Brief Announcement: The Degrading Effect of Forgetting on a Synchronizer

Matthias Függer, Alexander Kössler, Thomas Nowak, Martin Zeiner

Brief Announcement: The Degrading Effect of Forgetting on a Synchronizer*

Matthias Függer[1], Alexander Kößler[1], Thomas Nowak[2], and Martin Zeiner[1]

[1] ECS Group, TU Wien, Vienna, Austria
{fuegger,koe,mzeiner}@ecs.tuwien.ac.at
[2] Laboratoire d'Informatique, École polytechnique, Palaiseau, France
nowak@lix.polytechnique.fr

Abstract. A strategy to increase an algorithm's robustness against internal memory corruption is to let processes actively discard part of their accumulated knowledge during execution. We study how different strategies of forgetting affect the performance of a synchronizer in an environment with probabilistic message loss.

Introduction. Network synchronizers allow to tolerate asynchrony, as well as certain types of failures, while using programs devised for lock-step synchronous execution. We study a retransmission-based variant of the α-synchronizer introduced by Awerbuch [1] as the first in a series of synchronizer algorithms for asynchronous message-passing systems. Its main idea is that each process continuously broadcasts its current round number together with the corresponding application data. A process starts the next round when it has received the messages of its current round from all other processes.

In distributed systems such as low-power wireless sensor networks, one can observe two types of failures: (i) Messages can be dropped or corrupted, and (ii) the processes' internal memory can be corrupted. Failures of type (i) are already dealt with by the synchronizer itself: Since each process continuously retransmits its current round message until it starts its next round, dropping messages may only result in larger times between round switches, but not in incorrect behavior. The occurrence of corrupted messages can be made negligible by using error detection codes, often directly supported by transceiver chips. In this paper we thus assume that a message is either correctly received or dropped. Failures of type (ii) may occur for instance by ionized particle hits in memory cells. The longer the time between the write of a memory cell and its successive read, the higher is the likelihood that the read returns corrupted data. By actively resetting (part of) the nodes' internal memory, one reduces the likelihood of reading corrupted memory content. More "forgetful" strategies correspond to maintaining less internal state, which should increase robustness against failures but decrease performance.

* M. Függer, A. Kößler, and M. Zeiner were supported in part by the Austrian Science Fund (FWF): P21694-N23, S11405-N23.

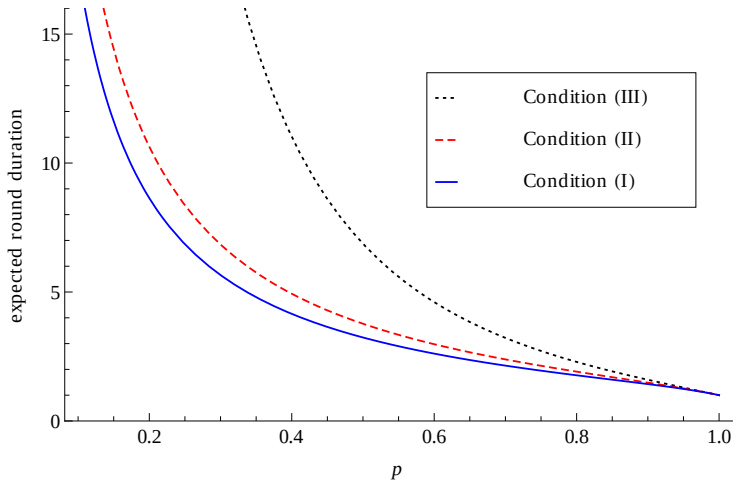

Fig. 1. Expected round durations in systems with three processes

System model and algorithm. Processes $1, \ldots, N$ take steps simultaneously at all integral times, but each message transmission succeeds only with constant independent probability p. Messages that do arrive have a transmission delay of 1. A step consists in (a) receiving messages from other processes, (b) performing local computations, and (c) broadcasting a message to the other processes (i.e., performing $N-1$ point-to-point message transmissions). Processes continuously broadcast their local round number and maintain a *knowledge vector* which contains the information on other processes' local round numbers accumulated via received messages. After updating its local round number, a process may *forget*, i.e., reset its knowledge vector. We consider three different conditions on when processes forget: (I) Never. (II) When starting a new local round. (III) Always, in every step.

Results. We state an explicit formula for the expected asympotic round duration for condition (III) and give efficiently computable bounds for the other two conditions. These bounds are shown to approximate the exact value well if the probability p of successful message transmission is high. We show that for all three conditions, the expected round durations collapse when $p \to 1$: All three expected round durations, as well as their first derivatives as a function of p, coincide in $p = 1$. We prove that for $p \to 0$, the expected round durations for conditions (I) and (II) follow the same order of growth, namely $\Theta(p^{-1})$, whereas condition (III) gives rise to $\Theta(p^{-(N-1)})$. Fig. 1 shows the behavior of the expected round duration in three-processor systems as a parameter of the probability parameter p. Monte Carlo simulations support our analytic results.

References

1. Awerbuch, B.: Complexity of Network Synchronization. J. ACM 32, 804–823 (1985)