Quantitative Modelling and Comparison of Communication Schemes to Guarantee Quality-of-Service in Networks-on-Chip

Mehmet Derin Harmanci, Nuria Pazos Escudero, Yusuf Leblebici and Paolo Ienne
Swiss Federal Institute of Technology Lausanne
Processor Architecture and Microelectronic Systems Laboratories
IN-F and EL-D Ecublens, 1015 Lausanne, Switzerland
Email: mehmetderin.harmanci/nuria.pazos/yusuf.leblebici/paolo.ienne@epfl.ch

***Abstract*— We address the quantitative comparison of two basic connection schemes for on-chip communication, connection-oriented and connection-less, regarding their enforcement of a defined *Quality of Service* for the communication. For such comparison, we have built similar models for the two approaches, whose simulation highlights the drawbacks and opportunities of each solution. We demonstrate that for variable bit-rate applications, the end-to-end delay for the individual flows is less stable in the connection-oriented scheme than in the connection-less scheme.**

I. Introduction

Considering the level of integration enabled by recent silicon technology advances, reliable communication of the system components becomes a major concern. On-chip communication has to face new challenges in a billion-transistor *System-on-Chip* (SoC) paradigm, such as scalability, performance, reliability and energy reduction. Reusing simple bus architectures for the communication does not satisfy the mentioned requirements. A promising alternative that has emerged in the last years is the design of *Networks-on-Chip* (NoC).

The performance constraints imposed to the related components require predictability of inter-block communication, which implies the assurance of a defined *Quality of Service* (QoS) for the communication. QoS is characterized by diverse parameters, such as availability, delay, jitter, packet loss, and throughput. It requires the implementation of additional mechanisms, which bring new overheads.

Over the past years, several NoC architectures supporting different communication schemes and techniques to guarantee QoS parameters in the on-chip traffic have emerged. A first classification of the existing techniques distinguishes between connection-oriented and connection-less communication schemes. We believe that a quantitative comparison of these two general techniques is necessary in the design process of an on-chip network with certain traffic guarantees. To our knowledge, there is no quantitative comparison between these two main communication schemes for on-chip traffic.

When pursuing an accurate comparison of the previous two on-chip communication approaches, it is first necessary to define a modelling and simulation framework where both communication schemes can be quantitatively analyzed. Therefore, we propose a common simulation platform and similar implementation models of both communication schemes whose analysis highlights their respective characteristics.

The rest of the paper is organized as follows: Section II introduces the existing techniques to provide service guarantees to on-chip communication. Section III surveys related work. In Section IV the modelling framework for comparing connection-oriented versus connection-less communication schemes is depicted. Section V shows the simulation results for both schemes in terms of end-to-end delay using as case study a MPEG-2 video decoder application. Finally, Section VI concludes the work.

II. Guaranteeing Quality-of-Service in NoCs

Providing *Quality-of-Service* (QoS) for on-chip communication is becoming an important concern, mainly driven by the necessity of global predictability. The aspect of predictability is crucial, for example, to enable design reuse within a SoC.

Early packet-switched NoCs mainly followed a connection-less communication scheme and provided *Best-Effort* (BE) traffic. This means that all packets are handled in the order they arrive at the system as long as there are sufficient resources available. BE traffic provides a good utilization of communication resources, but it is not able to guarantee any traffic parameter. To overcome this drawback, two main techniques can be applied: (1) building a connection-oriented communication on top of the packet-switched on-chip network (e.g., virtual circuits), or (2) implementing additional services to meet approximately the predefined QoS parameters (e.g., prioritization of flows).

The first technique (1) provides a connection-oriented distinction between flows. Connection-oriented communication is characterized by resource reservation. That is, flows must set up paths through the network and reserve resources at each networking-node. Although this scheme guarantees tight bounds for several traffic parameters, its main disadvantages are an inefficient resource reservation, a costly overhead due to connection setup, and its non-scalability.

The implementation of additional services (2) mainly consists of aggregating traffic into different classes at the network edge and scheduling the forwarding of packets for each class within the network. This communication-less communication scheme offers a better adaptation to varying network traffic and a potentially better utilization of network resources. Nevertheless, it provides a poorer QoS support than connection-oriented techniques in that it only offers relative service guarantees with different sensitivities to delay and loss.

In this work, we present a quantitative comparison of the two techniques to provide QoS support on NoCs. We demonstrate that in a connection-oriented network, an erroneus decision in terms of bandwidth reservation when setting up a connection might lead to an unexpected time penalty. While in a connection-oriented network, a non-optimal priority selection for the flows has less impact in terms of latency.

III. State-of-the-Art

After the emergence of NoC as an alternative to buses for on-chip communication [1], several solutions based on different topologies, router architectures, communication schemes, etc. have been proposed [2].

But not many research groups have addressed the importance of providing QoS in packet-switched on-chip networks, yet. One of the most relevant approaches in this area, Æthereal by the Philips Research Laboratories [3], supports *Guaranteed-Throughput* (GT) for real-time applications and *Best-Effort* (BE) communication. The GT is achieved by the implementation of virtual circuits on a connection-oriented system. Nevertheless, the main drawback of these connection-oriented communication systems is a poor scalability and an inefficient resource utilization [4]. This makes them not optimal for variable bit-rate applications. On the other hand, an attempt to provide relative guaranteed services on top of connection-less on-chip networks has been presented in [5]. It borrows several concepts of the DiffServ technology from Internet networks and adapts them to NoCs. Despite the fact that it does not provide tight guarantees, it improves the resource utilization of previous schemes and is better adapted to variable bit-rate applications.

IV. Simulation Models for Connection-oriented and Connection-less Communication Schemes

As mentioned in the introduction, we have built comparable models in SystemC for the two communication schemes, connection-oriented and connection-less. The first one is inspired to the architecture proposed by the Philips Research Laboratories (Æthereal [3]), whereas the second model relies on the prioritization of flows introduced in [5] on the so-called DiffServ-NoC. In both models, a $N \times M$ matrix of connection requests $C(i_n, o_m)$ is created, where i_n and o_m represents the input and the output of the router for the connection requested, and N and M are the total number of inputs and outputs of the router. For the connection-oriented model $C = f(Q, C_{GT})$, where $Q(i_n, o_m)$ is a $N \times M$ matrix made of *Best-Effort* (BE) requests, and $C_{GT}(i_n, o_m)$ is a $N \times M$ matrix made of *Guaranteed-Throughput* (GT) connections that have been reserved and are present in the current iteration. The entries of both matrices is set to 1 if the connection from i_n to o_m is requested. For the connection-less model $C = f(Q)$, where $Q(i_n, o_m)$ is a $N \times M$ matrix made of every connection request. In this case, the entries of the matrix are set to the priority number p associated with the connection request. The processing cycle of the router is divided in four slots, while the number of clock cycles assigned to one slot is parametrizable. A hop-by-hop flow control mechanism is not yet implemented and therefore the size of the input buffers is set to the worst case.

A. Connection-oriented Model (e.g., Æthereal)

Both, the router and the network interface models, built for analyzing the connection-oriented scheme, are somehow based on the interconnection concept presented in Æthereal. The router consists of two parts: the *Guaranteed-Throughput* (GT) and *Best-Effort* (BE) routers, which are combined in a single implementation sharing resources, such as the switch. Fig. 1 shows the control and data path of such packet-switched router. It uses virtual output queueing with packet scheduling for BE traffic and time-division multiplexing scheme for GT traffic. For GT traffic communication channels are statically set up to transport data between hosts, while BE traffic is never lost—but no latency or throughput is guaranteed. A deterministic routing algorithm is performed in the network interface and the path is added to the header information (i.e., source routing).

The crossbar switch, present in the virtual output queued architecture, is controlled by a contention resolution algorithm (implemented by the *arbiter* in Fig. 1) which computes which inputs and outputs must be connected. Firstly, the GT router communicates to the arbiter the connections reserved by GT traffic in the current iteration. Secondly, the BE requests not conflicting with the GT connections create a bipartite graph.

The bipartite graph consists of a vertex i_n and o_m for every input port and output port, and an edge (i_n, o_m) for every non-conflicting BE request. A *match* is a subset of these edges such that every node is incident to, at most, one edge. The matching algorithm applied is based on three stages: (a) *request* which represents all the connection requests, (b) *grant* which grants only one input for each output, and (c) *accept* which accepts only one output for each input (The policy implemented by the algorithm is not explained due to limited space). One example is shown in Fig. 2 for a bipartite graph of $N = M = 5$.

B. Connection-less Model (e.g., DiffServ-NoC)

For the analysis of the connection-less scheme, we have adapted the model presented in [5] for a connection-less router and network interface with priority-based routing, to present a similar structure as the previous connection-oriented system. Fig. 3 shows the control and data path of such packet-switched router, which uses virtual output queueing with priority-based packet scheduling. As in the previous solution, the network performs source routing on the network interface side. Besides, the network interface is in charge of classifying the packets before entering the network by performing the *Recursive Flow*

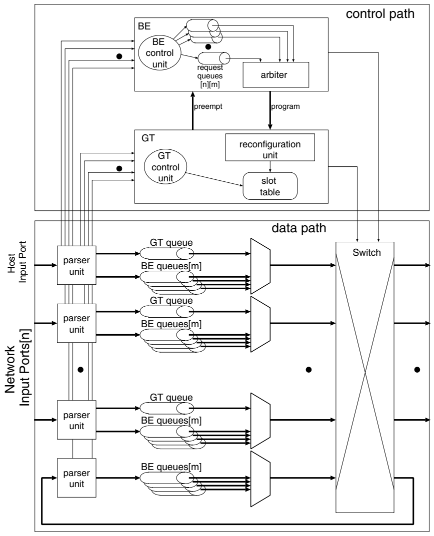

Fig. 1. Connection-oriented Router.

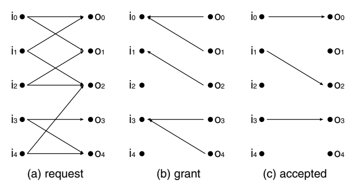

Fig. 2. Arbiter BE Connection-oriented.

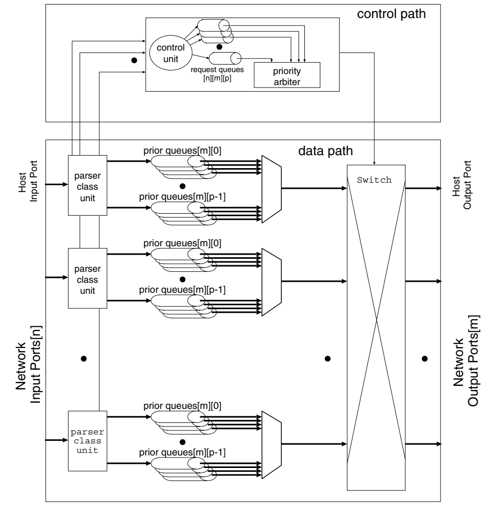

Fig. 3. Connection-less Priority-based Router.

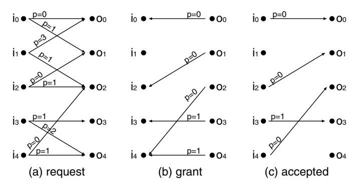

Fig. 4. Arbiter Priority-based Connection-less.

Classification (RFC) algorithm [6]. Depending on the assigned *class_id*, the packets will be forwarded with different priorities in the routers.

The crossbar switch is controlled by a contention resolution algorithm. Contrary to the previous model, it does not reserve any input-to-output connection for GT, but the different request queues compete to be accepted based on the assigned priority *p*. In this case, the priority-based bipartite graph consists of a vertex i_n and o_m for every input port and output port, and an edge (i_n, o_m) for every request, which has associated the priority value *p*. The matching algorithm ensures that every node is incident to, at most, one edge, taking the priorities of the edges into account when discriminating among convergent edges (p=0 represents the highest priority). It is based on the same three stages as the previous model. Fig. 4 depicts an example of a priority-based bipartite graph for $N = M = 5$.

V. Quantitative Comparison of Results

A. Case Study: MPEG-2 Video Decoder

The implementation of multimedia applications (e.g., MPEG-2 video decoder algorithm) on multi-core systems poses many challenges in terms of inter-core communication due to the variable bit-rate traffic. This is the main reason to use it as case study for comparing the target communication schemes. Fig. 6 shows the functional data flow of the video decoder algorithm as well as its partitioning into concurrent tasks communicating through a NoC infrastructure. Depending on the type of video frame, there will be different streams of data, some of them in parallel (e.g., flows (a) and (b)), flowing through the network).

For the comparison of connection-oriented and connection-less schemes, we have built two similar 4×4 networks, where the four nodes running the MPEG-2 algorithm are surrounded by nodes generating random noise (in terms of interval time, message size and type of service). For the connection-oriented approach, a flow from a source to a target node can select between GT service (reserving 25%, 50%, 75% or 100% of bandwidth) or BE traffic. For the connection-less solution four different classes of service or priorities have been defined.

B. Simulation Results

The presented results have been obtained running the parallelized MPEG-2 video decoder algorithm with a MPEG-2 sequence composed of three frames.

Fig. 5 shows the individual end-to-end delay for the parallel flows (a) and (b) when applying the two communication schemes to be compared. For the connection-less network (I) six simulations are run changing the priorities of the parallel flows under study and using i) low priority noise and ii) random noise. For the connection-oriented network (II) four experiments are performed changing the percentage of bandwidth reserved for the parallel flows and using either BE noise or 25% bandwidth GT noise. The delays are displayed as impuls at the time a flow of data is sent to the destination node. The data recorded corresponds to the transition between the first and the second frame of the MPEG-2 test sequence. The first frame (a I frame) does not require any *Motion*

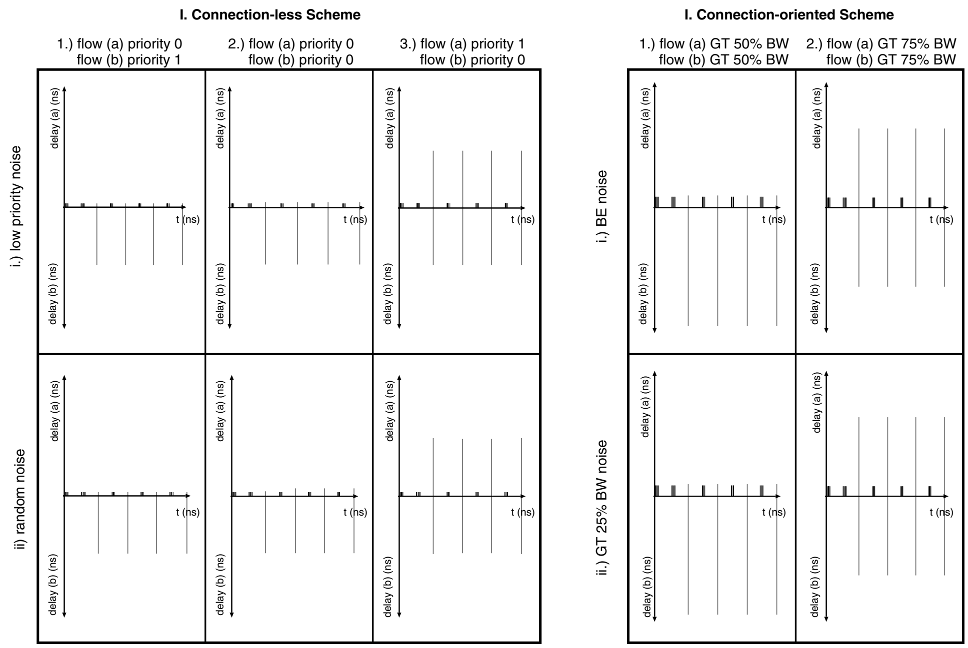

Fig. 5. End-to-End Delay for IDCT-ADD flow and MC-ADD flow in a connection-less and in a connection-oriented network

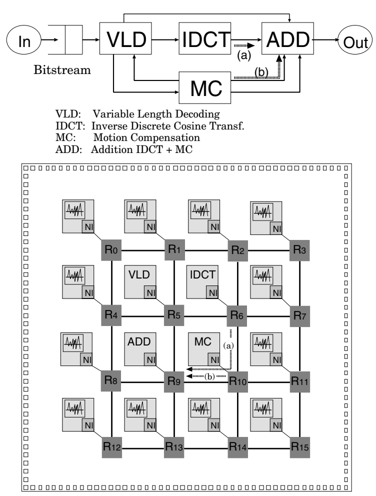

Fig. 6. Implementation of MPEG-2 Video Decoder on NoC.

Compensation (MC) and therefore there is no data flowing from MC to ADD (flow (b)), while in the second frame (a P frame) there is data sent from MC to ADD at the beginning of the processing of each macroblock, which shares the transmission link with flow (a).

From the differents graphs we can observe that: (i) the individual end-to-end delay for each flow is lower in the connection-less scheme than in the connection-oriented one, mainly due to the better adaptation of the first approach to variable bit-rate applications; (ii) the connection-less solution presents a higher stability towards a wrong decision in the type of service to be assigned to a flow (priority number in a connection-less network and BE/GT service—together with the percentage of bandwidth— for a connection-oriented network); and (iii) the two schemes seem to tolerate the presence of low priority or random generated noise.

VI. Conclusion

We have presented a quantitative comparison between two fundamental communication schemes in terms of end-to-end delay of individual flows. An accurate comparison has been possible thanks to a common simulation platform and similar implementation models for both schemes. The results have shown that by implementing additional services (e.g., prioritization of flows) on top of a connection-less communication network, this scheme is able to guarantee end-to-end delays for individual flows. Furthermore, the latency of the transmissions can be better predicted by defining priority classes (as in the connection-less scheme) than by assigning a certain bandwidth to the transmission (as in the connection-oriented approach).

References

[1] L. Benini and G. De Micheli, “Networks on chips: A new SoC paradigm,” *Computer*, vol. 35, no. 1, pp. 70–78, Jan. 2002.

[2] A. Jantsch and H. Tenhunen, Eds., *Networks On Chip*. Boston, Mass.: Kluwer Academic, 2003.

[3] E. Rijpkema, K. G. W. Goossens, A. Radulescu, J. Dielissen, J. van Meerbergen, P. Wielage, and E. Waterlander, “Trade-offs in the design of a router with both guaranteed and best-effort services for networks on chip,” in *Proceedings of the Design, Automation and Test in Europe Conference and Exhibition*, Munich, Mar. 2003, pp. 350–55.

[4] W. Dally and B. Towles, Eds., *Principles and Practices of Interconnection Networks*. Morgan Kaufmann Publishers, 2004.

[5] D. Harmanci, N. Pazos, P. Ienne, and Y. Leblebici, “Providing QoS to connection-less packet-switched NoC by implementing DiffServ functionalities,” in *Proceedings of the International Symposium on System-on-Chip*, Tampere, Finland, Nov. 2004, to appear.

[6] P. Gupta and N. McKeown, “Packet classification on multiple fields,” *ACM SIGCOMM Computer Communication Review*, vol. 29, no. 4, pp. 147–60, Oct. 1999.