Journal of High Speed Networks 14 (2005) 1–2
IOS Press

Guest Editorial

Self-Stabilizing Systems, Part 1

In the early part of the 1970's, Edsger W. Dijkstra identified an important property of some distributed systems. He called this property Self-Stabilization. A distributed system is self-stabilizing iff starting in any illegitimate state, the system is guaranteed to converge to a legitimate state in a finite, and hopefully small, number of steps. Moreover, once the system is in a legitimate state, the system is guaranteed to remain within its legitimate states indefinitely, or until some failure pushes the system outside its legitimate states and into its illegitimate states. If such a failure occurs and the system becomes in an illegitimate state, the property of self-stabilization guarantees that the system will return to its legitimate states in a finite number of steps, and the cycle repeats. Clearly, the property of self-stabilization provides distributed systems with a high degree of fault-tolerance.

In the 1980's and 1990's, it became clear that the property of self-stabilization is essential in making many network protocols, especially those employed in the Internet, fault-tolerant. For example, stateless protocols such as the Internet Protocol (IP) and the User Datagram Protocol (UDP) are self-stabilizing. Also, many routing protocols in the Internet, such as the Routing Information Protocol (RIP) and the Open Shortest Path First (OSPF), are self-stabilizing. The relationship between the self-stabilization of network protocols and the fault-tolerance of these protocols has become the research focus of many researchers of computing science.

In the Fall of 2002, Joffroy Beauquier and Colette Johnen (at the Université Paris-Sud) organized an International Seminar on Self-Stabilization at Luminy, France. The seminar, which was sponsored by the Centre International de Rencontres Mathématiques, was a great success uncovering new research directions concerning the self-stabilization of network protocols and distributed systems. As it happened, the seminar did not issue a proceedings, and the presented papers in the seminar have so far remained unpublished.

To remedy this unfortunate situation, Deepinder Sidhu, the Editor-in-Chief of the *Journal of High Speed Networks*, asked Colette Johnen and me to act as Guest Editors for a special issue of JHSN based on the presented papers in the Luminy seminar. Colette oversaw the reviewing process of these papers and selected only nine papers to be published in JHSN. Five of the accepted papers appear in the current issue of the journal, and the remaining four papers appear in the next issue.

The first paper in the current issue is co-authored by Joffroy Beauquier, Laurence Pilard and Brigitte Rozoy. In this paper, the authors describe how to design an external observer of any synchronous self-stabilizing distributed system. The function of the observer is to detect when the observed system has reached its legitimate states after being pushed, by some failure, into its illegitimate states.

The second paper in the current issue is co-authored by Yu Chen, Ajoy K. Datta and Sébastien Tixeuil. In this paper, the authors present a self-stabilizing version of the Border Gateway Protocol (BGP) used in inter-domain routing in the Internet. This new version of BGP solves the routing instability problems that may arise due to transient failures.

The third paper is co-authored by Jorge A. Cobb, Mohamed G. Gouda and Deepinder Sidhu. In this paper, the authors present a self-stabilizing version of the Hello protocol in the Open Shortest Path First (OSPF) routing protocol. In this new version of the Hello protocol, the hello and dead periods are allowed to change (rather than remain fixed) over time, and become consistent (rather than remain identical) in neighboring routers.

The fourth paper in the current issue is co-authored by I.S.W.B. Prasetya and S.D. Swierstra. In this paper, the authors propose to extend the UNITY logic with new operators that model self-stabilization. These operators suggest a number of useful strategies that can be utilized in designing self-stabilizing systems.

The fifth (and last) paper in the current issue is authored by Kleoni Ioannidou. In this paper, the author presents a new transformation of self-stabilizing systems. The presented transformation can be used to transform any self-stabilizing system from a message passing model to a shared memory model where each processor has bounded memory and where the processors share a finite number of bounded size registers.

We, Colette Johnen and I, hope that readers will find this collection of papers interesting, instructive, and worthy of their while.

Mohamed G. Gouda
Austin, TX, USA