ON THE QUALITY OF SERVICE OF
FAILURE DETECTORS

A Dissertation
Presented to the Faculty of the Graduate School
of Cornell University
in Partial Fulfillment of the Requirements for the Degree of
Doctor of Philosophy

by
Wei Chen
May 2000
Failure detectors are basic building blocks of fault-tolerant distributed systems and are used in a wide variety of settings. They are also the basis of a paradigm for solving several fundamental problems in fault-tolerant distributed computing such as consensus, atomic broadcast, leader election, etc.

In this thesis, we study the quality of service (QoS) of failure detectors. By QoS, we mean a specification that quantifies (a) how fast the failure detector detects actual failures, and (b) how well it avoids false detections. To the best of our knowledge, this is the first comprehensive and systematic study of the QoS of failure detectors that provides both a rigorous mathematical foundation and practical solutions.

We first study the QoS specification of failure detectors. In particular, we propose a set of QoS metrics that are especially suited for specifying failure detectors with probabilistic behaviors. We then provide a rigorous mathematical foundation based on stochastic modeling to support our QoS specification.

Next, we develop a new failure detector algorithm for systems with probabilistic behaviors (i.e., the behaviors of message delays and message losses follow some prob-
ability distributions). We perform quantitative analysis and derive closed formulas on the QoS metrics of the new algorithm. We show that among a large class of failure detectors, the new algorithm is optimal with respect to some of the QoS metrics. We then show how to configure the new failure detector algorithm to satisfy QoS requirements given by an application. In order to put the algorithm into practice, we further explain how to modify the algorithm so that it works when the local clocks of processes are not synchronized, and how to configure the failure detector even if the probabilistic behaviors of the system is not known. Finally, we run simulations of both the new algorithm and a simple failure detector algorithm commonly used in practice. The simulation results demonstrate that the new failure detector algorithm provides better QoS than the simple algorithm.
Biographical Sketch

Wei Chen was born on May 2, 1968 in Beijing, China. During most of his first twenty years, he lived with his parents in a lovely neighborhood north to the Long Tan Lake and three bus stops away from the famous Temple of Heaven, by which his wife Jian Han was brought up. In his early age, his mother fostered his interest in mathematics, while his father sent him to a nearby amateur sports school to receive regular soccer training. Since then, mathematics and soccer have been two of his long lasting interests, giving him many joy and excitement.

After six years at No.26 Middle School (later renamed to Hui Wen Middle School during the years when Jian was studying there), where he wrote his first program on an APPLE II computer, he entered Tsinghua University in 1986 and selected Computer Science as his major. He received his Bachelor of Engineering degree in July 1991 with the honor of “Excellent Graduate”, and then continued in Tsinghua for graduate study and received his Master of Engineering degree in March 1993. Only at around this time he finally met Jian, even though they had been brought up in nearby neighborhoods, and had attended the same elementary and middle schools.

After graduation, he worked in the Department of Computer Science and Technology, Tsinghua University as a Teaching and Research Associate. In August 1994,
he came to the States and pursue his Doctoral degree at the Department of Computer Science, Cornell University. One year later, he married Jian, who since then has accompanied and supported him through out his study at Cornell, and in the mean time pursues her own graduate degree in management science.
To my parents, Chen Chengda, Wang Zhengli
and my wife, Jian
Acknowledgements

More than any other person, I am indebted to my wife, Jian. Her love, understanding, and support have made my five years at Cornell much more joyful and much less frustrating than it could have been.

I am extremely grateful to my advisor, Sam Toueg, who has guided me through my research work. Sam has taught me everything important to a high quality academic research, from exploring new ideas, formulating the results, to writing every single sentence of a paper. I cannot imagine how I could have reached this point without his help and guidance.

I have also benefited a lot from the collaboration with Marcos Kawazoe Aguilera. Working with Marcos is always a pleasant and informative experience.

I extend my gratitude to my other committee members Robbert van Renesse, Joseph Halpern, David Shmoys, and Jon Kleinberg (as the proxy of Professor Shmoys at my defense), who carefully reviewed my thesis work and provided helpful input. I am also greatly benefited from the interactions through out the years with many people in or outside the department. Among others, they include Ken Birman, Tushar Chandra, Francis Chu, Vassos Hadzilacos, Narahari U. Prabhu, Michel Raynal, and Jean-Marie Sulmont.
I would like to offer my special thanks to my friend and soccer teammate Fang Xue, who has supplied me with many needed background knowledge in probability theory and stochastic processes. I would also like to thank Thomas Wan and Brian James, who read part of the thesis and helped me to improve my thesis presentation.

I would like to thank my teachers and advisors in Tsinghua, in particular, Professor Dai Yiqi, Lin Xingliang, Lu Kaicheng, Huang Liansheng, and Lu Zhongwan, who introduced me to computer science research.

This research work is partially supported by NSF grants CCR-9402896 and CCR-9711403, and by ARPA/ONR grant N00014-96-1-1014. Any opinions, findings, or recommendations presented in this thesis, however, are my own and do not necessarily reflect the views of any of the organizations mentioned in this paragraph.

Last but not the least, I thank my family, and my wife’s family, for their constant support to my graduate study at Cornell.
# Table of Contents

1 Introduction
   1.1 On the QoS Specification of Failure Detectors .......................... 4
   1.2 The Design and Analysis of a New Failure Detector Algorithm ....... 6
   1.3 Summary of Other Research Works ....................................... 8
      1.3.1 Failure Detection and Consensus in the Crash-Recovery Model 8
      1.3.2 Achieving Quiescence with the Heartbeat Failure Detector ..... 9
   1.4 Thesis Organization ..................................................... 11

2 On the Quality-of-Service Specification of Failure Detectors 12
   2.1 Introduction ............................................................. 12
      2.1.1 Background and Motivation ...................................... 13
      2.1.2 Related Work .................................................... 16
   2.2 Failure Detector Specification ......................................... 18
      2.2.1 The Failure Detector Model ..................................... 18
      2.2.2 Primary Metrics ................................................ 19
      2.2.3 Derived Metrics ................................................ 21
   2.3 Relations between Accuracy Metrics ................................. 25
   2.4 Discussion ............................................................. 28

3 Stochastic Modeling of Failure Detectors and Their Quality-of-
   Service Specifications 31
   3.1 Introduction ............................................................. 31
   3.2 Failure Detector Model ................................................ 32
      3.2.1 Failure Detector Definition ..................................... 33
      3.2.2 Failure Detector Histories as Marked Point Processes .......... 36
      3.2.3 The Steady State Behaviors of Failure Detectors ............... 38
   3.3 Failure Detector Specification Metrics .............................. 46
      3.3.1 Definitions of Metrics .......................................... 46
      3.3.2 Relations between Accuracy Metrics ......................... 50
4 The Design and Analysis of a New Failure Detector Algorithm 60

4.1 Introduction ................................................. 60
  4.1.1 A Common Failure Detection Algorithm and its Drawbacks 61
  4.1.2 The New Algorithm and its QoS Analysis ............... 62
  4.1.3 Related Work ........................................... 65

4.2 The Probabilistic Network Model ........................ 67

4.3 The New Failure Detector Algorithm and Its Analysis .... 68
  4.3.1 The Algorithm ......................................... 68
  4.3.2 The Analysis .......................................... 69
  4.3.3 An Optimality Result .................................. 84
  4.3.4 Configuring the Failure Detector to Satisfy QoS Requirements 88

4.4 Dealing with Unknown System Behavior and Unsynchronized Clocks 92
  4.4.1 Configuring the Failure Detector NFD-S When the Probabilistic Behavior of the Messages is Not Known .... 92
  4.4.2 Dealing with Unsynchronized Clocks .................... 96
  4.4.3 Configuring the Failure Detector When Local Clocks are Not Synchronized and the Probabilistic Behavior of the Messages is Not Known .................. 102

4.5 Simulation Results ......................................... 106
  4.5.1 Simulation Results of NFD-S .......................... 108
  4.5.2 Simulation Results of NFD-E .......................... 112
  4.5.3 Simulation Results of the Simple Algorithm ........... 118

4.6 Concluding Remarks ....................................... 124

A Theory of Marked Point Processes 127

Bibliography 137
List of Figures

2.1 Detection time $T_D$ ................................................. 14
2.2 $FD_1$ and $FD_2$ have the same query accuracy probability of .75, but
the mistake rate of $FD_2$ is four times that of $FD_1$ ................................................. 14
2.3 $FD_1$ and $FD_2$ have the same mistake rate 1/16, but the query accuracy
probabilities of $FD_1$ and $FD_2$ are .75 and .50, respectively .............. 15
2.4 Mistake duration $T_{M}$, Good period duration $T_{G}$, and Mistake recurrence time $T_{MR}$ ................................................. 20
4.1 Three scenarios of the failure detector output in one interval $[\tau_i, \tau_{i+1})$ 68
4.2 The new failure detector algorithm NFD-S, with synchronized clocks,
and with parameters $\eta$ and $\delta$ ................................................. 70
4.3 The new failure detector algorithm NFD-U, with unsynchronized
clocks and known expected arrival times, and with parameters $\eta$ and $\alpha$ 97
4.4 The new failure detector algorithm NFD-E, with unsynchronized
clocks and estimated expected arrival times, and with parameters $\eta$
and $\alpha$ ................................................. 98
4.5 The maximum detection times observed in the simulations of
NFD-S (shown by +) ................................................. 108
4.6 The average mistake recurrence times obtained from the simulations
of NFD-S (shown by +), with the plot of the analytical formula for
$E(T_{MR})$ of NFD-S (shown by −). ................................................. 109
4.7 The 99% confidence intervals for the expected values of mistake recurrence
times of NFD-S (shown by $\uparrow$), with the plot of the analytical
formula for $E(T_{MR})$ of NFD-S (shown by −). ................................................. 113
4.8 The change of the QoS of NFD-E when $n$ increases. Parameter $\alpha = 1.90$ 115
4.9 The maximum detection times observed in the simulations of NFD-E
(shown by $\times$) ................................................. 116
4.10 The average mistake recurrence times obtained from the simulations
of NFD-E (shown by $\times$), with the plot of the analytical formula for
$E(T_{MR})$ of NFD-S (shown by −). ................................................. 117
4.11 The maximum detection times observed in the simulations of SFD-L and SFD-S (shown by • and ○) ........................................... 119
4.12 The average mistake recurrence times obtained from the simulations of SFD-L and SFD-S (shown by -•- and -○-), with the plot of the analytical formula for $E(T_{MR})$ of NFD-S (shown by —). ........................ 120
Chapter 1

Introduction

Fault-tolerant distributed systems are designed to provide reliable and continuous service despite the failures of some of their components. A basic building block in these systems is the failure detector. Failure detectors are used in a wide variety of settings, such as network communication protocols [Bra89], computer cluster management [Pfi98], group membership protocols [ADKM92, BvR93, BDGB94, vRBM96, MMSA+96, Hay98], etc.

Roughly speaking, a failure detector consists of distributed modules such that each process has access to a local failure detector module that provides (possibly erroneous) information about which processes have crashed. This information is typically given in the form of a list of suspects. In general, due to the nondeterminism present in distributed systems, such as message delays and losses caused by network congestion, failure detectors are not reliable: a process that has crashed is not necessarily suspected and a process may be erroneously suspected even though it has not crashed.
Chandra and Toueg [CT96] provide the first formal specification of unreliable failure detectors and show how they can be used to solve some fundamental problems in distributed computing, such as consensus and atomic broadcast. This approach was later used and/or generalized in other works, e.g., [GLS95, DFKM96, FC96, ACT, ACT00, ACT99].

In all of the above works, the failure detector specifications are defined in terms of the eventual behaviors of failure detectors (e.g. a process that crashes is eventually suspected). These specifications are appropriate for purely asynchronous systems in which there is no timing assumption whatsoever. Practical distributed systems, however, usually do have certain timing constraints. In these systems, applications require more than just properties on the eventual behaviors of failure detectors. For example, a failure detector that starts suspecting a process one hour after the process crashes may still satisfy the properties necessary for solving asynchronous consensus, but it can hardly satisfy the requirement of any application in practice. Therefore, in practice, one needs to know the quality of service (QoS) of failure detectors. By QoS, we mean a specification that quantifies the behavior of a failure detector. More precisely, it specifies (a) how fast the failure detector detects actual failures, and (b) how well it avoids false detections.

In this thesis, we focus on the QoS of failure detectors. More specifically:

1. We study how to specify the QoS of failure detectors. In particular:

   (a) We propose a set of QoS metrics that are especially suited for specifying failure detectors with probabilistic behaviors.

   (b) We provide a rigorous mathematical foundation based on stochastic
modeling to support our QoS specification.

2. We develop a new failure detector algorithm, and study the QoS it provides. In particular:

   (a) We perform a quantitative analysis and derive closed formulas on the QoS metrics of the new algorithm.

   (b) We show that among a large class of failure detectors the new algorithm is optimal with respect to some of the QoS metrics.

   (c) We show how to configure the algorithm so that it meets the QoS required by an application. More precisely, given the QoS requirements of an application, we show how to use the closed formulas we derived to compute the parameters of the new algorithm to satisfy the requirements.

   (d) To widen the applicability of the new algorithm, we further explain how to configure the failure detector even if the probabilistic behavior of the system is not known, and how to modify the algorithm so that it works when the local clocks of processes are not synchronized.

   (e) We run simulations of both the new algorithm and a simple algorithm commonly used in practice, and from the simulation results we demonstrate that the new algorithm is better than the simple algorithm with respect to some QoS metrics.

To the best of our knowledge, this is the first comprehensive and systematic study of the QoS of failure detectors that provides both a rigorous mathematical foundation and practical solutions.
1.1 On the QoS Specification of Failure Detectors

How should one specify the QoS of a failure detector? As pointed out above, a failure detector may be slow in detecting a crash, and it may make mistakes, i.e., it may suspect some processes that are actually up. Thus the QoS specification should be given by a set of metrics that describes the failure detector’s speed (how fast it detects crashes) and its accuracy (how well it avoids mistakes). Note that, when specifying the QoS of a failure detector, we should consider the failure detector as a “black box”: the QoS metrics should refer only to the external behavior of the failure detector, and not to various aspects of its internal implementation.

A failure detector’s speed is easy to measure: this is the time elapsed from the moment when a process crashes to the time when the failure detector starts suspecting the process permanently. We call this QoS metric the detection time.

The accuracy metrics should measure how well a failure detector avoids erroneous suspicions of processes that are actually up. Therefore, when measuring the accuracy of failure detectors, we assume that the processes being monitored do not crash. It turns out that determining a good set of accuracy metrics is a subtle task. The subtleties are due to the variety of the accuracy aspects that applications might be interested in. For example, consider an application that at random times queries a failure detector about a process being monitored. For such an application, a natural measure of accuracy is the probability that, when queried at a random time, the failure detector does not suspect the process, i.e., the failure detector output is correct. We call this QoS metric the query accuracy probability. This metric, however, is not sufficient to fully describe the accuracy of a failure detector. In fact,
it is easy to find two failure detectors that have the same query accuracy probability, but one makes mistakes more frequently than the other. In some applications, every mistake of the failure detector causes a costly interrupt, and for such applications the mistake rate is an important accuracy metric. Mistake rate alone, however, cannot fully characterize the accuracy either: one can find two failure detectors that have the same mistake rate but different query accuracy probability. Moreover, even when used together, these two metrics are still not sufficient. It is easy to find two failure detectors such that one is better in both mistake rate and query accuracy probability, but the other is better in some other aspect of the accuracy.

These subtleties show that there are several different aspects of accuracy that may be important to applications, and each aspect has a corresponding accuracy metric. We identify six accuracy metrics, and then use the theory of stochastic processes to determine their relations. Based on these relations, we select two accuracy metrics as the primary ones in the sense that (a) they are not redundant (one cannot be derived from the other), and (b) together, they can be used to derive the other four accuracy metrics. These two accuracy metrics, together with the detection time, provide the QoS specification of failure detectors.

The QoS metrics we proposed are especially suited for specifying failure detectors with probabilistic behaviors (such probabilistic behaviors may be due to the fact that (a) message losses and delays follow a certain probability distribution, or (b) the failure detector algorithm itself uses randomization, as in [vRMH98]). We provide a solid mathematical foundation based on stochastic modeling to formally model the probabilistic behaviors of failure detectors and their QoS. More precisely, we use the theory of marked point processes to formally define the failure detector model and
the QoS metrics proposed, and then we perform a rigorous analysis on the relations between the accuracy metrics under this formal model.

1.2 The Design and Analysis of a New Failure Detector Algorithm

When designing a failure detector algorithm, one should strive to achieve both good speed and good accuracy. However, these are two conflicting objectives. To see this, note that in practice a failure detector typically works as follows: the failure detector waits for messages from the process being monitored, and if it does not receive any message from the process for a while, it starts suspecting the process. This suspicion could be a mistake since the messages from the process may be delayed or lost. If the failure detector waits for a longer period of time before suspecting the process, it reduces the chance of making a mistake, but it increases the detection time if the process actually crashes. Conversely, if the failure detector waits for a shorter period of time before suspecting the process, it reduces the detection time if the process actually crashes, but increases the chance of making a mistake. Thus to design a good algorithm design, one should find the right balance between these two conflicting objectives.

We first examine a simple failure detector algorithm commonly used in practice, and notice that when the variation of the message delays is large, this algorithm cannot achieve both good speed and good accuracy. We then design a new failure detector algorithm that overcomes the problem of the simple algorithm.

We analyze the QoS of the new algorithm in distributed systems with probabilistic
behaviors (i.e., the behaviors of message delays and message losses follow some probability distributions). We use the theory of stochastic processes in the analysis, and derive closed formulas on the QoS metrics of the new algorithm. We then show the following optimality result: Roughly speaking, among all failure detectors that send messages at the same rate and satisfy the same upper bound on the worst-case detection time, the new failure detector algorithm is optimal with respect to the query accuracy probability. This shows that the new failure detector algorithm provides both good speed and good accuracy. We then show that, given a set of QoS requirements by an application, we can use the closed formulas we derived to compute the parameters of the new algorithm to meet these requirements.

Next, we explain how to make the new algorithm applicable to more general settings. This involves the following two modifications: (a) When configuring the new failure detector algorithm to meet an application’s QoS requirements, the original configuration procedure requires the knowledge of the probabilistic behaviors of the system (i.e., the probability distributions of message delays and message losses). We show how to configure the new failure detector even if the probabilistic behavior of the system is not known. (b) The new failure detector algorithm is first given with the assumption that the local clocks of processes are synchronized. We show how to modify the new algorithm so that this assumption is no longer necessary.

Finally, we run simulations of both the new algorithm and the simple algorithm, and provide a detailed analysis on the simulation results. The conclusion we draw from these simulations are: (a) the simulation results of the new algorithm are consistent with our mathematical analysis of the QoS metrics; (b) the new algorithm that does not assume synchronized clocks provides similar QoS as the algorithm that
assumes synchronized clocks; and (c) when comparing the new algorithm with the
simple algorithm under the condition that both algorithms send messages at the
same rate and satisfy the same bound on the worst-case detection time, the new
algorithm provides (in some cases orders of magnitude) better accuracy than the
simple algorithm.

1.3 Summary of Other Research Works

Our research on the QoS of failure detectors aims to provide both a solid foundation
and useful solutions to practical systems. In the same spirit, our other research works
emphasize extending previous theoretical works to more practical computing models.
These works have appeared or will appear as the following journal papers [ACT00,
ACT, ACT99]. We only briefly summarize the main results of these research works
here.

1.3.1 Failure Detection and Consensus in the Crash-Recovery Model

The problem of solving consensus in asynchronous systems with unreliable failure
detectors was first investigated in [CT96, CHT96]. These works established the
paradigm of using failure detection to solve some fundamental problems in fault-
tolerant computing. However, these works only considered systems where process
crashes are permanent and links are reliable (i.e., they do not lose messages). In
practical distributed systems, processes may recover after crashing and links may
lose messages.
In [ACT00], we study the problems of failure detection and consensus in asynchronous systems in which processes may crash and recover, and links may lose messages. We first propose new failure detectors that are particularly suited for the crash-recovery model. We next determine the conditions under which stable storage is necessary to solve consensus in this model. Using the new failure detectors, we give two consensus algorithms that match these conditions: one requires stable storage and the other does not. Both algorithms tolerate link failures and are particularly efficient in the runs that are most likely in practice — those with no failures or failure detector mistakes. In such runs, consensus is achieved within $3\delta$ time units and with $4n$ messages, where $\delta$ is the maximum message delay and $n$ is the number of processes in the system.

1.3.2 Achieving Quiescence with the Heartbeat Failure Detector

An algorithm is quiescent if it eventually stops sending messages. Quiescence is an important property of an algorithm, but in asynchronous systems subject to both process crashes and message losses, quiescence is not easy to achieve.

In [ACT], we study the problem of achieving reliable communication with quiescent algorithms in asynchronous systems with process crashes and lossy links. We first show that it is impossible to solve this problem in purely asynchronous systems (with no failure detectors). We then show that, among failure detectors that output lists of suspects, the weakest one that can be used to solve this problem is $\Diamond P$, a failure detector that cannot be implemented. To overcome this difficulty, we
introduce an implementable failure detector called *Heartbeat* and show that it can be used to achieve quiescent reliable communication. Heartbeat is novel: in contrast to typical failure detectors, it does not output lists of suspects and it is implementable without timeouts. With Heartbeat, many existing algorithms that tolerate only process crashes can be transformed into quiescent algorithms that tolerate both process crashes and message losses. This can be applied to consensus, atomic broadcast, *k*-set agreement, atomic commitment, etc.

In [ACT99], we show how to achieve quiescent reliable communication and quiescent consensus in *partitionable* networks, in which not only processes may crash and messages may be lost, but also the network may be partitioned into disconnected components. We first tackle the problem of reliable communication for partitionable networks by extending the results in [ACT]. In particular, we generalize the specification of the heartbeat failure detector, show how to implement it, and show how to use it to achieve quiescent reliable communication. We then turn our attention to the problem of consensus for partitionable networks. We first show that, even though this problem can be solved using a natural extension of failure detector $\diamondsuit$ (the one used in [CT96] to solve consensus), such solutions are not quiescent — in other words, $\diamondsuit$ alone is not sufficient to achieve quiescent consensus in partitionable networks. We then solve this problem using $\diamondsuit$ and the quiescent reliable communication primitives that we developed.
1.4 Thesis Organization

In Chapter 2, we propose a set of metrics for the QoS specification of failure detectors. In Chapter 3, we present the formalization of the failure detector model and the QoS specification. In Chapter 4, we develop a new failure detector algorithm, analyze its QoS, show its optimality result, show how to configure the algorithm to satisfy QoS requirements given by an application, show how to make the algorithm applicable to more general settings, and show the simulation results that provide an empirical comparison between the new algorithm and the simple algorithm. In Appendix A, we summarize relevant definitions and results in the theory of marked point processes that are used in Chapter 3.
Chapter 2

On the Quality-of-Service Specification of Failure Detectors

2.1 Introduction

In this chapter, we study how to specify the quality of service (QoS) of failure detectors. In particular, we propose a set of QoS metrics that are especially suited for specifying failure detectors with probabilistic behaviors (such probabilistic behaviors may be due to the fact that (a) message losses and delays follow a certain probability distribution, or (b) the failure detector algorithm itself uses randomization, as in [vRMH98]).
2.1.1 Background and Motivation

We consider message-passing distributed systems in which processes may fail by crashing, and messages may be delayed or dropped by communication links. In such systems, failure detectors typically provide a list of processes that are suspected to have crashed so far. A failure detector can be slow, i.e., it may take a long time to suspect a process that has crashed, and it can make mistakes, i.e., it may erroneously suspect some processes that are actually up (such a mistake is not necessarily permanent: the failure detector may later remove this process from its list of suspects). To be useful, a failure detector has to be reasonably fast and accurate.

In this chapter, we propose a set of metrics for the QoS specification of failure detectors. In general, these QoS metrics should be able to describe the failure detector’s speed (how fast it detects crashes) and its accuracy (how well it avoids mistakes). Note that speed is with respect to processes that crash, while accuracy is with respect to processes that do not crash.

A failure detector’s speed is easy to measure: this is simply the time that elapses from the moment when a process \( p \) crashes to the time when the failure detector starts suspecting \( p \) permanently. This QoS metric, called detection time, is illustrated in Fig. 2.1.

How do we measure a failure detector’s accuracy? It turns out that determining a good set of accuracy metrics is a delicate task. To illustrate some of the subtleties involved, consider a system of two processes \( p \) and \( q \) connected by a lossy communication link, and suppose that the failure detector at \( q \) monitors process \( p \). The

---

1We assume that process crashes are permanent, or, equivalently, that a process that recovers from a crash assumes a new identity.
output of the failure detector at $q$ is either “I suspect that $p$ has crashed” or “I trust that $p$ is up”, and it may alternate between these two outputs from time to time. For the purpose of measuring the accuracy of the failure detector at $q$, suppose that $p$ does not crash.

Consider an application that queries $q$’s failure detector at random times. For such an application, a natural measure of accuracy is the probability that, when queried at a random time, the output of the failure detector at $q$ is “I trust that $p$ is up” — which is correct. This QoS metric is the *query accuracy probability*. For example, in Fig. 2.2, the query accuracy probability of $FD_1$ at $q$ is $12/(12 + 4) = .75$.

The query accuracy probability, however, is not sufficient to fully describe the
accuracy of a failure detector. To see this, we show in Fig. 2.2 two failure detectors $FD_1$ and $FD_2$ such that (a) they have the same query accuracy probability, but (b) $FD_2$ makes mistakes more frequently than $FD_1$. In some applications, every mistake causes a costly interrupt, and for such applications the mistake rate is an important accuracy metric.

Note, however, that the mistake rate alone is not sufficient to characterize accuracy: as shown in Fig. 2.3, two failure detectors can have the same mistake rate, but different query accuracy probabilities.

Even when used together, the above two accuracy metrics are still not sufficient. In fact, it is easy to find two failure detectors $FD_1$ and $FD_2$, such that (a) $FD_1$ is better than $FD_2$ in both measures (i.e., it has a higher query accuracy probability and a lower mistake rate), but (b) $FD_2$ is better than $FD_1$ in another respect: specifically, whenever $FD_2$ makes a mistake, it corrects this mistake faster than $FD_1$; in other words, the mistake durations in $FD_2$ are smaller than in $FD_1$. Having small mistake durations may be important to some applications.

---

2The failure detector at $q$ makes a mistake every time its output changes from “trust” to “suspect” while $p$ is actually up.
As it can be seen from the above, there are several different aspects of accuracy that may be important to applications, and each aspect has a corresponding accuracy metric.

In this chapter, we first identify six accuracy metrics (since the behavior of a failure detector is probabilistic, most of these metrics are random variables). We then use the theory of stochastic processes to determine their precise relation. This analysis allows us to select two accuracy metrics as the primary ones in the sense that: (a) they are not redundant (one cannot be derived from the other), and (b) together, they can be used to derive the other four accuracy metrics.

In summary, we show that the QoS specification of failure detectors can be given in terms of three basic metrics, namely, the detection time and the two primary accuracy metrics that we identified. Taken together, these metrics can be used to characterize and compare the QoS of failure detectors.

2.1.2 Related Work

There is not much previous work on the QoS specification of failure detectors.

In [CT96], unreliable failure detectors were introduced as an abstraction that can be used to solve some fundamental problems of fault-tolerant distributed computing, such as consensus, in asynchronous systems. This approach was later used and/or generalized in other works, e.g., [GLS95, DFKM96, FC96, ACT, ACT00, ACT99]. In all of these works, the failure detector specifications are defined in terms of the eventual behaviors of failure detectors (e.g., a process that crashes is eventually suspected). The eventual behavior, however, does not describe the QoS of failure detectors (e.g., how fast a process that crashes becomes suspected).
In [GM98], Gouda and McGuire measure the performance of some failure detector protocols under the assumption that the protocol stops as soon as some process is suspected to have crashed (even if this suspicion is a mistake). This class of failure detectors is less general than the one that we studied here: in our work, a failure detector can alternate between suspicion and trust many times.

In [vRMH98], van Renesse et. al. propose a gossip-style randomized failure detector protocol. They measure the accuracy of this protocol in terms of the probability of premature timeouts.\(^3\) The probability of premature timeouts, however, is not an appropriate metric for the specification of failure detectors in general: it is implementation-specific and it cannot be used to compare failure detectors that use timeouts in different ways. This point is further explained in Section 2.4.

In [VR00], Veríssimo and Raynal study timing failure detectors, which detect timing failures (such as the delay of a message or the execution time of a task being longer than a given time bound). The class of timing failure detectors is more general than the class of failure detectors that detect crash failures. They also study QoS, but their work differs significantly from ours in that: What they study is QoS-FD — failure detectors that detect quality-of-service failures. More precisely, they study failure detectors that output some index (and other derived information) to indicate the quality of service of some system services (e.g. network connectivity). What we study here is the quality of service of (crash) failure detectors, i.e. how good a failure detector is in terms of detecting process crashes, and how to configure the failure detector to satisfy QoS requirements given by an application.

\(^3\)This is called “the probability of mistakes” in [vRMH98].
The rest of the chapter is organized as follows. In Section 2.2, we propose a set of QoS metrics for failure detectors. We quantify the relation between these metrics in Section 2.3, and conclude the chapter with a brief discussion in Section 2.4.

In this chapter, we keep our presentation at an intuitive level. The formal definitions of our model and of our QoS metrics are developed using the theory of stochastic processes, and are given in Chapter 3.

2.2 Failure Detector Specification

We consider a system of two processes $p$ and $q$. We assume that the failure detector at $q$ monitors $p$, and that $q$ does not crash. Henceforth, real time is continuous and ranges from 0 to $\infty$.

2.2.1 The Failure Detector Model

The output of the failure detector at $q$ at time $t$ is either $S$ or $T$, which means that $q$ suspects or trusts $p$ at time $t$, respectively. A transition occurs when the output of the failure detector at $q$ changes: An $S$-transition occurs when the output at $q$ changes from $T$ to $S$; a $T$-transition occurs when the output at $q$ changes from $S$ to $T$. We assume that there are only a finite number of transitions during any finite time interval. A failure detector history describes the output of the failure detector in an entire run.

A failure pattern of process $p$ is just a number $F \in [0, \infty]$, denoting the time $t$ at which $p$ crashes; $F = \infty$ means that $p$ does not crash. A run in which $p$ does not crash is called a failure-free run. For each failure pattern, there is a corresponding
set of possible failure detector histories (as in [CT96]), and this set has a probability distribution. To understand this, consider the following example. Let $F$ be the failure pattern in which $p$ crashes at time 5. The failure detector at $q$ may “detect” this crash at time 6, or 6.74, or 9, etc. This is the set $H$ of failure detector histories corresponding to the failure pattern $F$. In a probabilistic system, some of the failure detector histories in $H$ are more likely than others, and this is given by the probability distribution on $H$. With this probability distribution, quantities like “the probability that the crash of $p$ is detected before time 8”, or “the expected detection time” are now meaningful.

We consider only failure detectors whose behavior eventually reaches steady state, as we now explain.\footnote{We omit the formal definition of steady state here; this definition is based on the theory of stochastic processes.} When a failure detector starts running, and for a while after, its behavior depends on the initial condition (such as whether initially $q$ suspects $p$ or not) and on how long it has been running. Typically, as time passes the effect of the initial condition gradually diminishes and its behavior no longer depends on how long it has been running — i.e., eventually the failure detector behavior reaches equilibrium, or steady state. In steady state, the probability law governing the behavior of the failure detector does not change over time. The QoS metrics that we propose refer to the behavior of a failure detector after it reaches steady state.

\subsection{Primary Metrics}

We propose three primary metrics for the QoS specification of failure detectors. The first one measures the speed of a failure detector. It is defined with respect to the
runs in which $p$ crashes.

**Detection time** ($T_D$): Informally, $T_D$ is the time that elapses from $p$’s crash to the time when $q$ starts suspecting $p$ permanently. More precisely, $T_D$ is a random variable representing the time that elapses from the time that $p$ crashes to the time when the final S-transition (of the failure detector at $q$) occurs and there are no transitions afterwards (Fig. 2.1). If there is no such final S-transition, then $T_D = \infty$; if such an S-transition occurs before $p$ crashes, then $T_D = 0$.

The next two metrics can be used to specify the accuracy of a failure detector. They are defined with respect to failure-free runs.\(^5\)

**Mistake recurrence time** ($T_{MR}$): this measures the time between two consecutive mistakes. More precisely, $T_{MR}$ is a random variable representing the time that elapses from an S-transition to the next one (Fig. 2.4). If no new S-transition occurs, then $T_{MR} = \infty$.

**Mistake duration** ($T_M$): this measures the time it takes the failure detector to correct a mistake. More precisely, $T_M$ is a random variable representing the time that elapses from an S-transition to the next T-transition (Fig. 2.4). If no S-transition

---

\(^5\)In Section 2.4, we explain why these metrics also measure the failure detector accuracy in runs in which $p$ crashes.
occurs, then $T_M = 0$; if no T-transition occurs after an S-transition, then $T_M = \infty$.

As we discussed in the introduction, there are many aspects of failure detector accuracy that may be important to applications. Thus, in addition to $T_{MR}$ and $T_M$, we propose four other accuracy metrics in the next section. We selected $T_{MR}$ and $T_M$ as the primary metrics because given these two, one can compute the other four (this will be shown in Section 2.3).

### 2.2.3 Derived Metrics

We propose the following four additional accuracy metrics (they are defined with respect to failure-free runs).

**Average mistake rate** ($\lambda_M$): this measures the rate at which a failure detector make mistakes, i.e., it is the average number of S-transitions per time unit. This metric is important to long-lived applications such as group membership and cluster management, where each mistake (each S-transition) results in a costly interrupt.

**Query accuracy probability** ($P_A$): this is the probability that the failure detector’s output is correct at a random time. This metric is important to applications that interact with the failure detector by querying it at random times.

Many applications are slowed down by failure detector mistakes. Such applications prefer a failure detector with long *good periods* — periods in which the failure detector makes no mistakes. This observation leads to the following two metrics.

**Good period duration** ($T_G$): this measures the length of a good period. More precisely, $T_G$ is a random variable representing the time that elapses from a T-transition to the next S-transition (Fig. 2.4). If no T-transition occurs, then $T_G = 0$; if no S-transition occurs after a T-transition, then $T_G = \infty$. 
For short-lived applications, however, a closely related metric may be more relevant. Suppose that an application is started at a random time, and that this happens to occur somewhere inside a good period. In this case, we are interested in measuring the *remaining portion* of this good period: if it is long enough, the short-lived application will be able to complete its task within this period. The corresponding metric is as follows.

**Forward good period duration** ($T_{FG}$): this is a random variable representing the time that elapses from a random time at which $q$ trusts $p$, to the time of the next S-transition. If no such S-transition occurs, then $T_{FG} = \infty$. If the probability that $q$ trusts $p$ at a random time is 0 (i.e. $P_A = 0$), then $T_{FG}$ is always 0.

At first sight, it may seem that, on the average, $T_{FG}$ is just half of $T_G$ (the length of a good period). But this is incorrect, and in Section 2.3 we give the actual relation between $T_{FG}$ and $T_G$.

We now give a simple example to illustrate how these definitions work.

**Example 1.** Consider the following simple failure detector algorithm $A$: process $p$ sends a heartbeat message to $q$ every one time unit; process $q$ suspects $p$ initially; every time $q$ receives a heartbeat message from $p$, $q$ trusts $p$ for one time unit, and by the end of the unit if $q$ has not received any new heartbeat message, then $q$ starts suspecting $p$.

Suppose that algorithm $A$ runs in the following simplified network environment: every heartbeat message is either lost, or is delivered instantaneously; each heartbeat message has an independent probability $p_L \in (0, 1)$ to be lost.

We now analyze all seven metrics of this failure detector. In this system, if $p$ does not crash, then $q$ keeps trusting $p$ if and only if the heartbeat messages are not lost.
Once a message is lost, \( q \) starts suspecting \( p \) immediately and the suspicion is kept until a new heartbeat message is received.

For the detection time \( T_D \), let \( T \) be the time elapsed between the time \( t \) when \( p \) sends the last message and the time \( t' \) when \( p \) crashes. \( T \) has a uniform distribution between 0 and 1. If the last heartbeat message is lost, then \( q \) starts suspecting \( p \) permanently at time \( t \) before \( p \) crashes, and so \( T_D = 0 \) in this case. If the last message is not lost, then \( q \) starts suspecting \( p \) permanently at time \( t + 1 \), and so \( T_D = 1 - T \) in this case. Therefore, the distribution of \( T_D \) is such that with probability \( p_L \), \( T_D = 0 \), and with probability \( 1 - p_L \), \( T_D = 1 - T \), where \( T \) has a uniform distribution between 0 and 1. Thus we have

\[
Pr(T_D \leq x) = \begin{cases} 
0 & x < 0 \\
p_L & x = 0 \\
p_L + x(1 - p_L) & 0 < x \leq 1 \\
1 & x > 1
\end{cases}
\]

For the accuracy metrics, suppose that \( p \) does not crash.

For the mistake duration \( T_M \), suppose that a message \( m_j \) is lost, which causes an S-transition of the failure detector. Then after \( m_j \), the first message that \( q \) receives causes the next T-transition. Since message losses are independent, we have that the probability that \( m_{j+i} \) is the first message that \( q \) receives after \( m_j \) is \( p_l^{i-1}(1 - p_L) \), for all \( i \geq 1 \). Since messages are sent every one unit of time and are delivered instantaneously if not lost, we know that the distribution of \( T_M \) is such that with probability \( p_l^{i-1}(1 - p_L) \), \( T_M = i \), \( i \geq 1 \). This is a geometric distribution with parameter \( 1 - p_L \).

The good period duration \( T_G \) has a distribution symmetric to the distribution of
Suppose that message $m_j$ is received and it causes a T-transition. Then after $m_j$, the first message that is lost causes the next S-transition. Thus the distribution of $T_o$ is such that with probability $(1 - p_L)^{i-1}p_L$, $T_o = i, i \geq 1$. This is a geometric distribution with parameter $p_L$.

For the mistake recurrence time $T_{MR}$, starting at an S-transition, the first message that is received causes the next T-transition, and then the first message that is lost causes the next S-transition. The length of the suspicion period is independent of the length of the trust period that follows, due to the independence of message loss. Thus $T_{MR}$ is the sum of two independent random variable $X$ and $Y$, where $X$ and $Y$ have geometric distributions with parameters $1 - p_L$ and $p_L$, respectively.

For the average mistake rate $\lambda_M$, in any unit time interval in steady state, there is either no S-transition or exactly one S-transition. Thus the average number of S-transitions in the unit interval is just the probability that one S-transition occurs in the interval. An S-transition occurs in the interval if and only if the message sent in the interval is lost and the previous message is not lost. Thus the probability that an S-transition occurs in the interval is $p_L(1 - p_L)$. Therefore, $\lambda_M = p_L(1 - p_L)$.

For the query accuracy $P_A$, $q$ trusts $p$ at a random time $t$ if and only if the message sent before $t$ is not lost. Therefore, $P_A = 1 - p_L$.

For the forward good period duration $T_{FG}$, suppose that $q$ trusts $p$ at a random time $t$. Let $T'$ be the time elapsed from $t$ to the time when next heartbeat message is sent. Then $T'$ has a uniform distribution from $0$ to $1$. From time $t$ on, an S-transition occurs when a heartbeat message is lost. Therefore, $T_{FG}$ has the distribution such that with probability $(1 - p_L)^i p_L$, $T_{FG} = T' + i, i \geq 0$. 

$\square$
2.3 Relations between Accuracy Metrics

In Theorem 2.1 below we state the relation between the six accuracy metrics that we defined in the previous sections. We then use this theorem to justify our choice of the primary accuracy metrics.

Henceforth, \( Pr(A) \) denotes the probability of event \( A \), and \( E(X) \), \( V(X) \), and \( \sigma(X) \) denote the expected value (or mean), the variance, and the standard deviation of random variable \( X \), respectively.

Parts (2) and (3) of Theorem 2.1 assume that in failure-free runs, the probabilistic distribution of failure detector histories is \textit{ergodic}. Roughly speaking, this means that in failure-free runs, the failure detector slowly “forgets” its past history: from any given time on, its future behavior may depend only on its recent behavior. We call failure detectors satisfying this ergodicity condition \textit{ergodic failure detectors}. In Chapter 3, we formally define the ergodicity condition, prove the following theorem, and also show the relations between our accuracy metrics in the case that ergodicity does \textit{not} hold.

**Theorem 2.1** For any ergodic failure detector, the following results hold:

1. \( T_G = T_{MR} - T_M \).

2. If \( 0 < E(T_{MR}) < \infty \), then:
   \[
   \lambda_M = \frac{1}{E(T_{MR})},
   \]
   \[
   P_A = \frac{E(T_G)}{E(T_{MR})} = \frac{E(T_{MR}) - E(T_M)}{E(T_{MR})}.
   \]

3. If \( 0 < E(T_{MR}) < \infty \) and \( E(T_G) = 0 \), then \( T_{FG} \) is always 0. If \( 0 < E(T_{MR}) < \infty \)
and $E(T_G) \neq 0$, then:

for all $x \in [0, \infty)$, \[ Pr(T_{FG} \leq x) = \frac{1}{E(T_G)} \int_0^x Pr(T_G > y)dy, \] (2.3)

\[ E(T^k_{FG}) = \frac{E(T^{k+1}_G)}{(k+1)E(T_G)}. \] (2.4)

In particular,

\[ E(T_{FG}) = \frac{E(T^2_G)}{2E(T_G)} = \frac{E(T_G)}{2} \left( 1 + \frac{V(T_G)}{E(T_G)^2} \right). \] (2.5)

The fact that $T_G = T_{MR} - T_M$ holds is immediate by definition. Equalities (2.1) and (2.2) are intuitive, but (2.3), (2.4) and (2.5), which describe the relation between $T_G$ and $T_{FG}$, are more complex. Moreover, (2.5) is counter-intuitive: one may think that $E(T_{FG}) = E(T_G)/2$, but (2.5) says that $E(T_{FG})$ is in general larger than $E(T_G)/2$ (this is a version of the “waiting time paradox” in the theory of stochastic processes [All90]).

We now explain how Theorem 2.1 guided our selection of the primary accuracy metrics. Equalities (2.1), (2.2) and (2.3) show that $\lambda_M$, $P_3$ and $T_{FG}$ can be derived from $T_{MR}$, $T_M$ and $T_G$. This suggests that the primary metrics should be selected among $T_{MR}$, $T_M$ and $T_G$. Moreover, since $T_G = T_{MR} - T_M$, it is clear that given the joint distribution of any two of them, one can derive the remaining one. Thus, two of $T_{MR}$, $T_M$ and $T_G$ should be selected as the primary metrics, but which two? By choosing $T_{MR}$ and $T_M$ as our primary metrics, we get the following convenient property that helps to compare failure detectors: if $FD_1$ is better than $FD_2$ in terms of both $E(T_{MR})$ and $E(T_M)$ (the expected values of the primary metrics) then we can be sure that $FD_1$ is also better than $FD_2$ in terms of $E(T_G)$ (the expected values of
the other metric). We would not get this useful property if $T_G$ were selected as one of the primary metrics.\footnote{For example, $FD_1$ may be better than $FD_2$ in terms of both $E(T_G)$ and $E(T_M)$, but worse than $FD_2$ in terms of $E(T_{MR})$.}

We now demonstrate parts (2) and (3) of Theorem 2.1 with the example in the previous section.

\textbf{Example 2.} Consider again the algorithm in Example 1. Since message losses are independent, the behavior of the failure detector does not depend on what happened in the past history, and so the distribution of failure detector histories in failure-free runs is ergodic.

$T_G$ has a geometric distribution with parameter $p_L$, so we have $E(T_G) = 1/p_L$ and $V(T_G) = (1 - p_L)/p_L^2$. Similarly, we have $E(T_{n}) = 1/(1 - p_L)$, and $E(T_{MR}) = 1/p_L + 1/(1 - p_L) = 1/[p_L(1 - p_L)]$. With $p_L \in (0, 1)$, we have $0 < E(T_{MR}) < \infty$ and $E(T_G) \neq 0$. Thus the conditions for Equalities (2.1)--(2.5) to hold are true.

From Example 1, we know that $\lambda_M = p_L(1 - p_L)$ and $P_A = 1 - p_L$. Since $E(T_{MR}) = 1/[p_L(1 - p_L)]$ and $E(T_G) = 1/p_L$, Equalities (2.1) and (2.2) hold for this failure detector.

We now check Equality (2.3). Given any $x \in [0, \infty)$, let $n = \lfloor x \rfloor$ and $r = x - n$. From Example 1, we know that $T_{FG}$ has the distribution such that with probability $(1 - p_L)^i p_L$, $T_{FG} = T' + i$, $i \geq 0$, where $T'$ has a uniform distribution from 0 to 1. Then

$$Pr(T_{FG} \leq x) = Pr(T_{FG} < n) + Pr(n \leq T_{FG} \leq x)$$

$$= \sum_{i=0}^{n-1} (1 - p_L)^i p_L + (1 - p_L)^n p_L Pr(0 \leq T' \leq r)$$

$$= 1 - (1 - p_L)^n + rp_L(1 - p_L)^n.$$
On the other side, for any $y \in [i - 1, i)$, $Pr(T_G > y) = \sum_{j=i}^{\infty} Pr(T_G = j) = \sum_{j=i}^{\infty}(1 - p_L)^{j-1}p_L = (1 - p_L)^{i-1}$. Thus
\[
\frac{1}{E(T_G)} \int_0^x Pr(T_G > y) \, dy = p_L \left[ \sum_{i=1}^{n} \int_{i-1}^{i} Pr(T_G > y) \, dy + \int_{n}^{n+r} Pr(T_G > y) \, dy \right]
\]
\[
= p_L \left[ \sum_{i=1}^{n+1} (1 - p_L)^{i-1} + r(1 - p_L)^n \right]
\]
\[
= 1 - (1 - p_L)^n + rp_L(1 - p_L)^n.
\]
Therefore, Equality (2.3) holds for this failure detector.

Equalities (2.4) and (2.5) are direct consequences of Equality (2.3), and we only check Equality (2.5) here. From the distribution of $T_{FG}$, we have $E(T_{FG}) = \sum_{i=0}^{\infty} E(T' + i)(1 - p_L)^i p_L = \sum_{i=0}^{\infty} (i + 1/2)(1 - p_L)^i p_L = (2 - p_L)/(2p_L)$. On the other side,
\[
\frac{E(T_G)}{2} \left( 1 + \frac{V(T_G)}{E(T_G)^2} \right) = \frac{1}{2p_L} \left( 1 + \frac{(1 - p_L)/p_L^2}{1/p_L^2} \right) = \frac{2 - p_L}{2p_L}.
\]
Therefore, Equality (2.5) holds for this failure detector.

Note that for this failure detector $E(T_{FG}) = (2 - p_L)/(2p_L)$ while $E(T_G) = 1/p_L$, so $E(T_{FG}) > E(T_G)/2$. \hfill \Box

\section{2.4 Discussion}

\textbf{On the Probability of Premature Timeouts}

For timeout-based failure detectors, the probability of premature timeouts is sometimes used as the accuracy measure: this is the probability that when the timer is set, it will prematurely timeout on a process that is actually up. The problem with this measure, however, is that (a) it is implementation-specific, and (b) it is not
useful to applications unless it is given together with other implementation-specific measures, e.g., how often timers are started, whether the timers are started at regular or variable intervals, whether the timeout periods are fixed or variable, etc. (many such variations exist in practice [Bra89, GM98, vRMH98]). Thus, the probability of premature timeouts is not a good metric for the specification of failure detectors, e.g., it cannot be used to compare the QoS of failure detectors that use timeouts in different ways. The six accuracy metrics that we identified in this paper do not refer to implementation-specific features, in particular, they do not refer to timeouts at all.

**Accuracy Metrics and Runs with Crashes**

To measure the accuracy of a failure detector that monitors a process $p$, we considered runs in which $p$ does not crash. In real systems, however, such runs rarely occur: $p$ is likely to crash eventually. Are our accuracy metrics applicable to such systems? The answer is yes, as we now explain.

Note that the output of any failure detector implementation at a time $t$ should not depend on what happens after time $t$, i.e., the implementation does not predict the future.\(^7\) Therefore, the steady state behavior of a failure detector before a process $p$ crashes is the same as its steady state behavior in runs in which $p$ does not crash. Thus, our accuracy metrics also measure the accuracy of a failure detector in runs in which $p$ eventually crashes (provided that this crash occurs after the failure detector has reached its steady state behavior).

\(^7\)Our model can enforce this assumption by imposing some restriction on the sets of failure detector histories and their associated distributions (see Chapter 3).
Good Periods versus Stable Periods

Recall that a good period of a failure detector is defined in terms of runs in which \( p \) does not crash. It starts when the failure detector trusts \( p \) (makes a T-transition) and terminates when the failure detector erroneously suspects \( p \) (makes an S-transition).

In contrast, a stable period of a system starts when the failure detector trusts \( p \) and \( p \) is up, and terminates when either: (a) the failure detector suspects \( p \), or (b) \( p \) crashes. The length of stable periods is an important measure for many applications. This measure, however, cannot be part of the QoS specification of failure detectors: since a failure detector has no control over process crashes, it cannot by itself ensure “long” stable periods, even if it is very accurate.

To measure the length of a stable period in a system, one can use measures on the accuracy of the failure detector and on the likelihood of crashes. For example, let \( T_G \) be the random variable representing the length of a good period of the failure detector, and \( C \) be the random variable representing the lifetime of process \( p \). Assume that \( C \) has an exponential distribution (so that at any given time at which \( p \) is still up, the remaining lifetime of \( p \) has the same distribution as \( C \)). Let \( S \) be the random variable representing the length of a stable period after the failure detector has reached steady state. Then the distribution of \( S \) is given by \( \Pr(S \leq x) = 1 - \Pr(C > x)\Pr(T_G > x) \). Intuitively, this is because a stable period terminates as soon as the failure detector makes a mistake or \( p \) crashes.
Chapter 3

Stochastic Modeling of Failure Detectors and Their Quality-of-Service Specifications

3.1 Introduction

In Chapter 2, we proposed a set of metrics for the QoS specification of failure detectors. The definitions of failure detectors and the QoS metrics were kept at an intuitive level to emphasize the main ideas of the QoS specification of failure detectors. In this chapter, we give a rigorous formalization of failure detectors and their QoS metrics based on stochastic modeling, in particular the theory of marked point processes (c.f. [Sig95]). Upon the first reading, readers can skip this chapter and read Chapter 4 without any difficulty.

In the formalization, we first define random failure detector histories that model
the probabilistic behaviors of failure detectors. We show that a random failure
detector history is an extension of a (particular type of) random marked point process.
We next define failure detectors as mappings from failure patterns to random failure
detector histories. This is an extension to the failure detector model of [CT96].
We then define some particular random failure detector histories as the steady state
behaviors of a failure detector and use them to define the QoS metrics. Some of
these random failure detector histories match with the stationary versions of random
marked point processes. Finally, we analyze the relation between the QoS metrics we
defined. The analysis is based on the results in the theory of marked point processes,
such as Birkhoff’s Ergodic Theorem for marked point processes, and the empirical
inversion formulas. The relations we present in this chapter are more general than
the results in Theorem 2.1 of Chapter 2.

The rest of the chapter is organized as follows. In Section 3.2, we define the failure
detector model, which includes the definitions of random failure detector histories,
failure detectors, and the steady state behaviors of failure detectors. In Section 3.3,
we define the QoS metrics and analyze the relation between these metrics.

In Appendix A, we summarize relevant definitions and results in the theory of
marked point processes.

3.2 Failure Detector Model

As in Chapter 2, we consider a system of two processes $p$ and $q$, and a failure detector
at $q$ that monitors $p$. We assume that $q$ does not crash. Real time is continuous and
ranges from 0 to $\infty$. 
3.2.1 Failure Detector Definition

As described in Section 2.2.1, the output of the failure detector is denoted as either $S$ or $T$, and it has two types of transitions: $S$-transitions and $T$-transitions. Roughly speaking, a failure detector history describes the output of the failure detector in an entire run, and it can be represented by the initial output and the times at which transitions occur.

More precisely, we define a failure detector history as follows. Let $\mathbb{R}$, $\mathbb{R}_+$, and $\mathbb{Z}_+$ denote the set of real numbers, nonnegative real numbers, and nonnegative integers, respectively. Let $K = \{S, T\}$. For $x \in K$, let $\overline{x}$ denote the element other than $x$ in $K$.

A failure detector history is given as $\psi = \langle k, \{t_n : n \in I\} \rangle$ such that

1. $k \in K$;
2. $I = \mathbb{Z}_+$ or $I = \{0, 1, \ldots, m - 1\}$ for some $m \in \mathbb{Z}_+$ (if $m = 0$, then $I = \emptyset$);
3. $t_n \in \mathbb{R}_+$ for all $n \in I$;
4. if $I = \mathbb{Z}_+$, then $0 \leq t_0 < t_1 < t_2 < \cdots$, and $\lim_{n \to \infty} t_n = \infty$;
5. if $|I| = m < \infty$, then $0 \leq t_0 < t_1 < t_2 < \cdots < t_{m-1}$.

In the representation of failure detector history $\psi$, $k$ represents the output of the failure detector at time 0, and the increasing sequence $\{t_n\}$ represents the times at which failure detector transitions occur. We call $t_n$ the $n$-th transition time (so $\psi$ starts with the zeroth transition). When $t_0 > 0$, $\psi = \langle k, \{t_n\} \rangle$ represents a run in which the failure detector outputs $k$ in the period $[0, t_0)$, makes a transition at time $t_0$, then outputs $\overline{k}$ in the period $[t_0, t_1)$, and then makes another transition at time $t_1$, and so on. When $t_0 = 0$, $\psi = \langle k, \{t_n\} \rangle$ represents a run in which failure detector has a transition at time 0, then outputs $k$ in the period $[t_0, t_1)$, makes a transition at
time \( t_1 \), and then outputs \( \overline{k} \) in the period \([t_1, t_2)\), and so on. Allowing a transition at time 0 is to conform with the representation of marked point processes (as in [Sig95]), which is the basic tool we use to model the failure detectors. Intuitively it makes sense when there is output before time 0, and this can happen if the time line is shifted. The requirement \( \lim_{n \to \infty} t_n = \infty \) in (4) enforces that there are only a finite number of transitions in any bounded time interval.

For a failure detector history \( \psi = \langle k, \{t_n : n \in I\} \rangle \), we define the \( n \)-th inter-transition time \( T_n \) of \( \psi \) as follows. If \(|I| = \infty\), then \( T_n \defeq t_{n+1} - t_n \) for all \( n \geq 0 \); if \(|I| = m < \infty\), then (a) if \( m = 0 \), then \( T_0 \defeq \infty \) and \( T_n \defeq 0 \) for \( n \geq 1 \); and (b) if \( m \geq 1 \), then \( T_n \defeq t_{n+1} - t_n \) for \( 0 \leq n \leq m - 2 \), \( T_{m-1} \defeq \infty \), and \( T_n \defeq 0 \) for \( n \geq m \).

To model the probabilistic behavior of a failure detector, we need to define random failure detector histories with probability distributions over the set of failure detector histories. To do so, we first need to define what are the subsets of failure detector histories that we can assign probability to. Formally, we need to define a \( \sigma \)-field which contains all measurable subsets of failure detector histories. This is done as follows.

Let \( H \) be the set of all failure detector histories. Let \( Z_\infty^{\infty} \defeq \mathbb{Z}_+ \cup \{\infty\} \) for \( m \in \mathbb{Z}_\infty^{\infty} \), let \( H^{(m)} \) be the set of all failure detector histories with exactly \( m \) transitions, i.e. \( H^{(m)} \defeq \{ \psi = \langle k, \{t_n : n \in I\} \rangle : |I| = m \} \). Thus \( \{H^{(m)} : m \in \mathbb{Z}_\infty^{\infty}\} \) forms a partition of \( H \). We next define the Borel \( \sigma \)-field \( \mathcal{B}(H^{(m)}) \) of \( H^{(m)} \) for each \( m \in \mathbb{Z}_\infty^{\infty} \).

When \( m < \infty \), \( H^{(m)} \) is a subset of \( K \times \mathbb{R}^m \), where \( \mathbb{R}^m \) is the \( m \)-dimensional Euclidean space with the Borel \( \sigma \)-field \( \mathcal{B}(\mathbb{R}^m) \). Let \( \mathcal{B}(K \times \mathbb{R}^m) \) be the product \( \sigma \)-field generated by \( \{K \times B : K \subseteq K, B \in \mathcal{B}(\mathbb{R}^m)\} \). It is easy to check that \( H^{(m)} \in \mathcal{B}(K \times \mathbb{R}^m) \). Then, we get the Borel \( \sigma \)-field \( \mathcal{B}(H^{(m)}) \defeq \{ \mathcal{E} : \mathcal{E} \in \mathcal{B}(K \times \mathbb{R}^m) \} \).
\( \mathbb{R}^m \), \( \mathcal{E} \subseteq H^{(m)} \).

When \( m = \infty \), \( H^{(\infty)} \) is a subset of \( K \times \mathbb{R}^+ \), where \( \mathbb{R}^+ \) is the set of all countably infinite sequences of real numbers. It is known (see e.g. [Sig95]) that \( \mathbb{R}^+ \) is a complete separable metric space, and the Borel \( \sigma \)-field \( \mathcal{B}(\mathbb{R}^+) \) is well defined. Let \( \mathcal{B}(\mathbb{K} \times \mathbb{R}^+) \) be the product \( \sigma \)-field generated by \( \{ K \times B : K \subseteq \mathbb{K}, B \in \mathcal{B}(\mathbb{R}^+) \} \).

It is easy to check that \( H^{(\infty)} \in B(\mathbb{K} \times \mathbb{R}^+) \). Then, we get the Borel \( \sigma \)-field \( \mathcal{B}(H^{(\infty)}) \) defined as \( \{ E : E \in \mathcal{B}(\mathbb{K} \times \mathbb{R}^+), E \subseteq H^{(\infty)} \} \).

With the above definitions of \( \mathcal{B}(H^{(m)}) \) for all \( m \in \mathbb{Z}^+ \), we then define the Borel \( \sigma \)-field \( \mathcal{B}(H) \) of \( H \) to be \( \{ \bigcup_{m \in \mathbb{Z}^+} E_m : E_m \in \mathcal{B}(H^{(m)}) \} \). Hence we have a measurable space \( (H, \mathcal{B}(H)) \) on the set of all failure detector histories.

Some simple examples of Borel sets in \( H \) are: (1) \( \{ \psi \in H : k = S \} \), the set of failure detector histories in which the output at time 0 is \( S \); (2) \( \{ \psi \in H : t_0 \leq x \} \), the set of failure detector histories in which the zeroth transition occurs within \( x \) time units; and (3) \( \{ \psi \in H : T_0 \leq x \} \), the set of failure detector histories in which the zeroth intertransition time is at most \( x \) time units.

It is easy to verify that \( \mathcal{B}(H) \) can also be generated from the following collection of the sets:

\[
\{ \psi = (k, \{ t_n : n \in \mathbb{I} \}) \in H : |\mathbb{I}| = m, k \in K, t_{n_0} \leq x_0, \ldots, t_{n_l} \leq x_l \},
\]

where \( m \in \mathbb{Z}^+_\infty, K \subseteq \mathbb{K}, l \in \mathbb{I}, 0 \leq n_0 < \cdots < n_l < m, x_i \in \mathbb{R}_+ \).

We define a random failure detector history to be a measurable mapping \( \Psi : \Omega \to H \), with \( (\Omega, \mathcal{F}, P) \) as the underlying probability space. With this definition, the random failure detector history \( \Psi \) has the probability distribution \( P(\Psi \in \mathcal{E}) \) defined for all \( \mathcal{E} \in \mathcal{B}(H) \). For convenience, we use \( \{ \Psi \in \mathcal{E} \} \) as a
short hand for \( \{ \omega \in \Omega : \Psi(\omega) \in \mathcal{E} \} \). Let \( \Psi \) be the set of all random failure detector histories.

A failure pattern \( F \) of process \( p \) is just a number \( F \in [0, \infty] \), denoting the time \( F \) at which \( p \) crashes; \( F = \infty \) means that \( p \) does not crash, and we call this pattern failure-free pattern. Let \( \mathcal{F} \) denote the set of all failure patterns. Thus \( \mathcal{F} = [0, \infty] \).

A failure detector \( \mathcal{D} \) is a mapping \( \mathcal{D} : \mathcal{F} \to \Psi \). Intuitively, the random failure detector history \( \mathcal{D}(F) \) characterizes the probabilistic behavior of the failure detector output when process \( p \) crashes at time \( F \). This is an extension of the failure detector definition in [CT96] to model the probabilistic behavior of the failure detector output.

### 3.2.2 Failure Detector Histories as Marked Point Processes

We now build the relation between failure detector histories and marked point processes.

Given a failure detector history \( \psi = \langle k, \{ t_n : n \in \mathbb{I} \} \rangle \) where \( \mathbb{I} \neq \emptyset \), let \( k_n \) be the output of the failure detector at time \( t_n \) for all \( n \in \mathbb{I} \). Thus, we know that the transition occurred at time \( t_n \) is a \( k_n \)-transition, and in period \( [t_n, t_{n+1}) \), the failure detector output is \( k_n \). The relation between \( k \) and \( k_n \)'s is: (1) if \( t_0 = 0 \), then \( k_0 = k_2 = \cdots = k \) and \( k_1 = k_3 = \cdots = k \); and (2) if \( t_0 > 0 \), then \( k_0 = k_2 = \cdots = k \) and \( k_1 = k_3 = \cdots = k \). For notational convenience, let \( t_{-1} \overset{\text{def}}{=} 0 \), and \( k_{-1} \overset{\text{def}}{=} k \).

With \( k_n \)'s, the failure detector history \( \psi \) can be equivalently represented as \( \psi = \{(t_n, k_n) : n \in \mathbb{I}\} \). When \( \mathbb{I} = \mathbb{Z}_+ \), this representation coincides with the representation of a simple marked point process as given in [Sig95]. In fact, A failure detector history with an infinite number of transitions can be directly modeled as a simple marked point process, with transitions as events and \( \mathcal{K} \) as the mark space.
Therefore, definitions and results for marked point processes can be directly applied to failure detector histories with an infinite number of transitions. For consistency and convenience, we extend some of the definitions to include failure detector histories with only a finite number of transitions.

One important extension is the shift mappings on failure detector histories with a finite number of transitions, as given below. Suppose \( \theta_s : H \to H \) is a shift mapping defined on all failure detector histories. Intuitively, for a failure detector history \( \psi \), \( \theta_s\psi \) is the failure detector history obtained from \( \psi \) by shifting the origin to \( s \), using the output at time \( s \) as the initial output, re-labeling transitions at and after \( s \) as \( t_0, t_1, \ldots \), and ignoring the portion of the failure detector history before \( s \). More precisely, if \( s = 0 \), then \( \theta_s \) is the identity mapping; if \( \psi \) has an infinite number of transitions, then \( \psi \) is also a simple marked point process, and thus \( \theta_s\psi \) is defined as in Appendix A. Now suppose \( s > 0 \) and \( \psi = \langle k, \{t_n : n \in I\} \rangle \) has only a finite number of transitions, i.e., \( |I| = m < \infty \). If \( t_{i-1} < s \leq t_i \) for some \( i \in I \), then
\[
\theta_s\psi \overset{\text{def}}{=} \langle k', \{t_{i+n} - s : 0 \leq n \leq m - 1 - i\} \rangle,
\]
where \( k' \) is the output at time \( s \), and \( k' = k_i \) if \( s = t_i \); \( k'_i = k_i \) if \( s < t_i \). If \( s > t_{m-1} \), then \( \theta_s\psi \overset{\text{def}}{=} \langle k_{m-1}, \emptyset \rangle \).

We now define shift mapping by event time \( \theta_j \) for \( j \geq 0 \). Intuitively, for a failure detector history \( \psi \), \( \theta_j\psi \) is a failure detector history obtained from \( \psi \) by shifting the origin to the time of \( j \)-th transition in \( \psi \), and if \( \psi \) does not have enough transitions, then the origin is shifted to the last transition of \( \psi \). More precisely, if \( \psi \) has at least \( j + 1 \) transitions, then \( \theta_j\psi \overset{\text{def}}{=} \theta_{t_j} \); if \( \psi \) has less than \( j + 1 \) transitions, then \( \theta_j\psi \overset{\text{def}}{=} \theta_{t_{m-1}} \), where \( m \) is the number of transitions in \( \psi \). We then let
\[
\psi_s \overset{\text{def}}{=} \theta_s\psi \quad \text{and} \quad \psi_j \overset{\text{def}}{=} \theta_j\psi. \tag{3.2}
\]
Note that $\psi(j)$ always has a transition at the origin, except the case when $\psi$ itself has no transition at all.

For a random failure detector history $\Psi : \Omega \to H$, let $\Psi_s : \Omega \to H$ be a random failure detector history obtained from $\Psi$ by shifting the origin to time $s$, that is, $\Psi_s(\omega) = \Psi(\omega)_s$ for all $\omega \in \Omega$. Similarly, let $\Psi_{(j)} : \Omega \to H$ be a random failure detector history obtained from $\Psi$ by shifting the origin to the time of the $j$-th transition, that is, $\Psi_{(j)}(\omega) = \Psi(\omega)_{(j)}$ for all $\omega \in \Omega$. Intuitively, $\Psi_s$ represents what you see if you always start observing $\Psi$ at time $s$, and $\Psi_{(j)}$ represents what you see if you always start observing $\Psi$ at the $j$-th transition.

Shift mapping is an important tool to the study of the steady state behaviors of failure detectors, as we discuss in the next section.

3.2.3 The Steady State Behaviors of Failure Detectors

We consider failure detectors whose behaviors eventually reach the steady state. Roughly speaking, when a failure detector starts running, and for a while after, its behavior depends on the initial condition (such as whether initially $q$ suspects $p$ or not) and on how long it has been running. Typically, as time passes the effect of the initial condition gradually diminishes and its behavior no longer depends on how long it has been running — i.e., eventually the failure detector behavior reaches equilibrium, or steady state.

Suppose that while $p$ is still up, the behavior of the failure detector reaches a steady state. We consider two kinds of behaviors in this case: First, if $p$ remains up, what would be the behavior of the failure detector? Second, if $p$ crashes, what would be the behavior of the failure detector in response of the crash of $p$?
We now formally define several random failure detector histories that capture such steady state behaviors. Let $\mathcal{D}$ be the failure detector in consideration.

The Steady State Behavior If $p$ Does Not Crash

Let $F = \infty$ be the failure-free pattern of $p$. Then $\Psi \overset{\text{def}}{=} \mathcal{D}(F)$ defines the behavior of the failure detector output under this failure-free pattern. Suppose the underlying probability space defining $\Psi$ is $(\Omega, \mathcal{F}, P)$. The steady state behavior of $\Psi$ is given by its event stationary version $\Psi^0$ and time stationary version $\Psi^*$, if they exist. Formally, they are defined by the following distributions (assuming they exist), just as the definitions in Section 2.3 of [Sig95]:

$$Pr(\Psi^0 \in \mathcal{E}) \overset{\text{def}}{=} \lim_{n \to \infty} \frac{1}{n} \sum_{j=0}^{n-1} P(\Psi(j) \in \mathcal{E}), \quad \text{for all } \mathcal{E} \in \mathcal{B}(H), \tag{3.3}$$

$$Pr(\Psi^* \in \mathcal{E}) \overset{\text{def}}{=} \lim_{t \to \infty} \frac{1}{t} \int_0^t P(\Psi_s \in \mathcal{E}) \, ds, \quad \text{for all } \mathcal{E} \in \mathcal{B}(H). \tag{3.4}$$

The event stationary version $\Psi^0$ is obtained by averaging the distribution of $\Psi(j)$ over all transition times, and the time stationary version $\Psi^*$ is obtained by averaging the distribution of $\Psi_s$ over all times. Such average distributions are referred to as empirical distributions in [Sig95]. The intuitive meanings of (3.3) and (3.4) are: if we randomly pick a transition and start observing $\Psi$ after this transition, the random failure detector history we observed is given by the event stationary version $\Psi^0$; if we randomly pick a real time $s$ and then observe the behavior of $\Psi$ after $s$, then the

\footnote{Note that in (3.3) and (3.4) we use the notation $Pr(\cdot)$ to avoid the complication of specifying the underlying probability spaces for $\Psi^0$ and $\Psi^*$. These stationary versions can be defined in probability spaces different from $\Psi$, but there is no need to specify them here since we are only interested in the probability distributions of $\Psi^0$ and $\Psi^*$. We will use the notation $Pr(\cdot)$ whenever it is convenient for us.}
random failure detector history we observed is given by the time stationary version $\Psi^*$. Using the expressions in [Sig95], $\Psi^0$ is the version of $\Psi$ when we randomly observe $\Psi$ way out at a transition, and $\Psi^*$ is the version of $\Psi$ when we randomly observe $\Psi$ way out in time.

Intuitively, $\Psi^0$ is event stationary, i.e., the distribution does not change if $\Psi^0$ is shifted by transition times (see Appendix A for its definition), because after already randomly observing $\Psi$ way out at a transition to obtain $\Psi^0$, observing $\Psi$ several transitions later makes no difference to the distribution of $\Psi^0$. Similarly, $\Psi^*$ is time stationary, i.e, the distribution does not change if $\Psi^*$ is shifted by time, because after already randomly observing $\Psi$ way out in time to obtain $\Psi^*$, observing $\Psi$ some time units later makes no difference to the distribution of $\Psi^*$.

**Lemma 3.1** $\Psi^0$ is event stationary and $\Psi^*$ is time stationary.

**Proof.** The proof is the same as the proof in [Sig95] p.26, except that the definitions of shift mappings are extended to include the case where the number of transitions are finite. \qed

We say that the behavior of the failure detector $D$ reaches steady state in failure-free runs if the distributions defined in (3.3) and (3.4) exist. The accuracy metrics of the failure detector are defined with respect to the steady state behavior of the failure detector in failure-free runs, i.e., with respect to stationary versions $\Psi^0$ and $\Psi^*$.

To further understand the stationary versions $\Psi^0$ and $\Psi^*$, we break down events in $B(H)$ into different categories and study them separately. From Section 3.2.1, we know that $\{H^{(m)} : m \in \mathbb{Z}^\infty_+\}$ is a partition of $H$, and $B(H) = \left\{ \bigcup_{m \in \mathbb{Z}^\infty_+} E_m : E_m \in B(H^{(m)}) \right\}$. Therefore, for any event $\mathcal{E} = \bigcup_{m \in \mathbb{Z}^\infty_+} \mathcal{E}_m$, if we know $Pr(\Psi^0 \in \mathcal{E}_m)$
for all \( m \in \mathbb{Z}_+^\infty \), then by the additivity of the probability measure we know that \( \Pr(\Psi^0 \in \mathcal{E}) = \sum_{m \in \mathbb{Z}_+^\infty} \Pr(\Psi^0 \in \mathcal{E}_m) \). The case for \( \Pr(\Psi^* \in \mathcal{E}) \) is similar. Thus, we now focus on \( \Pr(\Psi^0 \in \mathcal{E}) \) and \( \Pr(\Psi^* \in \mathcal{E}) \) with \( \mathcal{E} \in \mathcal{B}(\mathcal{H}^{(m)}) \), for each \( m \in \mathbb{Z}_+^\infty \).

Let \( \mathcal{E}_S \) and \( \mathcal{E}_T \) be the sets of all failure detector histories in which eventually the output is always \( S \) or \( T \), respectively. Thus \( \mathcal{E}_S \cup \mathcal{E}_T \) contains all failure detector histories with a finite number of transitions. Let \( p_S^\Psi \equiv P(\Psi \in \mathcal{E}_S) \) and \( p_T^\Psi \equiv P(\Psi \in \mathcal{E}_T) \). Thus \( p_S^\Psi \) and \( p_T^\Psi \) are the probabilities that eventually the output of the random failure detector history \( \Psi \) is always \( S \) or always \( T \), respectively. Let \( p_\infty^\Psi \equiv P(\Psi \in \mathcal{H}^{(\infty)}) \), the probability that \( \Psi \) has an infinite number of transitions.

**Proposition 3.2** \( p_S^\Psi + p_T^\Psi + p_\infty^\Psi = 1 \).

**Proof.** It is direct from the fact that \( \mathcal{E}_S \), \( \mathcal{E}_T \), and \( \mathcal{H}^{(\infty)} \) are disjoint and \( \mathcal{E}_S \cup \mathcal{E}_T \cup \mathcal{H}^{(\infty)} = \mathcal{H} \).

Probabilities \( p_S^\Psi \) and \( p_T^\Psi \) are used to characterize the steady state behavior of failure detector histories with only a finite number of transitions. Intuitively, if a run of the failure detector only has a finite number of transitions, then in steady state the failure detector should keep its final output value. In other words, when you randomly observe a failure detector history with a finite number of transitions way out in time or way out at a transition, with probability one what you observe is the portion in which the failure detector keeps its final output value. The probability that the output you observe is \( S \) or \( T \) is given by \( p_S^\Psi \) or \( p_T^\Psi \), respectively. This is formalized in the following lemma.

**Lemma 3.3**

\[(1) \ Pr(\Psi^* \in \{(S, \emptyset)\}) = p_S^\Psi, \text{ and } Pr(\Psi^* \in \{(T, \emptyset)\}) = p_T^\Psi; \]
(2) For all \( m \in \mathbb{Z}_+ \setminus \{0\} \), for all \( E \in \mathcal{B}(H^{(m)}) \), \( Pr(\Psi^* \in E) = 0 \);

(3) \( Pr(\Psi^0 \in \{(S, \emptyset), (S, \{0\})\}) = p_S^\Psi \), and \( Pr(\Psi^0 \in \{(T, \emptyset), (T, \{0\})\}) = p_T^\Psi \);

(4) For all \( m \in \mathbb{Z}_+ \setminus \{0\} \), for all \( E \in \mathcal{B}(H^{(m)}) \), if \( E \) does not contain \( (S, \{0\}) \) or \( (T, \{0\}) \), then \( Pr(\Psi^0 \in E) = 0 \).

**Proof.** (1) Let \( E = \{(S, \emptyset)\} \). We have that if \( s \leq t \), then \( \{\Psi_s \in E\} \subseteq \{\Psi_t \in E\} \), i.e., if a failure detector history remains the output \( S \) from time \( s \) on, then it of course remains the output \( S \) from a later time \( t \) on. Thus \( P(\Psi_s \in E) \leq P(\Psi_t \in E) \). It is clear that \( \{\Psi_t \in E\} \uparrow \{\Psi \in E_S\} \), i.e., as \( t \to \infty \), \( \{\Psi_t \in E\} \) monotonically increasing and tends to \( \{\Psi \in E_S\} \) from below. Then for integer valued \( n \), \( \{\Psi_n \in \E\} \uparrow \{\Psi \in \E_S\} \).

Since the probability measure is continuous from below (see e.g. [Bil95], p.25), we have \( Pr(\Psi_n \in E) \uparrow Pr(\Psi \in E_S) \), then it is also true that \( Pr(\Psi_t \in E) \uparrow Pr(\Psi \in E_S) \).

From (3.4), we have

\[
Pr(\Psi^* \in E) = \lim_{t \to \infty} \frac{1}{t} \int_0^t P(\Psi_s \in E) \, ds \leq \lim_{t \to \infty} \frac{1}{t} \int_0^t P(\Psi_t \in E) \, ds = \lim_{t \to \infty} P(\Psi_t \in E) = P(\Psi \in E_S) = p_S^\Psi.
\]

On the other hand, from \( P(\Psi_t \in E) \uparrow P(\Psi \in E_S) \), we have that for all \( \epsilon > 0 \), there exists \( K \) such that for all \( s \geq K \), \( P(\Psi_s \in E) \geq p_S^\Psi - \epsilon \). Then

\[
Pr(\Psi^* \in E) = \lim_{t \to \infty} \frac{1}{t} \int_0^t P(\Psi_s \in E) \, ds \geq \lim_{t \to \infty} \frac{1}{t} \int_K^t (p_S^\Psi - \epsilon) \, ds = p_S^\Psi - \epsilon.
\]

Let \( \epsilon \to 0 \), we have \( Pr(\Psi^* \in E) \geq p_S^\Psi \). Therefore, \( Pr(\Psi^* \in E) = p_S^\Psi \). Similarly, we can prove that \( Pr(\Psi^* \in \{(T, \emptyset)\}) = p_T^\Psi \).

(2) For all \( E \in \mathcal{B}(H^{(m)}) \) with \( m \in \mathbb{Z}_+ \setminus \{0\} \), since \( E \cap (H^{(\infty)} \cup \{(S, \emptyset), (T, \emptyset)\}) = \emptyset \), we have \( Pr(\Psi^* \in E) \leq 1 - Pr(\Psi^* \in H^{(\infty)}) - p_S^\Psi - p_T^\Psi \). Thus, to prove \( Pr(\Psi^* \in E) = 0 \), it is enough to show that \( Pr(\Psi^* \in H^{(\infty)}) = p_\infty^\Psi \), since we know that \( p_S^\Psi + p_T^\Psi + p_\infty^\Psi = 1 \).
To prove $Pr(\Psi^* \in H^{(\infty)}) = p_\Psi^\infty$, note that $\{\Psi_s \in H^{(\infty)}\} = \{\Psi \in H^{(\infty)}\}$, i.e., a failure detector history has an infinite number of transitions from time $s$ on if and only if itself has an infinite number of transitions. Then we have

$$Pr(\Psi^* \in H^{(\infty)}) = \lim_{t \to \infty} \frac{1}{t} \int_0^t P(\Psi_s \in H^{(\infty)}) \, ds$$

$$= \lim_{t \to \infty} \frac{1}{t} \int_0^t P(\Psi \in H^{(\infty)}) \, ds = P(\Psi \in H^{(\infty)}) = p_\Psi^\infty.$$ 

(3) and (4) have similar proofs as those of (1) and (2). \hfill \Box

We now look at failure detector histories with an infinite number of transitions. We know that with probability $p_\Psi^\infty$, $\Psi$ has an infinite number of transitions. If $p_\Psi^\infty = 0$, then $p_S^\infty + p_T^\infty = 1$, and in the stationary versions of $\Psi$, only trivial histories that never change the failure detector outputs have a nonzero probability.

We now consider the case when $p_\Psi^\infty > 0$. In this case, we restrict $\Psi$ onto $H^{(\infty)}$. More precisely, we first define the restricted probability space $(\Omega_\infty^\Psi, F_\infty^\Psi, P_\infty^\Psi)$ such that (1) $\Omega_\infty^\Psi = \Psi^{-1}(H^{(\infty)})$, (2) $F_\infty^\Psi = \{B : B \in F, B \subseteq \Omega_\infty^\Psi\}$, and (3) $P_\infty^\Psi(B) = P(B)/p_\Psi^\infty$ for all $B \in F_\infty^\Psi$. We then define the restricted random failure detector history $\Psi_\infty$ as the measurable mapping from $\Omega_\infty^\Psi$ to $H^{(\infty)}$ such that $\Psi_\infty(\omega) = \Psi(\omega)$ for all $\omega \in \Omega_\infty^\Psi$. Since a failure detector history in $H^{(\infty)}$ is also a simple marked point process, $\Psi_\infty$ is also a random marked point process as defined in [Sig95].

$\Psi_\infty$, as a random marked point process, has its own event stationary version $\Psi_\infty^0$ and time stationary version $\Psi_\infty^*$ (see definitions in Appendix A). The following lemma gives the relation between the distributions of $\Psi_\infty^0$, $\Psi_\infty^*$ and $\Psi_\infty^0$, $\Psi_\infty^*$.

**Lemma 3.4** If $p_\Psi^\infty > 0$, then for all $E \in B(H^{(\infty)})$, $Pr(\Psi_\infty^0 \in E) = p_\Psi^\infty * Pr(\Psi_\infty^0 \in E)$, and $Pr(\Psi_\infty^* \in E) = p_\Psi^\infty * Pr(\Psi_\infty^* \in E)$.
Proof. Direct from (3.3), (3.4), (A.5), (A.6) and the definition of the probability measure $P^\infty$.

The Steady State Behavior after $p$ Crashes

We now define a random failure detector history that represents the steady state behavior of the failure detector after $p$ crashes. Formally, a post-crash version $\Psi^c$ of failure detector $D$ is a random failure detector history defined by the following distribution (assuming it exists):

$$Pr(\Psi^c \in \mathcal{E}) \overset{\text{def}}{=} \lim_{t \to \infty} \frac{1}{t} \int_0^t Pr(D(s) \in \mathcal{E}) ds, \text{ for all } \mathcal{E} \in \mathcal{B}(H).$$

(3.5)

Intuitively, (3.5) means that if we randomly pick a time $s$ at which $p$ crashes and then observe the behavior of the failure detector after time $s$, the random failure detector history we observed is given by the post-crash version $\Psi^c$. So, similarly to $\Psi^0$ and $\Psi^*$, we say that $\Psi^c$ is the version of the failure detector $D$ when we randomly observe $D$ way out at a time when $p$ crashes. $\Psi^c$ is obtained by averaging the distribution of $D(s)$, the post-crash behavior of the failure detector $D$, over all crash times. Thus it is also an empirical distribution. We say that the failure detector $D$ has steady state behavior after $p$ crashes if the distribution defined in (3.5) exists.

One primary metric, the detection time, is defined with respect to the steady state behavior after $p$ crashes, i.e., with respect to the post-crash version $\Psi^c$.

Non-Futuristic Property

Before $p$ crashes, no failure detector implementation can tell whether $p$ will crash later or not, i.e., the failure detector cannot predict the future. Therefore, the behavior
of the failure detector up to any time $t$ at which $p$ is still up should be the same as
the behavior of the failure detector in the same period in failure-free runs. We now
formalize this idea.

For all $m \in \mathbb{Z}^+$ and for all $t \in \mathbb{R}^+$, let $H_{\leq t}^{(m)}$ be the subset of $H^{(m)}$ such that
the time of the last transition of any failure detector history in $H_{\leq t}^{(m)}$ is at most $t$,
i.e., $H_{\leq t}^{(m)} \overset{\text{def}}{=} \{ \psi \in H^{(m)} : t_{m-1} \leq t \}$. So $H_{\leq t}^{(m)}$ gives the set of failure detector history
prefixes up to time $t$ that contains exactly $m$ transitions. Clearly $H_{\leq t}^{(m)} \in \mathcal{B}(H^{(m)})$,
and so we can define the Borel $\sigma$-field $\mathcal{B}(H_{\leq t}^{(m)}) = \{ \mathcal{E} : \mathcal{E} \in \mathcal{B}(H^{(m)}), \mathcal{E} \subseteq H_{\leq t}^{(m)} \}$. Let
$H_{\leq t} = \bigcup_{m \in \mathbb{Z}^+} H_{\leq t}^{(m)}$. $H_{\leq t}$ contains all failure detector history prefixes up to time $t$.
The Borel $\sigma$-field of $H_{\leq t}$ is defined as $\mathcal{B}(H_{\leq t}) = \{ \bigcup_{m \in \mathbb{Z}^+} \mathcal{E}_m : \mathcal{E}_m \in \mathcal{B}(H_{\leq t}^{(m)}) \}$. We
define a prefix mapping $f_{\leq t} : H \to H_{\leq t}$, such that for any failure detector history
$\psi \in H$, $f_{\leq t}(\psi)$ is the failure detector history prefix that only contains the transitions
of $\psi$ up to time $t$. It is easy to verify that $f_{\leq t}$ is a measurable mapping.

We now formally define the probabilistic behavior of a failure detector history up
to some time $t$. Given a random failure detector history $\Psi : \Omega \to H$, the random
failure detector history prefix up to time $t$ is the measurable mapping $\Psi_{\leq t} : \Omega \to H_{\leq t}$
such that $\Psi_{\leq t} = f_{\leq t} \circ \Psi$.

We say that a failure detector $\mathcal{D}$ is non-futuristic (or not predicting the future)
if for all $t \in \mathbb{R}^+$, for all $t_1, t_2 \in \mathbb{R}^\infty$ and $t_1, t_2 > t$, $\mathcal{D}(t_1)_{\leq t}$ and $\mathcal{D}(t_2)_{\leq t}$ have
the same distribution, i.e., for all $\mathcal{E} \in \mathcal{B}(H_{\leq t})$, $Pr(\mathcal{D}(t_1)_{\leq t} \in \mathcal{E}) = Pr(\mathcal{D}(t_2)_{\leq t} \in \mathcal{E})$.
Intuitively, this means that as long as $p$ has not crashed yet by time $t$, the probabilistic
behavior of the failure detector up to time $t$ is the same no matter whether or when
$p$ may crash later. In other words, the failure detector does not provide hints on
whether or when process $p$ will crash in the future.
3.3 Failure Detector Specification Metrics

With the formal model of the failure detector given in the previous section, we are now ready to formally define the QoS metrics of the failure detector introduced in Chapter 2.

Let $\mathcal{D}$ be a failure detector. Let $R^\infty_+ \overset{\text{def}}{=} \mathbb{R}_+ \cup \{\infty\}$.

3.3.1 Definitions of Metrics

**Detection time** $(T_D)$: $T_D$ is defined from the post-crash version $\Psi^c$ of $\mathcal{D}$. Suppose $\Psi^c : \Omega^c \to \mathbb{H}$, with $(\Omega^c, \mathcal{F}^c, P^c)$ as the underlying probability space. We first define a measurable mapping $f_D : \mathbb{H} \to R^\infty_+$ such that for any failure detector history $\psi \in \mathbb{H}$, $f_D(\psi)$ is: (a) 0, if $\psi$ has no transition and the output is always $S$; or (b) the time of the last transition, if $\psi$ has a finite number of transitions and the output after the last transition is always $S$; or (c) $\infty$ otherwise. Then $T_D : \Omega^c \to R^\infty_+$ is the random variable such that $T_D = f_D \circ \Psi^c$. That is, given any particular post-crash history $\psi$, $T_D = f_D(\psi)$ is the time elapsed from the time of crash to the time when the failure detector starts suspecting $p$ permanently, and the distribution of $T_D$ is determined by the distribution of $\Psi^c$, which is defined in (3.5).

All accuracy metrics are defined with respect to the steady state behavior in failure-free runs, i.e., with respect to the stationary versions $\Psi^0$ and $\Psi^*$ of the random failure detector history $\Psi^\overset{\text{def}}{=} \mathcal{D}(\infty)$. For the convenience of studying the relations between the accuracy metrics in the next section, we assume that $\Psi$, $\Psi^0$ and $\Psi^*$ use the same underlying probability space $(\Omega, \mathcal{F}, P)$ (one can always construct some common space supporting all of them).
The following three accuracy metrics are defined in terms of the event stationary version \( \Psi^0 \). Recall from Section 3.2.1 that \( T_0 \) and \( T_1 \) are defined as the zeroth and the first intertransition time of a given failure detector history \( \psi \). Thus \( T_0 \) and \( T_1 \) are actually measurable mappings from \( H \) to \( \mathbb{R}^\infty_+ \).

**Mistake recurrence time** \( (T_{MR}) \): We define a measurable mapping \( f_{MR} : H \rightarrow \mathbb{R}^\infty_+ \) such that \( f_{MR} = T_0 + T_1 \). Then \( T_{MR} : \Omega \rightarrow \mathbb{R}^\infty_+ \) is the random variable such that \( T_{MR} = f_{MR} \circ \Psi^0 \). Intuitively, \( T_{MR} \) is the length of the first two consecutive periods, one trust period and one suspicion period, of \( \Psi^0 \). We call any two consecutive periods a recurrence interval. Since \( \Psi^0 \) is event stationary, the distribution of the length of any recurrence interval is the same, and thus we only take the first recurrence interval of \( \Psi^0 \). \( T_{MR} \) represents the length of the recurrence interval when we randomly observe the failure detector output way out at a transition in some failure-free run, and its distribution is determined by the distribution of \( \Psi^0 \), which is defined in (3.3). Note that when defining \( T_{MR} \) we do not restrict the recurrence interval to be started and ended with S-transitions. This is because in steady state whether you observe at an S-transition or a T-transition does not change the distribution of the length of the recurrence interval, and so we choose not to make this restriction for convenience.

**Mistake duration** \( (T_M) \): we define a measurable mapping \( f_M : H \rightarrow \mathbb{R}^\infty_+ \) such that for any failure detector history \( \psi = \langle k, \{t_n\} \rangle \in H \), \( f_M(\psi) = T_0(\psi) \) if \( k = S \), and \( f_M(\psi) = T_1(\psi) \) if \( k = T \). Then \( T_M : \Omega \rightarrow \mathbb{R}^\infty_+ \) is the random variable such that \( T_M = f_M \circ \Psi^0 \). Recall that after being shifted by a transition time, any failure detector history has a transition at origin (i.e., \( t_0 = 0 \)) except the histories with no transitions at all. Thus the definition of \( f_M \) guarantees that it always takes the length of the first suspicion (mistake) period from the event stationary version \( \Psi^0 \).
Therefore, $T_m$ represents the length of the mistake period when we randomly observe
the failure detector output way out at an S-transition in some failure-free run, and
its distribution is determined by the distribution of $\Psi^0$.

**Good period duration ($T_g$):** The definition of $T_g$ is symmetric to that of $T_m$.
we define a measurable mapping $f_g : H \rightarrow \mathbb{R}_+^\infty$ such that for any failure detector
history $\psi = \langle k, \{t_n\} \rangle \in H$, $f_g(\psi) = T_0(\psi)$ if $k = T$, and $f_g(\psi) = T_1(\psi)$ if $k = S$.
Then $T_g : \Omega \rightarrow \mathbb{R}_+^\infty$ is the random variable such that $T_g = f_g \circ \Psi^0$. Intuitively,
$T_g$ represents the length of the trust (good) period when we randomly observe the
failure detector output way out at a T-transition in some failure-free run, and its
distribution is determined by the distribution of $\Psi^0$.

The following three accuracy metrics are defined in terms of the time stationary
version $\Psi^*.$

**Query accuracy probability ($P_a$):** Let $B_T$ be the set of failure detector histo-
ries with output $T$ at time 0. Then $P_a \overset{\text{def}}{=} P(\Psi^* \in B_T)$. Intuitively, when we randomly
observe the failure detector output way out at a time $t$ in some failure-free run, the
probability that the output at time $t$ is $T$ is just the probability that the output is
$T$ at time 0 of the time stationary version $\Psi^*$. Therefore, $P_a$ is the probability that,
when queried at a random time in some failure-free run, the output of the failure
detector is $T$ (and thus is correct).

**Average mistake rate ($\lambda_m$):** We define a measurable mapping $N_S : H \rightarrow \mathbb{R}_+$
such that for any failure detector history $\psi$, $N_S(\psi)$ is the number of S-transitions
in the period $[0, 1]$. Thus $N_S \circ \Psi^*$ is a random variable representing the number of
S-transitions of $\Psi^*$ in the unit interval $[0, 1]$. Since $\Psi^*$ is time stationary, $N_S \circ \Psi^*$
is the number of S-transitions in any unit interval when we randomly observe the
failure detector output way out in time in some failure-free run. Then \( \lambda_M \) is defined as \( E(N_S \circ \Psi^*) \), the expected value of \( N_S \circ \Psi^* \).

**Forward good period duration** \((T_{fg})\): Roughly speaking, we define \( T_{fg} \) as the time from the origin to the first transition of \( \Psi^* \), conditioned on the fact that the output of \( \Psi^* \) at the origin is \( T \). Since \( \Psi^* \) is obtained when we randomly observe the failure detector output way out in time in failure-free runs, \( T_{fg} \) represents the time elapsed from a random time at which \( q \) trusts \( p \) to the time of the next \( S \)-transition. We now formally define \( T_{fg} \). If \( P_A = 0 \), then let \( T_{fg} \equiv 0 \), i.e., if the probability that \( q \) trusts \( p \) at a random time is 0, then \( T_{fg} \) is always 0. If \( P_A > 0 \), then we define a random failure detector history \( \Psi_T^* \) obtained by restricting \( \Psi^* \) onto \( B_T \). More precisely, we first define a restricted probability space \((\Omega_T, \mathcal{F}_T, P_T)\) such that (1) \( \Omega_T = \{ \Psi^* \in B_T \} \), (2) \( \mathcal{F}_T = \{ B : B \in \mathcal{F}, B \subseteq \Omega_T \} \), and (3) \( P_T(B) = P(B)/P_A \) for all \( B \in \mathcal{F}_T \). Then \( \Psi_T^* : \Omega_T \to B_T \) is the random failure detector history such that \( \Psi_T^*(\omega) = \Psi^*(\omega) \) for all \( \omega \in \Omega_T \). Intuitively, \( \Psi_T^* \) is the version of \( \Psi^* \) conditioned on the fact that the output at the origin is \( T \). Let \( f_{fg} : B_T \to R_+^\infty \) be the measurable mapping such that for all \( \psi = \langle k, \{ t_n \} \rangle \in B_T \), if \( \psi \) has at least one transition then \( f_{fg}(\psi) = t_0 \), else \( f_{fg}(\psi) = \infty \), i.e., \( f_{fg}(\psi) \) is the time from the origin to the zeroth transition of \( \psi \). Then \( T_{fg} : \Omega_T \to R_+^\infty \) is the random variable such that \( T_{fg} = f_{fg} \circ \Psi_T^* \).

We now give an example that is helpful for understanding the above definitions. It shows how these definitions are linked with the steady state distributions defined in Section 3.2.3, and how they match with the intuition. **Example 1.** Given a failure detector \( D \), suppose we want to know the probability that its mistake recurrence time is at least \( x \), i.e. \( P_r(T_{mr} \geq x) \), for some \( x \in R_+ \). Let
\[ E \overset{\text{def}}{=} \{ \psi \in H : T_0(\psi) + T_1(\psi) \geq x \} \], i.e., the set of failure detector histories in which the length of the very first recurrence interval is at least \( x \). Let \( \Psi \overset{\text{def}}{=} D(\infty) \) be the random failure detector history in failure-free runs, and let \( \Psi^0 \) be the event stationary version of \( \Psi \). By the definition of \( T_{\text{MR}} \), we have \( \Pr(T_{\text{MR}} \geq x) = \Pr(f_{\text{MR}} \circ \Psi^0 \geq x) = \Pr(\Psi^0 \in E) \). From (3.3), we have

\[
\begin{align*}
\Pr(T_{MR} \geq x) &= \lim_{n \to \infty} \frac{1}{n} \sum_{j=0}^{n-1} \Pr(\Psi(j) \in E) \\
&= \frac{1}{n} \sum_{j=0}^{n-1} \Pr(\Psi(j) \in E).
\end{align*}
\]

(3.6)

Note that \( \Pr(\Psi(j) \in E) \) is the probability that the length of the \( j \)-th recurrence interval is at least \( x \). Thus \( \Pr(T_{MR} \geq x) \) is obtained by averaging these probabilities over the first \( n \) recurrence intervals, and then taking the limit as \( n \) goes to infinity.

Equality (3.6) corresponds to what we would do if we want to obtain an estimate on \( \Pr(T_{MR} \geq x) \) by experiments. We would run the failure detector a number of times such that each run contains a large number of recurrence intervals. We then compute the ratio of the number of recurrence intervals that are at least \( x \) time units long over the total number of recurrence intervals, and use this ratio as the estimate of \( \Pr(T_{MR} \geq x) \). This ratio can be equivalently obtained by computing such ratios for the zeroth, first, second ... recurrence intervals, and then averaging these ratios. This matches the intuitive idea behind equality (3.6).

\[ \Box \]

### 3.3.2 Relations between Accuracy Metrics

We now analyze the relations between the accuracy metrics defined in the previous section. The analysis is based on the results in the theory of marked point processes, such as Birkhoff’s Ergodic Theorem for marked point processes, and the empirical inversion formulas.
Let $\Psi \overset{\text{def}}{=} \mathcal{D}(\infty)$ be the random failure detector history of some failure detector $\mathcal{D}$ under the failure-free pattern, and suppose that $\Psi$ has the event stationary version $\Psi^0$ and the time stationary version $\Psi^*$. Suppose that the underlying probability space for $\Psi$, $\Psi^0$ and $\Psi^*$ are $(\Omega, \mathcal{F}, P)$.

**Lemma 3.5** $T_{MR} = T_M + T_G$.

**Proof.** This is immediate from the fact that $f_{MR} = f_M + f_G$, where $f_{MR}$, $f_M$ and $f_G$ are measurable mappings used to define $T_{MR}$, $T_M$ and $T_G$ respectively. \(\square\)

Henceforth, we only consider the nondegenerated case in which $0 < E(T_{MR}) < \infty$. Intuitively, this means that the average time for a failure detector to make the next mistake is finite and nonzero.

**Proposition 3.6** If $E(T_{MR}) < \infty$, then $p_S^\Psi = 0$ and $p_\infty^\Psi = 1$.

**Proof.** Let $\mathcal{E} \overset{\text{def}}{=} \{\langle S, \emptyset \rangle, \langle S, \{0\} \rangle\}$. By Lemma 3.3 (3), $Pr(\Psi^0 \in \mathcal{E}) = p_S^\Psi$. For all $\omega \in \Omega$ such that $\Psi^0(\omega) \in \mathcal{E}$, $T_0(\Psi^0(\omega)) = \infty$, and thus $\{\omega \in \Omega : \Psi^0(\omega) \in \mathcal{E}\} \subseteq \{\omega \in \Omega : T_{MR}(\omega) = \infty\}$. Therefore $Pr(T_{MR} = \infty) \geq Pr(\Psi^0 \in \mathcal{E}) = p_S^\Psi$. If $p_S^\Psi > 0$, then $E(T_{MR}) = \infty$, which contradicts the assumption that $E(T_{MR}) < \infty$. So $p_S^\Psi = 0$. Similarly we have $p_I^\Psi = 0$. By Proposition 3.2, we have $p_\infty^\Psi = 1$. \(\square\)

From this proposition and Lemma 3.3, we know that the probability that the stationary version $\Psi^0$ or $\Psi^*$ has a finite number of transitions is zero. Formally,

**Corollary 3.7** Let $\mathcal{E} \overset{\text{def}}{=} \mathcal{H} \setminus \mathcal{H}(\infty)$. If $E(T_{MR}) < \infty$, then $Pr(\Psi^0 \in \mathcal{E}) = Pr(\Psi^* \in \mathcal{E}) = 0$.

Henceforth, we treat $\Psi^0$ and $\Psi^*$ as mappings from $\Omega$ to $\mathcal{H}(\infty)$, since $\{\Psi^0 \in \mathcal{H} \setminus \mathcal{H}(\infty)\}$ and $\{\Psi^* \in \mathcal{H} \setminus \mathcal{H}(\infty)\}$ have measure zero. In this case, $\Psi^0$ and $\Psi^*$ are just
simple random marked point processes, and so results from the theory of random marked point processes can be applied to $\Psi^0$ and $\Psi^*$ directly.

Let $\mathcal{I}$ be the invariant $\sigma$-field of $H^{(\infty)}$ (see Appendix A for the definition). Let $E_{\mathcal{I}}(X)$ denote the conditional expected value of $X$ given the $\sigma$-field $\mathcal{I}$ (see [Bil95] p.445 for a definition of the conditional expected value given a $\sigma$-field).

**Proposition 3.8** $E_{\mathcal{I}}(T_{MR}) = 2E_{\mathcal{I}}(T_0 \circ \Psi^0)$ a.s.

**Proof.** by definition, $E_{\mathcal{I}}(T_{MR}) = E_{\mathcal{I}}(f_{MR} \circ \Psi^0) = E_{\mathcal{I}}((T_0 + T_1) \circ \Psi^0)$. Since $E_{\mathcal{I}}((T_0 + T_1) \circ \Psi^0) = E_{\mathcal{I}}(T_0 \circ \Psi^0) + E_{\mathcal{I}}(T_1 \circ \Psi^0)$ a.s., it is enough to show that $E_{\mathcal{I}}(T_1 \circ \Psi^0) = E_{\mathcal{I}}(T_0 \circ \Psi^0)$ a.s. By (A.9) of Theorem A.5, we have

$$E_{\mathcal{I}}(T_1 \circ \Psi^0) = \lim_{n \to \infty} \frac{1}{n} \sum_{j=0}^{n-1} T_1 \circ \Psi(j) = \lim_{n \to \infty} \frac{1}{n} \sum_{j=1}^{n} T_0 \circ \Psi(j)$$

$$= \lim_{n \to \infty} \frac{n+1}{n} \cdot \frac{1}{n+1} \left[ \sum_{j=0}^{n} T_0 \circ \Psi(j) - T_0 \circ \Psi(0) \right]$$

$$= \lim_{n \to \infty} \frac{1}{n+1} \left[ \sum_{j=0}^{n} T_0 \circ \Psi(j) \right] = E_{\mathcal{I}}(T_0 \circ \Psi^0) \text{ a.s.}$$

We say that $\Psi$ is **ergodic** if $\Psi^0$ (or equivalently $\Psi^*$) is ergodic (see Appendix A for the definition). Informally, in this case we also say that the distribution of failure detector histories in failure-free runs is ergodic, or simply the failure detector is ergodic.

**Lemma 3.9**

$$\lambda_M = E\left[ \frac{1}{E_{\mathcal{I}}(T_{MR})} \right]. \quad (3.7)$$

If $\Psi$ is ergodic, then

$$\lambda_M = \frac{1}{E(T_{MR})}. \quad (3.8)$$
**Proof.** Let \(\lambda \overset{\text{def}}{=} E(N_1 \circ \Psi^*)\) be the arrival rate of \(\Psi\). From (A.16) and (A.15), we have \(\lambda = E(E_T(N_1 \circ \Psi^*))\) and \(E_T(N_1 \circ \Psi^*) = \lim_{t \to \infty} \frac{N_t}{t} \overset{\text{a.s.}}{=} \). Similarly, we can have \(\lambda_m = E(E_T(N_S \circ \Psi^*))\) and \(E_T(N_S \circ \Psi^*) = \lim_{t \to \infty} \frac{N_t}{t} \overset{\text{a.s.}}{=} \), where \(N_t^S : H \to R_+\) is a measurable mapping representing the number of S-transitions in the period \((0,t]\).

Since in any period \((0,t]\), the numbers of S-transitions and T-transitions differ at most by one, we have for any \(\psi \in H\), \(2N_t^S(\psi) - 1 \leq N_t(\psi) \leq 2N_t^S(\psi) + 1\). Thus \(\lim_{t \to \infty} \frac{N_t}{t} = \lim_{t \to \infty} \frac{N_t^S}{t} \overset{\text{a.s.}}{=} \) and so \(\lambda = 2\lambda_m\). By (A.16) and (A.15), we know that \(\lambda_m = \frac{1}{2} E(T^T(0 \circ \Psi^0)) = E(T^T(\{T_{br}\})^{-1}).\) If \(\Psi\) is ergodic, then by Proposition A.4, we know that \(\lambda_m = E(T_{br})^{-1}.\)

Recall that \(B_T\) is the set of failure detector histories with output \(T\) at time 0. Let \(B_S\) be the set of failure detector histories with output \(S\) at time 0. Let \(A_T \overset{\text{def}}{=} \{\omega \in \Omega : \Psi^0(\omega) \in B_T\}\) and \(A_S \overset{\text{def}}{=} \{\omega \in \Omega : \Psi^0(\omega) \in B_S\}\). Let \(X_T : \Omega \to R_+\) be the random variable such that \(X_T(\omega) = T_0(\Psi^0(\omega))\) for all \(\omega \in A_T\), and \(X_T(\omega) = 0\) for all \(\omega \in A_S\). Let \(X_S : \Omega \to R_+\) be the random variable such that \(X_S(\omega) = T_0(\Psi^0(\omega))\) for all \(\omega \in A_S\), and \(X_S(\omega) = 0\) for all \(\omega \in A_T\).

**Proposition 3.10** \(E_T(T_G) = 2E_T(X_T) \overset{\text{a.s.}}{=} \), and \(E_T(T_M) = 2E_T(X_S) \overset{\text{a.s.}}{=} \).

**Proof.** Define random variable \(X'_T : \Omega \to R_+\) such that \(X'_T(\omega) = 0\) for all \(\omega \in A_T\), and \(X'_T(\omega) = T_1(\Psi^0(\omega))\) for all \(\omega \in A_S\). Then by definition \(T_G = X_T + X'_T\). Thus to prove \(E_T(T_G) = 2E_T(X_T) \overset{\text{a.s.}}{=} \), it is enough to show that \(E_T(X_T) = E_T(X'_T) \overset{\text{a.s.}}{=} \). Let \(f : H^{(\infty)} \to R_+\) be the measurable mapping such that for all \(\psi = \langle k, \{t_n\} \rangle\), \(f(\psi) = T_0(\psi)\) if \(k = T\), and \(f(\psi) = 0\) if \(k = S\). Similarly let \(f' : H^{(\infty)} \to R_+\) be the measurable mapping such that for all \(\psi = \langle k, \{t_n\} \rangle\), \(f'(\psi) = 0\) if \(k = T\), and \(f'(\psi) = T_1(\psi)\) if \(k = S\). Thus \(X_T = f \circ \Psi^0\) and \(X'_T = f' \circ \Psi^0\). It is important to note
that \( f' \circ \Psi_j = f \circ \Psi_{j+1} \) for all \( j \geq 0 \). Using equality (A.9), we have

\[
E_I(X_T') = E_I(f' \circ \Psi^0) = \lim_{n \to \infty} \frac{1}{n} \sum_{j=0}^{n-1} f' \circ \Psi_j
= \lim_{n \to \infty} \frac{1}{n} \sum_{j=1}^{n} f \circ \Psi_j
= \lim_{n \to \infty} \frac{n+1}{n} \cdot \frac{1}{n+1} \left[ \sum_{j=0}^{n} f \circ \Psi_j - f \circ \Psi_0 \right]
= E_I(f \circ \Psi^0) = E_I(X_T) \text{ a.s.}
\]

We thus have \( E_I(T_G) = 2E_I(X_T) \text{ a.s.} \). Similarly we can prove that \( E_I(T_M) = 2E_I(X_S) \text{ a.s.} \).

\[\square\]

**Lemma 3.11** If \( 0 < E_I(T_{MR}) < \infty \), a.s., then

\[
P_A = E \left[ \frac{E_I(T_G)}{E_I(T_{MR})} \right]. \quad (3.9)
\]

If \( \Psi \) is ergodic and \( 0 < E(T_{MR}) < \infty \), then

\[
P_A = \frac{E(T_G)}{E(T_{MR})}. \quad (3.10)
\]

**Proof.** By definition, \( P_A \overset{\text{def}}{=} Pr(\Psi^* \in B_T) \). Since \( 0 < E_I(T_{MR}) < \infty \text{ a.s.} \), by Proposition 3.8, we know that \( 0 < E_I(T_0 \circ \Psi^0) < \infty \text{ a.s.} \). Then by the empirical inversion formula (A.19) we have

\[
Pr(\Psi^* \in B_T) = E \left[ \frac{E_I \left[ \int_{0}^{T_0(\Psi^0)} I_{B_T} \circ \Psi^0 \, ds \right]}{E_I(T_0 \circ \Psi^0)} \right]. \quad (3.11)
\]

We claim that \( X_T = \int_{0}^{T_0 \circ \Psi^0} I_{B_T} \circ \Psi^0 \, ds \text{ a.s.} \). In fact, from Proposition A.1, we know that with probability one \( \Psi^0 \) has a transition at time 0, i.e. \( Pr(t_0 \circ \Psi^0 = 0) = 1 \). Then with probability one, during the entire period \((0, T_0(\Psi^0(\omega)))\), the output of \( \Psi^0 \) is the same as the output of \( \Psi^0 \) at the origin. In other words, with probability one, if \( \Psi^0(\omega) \in B_T \), then \( I_{B_T}(\Psi^0(\omega)) = 1 \) for all \( s \in (0, T_0(\Psi^0(\omega))) \); if \( \Psi^0(\omega) \in B_S \),
then $I_{B_T}(\Psi_0(\omega)) = 0$ for all $s \in (0, T_0(\Psi_0(\omega)))$. Therefore, with probability one, 
\[ \int_0^{T_0 \Psi_0} I_{B_T} \circ \Psi_0^0 ds = T_0 \circ \Psi_0^0 if \Psi_0(\omega) \in B_T, and \int_0^{T_0 \Psi_0} I_{B_T} \circ \Psi_0^0 ds = 0 if \Psi_0(\omega) \in B_S. \]
Thus $X_T = \int_0^{T_0 \Psi_0} I_{B_T} \circ \Psi_0^0 ds a.s.$

By Proposition 3.10, we have 
\[ E(I_{T_G}) = 2E(I_{X_T}) = 2E(I_{\int_0^{T_0} I_{B_T} \circ \Psi_0^0 ds}) a.s. \]
By Proposition 3.8, we have 
\[ E(I_{T_MR}) = 2E(I_{T_0 \circ \Psi_0^0}) a.s. \]
Therefore, from (3.11) we have 
\[ P_A = E\left[ \frac{E(I_{T_G})}{E(I_{T_MR})} \right]. \]

If $\Psi$ is ergodic, then By Proposition A.4, we have 
\[ P_A = E(I_{T_G}) E(I_{T_MR}) a.s. \]

By definition, if $P_A = 0$, then $T_{FG} \equiv 0$. We now study the case $P_A > 0$.

**Lemma 3.12** If $P_A > 0$ and $0 < E(I_{T_MR}) < \infty$ a.s., then for all $x \in \mathbb{R}_+$,
\[ Pr(T_{FG} \leq x) = \frac{1}{P_A} E\left[ \frac{E(I_{min(T_G, x)})}{E(I_{T_MR})} \right]. \quad (3.12) \]

**Proof.** Let $f : H^{(\infty)} \to \mathbb{R}_+$ be the measurable mapping such that for all $\psi \in H^{(\infty)}$, 
\[ f(\psi) = f_{FG}(\psi) if \psi \in B_T, and f(\psi) = 0 if \psi \in B_S. \]
Let $Y \overset{def}{=} f \circ \Psi^*$. Thus under the condition \{\Psi^* \in B_T\}, $Y = T_{FG}$, and under the condition \{\Psi^* \in B_S\}, $Y = 0$. For all $x \in \mathbb{R}_+$, we have 
\[ Pr(Y \leq x) = Pr(Y \leq x \mid \{\Psi^* \in B_T\}) Pr(\Psi^* \in B_T) + 
Pr(Y \leq x \mid \{\Psi^* \in B_S\}) Pr(\Psi^* \in B_S) 
= Pr(T_{FG} \leq x) P_A + (1 - P_A). \]
Thus if $P_A > 0$, we have for all $x \in \mathbb{R}_+$,
\[ Pr(T_{FG} \leq x) = \frac{Pr(Y \leq x) - (1 - P_A)}{P_A}. \quad (3.13) \]
Let $\mathcal{E} = \{ \psi = (k, \{ t_n \}) \in \mathbf{H}^{(\infty)} : k = S, \text{ or } k = T \text{ and } t_0 \leq x \}$. Then $\{ Y \leq x \} = \{ f \circ \Psi^* \leq x \} = \{ \Psi^* \in \mathcal{E} \}$. Since $0 < E_T(T_{mn}) < \infty$ a.s., by Proposition 3.8, we know that $0 < E_T(T_0 \circ \Psi^0) < \infty$ a.s. Then by the empirical inversion formula (A.19) we have

$$Pr(Y \leq x) = Pr(\Psi^* \in \mathcal{E}) = E \left[ \frac{E_T \left( \int_0^{T_0 \circ \Psi^0} I_\mathcal{E} \circ \Psi_s^0 \, ds \right)}{E_T(T_0 \circ \Psi^0)} \right].$$

(3.14)

Let $X' : \Omega \to \mathbb{R}_+$ be the random variable such that for all $\omega \in A_T$, $X'(\omega) = \min(x, T_0(\Psi^0(\omega)))$, and for all $\omega \in A_S$, $X'(\omega) = 0$. We claim that $X_S + X' = \int_0^{T_0 \circ \Psi^0} I_\mathcal{E} \circ \Psi_s^0 \, ds$ a.s. In fact, from Proposition A.1, we know that with probability one $\Psi^0$ has a transition at time 0, i.e. $Pr(t_0 \circ \Psi^0 = 0) = 1$. Thus, with probability one, if $\omega \in A_S$ then $I_\mathcal{E}(\Psi_s^0(\omega)) = 1$ for all $s \in (0, T_0(\Psi^0(\omega)))$. So we have that with probability one, if $\omega \in A_S$ then $\int_0^{T_0(\Psi^0(\omega))} I_\mathcal{E}(\Psi_s^0(\omega)) \, ds = T_0(\Psi^0(\omega))$.

If $\omega \in A_T$, then $I_\mathcal{E}(\Psi_s^0(\omega)) = 1$ iff $\Psi_s^0(\omega) \in \mathcal{E}$, which means that starting from time $s$ at which the output is $T$, the time to the next transition in $\Psi^0(\omega)$ is at most $x$. So, with probability one, if $\omega \in A_T$, then $I_\mathcal{E}(\Psi_s^0(\omega)) = 1$ iff $T_0(\Psi^0(\omega)) - s \leq x$. There are two possible cases: (a) $T_0(\Psi^0(\omega)) \leq x$, in which case $I_\mathcal{E}(\Psi_s^0(\omega)) = 1$ for all $s \in (0, T_0(\Psi^0(\omega)))$, and so $\int_0^{T_0(\Psi^0(\omega))} I_\mathcal{E}(\Psi_s^0(\omega)) \, ds = T_0(\Psi^0(\omega))$; or (b) $T_0(\Psi^0(\omega)) > x$, in which case for all $s \in (0, T_0(\Psi^0(\omega)) - x)$, $I_\mathcal{E}(\Psi_s^0(\omega)) = 0$, and for all $s \in [T_0(\Psi^0(\omega)) - x, T_0(\Psi^0(\omega)))$, $I_\mathcal{E}(\Psi_s^0(\omega)) = 1$, and so $\int_0^{T_0(\Psi^0(\omega))} I_\mathcal{E}(\Psi_s^0(\omega)) \, ds = \int_{T_0(\Psi^0(\omega)) - x}^{T_0(\Psi^0(\omega))} 1 \, ds = x$.

Combining the cases (a) and (b), we have with probability one, if $\omega \in A_T$, then

$$\int_0^{T_0 \circ \Psi^0} I_\mathcal{E} \circ \Psi_s^0 \, ds = \min(x, T_0(\Psi^0(\omega))).$$

From the above separate cases for $\omega \in A_S$ and $\omega \in A_T$, we thus have $X_S + X' = \int_0^{T_0 \circ \Psi^0} I_\mathcal{E} \circ \Psi_s^0 \, ds$ a.s.

We now show that $E_T(\min(T_{\xi}, x)) = 2E_T(X')$ a.s. The proof is similar to the proofs of Propositions 3.8 and 3.10. Define random variable $X'' : \Omega \to \mathbb{R}_+$ such
that $X''(\omega) = 0$ for all $\omega \in A_T$, and $X''(\omega) = \min(x, T_1(\Psi^0(\omega)))$ for all $\omega \in A_S$. Then by definition $\min(T_G, x) = X' + X''$. Thus it is enough to show that $E_I(X'') = E_I(X')$ a.s. Let $f'$ and $f''$ be the corresponding measurable mappings such that $X' = f' \circ \Psi^0$ and $X'' = f'' \circ \Psi^0$. Note that $f'' \circ \Psi^0(j) = f' \circ \Psi^0(j + 1)$ for all $j \geq 0$. Using equality (A.9), we have

$$E_I(X'') = E_I(f'' \circ \Psi^0) = \lim_{n \to \infty} \frac{1}{n} \sum_{j=0}^{n-1} f'' \circ \Psi^0(j) = \lim_{n \to \infty} \frac{1}{n} \sum_{j=1}^{n} f' \circ \Psi^0(j)$$

$$= E_I(f' \circ \Psi^0) = E_I(X') \text{ a.s.}$$

We thus have $E_I(\min(T_G, x)) = 2E_I(X')$ a.s.

Therefore, from (3.14) and Propositions 3.8 and 3.10, we have

$$Pr(Y \leq x) = E \left[ E_I(X_S) + E_I(X') \right] = E \left[ \frac{E_I(T_M) + E_I(\min(T_G, x))}{E_I(T_{MR})} \right]$$

$$= 1 - P_A + E \left[ \frac{E_I(\min(T_G, x))}{E_I(T_{MR})} \right].$$

The last equality is due to Lemmata 3.5 and 3.11. Plugging the above result into (3.13), we then have (3.12).

**Corollary 3.13** If $\Psi$ is ergodic and if $0 < E(T_{MR}) < \infty$ and $E(T_G) \neq 0$, then

for all $x \in \mathbb{R}_+$, $Pr(T_{FG} \leq x) = \frac{1}{E(T_G)} \int_0^x Pr(T_G > y)dy,$ \hspace{1cm} (3.15)

$$E(T_{FG}^k) = \frac{E(T_G^{k+1})}{(k+1)E(T_G)}.$$ \hspace{1cm} (3.16)

In particular,

$$E(T_{FG}) = \frac{E(T_G^2)}{2E(T_G)} = \frac{E(T_G)}{2} \left( 1 + \frac{V(T_G)}{E(T_G)^2}) \right).$$ \hspace{1cm} (3.17)
Proof. By Lemma 3.11, if \( \Psi \) is ergodic and \( 0 < E(T_{mr}) < \infty \), then \( P_A = E(T_G)/E(T_{mr}) \). If \( E(T_G) \neq 0 \), then \( P_A > 0 \). Then by Proposition A.4 and Lemma 3.12, we have

\[
Pr(T_{rg} \leq x) = E\left(\frac{\min(T_G, x)}{E(T_G)}\right).
\]

We now use the fact (see e.g. [Bil95] p.275) that for any nonnegative random variable \( X \),

\[
E(X) = \int_0^\infty Pr(X > t) \, dt. \tag{3.18}
\]

We have \( E(\min(T_G, x)) = \int_0^\infty Pr(\min(T_G, x) > y) \, dy = \int_0^x Pr(T_G > y) \, dy \). Thus

\[
Pr(T_{rg} \leq x) = \frac{1}{E(T_G)} \int_0^x Pr(T_G > y) \, dy.
\]

To prove (3.16), we substitute \( X \) in (3.18) with \( X^k \) to have

\[
E(X^k) = \int_0^\infty Pr(X^k > t) \, dt = \int_0^\infty Pr(X > t^{1/k}) \, dt = \int_0^\infty Pr(X > t) \, d(t^{k-1}) = \int_0^\infty kt^{k-1}Pr(X > t) \, dt.
\]

Then together with (3.15) we have

\[
E(T_{rg}^k) = \int_0^\infty kt^{k-1}Pr(T_{rg} > t) \, dt = \int_0^\infty kt^{k-1}\left(1 - \frac{1}{E(T_G)} \int_0^t Pr(T_G > y) \, dy\right) \, dt
\]

\[
= \frac{1}{E(T_G)} \int_0^\infty kt^{k-1} \left(E(T_G) - \int_0^t Pr(T_G > y) \, dy\right) \, dt
\]

\[
= \frac{1}{E(T_G)} \int_0^\infty kt^{k-1} \left(\int_t^\infty Pr(T_G > y) \, dy\right) \, dt
\]

\[
= \frac{1}{E(T_G)} \int_0^\infty Pr(T_G > y) \left(\int_0^y kt^{k-1} \, dt\right) \, dy
\]

\[
= \frac{1}{E(T_G)} \int_0^\infty Pr(T_G > y)y^{k} \, dy = \frac{E(T_{rg}^{k+1})}{(k+1)E(T_G)}.
\]
(3.17) is obtained from (3.16) by setting \( k = 2 \) and using the fact that \( E(X^2) = E(X)^2 + V(X) \).

We now summarize all the results in the following theorem.

**Theorem 3.14** For any failure detector \( \mathcal{D} \), the following results hold:

1. \( T_{MR} = T_M + T_G \).
2. Suppose \( 0 < E(T_{MR}) < \infty \), a.s. Then
   \[
   \lambda_M = E \left[ \frac{1}{E(T_{MR})} \right],
   \]
   \[
   P_A = E \left[ \frac{E(T_G)}{E(T_{MR})} \right].
   \]
   If \( P_A = 0 \) then \( T_{FG} \equiv 0 \); if \( P_A > 0 \) then
   \[
   Pr(T_{FG} \leq x) = \frac{1}{P_A} E \left[ \frac{E(T_G, \min(T_G, x))}{E(T_{MR})} \right].
   \]
3. Suppose \( \Psi \overset{\text{def}}{=} \mathcal{D}(\infty) \) is ergodic and \( 0 < E(T_{MR}) < \infty \). Then
   \[
   \lambda_M = \frac{1}{E(T_{MR})},
   \]
   \[
   P_A = \frac{E(T_G)}{E(T_{MR})},
   \]
   If \( E(T_G) = 0 \) then \( T_{FG} \equiv 0 \); if \( E(T_G) \neq 0 \), then
   for all \( x \in \mathbb{R}_+ \),
   \[
   Pr(T_{FG} \leq x) = \frac{1}{E(T_G)} \int_0^x Pr(T_G > y)dy,
   \]
   \[
   E(T_{FG}^k) = \frac{E(T_G^{k+1})}{(k+1)E(T_G)}.
   \]
   In particular,
   \[
   E(T_{FG}) = \frac{E(T_G^2)}{2E(T_G)} = \frac{E(T_G)}{2} \left( 1 + \frac{V(T_G)}{E(T_G)^2} \right).
   \]

Theorem 2.1 in Chapter 2 is just the parts (1) and (3) of the above theorem.
Chapter 4

The Design and Analysis of a New Failure Detector Algorithm

4.1 Introduction

In Chapter 2, we proposed a set of specification metrics to measure the QoS provided by failure detectors. These metrics address the failure detector’s speed (how fast it detects process crashes) and its accuracy (how well it avoids erroneous detections). In this chapter, we design a new failure detector algorithm for distributed systems with probabilistic behaviors. We analyze the QoS of the new algorithm and derive closed formulas on its QoS metrics. We show that, among a large class of failure detector algorithms, the new algorithm is optimal with respect to some of these QoS metrics. Given a set of failure detector QoS requirements, we show how to compute the parameters of our algorithm so that it satisfies these requirements, and we show how this can be done even if the probabilistic behavior of the system is
not known. Finally, we simulate both the new algorithm and a simple algorithm commonly used in practice, compare the simulation results, and show that the new algorithm provides better QoS than the simple algorithm.

We consider a simple system of two processes $p$ and $q$ connected through a communication link. Process $p$ may fail by crashing, and the link between $p$ and $q$ may delay or drop messages. Message delays and message losses follow some probabilistic distributions. Process $q$ has a failure detector that monitors $p$. As in Chapter 2, the failure detector at $q$ outputs either “I suspect that $p$ has crashed” or “I trust that $p$ is up” (“suspect $p$” and “trust $p$” in short, respectively).

4.1.1 A Common Failure Detection Algorithm and its Drawbacks

We first consider the following simple failure detector algorithm commonly used in practice: at regular time intervals, process $p$ sends heartbeat messages to $q$; when $q$ receives a more recent heartbeat message, it trusts $p$ and starts a timer with a fixed timeout value $TO$; if the timer expires before $q$ receives a more recent heartbeat message from $p$, $q$ starts suspecting $p$.

This algorithm has two undesirable characteristics; one regards its accuracy and the other its detection time, as we now explain. Consider the $i$-th heartbeat message $m_i$. Intuitively, the probability of a premature timeout on $m_i$ should depend solely on $m_i$, and in particular on $m_i$’s delay. With the simple algorithm, however, the probability of a premature timeout on $m_i$ also depends on the heartbeat $m_{i-1}$ that precedes $m_i$! In fact, the timer for $m_i$ is started upon the receipt of $m_{i-1}$, and so if
$m_{i-1}$ is "fast", the timer for $m_i$ starts early and this increases the probability of a premature timeout on $m_i$. This dependency on past heartbeats is undesirable.

To see the second problem, suppose $p$ sends a heartbeat just before it crashes, and let $d$ be the delay of this last heartbeat. In the simple algorithm, $q$ would permanently suspect $p$ only $d + TO$ time units after $p$ crashes. Thus, the worst-case detection time for this algorithm is the maximum message delay plus $TO$. This is impractical because in many systems the maximum message delay is orders of magnitude larger than the average message delay (i.e., they have large variations of message delays).

The source of the above problems is that even though the heartbeats are sent at regular intervals, the timers to "catch" them expire at irregular times, namely the receipt times of the heartbeats plus a fixed $TO$. The algorithm that we propose eliminates this problem. As a result, the probability of a premature timeout on heartbeat $m_i$ does not depend on the behavior of the heartbeats that precede $m_i$, and the detection time does not depend on the maximum message delay.

4.1.2 The New Algorithm and its QoS Analysis

In this chapter, we design a new failure detector algorithm that has good worst-case detection time and good accuracy.

In the new failure detector algorithm, process $p$ sends heartbeat messages to $q$ periodically, as in the simple algorithm. Suppose $m_1, m_2, m_3, \ldots$ are heartbeat messages and $\eta$ is the intersending time between two consecutive messages. The new algorithm differs from the simple algorithm in the procedure that $q$ uses to decide whether to suspect $p$ or not. In the new algorithm, $q$ has a sequence of time points $\tau_1, \tau_2, \tau_3, \ldots$, called freshness points. Each freshness point $\tau_i$ is set to $\sigma_i + \delta$, where
σi is the time when \( m_i \) is sent and \( δ \) is a fixed parameter of the algorithm. That is, the freshness points are obtained by shifting the sending times of the heartbeat messages forward in time by a fixed \( δ \) time units. These freshness points are used to determine the failure detector output. Roughly speaking, at any time \( t \in [τ_i, τ_{i+1}) \), only messages \( m_i, m_{i+1}, m_{i+2}, \ldots \) can affect the failure detector output, and we say that only these messages are \textit{still fresh (at time} \( t \))\textit{), and messages } \( m_{1}, \ldots, m_{i−1} \) \textit{are stale (at time} \( t \)). At any time \( t \), process \( q \) trusts \( p \) if and only if some message that \( q \) received is still fresh at time \( t \). A detailed description of the algorithm is given in Section 4.3.1.

We analyze the algorithm in terms of the QoS metrics proposed in Chapter 2. The analysis uses the theory of stochastic processes, and provides some closed formulas on the QoS metrics of the new algorithm. We then show the following optimality result: among all failure detector algorithms that send messages at the same rate and satisfy the same upper bound on the worst-case detection time, our algorithm is optimal with respect to the \textit{query accuracy probability}. This shows that the new algorithm guarantees good worst-case detection time while providing good accuracy. We then show that, given a set of QoS requirements by an application, we can use the closed formulas we derived to compute the parameters of the new algorithm to meet these requirements. We first do so assuming that one knows the probabilistic behavior of the system (i.e., the probability distributions of message delays and message losses). We then drop this assumption, and show how to configure the failure detector to meet the QoS requirements of an application even when the probabilistic behavior of the system is not known.

The first version of our algorithm (described above) assumes that \( p \) and \( q \) have
synchronized clocks. This assumption is not unrealistic, even in large networks. For example, GPS clocks are becoming cheap, and they can provide clocks that are very closely synchronized (see e.g. [VR00]). When synchronized clocks are not available, we propose a modification to this algorithm that performs equally well in practice, as shown by our simulations. The basic idea is to use past heartbeat messages to obtain accurate estimates of the expected arrival times of future heartbeats, and then use these estimates to find the freshness points. This computation uses the same heartbeat messages used for failure detection, so it does not involve additional system cost.

The modified algorithm has another parameter, namely $n$, which is the number of messages used to estimate the expected arrival times of the heartbeat messages. As $n$ varies from 1 to $\infty$, we obtain a spectrum of algorithms. An important observation is that one end point of this spectrum ($n = 1$) corresponds to the simple algorithm, and the other end point ($n = \infty$) corresponds to the new algorithm with known expected arrival times of all heartbeat messages. As $n$ increases, the new algorithm moves away from the simple algorithm and gets closer to the new algorithm with known expected arrival times. This demonstrates that the problem of the simple algorithm is that it does not use enough information available (it only uses the most recently received heartbeat message), and by using more information available (using more messages received), the new algorithm is able to provide better QoS than the simple algorithm.

Finally, we run simulations of both the new algorithm and the simple algorithm, and provide detailed analysis on the simulation results. The conclusion we draw from the simulation results are: (a) the simulation results of the new algorithm are
consistent with our mathematical analysis of the QoS metrics; (b) the modified new algorithm for systems with unsynchronized clocks provides essentially the same QoS as the algorithm with synchronized clocks; and (c) when the new algorithm and the simple algorithm send messages at the same rate and satisfy the same upper bound on the worst-case detection time, the new algorithm provides (in some cases orders of magnitude) better accuracy than the simple algorithm.

4.1.3 Related Work

Heartbeat-style failure detectors are commonly used in practice. To keep both good detection time and good accuracy, many implementations rely on special features of the operating system and communication system to try to deliver heartbeat messages as regularly as possible (see discussion in Section 12.9 of [Pfi98]). This is not easy even for closely-connected computer clusters, and it is very hard in wide-area networks.

Some other failure detector algorithms and their analyses can be found in [vRMH98, GM98, RT99]. The gossip-style heartbeat protocol in [vRMH98] focus on the scalability of the protocol, and it falls into the category of the simple algorithm as given in Section 4.1.1. In this protocol, nodes in the network randomly pick some other node to forward a heartbeat message, so heartbeat messages generated by a source node may in some cases reach a destination node directly, while in some other cases may be forwarded by many other intermediate nodes before reaching the same destination node. Thus the protocol has a large variation of the end-to-end message delays, and therefore it has the problem of the simple algorithm pointed out in Section 4.1.1. Algorithms presented in [GM98] are different from one-way heartbeat algorithms we discussed in this paper, and they are used in a more limited
setting in which a single suspicion will terminate the protocol. The group membership failure detection algorithm presented in [RT99] detects member failures in a group: if some process detects a failure in the group (perhaps a false detection), then all processes report a group failure and the protocol terminates. The algorithm uses heartbeat-style protocol, and its timeout mechanism is the same as the simple algorithm that we described in Section 4.1.1.

The probabilistic network model used in this chapter is similar to the ones used in [Cri89, Arv94] for probabilistic clock synchronization. The method of estimating the expected arrival times of heartbeat messages is close to the method of remote clock reading of [Arv94].

The rest of the chapter is organized as follows. In Section 4.2, we define the probabilistic network model. In Section 4.3, we present the new failure detector algorithm, analyze it in terms of the QoS metrics, show the optimality result, and show how to configure the new algorithm to satisfy given QoS requirements. In Section 4.4 we show how to configure the failure detector algorithm when the probabilistic behavior of the messages is not known, and how to modify the algorithm so that it works when the local clocks are not synchronized. We present the simulation results in Section 4.5, and conclude the chapter with some discussions in Section 4.6.
4.2 The Probabilistic Network Model

We assume that process $p$ and $q$ are connected by a link that does not create or duplicate messages,\footnote{Message duplication can be easily taken care of: whenever we refer to a message being received, we change it to the \textit{first copy} of the message being received. With this modification, all definitions and analyses in this chapter go through, and in particular, our results remain correct without any change.} but may delay or drop messages. Note that the link here represents an end-to-end connection and does not necessarily correspond to a physical link.

We assume that the message loss and message delay behavior of any message sent through the link is probabilistic, and is characterized by the following two parameters: (a) \textit{message loss probability} $p_L$, which is the probability that a message is dropped by the link; and (b) \textit{message delay time} $D$, which is a random variable with range $(0, \infty)$ representing the delay from the time a message is sent to the time it is received, under the condition that the message is not dropped by the link. We assume that the expected value $E(D)$ and the variance $V(D)$ of $D$ is finite. Note that our model does not assume that the message delay time $D$ follows any particular distribution, and thus it is applicable to many practical systems.

Processes $p$ and $q$ have access to their own local clocks. For simplicity, we assume that there is no clock drift, i.e., local clocks run at the same speed as real time (our results can be easily generalized to the case where local clocks have bounded drifts). In Section 4.3, we further assume that clocks are synchronized. We explain how to remove this assumption in Section 4.4.2.

For simplicity we assume that the probabilistic behavior of the network does not change over time. In Section 4.6 we briefly discuss some issues related to the change
of network behavior, and explain how our algorithm can adapt to such changes.

4.3 The New Failure Detector Algorithm and Its Analysis

4.3.1 The Algorithm

In the new algorithm, the task of process $p$ is the same as in the simple algorithm: $p$ periodically sends heartbeat messages $m_1, m_2, m_3, \ldots$ to $q$ every $\eta$ time units, where $\eta$ is a parameter of the algorithm. Heartbeat message $m_i$ is tagged with its sequence number $i$. Let $\sigma_i$ be the sending time of message $m_i$.

The new algorithm differs from the simple algorithm in the task of process $q$. In the new algorithm, $q$ has a sequence of time points $\tau_1 < \tau_2 < \tau_3 < \ldots$, such that $\tau_i$ is obtained by shifting the sending time $\sigma_i$ forward in time by $\delta$ time units (i.e. $\tau_i = \sigma_i + \delta$), where $\delta$ is a fixed parameter of the algorithm. Time points $\tau_i$'s, together
with the arrival times of the heartbeat messages, are used to determine the output of the failure detector at \( q \), as we now explain. Consider the time period \([\tau_i, \tau_{i+1})\). At time \( \tau_i \), the failure detector at \( q \) checks whether \( q \) has received some message \( m_j \) with \( j \geq i \). If so, the failure detector trusts \( p \) in the period \([\tau_i, \tau_{i+1})\) (Fig. 4.1 (a)). If not, the failure detector starts suspecting \( p \) at time \( \tau_i \). During the period \([\tau_i, \tau_{i+1})\), if \( q \) receives some message \( m_j \) with \( j \geq i \), then the failure detector at \( q \) starts trusting \( p \) when the message is received, and keeps trusting \( p \) until time \( \tau_{i+1} \) (Fig. 4.1 (b)). If by time \( \tau_{i+1} \) \( q \) has not received any message \( m_j \) with \( j \geq i \), then the failure detector suspects \( p \) in the entire period \([\tau_i, \tau_{i+1})\) (Fig. 4.1 (c)). This procedure is repeated for every period.

Note that from time \( \tau_i \) to \( \tau_{i+1} \), only messages \( m_j \) with \( j \geq i \) can affect the output of the failure detector. For this reason, \( \tau_i \) is called a freshness point: from time \( \tau_i \) to \( \tau_{i+1} \), messages \( m_j \) with \( j \geq i \) are still fresh (useful), and messages \( m_j \) with \( j < i \) are stale (not useful). The core property of the algorithm is that \( q \) trusts \( p \) at time \( t \) if and only if some message that \( q \) received is still fresh at time \( t \).

The detailed algorithm, denoted by NFD-S, is given in Fig. 4.2.2

### 4.3.2 The Analysis

We now analyze the QoS metrics of the algorithm. For the analysis, we assume that the link from \( p \) to \( q \) satisfies the following message independence property: (a) the message loss and message delay behavior of any message sent by \( p \) is independent of whether or when \( p \) crashes later; and (b) there exists a known constant \( \Delta \) such that

---

2This version of the algorithm is convenient for illustrating the main idea and for performing the analysis. One can easily derive some equivalent version that is more efficient in practice.
Process $p$:
1. for some constant $\eta$, send to $q$ heartbeat messages $m_1, m_2, m_3, \ldots$ at regular time points $\eta, 2\eta, 3\eta, \ldots$ respectively;

Process $q$:
1. Initialization:
2. for all $i \geq 1$, set $\tau_i = \sigma_i + \delta$;
3. $\text{output} = S$;  \hspace{1cm} \{$\sigma_i = i\eta$ is the sending time of $m_i$} \hspace{1cm} \{$\text{suspect } p \text{ initially}$}
4. at every freshness point $\tau_i$:
5. if no message $m_j$ with $j \geq i$ has been received then
6. $\text{output} \leftarrow S$; \hspace{1cm} \{$\text{suspect } p \text{ if no fresh message is received}$}
7. upon receive message $m_j$ at time $t \in [\tau_i, \tau_{i+1})$:
8. if $j \geq i$ then $\text{output} \leftarrow T$; \hspace{1cm} \{$\text{trust } p \text{ when some fresh message is received}$}

Figure 4.2: The new failure detector algorithm NFD-S, with synchronized clocks, and with parameters $\eta$ and $\delta$

the message loss and message delay behaviors of any two messages sent at least $\Delta$ time units apart are independent. We assume that the intersending time $\eta$ is chosen such that $\eta \geq \Delta$, so that all heartbeat messages have independent delay and loss behaviors. For simplicity, we assume that the actions in lines 5–7 and lines 8–9 are executed instantaneously without interruption.

We adopt the following convention about transitions of a failure detector’s output:\footnote{This convention is already included in the formal model defined in Chapter 3.} when an S-transition occurs at time $t$, the output at time $t$ is $S$, and a symmetric convention is taken for T-transitions. With this convention, the output is right continuous: namely, if the output at a time $t$ is $X \in \{T, S\}$, then there exists $\epsilon > 0$ such that the output is also $X$ in the period $(t, t + \epsilon)$.

Henceforth, let $\tau_0 \overset{\text{def}}{=} 0$, and $\tau_i, i \geq 1$, are given as in line 3. The following core lemma states precisely our intuitive ideas about freshness points and fresh messages. All subsequent analyses are based on this lemma.
Lemma 4.1 For all $i \geq 0$ and all time $t \in [\tau_i, \tau_{i+1})$, $q$ trusts $p$ at time $t$ if and only if $q$ has received some message $m_j$ with $j \geq i$ by time $t$.

Proof. Fix an $i \geq 0$ and a time $t \in [\tau_i, \tau_{i+1})$. Suppose first that $q$ has received some message $m_j$ with $j \geq i$ by time $t$. Let $t' \leq t$ be the time when $m_j$ is received. Choose $i'$ such that $t' \in [\tau_{i'}, \tau_{i'+1})$. Thus $i' \leq i \leq j$. According to line 9, $q$ trusts $p$ at time $t'$. For every $\tau_\ell$ in the period $(t', t]$, since $m_j$ is received at $t'$ and $\ell \leq i \leq j$, the output of the failure detector does not change to $S$, according to lines 5–7. Therefore, $q$ trusts $p$ at time $t$.

Suppose now that $q$ has not received any message $m_j$ with $j \geq i$ by time $t$. Then at time $\tau_i$, $q$ suspects $p$ according to lines 5–7. During the period $(\tau_i, t]$, since no message $m_j$ with $j \geq i$ is received, the output of the failure detector does not change to $T$. So $q$ suspects $p$ at time $t$. \hfill \Box

The following definitions are used in the analysis, and they are all with respect to failure-free runs.\footnote{Recall that a failure-free run is a run in which $p$ does not crash, as defined in Section 2.2.1.} Note that even though $i$ appears in these definitions, the actual values of $i$ are irrelevant. This is made clear in Proposition 4.2.

Definition 4.1

(1) For any $i \geq 1$, let $k$ be the smallest integer such that for all $j \geq i + k$, $m_j$ is sent at or after time $\tau_i$.

(2) For any $i \geq 1$, let $p_j(x)$ be the probability that $q$ does not receive message $m_{i+j}$ by time $\tau_i + x$, for every $j \geq 0$ and every $x \geq 0$; let $p_0 = p_0(0)$.

(3) For any $i \geq 2$, let $q_0$ be the probability that $q$ receives message $m_{i-1}$ before time $\tau_i$. 

(4) Recall that a failure-free run is a run in which $p$ does not crash, as defined in Section 2.2.1.
(4) For any \(i \geq 1\), let \(u(x)\) be the probability that \(q\) suspects \(p\) at time \(\tau_i + x\), for every \(x \in [0, \eta)\).

(5) For any \(i \geq 2\), let \(p_s\) be the probability that an S-transition occurs at time \(\tau_i\).

**Proposition 4.2**

(1) \(k = \lceil \delta/\eta \rceil\).

(2) For all \(j \geq 0\) and for all \(x \geq 0\), \(p_j(x) = p_L + (1 - p_L)Pr(D > \delta + x - j\eta)\).

(3) \(q_0 = (1 - p_L)Pr(D < \delta + \eta)\).

(4) For all \(x \in [0, \eta)\), \(u(x) = \prod_{j=0}^{k} p_j(x)\).

(5) \(p_s = q_0 \cdot u(0)\).

Upon the first reading, readers can skip the rest of the analysis and go directly to Theorem 4.11.

We now analyze the accuracy metrics of the algorithm NFD-S, and to do so we assume that \(p\) does not crash.

**Proposition 4.3** (1) An S-transition can only occur at time \(\tau_i\) for some \(i \geq 2\), and it occurs at \(\tau_i\) if and only if message \(m_{i-1}\) is received by \(q\) before time \(\tau_i\) and no message \(m_j\) with \(j \geq i\) is received by \(q\) by time \(\tau_i\); (2) Lemma 4.1 remains true if \(j \geq i\) in the statement is replaced by \(i \leq j \leq i + k\); (3) part (1) above remains true if \(j \geq i\) in the statement is replaced by \(i \leq j < i + k\).

**Proof.** From the algorithm, it is clear that an S-transition can only occur at time \(\tau_i\) with \(i \geq 1\). An S-transition cannot occur at time \(\tau_1\), because if so, \(q\) suspects \(p\) at time \(\tau_1\), which implies from Lemma 4.1 that \(q\) does not receive \(m_i\) by time \(\tau_1\) for all \(i \geq 1\). Then \(q\) must also suspect \(p\) during the period \([0, \tau_1)\) — a contradiction.
An S-transition occurs at time $\tau_i$ if and only if (a) $q$ suspects $p$ at time $\tau_i$, and (b) for some $t' \in (\tau_{i-1}, \tau_i)$, $q$ trusts $p$ at time $t'$. Then by Lemma 4.1, (a) is true if and only if no message $m_j$ with $j \geq i$ is received by $q$ by time $\tau_i$, while (b) is true if and only if some message $m_j$ with $j \geq i - 1$ is received by $q$ by time $t' < \tau_i$. Combining (a) and (b), we know that an S-transition occurs at time $\tau_i$ if and only if message $m_{i-1}$ is received by $q$ before time $\tau_i$ and no message $m_j$ with $j \geq i$ is received by $q$ by time $\tau_i$.

(2) and (3) follow from the definition of $k$.  

Note that part (1) of the above proposition guarantees that during any bounded time period, there is only a finite number of transitions of failure detector output.

**Proof of Proposition 4.2.** (1) is immediate from the fact that $m_j$ is sent at time $\tau_i - \delta + (j - i)\eta$ for all $i \geq 1$.

(2) directly follows from the fact that $p_j(x)$ is the probability that either $m_{i+j}$ is lost, or $m_{i+j}$ is not lost but is delayed by more than $\sigma_i + \delta + x - (\sigma_i + j\eta) = \delta + x - j\eta$ time units.

(3) directly follows from the fact that $q_0$ is the probability that $m_{i-1}$ is not lost and is delayed less than $\delta + \eta$ time units.

(4) By Proposition 4.3 (2), $u(x)$ is the probability that $q$ does not receive any message $m_j$ with $i \leq j \leq i + k$ by time $\tau_i + x$. Then by the definition of $p_j(x)$ and the message independence property, we have $u(x) = \prod_{j=0}^{k} p_j(x)$.

(5) By Proposition 4.3 (1), $p_s$ is the probability that (a) message $m_{i-1}$ is received by $q$ before time $\tau_i$, and (b) no message $m_j$ with $j \geq i$ is received by $q$ by time $\tau_i$. By the message independence property, (a) and (b) are independent, and by Lemma 4.1, (b) is also the event that $q$ suspects $p$ at time $\tau_i$. Thus by the definitions of $q_0$ and
Proposition 4.4 \( u(0) \geq p_{0}^{k}, \) and for all \( x \in [0, \eta), \) \( u(0) \geq u(x). \)

Proof. By Proposition 4.2, \( p_{j}(0) \geq p_{0}(0) = p_{0}, \) \( p_{k}(0) = 1, \) and \( p_{j}(0) \geq p_{j}(x) \) for \( x \in [0, \eta). \) So \( u(0) = \prod_{j=0}^{k} p_{j}(0) \geq \prod_{j=0}^{k-1} p_{0} = p_{0}^{k}, \) and \( u(0) = \prod_{j=0}^{k} p_{j}(0) \geq \prod_{j=0}^{k} p_{j}(x) = u(x). \) \( \square \)

Lemma 4.5 (1) If \( p_{0} = 0, \) then with probability one \( q \) trusts \( p \) forever after time \( \tau_{1}; \) (2) If \( q_{0} = 0, \) then with probability one \( q \) suspects \( p \) forever; (3) If \( p_{0} > 0 \) and \( q_{0} > 0, \) then with probability one the failure detector at \( q \) has an infinite number of transitions.

Proof. (1) By definition, \( p_{0} = 0 \) means that for all \( i \geq 1, \) the probability that \( q \) does not receive \( m_{i} \) by time \( \tau_{i} \) is 0. Thus by Lemma 4.1, the probability that \( q \) keeps trusting \( p \) in the period \( [\tau_{i}, \tau_{i+1}) \) is 1. Therefore, with probability one \( q \) trusts \( p \) forever after time \( \tau_{1}. \)

(2) By definition, \( q_{0} = 0 \) means that for all \( i \geq 2, \) the probability that \( q \) receives \( m_{i-1} \) before time \( \tau_{i} \) is 0. For every \( j \geq i, \) message \( m_{j} \) is sent after \( m_{i-1}, \) so the probability that \( q \) receives \( m_{j} \) before time \( \tau_{i} \) is also 0. This implies that the probability that \( q \) receives some \( m_{j} \) with \( j \geq i - 1 \) before time \( \tau_{i} \) is 0. By Lemma 4.1, we have that for all \( i \geq 0, \) the probability that \( q \) keeps suspecting \( p \) in the period \( [\tau_{i}, \tau_{i+1}) \) is 1. Thus with probability one \( q \) suspects \( p \) forever.

(3) Suppose \( p_{0} > 0 \) and \( q_{0} > 0. \) For all \( i \geq 2, \) let \( A_{i} \) be the event that there is an S-transition at time \( \tau_{i}. \) By Proposition 4.3 (3), \( A_{i} \) is also the event that message \( m_{i-1} \) is received before time \( \tau_{i} \) but no messages \( m_{j} \) with \( i \leq j < i + k \) is received by time \( \tau_{i}. \) Hence \( A_{i} \) only depends on messages \( m_{j} \) with \( i - 1 \leq j < i + k, \) which
implies that \([A_{i(k+1)}, i \geq 2]\) are independent. By definition and Proposition 4.4, we have \(Pr(A_i) = q_0 \cdot u(0) \geq q_0 \cdot p_0^k > 0\). Therefore, with probability one, the failure detector at \(q\) has an infinite number of transitions. \(\square\)

The above lemma factors out the special case in which \(p_0 = 0\) or \(q_0 = 0\). We call this special case the degenerated case. From now on, we only consider the nondegenerated case in which \(p_0 > 0\) and \(q_0 > 0\), and only consider runs in which the output of the failure detector has an infinite number of transitions.

**Lemma 4.6** \(P_A = 1 - \frac{1}{\eta} \int_{0}^{\eta} u(x) \, dx\).

**Proof.** For all \(i \geq 1\), let \(P_i\) be the probability that at any random time \(T \in [\tau_i, \tau_{i+1})\), \(q\) is suspecting \(p\). Note that \(T\) is uniformly distributed on \([\tau_i, \tau_{i+1})\) with density \(1/(\tau_{i+1} - \tau_i) = 1/\eta\). Thus

\[
P_i = \frac{1}{\eta} \int_{\tau_i}^{\tau_{i+1}} u(x - \tau_i) \, dx = \frac{1}{\eta} \int_{0}^{\eta} u(x) \, dx.
\]

Note that the value of \(P_i\) does not depend on \(i\). Let this value be \(P\). Thus we have that \(P_A\), the probability that \(q\) trusts \(p\) at a random time, is \(1 - P\). This shows the lemma. \(\square\)

We now analyze the average mistake recurrence time \(E(T_{MR})\) of the failure detector. We will show that

**Lemma 4.7** \(E(T_{MR}) = \eta/p_s\).

If, at each time point \(\tau_i\) with \(i \geq 2\), the test of whether an S-transition occurs were an independent Bernoulli trial, then the above result would be very easy to obtain: \(p_s\) is the probability of success in one Bernoulli trial, i.e. an S-transition occurs at \(\tau_i\), and \(\eta\) is the time between two Bernoulli trials, and so \(\eta/p_s\) is the
expected time between two successful Bernoulli trials, which is just the expected
time between two S-transitions. Unfortunately, this is not the case because the tests
of whether S-transitions occur at $\tau_i$'s are not independent. In fact, by Proposition 4.3,
the event that an S-transition occurs at $\tau_i$ depends on the behavior of messages
$m_i, \ldots, m_{i+k-1}$. Thus two such events may depend on the behavior of common
messages, and so they are not independent in general.

To deal with this, we use some results in renewal theory, a branch in the theory
of stochastic processes. Besides proving Lemma 4.7, the analysis also reveals an
important property of the failure detector output: each recurrence interval between
two consecutive S-transitions is independent of other recurrence intervals.

The analysis proceeds as follows. We first introduce the concept of a renewal
process. A more formal account can be found in any standard textbook on stochastic
processes (see for example Chapter 3 of [Ros83]). Let $\{(T_n, R_n), n = 1, 2, \ldots\}$ be a
sequence of random variable pairs such that (1) a nonnegative $T_n$ denotes the time
between the $(n - 1)$-th and $n$-th occurrences of some recurrent event $A$, i.e., event
$A$ occurs at time $t_1 = T_1$, $t_2 = T_1 + T_2$, $t_3 = T_1 + T_2 + T_3$, \ldots; and (2) $R_n$ can be
interpreted as the reward associated with the $n$-th occurrence of event $A$. A delayed
renewal reward process is such a sequence satisfying: (1) The pairs $(T_n, R_n), n \geq 1$
are mutually independent; and (2) The pairs $(T_n, R_n), n \geq 2$ are identically distributed.
If $\{R_n\}$ is omitted, then the above process $\{T_n, n \geq 1\}$ is called a delayed renewal
process. Such processes are well studied in the literature, and are known to have
some nice properties.

Now consider S-transitions of the failure detector output as the recurrent events.
Let $T_{MR,n}$ be the random variable representing the time that elapses from the $(n - 1)$-
th S-transition to the \( n \)-th S-transition (as a convention consider time 0 to be the time at which the 0-th S-transition occurs). Let \( T_{x,n} \) be the random variable representing the time that elapses from the \((n-1)\)-th S-transition to the \( n \)-th T-transition. Thus
\[ T_{M,n} \leq T_{MR,n} \text{ for all } n \geq 1. \]

**Lemma 4.8** \( \{(T_{MR,n}, T_{M,n}), n = 1, 2, \ldots\} \) is a delayed renewal reward process.

We need the following technical result before proving this lemma.

**Proposition 4.9** Let \( \{A_i, i \geq 1\} \) be an event partition (i.e. disjoint and covers all events). Two random variables \( X \) and \( Y \) are independent if for all \( A_i \): (1) \( X \) is independent of \( A_i \), that is, if \( Pr(A_i) > 0 \) then for all \( x \in [\infty, \infty] \), \( Pr(X \leq x) = Pr(X \leq x \mid A_i) \); and (2) \( X \) and \( Y \) are independent when conditioned on \( A_i \), that is, if \( Pr(A_i) > 0 \) then for all \( x, y \in [\infty, \infty] \), \( Pr(X \leq x, Y \leq y \mid A_i) = Pr(X \leq x \mid A_i)Pr(Y \leq y \mid A_i) \).

**Proof.** For all \( x, y \in [\infty, \infty] \),
\[
Pr(X \leq x, Y \leq y) = \sum_{i=1}^{\infty} Pr(X \leq x, Y \leq y \mid A_i)Pr(A_i)
= \sum_{Pr(A_i) \succ 0} Pr(X \leq x \mid A_i)Pr(Y \leq y \mid A_i)Pr(A_i)
= Pr(X \leq x) \sum_{Pr(A_i) \succ 0} Pr(Y \leq y \mid A_i)Pr(A_i)
= Pr(X \leq x)Pr(Y \leq y).
\]
Thus \( X \) and \( Y \) are independent. \( \Box \)

Note that we can replace \( X \) and \( Y \) in the above proposition with random vectors and the result still holds.
Proof of Lemma 4.8. For all \( n \geq 1 \), by Proposition 4.3 (1), the \( n \)-th S-transition can only occur at time \( \tau_i \) for some \( i \geq 2 \). Let \( A_i^n \) be the event that the \( n \)-th S-transition occurs at time \( \tau_i \). Thus for each \( n \geq 1 \), \( \{A_i^n, i \geq 2\} \) is an event partition. Let \( B_i^n \) be the event consisting of all the runs in which the messages \( m_j \) with \( j < i \) behave in the same way as in some run in \( A_i^n \). Let \( C_i \) be the event that no message \( m_j \) with \( j \geq i \) is received by time \( \tau_i \). Since \( C_i \) and \( B_i^n \) are determined by completely different set of messages, \( C_i \) is independent of \( B_i^n \).

To complete the proof of the lemma, we now show the following five claims.

Claim 1. For all \( n \geq 1 \) and for all \( i \geq 2 \), \( A_i^n = B_i^n \cap C_i \).

Proof of Claim 1. By definition, \( A_i^n \subseteq B_i^n \). By Proposition 4.3 (1), \( A_i^n \) implies that no message \( m_j \) with \( j \geq i \) arrives by time \( \tau_i \), and thus \( A_i^n \subseteq C_i \). So \( A_i^n \subseteq B_i^n \cap C_i \).

For any run \( r_1 \) in \( B_i^n \cap C_i \), by the definition of \( B_i^n \), there is a run \( r_2 \) in \( A_i^n \) such that messages \( m_j \) with \( j < i \) behave exactly in the same way in both runs. Since \( r_1 \in C_i \), we know from the definition of \( C_i \) that in \( r_1 \) no messages \( m_j \) with \( j \geq i \) is received by time \( \tau_i \). Since \( r_2 \in A_i^n \), we know from the definition of \( A_i^n \) and Proposition 4.3 (1) that in \( r_2 \) no messages \( m_j \) with \( j \geq i \) is received by time \( \tau_i \). Thus in both runs \( r_1 \) and \( r_2 \), the failure detector outputs up to time \( \tau_i \) are the same. Therefore, in \( r_1 \) the \( n \)-th S-transition occurs at \( \tau_i \) just as in \( r_2 \), which means \( r_1 \in A_i^n \). Thus Claim 1 holds.

Claim 2. For all \( n, n' \geq 1 \), for all \( i, i' \geq 2 \), if \( Pr(A_i^n) > 0 \) and \( Pr(A_{i'}^{n'}) > 0 \), then for all \( x, y \in [-\infty, \infty] \),

\[
Pr(T_{MR,n+1} \leq x, T_{M,n+1} \leq y \mid A_i^n) = Pr(T_{MR,n'+1} \leq x, T_{M,n'+1} \leq y \mid A_{i'}^{n'}). \tag{4.1}
\]

Proof of Claim 2. Suppose \( Pr(A_i^n) > 0 \) and \( Pr(A_{i'}^{n'}) > 0 \). Let \( t_T \) and \( t_S \) be
two random variables representing the times at which the first T-transition and S-transition occur after time $\tau_i$, respectively. Since $A_i^n$ represents the event that the $n$-th S-transition occurs at time $\tau_i$, we have $Pr(T_{mr,n+i} \leq x, T_{m,n+i} \leq y \mid A_i^n) = Pr(t_S - \tau_i \leq x, t_T - \tau_i \leq y \mid A_i^n)$. Let $D_i$ be the event $\{t_S - \tau_i \leq x, t_T - \tau_i \leq y\}$. Equality (4.1) is thus equivalent to $Pr(D_i \mid A_i^n) = Pr(D_i \mid A_i^{n'})$.

By Lemma 4.1, the output of the failure detector after $\tau_i$ is completely determined by messages $m_j$ with $j \geq i$. Thus we know that $D_i$ is completely determined by messages $m_j$ with $j \geq i$. Since $C_i$ is completely determined by messages $m_j$ with $j \geq i$ while $B_i^n$ is completely determined by messages $m_j$ with $j < i$, we have that both $C_i$ and $C_i \cap D_i$ are independent of $B_i^n$. We claim that $Pr(D_i \mid A_i^n) = Pr(D_i \mid C_i)$. Indeed,

$$Pr(D_i \mid A_i^n) = Pr(D_i \mid B_i^n \cap C_i) = \frac{Pr(D_i \cap B_i^n \cap C_i)}{Pr(B_i^n \cap C_i)} = \frac{Pr(D_i \cap B_i^n \cap C_i)}{Pr(B_i^n) \cap Pr(C_i)} = Pr(D_i \mid C_i).$$

Similarly we have $Pr(D_i \mid A_i^{n'}) = Pr(D_i \mid C_{i'})$. Thus we only need to show that $Pr(D_i \mid C_i) = Pr(D_i \mid C_{i'})$.

$Pr(D_i \mid C_i)$ is the probability that, given that no messages $m_j$ with $j \geq i$ is received by time $\tau_i$, the first S-transition after $\tau_i$ occurs within $x$ time units after $\tau_i$ and the first T-transition after $\tau_i$ occurs within $y$ time units. Since the occurrences of these transitions are all determined by messages $m_j$ with $j \geq i$, and messages are sent at regular intervals, it is easy to verify that this probability is the same for every $i \geq 2$. Thus $Pr(D_i \mid C_i) = Pr(D_i \mid C_{i'})$, and Claim 2 holds.
Claim 3. The pairs \((T_{MR,n}, T_{M,n})\), \(n \geq 2\) are identically distributed.

Proof of Claim 3. This is a direct consequence of Claim 2. In fact, for all \(n \geq 2\),
for all \(x, y \in [-\infty, \infty]\), let \(p(x, y) = Pr(T_{MR,n} \leq x, T_{M,n} \leq y \mid A_{i}^{n-1})\) if \(Pr(A_{i}^{n-1}) > 0\).
This is well-defined by Claim 2. Then

\[
Pr(T_{MR,n} \leq x, T_{M,n} \leq y) = \sum_{i=2}^{\infty} Pr(T_{MR,n} \leq x, T_{M,n} \leq y \mid A_{i}^{n-1}) Pr(A_{i}^{n-1})
\]

\[
= \sum_{Pr(A_{i}^{n-1}) > 0} p(x, y) Pr(A_{i}^{n-1}) = p(x, y).
\]

Claim 4. For all \(n \geq 1\) and \(i \geq 2\), \((T_{MR,n+1}, T_{M,n+1})\) is independent of \(A_{i}^{n}\).

Proof of Claim 4. This is another direct consequence of Claim 2. Suppose that we
fix \(i\) and \(n\) and \(Pr(A_{i}^{n}) > 0\). Then we have for all \(x, y \in [-\infty, \infty]\),

\[
Pr(T_{MR,n+1} \leq x, T_{M,n+1} \leq y) = \sum_{j=2}^{\infty} Pr(T_{MR,n+1} \leq x, T_{M,n+1} \leq y \mid A_{j}^{n}) Pr(A_{j}^{n})
\]

\[
= \sum_{Pr(A_{j}^{n}) > 0} Pr(T_{MR,n+1} \leq x, T_{M,n+1} \leq y \mid A_{j}^{n}) Pr(A_{j}^{n})
\]

\[
= Pr(T_{MR,n+1} \leq x, T_{M,n+1} \leq y \mid A_{i}^{n}).
\]

This shows that \((T_{MR,n+1}, T_{M,n+1})\) is independent of \(A_{i}^{n}\).

Claim 5. For all \(n \geq 1\) and for all \(i \geq 2\), when conditioned on \(A_{i}^{n}\), \((T_{MR,n+1}, T_{M,n+1})\)
is independent of \(\{(T_{MR,j}, T_{M,j}), j \leq n\}\).

Proof of Claim 5. We already know that when conditioned on \(A_{i}^{n}\), \((T_{MR,n+1}, T_{M,n+1})\)
is completely determined by the distribution of messages \(m_{j}\) with \(j \geq i\). On the other
hand, when conditioned on \(A_{i}^{n}\), the occurrence of any S-transition or T-transition
before \(\tau_{i}\) is only determined by messages \(m_{j}\) with \(j < i\), because \(A_{i}^{n}\) implies that
all messages \(m_{j}\) with \(j \geq i\) do not arrive at \(q\) by time \(\tau_{i}\). Since the occurrences
of transitions before and after \(\tau_{i}\) are determined by disjoint set of messages, and
messages are independent of each other, Claim 5 holds.
From Claims 4 and 5 and Proposition 4.9, we know that \((T_{MR,n+i}, T_{M,n+i})\) is independent of \((T_{MR,j}, T_{M,j}), j \leq n\). Thus pairs \((T_{MR,n}, T_{M,n}), n \geq 1\) are mutually independent. From Claim 3, we know that \((T_{MR,n}, T_{M,n}), n \geq 2\) are identically distributed. Therefore, \{\((T_{MR,n}, T_{M,n}), n = 1, 2, \ldots\)\} is a delayed renewal reward process.

It is immediate from the above lemma that for all \(n \geq 2\), \(T_{MR} = T_{MR,n}, T_{M} = T_{M,n}\) and \(T_G = T_{MR,n} - T_{M,n}\). This provides more direct ways to analyze the distributions of these variables. Moreover, any delayed renewal reward process is ergodic (see for example Section 2.6 of [Sig95]), so Theorem 2.1 of Chapter 2 is applicable to our failure detector.

**Proof of Lemma 4.7.** For all \(i \geq 2\), let \(A_i\) be the event that an S-transition occurs at time \(\tau_i\). By definition and Proposition 4.4, we have that \(Pr(A_i) = p_s = q_0 \cdot u(0) \geq q_0 \cdot p_0^k\). Since in the nondegenerated case \(q_0 > 0\) and \(p_0 > 0\), we have \(Pr(A_i) > 0\).

By Proposition 4.3 (3), \(A_i\) is also the event that \(m_{i-1}\) is received before time \(\tau_i\) but no message \(m_j\) with \(i \leq j < i + k\) is received by time \(\tau_i\). This implies that \(A_i\) only depends on messages \(m_j\) with \(i - 1 \leq j < i + k\), which in turn implies that for every \(j \in \{2, \ldots, k + 2\}\), events \(A_i(k+1)+j, i \geq 0\) are independent.

For \(j \in \{2, \ldots, k + 2\}\), let \(B_j\) be the set of time points \(\{\tau_{i(k+1)+j} : i = 0, 1, \ldots\}\). Obvious \(B_j, j \geq 0\) is a partition of all time points \(\tau_i, i \geq 2\). Let \(N_j(t)\) be the random variable representing the number of S-transitions that occur at times in \(B_j\) by time \(t\). Let \(N(t)\) be the random variable representing the number of S-transitions by time \(t\). Thus \(N(t) = \sum_{j=2}^{k+2} N_j(t)\).

Consider \(N_j(t)\) for some \(j \in \{2, \ldots, k + 2\}\). For \(t \geq \tau_j\), the number of time points in \(B_j\) that are no greater than \(t\) is \(\lfloor (t - \tau_j)/(k + 1)\eta \rfloor + 1\). From the above, we know that at each of these time points, there is an independent probability of \(p_s\) that
an S-transition occurs. Therefore, the average number of S-transitions at these time points by time $t \geq \tau_j$ is given by

$$E(N_j(t)) = p_S \left( \left\lfloor \frac{t - \tau_j}{(k+1)\eta} \right\rfloor + 1 \right).$$

Hence, we have for $t \geq \tau_{k+2}$,

$$E(N(t)) = \sum_{j=2}^{k+2} p_S \left( \left\lfloor \frac{t - \tau_j}{(k+1)\eta} \right\rfloor + 1 \right).$$

By Lemma 4.8, $\{T_{MR,n}, n \geq 1\}$ is a delayed renewal process. Then by the Elementary Renewal Theorem (see for example [Ros83] p.61),

$$E(T_{MR}) = \lim_{t \to \infty} \frac{t}{E(N(t))}$$

$$= \lim_{t \to \infty} \frac{t}{\sum_{j=2}^{k+2} p_S \left( \left\lfloor \frac{t - \tau_j}{(k+1)\eta} \right\rfloor + 1 \right)}$$

$$= \frac{1}{\sum_{j=2}^{k+2} \frac{p_S}{(k+1)\eta}} \cdot \frac{\eta}{p_S}$$

By Lemma 4.7, we know that $0 < E(T_{MR}) < \infty$. Then we can apply Theorem 2.1 of Chapter 2 to obtain results on other metrics by our results on $P_A$ and $E(T_{MR})$.

The above is the analysis on the accuracy metrics of the new failure detector. We now give the bound on the worst-case detection time $T_D$.

**Lemma 4.10** $T_D \leq \delta + \eta$. Moreover, the inequality is tight when $q_0 > 0$, and $T_D$ is always 0 when $q_0 = 0$.

**Proof.** Suppose that process $p$ crashes at time $t$. Let $m_i$ be the last heartbeat message sent by $p$ before $p$ crashes. By definition, $m_i$ is sent at time $\sigma_i$, and $\sigma_i \leq t$. 
Since no messages with sequence number greater than $i$ are sent by $p$, $q$ does not receive these messages. Thus by Lemma 4.1, for all $t' \in [\tau_{i+1}, \infty)$, $q$ suspects $p$ at time $t'$. So the detection time is at most $\tau_{i+1} - t = \sigma_i + \delta + \eta - t \leq \delta + \eta$.

When $q_0 > 0$, with nonzero probability $m_i$ is received before $\tau_{i+1}$ and thus $q$ trusts $p$ just before $\tau_{i+1}$. In these cases, the detection time is $\sigma_i + \delta + \eta - t$. Since the time $t$ when $p$ crashes can be arbitrarily close to $\sigma_i$, we have that the bound $\delta + \eta$ is tight. When $q_0 = 0$, similar to Lemma 4.5 (2), we can see that in runs in which $p$ crashes $q$ also suspects $p$ forever. Therefore $T_D$ is always 0. \hfill $\Box$

All the above analytical results are summarized in Theorem 4.11.

**Theorem 4.11** The failure detector NFD-S given in Fig. 4.2 has the following properties:

1. $T_D \leq \delta + \eta$. Moreover, if $q_0 > 0$, then the inequality is tight, and if $q_0 = 0$, then $T_D$ is always 0.

2. If $p_0 > 0$ and $q_0 > 0$ (the nondegenerated case), then we have

$$E(T_{MR}) = \frac{1}{\lambda_M} = \frac{\eta}{p_s},$$

$$E(T_M) = (1 - P_A) \cdot E(T_{MR}) = \frac{\int_0^\eta u(x) \, dx}{p_s},$$

$$P_A = 1 - \frac{1}{\eta} \int_0^\eta u(x) \, dx,$$

$$E(T_G) = P_A \cdot E(T_{MR}) = \frac{\eta - \int_0^\eta u(x) \, dx}{p_s},$$

$$E(T_{FG}) \geq \frac{1}{2} E(T_{MR}) = \frac{\eta - \int_0^\eta u(x) \, dx}{2p_s}.$$

---

Even though $q_0$ is defined with respect to runs in which $p$ does not crash, it also applies to runs in which $p$ crashes by part (a) of the message independence property.
If \( p_0 = 0 \) or \( q_0 = 0 \) \((\text{the degenerated case})\), then we have: in failure-free runs, (a) if \( p_0 = 0 \), then with probability one \( q \) trusts \( p \) forever after time \( \tau_1 \); (b) if \( q_0 = 0 \), then with probability one \( q \) suspects \( p \) forever.

From these closed formulas, we can derive many useful properties of the QoS of the new failure detector. For example, we can derive bounds on the accuracy metrics \( E(T_{\text{MR}}) \), \( E(T_{\text{M}}) \), \( P_A \), \( E(T_{\text{G}}) \), and \( E(T_{\text{FG}}) \), as we later do in Theorem 4.17. From these bounds, it is easy to check that when \( \delta \) increases or \( \eta \) decreases, \( P_A \) increases exponentially fast towards 1, and \( E(T_{\text{MR}}) \), \( E(T_{\text{G}}) \) and \( E(T_{\text{FG}}) \) increases exponentially fast towards \( \infty \), while \( E(T_{\text{M}}) \) is bounded by a relative small value. The tradeoff is that: (a) when \( \delta \) increases, the detection time increases linearly; (b) when \( \eta \) decreases, the network bandwidth used by the failure detector increases linearly. Therefore, with a small (linear) increase in the detection time or in the network cost, we can get a large (exponential) increase in the accuracy of the new failure detector.

In Section 4.3.4, 4.4.1 and 4.4.3, we will show how these close formulas are used to compute the failure detector parameters to satisfy given QoS requirements.

### 4.3.3 An Optimality Result

Besides the properties given in Theorem 4.11, the new algorithm has the following important optimality property: among all failure detectors that send messages at the same rate and satisfy the same upper bound on the worst-case detection time, the new algorithm provides the best query accuracy probability.

More precisely, let \( \mathcal{C} \) be the class of failure detector algorithms \( A \) such that in
every run of $A$ process $p$ sends messages to $q$ every $\eta$ time units and $A$ satisfies $T_D \leq T_D^U$ for some constant $T_D^U$. Let $A^*$ be the instance of the new failure detector algorithm NFD-S with parameters $\eta$ and $\delta = T_D^U - \eta$ ($\delta$ can be negative). By part (1) of Theorem 4.11, we know that $A^* \in C$. We show that

**Theorem 4.12** For any $A \in C$, let $P_A$ be the query accuracy probability of $A$. Let $P^*_A$ be the query accuracy probability of $A^*$. Then $P^*_A \geq P_A$.

The core idea behind the theorem is the following important property of algorithm $A^*$: roughly speaking, if in some failure-free run $r$ of $A^*$ process $q$ suspects $p$ at time $t$, then for any $A \in C$, in any failure-free run $r'$ of $A$ in which the message delay and loss behaviors are exactly the same as in run $r$, $q$ also suspects $p$ at time $t$. With this property, it is easy to see that the probability that $q$ trusts $p$ at a random time in $A^*$ must be at least as high as the probability that $q$ trusts $p$ at a random time in any $A \in C$. We now give the more detailed proof.

A message delay pattern $P_D$ is a sequence $\{d_1, d_2, d_3, \ldots\}$ with $d_i \in (0, \infty]$ representing the delay time of the $i$-th message sent by $p$; $d_i = \infty$ means that the $i$-th message is lost. The distribution of message delay patterns are governed by the message loss probability $p_L$ and the message delay time $D$, and thus it is the same for all algorithms in $C$.

We first consider a subclass $C'$ of $C$ such that for any algorithm $A \in C'$, in any run of $A$ process $p$ sends messages to $q$ at times $\eta, 2\eta, 3\eta, \ldots$, just as in $A^*$. For any algorithm in $C'$, a message delay pattern completely determines the time and the order at which $q$ receives messages in failure-free runs. For $A^*$, this means that a message delay pattern uniquely determines a failure-free run of $A^*$. For some other
algorithm $A \in C'$, if $A$ is nondeterministic, then $A$ may have different failure-free runs with the same message delay pattern.

**Lemma 4.13** Given any message delay pattern $P_D$, let $r^*$ be the failure-free run of $A^*$ with $P_D$, and let $r$ be a failure-free run of some algorithm $A \in C'$ with $P_D$. Then for every time $t \geq T_D^U$, if $q$ suspects $p$ at time $t$ in run $r^*$, then $q$ suspects $p$ at time $t$ in run $r$.

**Proof.** Suppose that in run $r^*$ of $A^*$, $q$ suspects $p$ at time $t \geq T_D^U$. Note that $T_D^U = \eta + \delta = \tau_1$, so $t \geq \tau_1$. Suppose $t \in [\tau_i, \tau_{i+1})$ for some $i \geq 1$. By Lemma 4.1, in run $r^*$ $q$ does not receive any message $m_j$ with $j \geq i$ by time $t$. Since in run $r$ $p$ sends messages at the same times as in run $r^*$, and both runs have the same message delay pattern $P_D$, then in run $r$, by time $t$ $q$ does not receive any message sent by $p$ at time $j\eta$ with $j \geq i$.

Consider first that $t \in (\tau_i, \tau_{i+1})$. Suppose for a contradiction that $q$ trusts $p$ at time $t$ in run $r$. Let $\epsilon = t - \tau_i$, and let $t' = (i-1)\eta + \epsilon/2$. Thus $\epsilon > 0$. Consider another run $r'$ of $A$ in which $p$ crashes at time $t'$, and messages sent before $t'$ (those sent at times $j\eta$ with $j < i$) have the same loss and delay behaviors as in run $r$ (this is possible by part (a) of the message independence property). In both runs $r$ and $r'$ up to time $t$, $q$ receives the same messages at the same times. If $A$ is nondeterministic, we let $A$ make the same nondeterministic choices up to time $t$ in both runs. Thus $q$ cannot distinguish run $r'$ from $r$ at time $t$, and so $q$ trusts $p$ at time $t$ in run $r'$. The detection time in run $r'$, however, is at least $t - t' = (\tau_i + \epsilon) - ((i-1)\eta + \epsilon/2) = \eta + \delta + \epsilon/2 = T_D^U + \epsilon/2 > T_D^U$, contradicting the assumption that $A$ satisfies $T_D \leq T_D^U$. 
Now suppose \( t = \tau_i \). Since the failure detector output is right continuous, there exists \( \epsilon > 0 \) such that \( q \) suspects \( p \) in the period \((t, t + \epsilon)\) in run \( r^* \). Then by the above argument, \( q \) suspects \( p \) in the period \((t, t + \epsilon)\) in run \( r \). By the right continuity again, we have that \( q \) suspects \( p \) at time \( t \) in run \( r \).

**Corollary 4.14** For any \( A \in C' \), let \( P_A \) be the query accuracy probability of \( A \). Let \( P_A^* \) be the query accuracy probability of \( A^* \). Then \( P_A^* \geq P_A \).

**Proof (Sketch).** We first fix a message delay pattern \( P_D \). For the run \( r^* \) of \( A^* \) and any run \( r \) of \( A \) with message delay pattern \( P_D \), Lemma 4.13 shows that for any time \( t \geq T_D^U \), if \( q \) suspects \( p \) in \( r^* \) at time \( t \), then \( q \) suspects \( p \) in \( r \) at time \( t \). Thus, given a fixed message delay pattern \( P_D \), at any random time \( t \), the probability that \( q \) trusts \( p \) at time \( t \) in algorithm \( A^* \) is at least as high as the probability that \( q \) trusts \( p \) at time \( t \) in algorithm \( A \). So \( P_A^* \geq P_A \) given a fixed message delay pattern \( P_D \). When summing (or integrating) both sides of the inequality over all message delay patterns according to their distribution, we have \( P_A^* \geq P_A \).

The above corollary shows that the new algorithm \( A^* \) has the best query accuracy probability in \( C' \), the class of failure detector algorithms in which \( p \) sends messages at exactly the same times as in \( A^* \). We now generalize this result to class \( C \), where \( p \) still sends messages every \( \eta \) time units, but it may do so at times different from those in \( A^* \).

A **message sending pattern** \( P_S \) is a sequence of time points \( \{\sigma_1, \sigma_2, \sigma_3, \ldots\} \) at which \( p \) sends messages. The message sending pattern is determined by the algorithm. For any algorithm \( A \in C \), its message sending pattern is in the form \( \{s, s + \eta, s + 2\eta, \ldots\} \) for some \( s \in [0, \infty) \). Different runs of algorithm \( A \) may have different message sending patterns due to the possible nondeterminism of \( A \). Let \( A_s^* \) be the algorithm in which
p sends heartbeat messages according to the sending pattern \(\{s, s + \eta, s + 2\eta, \ldots\}\), and \(q\) behaves the same way as in \(A^*\). Thus \(A^*_s\) is a shifted version of \(A^*\), and so the behavior of the failure detector output in \(A^*_s\) is also a shifted version of that of \(A^*\). Since the behaviors of the two failure detectors only differ in some initial period, their steady state behaviors are the same. Therefore the QoS metrics of \(A^*_s\) and \(A^*\) are the same. In particular, they have the same query accuracy probability.

**Proof of Theorem 4.12 (Sketch).** We first fix a message sending pattern \(P_S = \{s, s + \eta, s + 2\eta, \ldots\}\). For any algorithm \(A \in C\), consider the runs of \(A\) with the sending pattern \(P_S\). In these runs \(p\) sends messages at exactly the same times as in algorithm \(A^*_s\). By the similar argument as in Lemma 4.13 and Corollary 4.14, we can show that the query accuracy probability of \(A^*_s\) is at least as high as the query accuracy probability of \(A\) given the message sending pattern \(P_S\). Since \(A^*_s\) and \(A^*\) have the same query accuracy probability, we have \(P^*_A \geq P_A\) given the message sending pattern \(P_S\). Since \(P_S\) is arbitrary, we thus have \(P^*_A \geq P_A\). \(\square\)

### 4.3.4 Configuring the Failure Detector to Satisfy QoS Requirements

Suppose we are given a set of failure detector QoS requirements (these QoS requirements could be given by an application). We now show how to compute the parameters \(\eta\) and \(\delta\) of the failure detector algorithm, so that these requirements are satisfied. We first assume that (a) the local clocks of processes are synchronized, and (b) one knows the probabilistic behavior of the messages, i.e., the message loss probability \(p_L\) and the distribution of message delays \(Pr(D \leq x)\). In Section 4.4, we
show how to remove these assumptions.

We assume that the QoS requirements are expressed using the primary metrics. More precisely, a set of QoS requirements is a tuple \((T_D^U, T_{MR}^L, T_M^U)\), where \(T_D^U\) is an upper bound on the worst-case detection time, \(T_{MR}^L\) is a lower bound on the average mistake recurrence time, and \(T_M^U\) is an upper bound on the average mistake duration. In other words, the requirements are that:

\[
T_D \leq T_D^U, \quad E(T_{MR}) \geq T_{MR}^L, \quad E(T_M) \leq T_M^U. \tag{4.7}
\]

In addition, we would like to have \(\eta\) as large as possible, to save network bandwidth. Using Theorem 4.11, this can be stated as a mathematical programming problem:

\[
\begin{align*}
\text{maximize} \quad & \eta \\
\text{subject to} \quad & \delta + \eta \leq T_D^U \tag{4.8} \\
& \eta \geq T_{MR}^L \tag{4.9} \\
& \int_0^\eta u(x) \, dx \leq T_M^U \tag{4.10} \\
& \eta \geq \Delta \tag{4.11}
\end{align*}
\]

where the values of \(u(x)\) and \(p_s\) are given by Proposition 4.2. Constraint (4.11) ensures that the heartbeat messages are independent, so that Theorem 4.11 can be applied. Computing the optimal solution for this problem, which means finding the largest \(\eta\) and some \(\delta\) that satisfy constraints (4.8)–(4.11), seems to be hard. Instead, we give a simple procedure that computes \(\eta\) and \(\delta\) such that they satisfy

\footnote{Note that the bounds on the primary metrics \(E(T_{MR})\) and \(E(T_M)\) also impose bounds on the derived metrics, according to Theorem 2.1 of Chapter 2. More precisely, we have \(\lambda_M \leq 1/T_{MR}^L\), \(P_A \geq (T_{MR}^L - T_M^U)/T_{MR}^L\), \(E(T_{MR}) \geq T_{MR}^L - T_M^U\), and \(E(T_{FG}) \geq (T_{MR}^L - T_M^U)/2\).}
constraints (4.8)–(4.11), but $\eta$ may not be the largest possible value. This is done by replacing constraint (4.10) with a simpler and stronger constraint to obtain a modified problem, and computing the optimal solution of this modified problem. The configuration procedure is as follows:

- **Step 1:** Compute $q'_0 = (1 - p_L) Pr(D < T^U_D)$, and let $\eta_{\text{max}} = q'_0 T^U_M$.

- **Step 2:** Let
  \[
  f(\eta) = \frac{\eta}{q'_0 \prod_{j=1}^{\lceil T^U_D/\eta \rceil - 1} [p_L + (1 - p_L) Pr(D > T^U_D - j\eta)]}
  \]  
  (4.12)

  Find the largest $\eta \leq \eta_{\text{max}}$ such that $f(\eta) \geq T^L_{MR}$.

  It is easy to check that when $\eta$ decreases, $f(\eta)$ increases exponentially fast towards infinity, so some simple numerical method (such as binary search) can be used to calculate $\eta$.

- **Step 3:** If $\eta \geq \Delta$, then set $\delta = T^U_D - \eta$; otherwise, the procedure does not find appropriate $\eta$ and $\delta$.

We now show that the parameters computed by the procedure are appropriate.

**Proposition 4.15** If $p_0 > 0$ and $q_0 > 0$ (the nondegenerated case), then $E(T^*_M) \leq \eta/q_0$.

**Proof.** By Proposition 4.4, we have for all $x \in [0, \eta)$, $u(0) \geq u(x)$. Thus by equality (4.3), we have

\[
E(T^*_M) = \frac{\int_0^\eta u(x) \, dx}{p_s} \leq \frac{\int_0^\eta u(0) \, dx}{q_0 u(0)} = \frac{\eta}{q_0}.
\]

\qed
Theorem 4.16 Consider a system in which clocks are synchronized, and the probabilistic behavior of messages is known. With the parameters $\eta$ and $\delta$ computed by the above procedure, the failure detector algorithm NFD-S of Fig. 4.2 satisfies the QoS requirements given in (4.7).

Proof. Suppose that the procedure finds parameters $\eta$ and $\delta$. Then by step 3 we have $T_D^U = \eta + \delta$. By part (1) of Theorem 4.11, $T_D \leq T_D^U$ is satisfied. By step 1 and Proposition 4.2, $q'_0 = (1 - p_L)Pr(D < \eta + \delta) = q_0$ (note that $q_0 > 0$: otherwise $\eta_{\text{max}} = 0$ and the procedure cannot find $\eta \geq \Delta$). Consider first that $p_0 > 0$. Then by Proposition 4.15, $E(T_M) \leq \eta/q_0 \leq \eta_{\text{max}}/q_0 = q_0T_M^U/q_0 = T_M^U$. So $E(T_M) \leq T_M^U$ is satisfied. Note that

$$\prod_{j=1}^{[T_D^U/\eta]-1} [p_L + (1 - p_L)Pr(D > T_D^U - j\eta)]$$

$$= \prod_{j=1}^{[(\eta+\delta)/\eta]-1} [p_L + (1 - p_L)Pr(D > \eta + \delta - j\eta)]$$

$$= \prod_{j=0}^{[\delta/\eta]-1} [p_L + (1 - p_L)Pr(D > \delta - j\eta)]$$

$$= \prod_{j=0}^{[\delta/\eta]} [p_L + (1 - p_L)Pr(D > \delta - j\eta)] = u(0).$$

Thus $f(\eta) = \eta/(q_0u(0)) = \eta/p_S = E(T_{MR})$, by equality (4.2). By step 2, $f(\eta) \geq T_M^L$, and so $E(T_{MR}) \geq T_M^L$ is satisfied.

Consider now that $p_0 = 0$. By Theorem 4.11, in failure-free runs, the failure detector keeps trusting $p$ after time $\tau_1$, and so $E(T_{MR}) = \infty$ and $E(T_M) = 0$. Thus the requirements in (4.7) are also satisfied. \qed
4.4 Dealing with Unknown System Behavior and Unsynchronized Clocks

So far, we assumed that (a) the local clocks of processes are synchronized, and (b) the probabilistic behavior of the messages (i.e., probability of message loss and distribution of message delays) is known. These assumptions are not unrealistic, but in some systems assumption (a) or (b) may not hold. To widen the applicability of our algorithm, we now show how to remove these assumptions.

4.4.1 Configuring the Failure Detector NFD-S When the Probabilistic Behavior of the Messages is Not Known

In Section 4.3.4, our procedure of computing $\eta$ and $\delta$ to meet some QoS requirements assumed that one knows the probabilistic behavior of the messages (i.e., probability $p_L$ of message loss and the probability distribution $Pr(D \leq x)$ of the message delay). If this probabilistic behavior is not known, we can still compute $\eta$ and $\delta$ as follows:

We first assume that message loss probability $p_L$, the expected value $E(D)$ and the variance $V(D)$ of message delay $D$ are known, and show how to compute $\eta$ and $\delta$ with only $p_L$, $E(D)$ and $V(D)$. We then show how to estimate $p_L$, $E(D)$ and $V(D)$ using heartbeat messages. Note that in this section we still assume that local clocks are synchronized.

With $E(D)$ and $V(D)$, we have an upper bound on $Pr(D > x)$, as given by the following One-Sided Inequality of probability theory (e.g., see [All90] p. 79): For any
random variable $D$ with a finite expected value and a finite variance,

$$Pr(D > x) \leq \frac{V(D)}{V(D) + (x - E(D))^2}, \text{ for all } x > E(D). \quad (4.13)$$

With the One-Sided Inequality, we derive the following bounds on the QoS metrics of algorithm NFD-S.

**Theorem 4.17** Assume $\delta > E(D)$. For algorithm NFD-S, in the nondegenerated cases of Theorem 4.11, we have

$$E(T_{mr}) \geq \frac{\eta}{\beta}, \quad (4.14)$$

$$E(T_m) \leq \frac{\eta}{\gamma}, \quad (4.15)$$

$$P_\delta \geq 1 - \frac{\beta}{\gamma}, \quad (4.16)$$

$$E(T_\sigma) \geq \frac{1 - \beta}{\beta} \eta, \quad (4.17)$$

$$E(T_{rc}) \geq \frac{1 - \beta}{2\beta} \eta, \quad (4.18)$$

where

$$\beta = \prod_{j=0}^{k_0} \frac{V(D) + p_L(\delta - E(D) - j\eta)^2}{V(D) + (\delta - E(D) - j\eta)^2}, \quad k_0 = \lceil (\delta - E(D))/\eta \rceil - 1,$$

and

$$\gamma = \frac{(1 - p_L)(\delta - E(D) + \eta)^2}{V(D) + (\delta - E(D) + \eta)^2}.$$

**Proof.** Note that for all $j$ such that $0 \leq j \leq k_0$, $\delta - j\eta > E(D)$. Then by the One-Sided Inequality (4.13), we have for all $j$ such that $0 \leq j \leq k_0$,

$$p_j(0) = p_L + (1 - p_L)Pr(D > \delta - j\eta) \leq p_L + (1 - p_L) \frac{V(D)}{V(D) + (\delta - E(D) - j\eta)^2} = \frac{V(D) + p_L(\delta - E(D) - j\eta)^2}{V(D) + (\delta - E(D) - j\eta)^2}.$$
Thus, $\prod_{j=0}^{k_0} p_j(0) \leq \beta$.

By Proposition 4.2 (4) and (5) and Proposition 4.4, and the fact that $k_0 \leq k - 1$, we have $u(x) \leq u(0) \leq \prod_{j=0}^{k_0} p_j(0) \leq \beta$, and $p_s \leq u(0) \leq \beta$. Therefore, from equality (4.2), $E(T_{m}) = \eta/p_s \geq \eta/\beta$. Similarly, when applying $u(x) \leq \beta$ and $p_s \leq \beta$ to equalities (4.4), (4.5) and (4.6), we obtain inequalities (4.16), (4.17) and (4.18), respectively.

To show inequality (4.15), first note that we can replace $Pr(D > x)$ in the One-Sided Inequality (4.13) with $Pr(D \geq x)$, and the inequality remains true. In fact, for all $\epsilon \in (0, x - E(D))$,

$$Pr(D \geq x) \leq Pr(D > x - \epsilon) \leq \frac{V(D)}{V(D) + (x - \epsilon - E(D))^2}.$$  

Let $\epsilon \to 0$, and we obtain the result.

Then from Proposition 4.2 (3) we have

$$q_0 = (1 - p_L)(1 - Pr(D \geq \delta + \eta)) \geq (1 - p_L)\left(1 - \frac{V(D)}{V(D) + (\delta - E(D) + \eta)^2}\right) = \gamma.$$  

Therefore, by Proposition 4.15, $E(T_m) \leq \eta/q_0 \leq \eta/\gamma$. □

Note that in Theorem 4.17 we assume that $\delta > E(D)$. This assumption is reasonable because if the parameter $\delta$ of NFD-S is set to be less than $E(D)$, then there will be a false suspicion every time the heartbeat message takes more than the average message delay, and so a failure detector with such a $\delta$ makes very frequent mistakes and is not useful.

**Computing Failure Detector Parameters $\eta$ and $\delta$**

The bounds given in Theorem 4.17 can be used to compute the parameters $\eta$ and $\delta$ of the failure detector NFD-S, so that it satisfies the QoS requirements given
in (4.7). The configuration procedure is given below. This procedure assumes that $T_D^{U} > E(D)$, i.e., the required detection time is greater than the average message delay, which is a reasonable assumption.

- **Step 1**: Compute \( \gamma' = (1 - p_L) (T_D^{U} - E(D))^2 / (V(D) + (T_D^{U} - E(D))^2) \) and let \( \eta_{\text{max}} = \min(\gamma' T_M^{U}, T_D^{U} - E(D)) \).

- **Step 2**: Let

\[
  f(\eta) = \eta \cdot \left[ \frac{(T_D^{U} - E(D))/\eta}{\prod_{j=1}^{V(D)} (V(D) + (T_D^{U} - E(D) - j\eta))^2} \right]^{-1}.
\]

Find the largest \( \eta \leq \eta_{\text{max}} \) such that \( f(\eta) \geq T_{L}^{L} \).

- **Step 3**: If \( \eta \geq \Delta \), then set \( \delta = T_D^{U} - \eta \); otherwise, the procedure does not find appropriate \( \eta \) and \( \delta \).

**Theorem 4.18** Consider a system in which clocks are synchronized, and the probabilistic behavior of messages is not known. With parameters \( \eta \) and \( \delta \) computed by the above procedure, the failure detector algorithm NFD-S of Fig. 4.2 satisfies the QoS requirements given in (4.7), provided that $T_D^{U} > E(D)$.

The proof of the theorem is straightforward.

**Estimating \( p_L \), \( E(D) \) and \( V(D) \)**

The configuration procedure given above assumes that \( p_L \), \( E(D) \) and \( V(D) \) are known. In practice, we can use heartbeat messages to compute close estimates of these quantities, as we now explain.
Estimating \( P_L \) is easy. For example, one can use the sequence numbers of the heartbeat messages to count the number of “missing” heartbeats, and then divide this count by the highest sequence number received so far.

Since local clocks are synchronized, \( E(D) \) and \( V(D) \) can also be easily estimated using the heartbeat messages of the algorithm. Suppose that when \( p \) sends a heartbeat \( m \), \( p \) timestamps \( m \) with the sending time \( S \), and when \( q \) receives \( m \), \( q \) records the receipt time \( A \). Thus the delay of message \( m \) is \( A - S \). Therefore, by taking the average and the variance of \( A - S \) of heartbeat messages, we obtain the estimates of \( E(D) \) and \( V(D) \).

### 4.4.2 Dealing with Unsynchronized Clocks

The algorithm NFD-S in Fig. 4.2 assumes that the local clocks are synchronized, so that \( q \) can set the freshness points \( \tau_i \)'s by shifting the sending times of the heartbeats by a constant. If the local clocks are not synchronized, \( q \) cannot set the \( \tau_i \)'s in this way. To circumvent this problem, we modify the algorithm so that \( q \) obtains the \( \tau_i \)'s by shifting the expected arrival times of the heartbeats, as we now explain.

We assume that local clocks do not drift with respect to real time, i.e., they accurately measure time intervals. Let \( \sigma_i \) denote the sending time of \( m_i \) with respect to \( q \)'s local clock time. Then, the expected arrival time \( EA_i \) of \( m_i \) at \( q \) is \( EA_i = \sigma_i + E(D) \), where \( E(D) \) is the expected message delay. We will show shortly how \( q \) can accurately estimate the \( EA_i \)'s by using past heartbeat messages.

Suppose for the moment that \( q \) knows the \( EA_i \)'s. Then \( q \) can set \( \tau_i \)'s by shifting the \( EA_i \)'s forward in time by \( \alpha \) time units, i.e., \( \tau_i = EA_i + \alpha \) (where \( \alpha \) is a new failure detector parameter replacing \( \delta \)). We denote the algorithm with this modification as
Process \( p \): \{using \( p \)'s local clock time\}

for some constant \( \eta \), send to \( q \) heartbeat messages \( m_1, m_2, m_3, \ldots \) at regular time points \( \eta, 2\eta, 3\eta, \ldots \) respectively;

Process \( q \): \{using \( q \)'s local clock time\}

 Initialization:

for all \( i \geq 1 \), set \( \tau_i = EA_i + \alpha \); \{\( EA_i \) is the expected arrival time of \( m_i \)\}  
\( output = S; \) \{suspect \( p \) initially\}

at every freshness point \( \tau_i \):

if no message \( m_j \) with \( j \geq i \) has been received then

\( output \leftarrow S; \) \{suspect \( p \) if no fresh message is received\}

upon receive message \( m_j \) at time \( t \in [\tau_i, \tau_{i+1}) \):

if \( j \geq i \) then \( output \leftarrow T; \) \{trust \( p \) when some fresh message is received\}

Figure 4.3: The new failure detector algorithm NFD-U, with unsynchronized clocks and known expected arrival times, and with parameters \( \eta \) and \( \alpha \)

NFD-U, and it is given in Fig. 4.3. Intuitively, \( EA_i \) is the time when \( m_i \) is expected to be received, and \( \alpha \) is a slack added to \( EA_i \) to accommodate the possible extra delay of message \( m_i \). Thus an appropriately set \( \alpha \) gives a high probability that \( q \) receives \( m_i \) before the freshness point \( \tau_i \), so that there is no failure detector mistake in the period \( [\tau_i, \tau_{i+1}) \) (see Fig. 4.1 (a)). If \( \alpha \) is large enough, it also allows subsequent messages \( m_{i+1}, m_{i+2}, \ldots \) to be received before time \( \tau_i \), so that there is no failure detector mistake in \( [\tau_i, \tau_{i+1}) \) even if \( m_i \) is lost. Of course \( \alpha \) cannot be too large because it adds to the detection time.

Note that in algorithm NFD-U, \( \tau_i = \sigma_i + E(D) + \alpha \). Therefore, it is easy to see that if we let \( \delta = E(D) + \alpha \), and consider all times referred in the analysis of the algorithm NFD-S to be with respect to \( q \)'s local clock time, then the analysis of NFD-S also applies to the algorithm NFD-U. In particular, the only changes of the
Process $p$: \{using $p$’s local clock time\}

for some constant $\eta$, send to $q$ heartbeat messages $m_1, m_2, m_3, \ldots$ at regular time points $\eta, 2\eta, 3\eta, \ldots$ respectively;

Process $q$: \{using $q$’s local clock time\}

Initialization:

\[ \tau_0 = 0; \]
\[ \ell = -1; \] \{$\ell$ keeps the largest sequence number in all messages $q$ received so far$\}$

upon $\tau_{\ell+1} =$ the current time:

\[ \text{if the current time reaches } \tau_{\ell+1}, \text{ then all messages received are stale} \]

\[ \text{output } \leftarrow S; \] \{suspect $p$ since all messages received are stale at this time$\}$

upon receive message $m_j$ at time $t$:

\[ \text{if } j > \ell \text{ then } \quad \{\text{received a message with a higher sequence number}\} \]
\[ \ell \leftarrow j; \]
\[ \text{compute } \hat{E}A_{\ell+1}; \quad \{\text{estimate the expected arrival time of } m_{\ell+1} \text{ using formula (4.24)}\} \]
\[ \tau_{\ell+1} \leftarrow \hat{E}A_{\ell+1} + \alpha; \]
\[ \text{if } t < \tau_{\ell+1} \text{ then output } \leftarrow T; \quad \{\text{trust } p \text{ since } m_{\ell} \text{ is still fresh at time } t\} \]

Figure 4.4: The new failure detector algorithm NFD-E, with unsynchronized clocks and estimated expected arrival times, and with parameters $\eta$ and $\alpha$

results are: (a) In Proposition 4.2, we now have

\[ k = \lceil (E(D) + \alpha) / \eta \rceil, \] 

(b) In part (1) of Theorem 4.11, the inequality is now

\[ T_D \leq E(D) + \alpha + \eta. \] 

Estimating the Expected Arrival Times

The expected arrival times can be estimated using heartbeat messages. The idea is to use the $n$ most recently received heartbeat messages to estimate the expected
arrival time of the next heartbeat message. To do so, we first modify the structure of the failure detector algorithm NFD-U in Fig. 4.3 to show exactly when q needs to estimate the expected arrival time of the next heartbeat.

The new version of the algorithm with estimated expected arrival times is given in Fig 4.4 and is denoted by NFD-E. In NFD-E, process q uses a variable ℓ to keep the largest heartbeat sequence number received so far, and τℓ+1 refers to the “next” freshness point. Note that when q updates ℓ, it also changes τℓ+1. If the local clock of q ever reaches time τℓ+1 (an event which might never happen), then at this time all the heartbeats received are stale, and so q starts suspecting p (lines 5–6). When q receives mj, it checks whether this is a new heartbeat (j > ℓ) and in this case, (1) q updates ℓ, (2) q computes the estimate \( \overline{E}A_{ℓ+1} \) of the expected arrival time of \( m_{ℓ+1} \) (the next heartbeat), (3) q sets the next freshness point τℓ+1 to \( \overline{E}A_{ℓ+1} + α \), and (4) q trusts p if the current time is less than τℓ+1 (lines 9–12).

Note that the algorithm NFD-E satisfies the same core property that q trusts p at time t if and only if some message that q received is still fresh at time t. Therefore, except the fact that it needs to estimate the expected arrival times, algorithm NFD-E is equivalent to algorithm NFD-U.

We now show how to estimate the expected arrival time of \( m_{ℓ+1} \) from the most recent n heartbeat messages that q received. Let \( m'_1, m'_2, \ldots, m'_n \) be these n messages. Note that \( m'_i \) is not \( m_i \) in general, and so the sequence number of \( m'_i \) is not necessarily i. Moreover, the sequence numbers of \( m'_1, m'_2, \ldots, m'_n \) may not be consecutive or monotonically increasing, because the heartbeat messages may be lost or received out of order. Let \( s_1, s_2, \ldots, s_n \) be the sequence numbers of \( m'_1, m'_2, \ldots, m'_n \) respectively. Let \( A'_i \) be the actual arrival time of \( m'_i \) with respect to q’s local clock time.
Let the expected arrival time of $m'_i$ at $q$ be $EA'_i$. Let $\epsilon_i = A'_i - EA'_i$. Thus $\epsilon_i$ is the deviation of the actual arrival time of $m'_i$ from its expected arrival time at $q$. Let $D_i$ be the actual delay time of message $m'_i$. Then we have $\epsilon_i = A'_i - EA'_i = D_i - E(D)$. For the expected arrival time $EA_{\ell+1}$ of $m_{\ell+1}$, we have that for every $i = 1, 2, \ldots, n$, $EA_{\ell+1} = EA'_i + (\ell + 1 - s_i)\eta = A'_i - \epsilon_i + (\ell + 1 - s_i)\eta$. By summing over all $i$'s on both side of the equality and then dividing both sides by $n$, we have

$$EA_{\ell+1} = \frac{1}{n} \sum_{i=1}^{n} A'_i - \frac{1}{n} \sum_{i=1}^{n} \epsilon_i + \frac{1}{n} \sum_{i=1}^{n} (\ell + 1 - s_i)\eta.$$ 

By choosing $\eta \geq \Delta$, we know that $D_i$'s are independent and identical to $D$. Thus $\frac{1}{n} \sum_{i=1}^{n} \epsilon_i = \frac{1}{n} \sum_{i=1}^{n} D_i - E(D)$, which is close to zero when $n$ is large. Therefore, we obtain the following estimator of $EA_{\ell+1}$:

$$\overline{EA}_{\ell+1} = \frac{1}{n} \sum_{i=1}^{n} A'_i + \frac{1}{n} \sum_{i=1}^{n} (\ell + 1 - s_i)\eta. \quad (4.24)$$

This is the formula used in line 10 of the algorithm NFD-E in Fig. 4.4 to compute the estimate of the expected arrival time of $m_{\ell+1}$.

How large the value of $n$ should be to obtain a reasonably good estimate? Note that $\overline{EA}_{\ell+1} - EA_{\ell+1} = \frac{1}{n} \sum_{i=1}^{n} \epsilon_i = \frac{1}{n} \sum_{i=1}^{n} D_i - E(D)$, where $D_i$'s are independent and identical to $D$. Thus the quality of the estimator $\overline{EA}_{\ell+1}$ is the same as the quality of the estimator $\frac{1}{n} \sum_{i=1}^{n} D_i$ for estimating $E(D)$. By the Sampling Theorem in statistics (see, e.g., [All90] p.432), we know that $\frac{1}{n} \sum_{i=1}^{n} D_i$ is an unbiased estimator of $E(D)$, and when $n$ is large, $\frac{1}{n} \sum_{i=1}^{n} D_i$ has approximately the normal distribution with mean $E(D)$ and standard deviation $\sigma(D)/\sqrt{n}$. When it is close to a normal distribution, about 95% of the estimated values are within $2\sigma(D)/\sqrt{n}$ away from the true value. The actual $n$ that makes the estimator close to a normal distribution depends on the distribution of $D$. A widely used rule of thumb is that $n$ be at least 30 ([All90] p.434).
For example, we simulate algorithm NFD-E with 32 messages for the estimation and
$D$ having an exponential distribution (Section 4.5.2). The simulation results show
that NFD-E provides essentially the same QoS as NFD-S (the new algorithm with
synchronized clocks), so the estimation does not compromise the QoS of the new
failure detector.

With $n$ as a parameter varying from 1 towards $\infty$, NFD-E is actually a spectrum
of algorithms. The larger the value of $n$ is, the better the estimates are. The
algorithm NFD-U corresponds to one end point of the spectrum when $n = \infty$.

The other end point of this spectrum, namely the algorithm with $n = 1$, is
worth some further discussion. When $n = 1$, formula (4.24) becomes $\hat{EA}_{\ell+1} =
A'_1 + (\ell + 1 - s_1)\eta$. According to the algorithm in Fig. 4.4, when $\hat{EA}_{\ell+1}$ is computed
at line 10, the most recent message $q$ received is $m_\ell$. Thus $s_1 = \ell$, $\hat{EA}_{\ell+1} = A'_1 + \eta$
and $\tau_{\ell+1} = A'_1 + \eta + \alpha$. This means that whenever a new heartbeat message with
a higher sequence number is received, $q$ sets a new freshness point $\tau_{\ell+1}$, which is a
fixed $\eta + \alpha$ time units away from the current time, such that if no new heartbeat
message is received by time $\tau_{\ell+1}$, then $q$ starts suspecting $p$. This is just the simple
algorithm!

Therefore, when $n$ varies from 1 towards $\infty$, the algorithm NFD-E spans a spec-
trum that includes the simple algorithm at one end ($n = 1$), and the new algorithm
NFD-U in which $q$ knows the expected arrival times of the heartbeat messages at
the other end ($n = \infty$). When the number of the heartbeat messages used in the
estimation increases, the new algorithm moves away from the simple algorithm and
gets closer to the algorithm NFD-U. This demonstrates that the problem of the sim-
ple algorithm is that it does not use enough information available (it only uses the
most recently received heartbeat message), and by using more information available (using more messages received), the new algorithm is able to provide a better QoS than the simple algorithm.

4.4.3 Configuring the Failure Detector When Local Clocks are Not Synchronized and the Probabilistic Behavior of the Messages is Not Known

We now consider systems in which local clocks are not synchronized and the probabilistic behavior of the messages is not known. Since local clocks are not synchronized, we cannot use algorithm NFD-S. In this section, we show how to compute the parameters $\eta$ and $\alpha$ of algorithm NFD-U to meet the QoS requirements in such systems. For algorithm NFD-E, when the number $n$ of messages used to estimate the expected arrival times are large, the estimates are very accurate, and thus for practical purposes the computation of $\eta$ and $\alpha$ for NFD-U also applies to NFD-E. The method used here is based on the one given in Section 4.4.1.

We first need to point out that in such systems, one cannot estimate $E(D)$ using only one-way heartbeat messages. This is because in such settings one cannot distinguish a system with small message delays but a large clock skew from another system with large message delays but a small clock skew, as we now further explain. Suppose in a system $S$ with message delay time $D$, one obtains an estimate $\mu$ of $E(D)$ using only one way messages from $p$ to $q$. Suppose in system $S$ $q$’s local clock is $s$ time units ahead of $p$’s local clock. Now consider another system $S’$ with message delay time $D’ = D + c$, where $c$ is a constant. That is, each message in $S’$ is delayed
by $c$ time units longer than in $S$. Suppose in $S'$ $q$'s local clock time is $s - c$ time units ahead of $p$'s local clock. Thus, in both systems $S$ and $S'$, a message sent by $p$ at the same $p$'s local clock time is received at the same $q$'s local clock time. Therefore, with unknown clock skews and only one way messages from $p$ to $q$, one cannot distinguish the two systems $S$ and $S'$, and so in $S'$ the estimate of $E(D')$ obtained is also $\mu$. But $\mu$ cannot be an estimate of both $E(D)$ and $E(D')$, since $E(D') = E(D) + c$ for an arbitrary constant $c$. This shows that $E(D)$ cannot be estimated when local clocks are not synchronized and only one way messages from $p$ to $q$ are used.

Fortunately, we do not need to estimate $E(D)$. Since the analysis of NFD-S applies to NFD-U if $\delta$ is replaced with $\alpha + E(D)$, with this replacement we obtain the following theorem from Theorem 4.17. From this theorem, it is clear that we only need $p_L$ and $V(D)$ to bound the QoS metrics of NFD-U.\(^7\)

**Theorem 4.19** Assume $\alpha > 0$. For algorithm NFD-U, in the nondegenerated cases of Theorem 4.11, we have

\[
E(T_{MR}) \geq \frac{\eta}{\beta}, \quad (4.25)
\]

\[
E(T_M) \leq \frac{\eta}{\gamma}, \quad (4.26)
\]

\[
P_A \geq 1 - \beta, \quad (4.27)
\]

\[
E(T_G) \geq \frac{1 - \beta}{\beta} \eta, \quad (4.28)
\]

\[
E(T_{FG}) \geq \frac{1 - \beta}{2\beta} \eta. \quad (4.29)
\]

where

\[
\beta = \prod_{j=0}^{k_0} \frac{V(D) + p_L(\alpha - j\eta)^2}{V(D) + (\alpha - j\eta)^2}, \quad k_0 = \lceil \alpha/\eta \rceil - 1,
\]

\(^7\)This is one reason why it is convenient to set the freshness points with respect to the expected arrival times as opposed to some other reference points (e.g. the median arrival times).
and
\[ \gamma = \frac{(1 - p_L)(\alpha + \eta)^2}{V(D) + (\alpha + \eta)^2}. \]

We first assume that \( p_L \) and \( V(D) \) are known, and show how to compute parameters \( \eta \) and \( \alpha \) of NFD-U using Theorem 4.17 to satisfy QoS requirements. We then show how to estimate \( p_L \) and \( V(D) \).

We consider a set of QoS requirements of the form:
\[ T_D \leq T_D^U + E(D), \quad E(T_{MR}) \geq T_{MR}^L, \quad E(T_M) \leq T_M^U. \]
(4.30)

These requirements are identical to the ones in (4.7), except that the upper bound requirement on the detection time is not just \( T_D^U \), but rather \( T_D^U \) plus the unknown average message delay \( E(D) \). This is justified as follows. First, it is not surprising that the detection time includes \( E(D) \): it is not reasonable to require a failure detector to detect a crash faster than the average delay of a heartbeat. Second, when local clocks are not synchronized and only one-way messages are used, an absolute bound \( T_D \leq T_D^U \) cannot be enforced by any failure detector. The reason is the same as the reason why \( E(D) \) cannot be estimated in such settings: one cannot distinguish a system with small message delays but a large clock skew from another system with large message delays but a small clock skew.

The following is the configuration procedure for algorithm NFD-U, modified from the one in Section 4.4.1.

- **Step 1:** Compute \( \gamma' = (1 - p_L)(T_D^U)^2/(V(D) + (T_D^U)^2) \) and let \( \eta_{\text{max}} = \min(\gamma'T_M^U, T_D^U) \).
• **Step 2:** Let

\[ f(\eta) = \eta \cdot \left[ \frac{T_D^U}{\eta} \right]^{-1} \prod_{j=1}^{V(D)} \frac{V(D) + (T_D^U - j\eta)^2}{V(D) + p_L(T_D^U - j\eta)^2}. \] (4.31)

Find the largest \( \eta \leq \eta_{\text{max}} \) such that \( f(\eta) \geq T_{\text{thr}}^L \).

• **Step 3:** If \( \eta \geq \Delta \), then set \( \alpha = T_D^U - \eta \); otherwise, the procedure does not find appropriate \( \eta \) and \( \alpha \).

**Theorem 4.20** Consider a system with unsynchronized, drift-free clocks, where the probabilistic behavior of messages is not known. With parameters \( \eta \) and \( \alpha \) computed by the above procedure, the failure detector algorithm NFD-U of Fig. 4.3 satisfies the QoS requirements given in (4.30).

**Estimating \( p_L \) and \( V(D) \)**

When local clocks are not synchronized, The message loss probability \( p_L \) and the variance \( V(D) \) of message delay can still be estimated using the heartbeat messages, in exactly the same way as the one given in Section 4.4.1. For \( p_L \), this is because we only use sequence numbers of the heartbeat messages to estimate \( p_L \), and so it is not affected by whether the clocks are synchronized or not. For \( V(D) \), we still use the variance of \( A - S \) of heartbeat messages as the estimate of \( V(D) \), where \( A \) is the time (with respect to \( q \)'s local clock time) when \( q \) receives a message \( m \), and \( S \) is the time (with respect to \( p \)'s local clock time) when \( p \) sends \( m \). This estimate method still works because here \( A - S \) is the actual delay of \( m \) plus a constant, namely the skew between the clocks of \( p \) and \( q \). Thus the variance of \( A - S \) is the same as the variance \( V(D) \) of message delay.
4.5 Simulation Results

We simulate both the new failure detector algorithm that we developed and the simple algorithm commonly used in practice. In particular, (a) we simulate the algorithm NFD-S (the one that sets the freshness points using the sending times of the heartbeat messages and synchronized clocks), and show that the simulation results are consistent with our QoS analysis of NFD-S in Section 4.3.2; (b) we simulate the algorithm NFD-E (the one that sets freshness points with respect to the expected arrival times of the heartbeat messages), show how the QoS of the algorithm changes as the number $n$ of messages used for estimating the expected arrival times increases, and show that, with appropriately chosen $n$, NFD-E provides essentially the same QoS as NFD-S; and (c) we simulate the simple algorithm and compare it to the different versions of the new algorithms, and show that when all algorithms send messages at the same rate and satisfy the same upper bound on the worst-case detection time, the new algorithms provide much better accuracy than the simple algorithm.

The settings of the simulations are as follows. For the purpose of comparison, we fix the intersending time $\eta$ of heartbeat messages in both the new algorithm and the simple algorithm to be 1. The message loss probability $p_L$ is set to 0.01. The message delay time $D$ follows the exponential distribution (i.e., $Pr(D \leq x) = 1 - e^{-x/E(D)}$ for all $x \geq 0$). We choose the exponential distribution because of the following two reasons: first, it has the characteristic that a large portion of messages have fairly short delays while a small portion of messages have large delays, which is also the characteristic of message delays in many practical systems [Cri89]; second, it has
simple analytical representation which allows us to compare the simulation results with the analytical results given in Theorem 4.11. The average message delay time $E(D)$ is set to be 0.02, which is a small value compared to the intersending time $\eta$. This corresponds to the practical situation in which message delays are in the order of tens of milliseconds (typical for messages transmitted over the Internet), while heartbeat messages are sent every few seconds. Note that since $D$ follows an exponential distribution, we have that the standard deviation $\sigma(D) = E(D) = 0.02$, and the variance $V(D) = \sigma(D)^2 = 4 \times 10^{-4}$.

We compare the accuracy of different algorithms when they all satisfy the same bound $T_D^U$ on the worst-case detection time. To do so, we run simulations for each algorithm as follows: (a) We first configure the algorithm using the given bound $T_D^U$. (b) We then run simulations to verify that the configuration is indeed correct, i.e., the given bound $T_D^U$ is satisfied. More specifically, we simulate the algorithm in 10,000 runs in which process $p$ crashes at some nondeterministic times, and obtain the maximum detection time observed among all these runs, and see if this observed maximum detection time is close to but not exceeding the given bound $T_D^U$. (c) Finally we obtain the average mistake recurrence time by simulating the algorithm in runs in which $p$ does not crash, and then taking the average of the lengths of 500 mistake recurrence intervals. We found that the average mistake recurrence time is representative for the purpose of comparing the accuracy of the algorithms we simulate, and thus we omit the simulation results on other accuracy metrics here. We vary the bound $T_D^U$ from 1 to 3.5, i.e., from exactly one intersending time of heartbeat messages to three and a half times of the intersending time, and show how the simulation results vary accordingly.
4.5.1 Simulation Results of NFD-S

It is easy to configure the parameters of NFD-S to meet the given upper bound $T^U_D$ on the worst-case detection time. In fact, since the intersending time $\eta$ is fixed (to 1), only parameter $\delta$ is configurable, and by Theorem 4.11 (1), we set $\delta = T^U_D - \eta = T^U_D - 1$.

Figure 4.5 shows the simulation results that checks the correctness of our configurations of NFD-S. The reference line represents the situation in which a failure
detector is perfectly configured: the maximum detection time is equal to the desired bound $T_{D}^{U}$. Figure 4.5 shows that all the maximum detection times observed in the simulations of NFD-S are very close to the reference line. Therefore NFD-S is correctly configured.

Figure 4.6 shows the simulation results on the average mistake recurrence times of algorithm NFD-S, together with the plot of the analytical formula for $E(T_{MR})$ that we derived in Section 4.3.2 (formula (4.2) of Theorem 4.11). The immediate
conclusions from Fig. 4.6 is: the simulation results of algorithm NFD-S matches the analytical formula for $E(T_{MB})$, i.e., formula (4.2) of Theorem 4.11.

Furthermore, note that the y-axis is in log scale, which means that when $T^U_D$ increases linearly, the overall tendency of the average mistake recurrence time is to increase exponentially fast. This increase, however, is not continuous: as $T^U_D$ increases, the average mistake recurrence time alternates between rapid increasing periods and flat (nonincreasing) periods — just as a step function. We now explain why the curve resembles the curve of a step function.

We separate the curve into the following periods according to the value of $T^U_D$, and explain them one by one.

1. When $T^U_D = 1$, the parameter $\delta$ is set to $T^U_D - \eta = 0$. Recall that the freshness point $\tau_i$ is set to be $\sigma_i + \delta$ where $\sigma_i$ is the sending time of $m_i$. So, in this case the freshness point $\tau_i$ is the same as the sending time $\sigma_i$. But it is impossible for message $m_i$ to arrive before time $\tau_i$, so $q$ suspects $p$ at every freshness point $\tau_i$. During the interval $(\tau_i, \tau_{i+1})$, $q$ is likely to receive the message $m_i$ (recall that the average message delay is only 0.02 and the message loss probability is only 0.01), and thus becomes trusting $p$ again. Therefore, when $T^U_D = 1$, the average mistake recurrence time is close to 1.

2. As $T^U_D$ increases from 1 to around 1.16, $\delta = T^U_D - \eta$ increases from 0 to 0.16 and the freshness points $\tau_i$’s are shifted forward in time accordingly. In this period, the probability that message $m_i$ arrives after time $\tau_i$ decreases very fast, from 1 to $e^{-8} = 0.0003$. Thus a small increase in $\delta$ reduces the probability that $m_i$ arrives late significantly, and therefore increase significantly the time between
consecutive mistakes.

3. When $T^U_D = 1.16$, $\delta = T^U_D - \eta = 0.16$, i.e., $\tau_i$’s are shifted forward in time $0.16$ time units from $\sigma_i$’s. This shift distance is $8$ times of the average message delay time, and thus if $m_i$ is not lost, there is a very high probability that $m_i$ is indeed received before $\tau_i$ (in fact, the probability is $1 - e^{-8} = 0.9997$). Since the message loss probability is $0.01$, we know that at this point the main cause of a failure detector mistake is that a message is lost. Since on average one out of every $100$ messages is lost, the average mistake recurrence time is close to $100$, as shown in Fig. 4.6.

4. From $T^U_D = 1.16$ to $T^U_D = 2.0$, $\delta$ increases from $0.16$ to $1$. In this period, a message is very unlikely to be delayed by more than $\delta$ time units, while a single message loss is enough to cause a failure detector mistake. Therefore, an increase in $\delta$ does not help much to gain a better mistake recurrence time, and the curve is almost flat.

The case is similar when $T^U_D$ increases from $2$ to $3$: (a) From $2$ to around $2.16$, a failure detector mistake is mainly caused by the loss of a message $m_i$ followed by the delay of message $m_{i+1}$. Thus an increase in $\delta$ increases the probability that message $m_{i+1}$ is received before time $\tau_i$, so that the failure detector does not make a mistake even if $m_i$ is lost. Therefore in this period the average mistake recurrence time increases sharply. (b) From $2.16$ to $3$, a failure detector mistake is mainly due to the loss of two consecutive messages, and thus an increase in $\delta$ does not help much to gain a better accuracy, and the average mistake recurrence time stays at about $10^4$. Other periods can be explained similarly.
As a summary, the flat portions of the curve correspond to the failure detector configurations in which the failure detector mistakes are mainly due to consecutive message losses, while the ascending portions of the curve correspond to the configurations in which the failure detector mistakes are mainly due to a sequence of consecutive message losses followed by the delay of the last message before the suspicion.

In Fig. 4.6, we only show the average mistake recurrence times obtained from the simulations. To further show that these simulation results are reliable, i.e., they are not just by chance very close to the theoretical analysis, we show their corresponding confidence intervals in Fig. 4.7. In this figure, we show the 99% confidence intervals for the expected values of mistake recurrence times of NFD-S, together with the plot of the analytical formula for $E(T_{mr})$ of NFD-S. The confidence intervals are computed using standard techniques (see e.g. [All90] p.445). The figure illustrates that all the confidence intervals are very small and surrounding the theoretical results. This shows that the simulation results are accurate and are not obtained by chance. The confidence intervals of the simulation results of other algorithms show the similar properties, and thus we do not include them in the thesis.

### 4.5.2 Simulation Results of NFD-E

Algorithm NFD-E has a parameter $n$ — the number of messages used for estimating the expected arrival times of the heartbeats — that also affects the QoS. To show this, we first run simulations in which the parameter $\alpha$ is fixed and $n$ takes the values 1, 4, 8, 12, 16, 20, 24, 28, 32 respectively, and see how the maximum detection times and the average mistake recurrence times change accordingly.
Figure 4.7: The 99% confidence intervals for the expected values of mistake recurrence times of NFD-S (shown by ⊤ ⊥), with the plot of the analytical formula for $E(T_{MR})$ of NFD-S (shown by —).
Figure 4.8 shows the simulation results for $\alpha = 1.90$. From the figure, we see that when $n$ increases, the average mistake recurrence times have no obvious change, while the maximum detection time observed decreases from above 3.00 (when $n = 1$) to about 2.93 (when $n = 32$). Note that according to the analytical results on algorithm NFD-U (the one that knows all the expected arrival times), we have $T_D \leq E(D) + \alpha + \eta$. Thus with $\alpha = 1.90$, we have $T_D \leq 2.92$ for algorithm NFD-U. So from $n = 1$ to $n = 32$, the maximum detection time observed changes from more than 0.08 (4 times of $E(D)$) above the bound 2.92, to within 0.01 (half of $E(D)$) above the bound 2.92. This suggests that when $n = 32$, the algorithm NFD-E is very close to the algorithm NFD-U. Simulations on other $\alpha$ values show the similar results, and so we choose $n = 32$ for the algorithm NFD-E.

Since when $n = 32$ NFD-E is very close to NFD-U, we use the bound $T_D \leq E(D) + \alpha + \eta$ of the algorithm NFD-U to compute the parameter $\alpha$ for the algorithm NFD-E. In particular, we set $\alpha = T^U_D - E(D) - \eta = T^U_D - 1.02$.

Figure 4.9 shows the simulation results that checks the correctness of our configurations of NFD-E. Since all simulation results are very close to the reference line, the algorithm NFD-E is correctly configured.

Figure 4.10 shows the simulation results on the average mistake recurrence times of algorithms NFD-E, together with the plot of the analytical formula for $E(T_{MR})$ that we derived for algorithm NFD-S in Section 4.3.2 (formula (4.2) of Theorem 4.11). From this figure, we see that with appropriately chosen $n$, the accuracy of algorithms NFD-S and NFD-E are essentially the same, when both algorithms send heartbeat messages at the same rate and satisfy the same upper bound on the worst-case detection time.
(a) The maximum detection time observed decreases when $n$ increases.

(b) The average mistake recurrence time stays about the same when $n$ increases.

Figure 4.8: The change of the QoS of NFD-E when $n$ increases. Parameter $\alpha = 1.90$. 
Figure 4.9: The maximum detection times observed in the simulations of NFD-E (shown by $\times$)
Figure 4.10: The average mistake recurrence times obtained from the simulations of NFD-E (shown by ×), with the plot of the analytical formula for $E(T_{MR})$ of NFD-S (shown by —).
4.5.3 Simulation Results of the Simple Algorithm

As discussed in the Introduction of this chapter, the worst-case detection time of the simple algorithm is the maximum message delay time plus the timeout value $TO$. This means that for many practical systems that have no upper bound on the message delay time, as well as for our simulation setting, the worst-case detection time of the simple algorithm is unbounded. Thus in these situations the simple algorithm as it stands is not suitable to satisfy QoS requirements that require an upper bound on the worst-case detection time.

In this section, we apply a straightforward modification to the simple algorithm so that it is able to provide an upper bound on the worst-case detection time. Since the unbounded worst-case detection time of the simple algorithm is caused by the messages with very large delays, we modify the algorithm such that these messages are discarded. More precisely, the modified algorithm has another parameter, the **cutoff time** $c$, such that all messages delayed by more than $c$ time units are discarded. We call messages delayed by at most $c$ time units *fast* messages, and messages delayed by more than $c$ time units *slow* messages. We assume that the simple algorithm is able to distinguish slow messages from fast messages.\(^8\)

With this modification, it is easy to see that the simple algorithm now has a worst-case detection time $c + TO$. Given a bound $T_D^U$ on the worst-case detection time, there is a tradeoff in setting the cutoff time $c$ and the timeout value $TO$: the larger the cutoff time $c$, the smaller the number of messages being discarded, but the shorter the timeout value $TO$, and vice versa. In the simulations, we choose two

---

\(^8\)This is not easy when local clocks are not synchronized. A fail-aware datagram service [FC97] may be used for this purpose.
Figure 4.11: The maximum detection times observed in the simulations of SFD-L and SFD-S (shown by ⋄ and ◦).

cutoff times $c = 0.16$ and $c = 0.08$, i.e., cutoff times of 8 and 4 times of the mean message delay time respectively. The timeout value $TO$ is then set to be $T_U - c$. The algorithm with $c = 0.16$ is denoted by SFD-L, and the one with $c = 0.08$ is denoted by SFD-S.

Figure 4.11 shows the simulation results of the observed maximum detection times of SFD-L and SFD-S. Since all simulation results are very close to the reference line at which the maximum detection time observed equals $T_U$, algorithms SFD-L and SFD-S are correctly configured.
Figure 4.12: The average mistake recurrence times obtained from the simulations of SFD-L and SFD-S (shown by -○- and -◦-), with the plot of the analytical formula for $E(T_{MR})$ of NFD-S (shown by —).
Figure 4.12 shows the simulation results on the average mistake recurrence times of SFD-L and SFD-S, together with the plot of the analytical formula for \( E(T_{MR}) \) of the new algorithm NFD-S (formula (4.2) of Theorem 4.11), which is the same plot as given in Fig. 4.6 and 4.10. From Fig. 4.6 and 4.10 we know that this plot also closely represents the simulation results of the two versions of the new algorithm NFD-S and NFD-E.

We have the following observations from Fig. 4.12.

1. The curves of SFD-L and SFD-S resemble the curves of some step functions. The reason is similar to the one that we give for algorithm NFD-S.

2. The flat portions of SFD-L are very close to those of NFD-S, but the flat portions of SFD-S are much lower than those of the other two curves, and the gap is orders of magnitude large and it is increasing.

   The reason is as follows. We already explained that the flat portions of a curve correspond to the cases in which failure detector mistakes are mainly due to message losses. More precisely, the first flat portion of each curve corresponds to the cases in which a mistake is mainly due to a single message loss; the second flat portion corresponds to the cases in which a mistake is mainly due to two consecutive message losses, and so on.

   For the modified simple algorithm, slow messages are equivalent to lost messages since they are discarded by the algorithm. In SFD-L with cutoff time \( c = 0.16 \), the probability that a message is slow is very small comparing to the probability that a message is really lost (in fact, it is \( e^{c/E(D)} = 3.4 \times 10^{-4} \) compared to the message loss probability 0.01). In SFD-S, however, the cutoff
time is $c = 0.08$ and the probability that a message is slow is 0.018, which is significant. For this algorithm, the combined message loss probability is $p_L + 0.018 = 0.028$. Under this message loss probability, a single message loss occurs about every 35 messages, and the event of two consecutive message losses occurs about every 1200 messages. This explains why the vertical position of the first flat portion of SFD-S is between 30 and 40 and the vertical position of the second flat portion is between 1000 and 2000. Since the difference in the probability of consecutive message losses between the algorithm SFD-S and the other two algorithms is increasing, the gap between SFD-S and the other two algorithms is increasing accordingly.

3. Regarding the ascending portions of the curves, as $T_D^U$ increases, an ascending portion of NFD-S always starts first, then followed by an ascending portion of SFD-S, and finally followed by an ascending portion of SFD-L. In these ascending portions, under the same value $T_D^U$, the average mistake recurrence time of the new algorithm could be orders of magnitude better than those of the simple algorithms.

This can be explained by the following example. Consider the point when $T_D^U = 1.08$. For the new algorithm NFD-S, $\delta = T_D^U - \eta = 0.08$, which means that the freshness points are shifted forward in time by 4 times the mean message delay. Under this setting, a message (if not lost) is very likely to be received before the corresponding freshness point and thus avoid a failure detector mistake (the exact probability is 0.982). For the simple algorithm SFD-S with $c = 0.08$, we have $TO = T_D^U - c = 1$. This means that after a
message is received, the timer will expire one time unit later. Since on average
the next message will arrive one time unit later than the receipt of the previous
message, \( TO = 1 \) means that about half of the messages will arrive after the
timer expires and thus cause failure detector mistakes. Thus the accuracy of
SFD-S is not good compared with NFD-S. For the simple algorithm SFD-L
with \( c = 0.16 \), we have \( TO = T_U^D - c = 0.92 \). Under this setting, the timeout
is too short, such that almost no message can arrive before the timer expires.
Thus the accuracy of SFD-L is even worse at this point. Similar explanations
can be applied to other points of \( T_U^D \) in the period from 1 to around 1.2, from
2 to around 2.2, etc.

Therefore, in general, under the same requirement of \( T_U^D \), the configuration of
the new algorithm always gives a lower probability of a failure detector mistake
caused either by message delay or by message loss, than the configuration of the
simple algorithm. For the simple algorithm, the larger the cutoff time is, the smaller
the timeout value, and thus the higher the probability of a failure detector mistake
caused by message delay. On the other hand, if the cutoff time is getting smaller,
more messages are discarded (it effectively increases the probability that a message
is lost), and this increases the probability of a failure detector mistake caused by
message losses.

From the above observations, it is not hard to see that when the cutoff time
of the modified simple algorithm increases, its curve is shifted further to the right;
when the cutoff time decreases, its curve is pressed further down towards the x-axis.
In all cases, the curve of the simple algorithm is always under the curve of the new
algorithm.
To summarize, the simulation results show that, when both algorithms send heart-beat messages at the same rate and satisfy the same upper bound on the worst-case detection time, the accuracy of the new algorithm (with or without synchronized clocks) always dominates the accuracy of the simple algorithm, and in some cases it is orders of magnitude better.

4.6 Concluding Remarks

On the Adaptiveness of the New Failure Detector

In this chapter, our network model assumes that the probabilistic behavior of the network does not change over time. In practice, the network behavior may change over time gradually. For example, during working days, a corporate network typically experiences heavier traffic, which means longer message delays and more message losses, while during nights and weekends, the network traffic is usually much lighter. However, for a short period of time, e.g., one hour, the change of network behavior is relatively small, and our model is a good approximation for such relatively short periods.

For the gradual changes of the network behavior in a long time period, our new failure detector algorithm has the ability to adapt to the changes and behave accordingly. This is because we can configure the failure detector so that it only uses recent heartbeat messages to estimate the relevant system parameters such as $p_L$, $E(D)$ and $V(D)$, and the expected arrival times of the heartbeats if necessary. Therefore, the algorithm can automatically adapt to the recent behavior of the network, and thus the QoS of the failure detector can be guaranteed even if the network behavior
changes gradually over time.

**On the QoS Requirements**

In Sections 4.3.4, 4.4.1 and 4.4.3, we consider some simple QoS requirements that take the form of the bounds on some QoS metrics. Applications may also have requirements in other forms. For example, an application may specify some objective function in terms of the QoS metrics, and require that the failure detector be configured such that the objective function is maximized. To deal with such more general QoS requirements, a decision-theoretic approach may be used in the configuration of the failure detector. Decision theory [Res87] provides mathematical tools for making decisions, and there have been some research works that apply decision theory in certain areas of computer science such as networking, distributed computing, and database systems (e.g., [MHW96, BBS98, BS98, CH98, CHS99]). A study on the QoS of failure detectors using decision theory is an interesting research direction, but it is beyond the scope of this thesis.

**On n-Process Systems**

In this thesis, we focus on two-process systems: a failure detector at a process $q$ monitors a process $p$. Many practical systems consist of more than two processes, and failure detection is required between every pair of processes. Our work on two-process systems can be used as a basis for the study of n-process systems. For example, in an n-process system, one may be interested in the time elapsed from the time when a process $p$ crashes to the time when all other processes detect the crash of
For this purpose, we can use our QoS metric — the detection time — of the failure detector on every process \( q \) that monitors \( p \), and then take the maximum of all these detection times to obtain the value we want. Of course, \( n \)-process systems present more complicated cases than two process systems, and more careful and creative study is necessary. We hope that this thesis can provide some helpful directions to the study of failure detection in more complicated distributed systems.
Appendix A

Theory of Marked Point Processes

Most of the notations, terminologies, and results concerning the theory of marked point processes are taken from [Sig95].

Marked Point Processes

Let \( \mathbb{R}_+ \) and \( \mathbb{Z}_+ \) denote the sets of nonnegative real numbers and integers, respectively. Let \( K \) denote a complete separable metric space called mark space.

A simple marked point process (mpp) on the nonnegative time line \( \mathbb{R}_+ \) is a sequence

\[
\psi = \{(t_n, k_n) : n \in \mathbb{Z}_+, t_n \in \mathbb{R}_+, k_n \in K\},
\]

such that \( 0 \leq t_0 < t_1 < t_2 < \cdots, \lim_{n \to +\infty} t_n = +\infty \). We call \( t_n \) an event time and \( k_n \) a mark associate with event time \( t_n \). By simple we mean that the event times are all different. Let \( M = M_K \) denote the collection of all simple mpp’s with mark space \( K \). The set of all Borel measurable subsets of \( M \) is denoted as \( \mathcal{B}(M) \).
We sometimes use the following *interevent time sequence representation* that is equivalent to (A.1):

\[
\{t_0, \{(T_n, k_n) : n \in \mathbb{Z}_+, T_n \equiv t_{n+1} - t_n\}\}, \tag{A.2}
\]

where \(T_n\) denotes the \(n\)-th interevent time.

Note that \(t_n, T_n, \text{ and } k_n\) are actually measurable mappings from \(M\) to \(\mathbb{R}_+\) or \(\mathbb{K}\).

**Shift Mappings**

A *shift mapping* \(\theta_s : M \to M\) is a mapping that shifts a mpp \(\psi\) to the left by \(s\) time units. More precisely, if \(s = 0\), then \(\theta_s\) is the identity mapping; if \(s > 0\), then for \(\psi = \{(t_n, k_n)\}\), suppose \(t_{i-1} < s \leq t_i\) for some \(i \in \mathbb{Z}_+\) (denote \(t_{-1} = 0\) for convenience). We then have

\[
\theta_s \psi \ \overset{\text{def}}{=} \ \{(t_{i+n} - s, k_{i+n}) : n \geq 0\}. \tag{A.3}
\]

That is, \(\theta_s \psi\) is the mpp obtained from \(\psi\) by shifting the origin to \(s\), re-labeling event times at and after \(s\) as \(t_0, t_1, \ldots\), and ignoring the events before time \(s\). Let \(\theta_{(j)} \overset{\text{def}}{=} \theta_{t_j}\) denote shifting by the event time \(t_j, j \geq 0\). We then let

\[
\psi_s \overset{\text{def}}{=} \theta_s \psi \quad \text{and} \quad \psi_{(j)} \overset{\text{def}}{=} \theta_{(j)} \psi. \tag{A.4}
\]

Let \(\theta_t^{-1} \mathcal{E} \overset{\text{def}}{=} \{\psi \in M : \theta_t \psi \in \mathcal{E}\}\), and \(\theta_{(j)}^{-1} \mathcal{E} \overset{\text{def}}{=} \{\psi \in M : \theta_{(j)} \psi \in \mathcal{E}\}\).

**Random Marked Point Processes**

Let \((\Omega, \mathcal{F}, P)\) be a probability space. A *random marked point process (rmpp)\( \Psi\) is a measurable mapping \(\Psi : \Omega \to M\). \(\Psi\) has the distribution \(P(\Psi \in \mathcal{E}) \overset{\text{def}}{=} P\{\omega \in \Omega : \theta_t^{-1} \mathcal{E}\).
ψ(ω) ∈ E\} defined for all E ∈ B(M). ψs is the rmpp obtained from ψ be shifting the origin to time s, that is, for all ω ∈ Ω, ψs(ω) = ψ(ω)s. Similarly, ψ(j) is the rmpp obtained from ψ by shifting the origin to the time of the j-th event, that is, for all ω ∈ Ω, ψ(j)(ω) = ψ(ω)(j).

Stationary Versions

A rmpp ψ is event stationary if P(ψ(j) ∈ E) = P(ψ ∈ E) for all j ∈ Z+ and all E ∈ B(M). ψ is time stationary if P(ψs ∈ E) = P(ψ ∈ E) for all s ∈ R+ and all E ∈ B(M).

The event stationary version ψ0 and the time stationary version ψ∗ of rmpp ψ are two rmpp’s defined by the following distributions (assuming they exist):

\[ Pr(ψ^0 ∈ E) \overset{\text{def}}{=} \lim_{n → ∞} \frac{1}{n} \sum_{j=0}^{n-1} P(ψ(j) ∈ E), \text{ for all } E ∈ B(M), \] (A.5)

and

\[ Pr(ψ^* ∈ E) \overset{\text{def}}{=} \lim_{t → ∞} \frac{1}{t} ∫_0^t P(ψ_s ∈ E) ds, \text{ for all } E ∈ B(M). \] (A.6)

As shown in [Sig95], ψ0 is event stationary and ψ∗ is time stationary.

**Proposition A.1** Any event stationary ψ has, with probability one, the event time t0 at the origin, i.e. Pr(t0 ◦ ψ = 0) = 1. Any time stationary ψ has, with probability one, no event at the origin, i.e. Pr(t0 ◦ ψ = 0) = 0.

Invariant σ-Field

The invariant σ-field of M with respect to the flow of shifts, \{θ_t : t ≥ 0\}, is denoted by I and defined by I \overset{\text{def}}{=} \{E ∈ B(M) : θ_t^{-1}E = E, t ≥ 0\}. The invariant σ-field of
with respect to the event shifts, \( \{\theta(j) : j \geq 0\} \), is denoted by \( \mathcal{I}_e \) and defined by
\[
\mathcal{I}_e \overset{\text{def}}{=} \{ E \in B(M) : \theta^{-1}(j) E = E, j \geq 0 \}.
\]

**Proposition A.2** \( \mathcal{I}_e = \mathcal{I} \).

Because of the above proposition, we use \( \mathcal{I} \) to denote the one and only invariant \( \sigma \)-field of \( M \).

If \( \Psi \) is defined on a probability space \((\Omega, \mathcal{F}, P)\), then the invariant \( \sigma \)-field on \( M \) can be lifted onto \( \mathcal{F} \) by taking the inverse image:
\[
I_\Psi \overset{\text{def}}{=} \Psi^{-1} \mathcal{I} = \{ \Psi^{-1} E : E \in \mathcal{I} \},
\]
where \( \Psi^{-1} E \overset{\text{def}}{=} \{ \omega \in \Omega : \Psi(\omega) \in E \} \). We omit the superscript \( \Psi \) whenever the context is clear. For example, for the notation of conditional expected value, we use \( E_{\mathcal{I}}(f \circ \Psi) \) instead of \( E_{\mathcal{I}_\Psi}(f \circ \Psi) \) (\( f \) is a measurable mapping from \( M \) to \( \mathbb{R}_+ \)).

**Ergodicity**

An event stationary \( \Psi^0 \) is called **ergodic** if for any two events \( E_1, E_2 \in B(M) \),
\[
\lim_{n \to \infty} \frac{1}{n} \sum_{j=0}^{n-1} P_r(\Psi^0 \in E_1, \Psi^0(j) \in E_2) = P_r(\Psi^0 \in E_1)P_r(\Psi^0 \in E_2). \tag{A.7}
\]

Similarly, a time stationary \( \Psi^* \) is called **ergodic** if for any two events \( E_1, E_2 \in B(M) \),
\[
\lim_{t \to \infty} \frac{1}{t} \int_0^t P_r(\Psi^* \in E_1, \Psi^*(s) \in E_2) \, ds = P_r(\Psi^* \in E_1)P_r(\Psi^* \in E_2). \tag{A.8}
\]

As suggested by Sigman [Sig95], the ergodicity should be regarded as a condition describing a kind of **loss of memory** as the event (or time) parameter tends to \( \infty \).

“For \( \Psi^0 \) this means that if you start with \( \Psi^0 \) and then randomly observe it way out at an event, then what you observe is an independent copy of \( \Psi^0 \) itself” ([Sig95] p.38). The same holds for \( \Psi^* \) when you randomly observe it way out in time. The follow
proposition shows that ergodicity can be equivalently defined by using invariant σ-field \( \mathcal{I} \).

**Proposition A.3** \( \Psi^0 \) is ergodic if and only if the invariant σ-field \( \mathcal{I} \) is 0-1 with respect to \( \Psi^0 \), i.e., iff for all \( E \in \mathcal{I} \), \( \Pr(\Psi^0 \in E) \in \{0, 1\} \). \( \Psi^* \) is ergodic if and only if the invariant σ-field \( \mathcal{I} \) is 0-1 with respect to \( \Psi^* \), i.e., iff for all \( E \in \mathcal{I} \), \( \Pr(\Psi^* \in E) \in \{0, 1\} \).

**Proposition A.4** For any measurable \( f : M \to \mathbb{R}^\infty_+ \), if \( \Psi^0 \) is ergodic, then \( E_{\mathcal{I}}(f \circ \Psi^0) = E(f \circ \Psi^0) \) a.s., and if \( \Psi^* \) is ergodic, then \( E_{\mathcal{I}}(f \circ \Psi^*) = E(f \circ \Psi^*) \) a.s.\(^1\)

The following is the version of the important Birkhoff’s Ergodic Theorem for random marked point processes. Henceforth, we assume that \( \Psi, \Psi^0 \) and \( \Psi^* \) use the same underlying probability space \((\Omega, \mathcal{F}, P)\) (one can always construct some common space supporting all of them).

**Theorem A.5**

1. If \( \Psi \) has the event stationary version \( \Psi^0 \), then for any measurable mapping \( f : M \to \mathbb{R}^\infty_+ \),

   \[
   \lim_{n \to \infty} \frac{1}{n} \sum_{j=0}^{n-1} f \circ \Psi(j) = E_{\mathcal{I}}(f \circ \Psi^0) \text{ a.s.} \quad (A.9)
   \]

   In particular, if \( \Psi^0 \) is ergodic, then

   \[
   \lim_{n \to \infty} \frac{1}{n} \sum_{j=0}^{n-1} f \circ \Psi(j) = E(f \circ \Psi^0) \text{ a.s.} \quad (A.10)
   \]

\(^1\)The notation a.s. stands for “almost surely”, which means that the equation is true with probability one.
(2) If $\Psi$ has the time stationary version $\Psi^*$, then for any measurable mapping $f : M \to R_+$ such that $\int_0^t f \circ \Psi_s ds < \infty, t \geq 0, a.s.$,

$$\lim_{t \to \infty} \frac{1}{t} \int_0^t f \circ \Psi_s ds = E(f \circ \Psi^*) \text{ a.s.} \quad (A.11)$$

In particular, if $\Psi^*$ is ergodic, then

$$\lim_{t \to \infty} \frac{1}{t} \int_0^t f \circ \Psi_s ds = E(f \circ \Psi^*) \text{ a.s.} \quad (A.12)$$

Note that $f \circ \Psi^0$ and $f \circ \Psi^*$ are measurable mappings from the underlying sample space $\Omega$ to $R_+$, and so they are random variables. So are $\lim_{n \to \infty} \frac{1}{n} \sum_{j=0}^{n-1} f \circ \Psi(j)$ and $\lim_{t \to \infty} \frac{1}{t} \int_0^t f \circ \Psi_s ds$. Similar mathematical expressions are used in the following theorems.

From the above theorem, we can have the following characterization of the ergodicities of $\Psi^0$ and $\Psi^*$. Let $I_E$ be the indicator function for some event $E \in B(M)$, i.e. for all $\psi \in M$, $I_E(\psi) = 1$ if $\psi \in E$, and $I_E(\psi) = 0$ if $\psi \notin E$.

**Proposition A.6** Suppose that $\Psi$ has event stationary version $\Psi^0$ and time stationary version $\Psi^*$. $\Psi^0$ is ergodic if and only if for all $E \in B(M)$,

$$Pr(\Psi^0 \in E) = \lim_{n \to \infty} \frac{1}{n} \sum_{j=0}^{n-1} I_E \circ \Psi(j) \text{ a.s.} \quad (A.13)$$

$\Psi^*$ is ergodic if and only if for all $E \in B(M)$,

$$Pr(\Psi^* \in E) = \lim_{t \to \infty} \frac{1}{t} \int_0^t I_E \circ \Psi_s ds \text{ a.s.} \quad (A.14)$$

**Proof.** Suppose $\Psi^0$ is ergodic. Then (A.13) is obtained by substituting $f$ in (A.10) with $I_E$. Now suppose (A.13) holds. Then for any $E \in I$, we claim that $I_E \circ \Psi(j) = I_E \circ \Psi$. In fact, for all $\omega \in \Omega$ where $\Omega$ is the underlying sample space for $\Psi$, $I_E \circ \Psi(j)(\omega) = 1$
iff $\Psi(\omega)(j) \in \mathcal{E}$ iff $\Psi(\omega) \in \theta^{-1}_j \mathcal{E}$ iff $\Psi(\omega) \in \mathcal{E}$ iff $I_{\mathcal{E}} \circ \Psi(\omega) = 1$. Thus from (A.13) we have $Pr(\Psi^0 \in \mathcal{E}) = I_{\mathcal{E}} \circ \Psi$ a.s., which implies that $Pr(\Psi^0 \in \mathcal{E}) \in \{0, 1\}$. By Proposition A.3, we know that $\Psi^0$ is ergodic. The proof for $\Psi^*$ is similar. 

Proposition A.6 suggests that the event stationary version $\Psi^0$ of some rmpp $\Psi$ is ergodic if and only if the distribution of $\Psi^0$, i.e. $Pr(\Psi^0 \in \mathcal{E})$, can be obtained (with probability one) from any single run $\Psi(\omega)$ of $\Psi$ as follows: observe $\Psi(\omega)$ at every event time $t_j$ (to obtain $\Psi(\omega)(j)$), check whether the event $\mathcal{E}$ is true when observed at $t_j$ (i.e., whether $I_{\mathcal{E}}(\Psi(\omega)(j)) = 1$ or not), and then use the ratio of the number of event times $t_j$'s at which $\mathcal{E}$ is true over the total number of event times as $Pr(\Psi^0 \in \mathcal{E})$. The distribution of $\Psi^*$ can be obtained in a similar way.

The following lemma shows that the ergodicities of $\Psi^0$ and $\Psi^*$ are equivalent.

**Lemma A.7** Suppose that $\Psi$ has event stationary version $\Psi^0$ and time stationary version $\Psi^*$. Then $\Psi^0$ is ergodic if and only if $\Psi^*$ is ergodic.

### Arrival Rates

Let $N_t : \mathcal{M} \rightarrow \mathbb{R}_+$ be the measurable mapping such that for all $\psi \in \mathcal{M}$, $N_t(\psi)$ is the number of event times of $\psi$ in the period $(0, t]$. Suppose rmpp $\Psi$ has the event stationary version $\Psi^0$ and the time stationary version $\Psi^*$. Let $\lambda \stackrel{\text{def}}{=} E(N_1 \circ \Psi^*)$, and $\lambda$ is called *arrival rate* or *intensity* of $\Psi$. Intuitively, $\lambda$ is the average number of event times or arrivals in a unit period in the time stationary version $\Psi^*$. Let $\lambda_\mathcal{E} \stackrel{\text{def}}{=} E_{\mathcal{E}}(N_1 \circ \Psi^*)$, and $\lambda_\mathcal{E}$ is called *conditional arrival rate* or *conditional intensity* of $\Psi$.

Recall that $T_n : \mathcal{M} \rightarrow \mathbb{R}_+$ is the measurable mapping such that $T_n(\psi) \stackrel{\text{def}}{=} t_{n+1}(\psi) -$
Lemma A.8 For the conditional arrival rate $\lambda_\Psi$, we have
\[
\lambda_\Psi = \lim_{t \to \infty} \frac{N_t \circ \Psi}{t} = \left( \lim_{n \to \infty} \frac{1}{n} \sum_{j=0}^{n-1} T_j \circ \Psi \right)^{-1} = \left\{ E_\Psi(T_0 \circ \Psi^0) \right\}^{-1} \quad a.s. \quad (A.15)
\]

For the arrival rate $\lambda$, we have
\[
\lambda = E(\lambda_\Psi). \quad \quad (A.16)
\]

Moreover, if $\Psi^0$ is ergodic (and so is $\Psi^*$), the we have
\[
\lambda_\Psi = \lambda = \left\{ E(T_0 \circ \Psi^0) \right\}^{-1} \quad a.s. \quad (A.17)
\]

Equalities (A.15) mean that the conditional arrival rate $\lambda_\Psi$ is a random variable, and it can be obtained either from the long run number of events per unit time ($\lim_{t \to \infty} N_t \circ \Psi / t$), or from the reciprocal of the long run average interevent time ($\left\{ \lim_{n \to \infty} \frac{1}{n} \sum_{j=0}^{n-1} T_j \circ \Psi \right\}^{-1}$). Equality (A.16) shows that the arrival rate is the expected value of the random variable $\lambda_\Psi$. Equalities (A.17) mean that, if the stationary versions of $\Psi$ are ergodic, then the conditional arrival rate $\lambda_\Psi$ is almost surely the constant $\lambda$, which is also the reciprocal of the expected value of the very first interevent time of $\Psi^0$.

**Empirical Inversion Formulas**

The empirical inversion formulas are used to connect the event stationary version $\Psi^0$ with the time stationary version $\Psi^*$. Roughly speaking, the results show that (a) a random marked point process $\Psi$ has the event stationary version $\Psi^0$ if and only if it has the time stationary version $\Psi^*$; (b) $\Psi^0$ is the event stationary version of
both $\Psi$ and $\Psi^*$, and $\Psi^*$ is the time stationary version of both $\Psi$ and $\Psi^0$; (c) the distributions of $\Psi^0$ and $\Psi^*$ are related by some inversion formulas. We now state these results formally.

**Theorem A.9** $\Psi$ has the event stationary version $\Psi^0$ and $0 < E_T(T_0 \circ \Psi^0) < \infty$ a.s., if and only if $\Psi$ has the time stationary version $\Psi^*$ and $0 < E_T(N_1 \circ \Psi^*) < \infty$ a.s.. In this case, we have $E_T(N_1 \circ \Psi^*) = \{E_T(T_0 \circ \Psi^0)\}^{-1} = \lambda_T$, $\Psi^0$ is also the event stationary version of $\Psi^*$, and $\Psi^*$ is also the time stationary version of $\Psi^0$.

**Theorem A.10** If $\Psi$ has the event stationary version $\Psi^0$ and $0 < E_T(T_0 \circ \Psi^0) < \infty$ a.s. (or equivalently $\Psi$ has the time stationary version $\Psi^*$ and $0 < E_T(N_1 \circ \Psi^*) < \infty$ a.s.), then for all $E \in \mathcal{B}(M)$, we have the following empirical inversion formulas:

$$
Pr(\Psi^0 \in E) = E \left[ \frac{E_T \left[ \sum_{j=0}^{N_1 \circ \Psi^*} I_E \circ \Psi^*_j \right]}{E_T(N_1 \circ \Psi^*)} \right], \quad (A.18)
$$

$$
Pr(\Psi^* \in E) = E \left[ \frac{E_T \left[ \int_{T_0 \circ \Psi^0} I_E \circ \Psi^0 \, ds \right]}{E_T(T_0 \circ \Psi^0)} \right]. \quad (A.19)
$$

For all measurable $f : M \to \mathbb{R}_+$, we have the following conditional empirical inversion formulas:

$$
E_T(f \circ \Psi^0) = \frac{E_T \left[ \sum_{j=0}^{N_1 \circ \Psi^*} f \circ \Psi^*_j \right]}{E_T(N_1 \circ \Psi^*)} \quad a.s., \quad (A.20)
$$

and if in addition, $\int_0^t f \circ \Psi^0 \, ds, t \geq 0, a.s.$, then

$$
E_T(f \circ \Psi^*) = \frac{E_T \left[ \int_{T_0 \circ \Psi^0} f \circ \Psi^0 \, ds \right]}{E_T(T_0 \circ \Psi^0)} \quad a.s. \quad (A.21)
$$

The following corollary states the empirical inversion formulas under the ergodicity condition.
Corollary A.11 If $\Psi$ has the event stationary version $\Psi^0$, and $\Psi^0$ is ergodic, and $0 < E(T_0 \circ \Psi^0) < \infty$ (or equivalently $\Psi$ has the time stationary version $\Psi^*$, and $\Psi^*$ is ergodic, and $0 < E(N_1 \circ \Psi^*) < \infty$), then for all $\mathcal{E} \in \mathbf{M}$, we have the following ergodic empirical inversion formulas:

\begin{align*}
Pr(\Psi^0 \in \mathcal{E}) &= \frac{E \left[ \sum_{j=0}^{N_1 \circ \Psi^*-1} I_{\mathcal{E}} \circ \Psi^*_j \right]}{E(N_1 \circ \Psi^*)}, \quad (A.22) \\
Pr(\Psi^* \in \mathcal{E}) &= \frac{E \left[ \int_{0}^{T_0 \circ \Psi^0} I_{\mathcal{E}} \circ \Psi^0_s ds \right]}{E(T_0 \circ \Psi^0)}. \quad (A.23)
\end{align*}

For all measurable $f : \mathbf{M} \to \mathbb{R}_+$, we have

\begin{align*}
E(f \circ \Psi^0) &= \frac{E \left[ \sum_{j=0}^{N_1 \circ \Psi^*-1} f \circ \Psi^*_j \right]}{E(N_1 \circ \Psi^*)}, \quad (A.24)
\end{align*}

and if in addition, $\int_{0}^{t} f \circ \Psi^0_s ds, t \geq 0, \text{a.s.}$, then

\begin{align*}
E(f \circ \Psi^*) &= \frac{E \left[ \int_{0}^{T_0 \circ \Psi^0} f \circ \Psi^0_s ds \right]}{E(T_0 \circ \Psi^0)}. \quad (A.25)
\end{align*}
Bibliography


