Brief Announcement: On the Possibility and the Impossibility of Message-Driven Self-Stabilizing Failure Detection

Martin Hutle
TU Wien
A-1040 Vienna, Austria, EU
hutle@ecs.tuwien.ac.at

Josef Widder
TU Wien
A-1040 Vienna, Austria, EU
widder@ecs.tuwien.ac.at

Categories and Subject Descriptors: C.2.4 [Computer-Communication Networks]: Distributed Systems; F.1.1 [Computation by Abstract Devices]: Models of Computation — *Relations between Models*; F.2.m [Analysis of Algorithms and Problem Complexity]: Miscellaneous

General Terms: Algorithms, Theory

Keywords: Self-Stabilization, Fault Tolerance, Unreliable Failure Detectors, Timing Models

Results: Distributed computing considers sets of distributed processes that execute algorithms where each execution consists of a sequence of events. In the context of reliable agreement problems, much work focuses on timing constraints of these events; e.g. upper bounds between send and reception events of messages. Another issue is event generation. We distinguish here two kinds of models: *time-driven* and *message-driven*. In time-driven models, events are triggered by clocks or timers. In contrast, when considering message-driven algorithms, after the algorithm was started, all events happen as immediate reaction to a received message while clocks are not employed.

The issues of timing constraints and event generation are orthogonal, however. Consider, e.g., the well known failure detector (FD) based consensus algorithms by Chandra and Toueg [3] which work in an asynchronous model of computation (often referred to as "time-free" model, reflecting the absence of timing bounds). These algorithms must be attributed as time-driven as steps can be taken — independently of the presence or absence of messages in input buffers — just by the passage of time respectively the progress of the program counter. It seems obvious that solutions to the same problem can be achieved with message-driven algorithms if (1) messages are immediately processed upon reception, (2) the FD module triggers the consensus algorithm if new suspicions have been added and (3) the FD implementation itself is message-driven. Most existing FD implementation in the literature are not message-driven as they periodically send messages (e.g. heartbeats). Exceptions [5] show that FDs can be implemented without autonomous event generation (i.e., timers or clocks).

We investigate whether time-driven and message-driven semantics are equivalent regarding expressiveness; in other words whether the same set of problems have solutions in both models. To this end, self-stabilizing implementations of FDs are considered. Such implementation where first discussed by Beauquier and Kekkonen-Moneta [1] who devised time-driven self-stabilizing FD implementations.

By providing an impossibility result for message-driven FD implementations we show in the full paper [4] that there are differences in the expressiveness. More precisely, we show that it is impossible to implement the weakest FD [2] that allows solving consensus, i.e., the eventually strong FD $\diamond\mathcal{S}$ under certain assumptions. These assumptions include: self-stabilization, message-driven semantics, unknown bound on link capacity, bounded memory of the failure detector algorithm, determinism, and timing uncertainty.

By devising two FD implementations we show how to circumvent this impossibility result. The first algorithm copes with an unbounded number of messages but requires unbounded space. The second one works with bounded space but requires knowledge of a known upper bound on the number of messages that may be in transit simultaneously.

Acknowledgments: This work is supported by the bmvit FIT-IT project 808198, and by the FWF project P17757.

1. REFERENCES

[1] J. Beauquier and S. Kekkonen-Moneta. Fault-tolerance and self-stabilization: Impossibility results and solutions using self-stabilizing failure detectors. *International Journal of Systems Science*, 28(11):1177–1187, 1997.

[2] T. D. Chandra, V. Hadzilacos, and S. Toueg. The weakest failure detector for solving consensus. *Journal of the ACM*, 43(4):685–722, June 1996.

[3] T. D. Chandra and S. Toueg. Unreliable failure detectors for reliable distributed systems. *Journal of the ACM*, 43(2):225–267, March 1996.

[4] M. Hutle and J. Widder. On the possibility and the impossibility of message-driven self-stabilizing failure detection. Research Report 34/2004, Technische Universität Wien, Institut für Technische Informatik, July 2004. http://www.ecs.tuwien.ac.at/projects/Theta/papers/.

[5] J. Widder, G. Le Lann, and U. Schmid. Failure detection with booting in partially synchronous systems. In *Proceedings of the 5th European Dependable Computing Conference (EDCC-5)*, volume 3463 of *LNCS*, pages 20–37, Budapest, Hungary, Apr. 2005. Springer Verlag.

PODC'05, July 17–20, 2005, Las Vegas, Nevada, USA.
ACM 1-59593-994-2/05/0007.