Brief Announcement: On the Possibility of Consensus in Asynchronous Systems with Finite Average Response Times

Christof FETZER
Computer Science Department
TU Dresden
D-01262 Dresden, Germany
cf2@inf.tu-dresden.de

Ulrich SCHMID
Institut für Technische Informatik
TU Wien
A-1040 Vienna, Austria
s@ecs.tuwien.ac.at

It has long been known that the consensus problem cannot be solved deterministically in completely asynchronous distributed systems, i.e., systems (1) without assumptions on communication delays and relative speed of processes and (2) without access to real-time clocks. We introduce a new asynchronous system model (we call it the Finite Average model or FA model) that

- does not bound the relative speed of processes or minimum speed of processes,
- does not postulate upper or lower bounds on the messages delivery times,
- does not assume that the system stabilizes, and
- does not assume clocks, failure detectors, or other extensions of the model.

Rather, it assumes an unknown finite average response time and unknown finite maximum speed for incrementing an integer. More precisely, the FA model makes the following non-standard assumptions:

- Incrementing an integer takes some unknown time \( \geq G > 0 \).
- The average response time is finite, i.e., the average time until the acknowledgment of a message sent between two correct processes arrives is finite.

Despite of those weak assumptions, the FA model permits the implementation of an eventually perfect failure detector. Hence, the consensus problem can be solved deterministically in this asynchronous system model.

The intuition why an eventually perfect failure detector can be implemented in the FA model is as follows. Each process can implement a clock with a bounded but unknown speed. To do this, a process iteratively increments an integer variable and uses the value of the variable as its clock value. Because incrementing an integer takes an unknown time \( \geq G > 0 \), the maximum speed of such a clock is bounded. The relative speed of two clocks is however unbounded because there is no bound on how slowly a clock can proceed. This clock is sufficient to provide a “subjective” notion of slow and fast messages. The acknowledgment of a slow message \( m \) arrives after the timeout for \( m \) expired and the acknowledgment of a fast message \( n \) arrives before the timeout for \( n \) expires. The timeout for messages is dynamically adapted according to the classification of earlier messages. However, unlike most other timeout-based failure detectors, a process increases the time-out when it receives a fast message but might decrease the timeout when it receives a slow message. One can show that this strategy can be used to ensure that the failure detector will eventually be perfect.

To guarantee the failure detector is eventually perfect, timeouts might have to grow quite large over time. One can decrease the expected time for termination of protocols without sacrificing the guaranteed termination by fusing our failure detector with an adaptive timeout-based failure detector, i.e., one that adjusts its timeouts according to the current system behavior but which does not generally guarantee termination in the FA model.