-A. Hsiun, "Round-based priority arbitration for predictable and reconfigurable Network-on-Chip," in *Proc. 8th IEEE Int. Conf. Field-Programmable Technol.*, 2009, pp. 403–406.

[37] B. Sudev, L. S. Indrusiak, and J. Harbin, "Network-on-chip packet prioritisation based on instantaneous slack awareness," in *Proc. IEEE 13th Int. Conf. Ind. Informat.*, 2015, pp. 227–232.

[38] B. Sudev and L. Indrusiak, "Low overhead predictability enhancement in non-preemptive network-on-chip routers using Priority Forwarded Packet Splitting," in *Proc. IEEE 9th Int. Symp. Reconfigurable Commun.-Centric Syst.-on-Chip*, 2014, pp. 1–8.

[39] Z. Lu and A. Jantsch, "Slot allocation using logical networks for TDM virtual-circuit configuration for network-on-chip," in *Proc. IEEE/ACM Int. Conf. Comput.-Aided Des.*, 2007, pp. 18–25.

[40] K. Goossens, J. Dielissen, and A. Radulescu, "The Æthereal network on chip: Concepts, architectures, and implementations," *IEEE Des. Test Comput.*, vol. 22, no. 5, pp. 414–421, Sep./Oct. 2005.

[41] N. Concer, A. Vesco, R. Scopigno, and L. Carloni, "A dynamic and distributed TDM slot-scheduling protocol for QoS-oriented Networks-on-Chip," in *Proc. IEEE 29th Int. Conf. Comput. Des.*, 2011, pp. 31–38.

[42] M. Millberg, E. Nilsson, R. Thid, and A. Jantsch, "Guaranteed bandwidth using looped containers in temporally disjoint networks within the nostrum network on chip," in *Proc. Des. Autom. Test Europe Conf. Exhib.*, 2004, pp. 890–895.

[43] A. Hansson, M. Subbaraman, and K. Goossens, "Ælite: A flit-synchronous network on chip with composable and predictable services," in *Proc. Des. Autom. Test Europe Conf. Exhib.*, 2009, pp. 250–255.

[44] R. A. Stefan, A. Molnos, and K. Goossens, "dÆlite: A TDM NoC supporting QoS, multicast, and fast connection set-up," *IEEE Trans. Comput.*, vol. 63, no. 3, pp. 583–594, Mar. 2014.

[45] K. Goossens and A. Hansson, "The aethereal network on chip after ten years: Goals, evolution, lessons, and future," in *Proc. 47th ACM/IEEE Des. Autom. Conf.*, 2010, pp. 306–311.

[46] E. Kasapaki, M. Schoeberl, R. B. Sørensen, C. Müller, K. Goossens, and J. Sparsø, "Argo: A real-time network-on-chip architecture with an efficient GALS implementation," *IEEE Trans. Very Large Scale Integration (VLSI)*, vol. 24 no. 2, pp. 479–492, Feb. 2016.

[47] H. M. G. Wassel, et al., "SurfNoC: a low latency and probably non-interfering approach to secure networks-on-chip," in *Proc. ISCA*, 2013, pp. 583–594.

[48] A. Psarras, I. Seitanidis, C. Nicopoulos, and G. Dimitrakopoulos, "PhaseNoC: TDM scheduling at the virtual-channel level for efficient network traffic isolation," in *Proc. Des. Autom. Test Europe Conf. Exhib.*, 2015, pp. 1090–1095.

[49] T. Bjerregaard and J. Sparso, "A router architecture for connection-oriented service guarantees in the MANGO clockless network-on-chip," in *Proc. Des. Autom. Test Europe Conf. Exhib.*, 2005, pp. 1226–1231.

[50] J. Heisswolf, R. Konig, M. Kupper, and J. Becker, "Providing multiple hard latency and throughput guarantees for packet-switching networks on chip," *Comput. Electr. Eng.*, vol. 39, no. 8, pp. 2603–2622, 2013.

[51] A. Lusala and J. Legat, "A hybrid NoC combining SDM-TDM based circuit-switching with packet-switching for real-time applications," in *Proc. IEEE 10th Int. New Circuits. Syst. Conf.*, 2012, pp. 17–20.

[52] F. Pakdaman, A. Mazloumi, and M. Modarressi, "Integrated circuit-packet-switching NoC with efficient circuit setup mechanism," *J. Supercomputing*, vol. 71, pp. 1–21, 2014.

[53] J. Yin, P. Zhou, S. Sapatnekar, and A. Zhai, "Energy-efficient time-division multiplexed hybrid-switched NoC for heterogeneous multicore systems," in *Proc. IEEE 28th Int. Parallel Distrib. Process. Symp.*, 2014, pp. 293–303.

[54] D. Goehringer, O. Oey, M. Huebner, and J. Becker, "Heterogeneous and runtime parameterizable star-wheels network-on-chip," in *Proc. Int. Conf. Embedded Comput. Syst.*, 2011, pp. 380–387.

[55] M. Ruaro, E. A. Carara, and F. G. Moraes, "Runtime adaptive circuit switching and flow priority in NoC-Based MPSoCs," *IEEE Trans. VLSI Syst.*, vol. 23, no. 6, pp. 1077–1088, Jun. 2015.

[56] J. P. Diguet, "Self-adaptive network on chips," in *Proc. 27th Symp. Integr. Circuits Syst. Des.*, 2014, Art. no. 24.

[57] J. Heisswolf, M. Singh, M. Kupper, R. Konig, and J. Becker, "Rerouting: Scalable NoC self-optimization by distributed hardware-based connection reallocation," in *Proc. Int. Conf. Reconfigurable Comput. FPGAs*, 2013, pp. 1–8.

[58] J. Zhan, et al., "Optimizing the NoC slack through voltage and frequency scaling in hard real-time embedded systems," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 33, no. 11, pp. 1632–1643, Nov. 2014.

[59] M. D. Grammatikakis, A. Papagrigoriou, P. Petrakis, and G. Kornaros, "Non-intrusive NoC DFS for soft real-time multimedia applications," in *Proc. 16th Euromicro Conf. Dig. Syst. Des.*, 2013, pp. 60–63.

[60] H. He, G. Yang, and J. Hu, "Algorithms for power-efficient QoS in application specific NoCs," in *Proc. IEEE/ACM Int. Symp. Low Power Electron. Des.*, 2014, pp. 165–170.

[61] M. N. S. M. Sayuti, L. S. Indrusiak, and A. Gracia-Ortiz, "An optimization Algorithm for minimizing energy dissipation in NoC-based hard real-time embedded systems," in *Proc. 21st Int. Conf. Real-Time Netw. Syst.*, 2013, pp. 3–12.

[62] M. N. S. M. Sayuti and L. S. Indrusiak, "Real-time low power task mapping in networks-on-chip," in *Proc. IEEE Comput. Soc. Annu. Symp. VLSI*, Aug. 2013, pp. 14–19.

[63] J. Palma, L. S. Indrusiak, F. G. Moraes, A. G. Ortiz, M. Glesner, and R. A. L. Reis, "Evaluating the impact of data encoding techniques on the power consumption in networks-on-chip," in *Proc. IEEE Comput. Soc. Annu. Symp. Emerging VLSI Technol. Archit.*, 2006, Art. no. 2.

[64] A. Berman, R. Ginosar, and I. Keider, "Order is power: Selective Packet Interleaving for energy efficient Networks-on-Chip," in *Proc. 18th IEEE/IFIP Int. Conf. VLSI Syst. Chip*, 2010, pp. 37–42.

[65] M. Palesi, G. Ascia, F. Fazzino, and V. Catania, "Data encoding schemes in networks on chip," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 30, no. 5, pp. 774–786, May 2011.

[66] M. Taassori, M. Taassori, and S. Uysal, "MFLP: A low power encoding for on chip networks," *Int. J. DAES*, vol. 20, no. 3, pp. 191–210, 2016.

[67] A. Sheibanyrad, A. Greiner and I. Miro-Panades, "Multisynchronous and fully asynchronous NoCs for GALS architectures," *IEEE Des. Test Comput.*, vol. 25, no. 6, pp. 572–580, Nov./Dec. 2008.

[68] E. Rijpkema, et al., "Trade-offs in the design of a router with both guaranteed and best-effort services for networks on chip," in *Proc. Des. Autom. Test Europe Conf. Exhib.*, 2003, pp. 350–355.

[69] L. S. Indrusiak, J. Harbin, and O. M. Dos Santos, "Fast simulation of networks-on-chip with priority-preemptive arbitration," *ACM Trans. Des. Autom. Electron. Syst.*, vol. 20, no. 4, pp. 56:1–56:22, 2015.

[70] L. Ost, G. Guindani, L. S. Indrusiak, C. Reinbrecht, T. Raupp, and F. Moraes, "A high abstraction high accuracy power estimation model for networks-on-chip," in *Proc. 22nd Annu. Symp. Integr. Circuits Syst. Des. Chip Dunes*, 2009, Art. no. 31.

[71] K. Heid, H. Ying, C. Hochberger, and K. Hofmann, "LatEst: Latency estimation and high speed evaluation for wormhole switched networks-on-chip," in *Proc. 9th Int. Symp. Reconfigurable Commun.-Centric Syst. Chip.*, 2014, pp. 1–7.

[72] A. Butko, et al., "A Trace-driven approach for fast and accurate simulation of Manycore architectures," in *Proc. 20th Asia South Pacific Des. Autom. Conf.*, 2015, pp. 707–712.

[73] J. Xu and D. L. Parnas, "Priority scheduling versus pre-run-time scheduling," *Real-Time Syst.*, vol. 18, no. 1, pp. 7–23, 2000.

[74] J. Diemer and R. Ernst, "A link arbitration scheme for quality of service in a latency-optimized network-on-chip," in *Proc. Des. Autom. Test Europe Conf. Exhib.*, 2009, pp. 574–577.

[75] C. Zeferino and A. Susin, "SoCIN: A parametric and scalable network-on-chip," in *Proc. 16th Symp. Integr. Circuits Syst. Des.*, 2003, 169–174.

[76] F. Moraes, N. Calazans, A. Mello, L. Möller, and L. Ost, "Hermes: an infrastructure for low area overhead packet-switching networks on chip," *Integr. VLSI J.*, vol. 38, no. 1, pp. 69–93, 2004.

[77] K. Goossens, J. Dielissen, O. Gangwal, S. Pestana, A. Radulescu, and E. Rijpkema, "A design flow for application-specific networks on chip with guaranteed performance to accelerate SOC design and verification," in *Proc. Des. Autom. Test Europe*, 2005, pp. 1182–1187.

[78] A. Hansson, M. Coenen, and K. Goossens, "Undisrupted quality-of-service during reconfiguration of multiple applications in networks on chip," in *Proc. Des. Autom. Test Europe Conf. Exhib.*, 2007, pp. 1–6.

[79] A. Hansson, K. Goossens, and A. Rădulescu, "A unified approach to mapping and routing on a network-on-chip for both best-effort and guaranteed service traffic," *VLSI Design*, vol. 2007, 2007, Art. no. 68432.

[80] R. B. Sørensen, J. Sparsø, M. R. Pedersen, and J. Højgaard, "A metaheuristic scheduler for time division multiplexed networks-on-chip," in *Proc. IEEE 17th Int. Symp. Object/Component/Service-Oriented Real-Time Distrib. Comput.*, 2014, pp. 309–316.

[81] H. Zhang, "Service disciplines for guaranteed performance service in packet-switching networks," *Proc. IEEE*, vol. 83, no. 10, pp. 1374–1396, Oct. 1995.

[82] N. Agarwal, T. Krishna, L.-S. Peh, and N. Jha, "GARNET: A detailed on-chip network model inside a full-system simulator," in *Proc. IEEE Int. Symp. Performance Anal. Syst. Softw.*, 2009, pp. 33–32.
[83] Z. Shi, A. Burns, and L. Indrusiak, "Schedulability analysis for real time on-chip communication with wormhole switching," *Int. J. Embedded Real-Time Commun. Syst.*, vol. 1, no. 2, pp. 1–22, 2010.
[84] M. Bekooij, et al., "Dataflow analysis for real-time embedded multiprocessor system design," in *Dynamic and Robust Streaming in and Between Connected Consumer-Electronic Devices*. Dordrecht, Netherlands: Springer , 2005, pp. 81–108.
[85] Y. Qian, Z. Lu, and W. Dou, "Analysis of worst-case delay bounds for best-effort communication in wormhole networks-on-chip," in *Proc. 3rd ACM/IEEE Int. Symp. Netw. Chip*, 2009, pp. 44–53.
[86] B. Nikolić, P. M. Yomsi, and S. M. Petters, "Worst-case communication delay analysis for NoC-based many-cores using a limited migrative model," *J. Signal Process. Syst.*, vol. 84, no. 1, pp. 25–46, 2015.
[87] E. A. Rambo and R. Ernst, "Worst-case communication time analysis of networks-on-chip with shared virtual channels," in *Proc. Des. Autom. Test Europe Conf. Exhib.*, 2015, pp. 537–542.
[88] C. Bienia, S. Kumar, J. S. Singh, and K. Li, "The PARSEC benchmark suite: Characterization and architectural implications," in *Proc. 17th Int. Conf. Parallel Archit. Compilation techn.*, 2008, pp. 72–81.
[89] S. Che, et al., "Rodinia: A benchmark suite for heterogeneous computing," in *Proc. IEEE Int. Symp. Workload Characterization*, 2009, pp. 44–54.
[90] A. Bakhoda, G. Yuan, W. Fung, H. Wong and T. Aamodt, "Analyzing CUDA workloads using a detailed GPU simulator," in *Proc. IEEE Int. Symp. Performance Anal. Syst. Softw.*, 2009, pp. 163–174.
[91] V. Aslot, M. J. Domeika, R. Eigenmann, G. Gaertner, W. B. Jones, and B. Parady, "SPEComp: A new benchmark suite for measuring parallel computer performance," in *Proc. Int. Workshop OpenMP Appl. Tools: OpenMP Shared Memory Parallel Program.*, 2001, pp. 1–10.
[92] S. Woo, M. Ohara, E. Torrie, J. Singh, and A. Gupta, "The SPLASH-2 programs: Characterization and methodological considerations," in *Proc. 22nd Annu. Int. Symp. Comput. Archit.*, 1995, pp. 24–36.
[93] N. Hardavellas, M. Ferdman, B. Falsafi, and A. Ailamaki, "Toward dark silicon in servers," *IEEE Micro*, vol. 31, no. 4, pp. 6–15, Jul. 2011.
[94] T. Ye, L. Benini, and G. De Micheli, "Analysis of power consumption on switch fabrics in network routers," in *Proc. 39th Annu. Des. Autom. Conf.*, 2002, pp. 524–529.
[95] A. Kahng, B. Li, L.-S. Peh, and K. Samadi, "ORION 2.0: A fast and accurate NoC power and area model for early-stage design space exploration," in *Proc. Conf. Des. Autom. Test Europe*, 2009, pp. 423–428.
[96] J. Fong, S. Nath, A. B. Kahng, and B. Lin, "ORION3.0: A power-performance simulator for interconnection networks." (2012). [Online]. Available: http://vlsicad.ucsd.edu/ORION3/
[97] N. Jiang, et al., "A detailed and flexible cycle-accurate network-on-chip simulator," in *Proc. IEEE Int. Symp. Performance Anal. Syst. Softw.*, 2013, pp. 86–96.
[98] OPNET Technologies. (2000). [Online]. Available: http://www.opnet.com
[99] C. Sun, et al., "DSENT - A Tool Connecting Emerging Photonics with Electronics for Opto-Electronic Networks-on-Chip Modeling," in *Proc. IEEE/ACM 6th Int. Symp. Netw. Chip*, 2012, pp. 201–210.
[100] "International Technology Roadmap for Semiconductors," 2012. [Online]. Available: http://www.itrs.net/reports.html.
[101] E. Salminen, E. Kulmala, and T. D. Hämäläinen, "On network-on-chip comparison," in *Proc. 10th Euromicro Conf. Dig. Syst. Des. Archit. Methods Tools*, 2007, pp. 503–510.
[102] J. Sparsø, E. Kasapaki, and M. Schoeberl, "An Area-efficient Network Interface for a TDM-based Network-on-Chip," in *Proc. Des. Autom. Test Europe Conf. Exhib.*, 2013, pp. 1044–1047.
[103] S. Hesham, D. Goehringer, and M. A. Abdel-Ghany, "ARTNoCs: An evaluation framework for hardware architectures of real-time NoCs," in *Proc. IEEE Int. Parallel Distrib. Process. Symp. Workshops*, 2016, pp. 259–264.
[104] H. Bokhari, H. Javaid, M. Shafique, J. Henkel, and S. Parameswaran, "Malleable NoC: Dark silicon inspired adaptable Network-on-Chip," in *Proc. Des. Autom. Test Europe Conf. Exhib.*, 2015, pp. 1245–1248.

Salma Hesham received the BSc and MSc degrees in electronics engineering from the German University, Cairo (GUC), Egypt, in 2010 and 2012, respectively. Currently, she is working towards the PhD degree in a bilateral frame between the Electronics Department, GUC, Egypt and the MCA (application-specific Multi-Core Architectures) research group at the Ruhr-University Bochum (RUB), Germany. Her current research interests include real-time networks-on-chip, multiprocessor systems-on-chip (MPSoCs), hardware design of digital circuits and systems and asynchronous circuits design.

Jens Rettkowski received the BSc degree in microtechnology from Westphalian University of applied Science, in 2011 and the MSc degree in electrical engineering and information technology from the Ruhr-University Bochum (RUB), Germany, in 2014. He is working toward the PhD degree at the MCA (application-specific Multi-Core Architectures) research group from the Ruhr-University Bochum, Germany. His research interests are self-adaptive networks-on-chip, multiprocessor systems-on-chip (MPSoCs) and reconfigurable computing.

Diana Goehringer received the master and PhD degrees in electrical engineering and information technology from the Karlsruhe Institute of Technology (KIT), Germany, in 2011 and 2006, respectively. She is an assistant professor and head of the MCA (application-specific Multi-Core Architectures) research group with the Ruhr-University Bochum (RUB), Germany. She is author and co-author of more than 70 publications in international journals, conferences and workshops. Additionally, she serves as technical program committee member in several international conferences and workshops. She is reviewer and guest editor of several international journals. Her research interests include reconfigurable computing, multiprocessor systems-on-chip (MPSoCs), networks-on-chip, hardware-software-codesign and parallel programming models.

Mohamed A. Abd El Ghany received the BS (honors) degree in electronics and communications engineering and the masters degrees in electronics engineering from Cairo University, Cairo, Egypt, in 2000 and 2006, respectively, and the Ph.D. degree in the area of high-performance VLSI/IC design from the German University, Cairo, (GUC) Egypt, in 2010. From 2003 to 2006, he was in National Space Agency of Ukraine, EGYPTSAT-1 project. From 2008 to 2009, he was an International Scholar at the Ohio State University, Electrical Engineering Department, Columbus. From 2012 to 2014, He awarded the Alexander von Humboldt Foundation Postdoctoral Fellowship, TU, Darmstadt, Germany. He is currently working as an assistant professor in GUC, Egypt. He is a project manager for two international projects between TU Darmstadt, Ruhr-Universität Bochum and GUC. His research interest is in network on chip design and related circuit level issues in high performance VLSI circuits, clock distribution network design, digital ASIC design, SoC/NoC design and verification, low-power design and embedded system design. He is the author of about 30 papers, two book chapters, two books in the fields of high throughput and low-power VLSI/IC design and NoC/SoC. He is a reviewer and program committee member of many IEEE intl. journals and conferences.