Reactive Networked Computation

Vincenzo Liberatore*

Networked systems are built from communicating components that are *reactive*. For example, a server is a continuously running daemon that responds to client requests online [9]. As another example, a TCP source adaptively and dynamically regulates its sending rate depending on the real-time stream of acknowledgment packets [8]. Major objectives of reactive distributed systems are, among others, its scalability, fault-tolerance [9], and real-time performance. However, the textbook definitions of computational complexity is particularly appropriate for modeling *batch* computation. In these models, the objective is to minimize the worst-case asymptotic running time given any one instance. Although batch computation will remain critical in Networked Computation, more work is needed to understand its reactive and adaptive aspects.

Current work in the Algorithms community has mostly focused on *online algorithms*, in which a sequence of decision is made with no knowledge of the future, and on their *competitive analysis*, which is basically worst-case relative performance [2]. The worst-case analysis often leads to trivial results and, in these cases, it is hard to justify (e.g., [6]), so that research has examined restricted adversarial models (e.g., [4]). Networked computing has also relied on *game theory* and *mechanism design* to model the competing demands of heterogeneous players in a distributed environment [7]. An orthogonal approach is *control theory*. For example, control theory has been used for load balancing problems (e.g., [5]). Control-theoretical method have been successfully employed in TCP congestion control, where traffic sources and network delays are modeled as a scalar linear system and a queue management mechanism is viewed as a controller for such a system (e.g., [1, 3]). Control-theoretical approaches benefit from the level of maturity of the field, from its use of real-time system properties, and also because it largely avoids worst-case analysis.

In summary, real-time, reactive, and online processing should be a cornerstone of the Theory of Networked Computation. There are research directions, often in distinct communities (e.g., online algorithms and control theory).

References

[1] Ahmad Al-Hammouri and Vincenzo Liberatore. Decentralized and dynamic bandwidth allocation in networked control systems. In *14th International Workshop on Parallel and Distributed Real-Time Systems (WPDRTS)*, 2006.

[2] Allan Borodin and Ran El-Yaniv. *Online Computation and Competitive Analysis*. Cambridge University Press, 1998.

[3] C.V. Hollot, V. Misra, D. Towsley, and W. Gong. Analysis and design of controllers for AQM routers supporting TCP flows. *IEEE Transactions on Automatic Control*, 47(6):945–959, 2002.

*Division of Computer Science, Case Western Reserve University. 10900 Euclid Avenue, Cleveland, Ohio 44106-7071, USA. Fax: (216) 368 6039. E-mail: vl@case.edu. URL: http://vincenzo.liberatore.org/NetBots/.

[4] Bala Kalyanasundaram and Kirk Pruhs. Speed is as powerful as clairvoyance. *Journal of the ACM*, 47(4):617–643, July 2000.

[5] Magnus Karlsson, Christos Karamanolis, and Xiaoyun Zhu. Triage: Performance isolation and differentiation for storage systems. In *Twelfth IEEE International Workshop on Quality of Service*, pages 67–74, 2004.

[6] Richard M. Karp. On-line algorithms versus off-line algorithms: How much is it worth to know the future? Technical Report TR-92-044, ICSI, July 1992.

[7] C. H. Papadimitriou. Game theory and mathematical economics: a theoretical computer scientist's introduction. In *Proceedings. 42nd IEEE Symposium on Foundations of Computer Science*, pages 4–8, 2001.

[8] Larry L. Peterson and Bruce S. Davie. *Computer Networks.* Morgan Kaufmann, 2000.

[9] Andrew S. Tanenbaum and Maarten Van Steen. *Distributed Systems. Principles and Paradigms.* Prentice-Hall, Upper Saddle River, NJ, 2002.