Router Architecture for High-Performance NoCs

Everton Carara carara@inf.pucrs.br
Ney Calazans calazans@inf.pucrs.br
Fernando Moraes moraes@inf.pucrs.br
Pontifícia Universidade Católica do Rio Grande do Sul (FACIN-PUCRS)
Av. Ipiranga, 6681 – 90619-900 – Porto Alegre – BRASIL

ABSTRACT

A considerable number of NoC designs are available, focusing on different aspects of this type of communication infrastructure. Example of relevant aspects considered during NoC design are quality-of-service achievement, the choice of synchronization method to employ between routers, power consumption reduction and application modules mapping. However, some design choices are common to many if not most NoC proposals: wormhole packet switching and the use of virtual channels. This work discusses trade-offs on using circuit and packet switching, arguing in favor of the former with fixed packet size. Next, it proposes and justifies the replacement of virtual channels by replicated channels, based on the abundance of wires expected in current and future deep sub-micron technologies. Finally, the work proposes the use of a session layer coupled to circuit switching. Results point out to reduced latency and router area, leading to a router architecture adapted for high-performance NoCs.

Categories and Subject Descriptors

B.7.1 [Integrated Circuits]: Types and Design Styles – advanced technologies, algorithms implemented in hardware, VLSI (very large scale integration).

General Terms

Design, Experimentation, Measurement, Performance, Theory, Verification.

Keywords

Networks on Chip, Switching Modes, Virtual Channels, Session Layer.

1. INTRODUCTION

The main performance figures used to evaluate interconnection infrastructures such as busses and NoCs are latency, throughput and jitter. NoC performance is a function of design choices concerning switching mode, physical channel allocation policy, buffering strategy, routing algorithm and arbitration policy.

SBCCI'07, September 3–6, 2007, Rio de Janeiro, Brazil.

Buffering strategies, routing algorithms and arbitration policies are router structural parameters. On the other hand, switching mode and physical channel allocation reflect the way data transmission occurs between routers.

Most NoC proposals employ layered stacks similar to OSI reference model [1] . The three lower layers (*physical*, *link* and *network*) are often implemented in hardware. The *physical* layer is responsible for providing the electrical media definitions to connect routers among them or routers to PEs. The *link* layer is responsible for the reliable transport of packets from one router/IP to another router/IP across the links while applying flow control, such as credit based or handshake. The *network* layer is responsible for path determination and logical addressing (routing algorithms). The *transport*/*session* layers respond for end-to-end connection, assembling and disassembling messages, and end-to-end error handling. *Transport*/*session* layers are not usually integrated in NoCs infrastructures.

This work has two main objectives. The first one is to discuss performance trade-offs for switching modes and physical channel allocation policies. The second objective is to propose the addition of a *session layer* over circuit switching, including the management of several simultaneous sessions per router, to improve the overall NoC performance. Buffering strategies, routing algorithms and arbitration policies are not discussed here.

This paper is organized as follows. Section 2 discusses switching modes employed in NoCs, highlighting pros and cons of circuit and packet switching. Section 3 presents NoCs employing Time Division (TDM) and Spatial Division (SDM) multiplexing, justifying the use of spatial multiplexing in current technologies. Section 4 contains the main contribution of this work, the proposition to add a session layer coupled to circuit switching. Section 5 presents experimental results and Section 6 concludes this paper.

2. SWITCHING MODES IN NOCS

Wormhole packet switching is the most common switching mode employed in NoCs [2]. Packet-switched networks often allow for high aggregate system bandwidth, as many packets can be in flight at a given instant [3]. However, they generally require congestion control and packet processing, which includes buffers to queue-up packets awaiting the availability of the routing resources. Correct buffer sizing is a fundamental parameter to optimize NoC performance. Small buffers increase network congestion and large buffers increase the area overhead. This switching mode supports well best-effort services [4], being more efficient for traffics with short and frequent packets. HERMES [5], Xpipes [6], MANGO [7] and SoCIN [8] are examples of NoCs employing wormhole packet switching.

Circuit switching provides throughput guarantees and latency bounds, since an exclusive path is allocated to data transfers between source and target IPs. In addition, the buffering requirement is typically a single register instead of a FIFO buffer, since when the circuit is established the NoC acts like a pipeline. However, the disadvantages of this switching mode are the channel bandwidth underutilization when traffic is transmitted at lower rates and the setup latency to establish a circuit, which is a function of the traffic in the path during circuit establishment. This switching mode is more efficient for traffics with long packets at high rates, with requirements for throughput and latency guarantees. Representative circuit-switching NoCs are: PNoC [3], Æthereal [9], SoCBUS [10] and Octagon [11]. Æthereal employs circuit switching only for traffic with QoS requirements, while BE traffic uses wormhole packet switching.

Table 1 summarizes the main pros and cons of circuit and packet switching.

Table 1 - Pros and cons of circuit and packet switching.

	Pros	Cons
circuit switching	– Guaranteed throughput and latency – Single register instead of FIFO buffers	– Static path reservation and possibly wasted bandwidth
wormhole packet switching	– Shared NoC resources, enabling to send multiple flows simultaneously	– Under heavy traffic, flits may block an important number of routers – Wasted bandwidth when the traffic initiator rate is slower than the channel rate

This paper proposes the use of circuit switching with fixed size packets, similar to the *cell* concept used in ATM [12]. Here, a *cell* is first buffered and then transmitted to its target using circuit switching. The advantages of using circuit switching with buffered *cells* are:

1. a cell is sent to its destination if and only if a path exists between source and target IPs, avoiding network congestion;
2. a cell is transmitted at the network rate, not at the IP rate, improving channel bandwidth allocation (burst transmission);
3. buffering in routers is reduced, due to the use of circuit-switching.

On the other hand, due to the burst transmission, the source IP must have a buffer to store at least one cell, which may increase packet latency.

A good trade-off, as in the Æthereal [9] network, is to combine the two switching modes. Packet switching may be employed for BE traffic while the proposed *cell-based* circuit switching deal with QoS traffic.

3. MULTIPLEXING STRATEGIES IN NOCS

NoCs may be modeled as graph G=<R,L>, where the vertex set *R* is a set of routers, and the edge set *L* represents its bidirectional communication links. Each link contains two unidirectional channels, enabling the communication between neighbor routers. Channels can be multiplexed, spatially or temporally, allowing the use of a same channel by different flows, improving the NoC performance. NoC literature describes the use of time and spatial division multiplexing.

Time division multiplexing (TDM) shares physical channels, dividing them into logical channels (or virtual channels - VCs) [13]. In this scheme, the time is discretized in equal-size periods called time-slots. During a time-slot, the available bandwidth is exclusively dedicated to a flow. TDM reduces congestion, and consequently improves NoC performance. The insertion of VCs also allows the use of special policies to reserve time slots for certain flows, i. e. bandwidth reservation, enabling QoS support. However, an individual buffer is required for each VC, and a time slot table is required to store VCs allocation (this table is required when a priority scheme is employed to guarantee QoS). The additional buffers and the slot table increase the power and silicon area [14]. Æthereal [9] and Nostrum [15] are two representative NoCs employing virtual channels.

In current technologies, a phit size equal to 32 or 64 bits underutilizes the amount of wires that can be implemented to connect neighbor routers. Consider for example a 90 nm technology, 140 nm wire pitch and 0.1 mm^2 router area [16]. Each router could be connected to its neighbor through 715 wires (Figure 1), considering the use of only one metal layer. Therefore, this scenario favors the use of spatial multiplexing in lieu of temporal multiplexing. NoC designs employ either Spatial Division Multiplexing (SDM) [14] or Lane Division Multiplexing (LDM) [17].

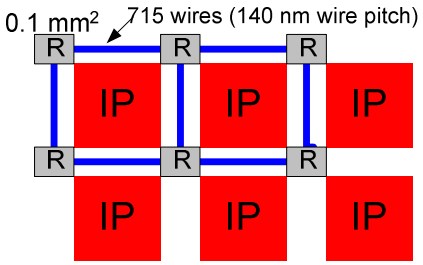

Figure 1 – Number of available wires to connect routers (R), in current technologies (90 nm example).

Leroy et al. [14] divide the channel in groups of wires. The number of wires assigned to each flow is a function of its required bandwidth. This method allocates each sub-set of wires for the whole connection lifetime (as in circuit switching). Data must be serialized and de-serialized at the source and target IPs, respectively. Results presented in this work, using as a case-study a video application, show a gain of 8% on energy consumption and 24% router area reduction, compared to a TDM router implementation. However, SDM increases the critical path by 37%.

In a similar work, Wolkotte et al. [17] propose the Lane Division Multiplexing (LDM) technique, also employing circuit switching. Differently from [14], this work divides each channel into fixed size lanes. Results presented by the Authors, comparing the circuit switching LDM router to two packet switching routers, show lower power consumption, a smaller chip area and higher maximum throughput. The disadvantages of LDM are the lack of flexibility in router design (fixed lane size) and no support for BE traffic.

Figure 2 illustrates a typical TDM router architecture. The main router components are: (*i*) a switch control; responsible for arbitration and routing; (*ii*) a crossbar, to connect the input ports to the output ports; (*iii*) input FIFO buffers for temporary flit storage. It is important to observe in this Figure the presence of de-multiplexers at the input ports, and multiplexers at the output ports, which significantly increase the router area.

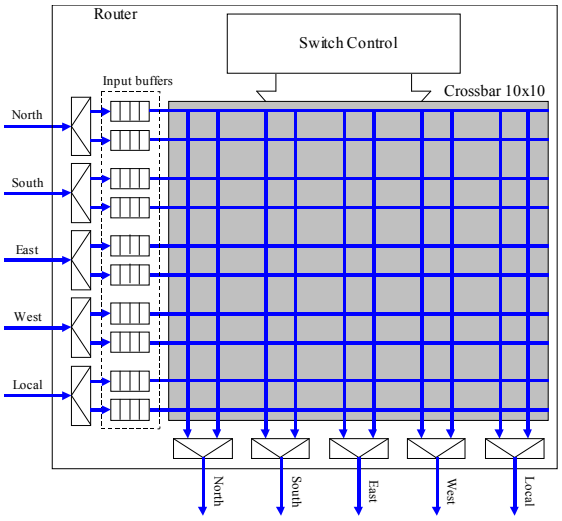

Figure 2 - Router with virtual channels (TDM).

SDM and LDM allocate wires for a given flow in function of the required bandwidth, requiring control circuitry and serialization and de-serialization modules. Given the amount of area available for wires, the *proposed router architecture* replicates the physical channels in all directions (N, S, E, W, Local), avoiding the extra circuitry of SDM and LDM. Figure 3 illustrates the router architecture employing replicated physical channels.

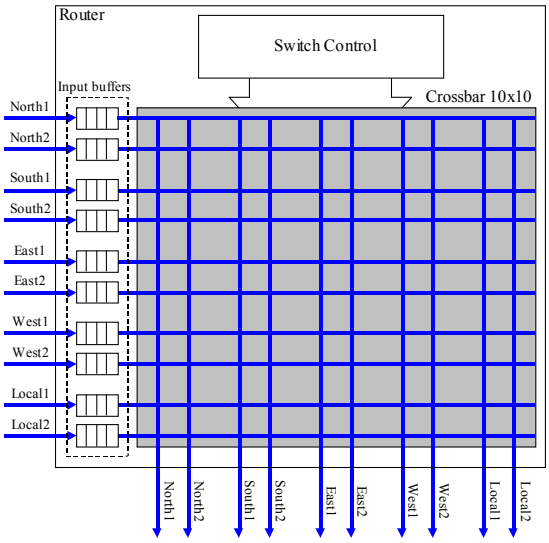

Figure 3 - Router with replicated physical channels.

Note in Figure 3 the suppression of de-multiplexers and multiplexers (compared to Figure 2), which significantly reduces the router area, as will be shown in the Results Section. The switch control complexity in both approaches is similar, since its main function is to control the internal crossbar. The input buffers of both approaches have the same size, requiring the same amount of silicon area.

The replicated channels approach doubles the router bandwidth, when compared to the same number of virtual channels. Also, as can observable in Figure 3, the Local port may receive *n* distinct flows, where *n* is the replication degree. This feature allows connecting *n* IPs to the same router, thus reducing the number of required routers and the total SoC area.

4. CIRCUIT SWITCHING AND SESSION LAYER

The reasoning behind the use of circuit switching coupled with session layer resides in the higher bandwidth of the NoC compared to individual application rates. Consider for example a 16-bit 200 MHz router: the available bandwidth per channel is 3.2 Gbps. In contrast, the rate of an application requiring a large amount of bandwidth, such as an HDTV stream (MPEG2), is 15 Mbps.

The basis of the proposed session layer is to couple the application rates to the channel rates. This is achieved by first packing data in the *source buffer* (at the output of the source traffic generator, Figure 5), and then transmit the packet to the NoC in burst. Figure 4 illustrates an application producing data with a rate inferior to the NoC link rate, and the corresponding packaging before transmission to the NoC. Here, fixed size packets (*cells*) are adopted. This simplifies buffer sizing and session management. This *source buffer* ensures data transmission (one cell) at the channel rate, avoiding idle time between flits, maximizing the use of channel bandwidth.

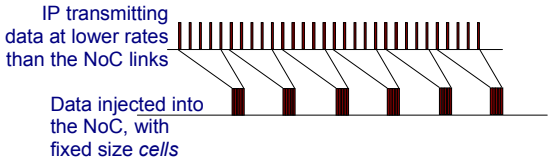

Figure 4 – Data packaging in cells, for coupling NoC and application rates.

Messages can be transmitted using connection (circuit switching) or connectionless methods (wormhole packet switching). With wormhole packet switching cells may be blocked inside the network, increasing the latency. The major benefit on using cells comes when circuit switching is associated to the use of a session layer. The proposed method employs the definitions detailed below.

Definition 1: *Physical connection*. Corresponds to the establishment of a circuit between the source and target IPs, *for each* cell of the message.

Definition 2: *Session*. Corresponds to the reservation of one of the Local ports at the target router (target IP) for all cells coming from the source IP. The session is established by the first cell of the message, being released in the last cell of the message. This reservation is necessary to avoid the interleaved reception of cells belonging to different source IPs at the same port.

For each cell, a control packet, using wormhole packet switching, establishes the *physical connection*. According to the cell position inside the message, different control packets are employed: (*i*) first cell; (*ii*) middle cell(s); (*iii*) last cell. When the control packet reaches the target IP, this router back propagates an **ACK** (acknowledge) signal, setting up the circuit. Next, the cell is transmitted, one flit per clock cycle per hop (circuit switching). A physical connection is closed when the last flit of the current cell is transmitted, using a sideband signal named EOP[1]. Note that control packets may find congestion, increasing the time to set up the circuit.

A *session* may be established when the first control packet requiring a physical connection arrives at the target IP. If the target IP is not reserved, the first acknowledge signal sets up both physical connection and session. If a session is already established with the target IP, a non-acknowledge is back propagated to the source IP indicating that, even if a path exist in the network, the target IP is already receiving data from another IP. The non-acknowledge signal releases all reserved resources between the source and target IPs. If no session is available, the source IP tries to set up a new session after a certain amount of time (in this implementation, a time proportional to the duration to transmit one cell). The session remains active up to the last message cell. The method to transmit messages with variable sizes (e.g. video frames, Ethernet packets, cache blocks) can be summarized as follows:

1. Store data in the source buffer and require a session establishment through a physical connection procedure.
2. Transmit the remaining cells, except the last one, through physical connections (one per cell), using the active session.
3. Transmit the last cell, through a physical connection, closing the active session.

The use of circuit switching, coupled with session layers improves network performance, because all cells are sent at the network rate. Resource reservation during circuit switching does not reduce performance, since the cell is already stored in the source buffer.

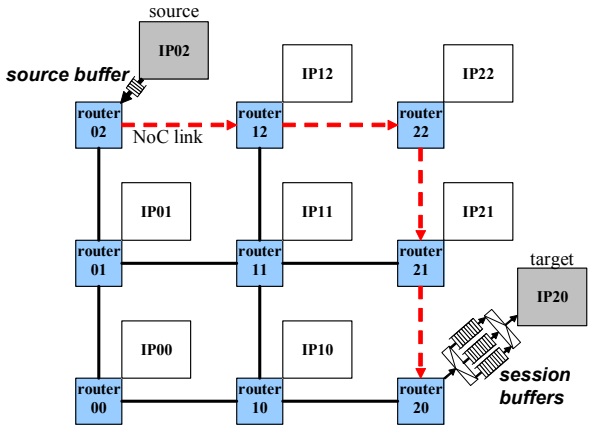

Figure 5 –Buffers included in the system when allowing multiple sessions per IP.

A bottleneck of the proposed approach arises when multiple sources try to simultaneously connect to the same target IP. The solution to this problem is to include in the target IP *session buffers*. Figure 5 illustrates the placement of source and target session buffers. Using session buffers, the target IP may receive *k* simultaneous flows, being *k* the number of session buffers. The *session buffers* must be sized to store at least one complete message (for example, an Ethernet packet or a cache block).

[1] The EOP (end of packet) signal enables the use of variable size packets. An exceptionally sized cell in the approach described is the last cell of a message, which can be smaller than the cell size.

5. RESULTS

A NoC implementation applying the proposed methods is available, and the results of evaluating it are the object of this Section. The NoC implements the methods directly in RTL VHDL, and derives its structure from the *HERMES* NoC [5] infrastructure. Performance figures like latency and total time to deliver messages derive from the use of RTL simulation of the code.

5.1 Virtual Channels versus Replicated Channels

This Section compares the architectures presented in Figure 2 (virtual channels) and Figure 3 (replicated channels). Table 2 presents the common features of both architectures. This experiment evaluates only multiplexing strategies, without employing circuit switching.

Table 2 - Common features for both architectures.

Flit/phit size	8 bits
Flow control	credit based
NoC topology	mesh 4x4
Routing algorithm	deterministic XY
Switching mode	packet switching/wormhole

Input buffers have the same size in both architectures. The virtual channel architecture has 8-flit deep buffers for each logical channel. The replicated channel architecture has 8-flit deep buffers for each physical channel.

The goal of using virtual or replicated channels is reducing congestion when different flows compete for the same path inside the network. Figure 6 illustrates the traffic scenario used to evaluate latency. This scenario is justified by the amount of concurrent flows in the same channel. Lines indicate the path taken by packets from source to target routers. Ellipses highlight channels where two flows compete for a link.

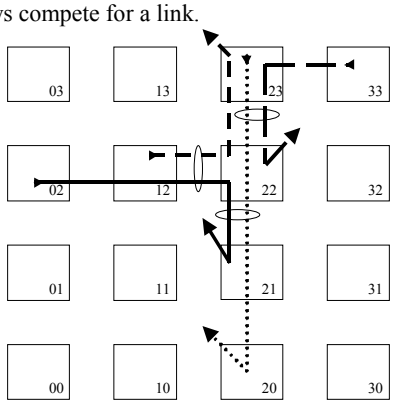

Figure 6 – Spatial traffic distribution for latency evaluation when comparing virtual and replicated channels.

Each source-target pair in Figure 6 transmits 500 257-flit packets (2 header flits and 255 payload flits). Packets enter the network at the channel rate. Table 3 presents the average latency to transmit one packet, in clock cycles. The presented latency includes the network latency, proportional to the number of hops, and the packet latency, proportional to the packet size.

Table 3 - Average latency values (clock cycles).

Source	Target	Virtual Channels Latency	Replicated Channels Latency
02	21	580	305
12	23	546	290
33	22	556	302
23	20	570	290

Replicated channels reduce 47.3% the average latency, when compared to virtual channels. This is an expected result, since the internal NoC bandwidth doubles. When there is no congestion, both approaches have equal latency.

Table 4 presents area consumption for FPGA mapping. For a single router (5 ports routers columns), a 12% area reduction is observed when using the replicated channels approach. For a 4*x*4 NoC the area reduction is 15%.

Table 4 – Area results for Virtual Channels (VC) and Replicated Channels (RC), targeting a Virtex 2VP30 FPGA.

Resource	5 ports router		4 x 4 mesh Noc		Available
	VC	RC	VC	RC	
Slices	861	758	10538	8904	13696
LUTs	1722	1515	21075	17808	27392
Flip Flops	455	398	5866	5057	29060

Table 5 presents area consumption data for ASIC mapping, considering the number of equivalent gates and a macro-cell "16x16 bits memory blocks" to implement the buffers (5 per router). For the single router and the 4*x*4 NoC, a 4% and 6.4% area reduction is observed when using the replicated channels approach.

Table 5 – Area results for Virtual Channels (VC) and Replicated Channels (RC), targeting to an ASIC library.

Resource	5 ports router		4 x 4 mesh Noc	
	VC	RC	VC	RC
Equivalent gates	6709	6416	83952	78759
16*x*16 bits memory blocks	5	5	64	64

5.2 Session layer evaluation

The second experiment evaluates the benefits of adding a session layer over circuit switching. The NoC has the features presented in Table 2 (except for the switching mode, which is now circuit switching), with 8-flit depth input buffers. Single links connect routers, with neither virtual channels nor replicated channels.

Figure 7 illustrates the traffic scenario used to evaluate the behavior of circuit switching coupled to session layer. All flows have at least one flow competing for the same resources. All six traffic initiators (represented as gray squares) send one 1280-byte message. The initiators IP00 and IP10 start transmitting first, inducing blocking situations for the remaining initiators.

As explained before, the goal of using circuit switching coupled with session layer is to reduce internal NoC resources reservation when IPs are transmitting data at rates lower than the NoC rate. The link rate is 400 Mbps (from a NoC frequency of 50 MHz and a flit size of 8 bits). The rate of traffic initiators varies from 66.4 Mbps (the rate relative to the link rate is 16.6 %) to 160 Mbps (relative rate equals 40%).

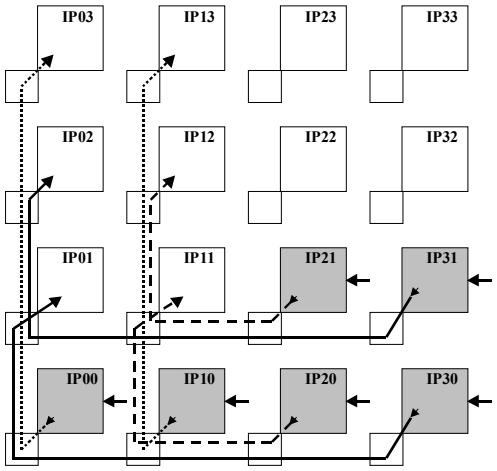

Figure 7 – Spatial traffic distribution used to evaluate circuit switching with session layer.

Figure 8 illustrates the number of clock cycles to transmit all messages as a function of the cell size (*CS*) and of the injection rates (*IR*) for the traffic scenario of Figure 7, where flows compete for the same links.

For lower injection rates (16.6 to 25%) the time spent transmitting messages increases linearly with the cell size (except at the point CL=32 and IR=25%). The idle time between cells in lower IR favors the sharing among flows in the same link and reduces the impact of circuit switching connection establishment time. The linear growth observed here is due to the latency of the last transmitted cell.

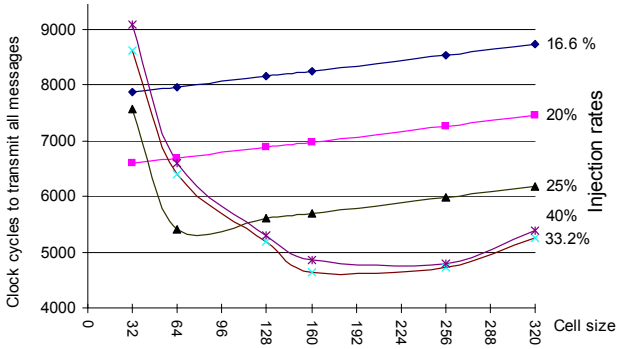

Figure 8 – Latency as a function of the cell size and injection rate.

The connection establishment time at higher IR penalizes smaller cell sizes. For example, the time spent to connect two routers within a 3-hop path is 25 clock cycles, if there is no contention. Transmitting a 32-flit cell requires 32 clock cycles. Consequently, each small cell has its latency doubled due to connection establishment. As cell size grows, more data flows per physical connection. However, for larger cells the link bandwidth is dominated by one flow, increasing again the time to transmit the messages (curves 33.2% and 40%).

This experiments points out to an intermediate cell size as the best compromise, for example 128 flits. Smaller cells penalize the performance, due to connection establishment time and bigger cells increase congestion inside the NoC.

The previous experiment creates only one session per IP, since all sources have different targets. A third experiment exploits multiple sessions. Here, IP30, IP20, IP10 and IP00 simultaneously transmit one 1280-byte message partitioned into 10 128-byte cells to IP03. Besides the same target IP, competition occurs also in internal links. The number of clock cycles to deliver all messages is 7440, 8740, 10305 and 12037 for 4, 3, 2 simultaneous sessions and for a single session, respectively.

If it exists only one session per IP, the target IP does not require a buffer session. However, the cost to add simultaneous sessions is one *buffer session* per session, each one sized to the longest possible message size.

In a NoC design, only few IPs are expected to receive simultaneous sessions. For example, in an MPSoC, shared memories may receive simultaneous write messages, or a communication IP may also receive simultaneous Ethernet packets to transmit to the external world. In such situations, simultaneous sessions are a solution to reduce hot spots, and the overall latency.

6. CONCLUSIONS AND FUTURE WORKS

This paper proposed methods to reduce the overall latency on NoCs. Results show significant performance gains, demonstrating the effectiveness of the propositions, even with higher injection rates and flows competing for the same physical channel. Both methods achieve latency reduction through congestion reduction. Replicated channels increase router bandwidth, whereas circuit switching coupled with a session layer maximizes the physical channel utilization.

Channels replication relies on available routing area in deep submicron technologies. The method reduces both latency and gate area, and it is an alternative to the use of virtual channels.

Session layers share the physical channels similarly to virtual channels. The main difference relies in the abstraction level. Virtual channels share the physical channels at the packet level, while a session layer shares the physical channels at the flow level. This technique can also be used to reduce hot spots, since it allows IPs to handle several simultaneous connections.

Future works include evaluating NoCs employing replicated channels together with session layers and the analytical definition of the cell size as a function of message sizes and input rates.

7. ACKNOWLEDGMENTS

This research was supported partially by CNPq (Brazilian Research Agency), project 300774/2006-0.

8. REFERENCES

[1] Dehyadgari, M.; et al. "*A new protocol stack model for network on chip*". In: Emerging VLSI Technologies and Architectures, 2006.

[2] Bjerregaard, T.; Mahadevan, S. "A survey of research and practices of Network-on-chip". ACM Computing Surveys, v.38(1), 2006, pp. 1-51.

[3] Hilton, C.; Nelson, B. "*PNoC: a flexible circuit-switched NoC for FPGA-based systems*". Computers and Digital Techniques, v. 153(3), May 2006, pp. 181-188.

[4] Jantsch, A.; Tenhunen, H. "*Networks on Chip*". Kluwer Academic Publishers, 2003, 303p.

[5] 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.

[6] Benini, L.; Bertozzi, D. "*Xpipes: A Network-on-Chip Architecture for Gigascale Systems-on-Chip*". In: DATE, 2004, pp. 18-31.

[7] Bjerregaard, T.; Spars, J. "*Router Architecture for Connection-Oriented Service Guarantees in the MANGO Clockless Network-on-Chip*". In: DATE, 2005, pp. 1226-1231.

[8] Zeferino, C. A.; Susin, A. A. *"SoCIN: a Parametric and Scalable Network-on-Chip*". In: SBCCI, 2003, pp. 169-174.

[9] Goossens, K.; et al. "*Æthereal network-on-chip: concepts, architectures, and implementations*". IEEE Design & Test of Computers, v. 22(5), Sept.-Oct. 2005, pp. 414-421.

[10] Wiklund, D.; Liu, D. "*SoCBUS: Switched Network-on-Chip for Hard Real Time Embedded Systems*". In: IPDPS, 2003, pp. 113-116.

[11] Karim, F.; Nguyen, A.; Dey, S. "*An Interconnect Architecture for Networking Systems on Chips*". IEEE Micro, v. 22(5), Sept.-Oct. 2002, pp. 36-45.

[12] Giroux, N.; Ganti, S. "*Quality of Service in ATM Networks: State-of-Art Traffic Management*". Prentice Hall, 1998, 252 p.

[13] Mello, A.; et al. "*Virtual channels in networks on chip: implementation and evaluation on HERMES NoC*". In: SBCCI, 2005, pp. 178-193.

[14] Leroy, A.; et al. "*Spatial Division Multiplexing: a Novel Approach for Guaranteed Throughput on NoCs*". In: CODES-ISSS, 2005, pp. 81-86.

[15] Millberg, M.; et al. "*Guaranteed Bandwidth Using Looped Containers in Temporally Disjoint Networks within the Nostrum Network-on-Chip*". In: DATE, 2004, pp. 890-895.

[16] Leibson, S. "The Future of Nanometer SOC Design". In: SOC, 2006, pp. 1-6.

[17] Wolkotte, P. T.; et al. "An Energy-Efficient Reconfigurable Circuit-Switched Network-on-Chip". In: IPDPS-RAW, 2005.