Analytical Approaches for Performance Evaluation of Networks-on-Chip

Abbas Eslami Kiasari
KTH Royal Institute of Technology, Sweden
kiasari@kth.se

Axel Jantsch
KTH Royal Institute of Technology, Sweden
axel@kth.se

Marco Bekooij
University of Twente, The Netherlands
marco.bekooij@nxp.com

Alan Burns
University of York, United Kingdom
alan.burns@york.ac.uk

Zhonghai Lu
KTH Royal Institute of Technology, Sweden
zhonghai@kth.se

ABSTRACT
This tutorial reviews four popular mathematical formalisms – *dataflow analysis, schedulability analysis, network calculus,* and *queueing theory* – and how they have been applied to the analysis of Network-on-Chip (NoC) performance. We review the basic concepts and results of each formalism and provide examples of how they have been used in on-chip communication performance analysis. The tutorial also discusses the respective strengths and weaknesses of each formalism, their suitability for a specific purpose, and the attempts that have been made to bridge these analytical approaches. Finally, we conclude the tutorial by discussing open research issues.

Categories and Subject Descriptors
C.4 [Performance of Systems]: Design studies, Modeling techniques, Performance attributes

General Terms
Design, Performance

Keywords
System-on-Chip, Network-on-Chip, Design methodology, Performance evaluation, Analytical modeling

1. INTRODUCTION
In modern system-on-chip (SoC), the on-chip communication infrastructure or network-on-chip (NoC) is a dominant factor for design, validation and performance analysis. SoC designers are interested in NoC performance evaluation since their goal is either to provide a minimum level of performance at lowest possible cost, or to provide the highest performance at a given cost. In both cases a reliable measure of performance is indispensible. However, in the first case the focus is typically on *worst-case performance*, while in the latter case the *average-case performance* is the main metric [8]. In real-time systems such as automotive or avionic applications, the worst-case execution time is of particular concern since it is important to know how much time might be needed in the worst-case to guarantee that the task will always finish its jobs before the predetermined deadline. However, the worst-case-based design results in resource over-dimensioning. Therefore, the average-case-based design methods are usually used for non-time critical applications to have a more efficient system.

CASES'12, October 7–12, 2012, Tampere, Finland.
ACM 978-1-4503-1424-4/12/09.

Performance estimation tools can be classified in *simulation models* and *analytical models*. SoC designers have tackled performance analysis by exploring the design space using detailed simulations. Simulation tools are flexible and accurate, but often have to be complemented by an analytical performance modeling approach. In particular, analytical models can analyze the worst-case. An appropriate analytical model can estimate very early in the design phase the desired performance metrics in a fraction of time that simulation would take. Although the use of high-level models conceals a lot of complex technological aspects, it facilitates fast exploration of the NoCs design space. Also, the analytical models provide not only the timing properties of the system, but also useful feedback about the system behavior. Hence, it can be invoked in any optimization loop for NoCs for fast and accurate performance estimations.

2. OVERVIEW OF THE TUTORIAL
This tutorial reviews the applicability and the application of dataflow models, schedulability analysis, network calculus, and queueing theory to NoC performance analysis. The key message of each presentation is described in the following subsections.

2.1 Dataflow Models (Marco Bekooij)
Timed dataflow models have been successfully applied to the derivation of the minimum throughput and maximum latency of network-on-chips [5]. Furthermore, these models are used to compute trade-offs between allocated bandwidth of the network connections and the required capacity of the buffers in the network [11]. Flow-control on the network connections results in cyclic dependencies in these dataflow models. However, such cyclic dependencies do not complicate dataflow analysis significantly. Also, the effects of starvation free arbitration are included in the dataflow models [13].

Dataflow models of networks can be created at different levels of abstraction depending on the required conciseness and accuracy of the model. Conservativeness of these levels of abstraction can be shown by making use of the earlier-the-better refinement relation and its transitivity property [4]. This refinement relation also implies that for proving the temporal requirements of a network, it suffices to show that an admissible schedule exists that adheres to these requirements. Approximation algorithms have been developed that compute these admissible schedules in polynomial time [12]. These algorithms are based on convex programming.

2.2 Schedulability Analysis (Alan Burns)

Scheduling analysis (SA) is a mathematical formalism used to confirm that all deadlines will be met in a real-time system. SA is usually applied to application tasks running on one or more CPUs/cores; but it is a general framework that allows the worst-case behavior of systems to be evaluated. Usually within SA, tasks are repetitive and are either released periodically or sporadically. Tasks can also suffer release jitter.

In this section we introduce a form of SA known as Response-Time Analysis (RTA) for analyzing resources that are scheduled by the common fixed priority dispatching policy. We then show how this analysis can be applied to determine the worst-case latencies for messages on a SoC. The analysis is then used to minimize the number of priority levels (and hence virtual channels) needed to deliver a system in which all messages are delivered by their deadlines. Background on the techniques to be introduced in this section can be found in standard textbooks [1]. The application of RTA to NoC message scheduling is described in [10].

2.3 Network Calculus (Zhonghai Lu)

Network calculus dealing with queuing systems is a formalism for design, analysis and implementation of performance guarantees in communication networks. The research was pioneered by Cruz in his seminal paper [3]. Chang systematically studied this subject [2]. In [6], stochastic network calculus generalizes the deterministic network calculus. Network calculus has been very successful when applied to achieve per-node and end-to-end QoS guarantees in Asynchronous Transfer Mode (ATM) networks, and Internet for both differentiated and integrated services. Recently it is applied to embedded real-time systems, off-chip networks such as wireless sensor networks, and on-chip networks [9].

This tutorial introduces the basics of network calculus within the context of on-chip networks. We begin by introducing the basic concepts such as arrival and service curves of network calculus to uncover the foundation for its elegance. Afterwards, we explain how closed-form formulas for calculating packet delay and backlog bounds can be obtained. With a clear-box approach on an example, we then orient our attention to its application to on-chip networks, analyzing service curves of an on-chip router and a concatenation of routers and further deriving per-flow end-to-end delay bound formula. Finally we give a short summary, reviewing its strength and pointing out future perspectives.

2.4 Queueing Theory (Abbas Eslami Kiasari)

Queueing theory is a branch of probability theory which is concerned with the mathematical modeling and analysis of systems that provide service to stochastic demands. Typically, a queueing model represents a system by probability models of customers' arrival time and service time. Since queueing theory deals with probability models, it is used to compute average-case performance metrics such as average packet latency, average throughput, average energy and power consumption, and average resource utilization.

This section starts with an introduction to queueing theory which demonstrates how to model events and resources in packet-switched networks. Then, we continue the tutorial by briefly reviewing related research where queueing theory is used for performance evaluation and optimization of NoCs. Using a state-of-the-art queueing model [7], we give a numerical example to estimate the average latency of packets in NoCs.

2.5 Bridging the Formalisms (Axel Jantsch)

The general trade-off between abstraction and accuracy can be observed in the comparison between these four formalisms. Since each of the reviewed formalisms has different advantages and difficulties, and since they also partially differ in purpose, none of them can easily replace all others. There are definitely point problems for each formalism that are worthy for further studies, but research on integrated approaches to the problems of system performance analysis is most urgent. Although each formalism can be extended in various directions, these extensions typically run into problems of complex mathematics or they are perceived to be unnatural and cumbersome. Therefore, we believe that comprehensive frameworks that combine two or more formalisms would be most desirable.

3. REFERENCE

[1] A. Burns and A. Wellings. *Real-Time Systems and Programming Languages*. Addison-Wesley, 4th Ed., 2009.

[2] C. S. Chang. *Performance Guarantees in Communication Networks*. Springer-Verlag, 2000.

[3] R. L. Cruz. A calculus for network delay, part I: network elements in isolation; part II: network analysis. *IEEE Transactions on Information Theory*, 37(1):114-141, 1991.

[4] M. Geilen, S. Tripakis, and M. Wiggers. The earlier the better: a theory of timed actor interfaces. In *Proceedings of the ACM International Conference on Hybrid Systems: Computation and Control*, pages 23-32, 2011.

[5] A. Hansson, M. Wiggers, A. Moonen, K. Goossens, and M. Bekooij. Enabling application-level performance guarantees in network-based systems on chip by applying dataflow analysis. *IET Computers & Digital Techniques*, 3(5):398-412, 2009.

[6] Y. Jiang and Y. Liu. *Stochastic Network Calculus*. Springer, 2008.

[7] A. E. Kiasari, Z. Lu, and A. Jantsch. An analytical latency model for networks-on-chip. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, Jan. 2012. doi: 10.1109/TVLSI.2011.2 178620

[8] A. E. Kiasari, A. Jantsch, and Z. Lu. Mathematical formalisms for performance evaluation of networks-on-chip. *Accepted for publication in the ACM Computing Surveys.*

[9] Y. Qian, Z. Lu, and W. Dou. Analysis of worst-case delay bounds for on-chip packet-switching networks. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems,* 29(5):802-815, 2010.

[10] Z. Shi and A. Burns. Real-time communication analysis for on-chip networks with wormhole switching. In *Proceeding of the IEEE International Symposium on Networks-on-Chip (NoCS),* pages 161-170, 2008.

[11] M. Wiggers, M. Bekooij, M. Geilen, and T. Basten. Simultaneous budget and buffer size computation for throughput constraint task graphs. In *Proceedings of the Design, Automation and Test in Europe Conference and Exhibition (DATE)*, pages 1669-1672, 2010.

[12] M. Wiggers, M. Bekooij, and G. Smit. Efficient computation of buffer capacities for cyclo-static dataflow graphs. In *Proceedings of the Design Automation Conference (DAC)*, pages 658-663, 2007.

[13] M. Wiggers, M. Bekooij, and G. Smit. Monotonicity and run-time scheduling. In *Proceedings of the International Conference on Embedded Software*, pages 177-186, 2009.