ED FD: Improving the Phi Accrual Failure Detector

Naixue Xiong and Xavier Défago

School of Information Science,
Japan Advanced Institute of Science and Technology (JAIST)

April 27, 2007

IS-RR-2007-007

Japan Advanced Institute of Science and Technology (JAIST)

School of Information Science
1-1 Asahidai, Nomi, Ishikawa 923-1292, Japan

http://www.jaist.ac.jp/

ISSN 0918-7553

ED FD: Improving the Phi Accrual Failure Detector

Naixue Xiong, Xavier Défago

School of Information Science
Japan Advanced Institute of Science and Technology
Email: {naixue, defago}@jaist.ac.jp

Abstract

Failure detection is a fundamental issue for supporting dependability in distributed systems, and often is an important performance bottleneck in providing generic service (similar to IP address lookup) in the event of node failure. It is very necessary to find an acceptable, optimized failure detector (FD) before the FD is actually implemented. Ensuring acceptable quality of service (QoS) is made difficult by the relative unpredictability of the network environment. This paper compares QoS of several parametric and adaptive failure detection schemes. Also, we introduce an optimization over the existing methods, called exponential distribution failure detector (ED FD), which significantly improves QoS, especially in the design of real-time mission-critical systems. Extensive experiments have been carried out over several kinds of networks (a cluster group, a wired LAN, a wireless LAN, and a WAN). The experimental results have shown the properties of the adaptive FDs, and demonstrated that the proposed ED FD outperforms the existing FDs in terms of short detection time, low mistake rate and high query accuracy probability.

keywords: Application requirements, Distributed computing, Failure detector, Fault tolerance, Quality of Service

1 Introduction

Fault-tolerant systems are designed to provide reliable and continuous services for distributed systems despite the failures of some of their components [1-5]. As an essential building block for fault-tolerant systems, failure detector (FD) plays a central role in the engineering of such dependable systems. It is very necessary to compare the existing FDs and find an optimized FD that can detect failures in a timely and correct way, before generic FD service can actually be implemented.

The design of dependable FDs is a hard task, mainly because the statistic behavior of communication delays is indefinable. Furthermore, asynchronous (i.e., no bound on the process execution speed or message-passing delay) distributed systems make it impossible to determine precisely whether a remote process has failed or has just been very slow [6].

FDs can be seen as one oracle per process. An oracle provides a list of processes that it currently suspects to have crashed. And the unreliable FD [6] can make mistakes by falsely suspecting correct processes or trusting crashed processes. It is of utmost importance to ensure acceptable quality-of-service (QoS) of unreliable FD, whose parameters are properly tuned for the most desirable QoS to be provided by the upper layers, because the QoS of FD greatly influences the QoS that upper layers may provide. Many fault-tolerant algorithms have been proposed (e.g., [7-10]) based on unreliable FDs.

In order to quantify the QoS of an FD, a set of metrics is proposed by Chen et al. in [11]: how fast it detects actual failures and how well it avoids false detections. In order to improve the QoS of FD, a lot of adaptive FDs have been proposed [12-15], such as Chen FD [11], Bertier FD [12, 16], and the φ FD [13]. Chen et al. in [11] proposed several implementations relying on the probabilistic behavior of the network system. The protocol uses arrival times sampled in the recent past to compute an estimation of the arrival time of the next heartbeat. The timeout is set according to this estimation and a constant safety margin, and it is recomputed for each interval. This technique provides a good estimation for the next arrival time. Furthermore, this paper assumed that the communication history was driven by uncorrelated samples with ergodic stationary behavior, and message delays followed some probabilistic distribution. However, it uses a constant safety margin, because the authors estimate that the model presents probabilistic behavior [12]. Then, Bertier FD [12, 16] provided an optimization of safety margin for Chen FD. It used a different estimation function, which combined Chen's estimation with Jacobson's estimation of the round-trip time (RTT). Bertier FD was primarily designed to be used over wired local area networks (LANs), where messages are seldom lost [13]. The self-tuned FDs proposed in [17] and [18] use the statistics of the previously-observed communication delays to continuously adjust timeout. In other words, they assume a weak past dependence on communication history. These three FDs dynamically predict new timeout values based on observed communication delays to improve the performance of the protocols.

Although the above FDs have important technical breakthroughs, they have obtained little success so far. So far as we know, there are two main reasons [13]: (1) an FD provides an information list of suspects about which processes have crashed. This information list is not always up-to-date or correct (e.g., an FD may falsely suspect a process that is alive). The reason, in practice, is due to the

high unpredictability of message delays, the dynamic and changing topology of the system, and the high probability of message losses. (2) the conventional binary interaction (i.e., trust and suspect) makes it difficult to meet the requirements of several distributed applications running simultaneously. In practice, many classes of distributed applications require the use of different QoS of failure detection to trigger different reactions (e.g., [19-21]). For instance, an application can take precautionary measures when the confidence in a suspicion reaches a given low level, while it takes more drastic actions once the doubt rises above a higher level [3]. However, the traditional output of the FDs (Chen FD [11] and Bertier FD [12, 16]) is of binary nature[1].

To resolve these questions, Hayashibara and Défago et al. [13] developed a φ FD, which assumes that the inter-arrival times follow a normal distribution, and computes a value φ with a scale that changes dynamically to match recent network conditions (i.e., this FD outputs suspicion level on a continuous scale, instead of traditional binary information. This is different from the other FDs). From the statistic analysis of the experimental results (see 4.2), we found the normal distribution is not a reasonable assumption for the approximation of the heartbeat inter-arrival time, especially in large scale distributed networks or unstable networks[2].

Therefore, this paper proposes a novel estimation of the distribution for the inter-arrival time, called the exponential distribution failure detector (ED FD), as an extension of φ FD [13]. Extensive experimental results have demonstrated that the proposed ED FD outperforms the existing adaptive failure detectors.

Briefly speaking, the ED FD works as follows. The protocol uses a sliding window to maintain the most recent samples of the arrival time, similarly to conventional adaptive FDs [3, 11-12]. The distribution of past samples in the sliding window is used as an approximation for the probabilistic distribution of future heartbeat messages. With this information, the suspicion level e_d is computed using a scale that changes dynamically to match recent network conditions. By design, ED FD can adapt well to changing network conditions, and the requirements of any number of concurrently running applications. The experimental results demonstrated that ED FD provides the flexibility required for implementing a truly generic failure detection service.

Therefore, the contribution of this paper is as follows. Firstly, an optimized accural FD, called ED FD, is proposed. Secondly, we have comparatively evaluated our failure detection scheme with existing schemes (Chen FD [11], Bertier FD [12, 16], and φ FD [13]) by extensive experiments in four cases: a cluster group, LAN,

[1] Bertier FD and Chen FD were aimed at other problems, which they both solved admirably well.

[2] Here the unstable networks means the networks have the high unpredictability of message delays, the great dynamic changing topology of system, and the high probability of message losses.

and wide area network (WAN), and wireless network. The experimental results have shown the properties of the different adaptive FDs, and demonstrated that the proposed ED FD outperforms the existing FDs in terms of short detection time, low mistake rate and high query accuracy probability.

The remainder of the paper is organized as follows: In Section 2, the system model and failure detection QoS metrics are introduced. Section 3 introduces several adaptive FDs. In Section 4, we analyze the sample data and present an optimization of φ FD. Section 5 carries out a lot of experiments in different network conditions (cluster group, LAN, wireless, and WAN). In Section 6, we discuss more work related to FD. Finally, we conclude our work and discuss further work in Section 7.

2 System model and basic concepts

2.1 System model

We consider a partially synchronous distributed system consisting of a finite set of processes $\Pi = \{p_1, p_2, p_3, ..., p_n\}$. A process may fail by crashing, here a crashed process does not recover. A process behaves correctly (i.e., according to the specification) until it (possibly) crashes.

We assume the existence of some global time (unbeknownst to processes) denoted by global stabilized time (GST), and that processes always make progress, furthermore, at least $\delta > 0$ time units elapse between consecutive steps (the purpose of the latter is to exclude the case where processes take an infinite number of steps in finite time) [14].

For simplicity and without loss of generality, [13-14] consider a simple system model that consists of only two processes called p and q, which are arbitrarily taken from the large system Π, where process q monitors process p (see Figure 1). p may periodically send a message to q, or is subject to crash. Here the sending period is called the heartbeat interval Δt. Process q suspects process p if it does not receive any heartbeat message from p for a period of time determined by the freshpoint. In the sequel, we consider the same system model.

In Figure 1, d_i is the transmission delay of heartbeat m_i from p to q. For the incoming heartbeat m_j ($1 \leq j \leq i$), q dynamically gives a response based on the new freshpoint FP_j, which is according to network conditions (e.g., the transmission delay d_j). This model describes three cases that may occur. The first one is that heartbeat message m_1 from the sending time σ_1 of process p arrives at q before q's freshpoint FP_1, then q trusts p from the m_1 arrival time (here we assume that p is suspected in the initial case). The second case is that heartbeat m_2 from p arrives at q after q's freshpoint FP_2, then q suspects p from FP_2 until the

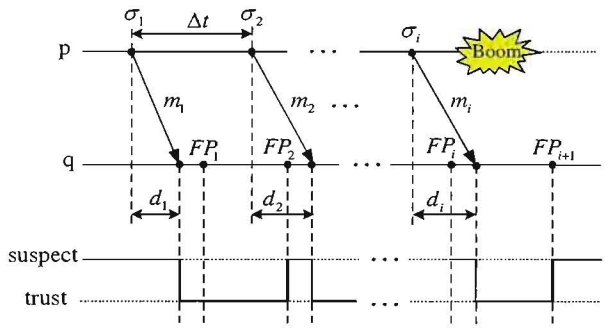


Figure 1: Basic heartbeat failure detection model.

m_2 arrival time. In the third case, after the sending time σ_i, p crashes, then q waits for that heartbeat m_{i+1} until its freshpoint FP_{i+1}, then q starts to suspect p.

In a common belief, the period Δt is a factor that contributes to the detection time. However, Müller [22] shows that, on several different networks, Δt is little determined by QoS requirements, but much by the characteristics of the underlying system, and [13] suggests that there exists, with every network, some nominal range for the parameter Δt with little or no impact on the accuracy of the FD.

In the conventional implementation of this model, the freshpoint is fixed. If the time between two next freshpoints is too short, the likelihood of wrong suspicions is high, though crashes are detected quickly. In contrast, if the time is too long, there is too much detection time, although there are fewer wrong suspicions.

An alternative implementation of this model sets the freshpoints based on the transmission delay of the heartbeat. The advantage is that the maximal detection time is bounded, but the disadvantage is that it relies on physical clocks with negligible drift[3] and a shared knowledge of the heartbeat interval Δt. The drawback is a serious problem in practice, when the regularity of the sending of heartbeats cannot be guaranteed, and the actual sending interval is different from the target one (e.g., timing inaccuracies due to irregular OS scheduling) [13].

The two methods have advantages and disadvantages, it is difficult to conclude which is better [13].

FDs are classified in a number of classes depending on two properties termed

[3]A straightforward implementation of clocks requires synchronized clocks. Chen et al. [11] shows the method to do it with unsynchronized clocks, but this still requires the drift between clocks to be negligible.

the completeness and accuracy properties. Completeness requires that an FD eventually suspects every process that eventually crashes. Accuracy restricts the mistakes that an FD can make. Here our ED FD has the properties of the FDs of class $\Diamond\mathcal{P}$ (eventually perfect), that is sufficient to solve the Consensus problem.

2.2 Failure detection QoS metrics

To evaluate the QoS of the adaptive FDs quantitatively, we use three main QoS metrics (i.e., detection time, mistake rate, and query accuracy probability) that are independent [11]. The first metric measures the impact of the model on the speed of the FD, and the other two metrics measure the impact on its accuracy. In detail, considering two processes p and q where q monitors p, the QoS of the FD at q (called fd_q) can be determined from its transitions between the "trust" and "suspect" states with respect to p (see Figure 2).

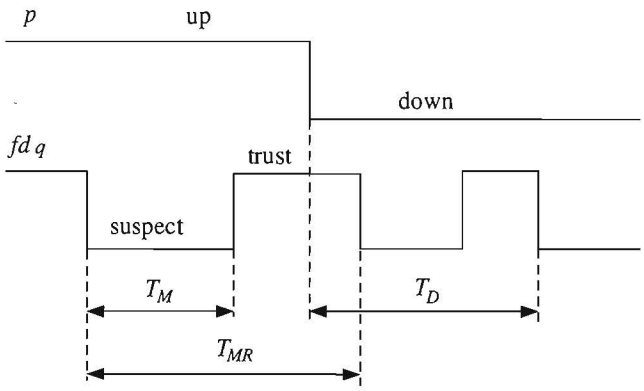


Figure 2: Basic Metrics for the QoS evaluation of an FD [11].

Detection Time (T_D): This is a random variable that represents the length of a period from the time when p starts crashing to the time when q starts suspecting p permanently by fd_q .

Mistake Rate (MR) This is a random variable that represents the number of mistakes that failure detector makes in a unit time, i.e,. it represents how frequent failure detector makes mistakes.

Query Accuracy Probability (QAP) This is a probability that, when queried at a random time, the FD at q indicates correctly that process p is up [11].

Failure Detection QoS Definition Based on [23], a particular FD performance

is defined in terms of its completeness and accuracy properties, and the QoS provided by each of its constituent failure detection modules is a tuple [23]:

$$QoS = (T_D, MR, QAP).$$

The QoS quantifies how fast a detector suspects a failure and how well it avoids false detection.

Relationship of $\Diamond\mathcal{P}$ and $\Diamond\mathcal{P}_{ac}$ Based on [14], an Accrual FD $\Diamond\mathcal{P}_{ac}$ has the two properties: (1) If process p is faulty, then eventually, the suspicion level is monotonously increasing at a positive rate. (2) If process p is correct, the process p always make progress in finite step after some global time, that means, the q eventually receives the heartbeat message from p. Furthermore, Défago et al. [14] proved that an Accrual FD $\Diamond\mathcal{P}_{ac}$ and a binary one of class $\Diamond\mathcal{P}$ have the same computational power, and the two FDs can transform each other.

3 Adaptive failure detectors

Recently, there are many research studies focused on the adaptive FD. The goal of adaptive FDs is to adapt to changing network conditions and application requirements [13]. In general, most adaptive FDs are based on a heartbeat strategy (although nothing precludes a query-response interaction style). Several main adaptive FDs work as follows.

3.1 Chen FD

Chen et al. [11] proposed an approach based on a probabilistic analysis of network traffic. The protocol uses arrival times sampled in the recent past to compute an estimation of the arrival time of the next heartbeat. The timeout is set according to this estimation and a safety margin, and recomputed for each interval.

The algorithm is described as follows: Assume process p sends heartbeat messages periodically to process q (see Figure 1). The n most recent heartbeat messages in a slide window, denoted by m_1, m_2, ..., m_n, are considered by process q. A_1, A_2, ..., A_n are their actual receiving times according to q's local clock. When at least n messages have been received, the theoretical arrival time $EA_{(k+1)}$ can be estimated by:

$$EA_{(k+1)} = \frac{1}{n} \sum_{i=k-n-1}^{k} (A_i - \Delta_i * i) + (k+1)\Delta_i, \qquad (1)$$

where Δ_i is the sending interval. The next timeout delay (which expires at the next freshness point $\tau_{(k+1)}$) is composed of $EA_{(k+1)}$ and the constant safety margin α.

One has

$$\tau_{(k+1)} = \alpha + EA_{(k+1)}. \tag{2}$$

This technique provides an estimation for the next arrival time based on a constant safety margin.

3.2 Bertier FD

Bertier et al. [12, 16] estimated the safety margin dynamically based on Jacobson's estimation of the RTT [23]. Bertier FD adapts the safety margin each time it receives a message. Simply speaking, the adaptation of the margin α is based on the variable $error$ in the last estimation. The recursive algorithm [12, 16] is as follows:

$$error_k = A_k - EA_{(k)} - delay_{(k)}, \tag{3}$$

$$delay_{(k+1)} = delay_{(k)} + \gamma \cdot error_{(k)}, \tag{4}$$

$$var_{(k+1)} = var_{(k)} + \gamma \cdot (|error_{(k)}| - var_{(k)}), \tag{5}$$

$$\alpha_{(k+1)} = \beta \cdot delay_{(k+1)} + \phi \cdot var_{(k)}, \tag{6}$$

and

$$\tau_{(k+1)} = EA_{(k+1)} + \alpha_{(k+1)}, \tag{7}$$

where the parameter γ represents the importance of the new measure with respect to the previous ones, the variable $delay$ represents the estimate margin, and var estimates the magnitude of errors. β and ϕ are used to adjust the variance var. Typical values β, ϕ and γ are 1, 4 and 0.1, respectively. Bertier's estimation provides a short detection time.

3.3 The φ FD

Different from the above schemes, the φ FD [13] outputs a suspicion level on a continuous scale, instead of providing information of a conventional binary nature (trust or suspect). The characteristic and the principal merit of this approach are that it favors a nearly complete decoupling between application requirements and the monitoring of the environment. The φ FD can not only dynamically adapt to current network conditions, but also satisfy different application requirements simultaneously based on the suspicion level expressed.

The specific implementation is as follows: Let T_{last} denote the time when the most recent heartbeat was received; t_{now} is the current time; and $P_{later}(t)$ denotes the probability that a heartbeat will arrive more than t time units after the previous one. Then, the value of φ is calculated as follows:

$$\varphi(t_{now}) = -log_{10}(P_{later}(t_{now} - T_{last})). \tag{8}$$

Here,

$$P_{later}(t) = \frac{1}{\sigma\sqrt{2\pi}} \int_{t}^{+\infty} e^{-\frac{(x-\mu)^2}{2\sigma^2}} dx = 1 - F(t), \tag{9}$$

where $F(t)$ is the cumulative distribution function of a normal distribution with mean μ and variance σ^2. Then, the value of φ at time t_{now} is computed by applying the Equation (8).

When the value of φ is returned to the applications, every application compares the value of ϕ with its threshold Φ. If $\varphi > \Phi$, some action is triggered. Therefore, for different applications with different Φ, different explanations are provided; for the same application, a different value of φ can trigger different actions.

4 Exponential Distribution Failure Detector

In this section, firstly an optimized ED FD over φ FD is presented, then we explain why to do such an optimization. Finally, we give a more precise description on the implementation of ED FD.

The φ FD can output suspicion information on a continuous scale, and can adapt equally well to changing network conditions and the requirements of any number of concurrently running applications. But we found the normal distribution in φ FD (see Figure 3(a)) is not a reasonable assumption for the approximation of the heartbeat inter-arrival, especially, in large scale distributed networks or unstable networks. Thus, this paper develops the optimization over φ FD, called ED FD, to estimate the arrival time of the coming heartbeat.

4.1 ED FD Algorithm

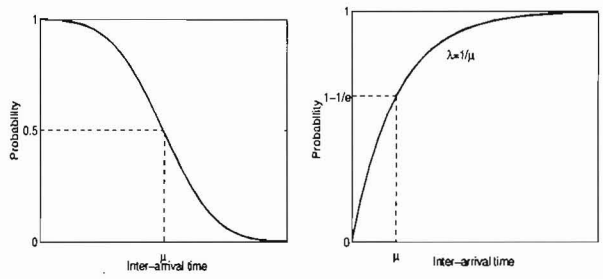
In this subsection, We assume that inter-arrival times follow an exponential distribution (ED) (see Figure 3(b)). In the next subsection, we will explain why this assumption is more reasonable than φ FD.

The ED FD implements the abstraction of an accrual FD in which the suspicion level is given by a value called e_d, expressed on a scale that is dynamically adjusted to reflect current network conditions. Then, the value of suspicion level e_d is calculated as follows.

$$e_d(t_{now}) \stackrel{\text{def}}{=} F(t_{now} - t_{last}), \tag{10}$$

where the $F(t)$ is an exponential distribution function, and one has

$$F(t) = 1 - e^{-\lambda t}, \tag{11}$$



(a) (b)

Figure 3: Probability distribution vs. inter-arrival time: (a) for ϕ FD [13]; (b) for ED FD, and here $\mu = 1/\lambda$.

where $t > 0$, and $\lambda = 1/\mu$ (μ is the average value of the past sampled inter-arrival times).

In this scheme, when $T_D = \mu$, it corresponds to a higher probability $(1 - 1/e)$ than 0.5. And when $T_D > \mu$, the sample data has larger probability than $T_D < \mu$. Then this scheme can catch the most sample data using a high probability, especially, the sample data that has the maximum probability of inter-arrival time. It is very reasonable and important for a good approximation.

From the theory view, the ED FD satisfied the property of accrual failure detector [14], accruement property and upper bound property. Here our ED FD belongs to the class $\Diamond\mathcal{P}_{ac}$, that is sufficient to solve the Consensus problem.

Theorem 1 ED FD implements an FD of class $\Diamond\mathcal{P}_{ac}$, on condition that the system is in accordance with the system model defined in Section 2 (see the proof of Theorem 1 in the Appendix).

4.2 Why ED FD is an optimization over φ FD

This section gives a comparative analysis of ED FD and φ FD based on the statistics of real sample data in several kinds of networks (a cluster group, wired LAN, WAN, and wireless), and shows the probability distribution properties of arrival interval periods. Based on the properties, we analyze the normal distribution in φ FD [13], and then present the improved exponential distribution scheme over φ FD.

4.2.1 Experiment setting and method

For a wired LAN case, the sending host was located at Tsurugi, Ishikawa, Japan, and the receiving host was located at Japan Advanced Institute of Science and

Technology (JAIST), Nomi, Ishikawa, Japan. There were 347,940 samples received (about 1 hour and 6 minutes). We find the average inter-arrival time is 10,782 μs (min.: 5 μs; max.: 488,865 μs). Then from the minimum to the maximum, every 50 μs as a unit, we can get 9,778 sample units, and the last one only has 10 μs time length. We can find the number is different in every statistic unit. So the total sample data divided by the different number in every time unit is a probability for this time unit. The other experimental setting for the wired LAN is shown in sub-Section 5.3.

For the cluster, wireless, and WAN cases, we used the same method to deal with the sample data, except that the WAN case used 100 μs as a statistic unit. Furthermore, the relevant experimental setting and sample data for cluster, wireless and WAN cases are shown in sub-Sections 5.1, 5.2, and 5.4, respectively.

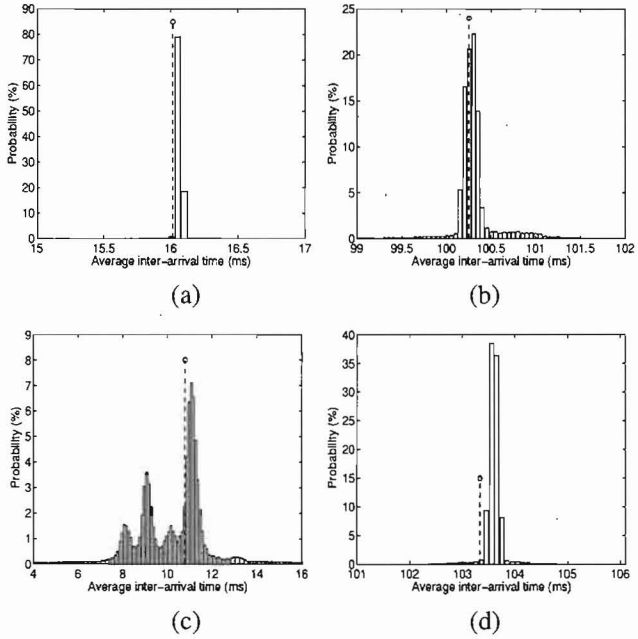


Figure 4: The experimental results of probability distribution vs. detection time : (a) for a cluster group, (b) for wireless, (c) for wired LAN, (d) for WAN.

4.2.2 Statistical experimental results

Based on the above experimental setting and statistic method, we got the experimental results of probability distribution vs. average detection time (see Figure 4). In each sub-figure of Figure 4, the circle point indicates the average value of all inter-arrival times. It is clear that, in general, probability values near the average value of all inter-arrival times are higher than those far from the average value, and most samples have an inter-arrival time that is near the average value. Furthermore, the inter-arrival time with the maximum probability is larger than the mean of all inter-arrival times, except that the inter-arrival time is similar to the mean of all inter-arrival times in wireless case.

4.2.3 The statistical analysis of sample data

There is an assumption that heartbeat inter-arrival times are influenced by a lot of independent unknown factors (central limit theorem) [13]. Chen el al. [11] and Hayashibara et al. [13] presented the estimation of the distribution for inter-arrival times: they follow a normal distribution. And the probability that a given heartbeat will arrive more than t time units later than the previous heartbeat is expressed in Equation (9).

Figure 3(a) shows the probability distribution vs. inter-arrival time for φ FD [13], and μ is the mean of all inter-arrival times. It is clear that the inter-arrival time is μ with 0.5 probability. Furthermore, when inter-arrival time is larger than μ, the probability is less than 0.5. Thus, the sample data whose inter-arrival time is larger than μ have less probability than that whose inter-arrival time is less than μ. So the inter-arrival time with the maximum probability in Figure 4 (inter-arrival time is less than μ) has a less probability in φ FD. It is obvious that the assumption does not present a reasonable probability for the inter-arrival time with the maximum probability, and the sample data whose inter-arrival time is larger than μ should have larger probability. For example, in φ FD, $t = 0$ corresponds to a higher probability than $t = \mu$, it is not reasonable.

In contract, ED FD (see Figure 3(b)) resolves above drawbacks. When inter-arrival time equals μ, the corresponding probability of inter-arrival time is $(1 - 1/e)$, it is much larger than 0.5. And when $t = 0$, the probability is 0, it is reasonable. Thus, ED FD estimation has a little less estimation errors than φ FD estimation.

Furthermore, we found local sample data in a window has similar statistical results as shown in Figure 4, and φ FD used the estimation Equations (8-9) to compute the value φ for the upper applications. Obviously, ED FD is more reasonable than φ FD.

4.3 Implementation of ED FD

This section first describes the architecture of ED FD, then presents the specific implementation algorithm of ED FD.

4.3.1 The architecture of ED FD

Conceptually, the implementation of ED FD on the receiving side q can be decomposed into three basic parts: Monitoring, Interpretation, and Action [13].

In traditional timeout-based FDs (Chen FD [11] and Bertier FD [12, 16]), the monitoring and the interpretation are combined within the FD, and the output is binary. While ED FD, as an accrual FD, provides a lower-level abstraction that avoids the interpretation of monitoring information. Some value, the suspicion level associated with each process, is then left for the applications to interpret [13].

Application processes set a suspicion threshold according to their own QoS requirements: a low threshold generates many wrong suspicions but with a quick detection of an actual crash; Conversely, a high threshold is prone to generate fewer mistakes, but needs more time to detect actual crashes.

4.3.2 The implementation of ED FD

As an accrual FD, the method used in ED FD is quite simple. After warm-up period, when a new heartbeat arrives, the inter-arrival time is put into a sampling slide window, and at the same time, the former oldest one is pushed out of the sampling window. Then the arrival time in the sampling window is used to compute the distribution of inter-arrival times, and get the average inter-arrival time μ in this slide window. After that, based on Equation (10) and Equation (11), we can compute the current value of e_d. At last, applications compare the e_d value and its threshold, then they will carry out some actions, or start to suspect the process. The detail information for the implementation of ED FD is shown in Figure 5.

5 Performance evaluation

In this section, we evaluate and comparatively analyze the performance of ED FD, φ FD [13], Chen FD [11], and Bertier FD [12, 16] in a cluster, a wireless network, a wired local area network (wired LAN) and a wide area network (WAN).

The experiments are carried out with two computers. One (process p) sends messages periodically using UDP, and the other one (process q) receives the messages from process p. In every experiment, the heartbeat sending and arrival times are logged into the log files. These log files are replayed for each FD scheme to

Initialization:
WS: window size;
Δt: sending interval;
$t_{last} = 0$; /*Last arrival time of last heartbeat*/
$Win\_arr[\,] = \perp$; /*empty slide window for inter-arrival times*/
Process p **(Sender)**:
For all $i \geq 1$, at time $(i \cdot \Delta t)$: Send heartbeat HT_i to q;
Process q **(Receiver)**:
Task 1: If q didn't receive message during a certain time period of q's clock

- Increase e_d; /*Suspect p*/

Task 2: Upon receiving heartbeat HT_j from p

- $t_{crt} = clock(\,)$; /*Get the current time*/
- $Win\_arr[\,] = (t_{crt} - t_{last})$;
- $\lambda = WS / \sum_{i=j-WS+1}^{i=j} Win\_arr[i]$;
- $e_d = 1 - e^{-\lambda * (t_{crt} - t_{last})}$;
- $t_{last} = t_{crt}$;

Application k:

- Compare e_d with the threshold E_d^k (from application requirement);
- Carry out some actions or start to suspect p;

Figure 5: Implementation of ED FD.

ensure the fairness of comparison. That means all the FDs are compared with the same experimental conditions: the same network model, the same heartbeat traffic, and the same experiment parameters (sending interval, slide window size, and communication delay, etc.). Thus, it provides an exactly fair experimental platform for every FD.

Interestingly, using the *traceroute* and *ping* commands, we observed that most of the traffic was actually routed with no network breakdowns. And we also used the *ping* command to check the RTT between the sender and the receiver. In addition, we found that the average CPU load was nearly constant during the experiments and was also below the full capacity of the two computers.

In the experiments, each FD scheme used a slide window to save past samples to compute their estimations for the future. All the experiments for the four FDs used the same fixed window size ($WS = 1,000$). Furthermore, it is reasonable to analyze the sampled data only after the slide window is full, because the network is unstable during warmup period.

The main parameters are as follows: In order to find the best QoS and compare with the others, here $E_d \in [10^{-4}, 10]$ for ED FD; For φ FD, the parameters are set the same as in [13]: $\Phi \in [0.5, 16]$; For Chen FD, the parameters are set the same as in [11]: $\alpha \in [0, 10000]$; For Bertier FD, the parameters are set the same as in [12, 16]: $\beta = 1$, $\phi = 4$, $\gamma = 0.1$. In each experiment, the other basic experimental parameters of FDs are the same.

In these experiments, we focus on the following key performance metrics: mistake rate, query accuracy probability and detection time. In every experiment result, different values of mistake rate, query accuracy probability and detection time were obtained with the respective parameters.

5.1 Experiment in a cluster group

This experiment was performed with two cluster nodes, the sending node (monitored) and the receiving node (monitoring) were located in a Cluster Group, at Japan Advanced Institute of Science and Technology (JAIST), in Japan. The two nodes transfer messages through a normal network connection.

5.1.1 Experiment setting: hardware/software/network

The two nodes were equipped with the same hardware and software: an Intel(R) Pentium(R) IV (CPU 2.80 GHz) and 512 Kb of cache size. The operating system was Fedora-Core 4 (Linux). The network is 1Gb/s. The heartbeat messages were generated at a target rate of one heartbeat every 10 ms (the sending interval).

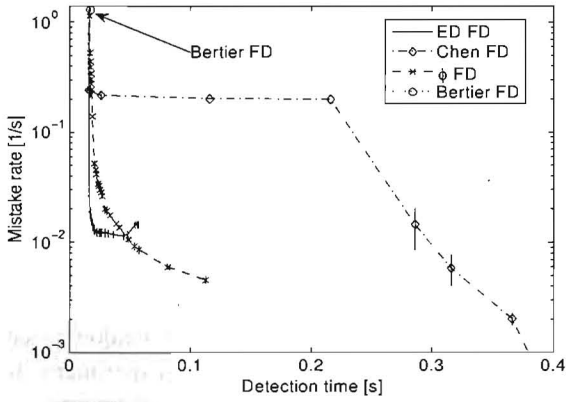


Figure 6: Mistake rate vs. detection time in a cluster.

In order to make the experiment general, we re-did the experiments 5 times with the same code, the same environment, and the same parameters for each FD scheme, but with different experiment times. The experiment periods were about 1 hour 1 minute, 1 hour 4 minutes, 5 hours 3 minutes, 7 hours 5 minutes, and 9 hours 3 minutes.

For the 5 experiments, here we show the detailed experimental data for one typical example (about 1 hour 4 minutes).

Heartbeat sampling We got 229,453 samples, and no heartbeats were lost. The average sending rate actually measured was of one heartbeat every 16.015 ms (standard deviation: 1.709 ms; min.: 0.005 ms; max.: 191.977 ms).

Round-trip time In the experiment, we measured the RTT. The average RTT was 0.387 ms. The standard deviation was 0.756 ms with a minimum of 0.100 ms, and a maximum of 3.101 ms.

5.1.2 Experimental results

The experimental results for detection time, mistake rate and query accuracy probability with a 95% confidence level are shown in Figures 6-7. Figure 6 shows mistake rate comparison of FDs, where the vertical axis is on a logarithmic scale. We believe that the best values are located toward the lower left corner, for that means this FD provides short detection time and has a low mistake rate. Figure 7 shows query accuracy probability comparison of FDs, where the vertical axis is on a linear scale. And the best values are located toward the higher left corner, which means that the FD provides short detection time and has a high query accuracy

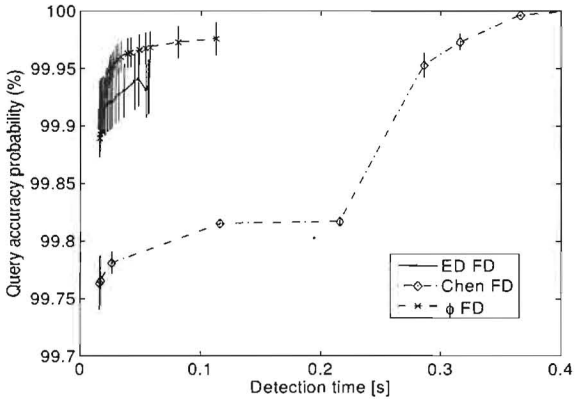


Figure 7: Query accuracy probability vs. detection time in a cluster.

probability.

In Figure 6, when $T_D < 48$ ms, the ED FD can obtain the lowest mistake rates among the four schemes with the same detection time; And when $T_D > 48$ ms, φ FD has the fewest mistakes. In Figure 7, ED FD and φ FD both obtain higher query accuracy probability than Chen FD with some junctions of ED FD and φ FD based on the confidence interval, both of ED FD and φ FD have good query accuracy probability.

Here the behavior of Bertier FD is plotted as a single point, because it has no tuning parameters. And obviously we found that Bertier FD doesn't perform very well compared with the others.

In summary, the ED FD behaves a little better than the other three FDs in the more aggressive range (i.e., $T_D \leq 48$ ms). Chen FD behaves slightly better than the other three FDs in the more conservative range (i.e., $T_D \geq 327$ ms). While, except for those two cases, the φ FD is a little better than others between the aggressive range and the conservative range (i.e., 48 ms $< T_D <$ 327 ms).

5.2 Experiment in a wireless network

These experiments involved two Mac computers and an AirMac Extreme base station. The two Mac computers were located in our lab, in JAIST, Japan. The AirMac extreme base station was used to build a private wireless LAN for two Mac computers.

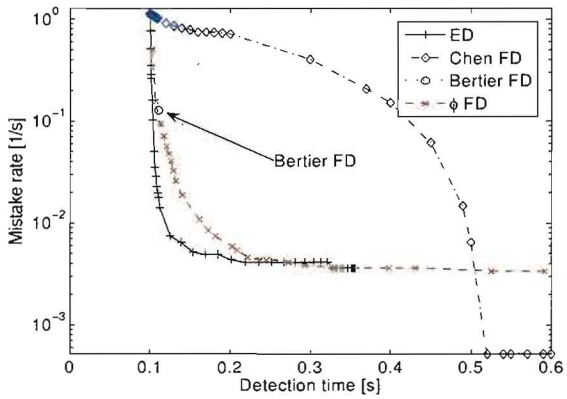


Figure 8: Mistake rate vs. detection time in a wireless network.

5.2.1 Experiment setting: hardware/software/network

The two Mac computers were equipped with the same hardware and software: a PowerPC G4 (CPU 1.5 GHz) and 768 MB DDR SDRAM of Memory. The operating system was Mac OS X (Version 10.4.8). The heartbeat messages were generated at a target rate of one heartbeat every 100 ms (the sending interval).

Heartbeat sampling We got 435, 799 samples, (about 12 hours, 10 minutes and 34 seconds), and no heartbeats were lost. The average sending rate actually measured was of one heartbeat every 100.359 ms (standard deviation: 8.453 ms; min.: 0.0088 ms; max.: 792.418 ms).

Round-trip time By measurement, the average RTT is 2.829 ms. The standard deviation is 12.095 ms with a minimum of 1.598 ms, and a maximum of 231.678 ms.

5.2.2 Experimental results

We show results for detection time, mistake rate and query accuracy probability in Figures 8-9. Figure 8 describes the relationship of mistake rate and detection time among these four FDs on a logarithmic scale. Figure 9 shows the change of the query accuracy probability with different detection time on a linear scale.

In Figure 8, with $T_D < 273$ ms, the ED FD has the lowest mistake rate, and when $T_D > 505$ ms, Chen FD has the fewest mistakes; While, when 273 ms $< T_D < 505$ ms, φ FD obtains the lowest mistake rates, except for some junctions of φ FD and ED FD.

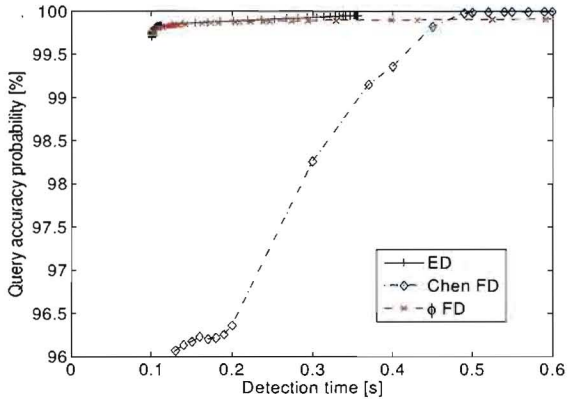


Figure 9: Query accuracy probability vs. detection time in a wireless network.

In Figure 9, for the same detection time, the ED FD has the highest query accuracy probability of all. When $T_D > 470$ ms, Chen FD achieves higher query accuracy probability than φ FD. Except for that case, i.e., when $T_D < 470$ ms, φ FD achieves higher query accuracy probability than Chen FD.

In summary, in this wireless case, ED FD has slightly better performance than the other three FDs in the aggressive range of FD. In the more conservative range (for examples, $T_D > 505$ ms), Chen FD behaves a little better than φ FD. While when 360 ms $< T_D < 470$ ms, φ FD has a little better performance than Chen FD.

5.3 Experiment in a LAN

This experiment also involved two Mac computers in a wired LAN. The sending host (monitored, on Floor 9) and the receiving host (monitoring, on Floor 6) were located at different labs in the same building, in JAIST, Japan.

5.3.1 Experiment setting: hardware/software/network

All the settings of the Mac computers are the same as those in the above wireless experiment. The two computers are connected through a single 100 Mbps Ethernet hub, with no other systems attached.

Heartbeat sampling There were 1,797,026 samples received (10 hours, 15 minutes and 43 seconds), and no heartbeats were lost. The target of heartbeat sending interval is 20 ms. The average sending rate actually measured was of one heartbeat every 20.019 ms (standard deviation: 13.683 ms; min.: 3.099 μs; max.:

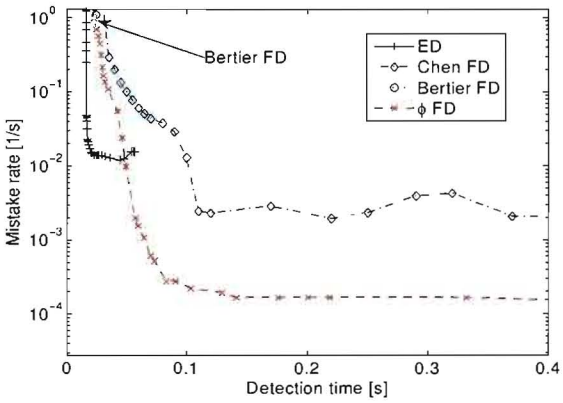


Figure 10: Mistake rate vs. detection time in a LAN.

17,950.169 ms).

Round-trip time The average RTT is 0.917 ms. The standard deviation is 0.146 ms with a minimum of 0.725 ms, and a maximum of 1.678 ms.

5.3.2 Experimental results

The results of the experiment are depicted in Figures 10-11. Figure 10 shows the relationship between detection time and mistake rate for the different FDs. In Figure 10, when $T_D < 49$ ms, ED FD has the lowest mistake rate, and when $T_D > 49$ ms, φ FD has lower mistake rate than other FDs. Figure 11 describes the relationship between detection time and query accuracy probability for each FD. In Figure 11, when $T_D < 54$ ms, ED FD has the highest query accuracy probability, after that period, φ FD first has the highest query accuracy probability with 54 ms $< T_D < 120$ ms, and when $T_D > 120$ ms, Chen FD and φ FD have similar query accuracy probability, with many junctions of φ FD and Chen FD.

In summary for LAN, in the aggressive range: ED FD has a slightly better performance than the other three FDs. In the conservative range, φ FD behaves a little better than the other three FDs.

5.4 Experiment in a WAN

All the above experiment environments are very stable, and there are no heartbeats lost. In order to further compare QoS of these four FDs, we carried out an experiment in WAN.

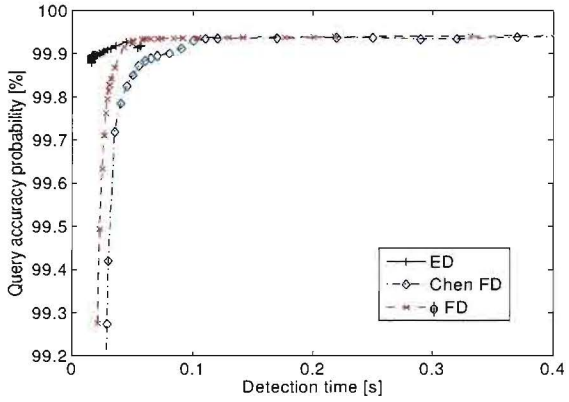


Figure 11: Query accuracy probability vs. detection time in a LAN.

In this experiment, we use the exactly same trace files from the paper about φ FD [13], and these trace files are publicly available on our lab website [25]. Therefore, this provides a common ground for evaluating the performance of ED FD, Chen FD [11], Bertier FD [12, 16] and φ FD [13].

5.4.1 Experiment setting: hardware/software/network

In detail, the trace files and relevant data were gotten from the following experiment setting.

This experiment involves two computers: one was located at the Swiss Federal Institute of Technology in Lausanne (EPFL), in Switzerland. The other one was located in JAIST, Japan. The two computers communicate through a normal inter-continental Internet connection. The two computers have the same equipment and the same operating systems as those in [13]. By analyzing the trace files, we found the average CUP load for the sending host and the receiving host were $1/67$ and $1/22$, respectively. So they were below the full capacity of the computers.

Heartbeat sampling The experiment started on April 3, 2004 at 2:56 UTC, and finished on April 10, 2004 at 3:01 UTC. During the one week experiment period, the heartbeats were generated at a target rate of one heartbeat every 100 ms (the sending interval). The average sending rate actually measured was one heartbeat every 103.501 ms (standard deviation: 0.189 ms; min.: 101.674 ms; max.: 234.341 ms). Furthermore, $5,845,713$ heartbeat messages were sent out, while only $5,822,521$ were received, so message loss rate was about $0.399\,\%$.

By checking the traces files more closely, one found the messages losses were

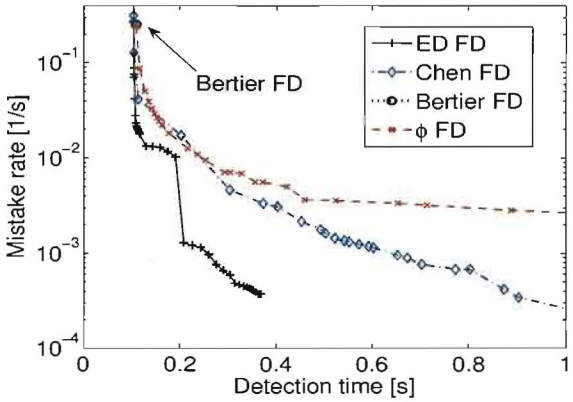


Figure 12: Mistake rate vs. detection time in a WAN.

because of 814 different bursts. The majority of total bursts were short length bursts. While the maximum burst-length was 1,093 heartbeats (only one), it lasted about 2 minutes. Furthermore, most of the heartbeats was not directly between Asia and Europe, but actually, routed through the United States.

Round-trip time The average RTT is 283.338 ms, with a standard deviation of 27.342 ms, a minimum of 270.201 ms, and a maximum of 717.832 ms.

5.4.2 Experimental results

The results of the experiment are depicted in Figures 12-13. Similar to the other experiments, we first give the relationship between detection time and mistake rate, as shown in Figure 12. In Figure 12, with the same detection time, ED FD obtains the lowest mistake rate among the four FDs, except for several initial junctions of ED FD, φ FD and Chen FD. When 148 ms $< T_D <$ 243 ms, φ FD obtains a lower mistake rate than Chen FD; except for that, i.e., when $T_D >$ 243 ms, Chen FD has the lower mistake rate compared with φ FD. The query accuracy probability comparison of FDs in a WAN is shown in Figure 13. From Figure 13, we find: when $T_D <$ 160 ms, ED FD and φ FD have the similar query accuracy probability with the same detection time. While, when $T_D >$ 160 ms, it is clear that ED FD has higher query accuracy probability than φ FD. Furthermore, Chen FD has a little lower query accuracy probability than ED FD and φ FD. In summary, in the aggressive range of FD: ED FD behaves a little better than the other three FDs in terms of short detection time, low mistake rate and high query accuracy probability.

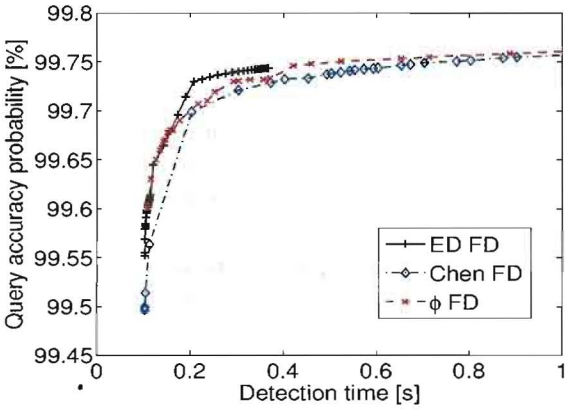


Figure 13: Query accuracy probability vs. detection time in a WAN.

5.5 Comparative analysis of the four FDs

From all the above experimental results, the following remarks can be made: The four kinds of experiments demonstrate that ED FD is an effective improvement over φ FD in terms of short detection time, low mistake rate and high query accuracy probability. In stable cases (no message loss and small variability of delay), such as in a cluster group, wireless system, and wired LAN, ED FD has slightly better performance in the aggressive range of FD. Especially, in unstable network environment (larger variability of heartbeat delay and some message loss), such as WAN, ED FD obviously performs better than the others in aggressive case.

In all, for the applications that need failure detection to be timely and highly accurate, ED FD is an efficient choice.

6 RELATED WORKS

Besides the works cited in Section 1 and Section 3, there have been some other alternate failure detection mechanisms. For example, [26] described a lot of experiments performed on Wide Area Network to assess and fairly compare QoS provided by a large family of FDs. The authors introduced choices for estimators and safety margins used to build several (30) FDs. Compared with [26], this paper considered comparing all kinds of adaptive FD schemes in different experiment environments.

Nunes et al. [27] evaluated the QoS of an FD based on timeout for different combinations of communication delay predictors and safety margins. As the results

show, to improve the QoS, the authors suggested that one must consider the relation between the pair predictor/margin, instead of each one separately. But we think it is (maybe) not very easy to find such proper pairs.

Fabio et al. [28] adapt FDs to communication network load fluctuations using Simple Network Management Protocol (SNMP) and artificial neural networks. The training patterns used to feed the neural network were obtained by using SNMP agents over Management Information Base variables. The output of such neural network is an estimation of the arrival time for the FD to receive the next heartbeat message from a remote process. This approach improves the QoS of the FD, while the training of neural network is a little more complex to achieve the same goal as in this paper.

Fetzer et al. [29] presented an adaptive failure detection protocol. This protocol enjoys the nice property of relying as much as possible on application messages to perform this monitoring. Differently from previous process crash detection protocols, it uses control messages only when no application message is sent by the monitoring process to the observed process. These measurements show that the number of wrong suspicions can be reduced by requiring each process to keep track of the maximum round trip delay between executions.

7 Conclusion

In this paper, we first proposed an optimized ED FD based on exponential distribution over the existing φ FD [13]. And the extensive experiments in cluster group, wireless, wired LAN, and WAN were carried out to compare these four FD schemes (Chen FD [11], Bertier FD [12, 16], φ FD [13], and ED FD). The experiment results have demonstrated that the proposed ED FD outperforms the existing adaptive FDs in terms of short detection time, low mistake rate and high query accuracy probability.

In future work, we would like to further explore FD properties and relation to software engineering applications, and then apply ED FD into an actual fault-tolerant distributed system.

Acknowledgments

We are indebted to Naohiro Hayashibara, Rami Yared and Takuya Katayama for their friendly providing the trace files of WAN, and we also would like to thank many colleagues for their constructive criticism and helpful suggestions for improving the overall quality of this paper.

Research supported by Japanese Government Foundation for Promoting Science and Technology of Ministry of Education, Culture, Sports, Science and Technology (MEXT), and Japan 21st Century COE (Strategic Development of Science and Technology) Foundation: Verifiable and Evolvable e-Society.

References

[1] I. Gupta, T. D. Chandra, G. S. Goldszmidt. On scalable and efficient distributed failure detectors. In *Proc. 20th ACM symp. on Principles of Distributed Computing*, pages 170–179, Newport, Rhode Island, United States, Aug. 26-29, 2001.

[2] M. K. Aguilera, W. Chen, and S. Toueg. Failure detection and consensus in the crash-recovery model. *Distributed Computing*, 13(2): 99–125, Springer-Verlag, 2000.

[3] P. Felber, X. Défago, R. Guerraoui, and P. Oser. Failure detectors as first class objects. In *Proc. 1st Intl. Symp. on Distributed-Objects and Applications (DOA'99)*, pages 132–141, Edinburgh, Scotland, Sept. 1999.

[4] C. Delporte-Gallet, H. Fauconnier, R. Guerraoui. A realistic look at failure detectors. In *Proc. Intl. Conf. on Dependable Systems and Networks (DSN'02)*, pages 345–353, Washington DC, Jun. 2002.

[5] M. J. Fischer, N. A. Lynch, and M. D. Paterson. Impossibility of distributed consensus with one faulty process. *Journal of ACM*, 32(2): 374–382, Apr. 1985.

[6] T. D. Chandra and S. Toueg. Unreliable failure detectors for reliable distributed systems. *Journal of the ACM*, 43(2): 225–267, Mar. 1996.

[7] M. K. Aguilera, W. Chen, and S. Toueg. Using the heartbeat failure detector for quiescent reliable communication and consensus in partitionable networks. *Theoretical Computer Science*, 220(1): 3–30, Jun. 1999.

[8] T. D. Chandra and S. Toueg. Unreliable failure detectors for reliable distributed systems. *Journal of the ACM*, 43(2):225–267, March 1996.

[9] R. Guerraoui, M. Larrea, and A. Schiper. Non-blocking atomic commitment with an unreliable failure detector. *Symposium on Reliable Distributed Systems*, pages 41–50, 1995.

[10] M. Larrea, A. Fernandez, and S. Arevalo. Optimal implementation of the weakest failure detector for solving consensus. In *Proceedings of the 19th Annual ACM Symposium on Principles of Distributed Computing (PODC-00)*, pages 334–334, NY, Jul. 2000.

[11] W. Chen, S. Toueg, and M. K. Aguilera. On the quality of service of failure detectors. *IEEE Trans. on Computers*, 51(5):561–580, May 2002.

[12] M. Bertier, O. Marin, and P. Sens. Implementation and performance evaluation of an adaptable failure detector. In *Proc. IEEE Intl. Conf. on Dependable Systems and Networks (DSN'02)*, pages 354–363, Washington, DC, USA, June 2002.

[13] N. Hayashibara, X. Défago, R. Yared, and T. Katayama. The φ accrual failure detector. In *Proc. 23rd IEEE Intl. Symp. on Reliable Distributed Systems (SRDS'04)*, pages 66–78, Florianpolis, Brazil, Oct. 2004.

[14] X. Défago, P. Urban, N. Hayashibara, T. Katayama. Definition and specification of accrual failure detectors. In *Proc. Intl. Conf. on Dependable Systems and Networks (DSN'05)*, pages 206–215, Yokohama, Japan, Jun. 2005.

[15] A. Basu, B. Charron-Bost, and S. Toueg. Simulating Reliable Links with Unreliable Links in the Presence of Process Crashes. In *Proc. Workshop on Distributed Algorithms (WDAG 1996)*, pages 105–122, Bologna, Italy, 1996.

[16] M. Bertier, O. Martin, P. Sens. Performance analysis of a hierarchical failure detector. In *Proc. Dependable Systems and Networks (DSN'03)*, pages 635–644, San Fra., USA, Jun. 2003.

[17] R. Macêdo. Implementing failure detection through the use of a self-tuned time connectivity indicator. TR, RT008/98, Laboratrio de Sistemas Distribudos - LaSiD, Salvador-Brazil, Aug. 1998.

[18] P. Felber. *The CORBA object group service - a service approach to object groups in CORBA*. PhD thesis, Départment D'informatique, Lausanne, EPFL, Swizerland, 1998.

[19] B. Charron-Bost, X. Défago, and A. Schiper. Broadcasting messages in fault-tolerant distributed systems: the benefit of handling input-triggered and output-triggered suspicions differently. In *Proc. 21st IEEE Intl. Symp. on Reliable Distributed Systems (SRDS'02)*, pages 244–249, Osaka, Japan, Oct. 2002.

[20] X. Défago, A. Schiper, and N. Sergent. Semi-passive replication. In *Proc. 17th IEEE Intl. Symp. Reliable Distributed Systems (SRDS'98)*, pages 43–50, West Lafayette, IN, USA, Oct. 1998.

[21] P. Urbán, I. Shnayderman, and A. Schiper. Comparison of failure detectors and group membership: Performance study of two atomic broadcast algorithms. In *Proc. IEEE Intl. Conf. on Dependable Systems and Networks (DSN'03)*, pages 645–654, San Francisco, CA, USA, June 2003.

[22] M. Müller. Performance evaluation of a failure detector using SNMP. Semester project report, École Polytechnique Fédérale de Lausanne, Lausanne, Switzerland, Feb. 2004.

[23] V. Jacobson. Congestion Avoidance and Control. In *Proc. ACM SIGCOMM '88*, pages 314–329, Stanford, CA, USA, Aug. 1988.

[24] L. M. R. Sampaio, Francisco V. Brasileiro, Walfredo Cirne, Jorge C. A. Figueiredo. How bad are wrong suspicions? toward adaptive distributed protocols. In *Proc. IEEE Intl. Conf. on Dependable Systems and Networks (DSN 2003)*, pages 551–560, San Francisco, CA, USA, Jun. 2003.

[25] *http://ddsg.jaist.ac.jp/en/jst/data.html.*

[26] L. Falai and A. Bondavalli. Experimental evaluation of the QoS of failure detectors on wide area network. In *Proc. of the Int. Conf. on Dependable Systems and Networks (DSN'05)*, Yokohama, Japan, pages 624–633, Jun. 2005.

[27] R. C. Nunes, I. Jansch-Porto. QoS timeout-based self-tuned failure detectors: the effects of the communication delay predictor and the safety margin. In *Proc. 2004 Intl. Conf. on Dependable Systems and Networks (DSN 2004)*, pages 753–761, Florence, Italy, Jun. 2004.

[28] L. Fabio, R. Macêdo. Adapting failure detectors to communication network load fluctuations using SNMP and artificial neural networks. In *Proc. Second Latin-American Symposium on Dependable Computing (LADC 2005)*, pages 191–205, Salvador, Brazil, Oct. 2005.

[29] C. Fetzer, M. Raynal, and F. Tronel. An adaptive failure detection protocol. In *Proc. IEEE the 8th Pacific Rim Symposium on Dependable Computing (PRDC-8)*, pages 146–153, Seoul, Korea, Dec. 2001.

Appendix

Proof of Theorem 1

An FD of class $\Diamond P_{ac}$ must verify the two properties represented by Property 1 and Property 2.

Property 1 (Accruement) *If process p is faulty, then eventually, the suspicion level $sl_{qp}(t)$ is monotonously increasing at a positive rate.*

$$p \in faulty(F) \Longrightarrow \exists K \exists Q \forall k \geq K(sl_{qp}(t_q^{qry}(k)) \leq sl_{qp}(t_q^{qry}(k+1)) \wedge$$
$$sl_{qp}(t_q^{qry}(k)) < sl_{qp}(t^{qry}(k+Q)))$$

Property 2 (Upper bound) *If process p is correct, then the suspicion level $sl_{qp}(t)$ is bounded.*

$$p \in correct(F) \Longrightarrow \exists SL_{max} : \forall t(sl_{qp}(t) \leq SL_{max})$$

The proof of Property 1:

Prove: If process p is faulty, the most recent arrival time of heartbeat t_{last} is constant, at time slot $t_q^{qry}(k)$, the suspicion level e_d is

$$sl_{qp}(t_q^{qry}(k)) = e_d(t_q^{qry}(k)) = F(t_q^{qry}(k) - t_{last}) = 1 - e^{-\lambda(t_q^{qry}(k) - t_{last})}.$$

With time flying, in time slot $t_q^{qry}(k+1)$, the suspicion level is:

$$sl_{qp}(t_q^{qry}(k+1)) = e_d(t_q^{qry}(k+1)) = F(t_q^{qry}(k+1) - t_{last}) = 1 - e^{-\lambda(t_q^{qry}(k+1) - t_{last})}.$$

Since $t_q^{qry}(k) \leq t_q^{qry}(k+1)$, we get

$$-\lambda(t_q^{qry}(k) - t_{last}) \geq -\lambda(t_q^{qry}(k+1) - t_{last}),$$

and

$$e^{-\lambda(t_q^{qry}(k) - t_{last})} \geq e^{-\lambda(t_q^{qry}(k+1) - t_{last})}.$$

Therefore,

$$1 - e^{-\lambda(t_q^{qry}(k) - t_{last})} \leq 1 - e^{-\lambda(t_q^{qry}(k+1) - t_{last})},$$

i.e.,

$$sl_{qp}(t_q^{qry}(k)) \leq sl_{qp}(t_q^{qry}(k+1)).$$

At time slot $t_q^{qry}(k+Q)$, $Q > 0$, $t_q^{qry}(k+Q) > t_q^{qry}(k)$. Using the above same method and conclusion, we can get

$$sl_{qp}(t_q^{qry}(k)) < sl_{qp}(t_q^{qry}(k+Q)).$$

Therefore, ED failure detector satisfied the accruement property.

The proof of Property 2:

If process p is correct, based on the system model, the process p always make progress in finite step after some global time GST, that means, the q eventually receives the heartbeat message from p. That is, there exists t_{max}, when the heartbeat message from p arrives at q. At any arbitrary time t, where $t \leq t_{max}$.

$$sl_{qp}(t_{max}) = e_d(t_{max}) = F(t_{max} - t_{last}) = 1 - e^{-\lambda(t_{max} - t_{last})}.$$

$$sl_{qp}(t) = e_d(t) = F(t - t_{last}) = 1 - e^{-\lambda(t - t_{last})}.$$

Based on Property 1, we know $sl_{qp}(t) \leq sl_{qp}(t_{max}) = SL_{max}$.

From the above proof, we can make the following remarks: The proposed ED FD satisfied the class $\Diamond P_{ac}$ of accrual failure detector. □