Survey of Network-on-chip Proposals

Erno Salminen, Ari Kulmala, and Timo D. Hämäläinen
Tampere University of Technology, P.O. Box 553, FIN-33101 Tampere, Finland

Abstract

This paper gives an overview of state-of-the-art regarding the network-on-chip (NoC) proposals. NoC paradigm replaces dedicated, design-specific wires with scalable, general purpose, multi-hop network. Numerous examples from literature are selected to highlight the contemporary approaches and reported implementation results. The major trends of NoC research and aspects that require more investigations are pointed out. A packet-switched 2-D mesh is the most used and studied topology so far. It is also a sort of an average NoC currently. Good results and interesting proposals are plenty. However, large differences in implementation results, vague documentation, and lack of comparison were also observed.

Keywords: network-on-chip, literature study, router, topology, comparison, implementation

CONTENTS

I. INTRODUCTION

System-on-chip (SoC) architectures are getting communication-bound both from physical wiring and distributed computation point of view. Wiring delays are becoming dominating over gate delays, which favors short links. The larger SoC the more probably the overall computation is heterogeneous and localized rather than evenly balanced over the chip. These two factors motivate Network-on-Chip (NoC) that brings the techniques developed for macro-scale, multi-hop networks into a chip.

NoCs have been widely reported in several special issues in journals, numerous special sessions on conferences and recently also a dedicated NoC symposium [1]. As an example of increased interest, Fig. 1 shows the hit count for search "network-on-chip" in IEEE Xplore document archive [2].

The early work and basic principles of NoC paradigm were outlined in various seminal articles, for example [99][34][35][25] [8][53][97][91][36] [72][73] and few text books [44][69][9]. However, the aforementioned sources do not present many implementation examples or conclusions about the current proposals. This article complements the surveys [12] [87] by providing a categorized, in-depth literature study of NoC proposals and their implementation results.

We analyzed well over 100 peer reviewed NoC publications and created classification of general properties. We present extensive comparisons for three purposes: to show major trends, what kind of details has been reported and what we should conclude for the future of NoC. Furthermore, this paper identifies the aspects calling for further research.

We have made a deliberate choice not to point out what we perceive as mistakes in others' work. Instead, we describe the potential pitfalls in general terms and without pointing to the sources.

II. COMPARISON CRITERIA

The major goal of communication-centric design and NoC paradigm is to achieve greater design productivity and performance by handling the increasing parallelism, manufacturing complexity, wiring problems, and reliability. The three critical challenges for NoC according to Owens *et al.* are: power, latency, and CAD compatibility [73]. The key research areas in network-on-chip design can be summarized, for example [72][12], as

- Communication infrastructure: topology and link optimization, buffer sizing, floorplanning, clock domains, power
- Communication paradigm: routing, switching, flow control, quality-of-service, network interfaces
- Benchmarking and traffic characterization for design- and runtime optimization
- Application mapping: task mapping/scheduling and IP component mapping.

A NoC consists of routers, links, and network interfaces. Routers direct data over several links (hops). Topology defines their logical lay-out (connections) whereas floorplan defines the physical layout. The function of a network interface (adapter) is to decouple computation (the resources) from communication (the network). Routing decides the path taken from source to the destination whereas switching and flow control policies define the timing of transfers. Task scheduling refers to the order in which the application tasks are executed and task mapping defines which processing element (PE) executes certain task. IP mapping, on the other hand, defines how PEs and other resources are connected to the NoC. An excellent text book about network basics in general is [24].

For illustrative purposes, Fig. 2 shows an example SoC with a NoC and nine heterogeneous IP blocks that are CPUs, memories, input/output devices, and HW accelerators. The size of one resource is, for example, 50-200 kilogates or larger. This means that tens of resources are possible in a single chip with modern $65\ nm$ processing technology. The good old shared bus is still very common in practical SoC implementations so it is well motivated to have a deeper look at NoC proposals and figure out what really matters.

Despite vast literature on the topic, most articles do not give clear definition what they mean with the term *NoC*. Sometimes it *seems* to mean only packet-switched mesh-like networks, but we do not make any limitation by topology or switching in

---

[1] www.nocs.org

[2] http://ieeexplore.ieee.org

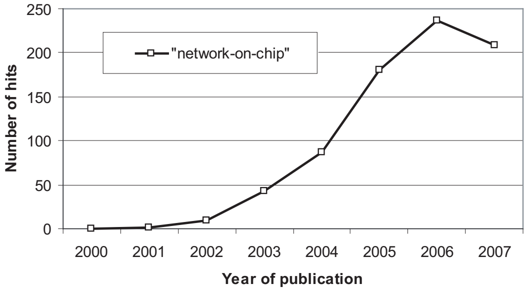

Fig. 1. Search hits for "network-on-chip" in IEEE Xplore archive.

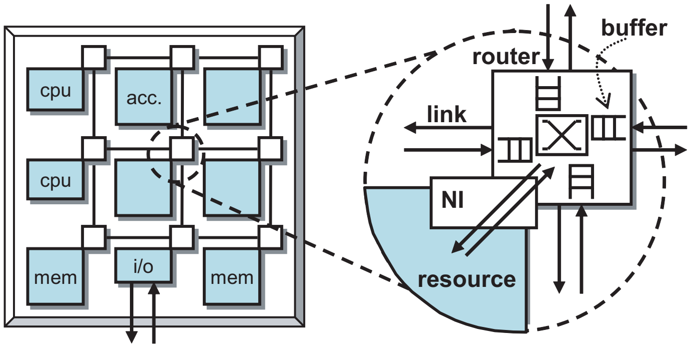

Fig. 2. An example SoC that has a 2-D mesh NoC with 9 resources. Network interface is denoted with NI.

this paper. Unfortunately, many papers give an impression that a "NoC" magically improves every design where it is utilized, but fail to show that or to charactecterize the minimal requirements of a NoC. To avoid such hype, this paper uses a simple, although loose, definition that "*network-on-chip is a communication network that is used on chip*". In [8], the network is the abstraction of the communication among components and must satisfy quality-of-service requirements, such as reliability, performance, and energy bounds.

For comparison, we consider architectural features and implementation specific figures of merit as well as the validation for each paper.

Due to limited space, the focus is on the design and implementation of network components and only single reference per NoC. Other aspects like design automation [10][59][72][74] [45], performance evaluation [75][33][87], encoded signaling [56][43], error detection and correction [67][11] wire optimization [56] [22][70][65], and programming models [77] are covered in more detail in the referenced publications. Interested readers are encouraged to study also the NoC-related on-line resources, especially OCP-IP's NoC bibliography **http://www.ocpip.org/university/biblio_main**. Another bibliography is collected by R. Mullins, Univ. Cambridge [3].

III. NoC ARCHITECTURE COMPARISON

Extensive summary of NoC proposals is listed in Table I. Topology, routing, switching scheme, as well as utilized evaluation methodology and criteria are given for each NoC. The bottom rows show the number and percentage of the papers reporting/using the given property. Furthermore, the percentages from another survey, [87], are also given.

Each property is explained in more detail in the following. Table lists support of various configurations if the papers explicitly report it. Any unclear or partially applying properties are marked in parentheses.

Although very extensive, Table I cannot be all-inclusive. However, we argue that the selected publications offer a representative view of NoC design.

A. Switching policy

Circuit-switching forms a path from source to destination prior to transfer by reserving the routers (switches) and links. All data follow that route and path is torn down after the transfer has completed. Packet-switching performs routing per-packet basis. The upper part of Table I lists the *circuit-switched* (c in the table) and lower part shows *packet-switched* (p) networks, both in alphabetical order. Switching policy is not clearly stated in all cases and sometimes both schemes are supported.

Packet-switching is more common and it is utilized in about 80% of the studied NoCs. Some sources assume packet-switching as key property of NoCs but we will consider also the circuit switched networks.

There are three choices how packets are forwarded and stored at routers: *store-and-forward*, *cut-through*, and *wormhole*, as shown in the next column. Store-and-forward (s) method waits for the whole packet before making routing decisions whereas cut-through (c) forwards the packet already when the header information is available. Both methods need buffering capacity for one full packet at minimum.

Wormhole switching (w) is the most popular and well suited on chip. It splits the packets into several *flits* (flow control digits). Routing is done as soon as possible, similarly to cut-through, but the buffer space can be smaller (only one flit at smallest). Therefore, the packet may spread into many consecutive routers and links like a worm.

Circuit-switching is best suited for predictable transfers that are long enough to amortize the setup latency, and which require performance guarantees. Circuit-switching scheme also reduces the buffering needs at the routers. Packet-switching necessitates buffering and introduces unpredictable latency (jitter) but is more flexible, especially for small transfers. It is an open research problem to analyze the break-even point (predictability and duration of transfers) between the two schemes.

---

[3] http://www.cl.cam.ac.uk/ rdm34/onChipNetBib/browser.htm

TABLE I

EXTENSIVE SUMMARY OF NETWORK-ON-CHIP PROPOSALS IN LITERATURE.

#	Network	Author	Ref.	Switching			Topologies $^{(2)}$					Rout.	Eval. method							Evaluation metrics								
				circuit/packet	$^{(1)}$s/c/w/d	Guarantees	bus, crossbar	mesh/torus	fat-tree/tree	(hier.) ring	custom	det. / adapt.	Analytical	Simulation	Synthesis	FPGA $^{(3)}$	ASIC $^{(3)}$	Applications	Comparison	area	frequency	latency	power/energy	throughput	runtime	utilization	other	# criteria
1	AET	Valtonen	[94]	c?	-	x		m				(d)	x		x			-	-	x	x						1	3
2	aSOC	Liang	[60]	c	-	x		m				(d)		x	x			5	x	x	x				x			3
3	Crossroad	Chang	[18]	c	-	x					c	(d)		x	x		5	3	x	x		x	x		x			4
4	dTDMA	Richardson	[82]	c	-	x	b					(d)	x	x				-	x		x	x						2
5	HIBI	Salm., Kulm.	[85,52]	c,p	(w)	x	hb					(d)	x	x	x	58		3	x	x	x	x		x	x	x		6
6	Hu	Hu	[40]	c	-	x					p	(d)		x	x			1	x	(x)			x				1	3
7	Nexus	Lines	[60]	c	-	x	x					(d)					16	-	-	x	x	(x)	x					4
8	Pnoc	Hilton	[38]	c	-	x					c	(d)		x	x	8		1	x	x	x				x			3
9	ProtoNoc	Castells-Rufus	[17]	c	-	x		m				(d)	x		x			-	-	x				x				2
10	SDM NoC	Leroy	[58]	c	-	x		(m)			(c)	(d)		(x)	x			1	x	x	x		x					3
11	SocBus	Wikl., Henr.	[98,37]	c	-	x		m				(d)		x	x		2	2	x	x	x	x		x				4
12	Wolkotte	Wolkotte	[100]	c	-	x		m				(d)		x	x			-	x	x	x		x					3
13	Aethereal	Rijpkema	[83]	p,(c)	w	x		(m)			(c)	n/a			x			-	-	x	x							2
14	Ahmad	Ahmad	[1]	p,c	w	x		t				a		x	(x)			-	-	(x)		(x)		x				3
15	ANOC	Beigne	[7]	p	w	x		(m)			(c)	a			x			-	-	x	x							2
16	Arteris	Arteris	[5]	p	w,s	-					c	n/a	x					-	x	x	x	x	x					4
17	Bartic	Bartic	[6]	p	c	-	x	m,t,h	tr			d*			x			1	x	x								1
18	Blackbus	Anjo	[4]	p	s	?		(m)			(c)	any		x				7	-								1	1
19	Bouhraoua	Bouhraoua	[15]	p	w	(x)			f			a		x	x			-	-	x		x						2
20	Butterfly FT	Pande	[76]	p	w				ftr			a		x	x			-	-	x				x				2
21	Carloni	Carloni	[16]	p	n/a	-					p	n/a	x					1	-					(x)				1
22	CDMA NoC	Kim, D.	[49]	p	n/a	-	x	sm				n/a	x		x			-	-	x								1
23	CDMA NoC(2)	Xin Wang	[101]	p	n/a	x	x					n/a		x	x			-	x	x		x	x					3
24	Chi, H-C	Chi, H-C	[19]	p	s	x		t, (m)				d		x			1	-	x	x	x	x						3
25	Cliche	Kumar	[53]	p	n/a	-		m				d		x				-	-								1	1
26	Crossbow	Crossbow	[23]	p	n/a	x		m				n/a			x	x		-	-	x	x							2
27	DyAD	Hu	[41]	p	w	-		m				d+a		x	x			1	x	x		x						2
28	Eclipse	Forsell	[30]	p?	s?	-		sparse m				d?	x	x				5	x						x	x		2
29	Ext. mesh	Ogras	[71]	p	w	-		em				d*	x	x	x	16		2	x	x		x	x			x		4
30	Faust	Lattard	[54]	p	w	x		m				d					23	1	-	x	(x)	(x)	x	x				5
31	Feero	Feero	[28]	p	w	-		3m				n/a	x	x	x			-	x	x		x	x	x				4
32	Felicijan	Felicijan	[29]	p	w	x		m				d		x				1	-			x		x			1	3
33	Gezel	Ching	[20]	p	w	-		(t)			(c)	d, (a)		x	x			-	x	x	x	x						3
34	GDB	Hosseinabady	[39]	p	w	-					db	a		x	x			-	x	x	x	(x)	(x)					4
35	Hermes	Moraes	[64]	p	w	-		m				d,a		x	x	4		-	-	x	x	x		x				4
36	Kavaldjiev	Kavaldjiev	[48]	p	w	x		m				d,a		x	x			1	-	x	x	x						3
37	Kim, J.	Kim, J.	[51]	p	w	-		m,t				a		x				-	x			x	x					2
38	Lee, H.G.	Lee, H.G.	[55]	p	n/a	-					c,p	d	x	x	x	8		1	x	x		x	x		x	x		5
39	Lochside	Mullins	[66]	p	w	-		m				d					16	-	-	(x)	x	x						3
40	Mango	Bjerregaard	[13]	p	w	x		m?				d			x			-	-	x	x							2
41	Microspider	Evain	[27]	p	w	x				er		d			x			-	(x)	x	x							2
42	Mocres	Janarthanan	[42]	p	c	-		m				d		x	x			-	x	x	x	x						3
43	Mondinelli	Mondinelli	[63]	p	s	-			ftr			d					64	-	-		x			x				2
44	Nostrum	Millberg, Pen.	[62,80]	p	s	x		m				a		x				1	-			x	x					2
45	Octagon	Karim	[46]	p,c	n/a	x				er		d	x	x				-	x			x					2	3
46	Orion/Luna	Soteriou	[92]	p	w,c	-		em,t,h	ftr	r,er		d,a		x	x			30	x	x		x	x					3
47	Pavlidis	Pavlidis	[78]	p	w	-		3m				d	x	x				-	x			x	x					2
48	PMCNOC	Wang	[96]	p	*	-		m				d		x	x			-	x	x	x	x		x				4
49	Proteo	Sigue., Ahon.	[90,2]	p	c	-				(r)	(c)	d		x	x			1	-	x								1
50	Qnoc	Bolotin, Rost.	[14, 84]	p	w	x		m				d	x	x	x			-	x	x		x	x			x		4
51	Ring road	Samuelsson	[88]	p	s	-				r		n/a		x				-	-							x		1
52	Slim-Spider	Lee, K.	[56]	p,c	n/a	x	x					(d)	x	x			9	-	x	x	x		x					3
53	SNA	Lee, S.	[57]	p	n/a	-	x	sm				n/a		x				1	x						x			1
54	SocIn/RaSoc	Zeferino	[103]	p	w	-		m				d			x			-	-	x	x							2
55	SPIN	Guerr., Andr.	[34, 3]	p	w	-			ftr			a	(x)	x	x			-	x	x		x		x				3
56	Star-mesh	Cidon	[21]	p	w	-		sm				d	x	x				-	x					x				1
57	TeraFLOPS	Vangal	[95]	p	w	-		m				d,a					80	4	x	x	x	(x)	x				1	5
58	XGFT	Kariniemi	[47]	p	w	-			ftr			a		x	x			-	x	x	x	x		x			1	5
59	Xpipes	Bertozzi	[10]	p	w	-					c	d		x	x			4	x	x		x	x				1	4
60	Xu	Xu	[102]	p	n/a	-	x					(d)	(x)	x	x			1	x	x	x		x		x		1	5
	# reported	60	60	60	50	28	9	37	7	5	8	51	18	43	39	6	9	24	36	46	30	33	21	15	8	6	10	60
	% reported	100	100	100	83	47	15	62	12	8	13	85	30	72	65	10	15	40	60	77	50	55	35	25	13	10	17	100
	% in [54]	44 papers	-	-	-	-	72	59	27	11	27	-	20	82	55	9	0	30	100	55	25	41	39	30	57	-	30	-

Notes: $^{(1)}$ w= wormhole, c=cut-through, s=store-and-forward

$^{(2)}$ see Table I for topology symbols

$^{(3)}$ number of NoC terminals in largest prototype

B. Topology

The most common topologies are 2-D mesh and torus which constitute over 60% of cases. Both have connections between 4 neighbor nodes but torus has wraparound links connecting the nodes on network edges and mesh does not. Custom, fat-tree, and crossbar have roughly even proportion followed by ring-based topologies. Otherwise similar distribution was observed in [87] except that buses were used commonly as a reference instead of multi-hop topologies. Table II lists the used topology symbols.

A crossbar (or star) is a non-blocking network with high performance. However, high implementation cost restricts the usage for local communication only instead of large systems. CDMA NoCs [49][101] are here considered as crossbars. Hierarchically constructed networks, for example Slim-spider (hierarchical star) [56], star-mesh [21], and SNA [57], differentiate local and global topologies. This way the topology better matches the differences between local and global communication.

Unlike general-purpose macronetworks, NoCs can be tailored according to the requirement of the application domain. Application-specific networks can obtain superior performance while minimizing both area and energy [40][10]. A topology can also be extended by systematically adding links between non-neighboring nodes [71][46][27]. Few approaches are customizable but the results are given for one topology only [4][7][20][58][90]. However, customization is contradictory to the early projections of simplified layout and wiring optimization that are possible with regular topologies [35][25][36]. Regular structure suites especially well general-purpose chip-multiprocessors (CMPs) with homogeneous resources [95]. Resources in contemporary SoCs, on the other hand, have often varying size and shape, see for example [40][53][10][31], which should be accounted in topology/floorplan analyses.

Physical restrictions in IC layout favor the utilization of 2-dimensional topologies and all but 3 papers in Table I have two-dimensional topology. However, Soteriou *et al.* found the 3-D mesh most suitable NoC in many cases [92]. Moreover, integrating ICs in 3-D fashion clearly favors 3-D NoCs [28][78].

It was found in the survey [87] that 3 networks are compared on average and the largest studies included 9 [56] and 12 networks [92]. Although 5 network sizes were included on average, half of the cases considered only one size (number of terminals). Hence, more disciplined evaluation is called for. It is worth noting no single fixed topology categorically outperforms all others. The properties of basic topologies are quite well-known but customization and IP mapping emphasize the importance of efficient design automation tools, application modeling, and further research.

C. Routing

With *deterministic* routing (d in the table), all packets follow the same route between given source-destination pair. With *adaptive* schemes (a) the route can vary. They reduce congestion and increase fault-tolerance but require special attention to avoid deadlocks. The studied circuit-switched NoCs do not commonly discuss how the circuits are setup. However, the transmitted data remains in-order since the circuit stays intact throughout the transfer, hence the notation (d).

Packet-switched networks mostly utilize deterministic routing (about 70 %) but adaptivity or some means for reprogramming the routing policy is necessary for fault-tolerance. Hermes supports selection of the scheme at design time [64]. Updating the routing tables at runtime which allows adaptivity event though the routing is deterministic (marked $d*$) [6][71]. DyAD [41] switches automatically between deterministic and adaptive at runtime and guarantees freedom from deadlock and livelock. Deflection routing forwards packets every cycle and uses adaptive misrouting (other than minimal path) in case that target

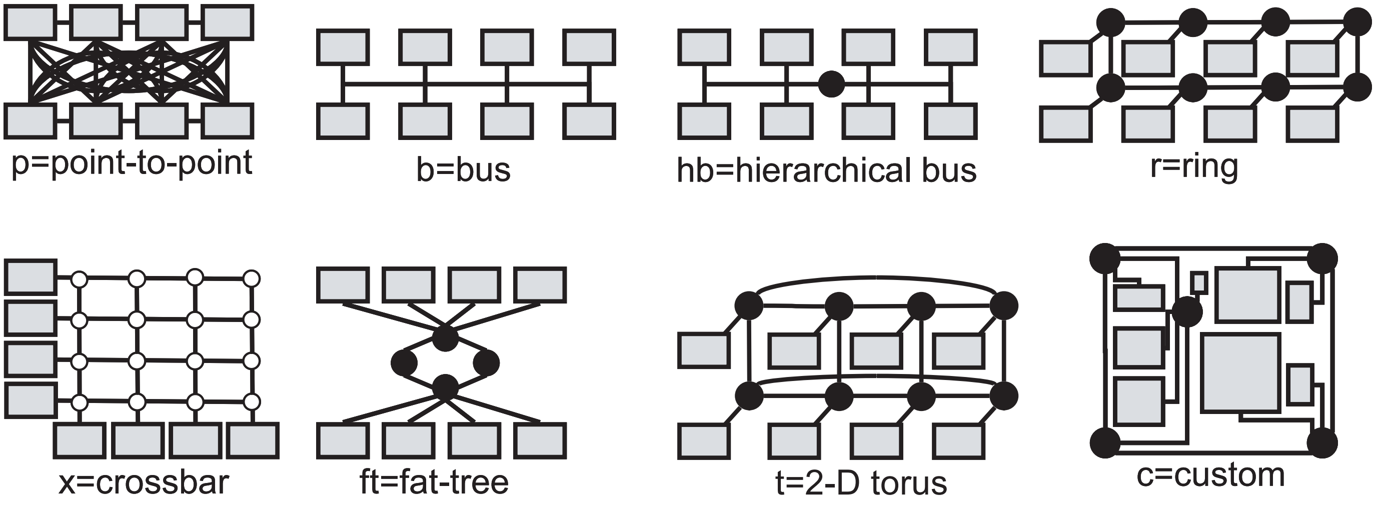

Fig. 3. Some basic network topologies. All examples have 8 terminals and bidirectional links.

TABLE II
THE TOPOLOGY SYMBOLS USED IN OTHER TABLES

Symbol	**Topology**
b, hb	(single shared) bus, hierarchical bus
x	crossbar
m, t	2-D mesh, 2-D torus
em, sm	2-D extended mesh, 2-D star-mesh
3m, h	3-D mesh, 3-D hypercube (i.e. torus)
tr, ftr	tree, fat-tree
r, er	ring, extended ring
c	custom
(c)	custom, but only 1 topol. is reported
p	point-to-point
db	de Bruijn graph

direction is occupied [62]. An interesting option is to split the traffic across several paths to reduce congestion on certain area of the network [10].

Although deadlock is generally avoided, out-of-order packet delivery is problematic with adaptive routing. Many sources neglect this phenomenon totally and others, for example [34][62], assume that network interface or software performs the re-ordering. The cost of reordering in terms of runtime and area overheads are unfortunately neglected which we find as major deficiency.

D. *Quality-of-Service*

Quality-of-service (QoS) refers to the levels of guarantees given for data transfers. Guarantees are related to timing (min. throughput, max. latency, max. latency jitter), integrity (max. error rate, max. packet loss), and packet delivery (in-order or out-of-order). Over half of the articles promise guarantees and concentrate mostly on timing aspects. Three methods are employed to give timing guarantees: network dimensioning, circuit-switching, and prioritized packet scheduling [61].

a) *Best effort* (BE) scheme forwards packets as soon as possible but no guarantees are given for latency or throughput in general case. This is the most common approach nowadays. If packet injection to network is restricted by the NI, a (loose) upper bound can be determined for the network latency but not for the waiting time at the NI. Prioritizing packets offers relative guarantees, for example lower latency for high-priority packets, but no exact guarantees either.

b) *Guaranteed throughput* (GT) scheme can offer minimum level of transfer capability through network. The fundamental feature of GT is the necessity of *a priori* knowledge about the traffic load conditions. For example, the network size (link width, topology, buffer sizes) is set to tolerate the *estimated worst case* conditions. Similarly, simultaneously activated circuits must be statically determined. Otherwise, the circuit setup may fail at runtime and lead to non-deterministic delay prior to transfer. Therefore, the complexity of traffic modeling and system simulation is high, and no guarantees can be given to new applications [61]. Almost half the articles mention QoS but only few actually discuss the implementation in detail or evaluate its efficiency.

More research is needed to obtain accurate, representative traffic models that can be used in benchmarking and in NoC design. Conservative models are especially critical when designing real-time systems. At the same, guaranteed services should coexist with BE and provided with minimal overhead. Most of QoS issues are coupled with routing and flow control policies.

E. *Testing and fault-tolerance*

Testing aims to detect the errors that occurred during fabrication, such as stuck-at-faults or short circuits. Fault-tolerance means the ability to operate in the presence of faults, either hard or soft. Link testing must be addressed in addition to NoC routers [31][74]. At the same time, test mechanism must meet the time, power, coverage, and temperature constraints. Using already tested routers to deliver test data and exploiting the NoC's parallelism for simultaneous testing of multiple nodes is presented in [32]. A special self-test mechanism for NoC was presented in [47].

There are five key elements to tolerate faults: avoidance, detection, containment, isolation, and recovery. Fault-tolerance can be achieved via error detection and correction, stochastic communication, adaptive routing, and both temporal and spatial redundancy [32]. Unfortunately, these issues are mostly neglected in the studied papers, except [60] and [47]. Decreasing feature size, higher frequencies, and increased process variation expose the modern SoC to various faults and countermeasures must be actively sought and studied.

IV. NoC EVALUATION METHODS

Simulation and *synthesis* are clearly the most popular evaluation methods. One third of the studies use analytical methods and one fourth have presented prototype chips. The simulation evidently leads to some simplifications and inaccuracies [87].

Such phenomena could be avoided with real prototypes although measurements are usually more complex than with a simulator and the costs are higher. Note that a prototype offers a valuable reference point for calibrating the simulation and analytical models.

Some reported prototypes omit the computation resources or results from real applications. Two ASIC examples include the network only; SocBus with 2 [37] and Nexus with 16 terminals [60]. Slim-Spider chip [56] includes the NoC and 9 resources and the same hierarchical star was used in object recognition system [50]. FAUST chip is targeted for baseband processing and it includes 23 IPs and an asynchronous 2-D mesh NoC [54].

Synthesizable traffic generators can be utilized in simulators but also if the complete system does not fit into (single) chip. For example, Nostrum NoC was prototyped with 16 traffic generators [68] and an ASIC prototype of 64-node fat-tree with traffic generators was presented in [63]. By far the largest NoC chip to date is the Intel's TeraFLOPS that has an 80-node 2-D mesh [95]. The chip is fabricated at $65\ nm$ technology, designed to runs at $4\ GHz$, and contains 100 million transistors occupying $275\ mm^2$ silicon area.

Modern FPGA devices are large enough for multiprocessor SoC implementation in the range of million gates and notably cheaper than ASICs in small series. They also allow special hardware structures such as monitors and traffic generators, which cannot be included in product ASICs. Two of the reported FPGA studies use 4 processing elements (PEs) but do not provide any application results [2][64]. An MPEG-2 encoder in FPGA with 8 hardwired PEs and either custom mesh or point-to-point network is presented in [55]. An FPGA-based MPEG-4 encoder with multiple Nios II processors, HW accelerators, and hierarchical bus has been presented in [85][52]. There can be 16 processors on a single FPGA and up to 35 processors with 23 other IPs when three FPGA boards are used.

In summary, prototypes are too scarce and lack concrete results from real applications. In our experience, FPGA prototypes are very valuable to ensure correct functionality of NoC, test the application, and include various overheads. Furthermore, the obtained (near) real-time execution of parallel applications allows longer runs and better credibility of the results. Hence, more effort towards prototyping and standardized evaluation methods in NoC community is strongly encouraged.

At the first stages, it is essential to perform verification with small systems, such as 4-16 terminals common nowadays. Nevertheless, NoCs must be utilized in their anticipated scale, which contains dozens of terminals, to properly analyze them. Note also that mere FPGA synthesis is not the same as running full applications on the FPGA (which naturally includes synthesis).

A. Evaluation metrics

The most important metrics for NoCs are *application runtime*, *silicon area*, *power consumption*, and *latency*. All these are to be minimized and usually appropriate trade-off is sought [87].

The required *silicon area* is the most commonly reported value (77%) followed by *latency* (55%) and maximum *operating frequency* (50%). The other metrics have lower occurrence. Category *others* denotes metrics that are included only in few articles, for example, error tolerance, wire length, latency jitter, or packet loss. The basics fault-tolerance metrics are discussed in [32] but they were not considered in the papers listed here. Similar observations on utilized metrics were done in [87] as shown on the bottom row.

About 3 criteria are reported on average which is inadequate for good comparison. At least, a comparison based solely on a single source is very limited. It is necessary to justify the proposals and to show that the results are in accordance with the related work.

Measurement setup and the definition of metrics vary between research groups and this complicates comparison. Over half of the studies provide some sort of comparison. Comparison is practically always done between in-house developed approaches which are not well documented. Therefore, third-party reference points are valuable, as in [59][38][57]. Similarly, using a freely available NoC (simulator), such as Hermes [26], FPGA-NoC[4], Noxim[5], and Netmaker[6], as a common reference might prove beneficial since most NoCs are proprietary and unavailable for public evaluation. Furthermore, researchers must aim for wide spectrum of results instead of common practice of reporting just a few, non-related values. For example, measuring area with sensitivity analysis over multiple router configurations.

B. Test cases

Majority of publications use synthetic traffic, such as uniform random traffic, which is a valid approach on the first steps of design. However, we encourage systematically evaluating a large set of (representative) traffic parameters. Actual applications give the best accuracy but their traffic profiles are also suitable. For example, the profiles of [10] have been used by many researchers. Unfortunately, applications are rare used, in 40% of cases (mere 30% in [87]).

Test cases can be divided into computation kernels and full applications. Examples in the first category include image binarization, segmentation, and smoothing, IIR, FFT, DCT, dot product, vector sum, and matrix multiplication. Unfortunately

[4] http://www.da.isy.liu.se/research/soc/fpganoc/

[5] http://noxim.sourceforge.net/

[6] http://www-dyn.cl.cam.ac.uk/ rdm34/wiki/

these applications are not realistic for a billion transistor multi-processor SoC (MPSoC). In addition, those applications represent too fine grained parallelism which is out of the projected NoC scope.

The latter group of applications includes video encoders (H.263, MPEG-2, MPEG-4) and decoders (motion-JPEG, MPEG-2), smart camera, OFDM, radar signal analysis, and large database processing. These give a better view to real NoC performance, but are still insufficient alone. Application results from a physical prototype are given only in [40][52][54][71][95] and the others rely on simulation. The largest study includes 30 application traces [92] but only 1.3 cases are used on average. In [87], most studies used statistical traffic generators and 3.1 cases on average.

None has yet reported a NoC running several large applications simultaneously. This is important step to bring evaluations closer to expected usage scenario of NoC-based systems. At the same, it necessitates including such aspects as (real-time) operating system, micronetwork stack [8] and other middleware that have a profound implications on runtime, scheduling, and memory usage. Similarly, the impact of more detailed resource models has not been widely explored yet. Standardized test case set is currently being defined by OCP-IP NoC Benchmarking group [33].

V. NOC ROUTER IMPLEMENTATION

Table III lists examples of reported implementation results for NoCs. FPGA results are omitted for brevity. The results are given per router or wrapper, but their number in the full network depends on the topology. The results are sorted according to processing technology [7], topology, and then alphabetically.

Network interface (NI) handles the packetization, and possibly re-ordering or retransmissions. Details about the needed network interface are too often omitted. However, interface may double the required silicon area, as in [81] and [83]. The NoC area in [54], the sum of router and the associated network and GASL interfaces, is 19+10+12=41 kilogates per NoC terminal.

Table III includes also results for networks that are used as reference in the studies (marked with prefix *Ref.*). Re-implementing the ideas from in literature is important to obtain fair comparison via direct control over parameters. Table includes few reference implementations synthesized for this paper and targeting 400 MHz frequency. The networks are described in [85][86] [26]. Reference buses support hierarchical structures and results are for one wrapper. The reference octagon is implemented for this paper based on [46] but uses packet switching only.

Power consumption is very important issue with NoCs [73] but omitted from the table since it is not a static metric in the same sense as the area or maximum frequency. Leakage power is proportional to area but dynamic power is related to the sustainable traffic rate of a NoC and hence application-dependent[8]. Furthermore, the power depends on the used wire model and required wire length which both require knowledge about the layout. Further analysis of power consumption can be found in [10][55][56][80][102]. The lowest frequency that meets the application demands may be set at runtime to minimize dynamic power. Estimating and lowering frequency prior to synthesis avoids over-design, for example large signal drivers. In addition, low-leakage logic gates can be opted with less stringent delay constraints.

A. Router parameters

For a given topology, the area is mainly defined by the flit width and the buffer size. The flit width refers here to the data payload alone and the control signals are excluded. The most common width is 32 bits which is significantly smaller than 256 bits assumed in [25]. Few approaches use wide flits, for example 96 bits [83], 128 bits [62], and 256 bits [19]. Note that the flits are used for flow control (reserving buffers and links) whereas the phit (physical digit) denotes the number of parallel wires between the routers. However, they have the same width in most cases.

The reported buffer depths range from 1 flit to 30 flits but all sources do not document the buffer sizes. However, several sources report buffers taking 50-90% of the router area and major part of the power consumption, see for example [48][56][90]. Buffers are usually constructed from flip-flops but using SRAM is also viable, especially on FPGAs. Special FIFO design can offer clear area savings [83].

When specified, the buffer size is listed as a product of number of ports, virtual channels per port and depth of a buffer in flits, In other words $ports \cdot VC \cdot flits$, average being $5\ x\ 3\ x\ 6$ flits per router. The number of ports is defined by the topology, mesh being the most popular (5 ports: PE, North, East, south, and West) [9]. Circuit-switched proposals are often called "bufferless" but 1-flit buffers at each output are assumed for them.

Multiple virtual channels (VCs) may be associated with each physical port. Each VC has a separate buffer which reduces blocking and hence improves performance. They are needed by certain routing algorithms ensure correct operation without deadlock. Virtual channels are omitted in most current implementations. However, Mango supports 8 VCs [13] and circuit-switched SDM up to 32 VCs per port [58]. Virtual channels are necessary in adaptive routing schemes to avoid deadlocks. Higher VC count also increases performance up to 4-8 VCs/port [75].

---

[7] Note that some feature sizes are grouped together, for example 90 nm and 100 nm.

[8] Leakage is also application-dependent if power supply gating is applied.

[9] Ref. mesh on the bottom has two buffers for PE, one for tx and the other for rx.

TABLE III

IMPLEMENTATION RESULT EXAMPLES OF NOC ROUTERS.

#	Network	Author	Ref.	Top.	Flit width [bit]	Buffering			Tech. [nm]	Min. lat.		Router area		Freq. [MHz]	Detail level[1]
						ports	VC	flits		[cycle]	[ns]	[mm2]	[kgates]		
1	Orion/Luna	Soteriou	[92]	m	64	5	4	16	50	7	0.7	0.11	-	10000	pred.
2	TeraFLOPS	Vangal	[95]	m	32	4	2	16	65	5	1.2	0.34	53	4270	C
3	Xpipes	Bertozzi	[10]	c	32	4	1	?	100	7	-	0.1	-	-	pred.
4	Example NoC	Dally	[25]	t	256	5	8	4	100	3?	-	0.59	-	200-2000	pred.
5	GDB	Hosseinaba	[39]	db	22?	n/a			90	-	-	0.1-0.15	-	510-590	S
6	HIBI v.2, rom	Salminen	[85]	b,(hb)	32	2	1	2,8	130	4	9.2	0.03, 0,05	4, 8	435	R
7	Ref. octagon	Salminen	this	er	32	4	1	2,8	130	4	9.2	0.04, 0.09	6, 15	435	R
8	Mondinelli	Mondinelli	[63]	ftr	16	8	1	8+1	130	-	-	<0.29	<21	200	C
9	SPIN	Andriah.	[3]	ftr	32	8	1	4?	130	-	-	0.24	-	200	L
10	XGFT	Kariniemi	[47]	ftr	32	3,6	1	7	130	-	-	0.16, 0.25	16, 21	400	R
11	Aethereal router	Rijpkema	[83]	m	3 x 32	5	1	8	120	-	-	0.26	-	500	L
12	Aethereal NI	Radulescu	[81]	m(NI)	3 x 32	4-8	1	8/3	130	4	8.0	0.17-0.25	-	500	L
13	Anoc	Beigne	[7]	m	32	n/a			130	-	2-2.5	0.25	20	250	L
14	Faust	Lattard	[54]	m	32	5	2	?	130	-	0.6	~0.6	19	async.	C
15	Kavaldjiev	Kavaldjiev	[48]	m	16?	5	4	4?	130	-	-	0.18	-	500	S
16	Mango	Bjerreg.	[13]	m	32	5	8	1	130	-	-	0.19	-	795	R
17	Ref. 2-D mesh	Salminen	[86]	m	32	6	1	2,8	130	4	9.2	0.08, 0.14	13, 24	435	R
18	Ref. Hermes	de Mello	[26, this]	m	32	5	1	2,8	130	10	23.0	0.05, 0.11	9, 18	435	R
19	SDM NoC	Leroy	[58]	m	32	5	1-32	1	130	-	-	0.03-0.14	-	2270	L
20	Wolk., circ, pkt	Wolkotte	[100]	m	16	5,5	1,4	1,4	130	-	-	0.05, 0.18	-	1075,507	R
21	Ref. seg. bus	Chang	[18]	b	69	n/a			180	-	3.2	0.18	-	-	R
22	SocBus	Sathe	[89]	c	16	3	1	1	180	6	setup	0.04-0.08	-	1200	R
23	Asoc	Liang	[59]	m	32	4	1	2	180	-	-	0.07	-	400	L
24	Crossroad	Chang	[18]	m	69	5	1	1	180	-	3.5	0.06	-	-	R
25	DyAD mesh	Hu	[41]	m	32	5	1	8	160	-	-	-	26	333	C
26	Lochside	Mullins	[66]	m	64	5	4	4	180	1	4.0	0.5	-	250	C
27	Ref. pkt mesh	Chang	[18]	m	69	5	1	8	180	-	29.7	0.65	-	-	R
28	Proteo	Siguenz.	[90]	r	8	3	1	7	180	-	-	0.20	-	-	R
29	Slim-spider	Lee, K.	[56]	x, c	8	n/a			180	-	-	-	-	1600	C
30	Chi	Chi	[19]	m,t	256	5	4	4	250	-	-	2.89	-	150	C
31	Chain	Rostislav	[84]	b	-	n/a			350	-	-	-	4.75	async.	R
32	AET	Valtonen	[94]	m	16	n/a			350	3	26.3	0.8	15	114	L
33	Hermes	Moraes	[64]	m	32	5	1	5-30	350	10	-	-	10-49	(25)	R
34	Qnoc, async	Rostislav	[84]	m	-	n/a			350	-	4-9	0.09-0.47	2-12	async.	R
35	Qnoc, sync	Rostislav	[84]	m	-	n/a			350	1	3.7	0.2-0.96	4-18	270-303	R
# reported		35	35	35	32	27			35	14	17.0	31	15	30	35
% reported		100	100	100	91	77			100	40	49	89	43	86	100
		AVG (all)			50.1	4.8	3.1	6.4	169.9	5.2	9.2	-	17.9	-	R
		AVG (130 nm)			57.4	-			-	5.5	12.3	0.14	14.9	596	-
		AVG (180 nm)			41.9	-			-	3.5	10.1	0.22	26.0	328[2]	-

[1] pred.=predicted, C=chip, L=layout, R=RTL

[2] 756.6 MHz including [89][56]

B. Minimum latency

The minimum header latency induced by a router is given in column *Min. lat.*, in clock cycles and/or nanoseconds. The shown values are for the optimum case without contention. Once the header has been forwarded, the payload is transmitted (usually) at the rate of one phit per clock cycle.

Latency depends on the level of pipelining being about 5 clock cycles on average but details are not usually available. A basic 5-cycle router pipeline has the following stages: input buffering, virtual channel handling, routing, output arbitration, and switching. Detailed analysis of router pipelining is presented in [79].

A novel design of router allows single-cycle latency [66] and look-ahead mechanism that performs routing decision for the next router is presented in [19]. SoCBus [89] reports a 6-cycle latency for circuit-setup whereas the latency for data transfer is most likely one cycle. A combined latency of 10 cycles for the two network interfaces (sending and receiving ends) was reported in [10]. The latency of Aethereal NI is $4-10$ cycles whereas SW implementation of packetization takes nearly 50 cycles [81]. In [54], both NI and GALS interface introduce a double latency compared to a single router ($12\ ns$ vs. $6\ ns$). The reported HW latencies are low enough in our opinion, especially when the overheads regarding the middleware and network stack are included. Minimizing the latency and memory overhead of micronetwork stack that still enables simple reuse is an open research problem.

C. Area

The router area is given in mm^2 or kilogates. These values are meant only for coarse-level comparison since the router parameters or inclusion/exclusion of wiring area in results are not always stated. This is rather counterintuitive to note because the area is the most often reported metric (cf. Table I).

The reported gate count per router is around 15 kilogates which is reasonably low for large MPSoCs. Reported average area at 130 nm technology is $0.14mm^2$. Circuit-switching allows even smaller routers due to absence of buffers, see for example [58][59][100]. Sylvester and Keutzer [93] have estimated PEs with complexity of about $50-100$ kilogates will allow dependable design and verification also in the future. Hence, an average router ($15-18$ kilogates) would impose a notable overhead in chip area, roughly $15-36\%$ even without NIs. Area overheads in the range $9-45\%$ were reported in [75]. The overhead gets larger if separate wiring channels are reserved for the signals between the routers. Hence, NoC area optimization is encouraged when NoCs are used with modest-sized IPs.

Many sources assume quite large blocks to be connected via NoC, for example $1-9mm^2$, which would clearly diminish the relative area overhead. For example, TeraFLOPS has 3 mm^2 tiles and 53-kilogate routers account about 17% of the transistors. Large tiles consist of few hundred kilogates, which is large enough to necessitate a local network, for example bus (for area-optimization) or crossbar/point-to-point (performance optimization). This is a natural spot for the hierarchical NoCs.

D. Operating frequency

The next column shows the reported operating frequencies. Similar to the area, wiring has notable impact on the frequency in deep sub-micron technologies. The results are for the router only assuming that the link delays do not affect. Few NoCs are asynchronous [7][13][29][54][60] [84][101]. Most NoCs use synchronous components that may be asynchronous with respect to each other (globally asynchronous, locally synchronous, GALS). GALS paradigm is crucial in deep submicron technologies [8] due to difficulties in clock distribution and to enable power saving. It requires synchronization logic that affects latency in addition to area. These penalties, as well as synchronization failures, must be accounted in evaluation.

The average frequency is about 600 MHz for 130 nm technology which seems adequate for cost-optimized, medium-performance devices. However, few very fast NoCs [58][100] increase to the average value notably.

There are great differences between the fastest and the slowest results; over $11x$ with 130 nm and $6x$ with 180 nm technology. An order of magnitude difference is so large that a reader cannot assume anything about the NoC's performance unless it is explicitly given. This emphasizes the importance of contrasting results proposals to the existing ones and providing commentary if large deviations from the average performance level are observed. At the same time, the synthesis settings and silicon conditions must be documented.

The last column shows the detail level. Most implementation results are obtained from synthesis tool but good results are obtained also in real silicon implementations: $1.6\ GHz$ for Slim-Spider at $180\ nm$ [56], $1.35\ GHz$ for Nexus at $130\ nm$ [60], and astonishing $4.27-5.67\ GHz$ for TeraFLOPS at $65\ nm$.

VI. DISCUSSION

The most remarkable observation is the lack of standardized NoC benchmarks, anecdotal nature and limited number of results, and lack of comparison. However, more and more high-quality papers are published continuously as the research matures. Most studies cited here originate from academia but a growing interest is observed also in the industry.

The main concepts for benchmarking NoCs in a systematic and comparable way are currently addressed by OCP-IP workgroup [33]. The group is formalizing a set of relevant metrics, associated measurement methodologies, and a set of parameterized reference inputs for the NoC benchmarks. These ensure meaningful comparison between various sources and the overall view can be determined in incremental steps.

Table IV shows an example data sheet for contemporary average NoC and our improvement suggestions. To illustrate their importance, there are few properties that were not included in the previous tables, such as packet length, buffer type, and the parameters related to implementation results.

Self-evident point is to note the importance of proper documentation and comparison. Unfortunately, this is also a commonly overlooked issue. To put it simply, we need more applications, larger setups, more prototypes, larger range of evaluated parameters and configurations, better models for traffic cases and for networks (especially for NIs, network stack, and synchronization), and continued motivation to keep the NoCs evolving.

VII. CONCLUSIONS

This paper gives an overview of state-of-the-art network-on-chips. An average set of currently utilized properties is identified and presented. Datasheet of a stereotypical NoC is also given to alleviate the reporting.

TABLE IV

DATASHEET FOR A STEREOTYPICAL NOC AND SUGGESTED RESEARCH DIRECTIONS.

Property	Average value	Necessary improvement
Switching	packet-switched	QoS, predictability
Topology	2-D mesh	synthesis, IP mapping
Links	bidirectional	pipelining, power reduction
Flow control	wormhole	QoS
Data delivery	in-order	(cannot be compromised)
Virtual channels	no	needed, e.g. 2-8 VC/port
Routing	deterministic	fault-tolerance
QoS	yes	overhead minimization
Clocking	synchronous (or GALS)	synchronization impact
Network interface	excluded	considered
Fault-tolerance	neglected	considered
Testability	neglected	considered
Evaluation	simulation, synthesis	analysis, prototype
Prototype	no	needed
Used metrics	area, latency	appl. runtime, power
Comparison	vs. in-house ref.	vs. third-party/std-reference
Applications	not used	standardized, several simultaneous, modeling styles
Flit width	32 bits	more detailed documentation
Phit width	same as flit	
Packet length [flit]	hdr:1, payload:n , tail:1	
Buffering per router	5 x 3 x 6 flits	
Buffer type	based on flip-flops	
Router area	0.14 mm^2 @ 130 nm, 15-18 kilogates	
Router frequency	600 MHz @ 130 nm	
Min. hdr.lat/router	5 cycles	
Power	?	
Results from	RTL	
Silicon conditions	n/a	
Wire+repeaters	n/a	

Packet-switched NoCs with two-dimensional topology and deterministic routing are the most common. NoCs are mostly evaluated with simulation and synthesis but they should be complemented with analytical studies and (FGPA) prototypes. Furthermore, evaluations with large NoC configurations running (multiple) representative applications are desired. Silicon area, application runtime, network latency, and power consumption are the most important metrics. Good results and interesting proposals are plenty. However, large differences in implementation results, vague documentation, and lack of comparison were also observed.

Network-on-chip is an very active research field with many practical applications in industry. Based on the study, the following topics were identified as especially crucial for continued development and success of NoC paradigm: procedures and test cases for benchmarking, traffic characterization and modeling, design automation, latency and power minimization, fault-tolerance, QoS policies, prototyping, and network interface design.

REFERENCES

[1] B. Ahmad, A. Erdogan, and S. Khawam, "Architecture of a dynamically reconfigurable NoC for adaptive reconfigurable MPSoC," in *AHS*, Jun. 2006, pp. 405–411.
[2] T. Ahonen and J. Nurmi, "Integration of a NOC-based multimedia processing platform," in *FPL*, Aug. 2005, pp. 606–611.
[3] A. Andriahantenaina and A. Greiner, "Micro-network for SoC: Implementation of a 32-port SPIN network," in *DATE*, Mar. 2003, pp. 1128–1129.
[4] K. Anjo *et al.*, "Black-bus: a new data-transfer technique using local address on networks-on-chips," in *IPDPS*, Arp. 2004, pp. 10–17.
[5] "A comparison of network-on-chip and busses," Arteris, white paper, 2005.
[6] T. Bartic *et al.*, "Topology adaptive network-on-chip design and implementation," *IEE Proc. Comput. Digit. Tech.*, vol. 152, no. 4, pp. 467–472, Jul. 2005.
[7] E. Beigne *et al.*, "An asynchronous NOC architecture providing low latency service and its multi-level design framework," in *ASYNC*, Mar. 2005, pp. 54–63.
[8] L. Benini and G. de Micheli, "Networks on chips: A new SoC paradigm," *IEEE Computer*, vol. 35, no. 1, pp. 70–78, Jan. 2002.
[9] ——, *Networks on chips: technology and tools*. Morgan Kaufmann, 2006.
[10] D. Bertozzi *et al.*, "NoC synthesis flow for customized domain specific multiprocessor systems-on-chip," *IEEE Trans. Parallel and Distributed Systems*, vol. 16, no. 2, pp. 113–129, Feb. 2005.
[11] D. Bertozzi, L. Benini, and G. de Micheli, "Low power error resilient coding for on-chip data buses," in *DATE*, 2002.
[12] T. Bjerregaard and S. Mahadevan, "A survey of research and practices of network-on-chip," *ACM Computing Surveys*, vol. 38, no. 1, p. article No. 1, Jun. 2006.
[13] T. Bjerregaard and J. Sparsoe, "A router architecture for connection-oriented service guarantees in the MANGO clockless network-on-chip," in *DATE*, vol. 2, Mar. 2005, pp. 1226–1231.

[14] E. Bolotin *et al.*, “QNoC: QoS architecture and design process for network on chip,” *Journal of Systems Architecture*, vol. 50, no. 2–3, pp. 105–128, Feb. 2004.

[15] A. Bouhraoua and M. E. Elrabaa, “A high-throughput network-on-chip architecture for systems-on-chip interconnect,” in *Intl. Symposium on Soc*, Nov. 2006, pp. 127–130.

[16] L. Carloni and A. Sangiovanni-Vincentelli, “Coping with latency in soc design,” *IEEE Micro*, vol. 22, no. 5, pp. 24–35, Sep.-Oct. 2002.

[17] D. Castells-Rufas, J. Joven, and J. Carrabina, “A validation and performance evaluation tool for ProtoNoc,” in *Intl. Symposium on Soc*, Nov. 2006, pp. 159–162.

[18] K.-C. Chang, J.-S. Shen, and T.-F. Chen, “Evaluation and design trade-offs between circuit-switched and packet-switched NOCs for application-specific SOCs,” in *DAC*, Jul. 2006, pp. 143–148.

[19] H.-C. Chi and J.-H. Chen, “Design and implementation of a routing switch for on-chip interconnection networks,” in *AP-ASIC*, Aug. 2004, pp. 392–395.

[20] D. Ching, P. Schaumont, and I. Verbauwhede, “Integrated modeling and generation of a reconfigurable network-on-chip,” in *IPDPS*, Apr. 2004, pp. 139–145.

[21] I. Cidon and I. Keidar, “Zooming in on network-on-chip architectures,” Technion Department of Electrical Engineering, Tech. Rep. CCIT 565, Dec. 2005.

[22] J. Cong, “An interconnect-centric design flow for nanometer technologies,” *Proc. IEEE*, vol. 89, no. 4, pp. 505–508, Apr. 2001.

[23] *2D-Fabric 402c Parallel Processing Interface*, Crossbow Technologies Inc., Sugar Land, TX, USA, Jul. 2002.

[24] W. J. Dally and B. Towles, *Principles and practices of interconnection networks*. Morgan Kaufmann Publishers, 2004.

[25] W. Dally and B. Towles, “Route packets, not wires: on-chip interconnection networks,” in *DAC*, Jun. 2001, pp. 684–689.

[26] A. V. de Mello and L. H. Möller, “Hermes project web page,” [Online] http://toledo.inf.pucrs.br/ gaph/Projects/Hermes/Hermes.html, 2004.

[27] S. Evain, J. P. Diguet, and D. Houzet, “μspider: a CAD tool for efficient NoC design,” in *Norchip*, Nov. 2004, pp. 218–221.

[28] B. Feero and P. Pande, “Performance evaluation for three-dimensional networks-on-chip,” in *ISVLSI*, 2007, pp. 305–310.

[29] F. Felicijan and S. Furber, “An asynchronous on-chip network router with quality-of-service (QoS) support,” in *SOCC*, Sep. 2004, pp. 274–277.

[30] M. Forsell, “A scalable high-performance computing solution for networks on chips,” *IEEE Micro*, vol. 22, no. 5, pp. 46–55, Sep.-Oct. 2002.

[31] C. Grecu, A. Ivanov, R. Saleh, and P. Pande, “Testing network-on-chip communication fabrics,” *IEEE Trans. Computer-Aided Design of Integrated Circuits and Systems*, vol. 26, no. 10, pp. 2201–2014, Dec. 2007.

[32] C. Grecu, L. Anghel, P. P. Pande, A. Ivanov, and R. Saleh, “Essential fault-tolerance metrics for noc infrastructures,” in *IOLTS 07*, Jul. 2007, pp. 37–42.

[33] C. Grecu, P. Pande, A. Ivanov, R. Marculescu, E. Salminen, and A. Jantsch, “Towards open network-on-chip benchmarks,” in *NOCS*, May 2007, p. 205.

[34] P. Guerrier and A. Greiner, “A generic architecture for on-chip packet-switched interconnections,” in *DATE*, Mar. 2000, pp. 250–256.

[35] A. Hemani, A. Jantsch, S. Kumar, A. Postula, J. Öberg, M. Millberg, and D. Lindqvist, “Network on chip: An architecture for billion transistor era,” in *Norchip*, 2000.

[36] J. Henkel, W. Wolf, and S. Chakradhar, “On-chip networks: a scalable, communication-centric embedded system design paradigm,” in *VLSI*, Jan. 2004, pp. 845–851.

[37] T. Henriksson, D. Wiklund, and D. Liu, “VLSI implementation of a switch for on-chip networks,” in *DDECS*, 2003.

[38] C. Hilton and B. Nelson, “A flexible circuit switched NOC for FPGA based systems,” in *FPL*, Aug. 2005, pp. 24–26.

[39] M. Hosseinabady, M. Kakoee, J. Mathew, and D. Pradhan, “Reliable network-on-chip based on generalized de Bruijn graph,” in *HLVDT*, Nov. 2007, pp. 3–10.

[40] J. Hu, Y. Deng, and R. Marculescu, “System-level point-to-point communication synthesis using floorplanning information,” in *ASP-DAC/VLSI*, Jan. 2002, pp. 573–579.

[41] J. Hu and R. Marculescu, “DyAD - smart routing for networks-on-chip,” in *DAC*, Jun. 2004, pp. 260–263.

[42] A. Janarthanan and K. A. Tomko, “MoCSYS: A multi-clock hybrid two-layer router architecture and integrated topology synthesis framework for system-level design of fpga based on-chip networks.”

[43] A. Jantsch, R. Lauter, and A. Vitkowski, “Power analysis of link level and end-to-end data protection in networks on chip,” in *ISCAS*, vol. 2, Kobe, Japan, 2005, pp. 1770–1773.

[44] A. Jantsch and H. Tenhunen, Eds., *Networks on Chip*. Dordrecht, The Netherlands: Kluwer Academic Publishers, 2003.

[45] T. Kangas *et al.*, “Uml-based multi-processor soc design framework,” *ACM Trans. Embedded Computing Systems*, vol. 5, no. 2, pp. 281–320, May 2006.

[46] F. Karim, A. Nguyen, and S. Dey, “An interconnect architecture for networking systems on chips,” *IEEE Micro*, vol. 22, no. 5, pp. 36–45, Sep.-Oct. 2002.

[47] H. Kariniemi, “On-line reconfigurable extended generalized fat tree network-on-chip for multiprocessor system-on-chip circuits,” Ph.D. dissertation, Tampere University of Technology, Sep. 2006.

[48] N. Kavaldjiev *et al.*, “A virtual channel network-on-chip for GT and BE traffic,” in *ISVLSI*, vol. 00, Mar. 2006.

[49] D. Kim, M. Kim, and G. E. Sobelman, “Design of a high-performance scalable CDMA router for on-chip switched networks,” in *ISSOC*, 2005, pp. 32–35.

[50] D. Kim, K. Kim, J.-Y. Kim, S. Lee, and H.-J. Yoo, “An 81.6 GOPS object recognition processor based on noc and visual image processing memory,” in *CICC*.

[51] J. Kim *et al.*, “A low latency router supporting adaptivity for on-chip interconnects,” in *DAC*, Jun. 2005, pp. 559–564.

[52] A. Kulmala, E. Salminen, and T. D. Hämäläinen, “Prototyping and evaluating large system-on-chips on Multi-FPGA platform,” in *SAMOS VII*, Jul. 2007, pp. 179–189.

[53] S. Kumar *et al.*, “A network on chip architecture and design methodology,” in *VLSI*, Apr. 2002, pp. 105–112.

[54] D. Lattard *et al.*, “A reconfigurable baseband platform based on an asynchronous network-on-chip,” *IEEE J. Solid-State Circuits*, vol. 43, no. 1, pp. 223–235, Jan. 2008.

[55] H. G. Lee *et al.*, “Design space exploration and prototyping for on-chip multimedia applications,” in *DAC*, Jul. 2006, pp. 137–142.

[56] K. Lee, S.-J. Lee, and H.-J. Yoo, “Low-power network-on-chip for high-performance SoC design,” *IEEE Trans. VLSI Syst.*, vol. 14, no. 2, pp. 148–160, Feb. 2006.

[57] S. Lee, C. Lee, and H.-J. Lee, “A new multi-channel on-chip-bus architecture for system-on-chips,” in *SOCC*, Sep. 2004, pp. 305–308.

[58] A. Leroy *et al.*, “Spatial division multiplexing: a novel approach for guaranteed throughput on nocs,” in *CODES+ISSS*, Sep. 2005, pp. 81–86.

[59] J. Liang *et al.*, “An architecture and compiler for scalable on-chip communication,” *IEEE Trans. VLSI Syst.*, vol. 12, no. 7, pp. 711–726, Jul. 2004.

[60] A. Lines, “Asynchronous interconnect for synchronous SoC design,” *IEEE Micro*, vol. 24, no. 1, pp. 32–41, Jan.-Feb. 2004.

[61] A. Mello, L. Tedesco, N. Calazans, and F. Moraes, “Evaluation of current QoS mechanisms in network on chip,” in *Intl. Symposium on Soc*, Nov. 2006, pp. 115–118.

[62] M. Millberg, R. T. E. Nilsson, and A. Jantsch, “Guaranteed bandwidth using looped containers in temporally disjoint networks within the Nostrum network on chip,” in *DATE*, Feb. 2004, pp. 890–895.

[63] F. Mondinelli, M. Borgatti, and Z. Vajna, “A 0.13 um 1Gb/s/channel store-and-forward network on-chip,” in *SOCC*, Sep. 2004, pp. 141–142.

[64] F. Moraes *et al.*, “Hermes: an infrastructure for low area overhead packet-switching networks on chip,” *Integration, the VLSI Journal*, vol. 38, no. 1, pp. 69–93, Oct. 2004.

[65] A. Morgenshtein, I. Cidon, A. Kolodny, and R. Ginosar, "Comparative analysis of serial vs parallel links in noc," in *Intl. Symposium on Soc*, 2004, pp. 185–188.
[66] R. Mullins, A. West, and S. Moore, "The design and implementation of a low-latency on-chip network," in *ASP-DAC*, Jan. 2006.
[67] S. Murali *et al.*, "Analysis of error recovery schems for networks on chips," *IEEE Design and test*, vol. 22, no. 5, pp. 434–442, 2005.
[68] E. Nilsson and J. Öberg, "Panacea - a case study on the panacea noc - a nostrum network on chip prototype," Royal Institute of Technology, Tech. Rep. 229, Apr. 2006.
[69] J. Nurmi, H. Tenhunen, J. Isoaho, and A. Jantsch, Eds., *Interconnect-Centric Design for Advanced SoC and NoC*. Dordrecht, The Netherlands: Kluwer Academic Publishers, 2004.
[70] S. Ogg, E. Valli, C. D'Alessandro, A. Yakovlev, B. Al-Hashimi, and L. Benini, "Reducing interconnect cost in noc through serialized asynchronous links," in *NOCS*, 2007, pp. 219–219.
[71] U. Y. Ogras *et al.*, "Communication architecture optimization: making the shortest path shorter in regular networks-on-chip," in *DATE*, Mar. 2006.
[72] U. Y. Ogras, J. Hu, and R. Marculescu, "Key research problems in NoC design: a holistic perspective," in *CODES*, Sep. 2005, pp. 69–75.
[73] J. Owens *et al.*, "Research challenges for on-chip interconnection networks," *IEEE Micro*, vol. 27, no. 5, pp. 96–108, Sep-Oct. 2007.
[74] P. P. Pande *et al.*, "Design, synthesis, and test of networks on chips," *IEEE Des. Test Comput*, vol. 22, no. 5, pp. 404–413, Aug. 2005.
[75] ——, "Performance evaluation and design trade-offs for network-on-chip interconnect architectures," *IEEE Trans. Computers*, vol. 54, no. 8, pp. 1025–1040, Aug. 2005.
[76] ——, "Switch-based interconnect architecture for future systems on chip," in *SPIE, VLSI Circuits and Systems*, vol. 5117, May 2003, pp. 228–237.
[77] P. G. Paulin *et al.*, "Parallel programming models for a multi-processor soc platform applied to high-speed traffic mangement."
[78] V. F. Pavlidis and E. G. Friedman, "3-d topologies for networks-on-chip," *IEEE Trans. VLSI Syst.*, 2007.
[79] L.-S. Peh and W. J. Dally, "A delay model for router micro-architectures," *IEEE Micro*, pp. 26–34, Jan/Feb. 2001.
[80] S. Penolazzi and A. Jantsch, "A high level power model for the Nostrum NoC," in *DSD*, Aug. 2006, pp. 673–676.
[81] A. Radulescu *et al.*, "An efficient on-chip NI offering guaranteed services, shared-memory abstraction, and flexible network configuration," *IEEE Trans. Computer-Aided Design of Integrated Circuits and Systems*, vol. 24, no. 1, pp. 4–17, Jan. 2005.
[82] T. Richardson *et al.*, "A hybrid SoC interconnect with dynamic TDMA-based transaction-less buses and on-chip networks," in *VLSI design*, Jan. 2006.
[83] E. Rijpkema *et al.*, "Trade offs in the design of a router with both guaranteed and best-effort services for network on chip (extended version)," *IEE Proc. Computers and Digital Techniques*, vol. 150, no. 5, pp. 294–302, Sep. 2003.
[84] D. Rostislav *et al.*, "An asynchronous router for multiple service levels networks on chip," in *ASYNC*, Mar. 2005, pp. 224–229.
[85] E. Salminen *et al.*, "HIBI communication network for system-on-chip," *Journal of VLSI Signal Processing-Systems for Signal, Image, and Video Technology*, vol. 43, no. 2, pp. 185–205, May 2006.
[86] ——, "Benchmarking mesh and hierarchical bus networks in system-on-chip context," *Journal of System Architectures (in press)*, vol. 53, no. 8, pp. 477–488, Aug. 2007.
[87] E. Salminen, A. Kulmala, and T. Hämäläinen, "On network-on-chip comparison," in *Euromicro DSD*, Aug. 2007, pp. 503–510.
[88] H. Samuelsson and S. Kumar, "Ring road noc architecture," in *Norchip*, 2004, pp. 16–19.
[89] S. Sathe, D. Wiklund, and D. Liu, "Design of a switching node (router) for on-chip networks," in *ASIC*, Oct. 2003, pp. 75–78.
[90] D. Sigüenza-Tortosa, T. Ahonen, and J. Nurmi, "Issues in the development of a practical NoC: the Proteo concept," *Integration, the VLSI Journal*, vol. 38, no. 1, pp. 95–105, Oct. 2004.
[91] J.-P. Soininen, A. Jantsch, M. Forsell, A. Pelkonen, J. Kreku, and S. Kumar, "Extending platform-based design to network on chip systems," in *VLSI Design*, Jan. 2003, pp. 401–408.
[92] V. Soteriou *et al.*, "Polaris: A system-level roadmapping toolchain for on-chip interconnection networks," *IEEE Trans. VLSI Syst.*, vol. 15, no. 8, pp. 855–868, Aug. 2007.
[93] D. Sylvester and K. Keutzer, "Impact of small process geometries on microarchitectures in systems on a chip," *Proc. IEEE*, vol. 89, no. 4, pp. 467–489, Apr. 2001.
[94] T. Valtonen *et al.*, "An autonomous error-tolerant cell for scalable network-on-chip architectures," in *Norchip*, Nov. 2001, pp. 198–203.
[95] S. Vangal *et al.*, "An 80-tile Sub-100-W TeraFLOPS processor in 65-nm CMOS," *IEEE J. Solid-State Circuits*, vol. 43, no. 1, pp. 29–41, Jan. 2007.
[96] N. Wang, A. Sanusi, P. Zhao, S. Mohamed, and M. A. Bayoumi, "PMCNOC: A pipelining multi-channel central caching network-on-chip communication architecture design," in *SiPS*, Oct. 2007, pp. 487–492.
[97] P. Wielage and K. Goossens, "Networks on silicon: blessing or nightmare?" in *DSD*, Sep. 2002, pp. 196–200.
[98] D. Wiklund, "Development and performance evaluation of networks on chip," Ph.D. dissertation, Linköping university, Linköping, Sweden, Apr. 2005.
[99] D. Wiklund and D. Liu, "Switched interconnect for system-on-a-chip designs," in *IP2000*, Oct. 2000, pp. 198–192.
[100] P. Wolkotte *et al.*, "An energy-efficient reconfigurable circuit-switched network-on-chip," in *IPDPS*, Apr. 2005, p. 155a.
[101] Xin Wang, T. Ahonen, and J. Nurmi, "Applying cdma technique to network-on-chip," *IEEE Trans. VLSI Syst.*, vol. 15, no. 10, pp. 1091–1100, Oct.
[102] J. Xu *et al.*, "Methodology for design, modeling, and analysis of networks-on-chip," in *ISCAS*, May 2005, pp. 1778–1781.
[103] C. Zeferino, M. Kreutz, and A. Susin, "Rasoc: a router soft-core for networks-on-chip," in *DATE*, vol. 3, Feb. 2004, pp. 198–203.