Evaluation of Current QoS Mechanisms in Networks on Chip

Aline Mello, Leonel Tedesco, Ney Calazans, Fernando Moraes
Faculdade de Informática
Pontifícia Universidade Católica do Rio Grande do Sul, PUCRS
Porto Alegre, Brazil
{alinev, ltedesco, calazans, moraes}@inf.pucrs.br

Abstract—**Several propositions of NoC architectures claim to provide quality of service (QoS) guarantees, which is essential for e.g. real time and multimedia applications. The most widespread approach to attain some degree of QoS guarantee relies on a two-step process. The first step is to characterize application performance through traffic modeling and simulation. The second step consists in tuning a given network template to achieve some degree of QoS guarantee. These QoS targeted NoC templates usually provide specialized structures to allow either the creation of connections (circuit switching) or the assignment of priorities to connectionless flows. It is possible to identify three drawbacks in this approach. First, it is not possible to guarantee QoS for new applications expected to run on the system, if those are defined after the network design phase. Second, even with end-to-end delay guarantees, connectionless approaches introduce jitter. Third, to model traffic precisely for a complex application is a very hard task. The objective of this paper is to evaluate the area-performance trade-off and the limitations of circuit switching and priority scheduling to meet QoS. Preliminary results show the need of more research in this field, by considering the aggregation of more explicit techniques to control QoS.**

I. Introduction

Most NoC implementations only provide support to best effort (BE) services [1], including those commercialized by Arteris [2]. BE services guarantee the transmission of all packets from a given source to a given target, without temporal bounds. The term QoS refers to the capacity of a network to control traffic constraints in order to meet design requirements of the application or of specific modules. Therefore, BE service is inadequate to satisfy QoS requirements for applications/modules with tight performance requirements.

External or internal mechanisms to the network may be used to meet QoS restrictions. When external mechanisms are employed, the network is designed according to the application using it. These mechanisms require accurate traffic modeling and network simulation to obtain the required bandwidth and latency figures for the target application. The results obtained through simulation allow to correctly dimension the network. The network synthesis occurs after the simulation step. In this case, it is still possible that no guarantee to meet QoS is given for new applications. Modern SoCs, such as 3G phones, support different applications profiles. Designing the network to support all traffic scenarios is unfeasible in terms of power and area. Thus, internal mechanisms, as admission control and/or traffic shaping, need to be used to enable the network to meet QoS requirements for a wide range of applications. These mechanisms are frequently used in IP and ATM networks. The main advantage of using internal mechanisms is to support new applications after network design, at the cost of extra area and dissipated power.

The objective of this paper is to evaluate the area-performance trade-off and the limitations of circuit switching and priority scheduling to meet QoS.

This paper is organized as follows. Section II presents related work in NoCs that offer guarantees of QoS. Section III details three NoC designs: best effort NoC, NoC employing a priority scheduling, and NoC employing circuit and packet switching. Section IV evaluates the latency, jitter and throughput for the NoCs designs. Section V presents conclusions and directions for future work.

II. Related Work

Current NoC designs employ one of three methods to provide QoS: (*i*) dimensioning the network to provide enough bandwidth to satisfy all IP requirements in the system; (*ii*) providing support to circuit switching for all or for selected IPs; (*iii*) making available priority scheduling for packet transmission.

The first method to provide QoS is advocated for example by the Xpipes NoC [3]. A designer sizes Xpipes according to application requirements, adjusting each channel bandwidth to fulfill the requirements. However, applying this method alone does not guarantee avoidance of local congestions (hot spots), even if bandwidth is largely increased.

The second method, circuit switching, provides a connection-oriented distinction between flows. This method is used in Æthereal [4], aSOC [5], Octagon [6] and SoCBUS [7]. The network creates connections for each or to selected flows. This scheme has the advantage to guarantee tight temporal bounds for individual flows. However, this method has three main disadvantages: (*i*) poor scalability, since router area grows proportional to the number of supported connections; (*ii*) inefficient bandwidth usage, because resource allocation is based in a worst case scenario; (*iii*) the setting up a circuit at runtime may long time (as will be shown later in Fig. 2) and in most cases unpredictable latency.

QNoC [8] and RSoC [9] are examples of NoCs adopting the third method, packet switching with priorities. This connectionless technique group traffic flows into different classes, with different services levels for each class. This scheme offers better adaptation to varying network traffic and a potentially better utilization of network resources. However, end-to-end latency and throughput cannot be guaranteed, except to the higher priority flows. When flows share resources, even higher priority flows can have an unpredictable behavior. Consequently, this method often provides a weaker QoS support than circuit switching.

Both circuit switching and priority methods do not guarantee QoS for multiple flows. When using the circuit switching method, the network may reject a number of flows, due to limited amount of simultaneously supported connections, even if network bandwidth is available. When multiple flows with the same priority compete for the same resources, priority-based networks have behavior similar to BE service networks, with no QoS guarantee. As mentioned before, networks using either methods (for example, Xpipes, Æthereal and QNoC) employ techniques external to the network to guarantee QoS. A network supporting these techniques guarantees QoS to the traffic scenario used before the network design. The drawbacks of external methods are: (*i*) the complexity of traffic modeling and system simulation is very high and (*ii*) the network does not guarantee QoS for new applications.

Finally, the main performance figures used in the above mentioned NoCs are end-to-end latency and throughput. Nonetheless, when QoS is considered, the variation in end-to-end latency (jitter) may be a mandatory consideration, as in video and audio applications. In connectionless networks, buffers introduce jitter. Therefore, networks using only priorities cannot guarantee controlled jitter. The solution to guarantee QoS is to use network internal methods, like admission control, congestion control and traffic shaping.

III. NoC Designs

A. Reference NoC

The Reference NoC is based on Hermes [10], a parameterizable infrastructure used to implement low area overhead wormhole packet switching NoCs with 2D mesh topology. The first and the second flits of a packet are header information, respectively containing the target address, and the payload size (up to $2^{(\text{flit size, in bits})}$) in flits.

The router has a centralized switching control logic and five bi-directional ports. The Local port establishes a communication between the router and its local core. The other ports of the router are connected to neighbor routers. A physical channel, including the local port, may support multiplexed VCs [11]. Each input port has a depth *d* buffer, for temporary flit storage. When *n* VCs are used, a buffer with *d/n* depth is associated to each VC.

Multiple packets may arrive simultaneously in a given router. A centralized round-robin arbitration grants access to incoming packets. The priority of a VC is a function of the last VC having a routing request granted. If the incoming packet request is granted by the arbiter, the XY routing algorithm is executed to connect the input port to the correct output port. When the algorithm returns a busy output port, the header flit and all subsequent flits of this packet are blocked. After routing execution, the output port allocates the bandwidth among the *n* VCs. Each VC, having flits to transmit occupies at least *1/n* of the physical channel bandwidth. If only one VC satisfies this condition, it occupies the whole physical channel bandwidth.

B. Priority NoC

The objective of this NoC is to add the ability to provide differentiated services to the flows, using a resource allocation mechanism based on priorities (similar to QNoC [8]). In the Priority NoC, each VC is associated to a fixed priority and served according to it. In this way, this NoC allows the network to differentiate *n* flows, where *n* is the number of VCs per physical channel.

To differentiate flows, the packet header is extended by a new field, named priority. This field determines which VC is used for packet transmission. The user may assign a value between zero and (*n*-1) to the priority field, zero being the lowest priority and (*n*-1) the highest. Only the source router verifies the priority field. The remaining routers transmit packets using the same VC allocated by the source router.

The assignment of priorities to virtual channels (VC) requires modification of the arbitration and scheduling router policies without modifying the reference NoC router interface. In priority-based arbitration, when multiple packets arrive simultaneously at the router input ports, the packet with higher priority is served first, even if other packets are waiting to be served. In priority-based scheduling, packets with higher priority are also served first. Then, data transmission in lower priority VCs depends on the load of the higher priority VCs, which can vary dynamically.

C. Circuit Switching NoC

The Circuit switching NoC adds the ability to differentiate services through connection establishment. The network offers a guaranteed throughput (GT) service to flows with QoS requirements. To flows without QoS requirements, the network offers a best effort (BE) service. This approach, GT plus BE, is similar to the one implemented in the Æthereal NoC.

This design employs two VCs, L1 and L2. VC L1 carries circuit switching data, while VC L2 is used to transmit packet switching data. GT flows have priority higher than BE flows, with end-to-end latency guarantee. When a given GT flow leaves the physical channel idle, BE flows may use this channel, without incurring in any significant penalty to GT data. The interface between routers has an additional signal to indicate connection establishment and release.

A GT flow requires connection establishment before starting data transmission. A connection between a source and a target node requires the reservation of VC L1 along the path between their respective routers. The circuitry to implement circuit switching is simpler than packet switching, since a single flit register can be used, and the control flow is simplified, requiring neither handshake nor credit control. The connections are established or released using BE control packets. These packets are differentiated from BE data packets by the most significant bit of the first header flit.

IV. EXPERIMENTAL RESULTS

The influence of traffic in system performance is greater than that of network structural parameters [12]. Thus, it is important to dispose of traffic generators to model the behavior of real traffic. This Section compares the performance of the described NoCs. Traffic injection and results capture is modeled with SystemC, while the NoC is modeled through RTL VHDL. The parameters for all NoCs are: 8x8 mesh topology; XY routing; 16-bit flits; 2 VCs; 8-flit buffers associated to each input VC.

A. Experimental Setup

Tab. I presents the flows used in the experiments. Flow A is characterized as a CBR service (Constant Bit Rate) and flow B is characterized as a VBR service (Variable Bit Rate). This VBR flow is modeled using Pareto distribution [13]. Flows A and B have QoS requirements, as latency and jitter. Nodes generating flows A and B transmit 200 packets. The results do not take into account the first 50 packets and the last 50 packets. They are discarded from results, since the traffic at the beginning and the end of the simulation does not correspond to regular load operation. Flow C is a BE flow, also modeled using a Pareto distribution. This flow is used to disturb flows with QoS requirements (A and B), being considered as noise traffic. For this reason, results for the C flow are not discussed.

TABLE I. FLOWS CHARACTERIZATION.

Type	Service	QoS	Load	Number of Packets	Packet Size	Target
A	CBR	Yes	Uniform (20%)	200	50	Single
B	VBR	Yes	Pareto (40% on)	200	50	Single
C	BE	No	Pareto (20% on)	Random	20	Random

Fig. 1 presents the spatial distribution of source and target nodes. In this scenario, two QoS flows originated at different nodes share part of the paths to targets. The remaining network nodes transmit C flows, disturbing the QoS flows.

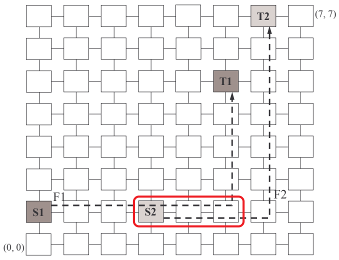

Figure 1. Spatial distribution of source and target nodes for flows with QoS requirements. Dotted lines indicate the path of each flow. Rounded rectangles highlight the area where flows compete for network resources.

Tab. II summarizes the experiments used to evaluate Priority and Circuit Switching NoCs. The priority column has no meaning in the reference NoC, which is a pure BE design. In the Circuit Switching NoC, flows with priority 1 are GT flows and flows with priority 0 are BE flows.

TABLE II. EXPERIMENTAL SCENARIOS.

Experiment	F1		F2		Noise flows	
	Type	Priority	Type	Priority	Type	Priority
I	A	1	A	0	C	0
II	A	1	A	1	C	0
III	B	1	B	1	C	0

NA = Not Applicable

This paper shows only the results for short packets (50 flits), but similar behavior is observable in experiments with long packets (1000 flits).

B. Priority Mechanism Analysis

Tab. III presents the minimum, average and maximum latencies, jitter and throughput for Experiment I. In the reference NOC, there is no differentiation between flows. Thus, the average latency, jitter and throughput of packets depend on the traffic conditions during transmission. In this way, the Reference NoC does not offer guarantees to any flow. In the Priority NoC, the highest priority flow F1 has average latency near to the minimum latency, jitter is closer to zero and average throughput is almost equal to the insertion rate (20%). This occurs because F1 has higher priority and exclusive usage of the L2 VC. This experiment shows that, even under disturbing traffic conditions (flows F2 and noise), a priority mechanism is efficient for guaranteeing QoS, as long as flows with a same priority do not compete. Throughput values higher than the 20% injection rate are normal and do not imply packet delivery faster than this rate. Rather, this occurs due to the way throughput is measured (at the receiver side) and is induced by network congestion, followed by subsequent packet burst transmission.

TABLE III. FLOWS F1 AND F2, EXPERIMENT I.

Performance Figures		Reference NoC		Priority NoC	
		F1	F2	F1	F2
Latency	Minimum (ck)	114,00	99,00	**99,00**	117,00
	Average (ck)	132,19	113,10	**101,88**	147,38
	Maximum (ck)	192,00	158,00	**111,00**	336,00
Jitter (ck)		16,09	12,80	**2,79**	43,21
Average throughput (%)		26,88	31,55	**19,21**	36,84

Tab. IV presents results for Experiment II. Flows F1 and F2 have the same priority, thus competing for VC L2. It is possible to observe that F1 and F2 have average latency near to minimum, jitter close to zero and average throughput in accordance with the insertion rates (20%). However, the F1 flow has higher latency. This occurs because F1 and F2 are CBR flows. Thus, they insert packets in the network at fixed intervals. As the F2 source node is closer to the region disputed by the flows, it is always served first.

However, when F1 and F2 are VBR flows (Experiment III) the results are quite different, as displayed in Tab. V. In this experiment, packets are inserted in the network at variable intervals using a 40% load for the ON period. The ON-OFF traffic model randomizes the packet injection instants. The main consequence is the increase in the jitter of both flows. Depending on the parameters that specify QoS for the flows, the usage of priority mechanism should be limited to specific situations, where competition among equal priority flows is avoidable or kept to a minimum.

TABLE IV. FLOWS F1 AND F2, EXPERIMENT II, CBR TRAFFIC.

Performance Figures		Reference NoC		Priority NoC	
		F1	F2	F1	F2
Latency	Minimum (ck)	114,00	99,00	**141,00**	**99,00**
	Average (ck)	132,19	113,10	144,23	101,78
	Maximal (ck)	192,00	158,00	154,00	113,00
Jitter (ck)		16,09	12,80	**2,66**	**3,04**
Average throughput (%)		26,88	31,55	**19,21**	**19,21**

TABLE V. FLOWS F1 AND F2, EXPERIMENT III, VBR TRAFFIC.

Performance Figures		Reference NoC		Priority NoC	
		F1	F2	F1	F2
Latency	Minimum (ck)	99,00	99,00	99,00	99,00
	Average (ck)	124,92	120,44	105,53	107,91
	Maximal (ck)	196,00	190,00	148,00	157,00
Jitter (ck)		21,99	22,13	**11,73**	**14,77**
Average throughput (%)		38,15	38,48	36,03	35,26

C. *Circuit Switching Mechanism Analysis*

The circuit switching mechanism guarantees QoS when flows do not compete for the same resources. Fig. 2 illustrates the amount of time required for connection establishment, data transmission and connection release, using flows of Experiment II, with F1 and F2 being GT flows, competing for the same VC. The amount of time to establish and release a connection, small in this experiment, can vary according to the network traffic, since these actions are controlled by BE packets. As illustrated in Fig. 2, F2 establish its connection after F1 release its connection. In SoCs, where video, audio and control signals flows are frequent and have QoS requirements, it is not possible to guarantee that will not exist competition between such QoS flows. Therefore, priority and circuit switching only provide QoS for specific situations.

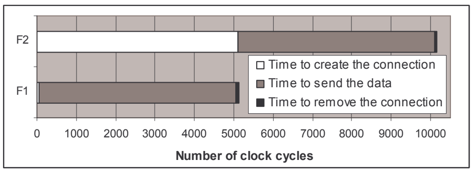

Figure 2. Time to connection establishment, data transmission and connection release for F1 and F2.

D. *Area Results*

Tab. VI presents the router area, obtained with Synplify synthesis tool.

TABLE VI. ROUTER AREA RESULTS FOR 2V1000 FPGA.

Resource	Mapping to Xilinx XC2V1000 FPGA device						
	Used			Available	Used /Available		
	Ref	Priority	CS		Ref	Priority	CS
Slices	1071	1158	967	5.120	20,92%	22,62%	18,89%
LUTs	1984	2150	1622	10.240	19,38%	21,00%	15,84%
Flip Flops	513	479	467	11.212	4,56%	4,27%	4,17%

Ref = Reference NoC; CS = Circuit Switching NoC

Router area is similar in terms of functions generators (LUTS), around 2000, for the Reference NoC and the Priority NoC. The Circuit Switching NoC has the smaller area, 1622 LUTS, since the input buffers of the circuit switching VC are replace by simple registers. The results point to the fact that priority and circuit switching do not significantly increase area, compared do the Reference NoC. Such mechanisms may be used to force the NoC to respect QoS requirements, not influencing final area.

V. CONCLUSIONS AND FUTURE WORK

This work evaluated two methods currently proposed to provide QoS for NoCs: (*i*) priority based resource allocation, and (*ii*) connection establishment support. Both methods present limitations, specially when flows with QoS requirements compete for network resources. As shown in Experiment I, if only one high priority flow has QoS requirements, priority mechanisms are effective. When flows with a same priority compete for resources, the priority mechanism does not provide rigid guarantees to none of the flows. It is possible to observe a minimal jitter when applying CBR traffic, but the consequence is latency penalization for one flow (as shown in Exp. II). When applying VBR flow, latencies near to the minimum appear, but with increased jitter (Exp. III). An alternative, increasing the number of priorities, implies increasing the amount of VCs, which can be prohibitive in terms of silicon area. In the method based in connection establishment, all QoS requirements are guaranteed after connection establishment. However, if some other flow not using connection establishment has deadlines to send data as QoS requirement then this method will be not able to guarantee this requirement. As a general conclusion, the state of the art in NoCs still does not present efficient solutions to provide QoS to applications when the network traffic is not known in advance.

REFERENCES

[1] Rijpkema, E.; et al. "*Trade Offs in the Design of a Router with Both Guaranteed and Best-Effort Services for Networks on Chip*". In: DATE, 2003, pp. 350-355.

[2] Arteris. "*Arteris Network on Chip Company*". Update, 2005. Available at http://www.arteris.net.

[3] Bertozzi, D.; Benini, L. "*Xpipes: A Network-on-chip Architecture for Gigascale Systems-on-Chip*". IEEE Circuits and Systems Magazine, 4(2), 2004, pp. 18-31.

[4] Goossens, K.; et al. "*Æthereal Network on Chip: Concepts, Architectures, and Implementations*". IEEE Design and Test of Computer, v.22(5), Sep.-Oct. 2005, pp. 414-421.

[5] Liang, J.; et al. "*aSOC: A Scalable, Single-Chip communications Architecture*". In: IEEE International Conference on Parallel Architectures and Compilation Techniques, pp. 37-46.

[6] Karim, F.; et al. "*An interconnect architecture for network systems on chips*". IEEE Micro, v.22(5), Sep.-Oct. 2002, pp. 36-45.

[7] Wiklund, D.; Liu D. "*SoCBUS: Switched Network on Chip for Hard Real Time Systems*". In: IPDPS, 2003.

[8] Bolotin E. et al. "*QNoC: QoS Architecture and Design Process for Network on Chip*". JSA, v.50(2-3), Feb. 2004, pp 105-128.

[9] Véstias, M.; Neto, H. "*A Reconfigurable SoC Platform Based on a Network on Chip Architecture with QoS*". In: XX DCIS, 2005.

[10] Moraes, F.; et al. "*Hermes: an Infrastructure for Low Area Overhead Packet-switching Networks on Chip*". Integration the VLSI Journal, v.38(1), Oct. 2004, pp. 69-93.

[11] Mello, A.; et al. "*Virtual Channels in Networks on Chip: Inplementantion and Evaluation on Hermes NoC*". In: SBCCI, 2005, pp. 178-183.

[12] Duato, J.; Yalamanchili, S.; Ni, L. "*Interconnection Networks*". Elsevier Science, 2002, 600 p.

[13] Pande, P.; et al. "*Performance Evaluation and design Trade-Offs for Network-on-Chip Interconnect Architectures*". IEEE Transactions on Computers, 54 (8), Aug. 2005, pp. 1025-104.