Is “Network” the Next “Big Idea” in Design?

Radu Marculescu[†], Jan Rabaey[‡] and Alberto Sangiovanni-Vincentelli[‡]

Extended Abstract

[†]Dept. of Electrical and Computer Engineering
Carnegie Mellon University
radum@ece.cmu.edu

[‡]Dept. of Electrical Engineering and Computer Science
University of California, Berkeley
{jan,alberto}@eecs.berkeley.edu

Abstract

As the complexity of nowadays systems continues to grow, we are moving away from creating individual components from scratch, toward methodologies that emphasize composition of re-usable components via the network paradigm. Complex component interactions can create a range of amazing behaviors, some useful, some unwanted, some even dangerous. To manage them, a “science” for network design is evolving, applicable in some surprising areas. In this paper, we consider a few application domains and discus the design challenges involved from a methodology standpoint. From large-scale hardware/software systems, to dynamically adaptive sensor networks, and network-on-chip architectures, these ideas find wide application.

1. Introduction

Any complex system from ambient intelligence to biological systems, from internet to transportation, from utility dispatching to telephony, has behaviors that may be highly (and dangerously) unpredictable due to the interaction of its components. Understanding these behaviors requires a deep analysis and understanding of the topology and pattern of communication among components. Since the early days of networks of workstations (Sun’s motto was then “the computer is the network”), the recognition of the importance of networking has been a key in developing new business models and ways of building reliable systems; we do now live in a deeply networked world!

Recently, the level of understanding of networking concepts needed to design and control complex systems, has reached unprecedented peaks. A holistic approach to the network paradigm is essential. This approach involves understanding the theoretical basis (*e.g.* graph theory, stochastic modeling and analysis), the essential properties (*e.g.* structure, dynamics, communication paradigm), and the metrics (*e.g.* energy, fault-tolerance, robustness) which are relevant to designing and characterizing different networks in either engineered or biological systems.

Starting from these overarching ideas, we aim at addressing the concept of “network” in a variety of contexts, from internet over embedded systems to silicon systems, and identify specific design principles and optimization techniques that are relevant to the design automation community at large. Understanding the structure and behavior of these seemingly different networks is crucial for our ability to master complex behaviors that characterize the newly emergent application domains. For Systems-on-Chip (SoCs), for instance, it has been suggested to replace the global interconnect with on-chip networks which allow communication via packet switching. As such, the *Network-on-Chip* (NoC) becomes the central concept and the optimization process needs to address issues related to the implementation platform, communication complexity, routing strategy, traffic patterns, *etc.*

Going beyond the SoC context, understanding and designing ambient intelligent systems has the network, again, at the forefront of any optimization approach. In this problem space, the *wireless network* is the focal point and various issues related to wireless communication, error rates, synchronization and coordination mechanisms, *etc.* become of central importance. Similarly, it has been recently suggested to treat even the living cells as complex networks too. Inside a living cell, complex molecular interactions are at the very basis of life as we all know it. Amazingly enough, we can learn a lot about biological systems by taking a network-centric approach too. Considering complex interactions between many cellular networks (truly, a network of networks scenario) can make things very complicated and, similarly to SoCs, simply enumerating the individual components (or parts) of the system is not sufficient to understand the underlying complexity of the networked life.

Germane to these ideas, the design automation methods can play an important part in the emerging field of networked applications [1]. Moreover, by arranging a collection of nodes (processing elements or sensors, for instance) in a two-dimensional regular array, or spreading it randomly across a wide geographical area, we open

3-9810801-0-6/DATE06 © 2006 EDAA

many possibilities for disseminating the information among them. Finding out the strategies which work best, the design metrics which are the most relevant, *etc.* require powerful algorithms able to work with incomplete data and answer statistical questions, while still providing meaningful results. Similar to the case of VLSI circuits, the design automation tools are poised to become the true design enablers for such network-based applications. Based on examples from real applications, we believe that many design issues need to be considered with respect to providing performance, scalability, fault-tolerance, robustness and cost-effectiveness. This way, the focus on the network paradigm becomes relevant to the more traditional issues in system-level design.

The paper is organized as follows. Section 2 discusses the interplay between static (*e.g.,* topology) and dynamic (*e.g.,* traffic, communication paradigm) properties of networks. The target application and its impact on network design is considered in Section 3. Finally, our conclusion appears in Section 4.

2. Networks structure and behavior

The science of networks can be traced more than two hundred years ago when the famous mathematician Leonhard Euler solved the Konigsberg bridges problem. By firstly proposing a graph-based solution and realizing that only *topology* (not distance) matters, Euler's formulation to this problem had, in the long run, a greater impact than the solution itself. Indeed, owning to the Euler's graph formulation, the modern science perceives the network as being an abstract graph consisting of nodes, links and a set of rules governing the internode communication. This graph-based view of the network is used by theoreticians to explain the nature of complex social, informational, biological, and technological networks [2].

Why network structure matters? Traditionally, the social and informational networks have been among the first real networks to receive a lot of attention. In recent years, however, significant research efforts have been directed toward understanding the structure of various technological networks (*e.g.* roads and railways, electric power grid, internet). As such, the change from a node-centric to a network-centric perspective became essential to better understand the networked world we live in. For instance, the WWW consists of more than 1 billion nodes but, due to its structure, it is much easier to navigate compared to other networks of equal size arranged in a 2D mesh configuration.

Understanding the structural properties of the network is crucial for mastering the complex behavior of many emergent application domains. For nanotechnologies, for instance, the global interconnect can cause unpredictable delays, propagation and synchronization errors, high power consumption. As such, it has been suggested to replace these long wires with on-chip networks and allow various heterogeneous components residing on the same chip communicate via packet switching. By routing packets instead of wires, the very basis of on-chip communication changes in a fundamental way. Consequently, a new design space opens up with the promise of achieving efficient on-chip communication via the NoC approach.

What are the relevant network properties? Most of the standard network architectures adopted so far in either multiprocessor or SoC domains are only partially scalable; that is, new nodes can be easily added to an already existing network, without distorting the regular structure or reducing the available bandwidth per node. However, since the *diameter* and the average *internode distance* of a $n\times n$ 2D mesh network are proportional to n, navigating these networks becomes easily a major problem as n increases [2]. This is particularly relevant for many technological networks as their *link distribution* per node peaks at very precise scales. Since the structure directly affects the network statistical properties, finding the best way to characterize and customize the network topology is of fundamental importance. For application-specific NoCs, for instance, the detailed understanding of the communication workload can be exploited to provide more performance and better resource utilization via topology customization. However, due to complexity of implementation, lack of standardized interfaces, *etc.*, the widespread use of such fully customized architectures may come at a hefty price.

How about network behavior? Between the regular and fully customized topologies, there exists a large class of networks, referred to as *small world networks,* which are characterized by a surprisingly interesting behavior [2]. Indeed, while the clustering coefficient for most standard topologies, such as meshes and hypercubes, is equal to zero, the small world networks are characterized by high clustering coefficients and small internode distances. (The clustering coefficient measures how tightly the neighbors of any node in the network are connected to each other.) Interestingly enough, the application-specific customization of the network topology can improve the network clustering to better match the application characteristics.

It is worth noting that the structure and behavior are so relevant to various research communities due to the underlying mechanisms which come at play when spreading information (packets, viruses, rumors, *etc.*) among the network nodes. For instance, epidemics proliferation in large populations, randomized protocols for lazy updates in replicated databases, sensor networks, *etc.* have all been studied in the context of the gossip-based multicast protocols for applications that can tolerate a small percentage of message losses, but need to be scalable and have a steady throughput.

3. Application-driven network design

Implementing the target application across a network, requires *mapping* the application tasks to the network nodes. From a design methodology standpoint, it is essential to bring together the theoretical concepts of network design and the application characteristics since, taken together, they can help understand and guide the overall design progress.

Why application matters? Understanding the target application is of fundamental importance for the efficient design of networked systems. For example, the energy consumption is a major design constraint for wireless applications, while it is hardly a design consideration for large scale data networks. Indeed, designing the sensor networks for minimal power consumption can make the network operation more robust, enable nodes operation using energy scavenging, *etc.* As pointed out recently, while the network design has been done traditionally by considering the different layers of the OSI stack in isolation, this cannot result in true energy efficiency so intra- and cross-layer optimizations are a must [4].

At chip-level, application mapping impacts heavily the communication performance of NoCs. Consequently, a set of IPs can be mapped onto a regular NoC architecture, while minimizing the total communication energy and guaranteeing performance through bandwidth reservation. If we relax the requirement for regularity, the network topology plays an important role when mapping the IP cores to the nodes of the network. For example, a constraint-driven communication synthesis approach based on point-to-point communication can result in optimized channels obtained by merging or separating the original point-to-point links. Similarly, one can decompose an entire application using just a few basic communication primitives (*e.g.,* gossiping or all-to-all communication, broadcasting or one-to-many communication, *etc.*) and then replace these primitives by their optimal implementations [3]. This way, the customized topology can achieve energy minimization, for instance, while meeting the performance and wiring constraints. The whole issue is that the topology selected to implement a particular class of applications must be the result of a rigorous analysis process rather than an arbitrary design choice.

What are the appropriate optimization metrics? Referring to the optimization process itself, it is important to note that most mapping algorithms considered so far for on-chip communication are based on average packet hop for either minimizing the communication energy consumption or improving the communication performance; this assumes implicitly that the network is not congested. In practice, however, the network is used in regimes closer to congestion for most on-chip applications. Consequently, the optimization metric should also consider the communication *dynamics* in order to produce meaningful results. Dealing with network dynamics has several important implications. For instance, a new performance model is needed since the average hop distance does not consider the waiting time of a packet, at the input buffers, before the router begins serving it. Also, the input buffers available at each router represent a major consumer of on-chip resources. Depending on the application workload, more buffering resources need to be allocated only to the heavy loaded channels in the network [3].

How about network traffic? Besides queuing effects, the traffic patterns generated by the application itself play a crucial part in network optimization [5]. Indeed, for both data networks and on-chip networks, using the Hurst parameter to characterize the degree of self-similarity helps finding the optimal buffer length distribution; this is a critical issue for designing the routers at each node in the network under multimedia traffic. At the same time, the synthetic trace generation can also benefit from using the appropriate traffic model since this can significantly reduce the simulation time for calculating the buffer loss probability and delay in the network.

4. Conclusion

In this paper we have considered some basic issues relevant to the “science” of network design which emphasizes connecting the right elements, in the right communication pattern, to achieve the right functionality. We have briefly discussed the role of network structure and behavior, as well as application impact in such a communication-based design scenario.

References

[1] K. Keutzer, et al., “System Level Design: Orthogonalization of Concerns and Platform-Based Design,” in IEEE Trans. on Computer-Aided Design of ICs and Systems, Vol.19, Dec. 2000.

[2] M. E. J. Newman, “The structure and function of complex networks,” in SIAM Review 45, 167-256 (2003)

[3] U. Y. Ogras, J. Hu, R. Marculescu, “Key Research Problems in NoC Design: A Holistic Perspective,” in Proc. CODES+ISSS, Jersey City, NJ, September, 2005.

[4] L.C. Zhong, J. Rabaey, A. Wolisz, “An integrated data-link energy model for wireless sensor networks,” in Proc. Intl. Conf. on Communications, Paris, June 2004.

[5] K. Park and W. Willinger, Eds., “Self-Similar Network Traffic and Performance Evaluation,” New York: Wiley, 2000.