Journal of Engineering Science and Technology
Vol. 9, No. 2 (2014) 206 - 219

FUZZY DECISION MAKING MODEL FOR BYZANTINE AGREEMENT

S. MURUGAN[1,*], V. RAMACHANDRAN[2]

[1]Department of Information Technology, Sathyabama University
Jeppiaar Nagar, Rajiv Gandhi Salai, Chennai – 600 119, Tamilnadu, India
[2]Department of Information Science and Technology, Anna University
College of Engineering, Guindy, Chennai – 600 025, Tamilnadu, India
*Corresponding Author: snmurugan@live.com

Abstract

Byzantine fault tolerance is of high importance in the distributed computing environment where malicious attacks and software errors are common. A Byzantine process sends arbitrary messages to every other process. An effective fuzzy decision making approach is proposed to eliminate the Byzantine behaviour of the services in the distributed environment. It is proposed to derive a fuzzy decision set in which the alternatives are ranked with grade of membership and based on that an appropriate decision can be arrived on the messages sent by the different services. A balanced decision is to be taken from the messages received across the services. To accomplish this, Hurwicz criterion is used to balance the optimistic and pessimistic views of the decision makers on different services. Grades of membership for the services are assessed using the non-functional Quality of Service parameters and have been estimated using fuzzy entropy measure which logically ranks the participant services. This approach for decision making is tested by varying the number of processes, varying the number of faulty services, varying the message values sent to different services and considering the variation in the views of the decision makers about the services. The experimental result shows that the decision reached is an enhanced one and in case of conflict, the proposed approach provides a concrete result, whereas decision taken using the Lamport's algorithm is an arbitrary one.

Keywords: Byzantine, Fuzzy entropy, Fault tolerance, Fuzzy Hurwicz rule.

1. Introduction

In today's complex distributed business applications, reliability of individual element is of major concern which is to be maintained at higher level to keep the

Nomenclatures

c_i	QoS parameters
d_i	Panel of decision makers
$\tilde{e}_j$	Entropy measure
fp_i	Fuzzy payoff value
g_i	Grade of membership
Hf_{ai}	Highest grade of membership
HV_i	Hurwicz value for the i^{th} alternative
Lf_{ai}	Lowest grade of membership
m_{ij}	No. of messages received by a service
P_{max}	Maximum of X
$\tilde{R}$	Normalized opinion matrix
S_i	Services
v_i	Message values of the decision variables commit or rollback
$\tilde{W}_j$	Weight of entropy measure
X	Set of fp_i
$\tilde{X}$	Opinion matrix
$\tilde{Z}$	Performance rating matrix

Greek Symbols

α	Optimism-pessmism index
γ	Index of optimism
$\wedge$	Minimum operator
$\vee$	Maximum operator

other components in the system intact and hence to make the entire system available. System failure is the major concern of reliability study which arises by several factors such as hardware, software, operator or network errors and reason for these faults are due to congestion in networks, server load, hardware or software fault at server end, denial of service attack and other such factors. Faults may be classified as transient faults, intermittent faults and permanent faults [1]. Fault detection and recovery are the major functions of any fault tolerant framework. Web based applications should adopt fault detection and recovery techniques to handle errors as propagated by the respective services and to ensure end-to-end Quality of Service (QoS). The faults which are Byzantine in nature cannot be handled using regular fault tolerant techniques because it is a system fault and even in the presence of fault, the faulty module continues to execute and the results produced are triggering false hope and arbitrary in nature. In the Byzantine failure model, a faulty process could forward incorrect information about messages received from other processes. These faults are difficult to handle. A system that can tolerate a Byzantine fault can tolerate any other fault [2].

Lamport et al. [3] proposed an algorithm to eliminate the Byzantine fault in which an agreement is arrived based on the messages that are exchanged between the processes. In case of tie, this algorithm uses an arbitrary tie-breaker for making the decision and apart from the messages received no other additional parameters are considered for decision making. A fuzzy based decision making approach for solving Byzantine Generals problem is proposed in this paper. Each process in the network is assigned a membership grade based on their non-functional quality attributes like reliability, security, data integrity and availability. The membership grade is computed using Fuzzy Entropy measure [4] and the final decision is made by applying fuzzy Hurwicz rule [5].

The Byzantine generals problem [3] is built around an imaginary General in defense who makes a decision to attack or retreat, and must communicate the decision to his lieutenants. The general and some of the lieutenants may be traitors. Traitors cannot be relied for proper communication of orders; worse yet, they may actively alter messages in an attempt to subvert the process. The generals are collectively known as *processes.* The general who initiates the order is the *source process.* The orders sent to the other processes are *messages*. The general and lieutenants those send faulty messages are traitorous and termed as *faulty processes.* Loyal general and loyal lieutenants are *correct processes*. The order to retreat or attack is a message with a single bit of information: 0 or 1.

Castro and Liskov [6] extended the synchronized method of tolerating Byzantine faults to an asynchronous environment by incorporating several optimizing techniques to improve the response time. The author proposed the state machine replication protocol by implementing a distributed file system for tolerating Byzantine faults in asynchronous networks. Wenbing Zhao [7] had developed a module based Byzantine Fault Tolerance framework for Web services along with security mechanisms. Lamport's algorithm does not employ additional parameters while taking the decision and there is no provision for considering the Quality of Service attributes like reliability, availability, security and integrity of a process which are vital in determining the rating of the processes. Selection of reliable Web services based on QoS parameters will lead to an undisputed result [8]. Gracia et.al [9] proposed an integrated ranking system that enables both users and developers to have control over the ranking process. The measurement of non-functional QoS parameters is very difficult and cannot be quantified numerically in a precise manner. Evaluating non-functional QoS parameters using traditional problem solving techniques may produce results, of ambiguous nature. So a fuzzy based decision making approach is adopted in this proposed work.

In the proposed work, the QoS attributes of the services are measured by various evaluators. The evaluations are based on the individuals, the type of service they are examining, the time in which it is executed, input parameters and various other factors. These values are different from one evaluator to other evaluator and this will result in ambiguity and biased opinions, and hence it becomes hard to evaluate the final outcome. The decision about the service is to be attained with this unbalanced set of inputs. Hurwicz criterion [5] is utilized in the proposed fuzzy decision making approach, that selects the minimum and maximum payoff given to each service. Hurwicz criterion balances the optimistic and pessimistic views of experts by predicting the outcome of changes in the decision making process. The rule uses optimism-pessimism index α ($0 \leq \alpha \leq 1$) in order to make the decision a balanced one. In this approach, maximax (optimistic) is used for selecting the best one out of the many bests and maximin (pessimistic) is used for picking the best out of the worst cases by varying the values of α.

Fuzzy entropy measure is used to rank the services, since the quantification of non-functional QoS parameters are very difficult to compute. The membership grade is assigned to each service that depends on this rank. An enhanced methodology is adopted for tolerating Byzantine faults by applying fuzzy based decision making approach with various QoS parameters.

2. Decision Making in Byzantine Environment

Currently, distributed applications are composed of many Web services, which are maintained and operated by different vendors with different levels of reliability, security and fault handling mechanisms. A service may invoke many other services sending appropriate response to the client. Let the service directly invoked by the client be the Primary Service Provider (PS) and let all the other coordinating services are Associated Service Providers (AS). Each associated service provider performs a specific task which in turn may invoke many other Web services. All these associated service providers respond back to the primary service provider. Based on the responses that are received from the associated services, the primary service provider may respond with the processed result or error message to the client. By assuming that the result is either commit or rollback, if all the intermediate services send a commit message, then the primary service provider commits the entire process and similarly for rollback. These services on execution, perhaps some of them may behave abnormally by sending wrong responses intentionally or some of them affected by malicious attack which is referred as a Byzantine fault.

A client sends a request to the primary service provider which may invoke ‘n’ other associated services. When any one of the associated services returns a rollback, all the responses from other services are ignored. The entire process is repeated until all the services agreed upon a common response. This leads to repetition of entire process. If the rollback is genuine, then repeating the entire set of operations is legitimate. Otherwise if the process is a faulty one and generates Byzantine behaviour then repeating the entire set of operations again and again is not an acceptable one. There are chances that the primary service provider may also exhibit Byzantine behaviour. This Byzantine behaviour can be solved by applying Lamport’s algorithm. Every service processes the request and the response is prepared. As per the Lamport’s algorithm this response is shared among all the other services. If the service is genuine then the message sent by that service to every other service is analogous. Otherwise the service may send the message commit to some services and rollback to some other services. Based on the received messages, every associated service takes its own decision and the final decision is taken by the primary service provider.

To explain this behaviour, a Web based order processing service is taken into consideration for analysis as the primary service provider which is to be invoked by the client directly. The associated services involved in the system are Warehousing service, Sales service and Accounts service. When the client sends a request to the order processing service, it has to validate the contract by getting approval from warehousing, sales and accounts services. If any of the service sends a negative response, then the contract is rejected. The response sent by the services may or may not be genuine. Some services may behave maliciously and intentionally produce incorrect response. To avoid this Byzantine behaviour, all the services will send each other’s result to every other associated service. These messages are to be validated and the final decision is based on the received responses.

A process may exhibit a malicious behaviour when it is affected by viruses, network attacks, human intervention and many other factors. Because of these attacks, the process produces faulty responses. In the example, Warehousing and Sales services send commit message to each other and to other services. The accounts service sends commit to all other services and rollback to order processing service. In this assumption, the accounting service is the malicious service which is not sending consistent message across all the services. As per the distributed services paradigm, when any one of the service returns a rollback all the results of associated services are ignored based on the decision on the requirement. The entire process is repeated until all the services agreed with consistent required message. To avoid this repetition of work when a malicious process is behaving abnormally, Byzantine agreement is applied for taking the final decision. Applying Lamport's algorithm for this example, the agreement is commit since majority of the messages that are received by the services are commit. In the same example, when the number of commit and rollback messages are equal, then the decision is made by using an arbitrary tie breaker and it may either be rollback or commit.

Decision taken by the Lamport's algorithm considers only the number of messages transmitted and is not providing a solution for the problem in the event of conflict. To overcome these limitations for solving the Byzantine agreement, a fuzzy based Hurwicz criterion is proposed to make the decision effectively.

3. Byzantine Agreement using Fuzzy Hurwicz Criterion

Let S_i (i = 1, 2, …, n) be the set of services which includes the primary service and the associated services. The primary service may invoke many other associated services in order to accomplish the task requested by the client. The response of each associated service is shared among / forwarded to other services except the primary service. This process continues till all the services receive the messages that are sent by every other service.

A payoff table is generated based on the total number of messages exchanged and the message values that are transmitted. In the payoff table, the set of alternatives are the message values shared among the services, i.e., 0 or 1 that represents commit or rollback and the events are the services in the group as shown in Table 1.

Table 1. Payoff Table.

Values / Services	S_1	S_2	…	S_n
v_0	m_{01}	m_{02}	…	m_{0n}
v_1	m_{11}	m_{12}	…	m_{1n}

where v_i is the Message values of the decision variables commit or rollback {i=0, 1}, S_i is the Services, and m_{ij} is the Number of messages received by a service. (i = 0, 1; j = 1, 2,…, n)

Each service S_i in the group is associated with a grade of membership g_i, $g_i \in [0, 1]$. The grade of membership g_i is derived using fuzzy entropy measure based on the non-functional QoS parameters of each service. The elements of the payoff matrix are awarded based on the subjective judgment of the services. The fuzzy payoff values associated with each decision variable (commit or rollback) is given in Eq. (1).

$$fp_i = \{(g_j \mid m_{ij}) \mid i=0,1; j=1, 2,\ldots, n\} \quad (1)$$

The set of possible payoff values (X) and the maximum of these values (P_{max}) is given by,

$$X = \{ m_{ij} \mid i = 0, 1; j = 1, 2,\ldots, n\} \quad (2)$$

$$P_{max} = \mathrm{SUP}_x (X) \quad (3)$$

The maximizing set for each alternative (i.e., decision variable) is computed using the following Eq. (4).

$$XM_i = \{((m_{ij} / P_{max}) \mid \mathrm{m_{ij}}) \mid i = 0, 1; j = 1, 2,\ldots, n\} \quad (4)$$

The fuzzy payoff (decision making) sets for each alterative is obtained by:

$$PS_i = \{ ((fp_i \wedge XM_i) \mid m_{ij}) \mid i = 0, 1; j = 1, 2, \ldots, n\} \quad (5)$$

where $\wedge$ is the minimum operator.

The highest grade (Hf_{ai}) of membership and the lowest grade (Lf_{ai}) of membership for each alternative are determined by Eqs. (6) and (7).

$$Hf_{ai} = \vee (g_j \in (PS_i \mid i = 0, 1) \mid j = 1, 2, \ldots, n) \quad (6)$$

where, $\vee$ is the maximum operator

$$Lf_{ai} = \wedge (g_j \in (PS_i \mid \mathrm{i} = 0, 1) \mid j = 1, 2, \ldots, n) \quad (7)$$

By applying Hurwicz rule [5], the Hurwicz balanced fuzzy decision set is arrived by using Eq. (8)

$$HV_i = \alpha\, Hf_{ai} + (1 - \alpha)\, Lf_{ai} \quad (8)$$

where HV_i is the Hurwicz value for the i^{th} alternative and α is the optimism-pessmism index, $\alpha \in [0, 1]$

4. Ranking the Services Using Fuzzy Entropy Measure

In this process, a group of decision makers analyzes the performance of a set of services based on non-functional QoS parameters. The performance of the services is evaluated and has been assigned with linguistic values based on the opinions of the decision makers. The non-functional parameters address various issues including availability, reliability, data integrity and security of primary and associated services. Since linguistic values have been assigned to the services to represent their weightage based on their performance, the computation of rank for the services using traditional computing methods is not feasible. In order to estimate the rank of the services based on the linguistic values, a fuzzy entropy measure [4] is proposed to quantify the linguistic values by means of grades of membership.

Each service 'S_i' is to be allotted a grade of membership based on its performance. A panel of decision makers (d_1, d_2, …, d_m) have to assess each S_i for its performance, i.e., to evaluate its QoS parameters (c_1, c_2, …, c_j). Every decision maker evaluates the services individually and assigns performance rating using linguistic terms. A set of linguistic values used to assign for the services to represent their performance level is as follows: {very poor, poor, fair, good, excellent}. These linguistic values are converted as triangular fuzzy number by applying (i-2, i, i+2), where the value 'i' depends on the rating given by the decision maker with the lower and upper bounds are restricted to 1 and 9 respectively.

For each service S_i, the performance rating is to be assigned by the decision makers with regard to the QoS parameters and hence a consolidated opinion matrix $\widetilde{X}$ has been formulated. The matrix $\widetilde{X}$ is to be normalized in order to convert the distinct QoS parameters into numerically comparable scale [4]. Utility oriented distinct QoS parameters namely reliability, security, integrity and availability are considered for analysis. The normalized opinion matrix is represented by $\widetilde{R}$ and it is computed using the set of QoS parameters for each service. By applying maximum entropy theory, the entropy measure of the j^{th} parameter is computed and represented as $\widetilde{e}_j$. The weight of each QoS parameter is to be obtained from the computed entropy measure and can be represented in fuzzy form $\widetilde{W}_j$. Fuzzy simple additive weighting method is applied on $\widetilde{W}_j$ for obtaining the priority of decision variables and based on this the final fuzzy performance rating matrix $\widetilde{Z}$ is computed in order to rank the services. The matrix $\widetilde{Z}$ is in fuzzy form and using α-cuts and interval arithmetic, this matrix has to be transformed into numerical values which enables to make a concrete decision on ranking the services. An index of optimism γ, $\gamma \in [0, 1]$ is considered to indicate the confidence level of the decision maker. By using Hurwicz rule and interval arithmetic with varying α-cuts, the precise performance rating matrix $\widetilde{\widetilde{z}}_{ij}^{\alpha}$ is formulated. The services are ordered based on the values in the matrix $\widetilde{\widetilde{z}}_{ij}^{\alpha}$. This order represents the rank of the services, using which, the grade of membership is assigned to the services.

5. Case Study: Order Processing Web Services

An order processing system is considered as case study for analyzing the Byzantine agreement using the proposed decision making approach. This system consists of four Web services namely Order processing service (S_1), Warehousing Service (S_2), Sales Service (S_3) and Accounts Service (S_4), which are coordinating with each other in order to accomplish their intended tasks. Let S_1 be the primary service and the services S_2, S_3 and S_4 be associated services. The client invokes the primary service which in turn invokes other services S_2, S_3 and S_4. The messages exchanged among these services are either 0 or 1, i.e., commit or rollback.

In this case study, it is assumed that the service S_2 is traitor which sends different messages to other services. Figure 1 shows the message exchange pattern among the services. The service S_1 sends a message 0 (commit) to the services S_2, S_3 and S_4. Since it is assumed that the service S_2 is traitor, it dispatches 1 to S_3 and 0 to S_4. Other services are sending/forwarding the messages to every other service without any modification.

A payoff table is created using the number of messages transmitted to each service, which is shown in Table 2.

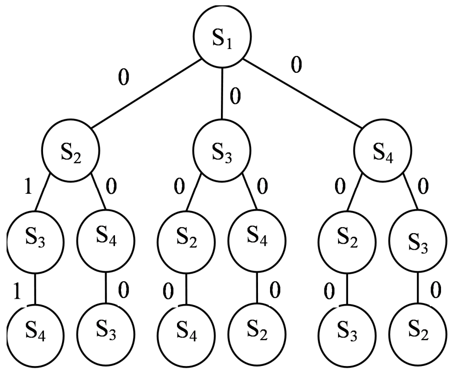

Fig. 1. Message Exchange Pattern among the Services.

Table 2. Pay Off Table.

Values / Services	S_2	S_3	S_4
v_0	5	4	4
v_1	0	1	1

where v_0 is the number of messages received with a decision variable "commit" and v_1 is the number of messages received with a decision variable "rollback".

The computation process for assigning the grade of membership to the services using fuzzy entropy measure is described as follows:

The services are evaluated by the decision makers based on the non-functional QoS parameters. The parameters used by the evaluators are data integrity (c_1), reliability (c_2), security (c_3) and availability (c_4). The performance ratings assigned by the decision makers for the services based on the QoS parameters using linguistic terms have been converted into triangular fuzzy numbers as represented in Table 3.

The consolidated opinion of the decision makers on each service with respect to each QoS parameter shall be obtained from the above matrix by taking the average of the respective value from the triangular fuzzy numbers. The consolidated opinion matrix thus obtained is shown in Table 4.

Normalized opinion matrix will be constructed by dividing each value of the fuzzy triangular number with respect to a particular QoS parameter by the maximum fuzzy triangular number associated with that QoS parameter. The normalized opinion matrix is shown in Table 5.

Fuzzy entropy theory [4] is applied to the normalized opinion values in order to estimate the weight of each QoS parameter using the Eq. (9) and the normalized weight of each QoS parameter is obtained using Eq. (10).

$$\tilde{e}_j = \frac{-1}{\ln(n)} \otimes [(\tilde{p}_{ij} \otimes \ln \tilde{p}_{ij}) \oplus ... \oplus (\tilde{p}_{nj} \otimes \ln \tilde{p}_{nj})] \quad (9)$$

where n is the number of services and the values of $\widetilde{p}_{ij}$ are computed by dividing each triangular fuzzy number of each column by the sum of normalized opinion values of that column.

$$\widetilde{w}_j = (1-\widetilde{e}_j)/[(1-\widetilde{e}_j) \oplus ... \oplus (1-\widetilde{e}_j)] \quad (10)$$

where $\widetilde{e}_j$ is the entropy measure obtained in Eq. (9). Using the normalized opinion matrix and $\widetilde{w}_j$, the final fuzzy performance rating matrix $\widetilde{Z}$ is formulated using which the services have been ranked. The matrix $\widetilde{Z}$ is in fuzzy form and using α-cuts and interval arithmetic, this matrix is transformed into crisp set which enables to make a concrete decision on ranking the services. By using Hurwicz rule and interval arithmetic with varying α-cuts, the precise performance rating matrix $\widetilde{z}_{ij}^{\alpha}$ is formulated and shown in Table 6.

Table 3. Triangular Fuzzy Number Representation of Performance Ratings of the Services.

Criteria	WS/DM	d_1			d_2			d_3			d_4		
c_1	S_1	1	1	3	1	3	5	3	5	7	5	7	9
	S_2	7	9	9	7	9	9	5	7	9	1	3	5
	S_3	1	1	3	3	5	7	1	3	5	1	1	3
	S_4	5	7	9	5	7	9	5	7	9	5	7	9
c_2	S_1	3	5	7	3	5	7	5	7	9	5	7	9
	S_2	7	9	9	1	1	3	5	7	9	5	7	9
	S_3	1	3	5	1	3	5	3	5	7	3	5	7
	S_4	5	7	9	5	7	9	5	7	9	5	7	9
c_3	S_1	1	3	5	5	7	9	5	7	9	3	5	7
	S_2	7	9	9	3	5	7	7	9	9	3	5	7
	S_3	1	3	5	1	3	5	1	3	5	5	7	9
	S_4	5	7	9	5	7	9	5	7	9	5	7	9
c_4	S_1	1	1	3	1	1	3	3	5	7	1	3	5
	S_2	5	7	3	1	3	5	5	7	9	1	3	5
	S_3	1	1	3	5	7	9	3	5	7	1	1	3
	S_4	1	3	5	1	3	5	7	9	9	1	1	3

Table 4. Consolidated Opinion Matrix.

	c_1			c_2			c_3			c_4		
S_1	2.50	4.00	6.00	4.00	6.00	8.00	3.50	5.50	7.50	1.50	2.50	4.50
S_2	5.00	7.00	8.00	2.00	4.00	6.00	2.00	4.00	6.00	3.00	5.00	5.50
S_3	1.50	2.50	4.50	4.50	6.00	7.50	5.00	7.00	8.00	2.50	3.50	5.50
S_4	5.00	7.00	9.00	5.00	7.00	9.00	5.00	7.00	9.00	2.50	4.00	5.50

Table 5. Normalized Opinion Matrix.

	c_1			c_2			c_3			c_4		
S_1	0.28	0.44	0.67	0.44	0.67	0.89	0.39	0.61	0.83	0.27	0.4	0.82
S_2	0.56	0.78	0.89	0.22	0.44	0.67	0.22	0.44	0.67	0.55	0.91	1.00
S_3	0.17	0.28	0.50	0.50	0.67	0.83	0.56	0.78	0.89	0.45	0.64	1.00
S_4	0.56	0.78	1.00	0.56	0.78	1.00	0.56	0.78	1.00	0.45	0.73	1.00

Table 6. Precise Performance Rating Matrix ($\tilde{z}_{ij}^{\alpha}$).

	$\alpha = 0.05$			$\alpha = 0.50$			$\alpha = 0.75$			$\alpha = 1$		
γ	0.20	0.50	1.00	0.20	0.50	1.00	0.20	0.50	1.00	0.20	0.50	1.00
S_1	3.05	6.99	13.55	2.64	4.71	8.16	2.41	3.44	5.17	2.18	2.18	2.18
S_2	3.10	7.02	13.56	2.85	4.92	8.36	2.71	3.75	5.47	2.58	2.58	2.58
S_3	3.14	7.09	13.68	2.77	4.85	8.32	2.56	3.60	5.34	2.36	2.36	2.36
S_4	3.88	8.75	16.86	3.49	6.06	10.32	3.28	4.56	6.69	3.06	3.06	3.06

The final ranking of services is assessed by comparing the results which is the maximum value against each optimism level that is obtained in Table 6 and hence the services are ranked in the order $\{S_4, S_2, S_3$ and $S_1\}$. According the fuzzy set of services is represented as follows:

$\{(1.00, S_4), (0.75, S_2), (0.50, S_3), (0.25, S_1)\}$

These membership grades indicate that S_1 is not the reliable one as it has the lowest grade of membership. In the example, it is assumed that S_2 exhibits Byzantine behaviour, but its reliability level is better than S_1. This is because of the values that are provided by the decision makers which are based on the previous transactions in which S_2 found to be reliable and genuine one. This is one of the reasons for adopting fuzzy based decision making approach.

The elements of the payoff matrix (Table 2) are awarded based on the subjective judgment derived using the rank of the services. Considering the associated services, the fuzzy payoff values associated with each alternative are represented as follows:

$fp_0 = [(0.75, 5), (0.50, 4), (1.00,4)]$

$fp_1 = [(0.75, 0), (0.50, 1), (1.00, 1)]$

The set of possible payoff values and the maximum payoff value are represented using Eq. (2) and Eq. (3) as follows:

$X = [0, 1, 4, 5]$, $P_{max} = 5$

The maximizing sets for various alternatives are determined as follows:

Each payoff value is divided by P_{max} and the result is the grade of membership value of the actual payoff value.

$XM_0 = [(5/5, 5), (4/5, 4), (4/5, 4)]$, i.e., $XM_0 = [(1, 5), (0.80, 4), (0.80, 4)]$

$XM_1 = [(0, 0), (0.20, 1), (0.20, 1)]$

The fuzzy payoff (decision making) sets are obtained (using Eq. 5) as follows:

$PS_0 = [(0.75, 5), (0.50, 4), (0.80, 4)]$

$PS_1 = [(0, 0), (0.20, 1), (0.20, 1)]$

The highest grade of membership for each alternative is determined as follows (using Eq. 6):

$Hf_{a0} = \vee\,(0.75, 0.50, 0.80) = 0.80$

$Hf_{a1} = \vee\,(0, 0.20, 0.20) = 0.20$

The decision variable 'commit' has the highest grade of membership, therefore it is the optimal decision for the given payoff table.

Therefore the fuzzy set of alternatives is given by,

$A = \{(0.80, v_0), (0.20, v_1)\}$

From Eq. (7), the lowest grade of membership for each alternative is determined as follows.

$Lf_{a0} = \wedge (0.75, 0.50, 0.80) = 0.50$

$Lf_{a1} = \wedge (0.00, 0.20, 0.20) = 0.00$

The Hurwicz values for $\alpha = 0.5$ can be determined using Eq. (8) and are given as follows:

$HV_i = 0.50\ X\,[0.80, 0.20] + (1 - 0.50)\ X\,[0.50, 0.00] = [0.40, 0.10] + [0.25, 0.00]$

According to algebraic sum of two fuzzy sets, $HV_i = [0.55, 0.10]$.

The largest Hurwicz value is 0.55 and hence v_0 is the best alternative for the given payoff table. As per the Hurwicz decision, the order processing service shall commit the request. With the above working principle the following computations are performed.

5.1. Byzantine agreement with four nodes

Table 7 shows the Byzantine agreement for four nodes. In 1a and 1b it is assumed that there exist two traitors labeled as P_2 and P_4. The similar membership grades are assigned to the services and the message exchange patterns are different in the cases 1a and 1b. Even though the message exchange pattern is different, the decision obtained using the proposed fuzzy approach matches with the existing Lamport's algorithm. In sl. no 2, it is assumed that the source process is the traitor and the membership grades assigned are similar to 1a and 1b. The final decision is agreeing with Lamport's algorithm. P_2 is treated as traitor in the sl. no's 3 and 4, in which the message sent by the source process is different and equal membership grades are assigned. By changing traitors, message exchange patterns and membership grades the decision acquired is similar to Lamport's algorithm.

Table 7. Byzantine Agreement for Four Nodes.

Sl. No	Traitor	Membership Grade			HV_i (1,0)	Selection	Proposed Approach Decision	Lamport's algorithm Decision
		P_2	P_3	P_4				
1a	P_2, P_4	0.25	0.50	0.75	(0.3438, 0.4531)	0.4531	0	0
1b	P_2, P_4	0.25	0.50	0.75	(0.3438, 0.4531)	0.4531	0	0
2	P_1	0.25	0.50	0.75	(0.3438, 0.4531)	0.4531	0	0
3	P_2	0.25	0.25	0.25	(0.1000, 0.2434)	0.2434	0	0
4	P_2	0.25	0.25	0.25	(0.2344, 0.1000)	0.2344	1	1

5.2. Byzantine agreement with five nodes

Table 8 shows the fuzzy based decision making when the number of processes involved in processing a request is five. All the samples produced similar results as Lamport's algorithm. When the traitors are similar, different message patterns are adopted and membership grades are assigned

Table 8. Byzantine Agreement for Five Nodes.

Sl. No	Traitor	Membership Grade				HV_i (1,0)	Selection	Proposed Approach Decision	Lamport's algorithm Decision
		P_2	P_3	P_4	P_5				
1a	P_2	0.20	0.40	0.60	0.80	(0.0625, 0.4600)	0.4600	0	0
1b	P_2	0.20	0.40	0.60	0.80	(0.1250, 0.4600)	0.4600	0	0
2a	P_2, P_5	0.20	0.40	0.60	0.80	(0.2267, 0.4600)	0.4600	0	0
2b	P_2, P_5	0.20	0.40	0.60	0.80	(0.3423, 0.4600)	0.4600	0	0
3	P_1	0.20	0.40	0.60	0.80	(0.1944, 0.4300)	0.4300	0	0
4a	P_2	0.25	0.25	0.25	0.25	(0.0625, 0.2344)	0.2344	0	0
4b	P_2	0.50	0.50	0.50	0.50	(0.0625, 0.4375)	0.4375	0	0
5a	P_2, P_5	0.25	0.25	0.25	0.25	(0.1542, 0.2344)	0.2344	0	0
5b	P_2, P_5	0.50	0.50	0.50	0.50	(0.2267, 0.4375)	0.4375	0	0
5c	P_2, P_5	0.75	0.75	0.75	0.75	(0.3536, 0.5914)	0.5914	0	0
5d	P_2, P_5	0.10	0.11	0.12	0.13	(0.0962, 0.1118)	0.1118	0	0
5e	P_2, P_5	0.13	0.11	0.12	0.10	(0.0913, 0.1118)	0.1118	0	0
5f	P_2, P_5	0.90	0.70	0.50	0.30	(0.2267, 0.5325)	0.5325	0	0
6a	P_1	0.25	0.25	0.25	0.25	(0.1542, 0.2344)	0.2344	0	0
6b	P_1	0.50	0.50	0.50	0.50	(0.1944, 0.4375)	0.4375	0	0
6c	P_1	0.60	0.20	0.40	0.10	(0.1944, 0.3350)	0.3350	0	0

5.3. Byzantine agreement with six nodes

Table 9 shows the fuzzy based decision when number of processes is six. All the samples produced similar results as Lamport's algorithm. When the traitors are similar, different message patterns and membership grades are assigned.

5.4. Byzantine agreement in case of a tie

In a specific case, an equal set of message values are transmitted between the processes (i.e., the count of one's and zero's are equal). In this scenario the proposed fuzzy based approach yields a concrete result whereas the Lamport's algorithm decision is based on a tie-breaker. Table 10 shows this illustration.

The proposed fuzzy decision approach for Byzantine Agreement has been tested by considering various numbers of services, modified message exchange patterns, assigning different grade of membership to the services (based on the rank of the services) and by changing the traitors. The decision results have been compared with the results obtained using Lamport's algorithm. In some cases, the Lamport algorithm for Byzantine agreement produces conflict results which are resolved using an option of tie breaker, whereas the proposed approach yields an accurate result and it does not require any further computation to arrive at the decision.

Table 9. Byzantine Agreement for Six Nodes.

Sl. No	Traitor	Membership Grade					HV_i (1,0)	Selection	Proposed Approach Decision	Lamport's algorithm Decision
		P_2	P_3	P_4	P_5	P_6				
1a	P_2	0.90	0.70	0.50	0.30	0.10	(0.0385,0.4775)	0.4775	0	0
1b	P_2	0.25	0.25	0.25	0.25	0.25	(0.0385,0.2344)	0.2344	0	0
1c	P_2	0.50	0.50	0.50	0.50	0.50	(0.0385,0.4375)	0.4375	0	0
1d	P_2	0.75	0.75	0.75	0.75	0.75	(0.0385,0.6094)	0.6094	0	0
1e	P_2	0.10	0.30	0.50	0.70	0.90	(0.0769,0.4519)	0.4519	0	0
1f	P_2	0.25	0.25	0.25	0.25	0.25	(0.0769,0.2344)	0.2344	0	0
2a	P_2, P_4	0.50	0.50	0.50	0.50	0.50	(0.1615,0.4375)	0.4375	0	0
2b	P_2, P_4	0.10	0.30	0.50	0.70	0.90	(0.1615,0.4458)	0.4458	0	0
2c	P_2, P_4	0.10	0.30	0.50	0.70	0.90	(0.2083.0.4063)	0.4063	0	0
2d	P_2, P_4	0.75	0.75	0.75	0.75	0.75	(0.2083,0.6094)	0.6094	0	0
3a	P_1	0.10	0.30	0.50	0.70	0.90	(0.1318,0.4137)	0.4137	0	0
3b	P_1	0.10	0.30	0.50	0.70	0.90	(0.3472,0.3566)	0.3766	0	0
3c	P_1	0.50	0.50	0.50	0.50	0.50	(0.2656,0.4375)	0.4375	0	0
3d	P_1	0.50	0.50	0.50	0.50	0.50	(0.2734,0.3828)	0.3828	0	0

Table 10. Byzantine Agreement in Case of Tie.

Traitor	Membership Grade				HV_i (1,0)	Selection	Proposed Approach Decision	Lamport's algorithm Decision
	*P*2	*P*3	*P*4	*P*5				
*P*1	0.2	0.4	0.6	0.8	(0.3700, 0.4600)	0.4600	0	0 or 1

6. Conclusions

In a real time system, where many Web services are coordinating to complete a specific task, it is hard to measure the performance of those services precisely with respect to non-functional QoS parameters. It is also not possible to predict the services exactly those are behaving differently, i.e., behaving as traitors. Hence to tolerate the Byzantine behavior of certain services and to take appropriate decisions where the services are exhibiting imprecise performance in a distributed environment, an enhanced fuzzy decision making approach is carried out to derive a concrete decision for the action to be taken with the presence of traitors. The opinions of the decision makers about the performance of the services have been balanced using Fuzzy Hurwicz rule. Fuzzy entropy measures are evaluated based on the non-functional QoS parameters of each service, and hence the services are ranked. Grade of membership is assigned for each service based on its evaluated rank. A systematic fuzzy decision making algorithm has been explained using an order processing system with four services. The algorithm provides a concrete decision whether to commit or rollback an action based on the messages received, while the Lamport's algorithm uses a tie breaker when there is a conflict.

References

1. Burns, A.; and Wellings, A. (2001). *Real-time systems and programming languages* (3rd ed.). Addison Wesley.
2. Garg, V.K. (2002). *Elements of distributed computing*. Wiley-Interscience, New York.
3. Lamport, L.; Pease, M.; and Shostak, R. (1982). The Byzantine Generals Problem. *ACM Transactions on Programming Languages and Systems,* 4(3), 382 - 401.
4. Wang, P.; Chao, K.-M.; Lo, C.-C.; Huang, C.-L.; and Li, Y. (2006). A fuzzy model for selection of QoS-aware web services. Proceedings of IEEE International Conference on e-Business Engineering. 585-593.
5. Ramachandran, V.; and Sankaranarayan, V. (1991). Fuzzy concepts applied to statistical decision making methods. *Proceedings of the 15th IFIP International Conference*. Zurich, Switzerland.
6. Castro, M.; and Liskov, B. (1999). Practical Byzantine fault tolerance. *Proceedings of the 3rd Symposium on Operating Systems Design and Implementation.* New Orleans, USA.
7. Zhao, W. (2007). Byzantine fault tolerant coordination for web services atomic transactions. *Proceedings of the 5th international conference on Service-Oriented Computing (ICSOC '07).* Springer-Verlag, Berlin, Heidelberg, 307-318.
8. Chen, M.; and Ludwig, S.A. (2012). Fuzzy-guided genetic algorithm applied to the web service selection problem. *Proceedings of the IEEE International Conference on Fuzzy Systems*. 1-8.
9. Maria Garcia, J.; Junghans, M.; Ruiza, D.; Agarwal, S.; and Ruiz-Cortes, A. (2013). Integrating semantic web services ranking mechanisms using a common preference model. *Knowledge-Based Systems*, 49, 22-36.