New Approach to Improve the Generalized Byzantine Agreement Problem

Hui-Ching Hsieh and Mao-Lun Chiang

***Abstract*—To achieve reliability, it is very important to study the agreement and fault-tolerance topic in distributed systems. This kind of problem is known as a Byzantine Agreement (BA) problem. It requires a set of processors to agree on a common value, even if some processors and transmission media are faulty. Basically, the traditional BA protocols require $\lfloor(n\text{-}1)/3\rfloor$ + 2 rounds of message exchange to make each processor reach an agreement. In other words, those protocols are inefficient and unreasonable, due to the number of message results in a large protocol overhead. In this study, we propose a novel and efficient protocol to reduce the amount of messages. Our protocol can compare and replace the messages received from other processors to find out the *reliable processors* and replace the value sent by the un-reliable processors through four rounds of message exchange even when the total number of faulty components or the total number of processors in the system is large. Finally, the agreement can be reached by using the minimum number of messages in the distributed system, while tolerating the maximum number of faulty components.**

***Index Terms*—Byzantine agreement problem, fault-tolerance, distributed system.**

I. Introduction

During the past decade, large-scale and complex distributed system has grown at an astonishing rate. Under such a circumstance, assuring high reliability and correctness is an important issue. Hence, providing some protocols to help all fault-free processors to reach an agreement and then do some corresponding activities, even if there exist some influences caused by the faulty components, becomes necessary. The Byzantine Agreement (BA) problem, one of the famous agreement schemes in distributed systems, was first studied by Lamport *et al*. [1] in 1982. This problem states that all fault-free processors can reach a common value under the n-processor distributed system in which at most $\lfloor(n\text{-}1)/3\rfloor$ processors can be faulty.

In the past, various protocols [1]-[13] for the BA problem have been developed to satisfy the following requirements:

(Agreement): All fault-free processors agree on a common value;

(Validity): If the source is correct, then the common value v should be the source's initial value v_s; i.e. $v = v_s$.

Basically, Fischer [14] proved that $\lfloor(n\text{-}1)/3\rfloor$ +1 rounds (The term "round" represents the interval of message exchange between any pair of processors [1]) of message exchange are the necessary and sufficient lower bound for agreement problems when there exists processors fault only in the distributed system. In other words, the proposed protocols [1], [8], [11], [14] need $\lfloor(n\text{-}1)/3\rfloor$ +1 rounds of message exchange for collecting messages. Under such protocols, each fault-free processor always requires $\lfloor(n\text{-}1)/3\rfloor$ +1 rounds of message exchange to reach agreement even if all processors are correct. Basically, the reason why the previous works [1], [8], [11], [14] always need $\lfloor(n\text{-}1)/3\rfloor$ +1 rounds of message exchange is because that these protocols [1], [8], [11], [14] only focusing on collecting messages to reduce the influences caused by the faulty components by exchanging messages continuously. They do not compare or examine the messages which may help the protocol to solve the BA problem more efficiently.

Furthermore, in practical situation, both processors and transmission media may be faulty simultaneously. However, the previous works [1], [7], [8], [11], [14] only consider the agreement problem when there exists processors fault only, and the faulty transmission media are treated as faulty processors. As a result, the treatment ignores the fact that the processor connected with a faulty transmission medium, which was called innocent processor, is still in correct. Such a treatment has following drawbacks:

1) If a fault-free processor is treated as faulty, then it will not be included in the process of reaching a common value. This violates the definition of BA problem that requires every fault-free processor agree on a common value.
2) The number of faulty processors is enlarged, so that some fault-free processors may not be able to agree on a common value as they should.

Based on above discussion, the transmission medium failure should be treated differently from the case of processor failure.

In order to solve the BA problem, various kinds of protocol are proposed, in [12], the authors proposed a weighted based protocol for the Byzantine Agreement Problem under various conditions. In the proposed protocol [12], a weight is assigned to the machines separately, and assumes that the total weight of the faulty processors is at most f=N (f: the number of faulty processors). If the weights are decided appropriately, the weighted Byzantine Agreement Problem can be applied to solve the agreement even when more than N=3 processors are faulty, if the total weight of the faulty processors is less than 1=3. Basically, the weighted based protocol can solve the agreement problem, however, the weight must be considered every time while executing the protocol. It is more redundant than previous

Manuscript received March 21, 2014; revised May 15, 2014.

Hui-Ching Hsieh is with the Department of Information Communication, Hsing Wu Universitiy, No. 101, Sec.1, Fenliao Rd., Lin Kou District, New Taipei City 244, Taiwan, R.O.C. (e-mail: luckyeva.hsieh@gmail.com).

Mao-Lun Chiang is with the Department of Information and Communication Engineering, Chaoyang University of Technology 168, Jifeng E. Rd., Wufeng District, Taichung County, 41349 Taiwan, R.O.C (e-mail: mlchiang@cyut.edu.tw).

protocol [1], [8], [11], [14]. Furthermore, it still needs f+1 rounds of message exchange to collect message for the worst case. Finally, the faulty component is limited to processors only; hence, the weighted based protocol is not suitable for the generalized Byzantine Agreement protocol.

In the past, there also have some protocols [5], [15], [16] been proposed to solve the BA problem even if the processors fault and transmission media fault exist simultaneously. Based on the previous protocols [5], [15], [16], each processor needs to exchange $\lfloor(n\text{-}1)/3\rfloor + 2$ rounds of message change to make all fault-free processors to reach an agreement. Besides, the total number of allowable faulty components must be less than or equal to $\lfloor n/2\rfloor\text{-}1$, in which the number of allowable faulty processor must be less than or equal to $\lfloor(n\text{-}1)/3\rfloor$ and the rest is the number of allowable faulty transmission media [15].

Based on the above result, there can be no doubt that the previous protocols [5], [15], [16] can solve the agreement problem. The issue we must consider next is improving the efficiency.

In this study, we proposed a New Generalized BA protocol (NGBA) to solve the BA problem when both of the processors fault and transmission media fault exist simultaneously. Here, NBGA can apply the function MAJ (α) to eliminate the influence caused by faulty transmission media. Furthermore, the NGBA protocol can compare and count the received values to find out the reliable processors correctly. Subsequently, the majority values of the reliable processors can be used to replace the values received from the un-reliable processors through four rounds of message exchange even if some processors and transmission media are faulty simultaneously. Finally, the agreement can be reached.

Noticeable, NGBA is more efficient and reasonable than previous works [5], [15], [16]. For example, the traditional protocols [5], [15], [16] require $\lfloor(n\text{-}1)/3\rfloor + 2$ rounds of message exchange to reach agreement in a n-processor distributed system, and the message complexity is O (n^n). Our protocol only requires four rounds of message exchange to reach agreement and the message complexity is O (n^3). Therefore, this is more suitable for the environment in which there exist a large number of processors.

The rest of this paper is organized as follows: The details of the proposed protocol are given in Section II. The correctness and complexity is shown in Section III. Finally, the conclusion is presented in Section IV.

II. The Details of the NGBA Protocol

In general case, both processors and transmission media may be faulty. If the fault-free processors want to reach a common value, they must remove the influence caused by the faulty transmission media first and then remove the influence caused by the faulty processors [16]. After that, all fault-free processors can find out the reliable processor, replace the values received from the un-reliable processors, and reach an agreement value. The procedures of removing the influences caused by the faulty components are shown in Fig. 1.

At the start of the second round of message, the messages in the first and second level of the ms-tree are influenced by the malicious faulty processors and malicious faulty links simultaneously. Now, all fault-free processors can apply the function MAJ (α) to remove the influence caused by the malicious faulty links in the first level of the ms-tree. And, the procedures are shown in Fig. 1(a) and Fig. 1(b). With the same principle, the function MAJ (α) can also remove the influences caused by the malicious faulty links in the second and third level of the ms-tree. And the procedures are shown in Fig. 1(c) to Fig. 1(f). After that, all fault-free processors must delete the messages in the fourth level of the ms-tree [7], and then the remaining messages in the three level of ms-tree will not be influenced by the malicious faulty links anymore.

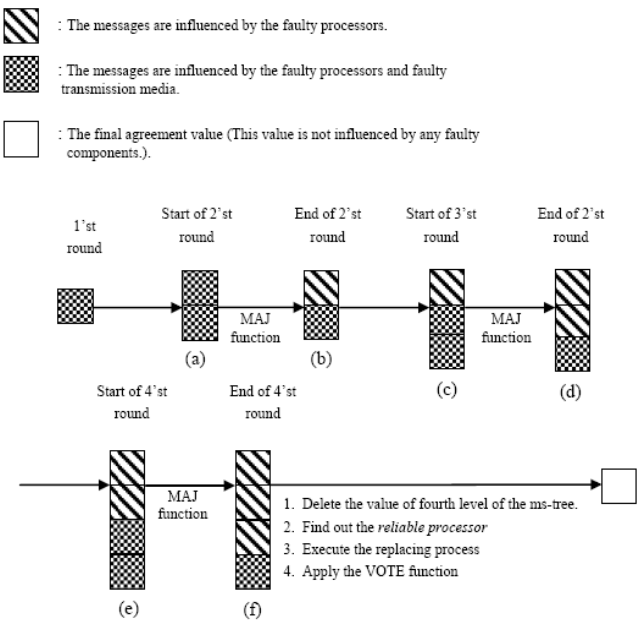

Fig. 1. The procedure of removing the influences caused by the faulty components and getting agreement value.

After finish removing the influence caused by the malicious links, all fault-free processors can find out the reliable processors by examining the values stored in the second and third level of the ms-tree. Subsequently, the messages received from the un-reliable processors can be replaced by the majority values of the reliable processors. Finally, all fault-free processors can reach an agreement by applying the VOTE function, and the details of the protocol are described below.

Basically, there are two phases in NGBA: the *message exchange phase* and the *decision making phase*. The goal of the *message exchange phase* is to collect four rounds of message exchange and store the received messages in the processor's ms-tree. Upon completion of the message exchange phase, the *decision making phase* is invoked. There are three objectives of this phase, shown as follow:

1) Decide which processors are reliable by comparing the messages sent from all processors.
2) Apply the replacing process to replace the values received from the un-reliable processors.
3) Apply the VOTE function from the third level to root of each fault-free processor's ms-tree and determine the agreement value.

After the decision making phase, each fault-free processor

can reach a common value in a distributed system even if the failure exists. Subsequently, the details are described in Fig. 2.

In previous protocols [5], [15], [16], $\lfloor (n\text{-}1)/3 \rfloor + 2$ rounds of message exchange are required to reach an agreement where the faulty processors and faulty transmission media exist simultaneously. Besides, the message complexity of previous protocols is O (n^n). As a result, the more number of message exchange are necessary when processors are getting larger. This will cause a large number of protocols overhead. However, each processor in NGBA protocol only needs to execute four rounds of message exchange to get an agreement, using the concept of reliable processor. In this case, the message complexity is O (n^3). Therefore, NGBA is more efficient than previous works [5], [15], [16] when the number of the processors is large in the network. This is because the NGBA protocol still requires only four rounds of message exchange, but $\lfloor (n\text{-}1)/3 \rfloor + 2$ rounds of message exchange. Due to less rounds of message exchange, the transmission time can be reduced, too. In view of time-saving, space-saving, the proposed protocol NGBA is more efficient than previous protocol.

***Message exchange phase*:**

$r = 1$ do:

- The source broadcasts its initial value v_s to other processors and itself.
- Each processor stores v_s in the root of its ms-tree;

For $r = 2$ to 4 do:

- Each processor broadcasts the value at level (r-1)th of its ms-tree to other processors and itself.
- Function $MAJ(\alpha)$ is used to each vertex at level r-1 for each processor's ms-tree and the values which have applied the function $MAJ(\alpha)$ are broadcasted to others and itself.
- Each processor stores the received values at level rth of its ms-tree.
- Delete the repeatable vertices in the ms-tree.

***Decision-making phase*:**

1. Each processor deletes the values at the fourth level of the ms-tree.
2. Determining *reliable processors*:

For each sub-tree of vertex $v(ax)$ in the second level of ms-tree. {
If $v(ax)=maj_{sib\_3}(ancestor_{ax})$ and $\#\ maj_{sib\_3}(ancestor_{ax}) \geq (n-\lfloor (n\text{-}1)/3 \rfloor\text{-}1)$ then {
Add processor x into RLP_x.
For each vertex whose parent is vertex $v(ax)$ {
If $v(axy) = v(ax)$ then
Add processor y to RLP_x }}}
For each processor z (denoted as p_z) {
Count $\#p_z$ from all of the RLP set
If $\#p_z \geq (n-\lfloor (n\text{-}1)/3 \rfloor)$ then {
Processor z is a *reliable processor* }}

3. The replacement process:

For each vertex whose parent is $v(ax)$ {
If processor y is not a *reliable processor* and $v(axy) \neq maj_{sib\_3\_RP}(ancestor_{ax})$ then
$v(axy) = maj_{sib\_3\_RP}(ancestor_{ax})$}

4. Applying function VOTE to the root of each processor's ms-tree, and common value is obtained.

Fig. 2. The proposed protocol NGBA.

III. The Correctness and Complexity

In this suction, we will prove the correctness and complexity of NGBA. Basically, each fault-free processor should be insulated from the influence of faulty processors and faulty transmission media. In our protocol, the influence of the faulty transmission media can be removed in the message exchange phase by using function $MAJ(\alpha)$, and the influence of faulty processors can be removed in the decision making phase, then the collected messages of fault-free processors are uninfluenced and an agreement is reached. Before analyzing the protocol, several terminologies must be defined.

A vertex α is defined as common [2] if each fault-free processor computes the same value for α. In other words, the value stored in vertex α of each fault-free processor's ms-tree is identical to all processors. When each fault-free processor has a common initial value from the source in the root of its ms-tree, an agreement is reached, because the root is common to all. Thus, the agreement (Agreement') and (Validity'), can be rewritten as:

(Agreement'): Root s is common, and (Validity'): $VOTE(s) = v_s$ for each fault-free processor, if the source is correct.

To prove that the vertex is common, the term common frontier [2] is defined as: when every root-to-leaf path of the ms-tree contains a common vertex, then the collection of the common vertices forms a common frontier. In other words, every fault-free processor has the same messages collected in the common frontier, if a common frontier does exist in a fault-free processor's ms-tree. Subsequently, by using the same voting function to compute the root value of the ms-tree, every fault-free processor can obtain the same root value, because they have the same input and the same computing function. Because the protocol can solve the BA problem, the above concepts can be used to prove the correctness of the proposed protocol NGBA.

Before proving the correctness of the protocol, the term correct vertex is defined as: vertex αi of a tree is a correct vertex if processor i is correct.

Lemma 1: At the rth round, $v(\alpha) = MAJ(\alpha i)$ is applied to the vertex in the (r-1)th level of each fault-free processor j's ms-tree, where $1 \leq j \leq n$, and $2 \leq r \leq 4$. The results applied by function $MAJ(\alpha)$ should be either $v(\alpha)$or $\neg\, v(\alpha)$.

Proof: Part 1. Transmission medium between processor i and processor j is correct.

If transmission medium between processor i and processor j is correct, processor j will receive $v(\alpha)$ from processor i in the round r-1, and $v(\alpha i) = v(\alpha)$. Meanwhile, processor i will broadcast the value $v(\alpha)$ to the others. There are (n-1) transmission media connected with a processor in which at most $\lfloor n/2 \rfloor$-1 transmission media are faulty in the system. In the next round, processor j receives at least $(n\text{-}1) - (\lfloor n/2 \rfloor\text{-}1) = \lceil n/2 \rceil$ $v(\alpha ik) = v(\alpha)$ in the children of vertex αi from the other processor k, for $1 \leq k \leq n$ and $k \neq j$. Since $v(\alpha ij) = v(\alpha i) = v(\alpha)$ for processor j, there are at least $\lceil n/2 \rceil$+1 $v(\alpha)$'s, $\lceil n/2 \rceil$ $v(\alpha ik)$'s and one $v(\alpha ij)$ are equal to $v(\alpha)$ in the children of vertex αi for processor j, and $MAJ(\alpha i)$ should be equal to $v(\alpha)$.

Part 2. Transmission medium between processor i and processor j is faulty.

In the case of transmission medium between processor i and processor j is faulty, there are two possible cases. Note that $v(\alpha ij) = v(\alpha i)$ for a fault-free processor j.

Case 1: $v(\alpha i) = v(\alpha)$

If there are at most $\lfloor n/2 \rfloor$-1 faulty components connected by fault-free processor j, there are at most $\lfloor n/2 \rfloor$-1 values, [$v(\alpha i1)$, $v(\alpha i2)$,…, $v(\alpha ik)$,…, $v(\alpha in)$] for $k \neq j$, that may be $\neg v(\alpha)$'s in the set of the children of vertex αi. Since $v(\alpha ij) = v(\alpha)$, the number of $v(\alpha)$'s is $[(n\text{-}1)\text{-}(\lfloor n/2 \rfloor\text{-}1)]+1 = \lceil n/2 \rceil + 1$ in this set. Hence, the majority of the value at such set is $v(\alpha)$, or MAJ(αi)=v(α).

Case 2: $v(\alpha i)=\neg v(\alpha)$

There are at most $\lfloor n/2 \rfloor$-1 faulty components in the network. At r-th round, the number of $\neg v(\alpha)$'s isn't beyond $(\lfloor n/2 \rfloor\text{-}1)+1 = \lfloor n/2 \rfloor$ and the number of $v(\alpha)$'s is at least $[(n\text{-}1)\text{-}(\lfloor n/2 \rfloor\text{-}1)] = \lfloor n/2 \rfloor$. If n is an even number, $\lfloor n/2 \rfloor = \lceil n/2 \rceil$, then the number of $v(\alpha)$'s may be equal to the number of $\neg v(\alpha)$'s in the set [$v(\alpha i1)$, $v(\alpha i2)$,…, $v(\alpha ij)$,…, $v(\alpha in)$] and the majority for vertex αi is undefined. By definition, MAJ(αi)= $\neg v(\alpha i)= v(\alpha)$. On the other hand, if n is an odd number, then $\lfloor n/2 \rfloor < \lceil n/2 \rceil$, hence the majority value, MAJ(αi), in the set is $v(\alpha)$.

Corollary 1: At the end of the fourth round, all fault-free processors can get a correct ms-tree, in which, each non-leaf vertex is not influenced by the faulty transmission media.

Lemma 2: All correct vertices of the ms-tree are common.

Proof: In the decision making phase, there are no repeatable vertices in the ms-tree by deleting the repeating vertices. At the second and third level, the correct vertex α has at least n-1 children in which at least $n\text{-}\lfloor (n\text{-}1)/3 \rfloor$ children are correct after applying the function MAJ(α). The value of these $n\text{-}\lfloor (n\text{-}1)/3 \rfloor$ correct vertices are in common, and the majority value of vertex α is common. The correct vertex α is common in the ms-tree, if the level of α is less than three. Thus, all correct vertices of the ms-tree are common.

Lemma 3: The common frontier does exist in the ms-tree.

Proof: There are three vertices along each root-to-leaf path of the ms-tree in the decision making phase, in which the root is labeled by the source name, and the others are labeled by a sequence of group names. Because at most $\lfloor (n\text{-}1)/3 \rfloor$ processors can fail, at least one vertex is correct along each root-to-leaf path of the ms-tree. By lemma 2, the correct vertex is common, and the common frontier exists in each fault-free processor's ms-tree.

Lemma 4: Let α be a vertex, if there is a common frontier in the sub-tree rooted at α, then α is common.

Proof: By induction on the height of α.

If the height of α is 0 and the common frontier (α itself) exists, α is common.

If the height of α is l, the children of α are all in common, based on the induction hypothesis with the height of the children at l-1; therefore, vertex α is common.

Lemma 5: The reliable processor can be obtained.

Proof: Since there are $n\text{-}\lfloor (n\text{-}1)/3 \rfloor$ fault-free processors can send the received values to others correctly and honestly in a distributed system. Therefore, there will have at least $n\text{-}\lfloor (n\text{-}1)/3 \rfloor$ vertices (In the third level of the ms-tree) that have the same value for each sub-tree which is derived from the corresponding vertex in the second level of the ms-tree. For each sub-tree in the third level of the ms-tree, if $maj_3(ax) = v(ax)$ and $\# maj_3(ax) \geq n\text{-}\lfloor (n\text{-}1)/3 \rfloor - 1$, it means that processor x (The value $v(ax)$ is received from processor x.) sends to at least $n\text{-}\lfloor (n\text{-}1)/3 \rfloor$ same values to others and there have $n\text{-}\lfloor (n\text{-}1)/3 \rfloor$ processors y (The value $v(axy)$ is received from processor y.) agree that processor x is reliable. Thus, processor x and processor y (which has the same value with processor x) can be added to the RLP. If $\#p_z$ (Each processor z in the RLP) $\geq n\text{-}\lfloor (n\text{-}1)/3 \rfloor$, it presents that the number of processors agree processor z is reliable are in the majority. Hence, the reliable processor z can be obtained in our protocol.

Lemma 6: The values replaced by the majority value of reliable processors are common.

Proof: By Lemma 2, Lemma 3, Lemma 4 and Lemma 5, all correct vertices of the ms-tree are common, and each fault-free processor's ms-tree also has the same common frontier. Furthermore, at least $n\text{-}\lfloor (n\text{-}1)/3 \rfloor$ processors are correct. Hence, all these fault-free processors must be reliable processors. Thus, the majority value of these reliable processors for each sub-tree in the third level must be common. Therefore, the values replaced by the majority value of reliable processors are common.

Corollary 2: If the common frontier exists in the ms-tree, then the root is common.

Theorem 1: All fault-free processors can determine the common set of reliable processors.

Proof: By Lemma 1, Lemma 2, Lemma 3, Lemma 4, Lemma 5, Corollary 1 and Corollary 2, the theorem is proven.

Theorem 2: The root of a fault-free processor's ms-tree is common.

Proof: By Lemma 1, Lemma 2, Lemma 3, Lemma 4, Lemma 5, Lemma 6, Corollary 1, Corollary 2 and Theorem 1, the theorem is proven.

Theorem 3: NGBA can solve the BA problem.

Proof: To prove the theorem, we show that NGBA can meet the agreements (Agreement') and (Validity').

(Agreement'): Root s is common.

By theorem 2, (Agreement') is satisfied.

(Validity'): VOTE(s) = v_s for all fault-free processors, if the source is correct.

If the source is correct, then it broadcasts the same initial value vs to all processors. The value of correct vertices for all fault-free processors' ms-tree is v_s. Thus, each correct vertex of the ms-tree is common (Lemma 2), and its value is v_s. Because the source is correct, the root of the ms-tree is also a correct vertex by lemma 2. By Theorem 2, this root is common. The computed value vote(s) = v_s is stored in the root for all fault-free processors. Thus, (Validity') is satisfied.

The complexity of the protocol is evaluated in terms of 1) the number of rounds about message exchange, 2) the number of allowable faulty processors and 3) the quantity of the messages that are generated during the execution of the NGBA protocol.

Theorem 4 describes the number of required rounds of message exchange and the fault tolerance capability of the protocol. Theorem 5 and Theorem 6 show that the protocol solves the BA problem, using four rounds of message

exchange and the maximum number of allowable faulty processors, respectively. Theorem 7 proves the complexity of NGBA

Lemma 7: The values sent by the fault-free processors are same as the majority value after applying the VOTE function.

Proof: There are at least n-$\lfloor(n-1)/3\rfloor$ fault-free processors in the distributed system. All fault-free processors transmit their values to the others correctly. In each round of message exchange n-$\lfloor(n-1)/3\rfloor$ fault-free processors can receive these values and send them again. Then, the majority values which are applied, using the VOTE function for the (i+1)th ($1 \le i \le \lfloor(n-1)/3\rfloor$) level of the ms-tree must be equal to the values in the ith level of the ms-tree. It is needless to send the values for $\lfloor(n-1)/3\rfloor + 2$ rounds, if the sender is a fault-free processor.

Theorem 4: NGBA requires four rounds of message exchanges to solve the BA problem in a distributed system.

Proof: 1) In the second round of the message exchange phase, the values are sent and received correctly by other n-$\lfloor(n-1)/3\rfloor$ fault-free processors, if the sending processors are correct. All these correct values are sent in the four rounds of the message exchange phase after applying the function MAJ(α). Then, these three (The values in the fourth level of the ms-tree are deleted) level ms-tree can be used to find the reliable processors. Here, they have n-1 number of vertices in the second level of the ms-tree. Thus, there will have n-1 number of RLP_x ($1 <= x <= (n-1)$). If the frequency of the processor x appearing in all RLP is greater than or equal to n-$\lfloor(n-1)/3\rfloor$, it implies that n-$\lfloor(n-1)/3\rfloor$ processors believe that processor x is reliable. Hence, the values sent from the reliable processors can be used to replace the values received from the un-reliable processors. Based on lemma 7, there is no need to send the values for $\lfloor(n-1)/3\rfloor + 2$ rounds if the sender is a fault-free processors. Furthermore, based on Lemma 5, and Theorem 3, the values replaced by the majority value of reliable processors are common, and the protocol can solve the BA problem. Hence, four rounds of message exchange can also solve the BA problem in a distributed system.

2) By Lemma 1, the influence caused by at most $\lfloor n/2\rfloor$-1 faulty transmission media can be resolved by function MAJ(α) while the ms-tree is established. In addition, by the lemma 2, the ambiguity due to at most $\lfloor(n-1)/3\rfloor$ faulty processors can be resolved. Hence, the theorem is proven.

Theorem 5: NGBA can solve BA problem by using four rounds of message exchanges and it is minimum.

Proof: Basically, one round of message exchange is not enough to determine the agreement value, because the source processor may be faulty. The source faulty processor may send 0's and 1's to others with the same frequency. It is impossible to determine the agreement value within two rounds of message exchange since the source processor may send equal amount of 0's and 1's to all processors in the distributed system. And the fault-free processor may not get a common value under such a circumstance. Furthermore, NGBA can find the reliable processors by comparing the messages in the third level of the ms-tree without regard to the number of processors, and can help all fault-free processors to reach an agreement when only processors fault exist. However, the values in the third round of message exchange still influenced by the faulty transmission media. Thus, it is impossible that the number of rounds required is three. Furthermore, by the result of Yan and Chin [15], two rounds of message exchange is the minimum number of rounds to solve the BA problem if all the processors are correct. In other words, all processors can remove the influences caused by faulty transmission media by using function MAJ(α) with every two rounds of message. Here, in order to remove the influence caused by faulty transmission media, all fault-free processors need to run one more rounds of message exchange to get the correct value for the third levels of the ms-tree. After finish four rounds of message exchange, all fault-free processors must delete the values in the fourth level of the ms-tree, and then all the reliable processors can be found out. The majority values of the reliable processors can be used to replace the values which are received from the un-reliable processors. After that, each processor can obtain the final agreement by using the VOTE function. Thus, four rounds is the minimum number of rounds required to solve the BA problem in the generalized fault assumption in which both the processors and the transmission media may be faulty.

Theorem 6: The total number of allowable faulty components ($\le \lfloor n/2\rfloor$-1) of the protocol NGBA is maximal to solve BA problem in the generalized faulty assumption. And the number of allowable faulty processors is $\lfloor(n-1)/3\rfloor$ in NGBA protocol, which is maximum.

Proof: In [15], Yan and Chin have pointed out that the allowable number of faulty transmission media ($\le \lfloor n/2\rfloor$-1) is the maximum number of allowable faulty transmission media to solve the BA problem in a transmission media failure system. Now, we use f_t to represent the total number of the faulty components, f_p to represent the total number of faulty processor, and f_l to represent the total number of the faulty transmission media. Here, f_t must be equal to $f_p + f_l$, and f_t must be less than or equal to $\lfloor n/2\rfloor$-1. Here, $f_t = f_p + f_l = 0 + f_l$. and f_t will be greater than or equal to $\lfloor n/2\rfloor$ when all processors are correct, if $\lfloor n/2\rfloor$-1 is not the maximum number of allowable faulty components. Then $f_l \ge \lfloor n/2\rfloor$ is a contradiction.

Furthermore, if the faulty processors are greater than $\lceil n/2\rceil$, then all faulty processors may send different values to each processor. Fault-free processors cannot get the common vertices or frontier. Thus, it cannot be sure that all fault-free processors can reach agreement. Furthermore, according to the constraints of the BA problem [1], the total allowable component is processor only, and the faulty processors cannot exceed $\lfloor(n-1)/3\rfloor$ when all the transmission media are correct. If the total number of faulty processors exceeds $\lfloor(n-1)/3\rfloor$, all fault-free processors cannot get a common value. Thus, the total number of allowable faulty processors is $\lfloor(n-1)/3\rfloor$ in NGBA.

To sum up, in order to make all fault-free processors to get a common value, to total number of faulty components must be less than or equal to $\lfloor n/2\rfloor$-1, in which the total number of faulty processors must be less than or equal to $\lfloor(n-1)/3\rfloor$.

Theorem 7: The message complexity is O(n^3).

Proof: In the first round of the message exchange phase, the source processor will send its initial value to others. Hence, one message must be generated. In the second rounds of the message exchange phase, all processors must send the received value in the first round of message exchange to others, and then send the value applied by the function MAJ(α). Hence, there will have 2**n* messages been generated. In the third rounds, 2(*n***n*) messages must be generated. There will have 2(*n***n***n*) messages been exchanged. Therefore, the total quantity of messages to be generated during the execution of NGBA is (1 + 2*n* + 2(*n***n*) +2(*n***n***n*)). The message complexity is O(n^3).

IV. Conclusion

To ensure that all fault-free processors reach agreement and perform corresponding actions in distributed systems is an important research topic. In previous studies [5], [15], [16], each fault-free processor could reach an agreement and tolerate $\lfloor n/2 \rfloor$-1 faulty components in which the number of faulty processors is less than or equal to $\lfloor (n-1)/3 \rfloor$ by using $\lfloor (n-1)/3 \rfloor + 2$ rounds of message exchange continuously even if the processor faults and transmission media faults exist simultaneously. However, it is unsuitable to some network topologies which have a large number of processors, such as a P2P, wireless and sensor networks.

In this study, we propose a novel agreement protocol, the NGBA protocol. Here, the NGBA protocol revisits the features that there will always have at least $n-\lfloor (n-1)/3 \rfloor$ fault-free processors and these fault-free processors will always sent the received values correctly and honestly. Furthermore, the total number of values sent by the fault-free processors will greater than the total number of values sent by the faulty processors. Based on these two features, NGBA can find out the reliable processors by comparing and counting the values in the second and third level of the ms-tree. After that, the majority values of the reliable processors can be used to replace the value sent by the un-reliable processors. This can help to reduce the influence caused by the faulty-processors. Hence, NGBA requires only four rounds of message exchange to make each fault-free processor reach agreement in a distributed system where the processor faults and transmission media faults exist. Besides, our protocol can reduce the complexity of message to O(n^3). With less rounds of message exchange, the generated message during executing the protocol can be reduced, too. This can help each processor reduce the storage to store the message. Therefore, NGBA is more suitable than previous works [5], [15], [16] to allow all fault-free processors to reach agreement, especially for the network systems, which have large numbers of processors.

Reference

[1] L. Lamport *et al*., "The Byzantine generals problem," *ACM Trans. Programming Languages and Systems*, vol. 4, no. 3, pp. 382-401, 1982.

[2] A. Bar-Noy *et al*., "Shifting gears: changing algorithms on the fly to expedite Byzantine agreement," in *Proc. Symp. on Principles of Distributed Computing*, 1987, pp. 42-51.

[3] D. Dolev, "Unanimity in an unknown and unreliable environment," *IEEE Foundations of Computer Science*, pp. 159-168, 1981.

[4] G. Pieri, J. da Silva Fraga, and Lau Cheuk Lung, "Consensus service to solve agreement prob," in *Proc. IEEE 16th International Conference on Parallel and Distributed Systems (ICPADS)*, 2010, pp. 267-274.

[5] H. S. Siu, Y. H. Chin, and W. P. Yang, "Byzantine agreement in the presence of mixed faults on processors and links," *IEEE Transactions on Parallel and Distributed Systems*, vol. 9, no. 4, pp. 335-345, 1998.

[6] H. Strong *et al*., "Byzantine agreement," *IBM Res. Rep. RJ-3714*, 1982.

[7] L. Lamport and P. Melliar-Smith, "Byzantine clock synchronization," in *Proc. ACM 3rd PODC*, 1984, pp. 10-16.

[8] M. Pease, R. Shostak, and L. Lamport, "Reaching agreement in presence of faults," *J. Ass. Compute. Mach.*, vol. 27, no. 2, pp. 228-234, 1980.

[9] R. Reischuk, "A new solution for the Byzantine generals problem," *IBM Res. Rep. RJ-3673*, 1982.

[10] R. Turpin and B. Coan, "Extending binary Byzantine agreement to multivalued Byzantine agreement," *Information Processing Letters*, vol. 18, no. 2, pp. 73-76, 1984.

[11] S. C. Wang, K. Q. Yan, and H. C. Hsieh, "The new territory of mobile agreement," in *Computer Standards & Interfaces*, vol. 26, no. 55, pp. 435-447, 2004.

[12] V. K. Garg and J. Bridgman, "The weighted byzantine agreement problem," in *Proc. 2011 IEEE International Parallel & Distributed Processing Symposium*, 2011, pp. 524-531.

[13] V. King and J. Saia, "Breaking the O(n^2) bit barrier: Scalable byzantine agreement with an adaptive adversary," *Journal of the ACM*, vol. 58, no. 4, 2011.

[14] M. Fischera and N. Lynch, "A lower bound for the assure interactive consistency," *Information Processing Letters*, vol. 14, no. 4, 1982.

[15] K. Q. Yan, Y. H. Chin, S. C. Wang, "Optimal agreement protocol in malicious faulty processors and faulty links," *IEEE Transactions on Knowledge and Data Engineering*, vol. 4, no. 3, pp. 266-280, 1992.

[16] K. Q. Yan, S. C. Wang, and S. S. Wang, "An optimal solution of byzantine agreement in a scale free network," in *Proc. IEEE 22nd International Conference on Advanced Information Networking and Applications*, Okinawa, Japan, March 25-28, 2008.

Hui-Ching Hsieh received her BS and MS degrees in information management from Chaoyang University of Technology, Taiwan in 2002 and 2004 respectively. She got her Ph.D. degree in computer science at National Tsing Hua University in Taiwan in 2010. Her research interests include distributed data processing, fault tolerant computing, and P2P network computing.

Mao-Lun Chiang received the M.S. degree in information management from Chaoyang University of Technology and the Ph.D. degree in the Department of Computer Science from National Chung-Hsing University, Taiwan. He is an assistant professor in the Department of Information and Communication Engineering at the Chaoyang University of Technology, Taiwan. His current research interests include Ad Hoc, mobile computing, distributed data processing, fault tolerant computing, and cloud computing.