A Predictable Communication Scheme for Embedded Multiprocessor Systems

Derin Harmanci, Nuria Pazos, Paolo Ienne, Yusuf Leblebici
Ecole Polytechnique Fédérale de Lausanne (EPFL)
Lausanne, Switzerland
Email: {mehmetderin.harmanci, nuria.pazos, paolo.ienne, yusuf.leblebici}@epfl.ch

Abstract— *Networks-on-Chip* (NoC) are emerging as a widely accepted alternative for the traditional bus architectures. However, their applicability by the system designers is far away from being intuitive due to their lack of predictability. This communication predictability can be obtained statically or dynamically. A dynamic allocation is more suitable for flexible multiprocessor systems and requires the implementation of a *Quality-of-Service* (QoS) mechanism. This paper explores the main QoS schemes suitable for such systems: connection-oriented and connectionless. The simulation results show that the connectionless scheme provides a better predictability in terms of message latency with an acceptable buffer requirement. This work provides the designer with valuable guidelines to choose a priori the QoS parameters such that they can be confident on the predicted results.

I. Introduction

Buses, traditional mainstream in system interconnect, are unable to keep up with increasing on-chip performance requirements. Interconnection networks —commonly referred as *Networks-on-Chip* (NoC)— are emerging as an attractive solution to meet current on-chip communication requirements and are becoming pervasive in digital systems. The use of an interconnect network allows the sharing of bandwidth by parallel flows and enforces regular, structured use of communication resources, making systems easier to design, debug, and optimize.

The need of global predictability in on-chip communication implies an efficient allocation of the system resources. Such allocation can be done statically or dynamically. Static allocation techniques have been broadly used in application specific systems where the communication demands between cores are calculated statically and the required resources are allocated accordingly [1] —see example for MPEG-4 decoder in Fig. 1 i). Nevertheless, in case the system has to be adapted to run different applications over time, dynamic allocation of the communication resources will be needed. For this purpose, the packets on the network will receive different services, allowing a more efficient allocation of resources. Using this differentiation among classes, the network guarantee a defined Quality-of-Service for the communication. Existing solutions supporting QoS for on-chip communication implement a semi-dynamic allocation. This connection-oriented scheme sets up long-term connections which guarantee the requested bandwidth during the connection lifetime. In order to avoid long connection setup delays, a statical pre-scheduling of the tasks should be performed — see Fig. 1 ii). However, in a more flexible multiprocessor system, where the tasks running on each node are not known a priori, a fully dynamic allocation technique might be needed — Fig. 1 iii). The question we target is whether the connection-oriented scheme is suitable for these more flexible multiprocessor systems or another allocation scheme better meets their requirements. A candidate scheme is the connectionless approach, which differentiates among services through the prioritization of flows.

In the current work, we address the previous question by exploring the effects of different on-chip workloads on message latency and buffer utilization for the two communication schemes. The synthetic traffic generated mimics different task traces encountered in embedded applications targeting on-chip multiprocessors. The results show that the connectionless scheme is more suitable for multiprocessor systems running various different tasks in parallel. It offers a better predictability in terms of message latency under different network load conditions and for different injections processes. Furthermore, in order to justify the overhead in terms of router buffers for the connectionless solution, the buffer occupancy as well as the buffer accesses for the two communication schemes at the same network load is analyzed. The results demonstrate a larger buffer requirement for the connectionless communication, but with a significantly smaller number of buffer accesses than that of the connection-oriented scheme.

The rest of the paper is organized as follows: Section II introduces different workloads representing different System-On-Chip applications. Section III presents the simulation models used for the analysis of the two communication schemes, connection-oriented and connectionless. In Section IV the simulation platform and the performed measurements are explained. The results for the message latency and buffer utilization are then depicted. Finally, Section V concludes the paper.

II. System-on-Chip Workloads

Behaviour of the network may differ considerably from one application to another (e.g., some applications running in multicomputers generate very long messages, while distributed, shared-memory multiprocessors with coherent caches generate very short messages). Up to now, there exists no standard traces for network evaluation and most performance analysis use synthetic workloads to mimic applications with different features.

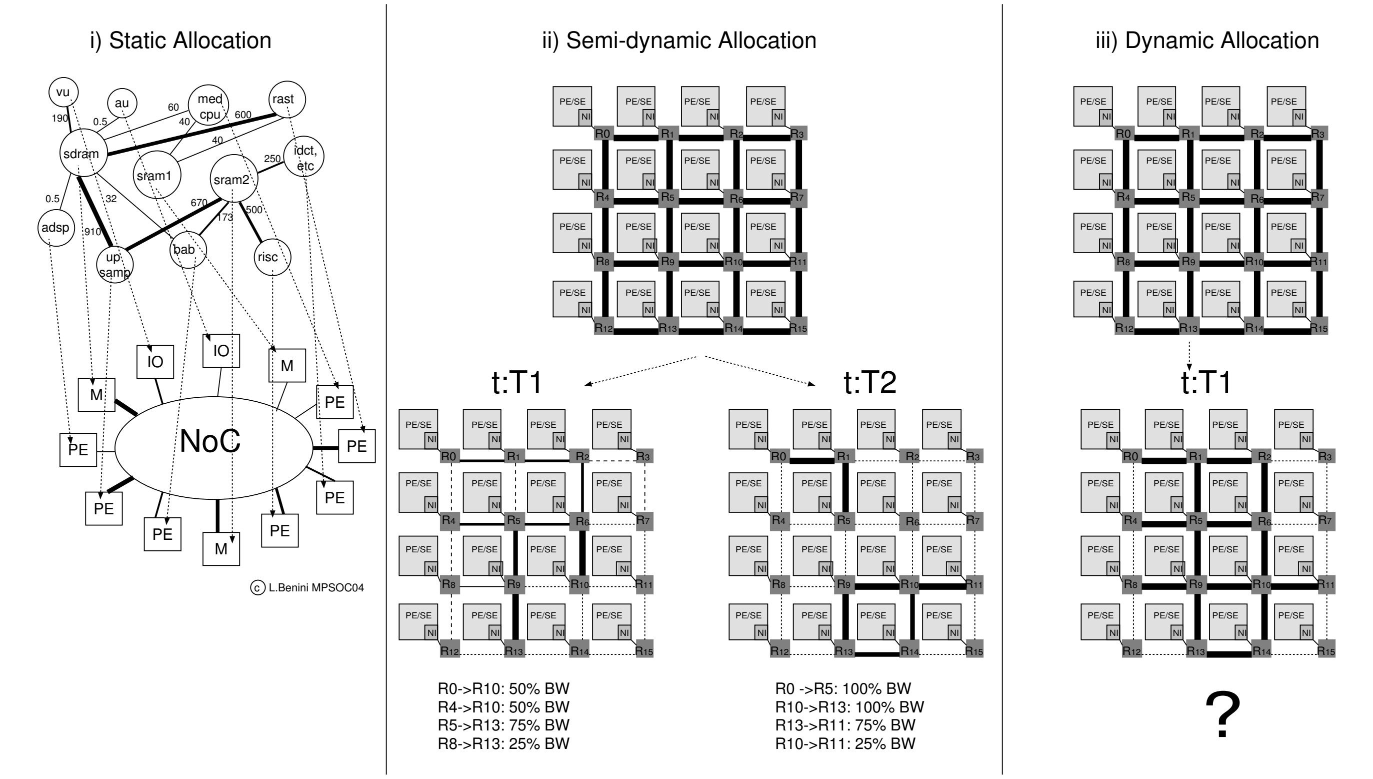

Fig. 1. Resource allocation techniques. i) Static allocation of required communication resources. ii) Semi-dynamic allocation of communication resources by setting up connections. iii) Dynamic allocation of communication resources by prioritization of flows

Network workload is the pattern of traffic applied at the network edges over time. For the selection of workload or input for a network simulation we have two main techniques [2]: (i) application-driven and (ii) synthetic generation. In (i), the sequence of messages applied to the network are generated directly from the intended application(s), which results in the most realistic traffic patterns. Nevertheless, it is difficult to achieve a thorough coverage of expected traffic and the feedback from the network can influence the workload. Synthetic traffic (ii) captures the salient aspects of the application-driven workloads, but are more easily designed and manipulated. If designed carefully, can capture the expected demands for the interconnection network and, at the same time, remains flexible. For the generation of synthetic traffic three main parameters are defined: injection process, distribution of destinations, and packet lengths [3]. For the purpose of the current work, the last technique, synthetic workload generation, is adopted in order to have a range of traffic profiles for different classes of traffic. The parameters applied as well as their implications are explained next.

A. Injection Process

- Bernoulli process: probability of injecting a packet is equal to the process rate. It results in geometrically spaced packet injections, but it lacks any state, i.e., not suitable for modelling time-varying or correlated traffic processes.
- Markov modulated process (MMP): popular model for modelling burstiness. The rate of a Bernoulli injection process is modulated by the current state of a Markov chain. During the bursts, injections occur with rate r_1. The injection process is quiet otherwise.
- Pareto process: Models long-range dependence, i.e., the probability of being in ON state for a given number of cycles decreases with polynomial law, and not geometrically as in ON/OFF Markov process [4]. For example, bursty traffic between on-chip modules in typical MPEG-2 video applications can be modelled as long-range dependent stochastic processes [5].

B. Distribution of Destinations

Destination for the next message at each node. Most frequently distribution is the uniform one, i.e., the probability of node i sending a message to node j is the same for all i and j, $i \neq j$.

C. Message Length

In most simulation runs, message length is chosen to be fixed (it may be varied from one run to another in order to study the effect of message length). Message length should be representative of the intended application.

III. Communication Schemes to Guarantee QoS

The models applied in the current work for analyzing the different communication schemes are described using the

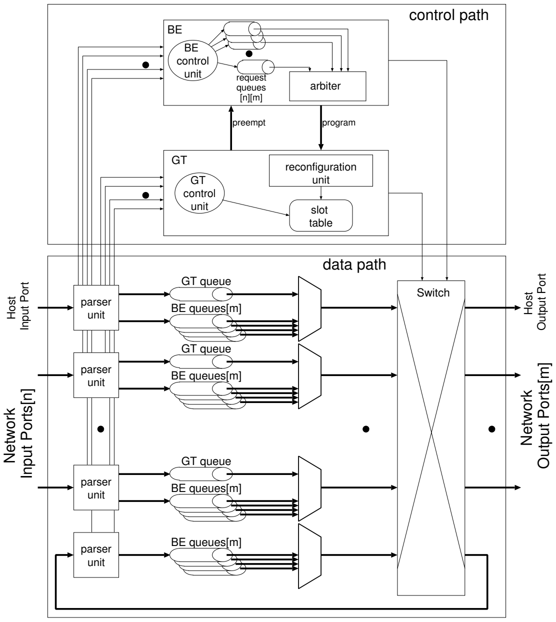

Fig. 2. Internal architecture of a router implementing connection-oriented communication.

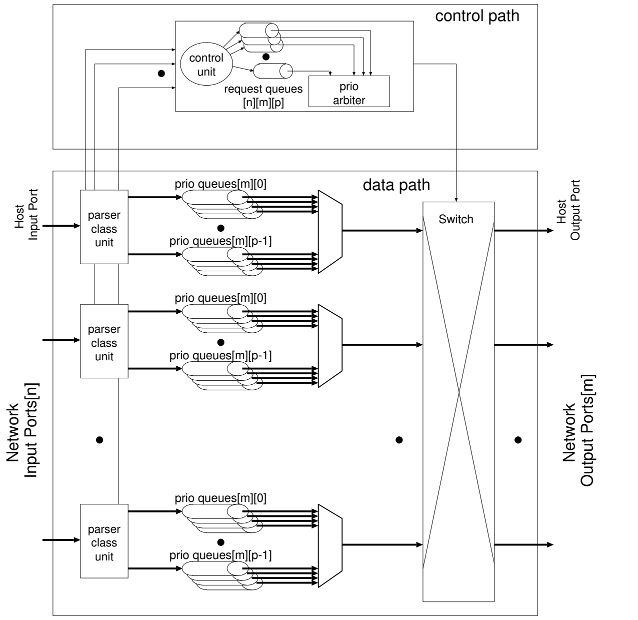

Fig. 3. Internal architecture of a router implementing connectionless communication based on prioritization of flows.

system level language *SystemC* at the abstraction level called transaction level modelling. Furthermore, they provide cycle-accuracy at the network interfaces, which is necessary for latency analysis. The connection-oriented model is inspired on the architecture proposed by the Philips Research Laboratories (Æthereal [6]), whereas the connectionless model relies on the prioritization of flows introduced in the so-called DiffServ-NoC [7].

A. Connection-oriented Simulation Model

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. 2 shows the control and data path of such packet-switched router. It uses virtual output queueing with packet scheduling for BE traffic and a time-division multiplexing scheme for GT traffic. For GT traffic communication channels are setup to transport data between hosts, while BE traffic is never lost —but no latency or throughput is guaranteed. The routing is performed statically in the network interface and the path is added to the header information (i.e., source routing).

A significant difference of our model with respect to Æthereal is that, contrary to the centralized connection setup policy implemented by Philips, our model performs a distributed policy (i.e., each network interface requests a connection setup without a priori knowledge of the connection being set or requested by the rest of the hosts in the network). A distributed policy decouples the functioning of the hosts and it is better scalable. The switch, present in the virtual output queued architecture, is controlled by a contention resolution algorithm (implemented by the *arbiter* in Fig. 2), that computes which inputs and outputs must be connected. At the entrance of the control path two boolean matrices are created: one consists of Best-Effort (BE) input-to-output requests, and a second one consists of Guaranteed-Throughput (GT) connections that have been reserved and are present in the current iteration. The entries of both matrix is set to 1 if the concerning input-to-output connection is requested. Then, the GT router communicates to the arbiter the connections reserved by GT traffic in the current iteration. Subsequently, the BE requests not conflicting with the GT connections create a bipartite graph. The bipartite graph consists of a vertex for every input port and output port, and an edge 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 reader is referred to [8] for more information about the details of the implemented policy).

B. Connectionless Simulation Model

For the analysis of the connectionless scheme, we have adapted the model presented in [7] for a connectionless 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. Depending on the assigned type of class, the packets will be forwarded with different priorities in the routers.

At the entrance of the control path a matrix of input-to-output requests is created. The entries of the matrix are set to the priority number associated with the connection request. The switch is controlled by a contention resolution algorithm (implemented by the *prio_arbiter* in Fig. 3). 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 (see [8] for more details).

IV. Analysis

A flexible test-bench platform is built for analyzing the two communication schemes supporting QoS, connection-oriented and connectionless, under different traffic conditions.

A. Simulation Platform

The simulation platform consists of a parameterized traffic generator and a sink module attached to each router, which generate/collect the packets to be sent/arriving to/from other nodes in the network.

1) Parameterized Traffic Generators: Each traffic generator contains a *timer* module, which defines the traffic injection rate. The statistical distribution followed by the injection rate can be set to any of the injection processes described in Section II: Bernoulli, Markov or Pareto process. The number of packets per message follows a normal distribution with a fixed mean of 10, while the destination node and the type of service are generated following a uniform random distribution.

2) Performance Measurement: A steady-state measurement technique has been applied to analyze the performance of the two communication schemes under study. This technique comprises three main phases: (i) *warm-up* phase to bring the network to equilibrium; (ii) *measurement* phase; and (iii) *chain* phase to allow all packets sent during the *measurement* phase to reach their destinations.

The main goal of the current work is to compare the two communication schemes in terms of message latency and, in particular, the predictability of this performance metric. Message latency is measured as the difference in time since the communication is requested by the source node until the last packet of the corresponding message arrives at the destination node.

B. Results

The network parameters used for the simulations are summarized in Table I. Extensive experiments with different synthetic traffic patterns created by diverse injection processes have been conducted. We believe that the most representative injection processes for embedded applications are the Markov modulated process, suitable for modelling time-varying or correlated traffic processes, and the Pareto process, capable of modelling long-range dependence, which is typical of multimedia applications (such as MPEG-2 video coding/decoding). Therefore, we have generated the injection rate of our experiments following these two distributions.

TABLE I
NoC Parameters

Timing	Slot number/frame = 4 Clock cycles/slot = 4
Topology	Router input number = 5 Router output number = 5 4 × 4 mesh
Routing	Static routing (source routing) X-Y routing
QoS parameters	Connectionless: priority levels = 4 Conn.-oriented: bandwidth levels = 4

1) Message Latency: The results of extensive simulations for the two communication schemes are depicted in Fig. 4 and Fig. 5 by latency histograms per service type. Both systems are stimulated with different set of injection rates generated with Markov and Pareto injection processes (with variable parameters), which lead to different loads of the network (first row is taken as reference unloaded state). Observing and comparing the latency figures, we draw the following conclusions:

- The mean latency for the highest priority level is insensitive to the load in the connectionless scheme, while all the guarantee classes in the connection-oriented approach are unacceptably sensitive to load.
- The mean latency increases, as expected, with decreasing the priority level in the connectionless approach, whereas there is almost no available differentiation between guarantee classes in the connection-oriented scheme.
- The effect of heavy tails in Pareto injection processes is imperceptible for the highest priority level in the connectionless communication, while it unpredictably changes the tendency of the mean latency for all guarantee classes in the connection-oriented solution.

From the previous observations, we can conclude that the connectionless scheme offers a better message latency predictability for different network workloads which mimic the behaviour of an homogeneous multiprocessor system running multiple embedded applications in parallel.

2) Buffer Utilization: The buffer utilization of each scheme is an important parameter, especially regarding the power consumption of the communication architecture. For this reason, two metrics related to the buffer occupancy are measured: (i) the maximum buffer size per buffer and per router, and (ii) the number of accesses to a buffer. The maximum buffer size is a measure of the size of required memory in the routers, while the number of accesses is a measure of the energy cost for performing the packet transfers.

The measurement of buffer sizes is done in terms of number of packets for Markov and Pareto traffic. In the connection-oriented scheme only the BE buffers (see Fig. 2) are considered, since the GT buffers carry data packets on reserved paths of a connection and, consequently, their size never exceeds one. Table II presents the maximum buffer size required among all the buffers of all routers (max) and the average of the max-

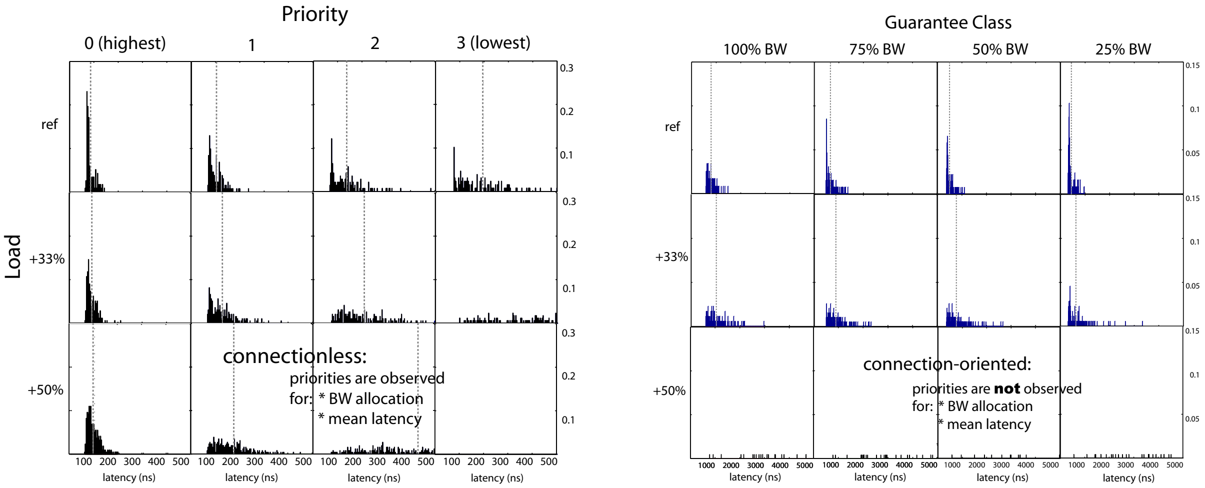

Fig. 4. Normalized message latency histograms for the connectionless and connection-oriented NoCs under Markov traffic. The rows represents different load points of the network. The columns depict the four types of service for each scheme.The service types are shown by packet priorities in connectionless scheme while they are expressed by the percentage of link bandwidth (BW) allocated for a connection in the connection-oriented scheme. The dotted lines show the mean latency of messages for each experiment.

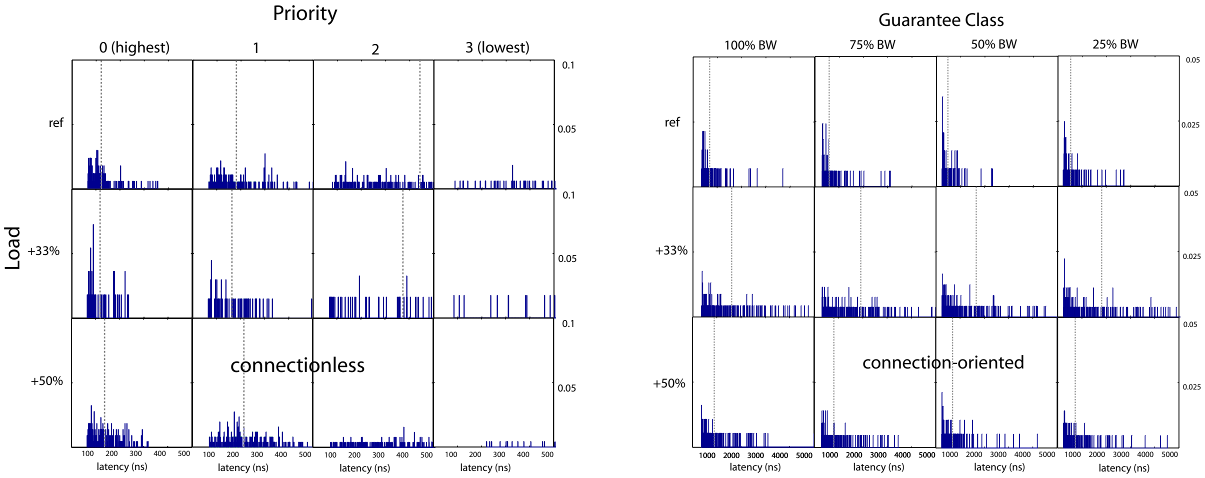

Fig. 5. Normalized message latency histograms for the connectionless and connection-oriented NoCs under Pareto traffic. The rows represents different load points of the network. The columns depict the four types of service for each scheme. The service types are shown by packet priorities in connectionless scheme while they are expressed by the percentage of link bandwidth (BW) allocated for a connection in the connection-oriented scheme. The dotted lines show the mean latency of messages for each experiment.

TABLE II

MAXIMUM BUFFER SIZE RESULTS

		Connectionless	Connection-oriented
Markov	Max	13	6
	Average	6.46	2.68
Pareto	Max	17	7
	Average	9.17	2.83

TABLE III

TOTAL NUMBER OF BUFFER ACCESSES

	Connectionless	Connection-oriented
Markov	8,659,479	22,447,058
Pareto	10,078,051	26,753,147

imum buffer size required by any of these buffers (average) for medium network load (the results obtained for other workloads follow the same trend). The table shows that the required maximum buffer size is not very sensitive to the type of application workload in the connection-oriented scheme, while the connectionless scheme requires larger buffers for long-range dependent traffic (Pareto distribution). Also a generally observed trend is that the maximum buffer size required by the connectionless scheme is 2 to 3 times higher than the one of the connection-oriented approach. Table III summarizes the total number of buffer accesses for the two communication schemes injecting Markov and Pareto traffic. These results demonstrate that the number of buffer accesses in the data path required for the connection-oriented scheme is about 3 times the number of accesses of the connectionless scheme. This is mainly due to the extra accesses to the BE buffers in the connection-oriented solution required by the setup and tear-down packets.

From the previous observations, we can conclude that predictably the connectionless scheme requires larger buffers in the routers. On the other hand, the connection-oriented scheme presents a significant overhead in terms of buffer accesses resulting in higher dynamic energy consumption.

V. Conclusions

The present work contrasts two techniques which dynamically allocate communication resources to improve on-chip predictability in a flexible multiprocessor system. We analyze the effects on message latency and buffer utilization of different embedded-application workloads for both QoS communication schemes. The results demonstrate a more direct applicability of the connectionless scheme for system designers to accurately predict the message latency of parallel flows. Furthermore, this predictability is provided with acceptable buffer sizes compared to the connection-oriented scheme, while the number of buffer accesses is significantly lower, possibly resulting in lower overall energy consumption.

References

[1] S. Murali and G. D. Micheli, “SUNMAP: A tool for automatic topology selection and generation for NoCs,” in *Proceedings of the 41st Design Automation Conference*, San Diego, Calif., June 2004, pp. 914–19.

[2] W. J. Dally and B. Towles, *Principles and Practices of Interconnection Networks*. Morgan Kaufmann, 2004.

[3] J. Duato, S. Yalamanchili, and L. Ni, *Interconnection Networks—An Engineering Approach*. Amsterdam: Morgan Kaufmann, 2003.

[4] S. Santi, B. Lin, L. Kocarev, G. M. Maggio, R. Rovatti, and G. Setti, “On the impact of traffic statistics on quality of service for networks on chip,” in *Proceedings of the IEEE International Symposium on Circuits and Systems*, Kobe, Japan, May 2005.

[5] G. Varatkar and R. Marculescu, “On-chip traffic modeling and synthesis for mpeg-2 video applications,” *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. VLSI-12, no. 1, pp. 108–119, Jan. 2004.

[6] 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.

[7] 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, pp. 37–40.

[8] M. D. Harmanci, N. Pazos Escudero, Y. Leblebici, and P. Ienne, “Quantitative modelling and comparison of communication schemes to guarantee Quality-of-Service in Networks-on-Chip,” in *Proceedings of the IEEE International Symposium on Circuits and Systems*, Kobe, Japan, May 2005, pp. 1782–85.