Fault Tolerance Mobile Agent Execution System (FTMAS) Modeling

Amal Moh'd Al. Dweik
College of Information Technology and Computer Engineering
Palestine Polytechnic University

Prof. Imane Aly Saroit Ismail
Information Technology Department
Faculty of Computers & Information
Cairo University

Prof. Sanaa. Hanafi Ahmed
Information Technology Department
Faculty of Computers & Information
Cairo University

ABSTRACT

Mobile Agent "MA" paradigm is said to give better performance advantages over other paradigms due to its features especially in the network and the internet applications. To integrate mobile agents into such applications, fault tolerant ability of an agent is one of the most important issues. In this paper we propose the Fault Tolerant Mobile Agent execution System "FTMAS" mathematical model. The paper states and defines the migration time, the Round trip time, the transfer time and the system throughput. They are defined relative to 3 performance metrics: the average connectivity of the network, the agent size and the probability of failure. The mathematical model of the FTMAS is stated as a promising model in the fault tolerant field of the Mobile Agent execution. In addition, a novel failure classification is stated to solve agent faults efficiently.

***KEY WORDS*:**
Mobile Agents, fault tolerance, Replication, migration time, round trip time, transfer time, system throughput.

1. INTRODUCTION

Modern applications can be no longer satisfied by the use of the traditional distributed computing paradigms. These paradigms are no longer adequate for the networking and information system applications. MAs are a type of software agent that migrates from one place to another with its data, state and code. In contrast to client-server paradigm, the mobile agent paradigm views computer-to-computer communication in both directions. During the agent itinerary, the agent may visit several places according to the tasks assigned for each agent. Mobile agent can suffer from different breakpoints during its execution. while failures should not prevent the mobile agent from continuing its itinerary to achieve its goals.

While a mobile agent is executing on a host h_i, a failure of h_i might interrupt the execution of the agent a_i and prevent any progress of this mobile agent execution. During the time h_i is down, the execution of a_i and consequently the entire mobile agent execution cannot proceed. Therefore, agent a_i may suffer from a problem causing its execution to terminate abnormally. When a_i crashed, then all the tasks done and all the results obtained by a_i lost. The Fault Tolerant Mobile Agent execution System ***FTMAS*** motivates the work of the MA reliability. The execution proceeds after the crash of the agent or crash of the host smoothly and correctly by the use of replication. The worker agent leaves some replica (s) in its itinerary while moving from host to another. The number of replica(s) and status is different depending on the approach used. This is performed by proposing 3 fault tolerant approaches: the Centralized, the Windowing of size-n and the Centralized Windowing approaches.

The remainder of this paper is organized as follows: the next section introduces the fault tolerant in MAs. Section 3 introduces the previous work while section 4 defines the FTMAS model. Section 5 introduces the suggested FTMAS mathematical model. The conclusion is introduced in section 6. Finally, an appendix that contains the most important terminologies used in the paper is presented in section 7 followed by the references.

2. FAULT TOLERANT IN MOBILE AGENTS

When fault does occur in a mobile agent system (MAS), interactions between agents may cause the fault to spread through the system randomly. These failures should not prevent the mobile agent from continuing its itinerary to achieve its goals as shown in figure 1.

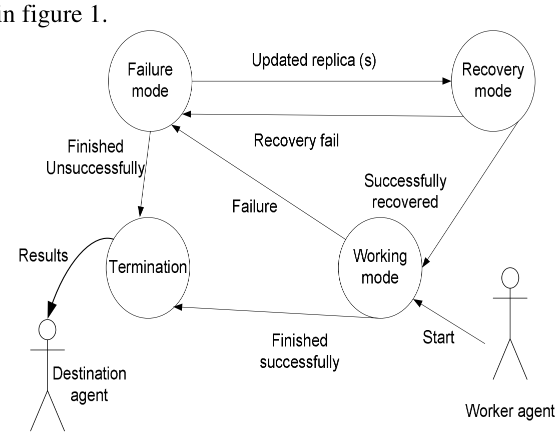

Figure 1. The Sate Diagram of the Agent Failure

2.1 Fault Problem

There are several types of faults that can occur in agent environments. The most important are: the infrastructure failures and the semantic failures. The work concentrates on the infrastructure failure.

Infrastructure Failures

Machines, places or agents can fail and recover later [1]. When infrastructure failures are detected, then it is generally encapsulated into a failure detector module. This module is defined in terms of completeness and accuracy properties [1]. Completeness is satisfied when any failed process is eventually suspected. However, accuracy relates to the number of the false suspicions (i.e., processes that are suspected but have not crashed). The objective here is to limit these numbers to the minimum.

3. PREVIOUS WORKS

For the past few years, researchers and developers have put considerable efforts in the development of the basic technology for mobile agents. Most of these efforts are subject to the reliability of the MA – based applications. Hence, there are a lot of researchers' efforts directed to work on fault tolerant approaches of MAs [2, 3,4, 5, 6, 7, and 8]. Two main works are highly related to our work: the work of Stefan Pleisch and the work of Wenyn Qu. Pleisch built FATOMAS as a fault tolerant MA execution prototype [1, 9, and 10]. FATOMAS sends a set of replicas to a set of places in the next stage to prevent blocking of the agent execution. However, this produces the problem of multiple executions of the agent and so necessitates a solution to overcome this problem. Pleisch measures the performance of the FATOMAS on a single MA execution by measuring the effect of the agent size and the size of the replicas. However, Qu in [11, 12, and 13] works on the performance analysis for the fault tolerant MA execution. It assumes that online communication is maintained between the previous node and the current node hosting the agent. However, this contradicts the offline processing feature of the MAs. In [11, 12, and 13] when the agent completes its execution at a node, then all its surveillances are killed. The problem arises when that agent is crashed before reaching to the final destination. Qu measures the performance using the migration time, life expectancy and population distribution of MAs for only one approach with general failure scenario. These factors are measured against the average network connectivity and the probability of failure.

4. FTMAS Model

The paper proposes a new execution model, FTMAS, which uses three proposed approaches to tolerate faults. The approaches are: the Centralized approach, the Windowing approach, and the Centralized Windowing approach. In addition, the FTMAS introduces a novel classification of failures that the agent may suffer from during and after execution.. Theses classifications are applied in the proposed approaches in the FTMAS execution model to recover failures in all cases efficiently.

4.1 FTMAS Execution Model

A mobile agent executes on a sequence of machines/hosts, where a host h ($1 < h < n$) provides the logical execution environment for the agent in its life cycle. The main host, Main Container which runs a set of services represents the bootstrap point of a platform. Executing the agent at a host h_i is called a stage s_i of the agent execution. The sequence of nodes visited between the agent source and destination (i.e., $\mathbf{h_0, h_1\ h_2 \ldots h_n}$) composes the agent itinerary.

The agents in the proposed system use dynamic itinerary, where there is no limit on the number of hosts to be visited by any agent. During the agent itinerary, it may visit several places according to the tasks assigned for each agent. For this work, the agent itinerary is not fully dynamic. The agent dynamically figures out the hosts before starting its itinerary. However, once it is started, then it will use static set of hosts to be visited. The visited hosts could be either remote hosts or local hosts to the agent platform. Once the agent completes its execution on a host h_i, the agent commits its execution. This commitment is accomplished by performing the replication operation according to the used approach. The agent replicas are not executing while the original executing agent is active. Therefore, only one execution of the agent will be guaranteed at the same time. This property ensures the exactly once execution of the agents. The non-blocking feature is also guaranteed even in the case of multiple failures by allowing the replica(s) of the crashed agent to replace it in order to continue execution even in case of agent failures.

4.2 Variations of the Failure Problem

The failure of the agent could happen after the updating of the agent replica(s), which is called optimistic failure. In this case, the agent commits its execution before crash. However, the failure of the agent could happen also before updating its replica(s), i.e., having uncommitted crashed agent, which in turn called pessimistic. In both cases the system should be able to tolerate faults successfully. However, it becomes a more complex and troublesome task to handle the pessimistic failures.

The paper suggests five novel failure cases for the agent a_i while it is executing at host h_i:

Case 1: [Safe Case]: *The agent successfully completes execution on host h_i and moves safely to the next host, h_{i+1}.*

Case 2: [Post-Failure - Optimistic]: *The agent crashes or fails after committing its execution at host h_i and before migrating to host h_{i+1}.*

Case 3: [Post-Failure - Pessimistic]: *The agent crashes or fails after completing execution at host h_i but before committing its execution.*

Case 4: [Pre-Failure]: *The agent crashes or fails before completing execution at host h_i.*

Case 5: [Host Crash]: *The host h_i crashes. This case is treated as having the agent being crashed. This is because the crash of the host means the crash of the* *operations, services and processes (agents) running on it.* Case 5 could be combined with case 2, case 3 or case 4.

5. FTMAS Proposed Approaches

FTMAS model suggests three main proposed approaches: Centralized Approach “CA”, Windowing Approach “WA” and Centralized – Windowing Approach “CWA”.

5. 1 Centralized Approach

Any agent created in any container, host, will be replicated in the main container. When the agent is migrated from any host to another, it commits its execution state by reflecting the changes to the replica in the main container. After crash, the replica is used to replace the crashed agent or crashed host with the most updated state of the crashed one.

5 .2 Windowing approach

This approach uses a snapshot window of the agent itinerary in order to recover that agent in case of failures. There are mainly two proposed techniques that could be applied. The first is to use the size -1 window snapshot and the second is to use size – n window.

A. Size-1 Approach

When any agent is committed after executing at a host, then a replica of this agent is created in the previously visited host. After the agent is migrating to another host, then the hosting container will be the previous and so on. In this approach, a window of agent replica following the agent in its migration with size 1 is formed.

B. Size-n Approach

This approach is based on the same principle as that of the size-1 approach. However, the window of size-**n** approach uses a variable or a fixed size of $\boldsymbol{n}$. In this approach, after the agent commits its execution, then a copy of this agent is created at the host $\boldsymbol{h_i}$ before migrating to host $\boldsymbol{h_{i+1}}$. Every time the agent is migrated, a copy of that agent is created in the $\boldsymbol{n}$ previously visited hosts with a number of $\boldsymbol{n}$ replica(s). In case of agent crash, the agent's replica in the $(\mathbf{n-1})^{\mathbf{th}}$ previously visited host in case that it is not crashed, is substituted the original agent by replicating it in the current host. Otherwise, i.e., if this $(\mathbf{n-1})^{\mathbf{th}}$ replica is crashed, the next previously agent's replica is used and so on.

There are two options to perform the update of the replicas: full or partial update. For the full update, all the changes are reflected to all replicas once the agent committed its execution. However, for the partial update, only the replica in the last visited host would be the fully updated replica. Other replicas are partially updated at the time of their creation.

5.3 Centralized-Windowing approach

This methodology was an integration of the two previously mentioned approaches: the centralized and the windowing approaches. It is based on using a sliding window of size *k*, and a centralized replica in the main container at the same time. As long as the main container's replica is available so use it. Otherwise, switch to the sliding window replicas. This tries to gain the advantages of the two approaches in order to increase the reliability and the efficiency to tolerate faults.

6. PERFORMANCE CALCULATION

Suppose that $LH(i)$ is the set of the neighboring hosts of the node h_i. After the agent finishes executing in node h_i, it travels to the first selected node, h_i, in the $LH(i)$ and executes there. This process will proceed repeatedly until the agent completes its execution successfully on all nodes in its itinerary.

6.1 FTMAS Performance Parameters

Four performance factors calculated: the Migration time MT, the Round trip time RTT, the transfer time TrT and the System throughput Th. The Migration time is defined as: “*How long it takes for the agent to migrate from a pair of nodes (h_i, h_{i+1}) and commits its execution at h_i”*. The typical definition used for the Round Trip Time is: *“How long it takes for the agent to migrate from a pair of nodes (h_i, h_{i+1}) and the source node, h_i, got the acknowledgment for the successful migration to destination node h_{i+1}”*. The transfer time is defined as: *“The time it takes to transmit or move data from one place to another. This time is dependent on the size of the data transfer and the rate at which it can be transmitted to/from the host”*. Finally, the system throughput is defined as: *“The average rate at which data is transferred through a system over a communication channel”*. All these performance factors are calculated against: the average network connectivity “*d*”, which is the number of nodes in the agent itinerary; the probability of failure “ ρ “; and the agent size.

6.2 Migration Time Calculations

The Migration Time "MT" between host h_i and the node h_{i+1} can be decomposed into travel time and execution time [11]. Suppose that the MT probability density function *(pdf)* is given by *l(t),* the travel time *pdfs* are given by *r(t)* and $\overline{r_i}(t)$ respectively, and the execution time *pdfs* is given by *e(t)* [11]. When the probability distribution of travel time and execution time are known, then the probability distributions of MT is calculable [12]. Thus the MT for moving the agent from h_i to h_{i+1}, denoted by l_i, can be expressed as $l_i = \sum_{j=1}^{v_i} l_j$ where v_i denotes the number of nodes that are selected by the agent in *LH (i).*

6.2.1 Windowing Approach of Size n

1. Safe Case (Case 1)

It states that no crash occurs either for the running agent or the hosting node. The agent will not die until it finds its destination. This means that the average MT of an agent is:

MA= r + e; (1)

2. Post-Failure – Optimistic (Case 2)

In case that the explicit distribution of *e, r* and $\overline{r}$ is known, the Laplace transform distribution of the MT, denoted by F*(s) can be expressed as:

$$F^*(s) = \sum_{l=1}^{d_i} \left[R^*(s).\overline{R}^*(s)\right]^{l-1} \left[R^*(s).E^*(s)\right] P(v_i = l) \quad (2)$$

The probability is computed by the inspiration of the work done in [12]. Where $P(v_i = l) = (1-\rho)\rho^{l-1}$ for $1 \le l \le d_i$ which is geometric distribution. Thus, probability distribution of MT can be easily gained by averaging Laplace transform of equation 3.

$$F^*(s) = \frac{(1-\rho)R^*(s).E^*(s)}{1-[\rho.R^*(s).\overline{R}^*(s)]} \cdot \left[1-(\rho.R^*(s).\overline{R}^*(s))^{d_i+1}\right] \quad (3)$$

In case that the explicit distribution of execution time and travel time is unknown, we still can estimate the mean of MT if the means of execution time and travel time are available. Since l_i is independent to v_i for $1 \le l \le v_i$ [3]. Due to the fact that $l_i = \sum_{j=1}^{v_i} l_i = \sum_{j=1}^{v_i} r^j{}_i + \sum_{j=1}^{v_{i-1}} \overline{r}^j{}_i + e_i^{v_i}$, then, the average MT of an agent satisfies:

$$E(l_i) = (r + \overline{r})\left(\frac{1-\rho^d}{1-\rho} - d\rho^d\right) - \overline{r} + e \quad (4)$$

3. Post-Failure – Pessimistic (Case 3)

Let the event $\{X_1^0 = 1\}$ indicates that the agent on the j[th] selected node fails before the agent commits its execution. Then the event $\{v_i=l\}$ gives the probability $P(v_i = l)$ equals $P\{v_i = j\}$ which in turn gives:

$$P(v_i = l) = [\rho(1-\rho)]^{j-1}(1-\rho) \quad [11](5)$$

Thus, if the explicit distribution of e, r and $\overline{r}$ is known, the Laplace transform distribution of MT is:

$$F^*(s) = \left(\begin{array}{c} \frac{(1-\rho)R^*(s)E^*(s).}{1-\rho(1-\rho)R^*(s)E^*(s)\overline{R}^*(s).} \cdot \\ \left[1-\left(\rho(1-\rho)R^*(s)E^*(s)\overline{R}^*(s)\right)^{d_i+1}\right] \end{array}\right) \quad (6)$$

If the explicit distribution of e, r and $\overline{r}$ are unknown, and if we use x= $\rho(1-\rho)$, then average MT can be expressed as:

$$E(l_i) = (r+\overline{r}+e)\left(\frac{1-\rho}{(1-x)^2}[1-X^d-(1-X)d.X^d]\right) - \overline{r} \quad (7)$$

4. Pre-Failure (Case 4)

The same sequence is done to recover the crashed agent as that done in post-failure – pessimistic case.

5. Host Crash (Case 5)

We have two scenarios: either to wait for the host recovery or not waiting for its recovery.

Ignoring the recovery of the host

In optimistic case and if the explicit distribution of execution time and travel time is known, the Laplace transform distribution of the MT can be as follows:

$$F^*(s) = \frac{(1-\rho)R^*(s).E^*(s)}{1-\rho.R^*(s)} \cdot \left[1-(\rho.R^*(s))^{d_i+1}\right] \quad (8)$$

However, in pessimistic case, we have:

$$F^*(s) = \frac{(1-\rho)R^*(s).E^*(s)}{1-\rho.(1-\rho)R^*(s)} \cdot \left[1-(\rho.(1-\rho)R^*(s))^{d_i+1}\right] \quad (9)$$

If explicit distribution of *e* and *r* is unknown, then in optimistic case, the average MT satisfies:

$$E(l_i) = r.\left(\frac{1-\rho^d}{1-\rho} - d\rho^d\right) + e \quad (10)$$

However, in pessimistic case, the average MT is::

$$E(l_i) = r.\left(\frac{1-\rho}{(1-x)^2}[1-X^d-(1-X)d.X^d]\right) + e \quad (11)$$

Waiting for the recovery of the host

There is no need for discussing the optimistic case here because it doesn't require any re-execution after the recovery of the host. The Laplace transforms distribution of the MT in pessimistic cases is:

$$F^*(s) = \left(\begin{array}{c} \frac{(1-\rho)R^*(s).E^*(s)}{1-[\rho(1-\rho)R^*(s)E^*(s)Cr^*(s)]} \\ \cdot\left[1-([\rho(1-\rho)R^*(s)E^*(s)Cr^*(s)])^{d_i+1}\right] \end{array}\right) \quad (12)$$

If explicit distribution is unknown, the average MT:

$$E(l_i) = (r+e+cr)\left(\frac{1-\rho}{(1-x)^2}[1-X^d-(1-X)d.X^d]\right) - cr \quad (13)$$

6.2.2 Windowing Approach of Size 1

The same calculation of the MT that is done for the windowing of size n approach is applied here. However, this is done with value of d=1. All these calculations are created, however, to summarize, we will not include these calculations in this paper.

6.2.3 Centralized Approach

The Centralized approach has the same calculations of that of the windowing of size 1 since we have one replica in the main container.

6.2.4 Centralized -Windowing Approach

Ck_i is used as a check time *pdf* to figure out whether to use the centralized approach or the windowing approach. Ck_i returns a value of υ which is either equal to 1 or *0*. It is used to test the availability of the main container's replica. If the check returns value of 1, then the centralized approach is applied. Otherwise, the windowing approach is performed.

1. Post-Failure – Optimistic

The Laplace transform distribution of the MT when the explicit distribution of e and r is known satisfies:

$$F^*(s)=\begin{pmatrix}(1-\rho)R^*(s)E^*(s)\\ \begin{pmatrix}\upsilon\dfrac{1-(R^*(s)CK^*(s)\overline{R}^*(s))^{d+1}}{1-(R^*(s)CK^*(s)\overline{R}^*(s))}+\\ (1-\upsilon)\dfrac{1-(\rho R^*(s)CK^*(s)\overline{R}^*(s))^{d+1}}{1-\rho R^*(s)CK^*(s)\overline{R}^*(s)}\end{pmatrix}\end{pmatrix} \quad (14)$$

Otherwise, the average MT satisfies:

$$E(l_i)=\begin{bmatrix}(r+\overline{r}+ck)\left(\upsilon(1-\rho)+(1-\upsilon)\left(\dfrac{1-\rho^d}{1-\rho}-d\rho^d\right)\right)\\ -\overline{r}-ck+e\end{bmatrix} \quad (15)$$

2. Post-Failure – Pessimistic / Pre-Failure

The Laplace transform distribution of MT if explicit distribution of e and r is known satisfies:

$$F^*(s)=\begin{pmatrix}(1-\rho)R^*(s)E^*(s)\\ \begin{pmatrix}\upsilon\dfrac{1-(R^*(s)E^*(s)CK^*(s)\overline{R}^*(s))^{d+1}}{1-(E^*(s)R^*(s)CK^*(s)\overline{R}^*(s))}+(1-\upsilon)\\ \dfrac{1-(\rho(1-\rho)R^*(s)E^*(s)CK^*(s)\overline{R}^*(s))^{d+1}}{1-\rho(1-\rho)E^*(s)R^*(s)CK^*(s)\overline{R}^*(s)}\end{pmatrix}\end{pmatrix} \quad (16)$$

Otherwise, the average MT satisfies:

$$E(l_i)=\begin{bmatrix}(r+\overline{r}+e+ck)\\ \begin{pmatrix}(1-\upsilon)\dfrac{1-\rho}{(1-x)^2}[1-X^d-(1-X)d.X^d]+\\ \upsilon\dfrac{1-\rho}{(1-x)^2}[1-2X+X^2]\end{pmatrix}\end{bmatrix}-\overline{r}-ck \quad (17)$$

3. Host Crash

When ignoring the recovery of the host

For optimistic case and in case that explicit distribution of e and r is known, the Laplace transform distribution of MT is:

$$F^*(s)=\begin{pmatrix}(1-\rho)R^*(s)E^*(s)\begin{pmatrix}\upsilon\dfrac{1-(R^*(s)CK^*(s))^{d+1}}{1-(R^*(s)CK^*(s))}+\\ (1-\upsilon)\dfrac{1-(\rho R^*(s)CK^*(s))^{d+1}}{1-\rho R^*(s)CK^*(s)}\end{pmatrix}\end{pmatrix} \quad (18)$$

However, in pessimistic case it is:

$$F^*(s)=\begin{pmatrix}(1-\rho)R^*(s)E^*(s)\\ \begin{pmatrix}\upsilon\dfrac{1-(R^*(s)E^*(s)CK^*(s))^{d+1}}{1-(E^*(s)R^*(s)CK^*(s))}+(1-\upsilon)\\ \dfrac{1-(\rho(1-\rho)R^*(s)E^*(s)CK^*(s))^{d+1}}{1-\rho(1-\rho)E^*(s)R^*(s)CK^*(s))}\end{pmatrix}\end{pmatrix} \quad (19)$$

If the explicit distribution is unknown, the average MT in optimistic case satisfies:

$$E(l_i)=(r+ck)\left(\upsilon(1-\rho)+(1-\upsilon)\left(\frac{1-\rho^d}{1-\rho}-d\rho^d\right)\right)-ck+e \quad (20)$$

But, the average MT in pessimistic case satisfies:

$$E(l_i)=\begin{bmatrix}(r+e+ck)\\ \begin{pmatrix}(1-\upsilon)\dfrac{1-\rho}{(1-x)^2}[1-X^d-(1-X)d.X^d]+\\ \upsilon\dfrac{1-\rho}{(1-x)^2}[1-2X+X^2]\end{pmatrix}\end{bmatrix}-ck \quad (21)$$

Waiting for the recovery of the host

The Laplace transform distribution of the migration time can be expressed as:

$$F^*(s)=\begin{pmatrix}(1-\rho)R^*(s)E^*(s)\\ \begin{pmatrix}\upsilon\dfrac{1-(R^*(s)E^*(s)CK^*(s)Cr^*(s))^{d+1}}{1-(E^*(s)R^*(s)CK^*(s)Cr^*(s))}+\\ (1-\upsilon)\dfrac{1-(\rho(1-\rho)R^*(s)E^*(s)CK^*(s)Cr^*(s))^{d+1}}{1-\rho(1-\rho)E^*(s)R^*(s)CK^*(s)Cr^*(s))}\end{pmatrix}\end{pmatrix} \quad (22)$$

Otherwise, the average MT satisfies:

$$E(l_i)=\begin{bmatrix}(r+e+cr+ck)\\ \begin{pmatrix}(1-\upsilon)\dfrac{1-\rho}{(1-x)^2}[1-X^d-(1-X)d.X^d]+\\ \upsilon\dfrac{1-\rho}{(1-x)^2}[1-2X+X^2]\end{pmatrix}-ck-cr\end{bmatrix} \quad (23)$$

6.3 Round Trip Time Calculations

6.3.1 Windowing Approach of Size n

1. Safe Case

For the safe case, the average RTT of the agent in its itinerary to accomplish its task successfully can be given by: $RTT = r$ (24)

2. Post – Failure / Pre - Failure Cases

Optimistic case:

The Laplace transform distribution of the round trip time satisfies:

$$F^*(s)=\frac{(1-\rho)R^*(s).E^*(s)}{1-[\rho.R^*(s).\overline{R}^*(s)]}.\left[1-(\rho.R^*(s).\overline{R}^*(s))^{d_i+1}\right] \quad (25)$$

Otherwise, the average RTT satisfies:

$$E(l_i)=(r+\overline{r})\left(\frac{1-\rho^d}{1-\rho}-d\rho^d\right)-\overline{r} \quad (26)$$

Pessimistic Case:

The Laplace transform distribution of the RTT satisfies:

$$F^*(s)=\begin{pmatrix}\dfrac{(1-\rho)R^*(s).E^*(s)}{1-[\rho(1-\rho)R^*(s).\overline{R}^*(s)]}.\\ \left[1-(\rho(1-\rho)R^*(s).\overline{R}^*(s))^{d_i+1}\right]\end{pmatrix} \quad (27)$$

In case that the explicit distribution of travel time is unknown, the average RTT satisfies:

$$E(l_i)=(r+\overline{r})\left(\frac{1-\rho}{(1-x)^2}[1-x^d-(1-x)dx^d]\right)-\overline{r} \quad (28)$$

3. Host Crash

Ignoring the recovery of the host

Optimistic Case:

The Laplace transforms distribution of RTT is:

$$F^*(s) = \frac{(1-\rho)R^*(s)}{1-R^*(s)}.\left[1-R^*(s)^{d_i+1}\right] \quad (29)$$

In case that the explicit distribution of travel time is unknown, the average RTT satisfies:

$$E(rtt_i) = r.\left(\frac{1-\rho^d}{1-\rho} - d\rho^d\right) \quad (30)$$

<u>**Pessimistic Case**</u>

However, for the pessimistic case, we have:

$$F^*(s) = \frac{(1-\rho)R^*(s)}{1-\rho(1-\rho)R^*(s)}.\left[1-\left(\rho(1-\rho)R^*(s)\right)^{d_i+1}\right] \quad (31)$$

In case that the explicit distribution of travel time is unknown, the average RTT satisfies:

$$E(rtt_i) = (r)\left(\frac{1-\rho}{(1-x)^2}[1-x^d-(1-x)dx^d]\right) \quad (32)$$

<u>**Waiting for the recovery of the host**</u>

For the pessimistic case, we have:

$$F^*(s) = \left(\begin{array}{l}\dfrac{(1-\rho)R^*(s)}{1-\rho(1-\rho)Cr^*(s)R^*(s)}. \\ \left[1-\left(\rho(1-\rho)Cr^*(s)R^*(s)\right)^{d_i+1}\right]\end{array}\right) \quad (33)$$

If explicit distribution of r is unknown, then:

$$E(l_i) = \left(\begin{array}{l}(r+Cr) \\ \left(\dfrac{1-\rho}{(1-x)^2}[1-x^d-(1-x)dx^d]\right) - Cr\end{array}\right) \quad (34)$$

6.3.2 Windowing Approach of Size 1

The same calculation of the RTT that is done for the windowing of size n is performed for the windowing of size 1 approach. However, this accomplished by using the value of d=1.

6.3.3 Centralized -Windowing Approach

1. Post-Failure – Pre-Failure Cases

<u>**Optimistic Case**</u>

The Laplace transform of the RTT satisfies:

$$F^*(s) = \left[\begin{array}{l}(1-\rho)R^*(s) \\ \left(\begin{array}{l}\upsilon\dfrac{1-(R^*(s)CK^*(s)\overline{R}^*(s))^{d+1}}{1-(R^*(s)CK^*(s)\overline{R}^*(s))} + \\ (1-\upsilon)\dfrac{1-(\rho R^*(s)CK^*(s)\overline{R}^*(s))^{d+1}}{1-\rho R^*(s)CK^*(s)\overline{R}^*(s)}\end{array}\right)\end{array}\right] \quad (35)$$

However, if the explicit distribution is unknown, the average RTT satisfies:

$$E(l_i) = \left[\begin{array}{l}(r+\overline{r}+ck) \\ \left(\upsilon(1-\rho)+(1-\upsilon)\left(\dfrac{1-\rho^d}{1-\rho}-d\rho^d\right)\right) - \overline{r} - ck\end{array}\right] \quad (36)$$

<u>**Pessimistic Case:**</u>

The Laplace transform of the RTT is:

$$F^*(s) = \left(\begin{array}{l}(1-\rho)R^*(s)E^*(s) \\ \left(\begin{array}{l}\upsilon\dfrac{1-(R^*(s)E^*(s)CK^*(s)\overline{R}^*(s))^{d+1}}{1-(E^*(s)R^*(s)CK^*(s)\overline{R}^*(s))} + \\ (1-\upsilon)\dfrac{1-(\rho(1-\rho)R^*(s)E^*(s)CK^*(s)\overline{R}^*(s))^{d+1}}{1-\rho(1-\rho)E^*(s)R^*(s)CK^*(s)\overline{R}^*(s)}\end{array}\right)\end{array}\right) \quad (37)$$

However, in case that the explicit distribution of travel time is unknown, the average RTT satisfies:

$$E(l_i) = \left(\begin{array}{l}(r+\overline{r}+ck) \\ \left(\begin{array}{l}(1-\upsilon)\dfrac{1-\rho}{(1-x)^2}[1-X^d-(1-X).d.X^d] + \\ \upsilon\dfrac{1-\rho}{(1-x)^2}[1-2X+X^2]\end{array}\right) - \overline{r} - ck\end{array}\right) \quad (38)$$

2. Host Crash

<u>**Ignoring the recovery of the host**</u>

The Laplace transform distribution of RTT can be expressed in optimistic case by:

$$F^*(s) = (1-\rho)R^*(s)\left(\begin{array}{l}\upsilon\dfrac{1-(R^*(s)CK^*(s))^{d+1}}{1-(R^*(s)CK^*(s))} + \\ (1-\upsilon)\dfrac{1-(\rho R^*(s)CK^*(s))^{d+1}}{1-\rho R^*(s)CK^*(s)}\end{array}\right) \quad (39)$$

In case that the explicit distribution of travel time is unknown, the average RTT satisfies:

$$E(l_i) = (r+ck)\left(\upsilon(1-\rho)+(1-\upsilon)\left(\frac{1-\rho^d}{1-\rho}-d\rho^d\right)\right) - ck \quad (40)$$

However, for the pessimistic case we have:

$$F^*(s) = \left(\begin{array}{l}(1-\rho)R^*(s) \\ \left(\begin{array}{l}\upsilon\dfrac{1-(R^*(s)CK^*(s))^{d+1}}{1-(R^*(s)CK^*(s))} + (1-\upsilon) \\ \dfrac{1-(\rho(1-\rho)R^*(s)CK^*(s))^{d+1}}{1-\rho(1-\rho)R^*(s)CK^*(s))}\end{array}\right)\end{array}\right) \quad (41)$$

In case that the explicit distribution of travel time is unknown, the average RTT satisfies:

$$E(l_i) = \left[(r+ck)\left(\begin{array}{l}(1-\upsilon)\dfrac{1-\rho}{(1-x)^2}[1-X^d-(1-X).d.X^d] + \\ \upsilon\dfrac{1-\rho}{(1-x)^2}[1-2X+X^2]\end{array}\right) - ck\right] \quad (42)$$

<u>**Waiting for the recovery of the host**</u>

The RTT when the explicit distribution of the travel time is known can be expressed as:

$$F^*(s) = \left[\begin{array}{l}(1-\rho)R^*(s) \\ \left(\begin{array}{l}\upsilon\dfrac{1-(R^*(s)CK^*(s)Cr^*(s))^{d+1}}{1-(R^*(s)CK^*(s)Cr^*(s))} + (1-\upsilon) \\ \dfrac{1-(\rho(1-\rho)R^*(s)CK^*(s)Cr^*(s))^{d+1}}{1-\rho(1-\rho)R^*(s)CK^*(s)Cr^*(s))}\end{array}\right)\end{array}\right] \quad (43)$$

Otherwise, the average RTT satisfies:

$$E(l_i) = \left(\begin{array}{l}(r+cr+ck) \\ \left(\begin{array}{l}(1-\upsilon)\dfrac{1-\rho}{(1-x)^2}[1-X^d-(1-X).d.X^d] + \\ \upsilon\dfrac{1-\rho}{(1-x)^2}[1-2X+X^2]\end{array}\right) \\ -ck-cr\end{array}\right) \quad (44)$$

6.4 Transfer Time Calculations

6.4.1 Windowing Approach of Size n

1. Safe Case

For the safe case, the average TrT of the agent in its itinerary to accomplish its task successfully can be given by: $Tr_i = r + C$ (45)

2. Post – Failure / Pre - Failure Cases

Optimistic case:

The Laplace transform distribution of the transfer time satisfies:

$$Tr^*(s) = \frac{(1-\rho)R^*(s)E^*(s)}{1-[\rho R^*(s)\overline{R}^*(s)]}.\left[1-(\rho R^*(s)\overline{R}^*(s))^{d_i+1}\right] + C \quad (46)$$

Otherwise, the average TrT satisfies:

$$E(l_i) = (r+\overline{r})\left(\frac{1-\rho^d}{1-\rho} - d\rho^d\right) - \overline{r} + C \quad (47)$$

Pessimistic Case:

The Laplace transform distribution of the TrT satisfies:

$$F^*(s) = \left(\begin{array}{l}\dfrac{(1-\rho)R^*(s).E^*(s)}{1-[\rho(1-\rho)R^*(s).\overline{R}^*(s)]}. \\ \left[1-(\rho(1-\rho)R^*(s).\overline{R}^*(s))^{d_i+1}\right]\end{array}\right) + C \quad (48)$$

In case that the explicit distribution of travel time is unknown, the average TrT satisfies:

$$E(l_i) = (r+\overline{r})\left(\frac{1-\rho}{(1-x)^2}[1-x^d-(1-x)dx^d]\right) - \overline{r} + C \quad (49)$$

3. Host Crash

Ignoring the recovery of the host

Optimistic Case:

The Laplace transforms distribution of TrT is:

$$F^*(s) = \frac{(1-\rho)R^*(s)}{1-R^*(s)}.\left[1-R^*(s)^{d_i+1}\right] + C \quad (50)$$

In case that the explicit distribution of travel time is unknown, the average TrT satisfies:

$$E(Tr_i) = r.\left(\frac{1-\rho^d}{1-\rho} - d\rho^d\right) + C \quad (51)$$

Pessimistic Case

However, for the pessimistic case, we have:

$$F^*(s) = \frac{(1-\rho)R^*(s)}{1-\rho(1-\rho)R^*(s)}\left[1-\left(\rho(1-\rho)R^*(s)\right)^{d_i+1}\right] + C \quad (52)$$

In case that the explicit distribution of travel time is unknown, the average TrT satisfies:

$$E(rtt_i) = (r)\left(\frac{1-\rho}{(1-x)^2}[1-x^d-(1-x)dx^d]\right) + C \quad (53)$$

Waiting for the recovery of the host

For the pessimistic case, we have:

$$F^*(s) = \left(\begin{array}{l}\dfrac{(1-\rho)R^*(s)}{1-\rho(1-\rho)Cr^*(s)R^*(s)}. \\ \left[1-\left(\rho(1-\rho)Cr^*(s)R^*(s)\right)^{d_i+1}\right] + C\end{array}\right) \quad (54)$$

If explicit distribution of r is unknown, then:

$$E(l_i) = (r+Cr)\left(\frac{1-\rho}{(1-x)^2}[1-x^d-(1-x)dx^d]\right) - Cr + C \quad (55)$$

6.3.2 Windowing Approach of Size 1

The same calculation of the TrT that is done for the windowing of size n is performed for the windowing of size 1 approach. However, this accomplished by using the value of d=1.

6.3.3 Centralized -Windowing Approach

1. Post-Failure – Pre-Failure Cases

Optimistic Case

The Laplace transform of the TrT satisfies:

$$F^*(s) = (1-\rho)R^*(s)\left(\begin{array}{l}\upsilon\dfrac{1-(R^*(s)CK^*(s)\overline{R}^*(s))^{d+1}}{1-(R^*(s)CK^*(s)\overline{R}^*(s))} \\ + \\ (1-\upsilon)\dfrac{1-(\rho R^*(s)CK^*(s)\overline{R}^*(s))^{d+1}}{1-\rho R^*(s)CK^*(s)\overline{R}^*(s)}\end{array}\right) + C \quad (56)$$

However, if the explicit distribution is unknown, the average TrT satisfies:

$$E(l_i) = (r+\overline{r}+ck)\left(\upsilon(1-\rho)+(1-\upsilon)\left(\frac{1-\rho^d}{1-\rho} - d\rho^d\right)\right) - \overline{r} - ck + C \quad (57)$$

Pessimistic Case:

The Laplace transform of the TrT is:

$$F^*(s) = (1-\rho)R^*(s)E^*(s)\left(\begin{array}{l}\upsilon\dfrac{1-(R^*(s)E^*(s)CK^*(s)\overline{R}^*(s))^{d+1}}{1-(E^*(s)R^*(s)CK^*(s)\overline{R}^*(s))} \\ + \\ (1-\upsilon)\dfrac{1-(\rho(1-\rho)R^*(s)E^*(s)CK^*(s)\overline{R}^*(s))^{d+1}}{1-\rho(1-\rho)E^*(s)R^*(s)CK^*(s)\overline{R}^*(s)}\end{array}\right) + C \quad (58)$$

However, in case that the explicit distribution of travel time is unknown, the average TrT satisfies:

$$E(l_i) = (r+\overline{r}+ck)\left(\begin{array}{l}(1-\upsilon)\dfrac{1-\rho}{(1-x)^2}[1-X^d-(1-X).d.X^d] + \\ \upsilon\dfrac{1-\rho}{(1-x)^2}[1-2X+X^2]\end{array}\right) - \overline{r} - ck + C \quad (59)$$

2. Host Crash

Ignoring the recovery of the host

The Laplace transform distribution of TrT can be expressed in optimistic case by:

$$F^*(s) = (1-\rho)R^*(s)\left(\begin{array}{l}\upsilon\dfrac{1-(R^*(s)CK^*(s))^{d+1}}{1-(R^*(s)CK^*(s))} \\ +(1-\upsilon)\dfrac{1-(\rho R^*(s)CK^*(s))^{d+1}}{1-\rho R^*(s)CK^*(s)}\end{array}\right) + C \quad (60)$$

In case that the explicit distribution of travel time is unknown, the average TrT satisfies:

$$E(l_i) = (r+ck)\left(\upsilon(1-\rho)+(1-\upsilon)\left(\frac{1-\rho^d}{1-\rho} - d\rho^d\right)\right) - ck + C \quad (61)$$

However, for the pessimistic case we have:

$$F^*(s)=(1-\rho)R^*(s)\begin{pmatrix} \upsilon\dfrac{1-(R^*(s)CK^*(s))^{d+1}}{1-(R^*(s)CK^*(s))}+ \\ (1-\upsilon)\dfrac{1-(\rho(1-\rho)R^*(s)CK^*(s))^{d+1}}{1-\rho(1-\rho)R^*(s)CK^*(s))} \end{pmatrix}+C \quad (62)$$

In case that the explicit distribution of travel time is unknown, the average TrT satisfies:

$$E(l_i)=(r+ck)\begin{pmatrix} (1-\upsilon)\dfrac{1-\rho}{(1-x)^2}[1-X^d-(1-X).d.X^d]+ \\ \upsilon\dfrac{1-\rho}{(1-x)^2}[1-2X+X^2] \end{pmatrix} \quad (63)$$

$$-ck+C$$

<u>Waiting for the recovery of the host</u>

The TrT when the explicit distribution of the travel time is known can be expressed as:

$$F^*(s)=(1-\rho)R^*(s)$$

$$\begin{pmatrix} \upsilon\dfrac{1-(R^*(s)CK^*(s)Cr^*(s))^{d+1}}{1-(R^*(s)CK^*(s)Cr^*(s))} \\ +(1-\upsilon)\dfrac{1-(\rho(1-\rho)R^*(s)CK^*(s)Cr^*(s))^{d+1}}{1-\rho(1-\rho)R^*(s)CK^*(s)Cr^*(s))} \end{pmatrix}+C \quad (64)$$

Otherwise, the average RTT satisfies:

$$E(l_i)=(r+cr+ck)$$

$$\begin{pmatrix} (1-\upsilon)\dfrac{1-\rho}{(1-x)^2}[1-X^d-(1-X).d.X^d]+ \\ \upsilon\dfrac{1-\rho}{(1-x)^2}[1-2X+X^2] \end{pmatrix} \quad (65)$$

$$-ck-cr+C$$

5.5 System Throughput Calculations

The system throughput calculations are done in the same manner as those done for the previous performance factors, and since it is equal to:

$$Thr_i=(Tr_i/TrSize) \quad (66)$$

$$Thr_i=RTT_i.\frac{1}{TrSize}+\frac{1}{BW} \quad (67)$$

$$Thr_i=RTT_i.c2+c1 \quad (68)$$

Then, for simplicity, we will not mention them here.

7. CONCLUSION

The novel mathematical model of the FTMAS is proposed. The paper states and defines the equations of the migration time, the Round trip time, the transfer time and the system throughput. These performance factors were defined relative to 3 performance metrics: the average connectivity of the network, the agent size and the probability of failure. The mathematical model of the FTMAS is stated as a promising model in the fault tolerant field of the mobile agent execution. In addition, the paper defined the equations of the novel failure classifications in terms of the performance factors and the performance metrics for each classified case. FTMAS model is to be analyzed and proven later.

REFERENCES

[1] Stefan Pleisch, "Fault-Tolerant and Transactional Mobile Agent Execution", Phd Dissertation, Lausanne, EPFL. 2002.

[2] S. S. Manvi and P. Venkataram, "Applications of agent technology in communications: A review", Computer communications, 5th-June- 2004, pages: 1493-1508. http://eprints.iisc.ernet.in/10925/

[3] Mosaab Daoud, Qusay H. Mahmoud, "Monte Carlo Simulation-Based Algorithms for Estimating The Reliability Of Mobile Agent-Based Systems", Journal Of Network And Computer Applications, Vol. 31, Issue 1 , Pp. 19-31, University Of Guelph, Department Of Computing and Information Science, Guelph, Canada, 2008. http://portal.acm.org/citation.cfm?id=1321933

[4] Jens Hofflinger, "Mobile Shadows: Strategies to Extend the Shadow Protocol for Orphan Detection and Termination in A Mobile Agent System", Thesis, Stuttgart University, 1 April 1998. http://elib.uni-stuttgart.de/opus/frontdoor.php?source_opus=363

[5] Alan Fedoruk and Ralph Deters, "Improving Fault- Tolerance by Replicating Agents", AAMAS'02, July 15-19, Bologna, Italy, 2002. http://portal.acm.org/citation.cfm?id=544917

[6] M. Erdem Türsem, M. Hadi Güneş, Mustafa Yıldız and Selahattin Kuru, "Performance Analysis Of Mobile Agents Using Simulation", Thesis, Boğaziçi University, Department Of Computer Engineering, Işık University, Department Of Computer Engineering, Istanbul, Turkey.

[7] Simon Pears, Jie Xu and Cornelia Boldyreff, "A Dynamic Shadow Approach for Mobile Agents to Survive Crash Failures", ACM real time systems, Vol. 32, Issue 3, pp. 235 - 252, 2006. http://portal.acm.org/citation.cfm?id=1121238

[8] Peng Xu and Ralph Deters, "Using Event-Streams for Fault Tolerant Management In MAS", IAT Proceedings Of The Intelligent Agent Technology, IEEE/WIC/ACM International Conference, Department Of Computer Science, pp. 433 – 436, 2005. http://portal.acm.org/citation.cfm?id=1019906

[9] S. Pleisch and A. Schiper, "FATOMAS - A Fault-Tolerant Mobile Agent System Based On The Agent-Dependent Approach," Proceedings of Int'l Conference On Dependable Systems and Networks, pp. 215-224, 2001.http://lsewww.epfl.ch/Documents/acrobat/PS01.pdf

[10] Stefan Pleisch and Andre´ Schiper, "Fault-Tolerant Mobile Agent Execution", IEEE Transactions on Computers, Vol. 52, No. 2, February 2003. http://ieeexplore.ieee.org/iel5/12/26428/01176987.pdf

[11] Wenyn Qu and Hong Shen, "Performance Modeling of a Fault Tolerance Agent – Driven System", IEEE International Conference on Communications, Graduate School Of

Information Science. Japan, Vol. 1, Issue 1, pp. 153 – 157, 16-20 May 2005. http://ieeexplore.ieee.org/iel5/9996/32109/01494338.pdf

[12] Wenyu Qu, Hong Shen and Zhiguang Shan, "A Novel Fault-Tolerant Execution Model by Using of Mobile Agents", Journal of Network and Computer Applications, vol. 42, no. 2, pp. 424 - 442, March 2009. http://portal.acm.org/citation.cfm?id=1304461

[13] Wenyn Qu and Hong Shen, "Analysis of Mobile Agents' Fault Tolerant Behavior", IAT Proceedings of the Intelligent Agent Technology, IEEE/WIC/ACM International Conference, IEEE Computer Society. Washington, DC, USA, pp. 377 – 380, 2004. www.springerlink.com/index/BEUEJ8HMPJDK54PC.pdf