情報処理学会研究報告
IPSJ SIG Technical Report

Vol.2009-DPS-141 No.28
Vol.2009-GN-73 No.28
Vol.2009-EIP-46 No.28
2009/11/27

アクルーアル故障検出器を用いた故障検出精度の制御

林　原　尚　浩[†1]

本研究では，様々なネットワーク環境における故障検出器 ACCMOS の性能評価を行った．ACCMOS は，アクルーアル故障検出方式に基づき，ネットワークの状況に適応した適切なシステムの監視を行う故障検出器である，また，この故障検出器はタイムアウトを用いず，監視対象が故障している度合いを示す suspicion level という値によって故障を判定する．様々なネットワーク環境で実験を行うことによって，ACCMOS の性能評価に加え，suspicion level とタイムアウトの関係などを考察する．

Control of Accuracy with the ACCMOS Failure Detector

Naohiro HAYASHIBARA[†1]

In this paper, I show performance results of the ACCMOS failure detector in several environments. ACCMOS can dynamically adapt to the network condition that is dynamically changed, and can adapt to diverse requirements of users simultaneously. It provides information of monitored nodes as the continuous scale, called suspicion level, instead of binary values. We discuss the suspicion level and the performance result derived from different environments.

1. はじめに

企業などのサービスを提供しているシステムは，サーバ等の故障によるサービスの停止で多大な損害を被る可能性があるため，管理者を置いて常時，システムの監視を行っており，管理者支援のためのモニタリングシステムも多く開発されている．一方，FTTH などの広帯域ネットワークの普及により，家庭内でも小規模なネットワーク (ホームネットワーク) を構築し，コンピュータなどを接続しているが，ホームネットワークでは企業で導入しているようなモニタリングシステムは，設置や監視のための設定などが複雑なため導入することは困難である．また，このようなモニタリングシステムは機能が豊富である一方で，システム管理の経験がない人にとっては不必要な情報まで提供するため，逆にシステムの状況を理解することが難しくなる場合もある．

本研究では，ホームネットワーク向けのユーザビリティの高いモニタリングシステムの開発を目的としている．その第一段階として，監視対象ノードをモニタする故障検出器 ACCMOS の実装を行い，性能評価を行った．この故障検出器は，2004 年に林原らによって提案され，近年 Cassandra [6] や OurGrid [2], APPIA [3] などの実用的なサービスやプラットフォームで採用されている，アクルーアル故障検出器 (Accrual Failure Detectors) [4,5] に基づいて実装されている．

2. 故障検出器

2.1 タイムアウト型故障検出器

一般的な故障検出器の多くは，このタイムアウト型故障検出器に分類される．このタイプの故障検出器は，まず監視対象ノード p の故障を判定するためのタイムアウト値 Δ_{to} をパラメータとしてユーザ，もしくはプロセスから与えられる．初期状態において，全ての監視対象ノードは $trust$ という状態に設定されている．

その後，一定時間 Δ_i 毎に p と q の間でメッセージ通信を行い，p からのメッセージが q において Δ_{to} 以内に受信できない場合，p が故障の疑いがあるため，$suspect$ という状態へ移行する．ユーザやプロセスが p の状態を問い合わせた場合，故障検出器はその時点での p の状態 $\{trust, suspect\}$ を返す．

2.1.1 タイムアウト型故障検出器の特徴

タイムアウトを用いたモニタリングシステムや故障検出器における故障検出の精度は，タイムアウトが適切に設定されているかどうかに依存する．しかし，監視対象ノードの所属するネットワークの特性を熟知した管理者の経験や勘がないと，ネットワークの状況に最適なタイムアウトの値を設

†1 京都産業大学 コンピュータ理工学部
Faculty of Computer Science and Engineering, Kyoto Sangyo University

Vol.2009-DPS-141 No.28
Vol.2009-GN-73 No.28
Vol.2009-EIP-46 No.28
2009/11/27

定することは困難である．また，ネットワークの状況も絶えず変化することが多い．ネットワーク適応型の故障検出器も存在するが，多くのユーザの異なる故障検出に対する要求 [⋆1]を同時に実現するためには，それぞれの要求に対応したタイムアウトを管理する必要があり，スケーラビリティに難がある．

2.2 アクルーアル故障検出器

アクルーアル故障検出器は，故障検出器の理論的なモデルである [4]．従来の非信頼性故障検出器モデルが非同期システムでは実装不可能なモデルであるのに対し，このモデルは実装可能であり，さらに非信頼性故障検出器モデルへの変換アルゴリズムを持つ [5]．これは非信頼性故障検出器を前提とした全ての耐故障分散アルゴリズムに適用可能であることを示し，理論と実装の橋渡しをする役割を担っている．

アクルーアル故障検出器は，監視対象ノード (もしくは，プロセス) の故障している度合いを示す *suspicion level* という値を出力する故障検出器として定義されている．今，監視元ノード q が監視対象ノード p を監視している場合を考える．アクルーアル故障検出器はノード q 上で動作している．p について時刻 t に問い合わせを行った場合，アクルーアル故障検出器は時刻 t における p の故障している度合いを示す連続的な値を出力する．

2.2.1 アクルーアル型 φ 故障検出器

φ 故障検出器 (φ Failure Detector) [4] は，アクルーアル故障検出器モデルに基づいた故障検出器のプロトタイプ実装である．この故障検出器は，故障検出器とユーザやアプリケーションを含めた故障検出に関する一連の処理を以下の 3 つのステップに分けて設計されている．

(1) **モニタリング**. 監視対象ノードと通信を行うことによって，その状態を予測するための情報の収集，解析を行う．
(2) **インタープリテーション**. モニタリングによって得た情報により，監視対象ノードの故障判定を行う．
(3) **アクション**. ユーザやアプリケーションが，故障判定を受けて行う動作を指す．

⋆1 「故障検出に時間がかかっても良いから誤検出を低減したい」,「精度よりも故障検出にかかる時間を短くしたい」などの要求が考えられる．

タイムアウト型故障検出器モデルに基づいた多くの実装は，上記のモニタリングとインタープリテーションを故障検出器内部で行っている．特に，インタープリテーションはタイムアウトによって故障判定するため，ボトルネックになり監視対象ノードの情報を要求するユーザやアプリケーションが増えるとパフォーマンスが低下する．

φ 故障検出器はこの問題を解決するために，インタープリテーションを故障検出器から切り離し，ユーザやアプリケーションにおいて行うことによってスケーラビリティの向上を図っている．φ 故障検出器は，モニタリングにおいて監視対象ノード p から送られてくるハートビートの受信間隔を蓄積する．蓄積したデータが正規分布に従っていると仮定し，正規分布の累積密度関数を用いて時刻 t_{now} における $susp\_level_p(t_{now})$ を表す値 φ_p をユーザやアプリケーションに提供する．

ユーザやアプリケーションは個別に φ_p に対するしきい値 Φ_p を設定し，独立して故障判定を行う（例えば，$\varphi_p > \Phi_p$ となったときに故障と判定）．また，φ_p は連続的な値であるため，一つのアプリケーションに複数のしきい値 $\Phi_p^1, \Phi_p^2, \ldots, \Phi_p^n$ を設定し，φ_p の推移に応じて段階的に，監視対象ノード p の故障に備えた処理を行うこともできる．

φ 故障検出器は，タイムアウト型故障検出器が時間軸であるタイムアウトを固定して故障判定を行うのに対し，要求する故障検出精度を入力として与え，ネットワークの状況と入力された故障検出精度に合わせた故障検出を行う画期的な故障検出器である．

φ 故障検出器と前述の適応型故障検出器との比較実験の結果，ネットワーク適応性に関して同等の性能を持つことが確認された [4]．これによって，φ 故障検出器はアクルーアル故障検出器モデルが持つ，ユーザ要求への適応性やスケーラビリティなどの利点に加え，ネットワーク適応性においても高い性能を持つことを示した．

2.2.1.1 アクルーアル故障検出器モデル及び φ 故障検出器の実用例

世界中で広くサービスを提供するソーシャルネットワーキングサービス Facebook [1] の大規模分散データベース管理システム Cassandra [6] や通信フレームワーク APPIA [3]，グリッドミドルウェアである Our Grid [2] においては透過的なグリッドノードへのアクセスを提供する JIC(Java Internet Communication) [7] などでアクルーアル故障検出器モデルに基づ

Vol.2009-DPS-141 No.28
Vol.2009-GN-73 No.28
Vol.2009-EIP-46 No.28
2009/11/27

いた故障検出器が実装されている．しかし，これらはそれぞれのプラットフォーム依存の実装であり，一般的なモニタリングシステムとして使用することはできない．

3. ACCMOS の設計と実装

本研究では，ホームネットワークなどの中小規模のシステム向けのモニタリングシステムを開発している．このモニタリングシステムは，初期設定の自動化機構，アクルーアル型故障検出器，GUI などのコンポーネントにより構成される．

本稿では，このモニタリングシステムの中核となる故障検出器 ACCMOS の実装と性能評価に焦点を当てる．

アクルーアル故障検出器モデルに基づいて実装された ACCMOS は，従来のタイムアウト型故障検出器が監視対象ノード p の状態を二値 (例えば，$\{trust, suspect\}$) で表現していたのに対し，φ 故障検出器同様，suspicion level φ_p という連続的な値で表現することによって，より詳細な監視対象ノードの状態を提供することができる．

本章では，一般的なモニタリングシステムの中核となる故障検出器 ACCMOS の実装の詳細について述べる．

3.1 監視対象のモニタリング

今，故障検出器 ACCMOS が動作しているノードを q とし，監視対象ノードを p とする．q が p を監視するためには，それらの間で通信を行う必要がある．φ 故障検出器ではハートビート型の実装を行っている [4]．この場合は，監視対象ノード p にハートビートメッセージを一定間隔 Δ_i に送信する実装を導入し，Δ_i は p と q の間で共有する必要がある．Ping 型の実装は，RFC792 で標準化されている ICMP (Internet Control Message Protocol) を用いて行うため，p に特別な実装を施す必要が無く，また多種のノードを監視することができる．そのうえ，Δ_i は q の故障検出器のみ知っていればよいため，メッセージの送信間隔は故障検出器が任意に設定することができる．

ACCMOS では，より多種のノードを監視することが可能な Ping 型の通信によって監視対象ノードのモニタリングを行う．ACCMOS の実装は，基本的に Java で行ったが，Ping 型の通信に関しては C 言語で ICMP パケットの送受信を実装し，JNI を用いて呼び出すようにした．

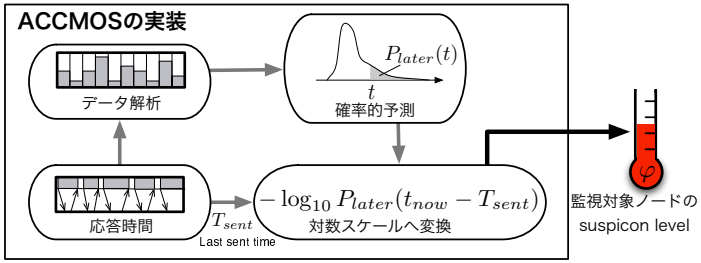


図 1 ACCMOS の実装

3.2 データの蓄積と suspicion level φ の計算

ACCMOS の実装の概略を図 1 に示す．

まず，ICMP Echo/Reply メッセージによってノード q, p 間のラウンドトリップ時間を得る．ACCMOS は，φ_p を計算するために，ラウンドトリップ時間を蓄積する．蓄積するデータ sum には次のように重み付けされる．

$$sum = sum \times \omega + RTT_{new} \times (1 - \omega) \quad (1)$$

ω $(0 < \omega < 1)$ は既に蓄積されたラウンドトリップ時間の重みを示し，新たに得たラウンドトリップ時間 RTT_{new} の重みは $1-\omega$ となる．ICMP Echo パケットを最後に送信した時刻を T_{sent}，現在の時刻を t_{now} としたとき，t_{now} における p が正常である確率[⋆1]は $P_{later}(t_{now} - T_{sent})$ となる (図 1 参照)．時刻 T_{sent} に q から p へ送信した ICMP Echo パケットに対応する Reply パケットが，q において既に受信されている場合，$P_{later} = 1$ となる．この値は，蓄積されたラウンドトリップ時間から確率分布関数を用いることによって計算される．ある ICMP Echo/Reply パケットによって計測されるラウンドトリップ時間は他の ICMP パケットと独立でかつ ICMP パケットはランダムに遅延すると仮定する．そのため確率分布関数として指数分布関数を用いて実装を行った．

suspicion level φ_p は監視対象ノード p が故障している度合いを示す．従って，P_{later} が減少するにつれて φ_p は増加しなければならない．ただ

⋆1 つまり，q において p からの ICMP Reply パケットが t_{now} 以降に到着する確率

Vol.2009-DPS-141 No.28
Vol.2009-GN-73 No.28
Vol.2009-EIP-46 No.28
2009/11/27

し，ネットワーク越しに監視している故障検出器は p の故障を 100%検出することはできないので，P_{later} は 0 に限りなく近づくが，0 にはならない．φ_p はこの様な性質を満たすために，以下に示すとおり，P_{later} を対数スケールに変換して算出される．

$$\varphi_p(t_{now}) \stackrel{\text{def}}{=} -\log_{10} P_{later}(t_{now} - T_{sent}) \quad (2)$$

監視対象ノード p が故障している場合，ノード q から送信された ICMP Echo パケットに対応する ICMP Reply パケットは戻ってこないので，φ_p は時間の経過と共に ∞ に近づく．一方，p が正常であれば，q において p からの ICMP Reply パケットが受信されるため，$\varphi_p = 0$ となる．

3.2.1 監視対象ノードの故障判定

ACCMOS は監視対象ノード p の故障判定をユーザやアプリケーション側で個別に行うことによって，ユーザの要求に適応した故障検出サービスの提供とスケーラビリティを両立させている．監視対象ノード p の suspicion level φ_p は，ACCMOS に問い合わせを行うことで得ることができる．ユーザやアプリケーションはそれぞれの要求を反映したしきい値 Φ_p を持ち，$\varphi_p > \Phi_p$ となったときに故障と判定する．しきい値 Φ_p は，監視対象ノード p の故障している度合いがどのくらい高まれば故障と判定するかを示す．例えば，$\Phi_p = 1.0$ ならば，故障している確率が 90% 以上 ($P_{later} < 0.1$) になれば故障と判定することを意味する．Φ_p の値を上げれば，故障検出に時間はかかるが，精度の高い故障検出を行うことができる．一方，Φ_p の値を下げれば，精度は下がるが，故障検出にかかる時間は短くなる．

Φ_p はそれぞれのユーザやアプリケーションが独自に設定するため，ACCMOS は φ_p を提供するだけで，個々の要求した故障検出精度に応じた故障検出サービスを同時に提供することができる．

3.2.2 ACCMOS の利点

ACCMOS は監視対象ノード p の情報を φ_p という連続的な値として提供するため，この値を温度計のように表示することで監視対象ノードの状態を分かりやすくすることができる (図 1 参照)．また，ACCMOS は故障判定のために要求する故障検出精度 Φ_p をパラメータとして与えるため，トラフィックの傾向が変化しても与えられた精度で適切に監視し，ユーザによる設定変更を必要としない．

環境	RTT 平均値 (ms)	RTT 中央値 (ms)	RTT 標準偏差	ロス率 (%)
Env. 1	20.56	10.00	163.79	0.41
Env. 2	46.31	43.00	86.30	0.09
Env. 3	31.02	30.00	62.46	0.05

表 1 各実験環境におけるラウンドトリップ時間 (RTT) の概要とパケットロス率

4. 評価実験

4.1 故障検出器の評価指標

評価基準は Chen らによって提案された故障検出器の QoS 指標を用いる [8]．本実験では以下の評価指標を用いる．

Definition1 (平均誤検出レート (Mistake Rate) λ_M) 故障検出器が単位時間あたりに故障していない監視対象ノードを誤検出する平均回数

Definition2 (正検出確率 (Query Accuracy Probability) P_A) 故障検出器が監視ノード p の状態に関する正しい情報を提供する確率

Definition3 (平均検出時間 (Average Detection Time) T_D) 監視ノード p が故障していた場合，故障検出器が p に対して故障判定を行うまでの平均時間

これらの指標によって ACCMOS における故障判定の偽陽性 (false positive) を評価する．故障判定の偽陽性は分散合意アルゴリズムの性能低下を招くなどの弊害があり [9]，λ_M や P_A は故障検出器の性能評価指標として広く用いられている．

4.2 実験環境と ACCMOS の設定

実験は異なる 3 つの環境で行った．実験を行った環境におけるラウンドトリップ時間の分布やパケットロス率 (ロス率) は表 4.2 および図 2 の通りである．

監視対象ノード p と ACCMOS が動作しているノード q は共に常に正常に動作している．実験の結果としては，q において動作している ACCMOS の平均誤検出レート λ_M と正検出確率 P_A を得ることができる．

ACCMOS の設定として，重みを $\omega = 0.8$ とした．ユーザが与える故障判定のしきい値は，$\Phi_p \in \{0.7, 1.0, 2.0, 3.0\}$ (それぞれ，$P_{later} = \{0.2, 0.1, 0.01, 0.001\}$ に対応) として，各設定に対して 2.5 時間実験を行っ

Vol.2009-DPS-141 No.28
Vol.2009-GN-73 No.28
Vol.2009-EIP-46 No.28
2009/11/27

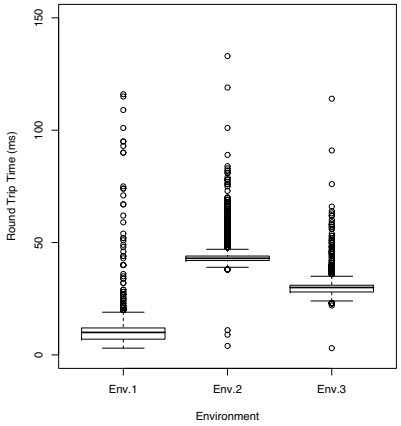


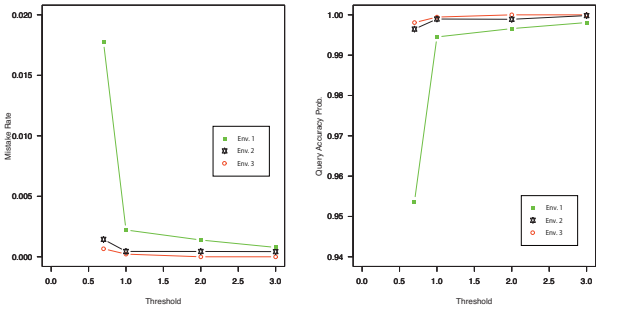
図 2 各実験環境におけるラウンドトリップ時間 (RTT) の分布

た．このとき設定したしきい値 Φ_p に対応する平均検出時間 (T_D) の変化を計測した．

4.3 実験結果

前述のネットワーク環境において実験を行い，実験結果として図 3 が得られた．平均誤検出レート λ_M は，ACCMOS が 1 秒間に起こす誤検出の平均回数を示している．Env. 1 と Env. 2 に関しては，ラウンドトリップ時間 (RTT) のばらつきがそれほど差がない (図 4.2 参照) ため，図 3 の λ_M および P_A はほぼ同様のかなり良好な結果が得られた．特に，Env. 3 の $\Phi_p \in 2.0, 3.0$ に関しては，パケットロスがなく，$P_A = 100(\%)$ となった．つまり，RTT がこの程度のばらつきであれば，パケットロスがない環境で，誤検出なく監視することができる．それ以外の設定においても，要求する故障検出精度とほぼ同等か大幅に上回る精度を得られることが P_A の値によって示されている．

RTT の標準偏差が Env. 2 より約 2 倍大きい Env. 1 においても，



指標	ω	$\Phi_p = 0.7$	$\Phi_p = 1.0$	$\Phi_p = 2.0$	$\Phi_p = 3.0$
λ_M	Env. 1	0.01773	0.00222	0.00139	0.00078
	Env. 2	0.00144	0.00044	0.00044	0.00043
	Env. 3	0.00067	0.00022	0.00000	0.00000
P_A (%)	Env. 1	95.37	99.45	99.65	99.80
	Env. 2	99.64	99.89	99.88	99.98
	Env. 3	99.80	99.94	100.00	100.00

図 3 実験結果: 平均誤検出レート λ_M と正検出確率 P_A

$\Phi_p = 0.7$ (80%) のとき $P_A = 95.37(\%)$，$\Phi_p = 1.0$ (90%) のとき $P_A = 99.45(\%)$，$\Phi_p = 2.0$ (99%) のとき $P_A = 99.65(\%)$，$\Phi_p = 3.0$ (99.9%) のとき $P_A = 99.80(\%)$，という結果が得られており，$\Phi_p = 3.0$ のときは P_A が若干要求した故障検出精度を下回ったが，それ以外では P_A が要求した故障検出精度を上回っている．

図 4 は，各実験環境における平均検出時間 T_D と正検出確率 P_A を示している．これによると，$0.7 \leq \Phi_p \leq 1.0$ では，T_D の違いが比較的小さいが，図 3 において P_A や λ_M の違いは大きい．一方で，$1.0 \leq \Phi_p \leq 3.0$ における T_D は，Φ_p の増加に比例して大幅に増加しているが，P_A や λ_M はそれほど変化しないことがわかる．つまり，今回の実験環境では，$\Phi_p = 1.0$ でも十分正確な故障検出ができることを示しており，$\Phi_p \geq 2.0$ を設定しても故障検出精度はそれほど向上しないにも関わらず，故障検出にかかる時間だけが増加するということが言える．

Vol.2009-DPS-141 No.28
Vol.2009-GN-73 No.28
Vol.2009-EIP-46 No.28
2009/11/27

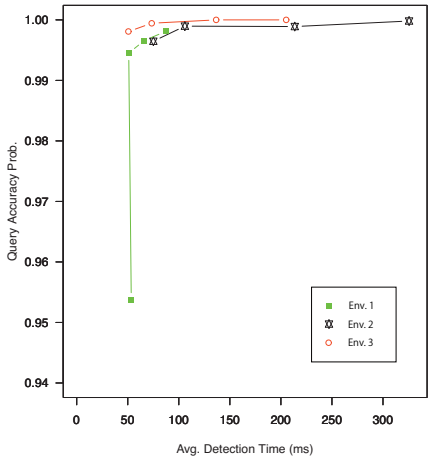


図 4 平均検出時間 (Avg. Detection Time) T_D と正検出確率 (Query Accuracy Prob.) P_A

5. まとめと展望

本稿では，3 つの異なる環境における，アクルーアル型故障検出器 ACCMOS の性能評価を行った．

ACCMOS において，監視対象ノード p の故障検出を行うためには，要求する故障検出精度をしきい値 Φ_p という値で設定する．今回の実験では，それぞれのネットワーク環境における Φ_p と故障判定の疑陽性 (P_A, λ_M) の関係に加えて，平均検出時間 (T_D) がどのように変化するかを測定した．これによって，Φ_p を高い値 (特に，$\Phi_p \geq 2.0$) にしても，故障検出にかかる時間が増大する一方，P_A や λ_M はそれほど向上しないことが実験結果から分かった．

今後，様々なパターンのネットワークトラフィックにおいてどのような結果が得られるかを蓄積することによって，ACCMOS における Φ_p の設定に一定の指標を与えることができると考えている．また，パケットロスや定常状態から著しく外れる不安定なネットワーク状態に対処する機構などについても検討する必要がある．また，ACCMOS を用いて実現するモニタリングシステムには，suspicion level を温度計のように表示することで直感的に監視対象の状況を把握できる GUI を実装し，ユーザビリティの高いモニタリングシステムの実現を目指す．

謝辞 本研究は科研費 (No. 20700072)，及び京都産業大学 総合研究支援制度 (No. 530) の助成を受けたものである．

参 考 文 献

1) “Facebook.” http://www.facebook.com.
2) “Our grid.” http://www.ourgrid.org.
3) “Appia.” http://appia.di.fc.ul.pt.
4) N.Hayashibara, X.Défago, R.Yared, and T.Katayama, “The φ accrual failure detector,” in *Proc. 23nd IEEE Int’l Symp. on Reliable Distributed Systems (SRDS’04)*, (Florianópolis, Brazil), pp.66–78, October 2004.
5) X.Défago, P.Urbán, N.Hayashibara, and T.Katayama, “Definition and specification of accrual failure detectors,” in *Proc. Int’l Conf. on Dependable Systems and Networks (DSN’05)*, (Yokohama, Japan), pp.206–215, June 2005.
6) “Cassandra.” http://wiki.apache.org/cassandra/.
7) R.Lima, W.Cirne, F.Brasileiro, D.Fireman, and L.D.S. Distribudos, “A case for event-driven distributed objects,” in *In Proc.of DOA’06, LNCS 4276*, pp.1705–1721, Springer-Verlag, 2006.
8) W.Chen, S.Toueg, and M.K. Aguilera, “On the quality of service of failure detectors,” *IEEE Transactions on Computers*, vol.51, pp.561–580, May 2002.
9) L. M. R. Sampaio, F. V. Brasileiro, W. Cirne, and J. C. A. Figueiredo, “How bad are wrong suspicions? towards adaptive distributed protocols,” in *In Proc. IEEE Intl. Conf. on Dependable Systems and Networks (DSN’03)*, pp.551–560, 2003.
10) N.Hayashibara and M.Takizawa, “Performance evaluation of the φ failure detector,” in *In Proc. of the 8th IEEE ICDCS Workshops (MNSA’06)*, July 2006.