CCIT Report #586
May 2006

How to Choose a Timing Model?

Idit Keidar Alexander Shraer
{idish@ee, shralex@tx}.technion.ac.il
Department of Electrical Engineering, Technion, Haifa 32000, Israel

Abstract

When employing a consensus algorithm for state machine replication, should one optimize for the case that all communication links are usually timely, or for fewer timely links? Does optimizing a protocol for better message complexity hamper the time complexity? In this paper, we investigate these types of questions using mathematical analysis as well as experiments over PlanetLab (WAN) and a LAN. We present a new and efficient leader-based consensus protocol that has $O(n)$ stable-state message complexity (in a system with n processes) and requires only $O(n)$ links to be timely at stable times. We compare this protocol with several previously suggested protocols. Our results show that a protocol that requires fewer timely links can achieve better performance, even if it sends fewer messages.

Keywords: synchrony assumptions, eventual synchrony, failure detectors, consensus algorithms, FT Middleware.

1 Introduction

Consensus is an important building block for achieving fault-tolerance using the state-machine paradigm [20]. It is therefore not surprising that the literature is abundant with fault-tolerant protocols for solving this problem. But how does a system designer choose, among the multitude of available protocols, the right one for her system? This decision depends on a number of factors, e.g., time and message complexity, resilience to failures (process crashes, message loss, etc.), and robustness to unpredictable timing delays.

In this paper we focus on the latter, namely the assumptions the protocol makes about timeliness. These are captured in a *timing model*. We study the impact of the choice of timing model on performance in terms of time and message complexity. It is important to note that although the physical system is often given, the system designer has freedom in choosing the timing model representing this system. For example, one seldom comes across a system where the network latency can exceed an hour. This suggests that in principle, even the most unpredictable systems can be modeled as synchronous, with an upper bound of an hour on message latency. Although a round-based synchronous protocol works correctly in this system, it can take an hour to execute a single communication round, and hence may not be the optimal choice. Indeed, measurements show that timely delivery of 100% of the messages is feasible neither in WANs nor under high load in LANs[10, 6, 4]. Instead, systems choose timeouts by which messages *usually* arrive (e.g., 90% or 99% of the time); note that by knowing the typical latency distribution in the system, a designer can fine-tune the timeout to achieve a desired percentage of timely arrivals. One can then employ protocols that ensure safety even when messages arrive late [10, 21, 15]. Such protocols are called indulgent [17].

While indulgent protocols ensure safety regardless of timeliness, they do make some timeliness assumptions in order to ensure progress. Periods during which these assumptions hold are called *stable*. For example, it is possible to require *Eventual Synchrony (ES)* [15, 10], where messages among all pairs of processes are timely in stable periods. Alternatively, one can use weaker majority-based or leader-based models, where only part of the links are required to be timely in stable periods. This defines a tradeoff: whereas weaker models may require more communication rounds for decision, they may also be stable more often (that is, their timeliness requirements will be satisfied more often). A second consideration is message complexity: protocols that send more messages per round may require fewer rounds. Thus, there may also be a tradeoff between the time and message complexities.

In order to provide insights into such tradeoffs, this paper (1) defines a new timing model, (2) introduces a novel time and message efficient algorithm, and (3) presents an evaluation of different consensus algorithms using probabilistic analysis, as well as concrete mea-