Probabilistic Snap-Stabilizing Algorithms for Local Resource Allocation Problems

Master degree subject

Karine Altisen Stéphane Devismes

Verimag Lab
Contacts: {Karine.Altisen, Stephane.Devismes}@imag.fr

Scientific Context. Modern networks are very large-scale (about 100 000 nodes). Now, the more a network contains nodes, the greater the probability of failures is. Hence, today fault-tolerance is a main concern for distributed algorithm designers. In particular, since several decades, there is a growing interest for self-repairing methods. For example, (deterministic) *self-stabilization* [Dij74] is a versatile property, enabling a distributed algorithm to withstand transient faults in a network. Indeed, a self-stabilizing algorithm, after transient faults (*e.g.*, memory corruptions, message losses, *etc.*) hit and place the network in some arbitrary state, enables the network to recover without external (*e.g.*, human) intervention in finite time.

For example, consider a self-stabilizing circulation of a single token. Such an algorithm can be used to implement mutual exclusion in the network: a node can access to the critical section only if it is the token holder. Now, after a transient fault, the token may be duplicated. Consequently, the exclusive access to the critical section is (temporarily) no more guaranteed. The self-stabilizing property ensures in this case that the network recovers a legitimate configuration containing a single token within finite time.

However, self-stabilization has a major drawback: after faults, there is a convergence phase during which no safety property can be ensured despite no fault occurs in the network.

Recent research has been made to overcome this drawback leading to propose new properties that offer more guarantees, *e.g.*, (deterministic) *Snap-stabilization* [BDPV99]. After a finite number of transient faults, a snap-stabilizing algorithm *immediately* operates correctly, without any external (*e.g.* human) intervention. By contrast, self-stabilization only guarantees that the network *eventually* recovers to a correct behavior.

Snap-stabilization is a powerful technique. But, as for deterministic self-stabilization, many problems have no deterministic snap-stabilizing solution. This is in particular true in anonymous networks, where graph coloring and token passing are known to be impossible to solve under the deterministic (self- or snap-) stabilizing setting.

To cope with these impossibility results, we are currently introducing *probabilistic snap-stabilization*, which basically means that *safety properties* of each task started after the last fault will be (deterministically) guaranteed, while *liveness properties* of the task will be *only* guaranteed with probability 1.

Subject. The subject consists in designing and proving probabilistic snap-stabilizing algorithms for anonymous networks.

We propose to focus on local resource allocation problems, in particular the Local Mutual Exclusion (LME). LME consists of ensuring the access of processes to a resource in such a way that no two *neighboring* processes can access the resource simultaneously.

The aim will be to propose, prove, and analyze a probabilistic snap-stabilizing local mutual exclusion algorithm for anonymous networks.

Precisely, the subject involves:

- A bibliographical study: Understand the computational model, the problems to be solved, the expected properties, ...
- The design of a distributed algorithm.
- The analysis of the proposed distributed algorithm:
  - Proof of its correctness.
  - Analysis of its complexity.

Required Skills. An important background about sequential algorithmic, in particular proof of algorithms, is mandatory. This subject also requires background about distributed systems and probabilistic tools (such as Markov Chains).

Working context. The students will be integrated in the lab VERIMAG[1] in the "synchronous" team. Possible extensions into a PhD thesis.

2012-2013

References

[BDPV99] Alain Bui, Ajoy Kumar Datta, Franck Petit, and Vincent Villain. State-optimal snap-stabilizing pif in tree networks. In Anish Arora, editor, *WSS*, pages 78–85. IEEE Computer Society, 1999.

[Dij74] Edsger W. Dijkstra. Self-stabilizing systems in spite of distributed control. *Commun. ACM*, 17(11):643–644, 1974.

[1] http://www-verimag.imag.fr/