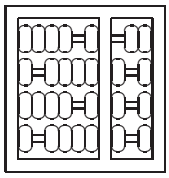

INSTITUTO DE COMPUTAÇÃO
UNIVERSIDADE ESTADUAL DE CAMPINAS

A Markov Model for Providing Quality of Service for Failure Detectors under Message Loss Bursts

I. Sotoma *E. R. M. Madeira*

Technical Report - IC-06-013 - Relatório Técnico

September - 2006 - Setembro

A Markov Model for Providing Quality of Service for Failure Detectors under Message Loss Bursts

Irineu Sotoma[a,1,*], Edmundo Roberto Mauro Madeira[a]

[a] *Institute of Computing, University of Campinas, 13084-971, Campinas, SP, Brazil*

Abstract

The quality of service (QoS) of failure detectors determines how fast a failure detector q detects the crash of a process p, and how accurate q informs the p crash. In wide area networks and wireless networks, the occurrence of process crashes, high delay variations and burst losses in message packets are common. So, this paper proposes a failure detector configurator which takes into account the probability distribution of loss burst lengths of message packets, by using a Markov model. The simulation results show that the parameters provided by the proposed configurator lead the failure detector to really satisfy the QoS requirements in networks subject to message loss bursts, while previous works do not satisfy them in some cases.

Keywords: Failure detectors; Quality of service; Fault tolerance; Distributed system.

1 Introduction

Failure detector is a system component which frequently monitors the state of other components. The monitorable components may belong to the same or to other systems. The goal of failure detectors usually is to inform timely and accurately about the ocurrence of failure in processes (processors), under reliable or unreliable communication channels. The areas of computer networks and distributed systems are plenty of examples which rely on failure detectors.

The field of computer networks uses timers as a kind of failure detectors, whereas the basic utilization of timers is present on the retransmission of TCP packets [38, 31]. Some other examples are congestion control [17], rate-based clocking [4], and soft-timer-based network polling [4].

The distributed systems field has extensively used failure detectors as a basic building block to solve several problems, e.g. consensus and group membership. In the consensus problem [16, 14, 15, 8], a set of processes proposes values to each other and they should agree on some value among those proposed. In the group membership problem [33, 12, 5], a set of processes should keep update a list of the currently active and connected processes.

[1] Present address: Department of Computing and Statistics, Federal University of Mato Grosso do Sul, 79070-900, Campo Grande, MS, Brazil.

[*] Corresponding author. Tel: +55 67 33457506; fax: +55 67 33457455. This research was supported by FAPESP, process number 00/05369-2, and CNPq, process number 62.0123/04-4. The email addresses are: isotoma@dct.ufms.br (I. Sotoma) and edmundo@ic.unicamp.br (E. R. M. Madeira).

There are also some works [32, 36] which study the cases in which is better to use only failure detectors or also other mechanisms to implement group membership. Several other problems can be solved in asynchronous systems extended with failure detectors, such as non-blocking atomic commitment, terminating reliable broadcast, and atomic register [22, 13].

Particularly, we are interested in the quality of service (QoS) of failure detectors. The Chen et al paper [11] formalizes the QoS of failure detectors by defining metrics to measure and evaluate the speed of crash detection and accuracy of the suspicion of process crashes. They developed a new failure detector (NFD-S) algorithm for synchronized clocks, and other algorithms for unsynchronized ones (NFD-U and NFD-E). They assumed only *message mean loss probability* as the system parameter for message losses. However, there are networks, such as WANs [2, 37, 39] and wireless networks [3, 31], where the occurrence of message loss bursts is frequent. Therefore, alternative models should be developed to take into account the loss bursts.

Markov chain models have been noted be adequate to model loss bursts in WANs [37, 39] and wireless networks [20, 24]. Sanneck [30] proposes an economic Markov chain model to loss bursts which needs only $m + 1$ states, unlike traditional ones which need 2^m states. m is the order of the Markov chain, and it represents the last consecutive losses which are considered by the Markov chain. His approach uses the probability distribution of loss burst lengths to approximate both state and state transition probabilities. Sotoma and Madeira [34] proposes a Markov model, based on the limited state space model of Sanneck, to model the QoS of failure detectors in the presence of message loss bursts.

In order to adequately cope to message loss bursts, we also use a Markov model based on the unlimited state space model of Sanneck, built from the probability distribution of loss burst lengths. The basic idea of the proposed model is similar to that of Sotoma and Madeira [34], which used a limited state space model. However, the use of the unlimited state space model, in this paper, required whole modification in that previous work.

A failure detector configurator takes as input a set of QoS requirements and network characteristics and outputs the failure detector parameters. This paper shows a new configurator which guarantees the QoS requirements even under the occurrence of loss bursts in the network. This guarantee is evaluated by simulations with geometric and Pareto probability distributions of loss burst lengths. The geometric probability distribution intends to assume a network less bursty (few long loss bursts), while the Pareto one intends to assume a network more bursty. The simulation results show that, under these network conditions, the proposed configurator lead the failure detector to satisfy the QoS requirements in all cases, while the Chen et al configurator [11] and the Sotoma and Madeira configurator [34] do not work well in some cases.

This paper is organized as follows. Section 2 shortly describes the basic Sanneck model, the Chen et al work, and related works which use the QoS metrics. Section 3 presents the new proposed Markov chain model for the configurators to NFD-S, and to NFD-U and their optimization. Section 4 details the simulation settings, and presents some analysis of the results. Finally, Section 5 offers some conclusions.

2 Background

The following subsections provide a short description of the loss burst model of Sanneck, the quality of service of failure detectors, and some related work on failure detectors and QoS.

2.1 Loss Run-Length Model

Sanneck [30] defined a model for loss run-length with a Markov chain with unlimited state space ($m + 1$ states) (see Figure 1). The random variable X is defined as follows: $X = 0$ means *no* lost packet, $X = z$ $(0 < z < m)$ means *exactly* z consecutive lost packets, $X \geq z$ means *at least* z consecutive lost packets. A state transition occurs depending on transition probabilities p_{ij}, with $i < j$ (for loss burst lengths lower than or equal to m) or $i \geq j = 0$ (for a packet arrival). The state probability of the system for $0 < z \leq m$ is $Pr(X \geq z)$, and for $z = 0$ is $Pr(X = 0)$.

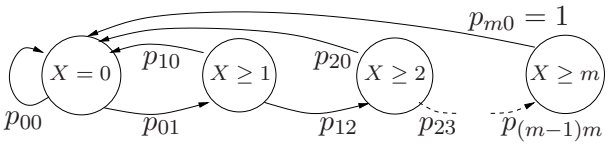

Figure 1. Sanneck model with unlimited state space.

2.2 QoS of Failure Detectors

This Section shortly describes the Chen et al assumptions to QoS of failure detectors. There is a failure detector process q which periodically verifies whether the monitored process p has crashed. They assume the following probabilistic network model:

1) p and q are connected by a link that does not create or duplicate messages, but may delay or drop messages;

2) *message loss probability* p_L is the probability of a message be dropped by the link, and *message delay* D is the delay from the time a message is sent to the time it is received, since the message is not dropped by the link;

3) the expected value $E(D)$ and the variance $V(D)$ of D are finite;

4) p and q have access to their own local clocks, which have no drift;

5) the probabilistic behavior of the network does not change over time;

6) the crashes can not be predicted;

7) the delay and loss behaviors of the messages that a process sends are independent of whether (and when) the process crashes;

8) the link from p to q has the *message independence* property: the behaviors of any two heartbeat messages sent by p are independent.

From the monitoring, q can output at a time t either S (suspicion of p crash) or T (trustiness that p is alive). An alternated changing between these outputs is called a *transition*. An *S-transition* occurs when the output of q changes from T to S; and a *T-transition* occurs when the output of q changes from S to T. The definitions of *S-transition* and

T-transition are used to define QoS metrics which characterize the speed and accuracy of the failure detectors. The QoS metrics are defined as random variables.

Detection time (T_D) is the QoS metric for failure detector speed, and describes how fast q detects the p crash. T_D represents the time that elapses from the time that p crashes to the time when the final *S-transition* (of q) occurs and there are no transitions afterward. There are QoS metrics for *accuracy* which describe how well q avoids mistakes. A mistake occurs when q outputs S, but p is still alive.

The two primary accuracy metrics are:

1) *Mistake recurrence time* (T_{MR}): the time that elapses from an *S-transition* to the next one. This indicates how long the failure detector q elapses between two mistakes about the p crash; and

2) *Mistake duration* (T_M): the time that elapses from an *S-transition* to the next *T-transition*. This indicates how long the failure detector q takes to change the incorrect information about the p crash.

Besides these two primary accuracy metrics, there are four accuracy metrics which are derived from T_{MR} and T_M:

1) *Average mistake rate* (λ_M): the rate at which a failure detector mistakes;

2) *Query accuracy probability* (P_A): the probability that the failure detector's output is correct at a random time;

3) *Good period duration* (T_G): the time that elapses from a *T-transition* to the next *S-transition*; and

4) *Forward good period duration* (T_{FG}): the time that elapses from a random time at which q trusts p to the time of the next *S-transition*.

Theorem 1 of Chen et al, at next, explains how the six accuracy metrics are related. $Pr(A)$ denotes the probability of event A; $E(X)$, $E(X^k)$, and $V(X)$ denote the expected value (or mean), the kth moment, and the variance of random variable X, respectively. A failure detector history is a sequence of outputs (S or T) which the failure detector provides. In failure-free runs, an ergodic failure detector is that which outputs histories which follow an ergodic probabilistic distribution. 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.

Theorem 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* $\lambda_M = 1/E(T_{MR})$ *and* $P_A = E(T_G)/E(T_{MR})$. *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 3a) for all* $x \in [0, \infty)$, $Pr(T_{FG} \leq x) = \int_0^x Pr(T_G > y)dy/E(T_G)$, *3b)* $E(T_{FG}^k = E(T_G^{k+1})/[(k+1)E(T_G)]$. *In particular, 3c)* $E(T_{FG}) = [1 + V(T_G)/E(T_G)^2]E(T_G)/2$.

The Figure 2 shows the NFD-S algorithm of Chen et al which assumes synchronized clocks. NFD-S has two parameters: η and δ. p sends heartbeat messages $m_1, m_2, \ldots$ to q every η time units. Henceforth, σ_i denotes the sending time of the message m_i. q shifts the σ_i's forward by δ to obtain the sequence of times $\tau_1 < \tau_2 < \ldots$, where $\tau_i = \sigma_i + \delta$, for $i \geq 1$. For $i = 0$, $\tau_0 = 0$. q uses the τ_i's and the times it receives heartbeat messages to determine if it trusts or suspects p, by using every time period $[\tau_i, \tau_{i+1})$. From time τ_i to τ_{i+1}, only messages m_j with $j \geq i$ can affect the output of q.

Process p:

1 for all $i \geq 1$, at time $\sigma_i = i\eta$, send heartbeat m_i to q;

Process q:

2 Initialization: $output = S$; {suspect p initially}

3 for all $i \geq 1$, at time $\tau_i = \sigma_i + \delta$:

4 **if** did not receive m_j with $j \geq i$ **then** $output \leftarrow S$; {suspect p if no fresh message is received}

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

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

Figure 2. Failure detector algorithm NFD-S with parameters η and δ (clocks are synchronized).

2.3 Failure Detectors and QoS

Unreliable failure detectors [8] allow monitored processes to be in and out of the list of suspected processes. By this way, even consensus problems can be solved in asynchronous systems extended with failure detectors where processes can crash and stop, and the communication channels are reliable. However, these failure detectors have their features classified according to completeness and accuracy properties independent of real implementation.

The usual implementation of unreliable failure detectors require some additional assumptions, e. g. partial synchrony, which assumes the system is not purely asynchronous. Even recent implementations of these failure detectors in partially synchronous systems [6, 23, 27] and in asynchronous systems [26] still assume that the communication channel is reliable. However, real communication channels are not reliable, but it is possible to construct quasi-reliable communication channels [32]. The work of Aguilera et al [1] is one of the first to address the development of failure detectors assuming process crashes and fair lossy links in asynchronous distributed systems. They define and show how to implement a failure detector called Heartbeat (HB), which should satisfy the properties of *HB-completeness* and *HB-accuracy*. However, these properties are not equivalent to completeness and accuracy properties defined by Chandra and Toueg [8].

Mostefaoui et al. [25] present the notion of oracles for asynchronous distributed systems. Barely, an oracle, defined by abstract properties, is a device with an interface, a set of operations, which allow to solve problems in asynchronous systems subject to failures. They present four oracles: a guessing failure detector (Θ), a hiding failure detector (Heartbeat), a failure detector ($\diamond S$), and a random oracle. These oracles solve some problems in asynchronous distributed systems subject to process crashes and fair lossy channels. The two first are used to solve quiescent URB (uniform reliable broadcast), and the other two

to solve consensus. The oracle $\diamond S$ is the unreliable failure detector $\diamond S$ defined by Chandra and Toueg [8]. Another oracle, defined to solve consensus in asynchronous distributed systems subject to process crashes and with reliable channels, is the leader oracle (Ω) [18, 9, 21].

Research on application of QoS metrics of failure detectors are beginning to arise. Bertier et al. [7] use an adaptation procedure, based on the Chen et al configurator for NFD-U algorithm, to calculate the safety margin and the emission interval, which are similar respectively to δ and η parameters (see Section 2.2). Then they evaluate their proposed hierarchical failure detection architecture by using the QoS metrics. They do not show if the required QoS is satisfied from the parameters outputted by the adaptation procedure, because this was not the focus of the paper. Nunes and Jansch-Pôrto [28], and Hayashibara et al. [19] utilize QoS metrics only to evaluate their proposed failure detectors. These works are interested in the adaptation of the failure detector parameters on-the-fly according to network changing conditions.

To the best of the authors' knowledge, the only papers, after the Chen et al work, which configure the failure detector from QoS requirements and then verify if the failure detector really satisfies the requirements, are the work of Sotoma and Madeira [34] and this paper. Additionally, these works are the first to address explicitly the problem of how to provide failure detector parameters from QoS requirements when the network conditions lead to message loss bursts.

3 A Model of Loss Bursts for Failure Detectors

This Section uses the Chen et al paper [11] as framework, and Chen thesis [10] to some proofs. Because Proposition 6 and Theorem 19 are the most important of the proposed model, its proofs are just after them, while the other proofs of the proposed model can be found in the Appendix A.

3.1 Modified Probabilistic Network Model

The probabilistic network model considered in the proposed model is the same of the Chen et al one (see Section 2.2), except by the following changes:

1) Besides the message loss probability (p_L) and message delay (D), the link between p and q also has the additional probability distribution of loss burst lengths, given by all $p_{L,z}$'s, according to Table 1 of Section 3.2.1. z is the length of a loss burst.

2) The *message independence* property is not required. There can be either an independent behavior of any two messages, or a dependent behavior of each message only with its predecessor one.

3.2 The Markov Model for Loss Bursts

The folowing subsections detail the basic Markov chain model, the NFD-S model for loss bursts, and the proposed NFD-S configurator.

3.2.1 The Basic Markov Chain Model

The Markov model of Sanneck [30] (see Section 2) is the basis for the Definition 2. The proposed Markov model has h states (1 to h) to message losses, and the state 0 to message arrivals. A sequence of message arrivals keeps the Markov chain in state 0. At each one more message loss, the state advances in the Markov chain, until the maximum loss burst length perceived (state h). Because this h definition, the probability of a state transition to a state greater than h is zero. At any state, if a message is received, a transition takes place and the Markov chain goes to state 0. The proposed NSM-NFD-S configurator, to be presented in Section 3.2.3, assumes the whole information in Definition 2 is already available when the failure detector configuration starts. This is possible due to approximations of the state and state transition probabilities, according to Table 1.

Definition 2.

1. *Z_n is a sequence of random variables with values within space $F = \{0, 1\}$. $Z_n = 0$ means a message was received by q, and $Z_n = 1$ means a message was lost.*

2. *h is the highest loss burst length which has been noted by q until the time at which the configuration takes place.*

3. *$S = [0, h]$, with $S \subseteq \mathbf{N}$, is the set of the possible states in the Markov chain.*

4. *$X_{n+1} = f(X_n, Z_{n+1})$ is the random variable which defines a Markov chain, with $X_n \in S$ and X_0 is the first observed state. If $X_n < h$, then $X_{n+1} = Z_{n+1}X_n + Z_{n+1}$; else if $X_n = h$, then $X_{n+1} = 0$, for $Z_{n+1} = 0$. If $X_n = h$ and $Z_{n+1} = 1$ the markov chain is not defined, according to item 2.*

5. *The meaning of the random variable X_{n+1}, defined in item 4, is as follows: $X_{n+1} = 0$ means no lost message, $X_{n+1} = z$ $(0 < z \leq h)$ means exactly z consecutive lost messages, $X_{n+1} \geq z$ means at least z consecutive lost messages.*

6. *The meaning of state and state transition probabilities, defined in item 4, is as follows. State transitions occur depending on transition probabilities p_{ij}, with $i < j$ (for loss burst lengths lower than or equal to h) or $i \geq j = 0$ (for a message arrival). The state probability of the system for $0 < z \leq h$ is $Pr(X_{n+1} \geq z)$, and for $z = 0$ is $Pr(X_{n+1} = 0)$.*

The Definitions 3 and 4, at next, simplifly the notation for the state transition probabilities of the Markov chain of the Definition 2. The Definition 3 shows the probability of performing state transitions corresponding to a loss burst with length $es - bs + 1$.

Definition 3. *The probability of forward state transitions, from a state bs to a state $es \geq bs$, is defined as $forw(bs, es) = \prod_{n=bs}^{es-1} p_{n(n+1)}$. From the* Definition 2*, when $n \geq h$, $forw(bs, es) = 0$.*

The Definition 4 shows the probability of performing a transition to state 0 when a message is received after a loss burst with length i, or when the message received follows another previously received message.

Definition 4. *The probability of a backward state transition, from a state i to the state 0 is defined as $to0(i) = p_{i0}$, from the* Definition 2*. $p_{i0} = 1 - p_{i(i+1)}$, for $i < h$, and $p_{h0} = 1$, for $i = h$.*

The Table 1, based on Sanneck [30], shows how the probabilities used by the Markov chain of the Definition 2 could be approximated, by using only the loss probabilities, called $p_{L,z}$, for every loss burst of length z. If this information is not available in advance, some probability distribution (e.g. uniform) could be assumed before the model usage.

Table 1. The Markov chain probability calculation.

Markov model with $h+1$ state	*a is the highest valid heartbeat message received*	$a \to \infty$
Burst loss $(0 < z \leq h)$	$p_{L,z} = \frac{o_z}{a}$ (o_z is the number of loss bursts of length z)	$Pr(X_{n+1} = z)$
Mean loss	$p_L = \sum_{z=1}^{h} \frac{z o_z}{a} = \sum_{z=1}^{h} z p_{L,z}$	$E[X_{n+1}]$
Cumulative loss $(0 < z \leq h)$	$p_{L,cum}(z) = \sum_{n=z}^{h} \frac{o_n}{a} = \sum_{n=z}^{h} p_{L,n}$	$Pr(X_{n+1} \geq z)$ (state probability)
Cumulative loss $(z = 0)$	$p_{L,cum}(0) = 1 - p_L$	$Pr(X_{n+1} = 0)$ (no loss case)
Conditional loss $(0 < z \leq h)$	$p_{L,cond}(z-1, z) = \frac{p_{L,cum}(z)}{p_{L,cum}(z-1)}$	$Pr(X_{n+1} \geq z \mid X_n \geq z-1)$ (state transition probability $p_{(z-1)z}$)

3.2.2 The NFD-S Model to Cope Loss Bursts

The Definition 5 uses the Markov chain of the Definition 2 for QoS of failure detectors in the presence of loss bursts. The Definition 5 characterizes the probability of an *S-transition* occurs in time τ_i, and what is needed to do it. k defines how many messages could be received within η, assuming a delay of δ. An *S-transition* corresponds to a message receipt before τ_i followed by no message receipt before τ_{i+1}. The first is determined by the probability q_0, and the second by $u(x)$, which is the probability of no one of the k messages arrives before τ_i due to message loss(es) or a delayed message.

Definition 5.

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 τ_i.*

2. *For any $i \geq 2$, let q_0 be the probability that q receives the message m_{i-1} before time τ_i. In this case, the Markov chain is in state 0.*

3. *For any $i \geq 1$, let $u(x)$ be the probability that q suspects p, by receiving no one of the messages m_{i+j}, for every $0 \leq j \leq k-1$, at time $\tau_i + x$, for all $x \in [0, \eta)$. This definition assumes the Markov chain is already in state 0 (definitions 5.2 and 5.4). Therefore, from state 0, the Markov chain takes transitions.*

4. *For any $i \geq 2$, let p_s be the probability that an S-transition occurs at time τ_i. This characterizes the whole Markov chain.*

The Proposition 6 at next only mathematically describes the Definition 5 in a way independent of i.

Proposition 6.

1. $k = \lceil \delta/\eta \rceil$.

2. $q_0 = Pr(X_{n+1} = 0)Pr(D < \delta + \eta)$.

3. *For all $x \in [0, \eta)$, and w initially equal to k, $u(x) = u_w(x)$. $u_w(x)$ is defined as follows:*

$$
\begin{aligned}
u_1(x) &= forw(0,1) + to0(0)Pr(D > \delta + x - (k-1)\eta), \text{ for } w = 1; \\
u_w(x) &= to0(0)Pr(D > \delta + x - (k-w)\eta)u_{w-1}(x) \\
&\quad + \sum_{a=1}^{w-2} forw(0,a)to0(a)Pr(D > \delta + x - (a+k-w)\eta)u_{w-(a+1)}(x) \\
&\quad + forw(0,w-1)to0(w-1)Pr(D > \delta + x - (k-1)\eta) \\
&\quad + forw(0,w), \text{ for } w > 1.
\end{aligned}
$$

4. $p_s = q_0 u(0)$.

In the Proposition 6.3: i) k messages could be received within $[\tau_{i-1}, \tau_i + x)$; ii) the total number of permutation of losses (represented by bits 1) and delays (represented by bits 0) is 2^k; and iii) the w index indicates how many messages are being considered, and from what one among $k - w$ to $k - 1$. For example, $w = k$ considers the messages 0 to $k - 1$, and $w = k - 1$ considers the messages 1 to $k - 1$. In the following, the Proposition 6.3 is detailed.

$u_1(x) = forw(0,1) + to0(0)Pr(D > \delta + x - (k-1)\eta)$ represents the probability of the $(k-1)$-th message be lost, or be delayed with a delay greater than $\delta + x - (k-1)\eta$ time units to it be not received within $[\tau_{i-1}, \tau_i + x)$.

$forw(0, w-1)to0(w-1)Pr(D > \delta + x - (k-1)\eta)$ in $u_w(x)$ considers the probability of patterns with suffix $1^{w-1}0$. $forw(0, w-1)$ gives the probability of the sequence of $w-1$ losses of messages $k-w$ to $k-2$. So $to0(w-1)Pr(D > \delta + x - (k-1)\eta)$ gives the probability of the $(k-1)$-th message be received with delay greater than $\delta + x - (k-1)\eta$.

$forw(0, w)$ in $u_w(x)$ considers the probability of patterns with suffix 1^w, i.e., the probability of the sequence of w losses of messages $k - w$ to $k - 1$.

$to0(0)Pr(D > \delta + x - (k-w)\eta)u_{w-1}(x)$ in $u_w(x)$ considers the probability of patterns with preffix 0. $to0(0)Pr(D > \delta + x - (k-w)\eta)$ means the probability of the delay of the $(k-w)$-th message be greater than $\delta + x - (k-w)\eta$ time units. $u_{w-1}(x)$ represents the recurrence which calculates the probabilities of the 2^{w-1} combinations of the following $w-1$ messages. This recurrence finishes when $w = 2$, by calling $u_1(x)$.

$\sum_{a=1}^{w-2} forw(0,a)to0(a)Pr(D > \delta + x - (a+k-w)\eta)u_{w-(a+1)}$ in $u_w(x)$ considers the probability of patterns which have prefix 1^a0, where a, $1 \leq a \leq w-2$, is the number of consecutive losses of messages. The probability resulting from losses are described by

$forw(0,a)$. $to0(a)Pr(D > \delta+x-(a+k-w)\eta)$ represents the probability of the $(a+k-w)$-th message be delayed more than $\delta + x - (a+k-w)\eta$. $u_{w-(a+1)}(x)$ represents the probability of the $2^{w-(a+1)}$ combinations of the following $w-(a+1)$ messages. This recurrence finishes when $a = w-2$, by calling $u_1(x)$.

Proof of Proposition 6.

1) The proof of the Proposition 6.1 is the same of Chen thesis: it is immediate from the fact that m_j is sent at time $\tau_i - \delta + (j-i)\eta$ for all $i \geq 1$.

2) The proof of the Proposition 6.2 directly follows from the fact that q_0 is the probability of m_{i-1} is not lost and is received with a delay less than $\delta + \eta$ time units, which leads to state 0 ($Pr(X_{n+1} = 0)$).

3) A strong induction proof follows to verify if the recurrence works. In the induction base, when $w = k = 1$, $u_1(x)$ represents the probability of the $(k-1)$-th message be lost, or be not received within $[\tau_{i-1}, \tau_i + x)$ because its delay is greater than $\delta + x - (k-1)\eta$ time units. When $w = k = 2$, $u_2(x)$ clearly uses the first, third, and fourth terms of $u_w(x)$ definition, and only $u_1(x)$ in the first term. In the induction hypothesis, we consider $u_w(x)$ works when $2 \leq w \leq k-1$. In the induction step, we verify if $u_w(x)$ works when $2 \leq w \leq k$. It is clear that when $w = k$, the first term of $u_w(x)$ uses $u_{w-1}(x)$, which by the induction hypothesis is calculated by $u_{k-1}(x)$. When $w = k$, the second term of $u_w(x)$ uses $u_{w-(a+1)}(x)$, where $w-(a+1)$ varies from $k-2$ to 1. Therefore, by the induction hypothesis, $u_{w-(a+1)}(x)$ is correctly calculated. The proofs of third and fourth terms directly follows. So, $u_w(x)$ works when $2 \leq w \leq k$.

4) From the Proposition 6.2, q outputs T, and from the Proposition 6.3, q outputs S, leading to an *S-transition*. So, p_s really is the probability that an *S-transition* occurs at time τ_i. □

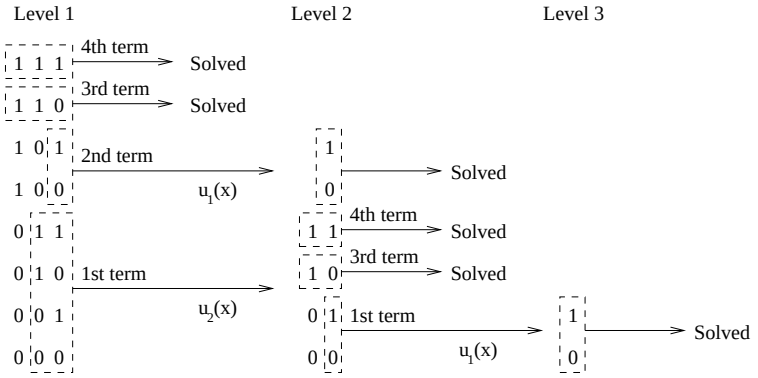

Figure 3. An example for the Prop. 6.3 when $k = 3$.

Figure 3 presents an example which uses the terms of the Proposition 6.3. $k = 3$, which leads to 3 levels of recursion and 2^3 permutations, each one with length k. This example shows the recursion in the building of the bit pattern:

At level 1: a) the first term treats patterns which begin with 0 (011, 010, 001, and 000), by

calling $u_2(x)$; b) the second term deals with patterns which begin with sequences of 1's and are followed by a sequence of 0's (101 and 100), except those patterns which have sequences of 1's and only the last bit is 0; the patterns 1 and 0 are treated by calling $u_1(x)$; c) the third term treats patterns where all bits are 1, except the last one which is 0 (110); and d) the fourth term manages patterns where there is only 1's (111).
At level 2: The patterns (11, 10, 01, and 00) are treated by $u_2(x)$, whose: a) first term treats the patterns 01 and 00, and calls $u_1(x)$; b) second term solves the pattern 10; and c) fourth term solves the pattern 11. The patterns 1 and 0 are solved by $u_1(x)$.
At level 3: the patterns 1 and 0 are solved by $u_1(x)$.

From the Figure 3 and Proposition 6.3, it should be clear that the idea is to generate all permutations of 0's and 1's, for a given k, and then make transitions in the Markov chain according to message losses (1's), and message receipts (0's). The probability of a forward transition is $forw$, and of a backward transition is $to0$. However, for a message receipt after $\tau_i + x$, for all $x \in [0, \eta)$, the backward transition becomes $to0Pr(D > y)$, where $y \in [0, \delta)$. Each permutation of length k has a probability to occur, and $u_k(x)$ is the sum of all the 2^k probabilities.

The Proposition 7 describes the nondegenerated cases where q eventually receives a valid message from p before the time τ_i ($q_0 > 0$), and eventually q suspects p ($u(0) > 0$). This eliminates the cases when q suspects p forever with probability one ($q_0 = 0$), and when q trusts p forever with probability one ($u(0) = 0$). It is used later on by the Proposition 17.

Proposition 7. *The nondegenerated cases $q_0 > 0$ and $u(0) > 0$, in the proposed model, occur when $Pr(D > \delta) > 0$, $Pr(D < \delta + \eta) > 0$, $0 < Pr(X_{n+1} = 0) < 1$, and $0 < Pr(X_{n+1} \geq 1) < 1$.*

The Definition 8 characterizes the probability for q to suspect p, and the Proposition 9 mathematically describes this probability. The main difference between Definition 5.3 and Definition 8 is that in the former the Markov chain is only in state 0 due to Definitions 5.2 and 5.4, and in Definition 8 the Markov chain can be in any state $s \in S$ (see Definition 2).

Definition 8. *For any $i \geq 1$, let $v(x)$ be the probability that q suspects p at time $\tau_i + x$, for every $x \in [0, \eta)$. This suspicion occurs when no one of the messages m_{i+j} is received by time $\tau_i + x$, for every $0 \leq j \leq k - 1$. Our $u'(x)$ assumes the Markov chain can be in any initial state $s \in S$.*

The Proposition 9 is similar to Proposition 6 but the Markov chain can be in any state $s \in S$.

Proposition 9. $v(x) = Pr(X_{n+1} = 0)v_{0,k}(x) + \sum_{s=1}^{h} Pr(X_{n+1} \geq s)v_{s,k}(x)$. *$u_{w-1}(x)$ and $u_{w-(a+1)}(x)$ use the $u(x)$ definition in the Proposition 6 and the Definition 8. $v_{s,w}(x)$,*

with w initially equal to k, is defined as follows:

$$v_{s,1}(x) = forw(s, s+1) + to0(s)Pr(D > \delta + x - (k-1)\eta), \text{ for } w = 1;$$

$$\begin{aligned} v_{s,w}(x) = {} & to0(s)Pr(D > \delta + x - (k-w)\eta)u_{w-1}(x) \\ & + \sum_{a=1}^{w-2} forw(s, s+a)to0(s+a)Pr(D > \delta + x - (a+k-w)\eta)u_{w-(a+1)}(x) \\ & + forw(s, s+w-1)to0(s+w-1)Pr(D > \delta + x - (k-1)\eta) \\ & + forw(s, s+w), \text{ for } w > 1. \end{aligned}$$

The following Lemma 14, which corresponds to the Lemma 16 of Chen et al paper, is still valid in our model, by using the p_s definition in the our Proposition 6.

Lemma 14. $E(T_{MR}) = \eta/p_s$.

The proof of Chen et al paper for its Lemma 18 is still valid for our following Lemma 15, because the same NFD-S algorithm is used.

Lemma 15. *$T_D \leq \delta + \eta$ and this bound is tight.*

The following theorem summarizes our QoS analysis of the NFD-S, under loss bursts, by using the previous definitions, lemmata, and propositions which follow the Definition 2.

Theorem 16. *Consider a system with synchronized clocks, where the probability of message loss p_L, the distribution of message delays $Pr(D \leq x)$, and the probability distribution of loss burst lengths are known. The failure detector NFD-S with parameters η and δ has the following properties:*

1. *The detection time is bounded as follows and the bound is tight: $T_D \leq \delta + \eta$.* (1.1)
2. *The average mistake recurrence time is: $E(T_{MR}) = \frac{\eta}{p_s}$.* (1.2)
3. *The average mistake duration is: $E(T_M) = \frac{\int_0^\eta v(x)d(x)}{p_s}$.* (1.3)

3.2.3 The Proposed NSM-NFD-S Configurator to Cope with Loss Bursts

Our goal is to find a configuration procedure, which takes as input the probabilistic behavior of heartbeats and the QoS requirements (T_D^U, T_{MR}^L, T_M^U), and outputs η and δ. Hereafter, we call configurator as a short for configuration procedure. T_D^U is an upper bound on the 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 QoS requirements are that:

$$T_D \leq T_D^U, E(T_{MR}) \geq T_{MR}^L, E(T_M) \leq T_M^U. \quad (1.4)$$

From the Theorem 16, the goal can be restated as a mathematical programming problem:

maximize η:

subject to $\delta + \eta \leq T_D^U$ (1.5)

$$\frac{\eta}{p_s} \geq T_{MR}^L \tag{1.6}$$
$$\frac{\int_0^\eta v(x)dx}{p_s} \leq T_M^U \tag{1.7}$$

where the value of $v(x)$ is given by the Proposition 9, and the value of p_s is given by the Proposition 6. Similar to Chen et al, the problem (1.7), which is hard to solve, was replaced by a simpler and stronger constraint as follows.

Proposition 17. *In the nondegenerated cases of the Proposition 7,* $E(T_M) \leq \frac{v(0)\eta}{q_0 u(0)}$.

From the problem (1.5) and Propositions 6 and 17, we obtain the following Proposition 18, which is used later on by the NSM-NFD-S configurator.

Proposition 18. *Let be* $k' = \lceil T_D^U/\eta \rceil - 1$. *At next,* $v'(0)$ *and* $u'(0)$ *consider, like Chen et al, only the messages* 0 *to* $k-1$. $v'(0) = Pr(X_{n+1} = 0)v_{0,k'}(0) + \sum_{s=1}^{h} Pr(X_{n+1} = s)v'_{s,k'}(0)$. $v'_{s,w}(0)$, *which is based on Proposition 9, is defined as follows:*

$$\begin{aligned}
v'_{s,1}(0) &= forw(s, s+1) + to0(s)Pr(D > T_D^U - k'\eta), \text{ for } w = 1;\\
v'_{s,w}(0) &= to0(s)Pr(D > T_D^U - (k' - w + 1)\eta)u'_{w-1}(0)\\
&\quad + \sum_{a=1}^{w-2} forw(s, s+a)to0(s+a)Pr(D > T_D^U - (a + k' - w + 1)\eta)u'_{w-(a+1)}(0)\\
&\quad + forw(s, s+w-1)to0(s+w-1)Pr(D > T_D^U - k'\eta)\\
&\quad + forw(s, s+w), \text{ for } w > 1.
\end{aligned}$$

The terms $u'_{w-1}(0)$ *and* $u'_{w-(a+1)}(0)$ *of* $v'_{s,w}(0)$ *use the following* $u'(0)$ *definition, which is based on Proposition 6:*

$$\begin{aligned}
u'_1(0) &= forw(0, 1) + to0(0)Pr(D > T_D^U - k'\eta), \text{ for } w = 1;\\
u'_w(0) &= to(0)Pr(D > T_D^U - (k' - w + 1)\eta)u'_{w-1}(0)\\
&\quad + \sum_{a=1}^{w-2} forw(0, a)to0(a)Pr(D > T_D^U - (a + k' - w + 1)\eta)u'_{w-(a+1)}(0)\\
&\quad + forw(0, w-1)to0(w-1)Pr(D > T_D^U - k'\eta)\\
&\quad + forw(0, w), \text{ for } w > 1.
\end{aligned}$$

From the problems (1.4), (1.5), (1.6), (1.7) and Propositions 17 and 18, we obtain the following configurator, called NSM-NFD-S configurator, to find η and δ:
Step 1: Compute $q'_0 = Pr(X_{n+1} = 0)Pr(D < T_D^U)$ and let $g(\eta) = v'(0)\eta/q'_0u'(0)$, where $v'(0) = v'_{k'}(0)$. If $q'_0u'(0) = 0$, then output *"QoS cannot be achieved"* and stop. Otherwise, find the largest $\eta_{max} \leq T_D^U$ such that $g(\eta_{max}) \leq T_M^U$.
Step 2: Let $f(\eta) = \eta/q'_0u'(0)$, find the largest $\eta \leq \eta_{max}$ such that $f(\eta) \geq T_{MR}^L$.
Step 3: Set $\delta = T_D^U - \eta$ and output η and δ.

Theorem 19. *Consider a system in which clocks are synchronized and the probability of message loss p_L, the distribution of message delays $Pr(D \leq x)$, and the probability distribution of loss burst lengths are known. Suppose we are given a set of QoS requirements as in (1.4). The NSM-NFD-S configurator has two possible outcomes: 1) It outputs η and δ. In this case, with parameters η and δ, the failure detector NFD-S satisfies the given QoS requirements. 2) It outputs "QoS cannot be achieved".*

Proof. We prove the theorem, based on Chen et al [11], in the following two parts:

1. Suppose that the NSM-NFD-S configurator outputs *"QoS cannot be achieved"*. Then, the configurator stops at *Step 1* and, thus, $q'_0u'(0) = 0$. $q'_0 = 0$ implies $Pr(D < T_D^U) = 0$ or $Pr(X_{n+1} = 0) = 1 - p_L = 0$. In these conditions, to satisfy $T_D \leq T_D^U$, q has to suspect p at a time $t > T_D^U$. Hence, for any failure detector, we have $E(T_M) = \infty$ and thus, it fails to satisfy $E(T_M) \leq T_M^U$. $u'(0) = 0$ implies there is no suspicion. So, $E(T_M) = g(\eta) = \infty$, which leads to the failure detector to fail to satisfy $E(T_M) \leq T_M^U$. Therefore, the failure detector can not satisfy the given QoS in this case.

2. Suppose that the NSM-NFD-S configurator outputs parameters η and δ. Then, by *Step 3*, we have $T_D^U = \eta + \delta$. By part 1 of Theorem 16, $T_D \leq T_D^U$ is satisfied. By *Step 1* and Proposition 6, $q'_0 = Pr(X_{n+1} = 0)Pr(D < \eta + \delta) = q_0$, and by *Step 1* and Proposition 18, $u'(0) = u(0)$. Note that we have $q'_0u'(0) > 0$ since, otherwise, $g(\eta) = \infty$, and the NSM-NFD-S configurator would output *"QoS cannot be achieved"* instead of η and δ. By Proposition 17 and *Step 1*, $E(T_M) \leq g(\eta) = \frac{v'(0)\eta}{q'_0u'(0)} = \frac{v(0)\eta}{q_0u(0)} \leq T_M^U$. So, $E(T_M) \leq T_M^U$ is satisfied. Thus, $f(\eta) = \eta/q'_0u'(0) = \eta/q_0u(0) = \eta/p_s = E(T_{MR})$ by (1.2). By *Step 2*, $f(\eta) \geq T_{MR}^L$ is satisfied. □

3.3 Optimization of the $u'(0)$ in Proposition 18

The $u'(0)$ in Proposition 18 takes exponential time to be executed, even if $forw$, $to0$, and the $Pr(D > y)$ are previously calculated. This occurs due to the recursive calls to $u'_{w-1}(0)$ and $u'_{w-(a+1)}(0)$. However, by analysing the recurrence, it is possible to devise an iterative solution (see the example in the Figure 3 of Section 3.2.2).

For all $x \in [0, \eta]$, and let a_u be an array which stores the $u'_w(x)$ values calculated for all $w \in [1, k']$. The terms $prefix\_0$, $prefix\_1^a0$, $suffix\_1^{w-1}0$, and $suffix\_1^w$ correspond, respectively to, the first, second, third and fourth terms of the $u'(0)$ definition in Proposition 18. To more details please refer to the explanation about the Proposition 6 in Section 3.2.2.

$u'(0) = calc\_u'(x, k')$, which is defined as follows:

$procedure\ calc\_u'(x, k')$

1 $a\_u[1] \leftarrow forw(0,1) + to0(0)Pr(D > T_D^U - k'\eta)$;
2 $for\ w \leftarrow 2\ to\ k'$
3 $prefix\_0 \leftarrow to(0)Pr(D > T_D^U - (k' - w + 1)\eta)a\_u[w-1]$;
4 $prefix\_1^a0 \leftarrow 0$;
5 $for\ a \leftarrow 1\ to\ w-2$
6 $prefix\_1^a0 \leftarrow prefix\_1^a0$
7 $+ forw(0,a)to0(a)Pr(D > T_D^U - (a + k' - w + 1)\eta)a\_u[w-(a+1)]$;
8 $suffix\_1^{w-1}0 \leftarrow forw(0, w-1)to0(w-1)Pr(D > T_D^U - k'\eta)$;
9 $suffix\_1^w \leftarrow forw(0, w)$;
10 $a\_u[w] \leftarrow prefix\_0 + prefix\_1^a0 + suffix\_1^{w-1}0 + suffix\_1^w$;

Assume that $forw$, $to0$, and the $Pr(D > y)$ are previously calculated before calling $calc\_u'(x, k')$, and the respective results put in additional arrays, which are used in a real implementation of $calc\_u'(x, k')$. So, from the two loops within the procedure, it is clear that this procedure takes quadratic (polynomial) time to execute.

4 Simulation of the Proposed Model

In this Section, we have plotted in Figures 4 to 7 the analytical outputs from the Chen et al's configurator, hereafter called CHEN-NFD-S and from our NSM-NFD-S configurator. Additionally, the figures present the behavior of the algorithm NFD-S (see Figure 2) under several loss burst length probability distributions. The Figure 8 compares the limited ([34]) and the unlimited Markov state space (this work) models.

4.1 The Simulation Settings

The simulations have basic settings similar to Chen et al [11]: the message delay D follows the exponential distribution, $T_D^U \in \{1, 3.5\}$ with steps of 0.1, $E(D) = 0.02$, $V(D) = 0.004$, $\eta = 1$, and $\delta = T_D^U - \eta = T_D^U - 1$. The exponential distribution was used due to its simple analytical representation: $Pr(D < x) = 1 - e^{-\lambda x}$, for $x \geq 0$. In our case we use $\lambda = 1/E(D)$. Moreover, this distribution is often used to model delays in communication channels [35]. Because we have performed simulations, the time unit is not used. It can be assumed that

all time values refer to the same time unit, e. g. seconds. The SM configurator used in the simulations was that defined in Section 3.3.

$h \in \{4, 8, 12\}$ is the maximum loss burst length used to generate the bursts (see Definition 2). In Figures 4 and 5, and Figures 6 and 7, for each value of T_D^U, respectively, we plotted $E(T_{MR})$ to consider the average of 300 mistake recurrence intervals, and $E(T_M)$ to consider the average of 300 mistake duration intervals. The loss burst length distributions used are the geometric [2] and Pareto [29]. As the burst length increases, the geometric distribution provides a sharp decrease in the loss burst probabilities, while the Pareto distribution provides a smooth one. We call long loss bursts to those ones that are close to h. So, the geometric distribution models a low occurrence of long loss bursts, and the Pareto one models a high occurrence of long loss bursts.

We have also performed simulations to verify if the required bound T_D^U was satisfied. For each value of T_D^U, h, p_L and both loss length distributions, in 50 runs with arbitrary crash times, the NFD-S algorithm always satisfied T_D^U.

For each simulation setting, the following sequence of steps was performed:

1) Generate randomically the network workload (the delays and the loss bursts);

2) Simulate the execution of the algorithm NFD-S using the workload. The $E(T_{MR})$ and $E(T_M)$ are calculated;

3) Execute the configurators (CHEN-NFD-S and NSM-NFD-S) and get the analytical QoS values to T_{MR}^L and T_M^U;

4) Plot the results from steps 2 and 3 in graphics;

5) Simulate the execution of the algorithm NFD-S using the workload. The T_D^U is calculated during this execution;

6) Plot the results from step 5 in graphics; and

7) Analyse the graphics from steps 4 and 6 by comparing the analytical QoS values with the $E(T_{MR})$, $E(T_M)$, and T_D values.

4.2 Calculation of Loss Burst Probabilities

The geometric distribution was calculated by using the following formula, adapted from Trivedi [35]: $loss\_burst\_probability_i = (1 - p_L)(p_L^i)/i$, for $i \in [1, h]$. The Pareto distribution was calculated indirectly by using the formula at next, also based on Trivedi [35]: $state\_probability_i = \alpha\beta^\alpha i^{-\alpha-1}$, for $i \geq \beta$, $\alpha > 0, i \in [1, h]$. β is a location paramater which indicates the least possible value to loss burst lengths. α is a shape parameter of Pareto distribution. From the $state\_probability_i$'s, the $loss\_burst\_probability_i$'s were calculated starting with $i = h$ until $i = 1$, by using the cummulative loss formula in Table 1.

The Pareto distribution used the parameter values $\alpha = 1.06$ and $\beta = 1$, which were found by Paxson [29] in TCP packets in Internet. Both calculations took care that the loss burst probabilities fitted within the p_L value, according to the mean loss calculation in Table 1.

4.3 Analysis of Mistake Recurrence Intervals

In this Section, because the results with $h \in \{4, 12\}$ are similar to those with $h = 8$, we show only the results to $h = 8$. The Figures 4 to 7 present the curves about mistake recurrence intervals. From the eq. (1.4) in Section 3.2.3, we say a configurator satisfies T_{MR}^L when its analytical output, indicated as analytic in the Figures 4 to 7, is lower than or equal to $E(T_{MR})$ obtained from the execution of the algorithm NFD-S.

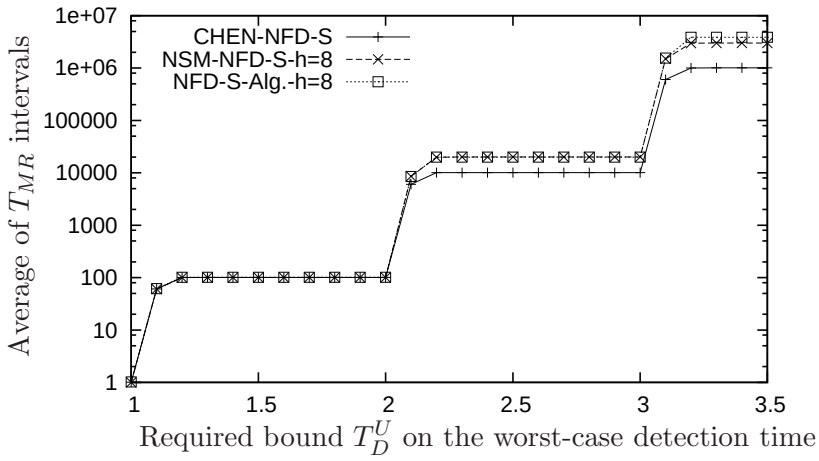

Figure 4. Check if $E(T_{MR}) \geq T_{MR}^L$ for geometric distribution and $p_L = 0.01$.

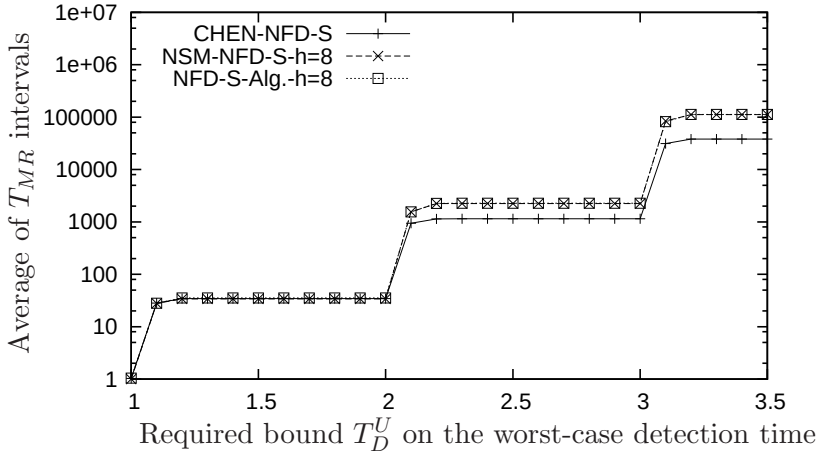

Figure 5. Check if $E(T_{MR}) \geq T_{MR}^L$ for geometric distribution and $p_L = 0.03$.

The Figures 4 and 5 show, respectively, the simulation results for $p_L = 0.01$ and $p_L = 0.03$, under geometric distribution on loss burst lengths. In these cases, both configurators satisfy the T_{MR}^L. In fact, for $T_D^U \in [1, 2]$, both configurators are equivalent. However, for $T_D^U \in [2.1, 3.5]$, the NSM-NFD-S configurator matches better with the NFD-S behavior under the workload. In this interval, the CHEN-NFD-S configurator underestimates the NFD-S behavior.

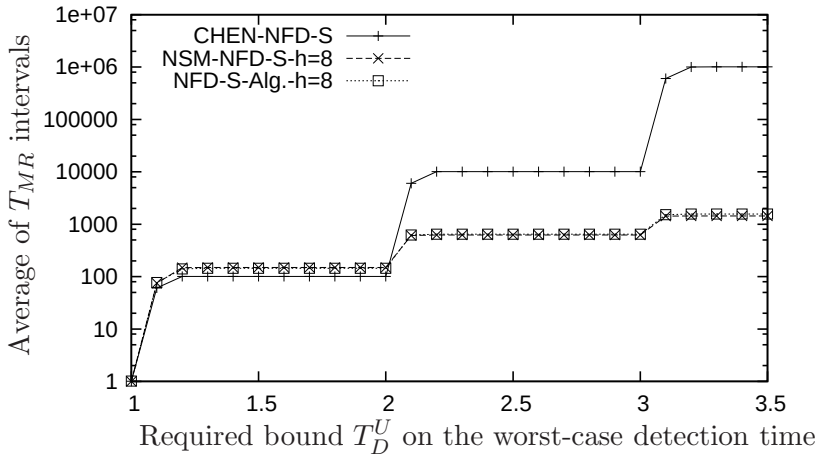

Figure 6. Check if $E(T_{MR}) \geq T_{MR}^L$ for Pareto distribution and $p_L = 0.01$.

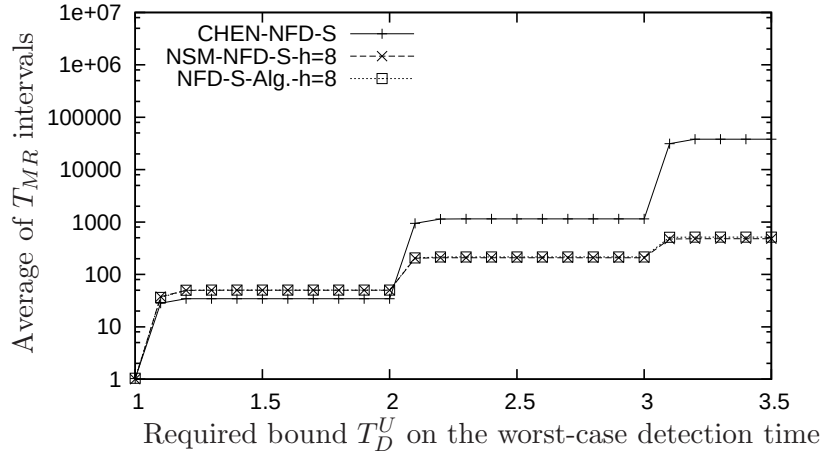

Figure 7. Check if $E(T_{MR}) \geq T_{MR}^L$ for Pareto distribution and $p_L = 0.03$.

The Figures 6 and 7 show, respectively, the simulation results for $p_L = 0.01$ and $p_L = 0.03$, under Pareto distribution on loss burst lengths. In these cases, only the NSM-NFD-S configurator satisfies the T_{MR}^L in all cases. In fact, for $T_D^U \in [1, 2]$, both configurators satisfy. However, for $T_D^U \in [2.1, 3.5]$, the NSM-NFD-S configurator matches better with the NFD-S behavior under the workload. In this interval, the CHEN-NFD-S configurator overestimates

the NFD-S behavior, leading to do not satisfy the T_{MR}^{L}.

The NSM-NFD-S configurator satisfies the T_{MR}^{L}, according to the Figures 4 to 7, within a 99% confidence interval, the same one obtained by the Chen et al paper for their simulation.

4.4 Analysis of Mistake Duration Intervals

The Figures 8 to 10 present the curves about mistake duration intervals. From the eq. (1.4) in Section 3.2.3, we say a configurator satisfies T_{M}^{U} when its analytical output, indicated as analytic in the figures 8 to 10, is greater than or equal to $E(T_M)$ obtained from the execution of the algorithm NFD-S.

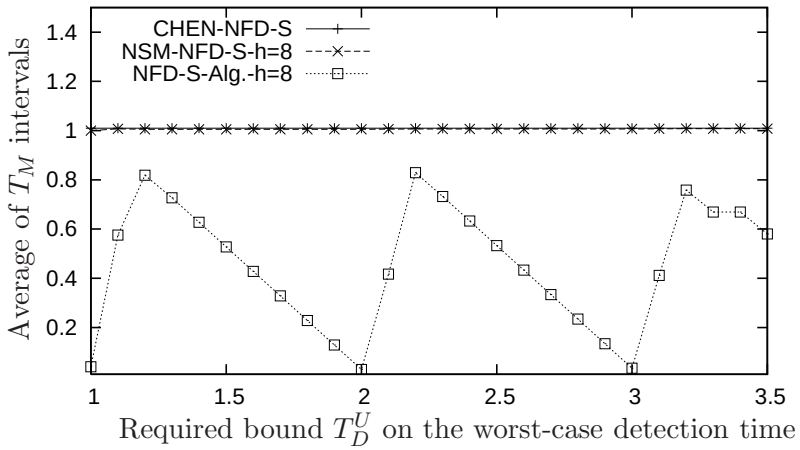

Figure 8. Check if $E(T_M) \leq T_M^U$ for geometric distribution and $p_L = 0.01$.

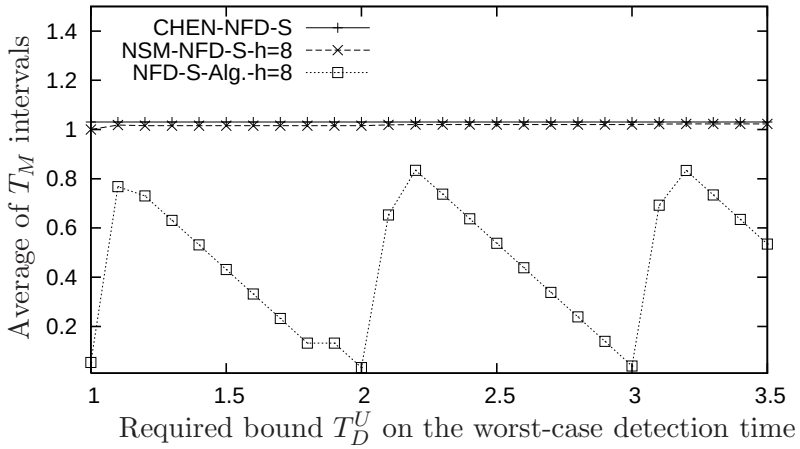

Figure 9. Check if $E(T_M) \leq T_M^U$ for geometric distribution $p_L = 0.03$.

The Figures 8 and 9 show, respectively, the simulation results for $p_L = 0.01$ and $p_L = 0.03$, under geometric distribution on loss burst lengths. Because in this case the results

with $h \in \{4, 12\}$ are similar to those with $h = 8$, we show only the results to $h = 8$ in Figure 8. The Figures 8 and 9 indicate that both NSM-NFD-S and CHEN-NFD-S configurators satisfy the T_M^U, and for $T_D^U \in [1, 3.5]$ both configurators are equivalent.

The Figures 10 and 11 show, respectively, the simulation results for $p_L = 0.01$ and $p_L = 0.03$, under Pareto distribution on loss burst lengths. Because in this case the results with $h \in \{4, 8, 12\}$ differ each other, we show all these in the Figures 10 and 11. The Figures 10 and 11 indicates that only the NSM-NFD-S configurator satisfies the T_M^U in all cases. In fact, in the $26 \times 3 = 78$ points for NFD-S behavior, the Chen configurator only satisfies the T_M^U in 33 cases for $T_D^U \in \{1, 1.1, 2.5\} \cup [1.4 - 2] \cup [2.6 - 3]$ in Figure 10, and 31 cases for the same intervals in Figure 11.

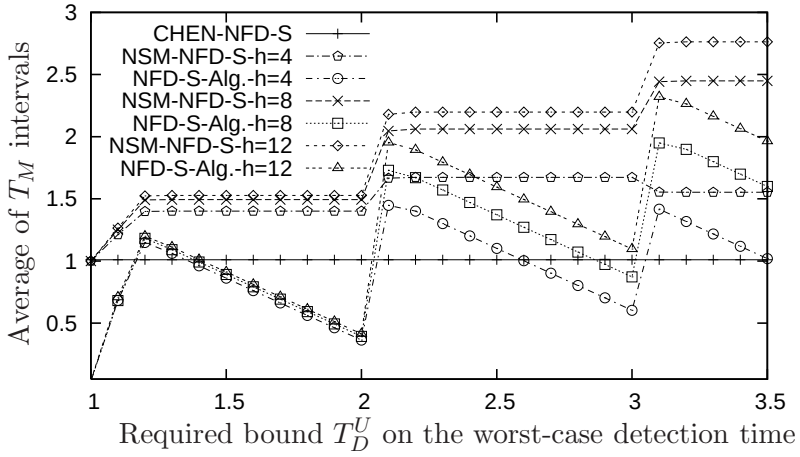

Figure 10. This simulation shows if $E(T_M) \leq T_M^U$ for Pareto distribution and $p_L = 0.01$.

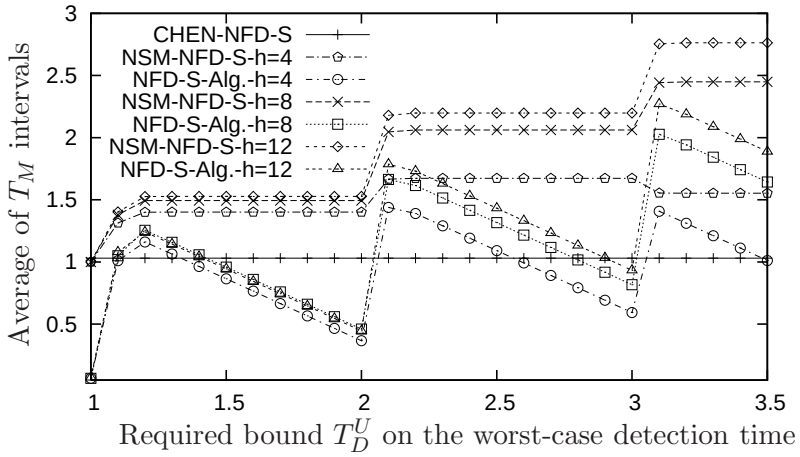

Figure 11. This simulation shows if $E(T_M) \leq T_M^U$ for Pareto distribution and $p_L = 0.03$.

4.5 Overall Discussion of the Simulation Results

The previously described results show that the NSM-NFD-S configurator leads the NFD-S algorithm to satisfy the QoS requirements both in detection time and detection accuracy. However, the CHEN-NFD-S configurator leads the NFD-S to satisfy the requirements of detection time, but fails in accuracy (mistake recurrence and mistake duration times).

The better performance of the NSM-NFD-S configurator is due the use of loss burst lenghts in the calculation of the analytical values, which does not occur with the CHEN-NFD-S configurator. Moreover, longer the loss burst length is, worser the CHEN-NFD-S configurator is. A similar behavior is obtained when the loss burst length probability distribution changes from a geometric to a Pareto one. Finally, the NSM-NFD-S configurator is better than CHEN-NFD-S, about the mistake duration time, because the occurrence of loss bursts leads the NFD-S algorithm to delay to recover from an S-transition.

4.6 Comparison between the Markov Model with Limited State Space and the Unlimited One

The Markov Model used in this paper was the loss run-length with unlimited state space (Section 2.1), unlike the Sotoma and Madeira [34] work, which used the limited one. The unlimited model proposed in this paper required a whole modification of that previous work. This section shows that the unlimited model works better than the limited model.

The work on the limited model only used in simulations the uniform distribution on loss burst lengths, while the current one uses geometric and Pareto distributions. There, the simulations were done with some arbitrary values of T_M^U, T_{MR}^L, and T_D^U, which did output both η and δ parameters different for each QoS requirement. Unlike that, the current work on the unlimited model uses an approach similar to the Chen et al work, which makes simulation analysis from a fixed η.

By using the same simulation settings of previous sections, we have compared the behavior of unlimited and limited models. The SM configurator, for the NFD-S algorithm, of Sotoma and Madeira [34] was reexecuted with the simulation settings of this paper. The Figure 12 shows the curves of T_{MR} to Pareto distribution, for $h = 4$ and $p_L = 0.01$. The proposed unlimited model fits better the NFD-S behavior, while the limited model gives lower values in $T_D^U \in [3.1, 3.5]$. This means the limited model gives lower η parameters in these cases, which lead to higher cost due to the higher heartbeat sending rate.

The Figures 12 and 13 shows the curves of T_M to Pareto distribution, for $h = 4$ and $p_L = 0.01$. The proposed unlimited model satisfies $E(T_M) \leq T_M^U$ for all cases, while the limited model fails to satisfy in $T_D^U \in [3.1, 3.3]$.

For both T_M and T_{MR}, what follows at next is valid. The simulation with Pareto distribution, for $h = 4$ and $p_L = 0.03$, also behaves similarly and is not shown. In the other simulations with the other settings, the limited model behaves similarly to the behavior of the unlimited model in Sections 4.3 and 4.4.

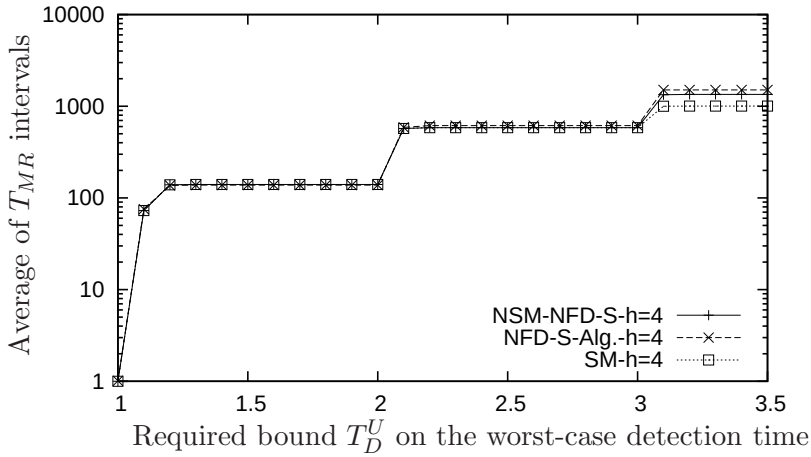

Figure 12. T_{MR} **comparison with** $p_L = 0.01$ **in unlimited and limited models.**

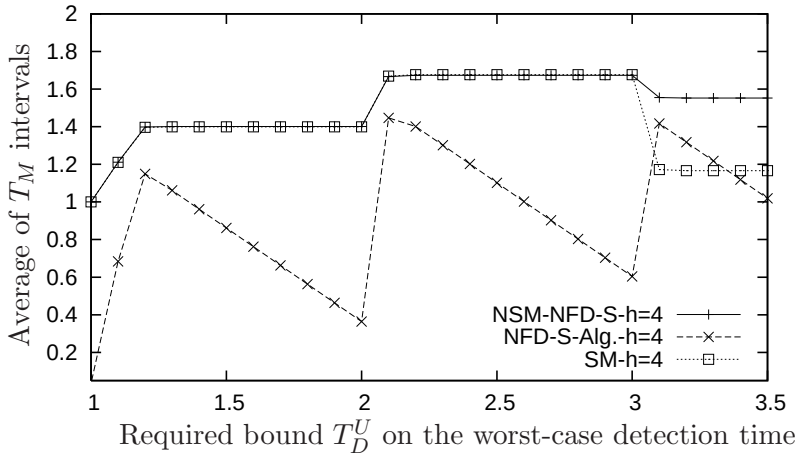

Figure 13. T_M **comparison with** $p_L = 0.01$ **in unlimited and limited models.**

5 Conclusions

This paper has extended the paper of Chen et al [11], by proposing a new failure detector configurator, which uses a Markov model, for QoS of failure detectors suitable to occurrence of loss bursts. The proposed Markov model is based on the Sanneck 30, and uses only $h+1$

states, where h is the maximum loss burst length perceived. The proposed configurator requires only a polynomial time to execute. After presenting the proposed configurator and its optimization, this configurator was evaluated under the geometric and Pareto distributions on loss burst lengths.

The Chen et al configurator [11], whose main goal was not to cope explicitly with the loss burst lengths, works well when the loss burst lengths follow a geometric distribution. In the presence of more agressive probability of loss burst lengths, like Pareto, the new proposed configurator provides QoS guarantees to the failure detector better than the Chen et al and Sotoma and Madeira [34] ones.

The proposed configurator is an important step to build configurators for wide area networks and wireless networks, where there are high delay variations and burst losses in message packets, since the delay probability distribution is known. Future research could focus on new configurators which do not require the delay distribution be known.

6 Acknowledgments

The authors would like to thank Jorge Stolfi by his useful discussion about some formulas in an earlier version of this work, Nancy Lopes Garcia by her helpful comments about Markov models and Probability.

7 References

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

[2] J. Andrén, M. Hilding, and D. Veitch. Understanding End-to-End Internet Traffic Dynamics. *Proceedings of IEEE 1998 Global Communications Conference (GLOBECOM 98)*, vol. 2, pp. 1118–1122. Sidney, Australia, November 1998.

[3] F. Anjum and L. Tassiulas. Comparative Study of Various TCP Versions Over a Wireless Link With Correlated Losses. *IEEE/ACM Transactions on Networking*, 11(3), pp. 370–383, June 2003.

[4] M. Aron, and P. Druschel. Soft Timers: Efficient Microsecond Software Timer Support for Network Processing. *ACM Transactions on Computer Systems*, 18(3):197–228, August 2000.

[5] Ö. Babaoglu, R. Davoli, and A. Montresor. Group Communication in Partitionable Systems: Specification and Algorithms. *IEEE Transactions on Software Engineering*, 27(4):308–336, April 2001.

[6] M. Bertier, O. Marin, and P. Sens. Implementation and performance evaluation of an adaptable failure detector. *Proceedings of the 2002 International Conference on Dependable Systems and Networks (DSN'02)*, pp. 354–363. June 2002.

[7] M. Bertier, O. Marin, and P. Sens. Performance Analysis of a Hierarchical Failure Detector. *Proceedings of the 2003 International Conference on Dependable Systems and Networks (DSN'03)*, pp. 635–644. June 2003.

[8] T. D. Chandra and S. Toueg. Unreliable Failure Detectors for Reliable Distributed Systems. *Journal of the Association for Computing Machinery*, 43(2):225–267, March 1996.

[9] T. D. Chandra, V. Hadzilacos and S. Toueg. The Weakest Failure Detector for Solving Consensus. *Journal of the Association for Computing Machinery*, 43(4):685–722, July 1996.

[10] W. Chen. *On the Quality of Service of Failure Detectors*. PhD thesis, Cornell University, Cornell, May 2000.

[11] W. Chen, S. Toueg, and M. K. Aguilera. On the Quality of Service of Failure Detectors. *IEEE Transactions on Computers*, 51(5):561–580, May 2002.

[12] G. V. Chockler, I. Keidar, and R. Vitenberg. Group Communication Specifications: A Comprehensive Survey. *ACM Computing Surveys*, 33(4):427–469, December 2001.

[13] C. Delport-Gallet, H. Fauconnier, R. Guerraoui, V. Hadzilacos, P. Kouznetsov, and S. Toueg. The Weakest Failure Detectors to Solve Certain Fundamental Problems in Distributed Computing. *Proceedings of the twenty-third annual ACM symposium on Principles of distributed computing(PODC'04)*, pp. 338–346. July 2004.

[14] D. Dolev, C. Dwork, and L. Stockmeyer. On the Minimal Synchronism Needed for Distributed Consensus. *Journal of the Association for Computing Machinery*, 34(1):77–97, January 1987.

[15] C. Dwork, N. Lynch, and L. Stockmeyer. Consensus in the Presence of Partial Synchrony. *Journal of the Association for Computing Machinery*, 35(2):288–323, April 1988.

[16] M. J. Fischer, N. A. Lynch, and M. S. Paterson. Impossibility of Distributed Consensus with One Faulty Process. *Journal of the Association for Computing Machinery*, 32(2):374–382, April 1985.

[17] S. Floyd. A Report on Recent Developments in TCP Congestion Control. *IEEE Communications Magazine*, 39(4):84–90, April 2001.

[18] R. Guerraoui and M. Raynal. The Information Structure of Indulgent Consensus. *IEEE Transactions on Computers*, 53(4):453–466, April 2004.

[19] N. Hayashibara, X. Défago, R. Yared, and T. Katayama. The φ Accrual Failure Detector. *Proceedings of the 23rd International Symposium on Reliable Distributed Systems (SRDS'04)*, pp. 66–78. October 2004.

[20] A. Konrad, B. Y. Zhao, A. D. Joseph, and R. Ludwig. A Markov-Channel Model Algorithm for Wireless Networks. *Wireless Networks*, 9, pp. 189–199, 2003.

[21] L. Lamport. The Part-Time Parliament. *ACM Transactions on Computer Systems*, 16(2):133–169, May 1998.

[22] M. Larrea. On the weakest failure detector for hard agreement problems. *Journal of Systems Architecture*, 49:345–353, 2003.

[23] M. Larrea, A. Fernandéz, and S. Arévalo. On the Implementation of Unreliable Failure Detectors in Partially Synchronous Systems. *IEEE Transactions on Computers*, 53(7):815–828, July 2004.

[24] H. Lee and P. K. Varshney. Gap-based Modeling of Packet Losses over the Internet. *Proceedings of 10th IEEE International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunications Systems (MASCOTS 2002)*, pp. 507–510. October 2002.

[25] A. Mostefaoui, E. Mourgaya, and M. Raynal. An introduction to oracles for asynchronous distributed systems. *Future Generation Computer Systems*, 18:757–767, 2002.

[26] A. Mostefaoui, E. Mourgaya, and M. Raynal. Asynchronous Implementation of Failure Detectors. *Proceedings of the 2003 International Conference on Dependable Systems and Networks (DSN'03)*, pp. 351–360. June 2003.

[27] A. Mostefaoui, D. Powell, and M. Raynal. A Hybrid Approach for Building Eventually Accurate Failure Detectors. *Proceedings of the 10th IEEE Pacific Rim International Symposium on Dependable Computing (PRDC'04)*, pp. 57–65. March 2004.

[28] R. C. Nunes, I. Jansch-Pôrto. QoS of Timeout-based Self-Tuned Failure Detectors: the Effects of the Communication Delay Predictor and the Safety margin. *Proceedings of the 2004 International Conference on Dependable Systems and Networks (DSN'04)*, pp. 753–761. June 2004.

[29] V. Paxson. End-to-end Internet Packet Dynamics. *IEEE Transactions on Networking*, 7(3):277–292, June 1999.

[30] H. Sanneck. *Packet Loss Recovery and Control for Voice Transmission over the Internet*. PhD thesis, Technischen Universität Berlin, Berlin, October 2000.

[31] P. Sarolahti, M. Kojo, and K. Raatikainen. F-RTO: An Enhanced Recovery Algorithm for TCP Retransmission Timeouts. *ACM SIGCOMM Computer Communications Review*, 33(2):51–63, April 2003.

[32] A. Schiper. Failure Detection vs Group Membership in Fault-Tolerant Distributed Systems: Hidden Trade-Offs. *Proceedings of Process Algebra and Probabilistic Methods. Performance Modeling and Verification: Second Joint International Workshop PAPM-PROBMIV 2002*, LNCS 2399, pp. 1–15, July 2002.

[33] A. Schiper. Group Communication: where are we today and future challenges. *Proceedings of the Third IEEE International Symposium on Network Computing and Applications (NCA'04)*, pp. 109–117, August-September 2004.

[34] I. Sotoma and E. R. M. Madeira. A Markov Model for Quality of Service of Failure Detectors in the Presence of Loss Bursts. *Proceedings of 18th International Conference on Advanced Information Networking and Applications (AINA 2004)*, vol. 2, pp. 62–67. Fukuoka, Japan, March 2004.

[35] K. S. Trivedi. *Probability and Statistics with Reliability, Queuing, and Computer Science Applications*. Second edition. New York: John Wiley & Sons. 2002.

[36] P. Urbán, I. Shnayderman, and A. Schiper. Comparison of Failure Detectors and Group Membership: Performance Study of Two Atomic Broadcast Algorithms. *Proceedings of the 2003 International Conference on Dependable Systems and Networks (DSN'03)*, pp. 645–654. June 2003.

[37] M. Yajnik, S. B. Moon, J. Kurose, and D. Towsley. Measurement and modeling of the temporal dependence in packet loss. *Proceedings of Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM'99)*, vol. 1, pp. 345–352. March 1999.

[38] L. Zhang. Why TCP Timers Don't Work Well. *ACM SIGCOMM Computer Communications Review*, 16(3):397–405, August 1986.

[39] Y. Zhang. *Characterizing End-to-end Internet Performance*. PhD thesis, Cornell University, Cornell, August 2001.

A Proof of Proposition 7

Proposition 7. *The nondegenerated cases $q_0 > 0$ and $u(0) > 0$, in the proposed model, occur when $Pr(D > \delta) > 0$, $Pr(D < \delta + \eta) > 0$, $0 < Pr(X_{n+1} = 0) < 1$, and $0 < Pr(X_{n+1} \geq 1) < 1$.*

Proof. If $q_0 = 0$ or $u(0) = 0$ then no *S-transition* would occur because $p_s = q_0 u(0)$ would be zero. Then we eliminate these degenerated cases and assume $q_0 > 0$ and $u(0) > 0$. From Proposition 6, $q_0 = Pr(X_{n+1} = 0)Pr(D < \delta + \eta)$. Since we assume $Pr(D < \delta + \eta) > 0$ and $Pr(X_{n+1} = 0) > 0$, and $Pr(X_0 = 0) = Pr(X_{n+1} = 0) > 0$ then $q_0 > 0$. From $u(x)$ definition in Proposition 6.4, $u(0) = u_k(0)$. In this case we also consider messages 0 to $k-1$. $u_w(0)$, with w initially equal to k, is then:

$$
\begin{aligned}
u_1(0) &= forw(0,1) + to0(0)Pr(D > \delta - (k-1)\eta), \text{ for } w = 1; \\
u_w(0) &= to0(0)Pr(D > \delta - (k-w)\eta)u_{w-1}(0) \\
&\quad + \sum_{a=1}^{w-2} forw(0,a)to0(a)Pr(D > \delta - (a+k-w)\eta)u_{w-(a+1)}(0) \\
&\quad + forw(0,w-1)to0(w-1)Pr(D > \delta - (k-1)\eta) \\
&\quad + forw(0,w), \text{ for } w > 1.
\end{aligned}
$$

When $k = 1$, the calculation of $forw(0,1)$ in Table 1 by using $0 < Pr(X_{n+1} = 0) < 1$ and $0 < Pr(X_{n+1} \geq 1) < 1$, leads to $forw(0,1)$ be greater than zero. So, $u(0) = u_1(0) > 0$. When $k > 1$, it is enough to prove that one of the terms of $u_w(0)$ is greater than 0. Then, by using Table 1 and only the first term of $u_w(0)$, when we generate the pattern 0^k, assuming $Pr(D > \delta) > 0$, $to0(0)Pr(D > \delta - (k-w)\eta)u_{w-1}(0) > 0$. So, $u(0) > 0$. □

B Proof of Proposition 9

Proposition 9. *$v(x) = Pr(X_{n+1} = 0)v_{0,k}(x) + \sum_{s=1}^{h} Pr(X_{n+1} \geq s)v_{s,k}(x)$. $u_{w-1}(x)$ and $u_{w-(a+1)}(x)$ use the $u(x)$ definition in the Proposition 6 and the Definition 8. $v_{s,w}(x)$, with w initially equal to k, is defined as follows:*

$$
\begin{aligned}
v_{s,1}(x) &= forw(s, s+1) + to0(s)Pr(D > \delta + x - (k-1)\eta), \text{ for } w = 1;\\
v_{s,w}(x) &= to0(s)Pr(D > \delta + x - (k-w)\eta)u_{w-1}(x)\\
&\quad + \sum_{a=1}^{w-2} forw(s, s+a)to0(s+a)Pr(D > \delta + x - (a+k-w)\eta)u_{w-(a+1)}(x)\\
&\quad + forw(s, s+w-1)to0(s+w-1)Pr(D > \delta + x - (k-1)\eta)\\
&\quad + forw(s, s+w), \text{ for } w > 1.
\end{aligned}
$$

Proof. The proof of $v(x)$ is immediate from the fact that from any Markov chain state is possible to loss a message (forward transition) or to receive a message (transition to state 0). Additionally, the $u(x)$ definition in Proposition 6 can be used because the Markov chain is always in state 0 when $u_{w-1}(x)$ and $u_{w-(a+1)}(x)$ are called in $v_{s,w}(x)$ definition. □

C Proof of Proposition 10

Proposition 10 is used later on in the proof of Proposition 17. The intuition here is that with more time to wait ($x > 0$), the probability of we have a suspicion ($u(x)$) is lower.

Proposition 10. *For all $x \in [0, \eta)$, $u(0) \geq u(x)$.*

Proof. From $u(x)$ definition in Proposition 6.4, $u(0) = u_k(0)$. In this case we also consider messages 0 to $k-1$. $u_w(0)$, with w initially equal to k, is then:

$$
\begin{aligned}
u_1(0) &= forw(0, 1) + to0(0)Pr(D > \delta - (k-1)\eta), \text{ for } w = 1;\\
u_w(0) &= to0(0)Pr(D > \delta - (k-w)\eta)u_{w-1}(0)\\
&\quad + \sum_{a=1}^{w-2} forw(0, a)to0(a)Pr(D > \delta - (a+k-w)\eta)u_{w-(a+1)}(0)\\
&\quad + forw(0, w-1)to0(w-1)Pr(D > \delta - (k-1)\eta)\\
&\quad + forw(0, w), \text{ for } w > 1.
\end{aligned}
$$

To prove that for all $x \in [0, \eta)$, $u(0) \geq u(x)$, it is enough to note $u(x)$ is influenced by $Pr(D > \delta + x - y\eta)$, where y can be $k-w$, $a+k-w$, or $k-1$. Since $0 \leq y \leq \delta$, $\delta + x - y\eta \geq 0$ for all $x \in [0, \eta)$. As greater the $\delta + x - y\eta$ value is, lower or equal is the $Pr(D > \delta + x - y\eta)$ value. Moreover, $Pr(D > \delta + x - y\eta)$ is always used in

multiplications followed by sums of positive numbers. Therefore, if $x_1 = 0$ and $x_2 \in [0, \eta)$, $Pr(D > \delta + x_1 - y\eta) \geq Pr(D > \delta + x_2 - y\eta)$, which implies $u(x_1) \geq u(x_2)$, leading to $u(0) \geq u(x)$. □

D Lemma 11

The following Lemma 11 is used later on by Theorem 16.3. Its proof is the same to that of Lemma 15 of Chen et al paper, except by the use of $v(x)$ of the our Proposition 9 instead $u(x)$. This Lemma provides the query accuracy probability, which is the probability of q is trusting p at a random time.

Lemma 11. $P_A = 1 - \frac{1}{\eta} \int_0^\eta v(x)dx$.

E Lemma 12

The following Lemma 12, which is the Lemma 17 of Chen et al paper, is still valid in our model. This Lemma, which is used later on by Theorem 14, presents a stochastic process defined by the random variables $T_{MR,n}$ and $T_{M,n}$. These variables represent, respectively, the period between two consecutive *S-transitions* and the period between a *S-transition* until a next *T-transition*. Thus $T_{M,n} \leq T_{MR,n}$ for all $n \geq 1$, and we get that for all $n \geq 2$, $T_{MR} = T_{MR,n}$, $T_M = T_{M,n}$, and $T_G = T_{MR} - T_M$, which is the relation found in Theorem 1.1 [10].

Lemma 12. $\{(T_{MR,n}, T_{M,n}), n = 1, 2, \ldots\}$ *is a delayed renewal reward process.*

F Lemma 13

The following Lemma 13, which is the Lemma 4 of Chen et al paper, is still valid in our model because our model uses the same NFD-S algorithm (see Figure 2). This Lemma allows the NFD-S analysis to use the Theorem 1 (see Section 2.2).

Lemma 13. *NFD-S is an ergodic failure detector.*

G Proof of Theorem 16

Theorem 16. *Consider a system with synchronized clocks, where the probability of message loss p_L, the distribution of message delays $Pr(D \leq x)$, and the probability distribution of loss burst lengths are known. The failure detector NFD-S with parameters η and δ has the following properties:*

1. *The detection time is bounded as follows and the bound is tight:* $T_D \leq \delta + \eta$. (1.1)
2. *The average mistake recurrence time is:* $E(T_{MR}) = \frac{\eta}{p_s}$. (1.2)
3. *The average mistake duration is:* $E(T_M) = \frac{\int_0^\eta v(x)d(x)}{p_s}$. (1.3)

Proof. The parts 1 and 2 of the theorem are direct from Lemmas 15 and 14. Part 3 is derived from the relation between $E(T_M)$, P_A, and $E(T_{MR})$, as given in part 2 of the Theorem 1 and the results on P_A and $E(T_{MR})$ as given by Lemmas 11 and 14. □

H Proof of Proposition 17

Proposition 17. *In the nondegenerated cases of the Proposition 7,* $E(T_M) \leq \frac{v(0)\eta}{q_0 u(0)}$.

Proof. This proof is the same of that of Lemma 16 of Chen et al paper, except by the use of $u(0) > 0$ of the Proposition 7. For all $i \geq 2$, they let A_i be the event that an *S-transition* occurs at time τ_i. By the use of $u(0) > 0$ is possible to assert that $Pr(A_i) = p_s = q_0 u(0) > 0$ in nondegenerated cases. By Proposition 10, $u(0) \geq u(x)$, for all $x \in [0, \eta)$, which is also valid to $v(0) \geq v(x)$. Thus, from equality (1.3) and Proposition 7: $E(T_M) = \frac{\int_0^\eta v(x)dx}{p_s} \leq \frac{\int_0^\eta v(0)dx}{q_0 u(0)} = \frac{v(0)\eta}{q_0 u(0)}$.

□

I Proof of Proposition 18

Proposition 18. *Let be* $k' = \lceil T_D^U/\eta \rceil - 1$. *At next,* $v'(0)$ *and* $u'(0)$ *consider, like Chen et al, only the messages* 0 *to* $k-1$. $v'(0) = Pr(X_{n+1} = 0)v_{0,k'}(0) + \sum_{s=1}^{h} Pr(X_{n+1} \geq s)v'_{s,k'}(0)$. $v'_{s,w}(0)$, *which is based on Proposition 9, is defined as follows:*

$$
\begin{aligned}
v'_{s,1}(0) &= forw(s, s+1) + to0(s)Pr(D > T_D^U - k'\eta), \text{ for } w = 1; \\
v'_{s,w}(0) &= to0(s)Pr(D > T_D^U - (k' - w + 1)\eta)u'_{w-1}(0) \\
&\quad + \sum_{a=1}^{w-2} forw(s, s+a)to0(s+a)Pr(D > T_D^U - (a + k' - w + 1)\eta)u'_{w-(a+1)}(0) \\
&\quad + forw(s, s+w-1)to0(s+w-1)Pr(D > T_D^U - k'\eta) \\
&\quad + forw(s, s+w), \text{ for } w > 1.
\end{aligned}
$$

The terms $u'_{w-1}(0)$ and $u'_{w-(a+1)}(0)$ of $v'_{s,w}(0)$ use the following $u'(0)$ definition, which is based on Proposition 6:

$$\begin{aligned}
u'_1(0) &= forw(0,1) + to0(0)Pr(D > T_D^U - k'\eta), \text{ for } w = 1;\\
u'_w(0) &= to(0)Pr(D > T_D^U - (k' - w + 1)\eta)u'_{w-1}(0)\\
&\quad + \sum_{a=1}^{w-2} forw(0,a)to0(a)Pr(D > T_D^U - (a + k' - w + 1)\eta)u'_{w-(a+1)}(0)\\
&\quad + forw(0,w-1)to0(w-1)Pr(D > T_D^U - k'\eta)\\
&\quad + forw(0,w), \text{ for } w > 1.
\end{aligned}$$

Proof. By part 1 of Theorem 16, we use $T_D \le T_D^U = \eta + \delta$ to prove $u'(0) = u(0)$ in the following:
When $w = k = 1$:

$$\begin{aligned}
u_1(0) &= forw(0,1) + to0(0)Pr(D > \delta - (k-1)\eta), \text{ for } k = \lceil \delta/\eta \rceil;\\
&= forw(0,1) + to0(0)Pr(D > \eta + \delta - k'\eta), \text{ for } k = \lceil \delta/\eta \rceil = \lceil 1 + \delta/\eta \rceil - 1 = k';\\
&= forw(0,1) + to0(0)Pr(D > \eta + \delta - k'\eta), \text{ for } k' = \lceil (\eta + \delta)/\eta \rceil - 1;\\
&= forw(0,1) + to0(0)Pr(D > T_D^U - k'\eta), \text{ for } k' = \lceil T_D^U/\eta \rceil - 1;\\
&= u'_1(0).
\end{aligned}$$

When $w = k > 1$:

$$\begin{aligned}
u_w(0) &= to0(0)Pr(D > \delta - (k-w)\eta)u'_{w-1}(0)\\
&\quad + \sum_{a=1}^{w-2} forw(0,a)to0(a)Pr(D > \delta - (a + k - w)\eta)u'_{w-(a+1)}(0)\\
&\quad + forw(0,w-1)to0(w-1)Pr(D > \delta - (k-1)\eta)\\
&\quad + forw(0,w), \text{ for } k = \lceil \delta/\eta \rceil;\\
&= to0(0)Pr(D > \eta + \delta - (k' - w + 1)\eta)u'_{w-1}(0)\\
&\quad + \sum_{a=1}^{w-2} forw(0,a)to0(a)Pr(D > \eta + \delta - (a + k' - w + 1)\eta)u'_{w-(a+1)}(0)\\
&\quad + forw(0,w-1)to0(w-1)Pr(D > \eta + \delta - k'\eta)\\
&\quad + forw(0,w), \text{ for } k = \lceil \delta/\eta \rceil = \lceil (\eta + \delta)/\eta \rceil - 1 = k';
\end{aligned}$$

$$
\begin{aligned}
&= to0(0)Pr(D > T_D^U - (k' - w + 1)\eta)u'_{w-1}(0) \\
&+ \sum_{a=1}^{w-2} forw(0,a)to0(a)Pr(D > T_D^U - (a + k' - w + 1)\eta)u'_{w-(a+1)}(0) \\
&+ forw(0,w-1)to0(w-1)Pr(D > T_D^U - k'\eta) \\
&+ forw(0,w), \text{ for } k' = \lceil T_D^U/\eta \rceil - 1; \\
&= u'_w(0).
\end{aligned}
$$

So, $u'(0) = u(0)$. The proof of $v'(0)$ follows directly from the proof of $v(0)$ of Proposition 9 and the proof above on $u'(0) = u(0)$. The only change is the use of the state s on probabilities $to0$ and $forw$. □