Chapter 64
Failure Detectors of Strong S and Perfect P Classes for Time Synchronous Hierarchical Distributed Systems

Anshul Verma
Banaras Hindu University, India

Mahatim Singh
Banaras Hindu University, India

Kiran Kumar Pattanaik
https://orcid.org/0000-0003-3920-1873
Atal Bihari Vajpayee Indian Institute of Information Technology and Management Gwalior, India

ABSTRACT

Present failure detection algorithms for distributed systems are designed to work in asynchronous or partially synchronous environments on mesh (all-to-all) connected systems and maintain status of every other process. Several real-time systems are hierarchically connected and require working in strict synchronous environments. Use of existing failure detectors for such systems would generate excess computation and communication overhead. The chapter describes two suspicion-based failure detectors of Strong S and Perfect P classes for hierarchical distributed systems working in time synchronous environments. The algorithm of Strong S class is capable of detecting permanent crash failures, omission failures, link failures, and timing failures. Strong completeness and weak accuracy properties of the algorithm are evaluated. The failure detector of Perfect P class is capable of detecting crash failures, crash-recovery failures, omission failures, link failures, and timing failures. Strong completeness and strong accuracy properties of the failure detector are evaluated.

DOI: 10.4018/978-1-7998-5339-8.ch064

INTRODUCTION

In distributed systems failure detectors are used to maintain information about the operational states of other processes. Information provided by a failure detector is assumed unreliable because it can suspect a correct process or not suspect a faulty process. The operational status information of a process provided by two failure detectors at different processes may differ (Cortinas, 2011). In such scenarios *completeness* and *accuracy* are the two properties to assess the reliability of failure detectors. Completeness has been further defined into two variations: strong and weak; while, accuracy has been defined into four variations: strong, weak, eventual strong, and eventual weak (Chandra & Toueg, 1996). The *strong completeness* represents that eventually every process that crashes is permanently suspected by every correct process. Whereas, *strong accuracy* represents that no correct process is suspected by any process. *Weak accuracy* represents that some correct process is never suspected, means some correct processes can be suspected. The failure detectors that satisfy *strong completeness* and *weak accuracy* properties belong to *Strong S* class. However, those satisfy *strong completeness* and *strong accuracy* properties belong to the *Perfect P* class. Similarly, there are eight pairs, each pair forming a new failure detector class (see Table 1) formed by selecting one of the two completeness properties and one of the four accuracy properties.

Failure detectors adopt mainly two methods for status monitoring of other processes: *polling* and *heartbeat*. *Polling* is basically a *query/reply* (or pull) based status monitoring technique (Larrea, Arévalo, & Fernández, 1999; Larrea, Fernández, & Arévalo, 2004). Whereas, in *heartbeat*, every process *q* periodically sends a heartbeat message to all its neighbours processes *p* to inform them that *q* is alive, thus termed as push based. Absence of the heartbeat message implies a fault (Aguilera, Chen, & Toueg, 1997; Soraluze, Cortiñas, Lafuente, Larrea, & Freiling, 2011). Some failure detectors return a list of suspected processes as output fall under *suspicion based* (Chandra & Toueg, 1996), and those return a list of trusted (correct) processes as output fall under *trust based* failure detectors (Chandra, Hadzilacos, & Toueg, 1996).

Table 1. Classification of failure detectors

Completeness	**Accuracy**			
	Strong	**Weak**	**Eventual Strong**	**Eventual Weak**
Strong	*Perfect P*	*Strong S*	*Eventually Perfect* $\Diamond P$	*Eventually Strong* $\Diamond S$
Weak	*Q*	*Weak W*	$\Diamond Q$	*Eventually Weak* $\Diamond W$

(Chandra & Toueg, 1996)

Taxonomy of distributed systems is presented in Figure 1 which is based on the physical arrangement of nodes (*topology aspect*), and events' completion time bound (*time aspect*). In *time aspects* based classification, systems are classified into three categories: synchronous, asynchronous, and partially synchronous, on the basis of two time attributes. First, the time taken for message transmission between two processes, and second the time taken by a processor to execute a task (Cortinas, 2011). Synchronous systems have lower and upper time bound defined for message transmission and task execution (Hadzilacos & Toueg, 1994). Whereas, asynchronous systems do not have time bound for message