A FAIL-AWARE DATAGRAM SERVICE

Christof Fetzer and Flaviu Cristian

University of California, San Diego
Computer Science Department
La Jolla, CA 92093—0114

cfetzer@cs.ucsd.edu, flaviu@cs.ucsd.edu
http://www-cse.ucsd.edu/users/\{cfetzer,flaviu\}

Abstract: In timed asynchronous distributed systems, it is often useful for a process \( p \) to know that another process \( q \) will not use a certain piece of information \( p \) has sent to \( q \) beyond a certain deadline. Since \( p \) learns about the occurrence of the deadline by simply measuring the passage of time on its own local clock, we call this kind of interprocess communication “communication by time”. Knowledge of computed upper bounds on one-way message transmission delays is a necessary prerequisite for this kind of communication.

We introduce a fail-aware datagram service that supports communication by time by delivering all messages whose computed one-way transmission delays are smaller than a given bound as “fast” and all other messages as “slow”. We specify the service and provide an efficient implementation for it.

INTRODUCTION

In synchronous systems, characterized by certain interprocess communication (Cristian, 1996), processes frequently communicate by measuring the passage of time. For example, consider that a correct process \( p \) broadcasts every \( \pi \) time units an “alive”-message and that the transmission delay of a message is at most \( \Delta_{\text{max}} \): when a process \( q \) has been waiting for an alive-message from \( p \) for more than \( \pi + \Delta_{\text{max}} \) time units, \( q \) knows that \( p \) has crashed. Indeed, if \( p \) were alive and had sent its “alive”-message, the certain communication would ensure that \( q \) would have received it.

In timed asynchronous systems (Fetzer and Cristian, 1997b) characterized by uncertain communication, it is not obvious that processes can communicate by measuring the passage of time on their local clocks, since if a process sends a message to another process, the underlying communication system does not guarantee delivery within a bounded time. We show that communication
by time is also possible in timed asynchronous systems despite the uncertain communication.

One important mechanism for communication by time is (time) locking (Fetzer and Cristian, 1997b): a process $p$ guarantees not to change the value of some variable $V$ for at least, say, $lt$ time units after $p$ has sent a message $m$ to another process $q$. Process $q$ knows at the reception of $m$ that $p$ will not change $V$ for $lt - td(m)$ time units, where $td(m)$ is the transmission delay of $m$. To use (time) locking, a process has to be able to determine an upper bound on the one-way transmission delays of messages.

In this paper we describe a fail-aware datagram service that calculates an upper bound for every message it delivers. In particular, the service classifies delivered messages as either “slow” or “fast”. It is a basic distributed communication service that we have successfully used in the design and implementation of several other distributed services such as a clock synchronization service (Fetzer, 1996), a local leader election service (Fetzer and Cristian, 1997c) and a node membership service (Fetzer and Cristian, 1997a). We propose a specification for this service and a protocol that implements it efficiently using local, unsynchronized hardware clocks.

RELATED WORK

Fail-awareness (Fetzer and Cristian, 1996b) is a general method for extending the properties of a fault-tolerant synchronous service by an exception indicator so that the new, extended service becomes implementable in timed asynchronous systems (Cristian and Fetzer, 1997). The idea is that the indicator tells a server and its clients whether a property currently holds or if it might be violated because the system has suffered “too many performance failures”. For a “synchronous” datagram service we can define the following property (T): \textit{the transmission delay of any message delivered by the service is at most $\Delta$ time units}. Since in asynchronous systems there exists no upper bound on the transmission delay of messages, one cannot guarantee that all messages are delivered within $\Delta$ time units. However, we can extend property (T) by an indicator that says if the message was delivered within $\Delta$ time units: \textit{the transmission delay of any message delivered by the service as “fast” is at most $\Delta$ time units}.

The fail-aware datagram service calculates an upper bound $ub(m)$ on the transmission delay of each message $m$ it delivers and classifies $m$ as “slow” if $ub(m) > \Delta$ and as “fast” if $ub(m) \leq \Delta$, where $\Delta$ is an a priori given constant. The protocol we propose is implementable on top of a conventional datagram service like UDP (Postel, 1980). The calculation of the upper bound is based on the idea which underlies probabilistic remote clock reading (Cristian, 1989), namely that the measurement of round-trip message delays allows the calculation of upper bounds on the one-way transmission delays.

The classification of a message as “slow” or “fast” can also be performed with the help of internally synchronized clocks. Even though internal clock synchronization can be achieved in an asynchronous systems by probabilistic