Pakistan Journal of Information and Technology 2 (2): 104-115, 2003
ISSN 1682-6027

Byzantine Agreement under Unreliable Multicasting Network

S.C. Wang, [1]K.Q. Yan and [2]C.F. Cheng
Department of Information Management, [1]Department of Business Administration,
[2]Department of Computer Science and Information Engineering,
Chaoyang University of Technology,168 Gifeng E. Rd.,
Wufeng, Taichung County, Taiwan 413, R.O.C.

Abstract: In practice, the communication media in the network are fallible. However, in previous results of the Byzantine Agreement (BA), most of the researches are focus on the fallible component in the network is processor only. This treatment would make an innocent processor would not be able to reach a common agreement due to its faulty communication media. In this paper, we revisit the BA problem in an unreliable MultiCasting Network (MCN) and we also enlarge the fault-tolerant capability by allowing dormant faults and malicious faults to exist in an MCN. The proposed protocols can make each fault-free processor reach a common agreement value by only one round of message exchange and can tolerate the maximum number of faulty communication media.

Key words: Byzantine agreement, parallel processing, dormant fault, malicious fault, dual failure mode, multicasting network

Introduction

In order to provide the reliable distributed computing environment to cope with the faulty components, Byzantine Agreement (BA) is one of the fundamental problems to reach a common agreement in presence of faults Silberschatz (2002). Such an agreement problem was first studied by Pease, Shostak and Lamport (1980). The definitions of the BA problem by Pease, Shostak and Lamport (1980) are shown as follows: There are n (n=4) processors in a distributed system and there is an initial value needs to be set in one of the processors which called as the source processor. The source processor can send this initial value v_s to the other processors through the reliable Fully Connected Network (FCN). When the other processors receive the initial value from the source processor, they can exchange their message(s) with each other to get enough information. There are at most one-third of the total number of processors could fail, so a faulty processor may send incorrect message(s) to other processors. After t+1 ($t=\sqrt{[(n-1)/3N]}$) rounds of message exchange, a common agreement can be reached among all the fault-free processors. That is, the BA is achieved when the following constraints are satisfied.

Agreement: All fault-free processors agree on the same common agreement value v_s,
Validity: If the source processor is fault-free, then the common agreement should be the initial value v_s of the source processor.

In practice, the fallible component in the network would not be processor only; the communication media are also fallible. Treating a communication medium fault as a processor fault violates the definition of BA, due to the innocent processor (A fault-free processor is treated as faulty processor due to its faulty communication media) will be excluded from the common agreement. Moreover, the reliability of the connection state of the network is also an important topic in designing the distributed system. We can transmit the message correctly and on time when the connection state of the network is stable. Otherwise, the message from faulty communication media would be not able to arrive on time and getting changed.

Furthermore, most of the network structure may not be fully connected and the network structure may have the feature of grouping. Hence, in this study, we will revisit the BA problem with fallible communication media in a MultiCasting Network (MCN), which is the most practical network structure in the real world. In Section 2, we will describe the various network structures more detail.

In this study, the BA problem is to be solved in an MCN with fallible communication media. The proposed protocols called Efficient Multicasting Agreement Protocol for Communication Media ($EMAP_{cm}$) and Relay Fault-tolerance Channel for Communication Media (RFC_{cm}) can solve the BA problem by only one round of message exchange and can tolerate a maximum number of allowable faulty communication media.

The rest of this paper is organized as follows. Section 2 will serve to introduce the conditions for BA problem with fallible communication media. Then, our new protocols will be brought up and illustrated in detail in Section 3. In Section 4, gives an example of executing the proposed protocols. Section 5 is responsible for proving the correctness and complexity of our new protocols. Finally, in Section 6, we shall come to the conclusion.

The conditions for byzantine agreement

Before the BA problem can be solved, some assumptions and definitions need to be made. They are the failure types of communication medium, the network structure, parameters and constraint. The BA problem is considered in a synchronous network in which the bounds on processing time and communication delays of each fault-free component are finite (2002). Fischer, Paterson and Lynch (1985) also report that it is impossible to reach a common agreement in the network even if only one crash faulty component. The reason is that each fault-free processor would not give up to wait for the message from the crash faulty component.

The failure types of communication medium

The symptoms of a faulty communication medium can be divided into two types. They are dormant fault (include crashes and stuck-at) and malicious fault.

The symptoms of dormant fault on communication medium

The symptoms of a faulty communication medium may be dormant faults (both crash and stuck-at). A crash fault happens when a communication medium is broken. A stuck-at fault takes place when the message received from a certain communication medium is always a constant

value. In the synchronous system, each fault-free processor can detect the messages from dormant faulty communication medium, if the protocol appropriately encodes a transmitted message by Manchester code (1995) before transmission.

The symptoms of malicious fault on communication medium

A communication medium with the malicious fault (also called as the Byzantine fault and arbitrary fault) is one whose behavior is arbitrary and unrestricted. It is also the most damaging failure type of all and causes the worst problem. If a common agreement can be reached at the presence of a malicious fault, then a common agreement in the other failure modes can also be reached. That is, a fault-free communication medium can transmit messages on time and correctly, but the message which is transmitted by a faulty communication medium may be changed or delayed.

Network structure

There are many kinds of network structures. They can be classified into either group of FCN, BCN, GCN by Wang (1995) and GCN by Siu (1996). However, each network has its own drawbacks due to the hardware structure. Large numbers of I/O ports and communication media make the FCN inapplicable; however, in the BCN, the traffic of message exchange on the bus is too heavy when there are hundreds of processors or more in it. As for the GCN by Wang (1995), its major drawbacks are that the number of processors in each group must be the same and the connection state of each group must be fully connected. The major drawback for the GCN by Siu (1996) is that it does not have the feature of grouping. And the network structure of FCN, BCN and the GCN are all special cases of the MCN.

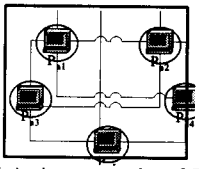
- A 5-processor FCN model can be seen as five groups in an MCN model with the connectivity of each processor being 4, as shown in Fig. 1(a).
- A 5-processor BCN model can be seen as a one-group MCN model and the processors in the BCN communicate with each other through the local bus, as shown in Fig. 1(b).
- A 10-processor GCN by Wang can be seen as a five-group MCN model with two processors in each group with the connectivity of each group being 4, as shown in Fig. 1(c).
- A 6-processor 2-connectivity GCN by Siu can be seen as a six-group MCN model with one processor in each group with the connectivity of each group being 2, as shown in Fig. 1(d).

That is, the MCN is a more generalized network. An example of the MCN is shown in Fig. 1(e). It has the following features:

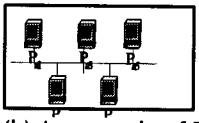
Grouping: A local bus links the processors of a same group. The number of groups (denoted as g) may be varied in the system, say $1=g=n$, where n stands for the total number of processors in the MCN.

Group member: The number of processors μ_i in group i can be different from each other, $\mu_1+\mu_2+YY\mu_g=n$ where $1=\mu_i=n$.

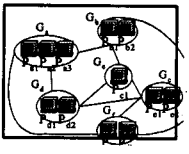
Connectivity: It allows the MCN a bounded connectivity c, where c is a constant and 1□c□g-1. Each group communicates with each ,other through the communication media.



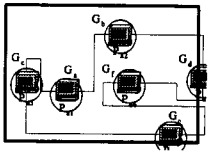
(a) An example of FCN medel



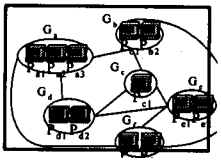
(b) An example of BCN



(c)An example of GCN by Wang



(d) An exmple of 2-connectivity



(e) An example of MCN model

Fig. 1: Example of network model FCN, BCN, GCN and MCN

The assumptions and parameters

The assumptions and parameters of our protocols are listed as follows:

- The underlying network is synchronous.
- Let N be the set of all processors in the network and |N|= n, where n is the number of processors in the underlying network.
- Each processor in the network can be identified uniquely.
- Let G be the set of all groups in the network and |G|= g, where g is the number of groups in the underlying network and g□4.
- Each group in the network can be identified.
- The processors of the underlying network are assumed to be fault-free.
- The communication media of the underlying network are assumed to be fallible.
- A processor that transmits messages is called a sender processor.
- The transmission protocol RFC_{cm} encodes a transmitted message by Manchester code before transmission.
- There is only one source processor who transmits the message at the first round in the BA problem.
- Let C_m be the maximum number of malicious faulty communication media allowed.
- Let C_d be the maximum number of dormant faulty communication media allowed.
- Let c be the connectivity of the MCN.
- A processor does not know the faulty status of communication media, while the message(s) from dormant faulty communication media can be detected.

The constraint

The number of faulty components can be allowed in the network is determined by the connectivity of the network topology. In this paper, $EMAP_{cm}$ is considered in the MCN with fallible communication media. So that, the constraint with the connectivity of an MCN is based on the number of malicious faulty communication media C_m and the number of dormant faulty communication media C_d. In addition, the total number of malicious faulty groups must be smaller than half of $c\text{-}G_d$. Hence the constraint of the connectivity of an MCN is $c > 2C_m+C_d$.

The approaches

In this section, the proposed protocols Efficient Multicasting Agreement Protocol for Communication Media ($EMAP_{cm}$) and Relay Fault-tolerance Channel for Communication Media (RFC_{cm}) are introduced to solve the BA problem by one round of message exchange in an unreliable MCN.

The transmit protocol: relay fault-tolerance channel for processor (RFC_{CM})

In $EMAP_{cm}$, RFC_{cm} is used to transmit messages. RFC_{cm} can provide a virtual channel to help each processor to transmit message to each other without influence from faulty inter-medium in an un-fully connect network. The idea of our transmission protocol RFC_{cm} comes from the concept of virtual link by Meyer and Pradhan (1991), which is a concept to let a reliable (without faulty communication media) un-fully connected network just like a fully connected network. The definition of our RFC_{cm} is shown in Fig. 2. The functions of RFC_{cm} are shown as follows:

RFC_{cm} can let an un-fully connected network work just like a fully connected network

In general, each processor has the common knowledge of graphic information of the underlying network and there are at least c disjoint paths between sender processor and destination processor if the connectivity of the network is c [Deo, 1974; Meyer and Pradhan, 1991). Therefore, each processor in the network can transmit its message(s) to other processors through the relay processors (the processors between the sender processor and destination processor). That is, virtual link can let an un-fully connected network work just like a fully connected network by relay.

RFC_{cm} can remove the influence from faulty inter-medium (communication media)

The message(s) from the dormant faulty communication media can be detected by fault-free processor if RFC_{cm} encodes a transmitted message by Manchester code before transmission (Halsall, 1995). And, we can remove the influences from the malicious faulty communication media between any pairs of sender processors and destination processors in each round of message exchange if $c>2C_m+ C_d$. The reason is that the fault-free sender processor sends c copies of a message to fault-free destination processors. In the worst case, a fault-free destination processor can receive $c\text{-}C_d$ messages transmitted by the fault-free sender processor. So that, a fault-free destination processor can decide which the correct messages are by taking the majority value.

Definition

- Each processor has the common knowledge of graphic information G=(E,G), where,G is the set of groups in the network and E is a set of group pairs (G_i,G_j) indicating a communication medium between group G_i and group G_j,where $1 \le i,j \le n$.
- There are c ($c>2C_m+ C_d$) paths from sender processor to destination group.
- The c disjoint paths between the sender processor to destination group can be predefined [3][8].
- These c paths from sender processor to destination group are group-disjoint paths [3].
- Each intermediate group on these c paths should not be passed through more than once[3].

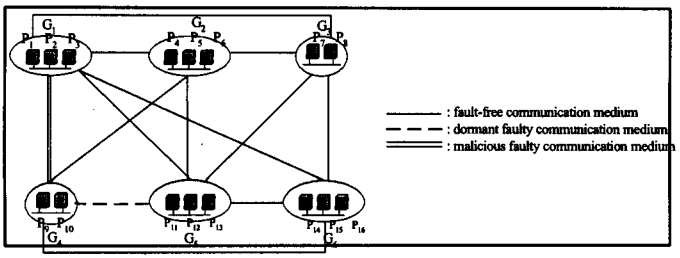
Message exchange

- The source processor P_s transmits initial v_s to the destination group through c group-disjoint paths.
- If a group-disjoint path from sender processor to destination group passes through any dormant faulty communication medium, then replace λ^0.
- The processors in the destination group take the majority value from the same group-disjoint paths and then construct the vector $V_i = [v_{path\ 1}, v_{path\ 2}, Y, v_{path\ c-1}, Y, v_{path\ c}]$, for $c>2C_m+C_d$.

Fig. 2: The proposed protocol RFC_{cm}

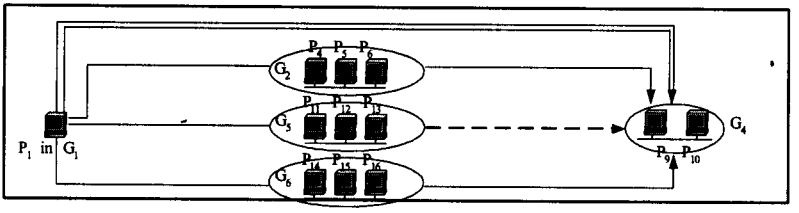
An example of executing RFC_{cm}

An example of executing RFC_{cm} is shown in Fig. 3. Fig. 3(a) illustrates a 4-connectivity MCN model with six groups where the number of processors in a group is two or three and Fig. 3(b) illustrates the sender processor P_1 in G_1 using RFC_{cm} to transmit message(s) to destination group G_4=s processors. There are four adjacent paths to G_4, so G_4=s processors can receive four values from the sender processor. In Fig. 3(b), processor P_1 can acquire these four values by using RFC_{cm} to transmit messages to G_4 directly, through G_2 to G_4, through G_5 to G_4 and through G_6 to G_4. Processors P_9 and processor P_{10} in G_4 can receive the vector $V_1= [v_1, v_2, v_5, v_6]$.



(a) An example of MCN (n=16,g=6,p={2,3} and c=4)

Fig. 3: An example of executing RFC_{cm} (Cont=d.)



(b) The source processor P_1 in G_1 uses RFC_{cm} transmit messages to destination group G_4's processors. The destination group G_4 with processor P_9 and processor P_{10} receives the vector $[V_1=v_1, v_2, v_5, v_6]$

Fig. 3: An example of executing RFC_{cm}

The BA protocol efficient multicasting agreement protocol for communication media ($EMAP_{cm}$)

There are two phases in the $EMAP_{cm}$, there are the message exchange phase and the decision making phase. In the message exchange phase, the source processor transmits its initial value to each other processors. To avoid the case where part of some messages being transmitted gets forged by some faulty communication media, the decision value must be dominated. Therefore, we use RFC_{cm} to transmit message.

In the decision making phase, each processor P_i takes the majority value in the vector V_i as the decision value. The detailed definition of protocol $EMAP_{cm}$ is shown in Fig. 4.

Protocol $EMAP_{cm}$

Message Exchange Phase

Round 1: The source processor uses RFC_{cm} to transmit v_s to all processors, then each processor P_i receives c values of v_s to construct vector $V_i = [v_{path\ 1}, v_{path\ 2}, Y, v_{path\ c-1}, Y, v_{path\ c}]$, where $1 \le i \le n$.

Decision Making Phase

Step 1 Eliminate all λ^0=s to lessen the influence of faulty behavior and take the majority value of vector V_i

Step 2: The decision value DEC_i =majority value of vector V_i.

Fig. 4. The proposed protocol $EMAP_{cm}$

An example of reaching byzantine agreement

Here is an example of how our $EMAP_{cm}$ and RFC_{cm} are applied. Assume the initial value of the source processor is assumed as 0. An MCN is shown in Fig. 5(a), where there is no connectivity between G_1 and G_6, between G_2 and G_4, as well as between G_3 and G_5. That is, the connectivity c is 4. There is a dormant communication medium on the way from G_3 to G_4 and a malicious communication medium on the way from G_1 to G_3. The source processor is assumed as processor P_1.

In the message exchange phase, the source processor P_1 transmits its initial value v_1 to the other groups through c neighboring paths by RFC_{cm}. For example, the source processor P_1 in G_1 uses RFC_{cm} to transmit message to the destination group G_2. Processor P_1 transmits its initial value v_1 to G_2 through c group-disjoint paths and these c group-disjoint paths from G_1 to G_2 are: from G_1 to G_2 directly, from G_1 to G_2 through G_3, from G_1 to G_2 through G_5, from G_1 to G_2 through G_4 and G_6. Then, the destination group G_2=s processors get c copies of the messages from the source processor P_1 in G_1.

After the message exchange, each processor P_i constructs the vector V in Fig. 5(b). In the message exchange phase, each processor in the same group will construct the same vector so that $V_1=V_2=V_3$, $V_4=V_5=V_6$ and so on. For example, processor P_7 in G_3 receives messages from 4 group-disjoint paths. From path 1, processor P_7 receives processor P_1=s value directly, but the communication medium between G_1 and G_3 is in malicious fault and so the value may be incorrect. From path 2, processor P_7 receives processor P ₹s value 0 through G . From path 3, processor P_7 receives processor P_1=s value through G_4, but the communication medium between G_3 and G_4 is in dormant fault, so the value is also not correct but can be detected and so we replace λ^0. From path 4, processor P_7 receives processor P_1=s value 0 through G_5 and G_6 and so on to receive values from other processors.

In the decision making phase, the decision value is the majority value of vector V_i as shown in Fig. 5(c). Finally, the decision value of this example is value 0, so that the BA is reached.

The correctness and complexity

The following lemmas and theorems are used to prove the correctness and complexity of RFC_{cm} and $EMAP_{cm}$. It tolerates C_m malicious faults and C_d dormant faults simultaneously in an MCN, where $c>2C_m+C_d$ and costs only one round of message exchange to reach BA. For all the processors in the same group, the received messages are the same due to multicasting. That is, if a processor achieves BA, the other processors in the same group certainly reach the same BA. Therefore, we only need to discuss one processor in a group.

Lemma CM-1: Let the initial value of a sender processor P_i is v_i. By using RFC_{cm}, the destination group=s processors can receive the value v_i from sender processor P_i if $c>2C_m+C_d$.

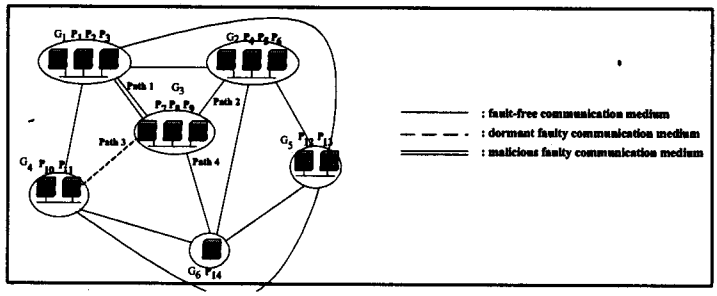
Proof

By using RFC_{cm}, the sender processor can transmit c copies of the same value to the destination group=s processors through c group-disjoint paths. According to the assumption of $c>2C_m+C_d$, the processors in the destination group, in the worst case, can get $c-C_d$ copies of the value from the sender processor. Since $c- C_d >2C_m$, we can take the majority value on these $c-C_d$ values and let each of the processors in the destination group get the value v_i.

Lemma CM -2: The decision value DEC_i =majority value.

Proof

By the definition of the BA problem, Lemma CM-2 is proven.



(a) An Example of MCN (n=14, g=6, p={1,2,3} and c=4)

Fig. 5: An example of executing $EMAP_{cm}$ and RFC_{cm} (Cont=d.)

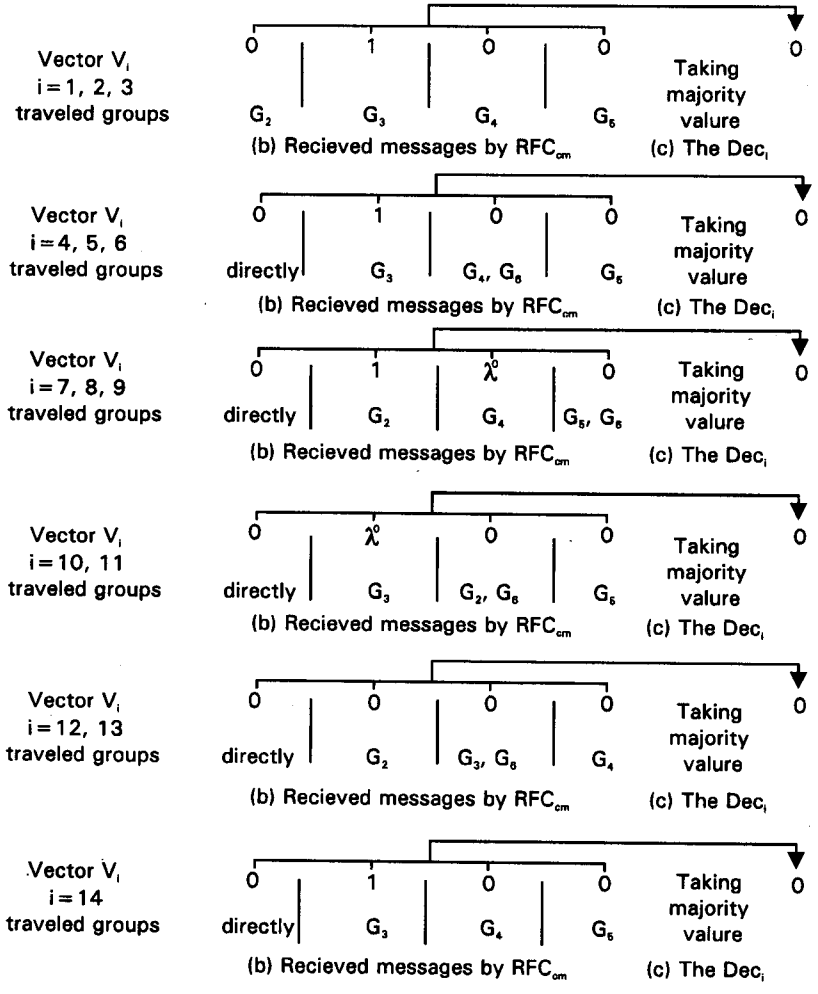


Fig. 5: An example of executing $EMAP_{cm}$ and RFC_{cm}

Table 1: The constraints for various protocols on communication media failures

Network topology	Failure types: Malicious Faults FCN	GCN	MCN	Dual Failure Faults FCN	GCN	MCN
Yan *et al.* [15]	$C_m \leq T(n-3)/2Z$	N.A.	N.A.	$C_m \leq T(n- C_d -3)/2Z$	N.A.	N.A.
Wang *et al.*[12]	$C_m \leq T(c-2)/2Z$	$C_m \leq T(c-2)/2Z$	N.A.	$C_m \leq T(c- C_d -2)/2Z$	$C_m \leq T(c- C_d -2)/2Z$	N.A.
$EMAP_{cm}$	$c>2C_m$	$c>2C_m$	$c>2C_m$	$c>2C_m + C_d$	$c>2\ C_m + C_d$	$c>2\ C_m + C_d$

N.A.= Not Applicable

Table 2: The number of rounds of message exchange for various protocols

	Number of rounds of message exchange: FCN	GCN	MCN
Yan *et al.* [15]	2	N.A.	N.A.
Wang *et al.* [12]	2	2	N.A.
$EMAP_{cm}$	1	1	1

Theorem CM -1: Protocol $EMAP_{cm}$ is valid.

Proof

According to Lemmas CM-1 and CM-2, the validity of $EMAP_{cm}$ is confirmed.

Theorem CM -2: Protocol $EMAP_{cm}$ can reach a BA.

Proof

If a processor agrees on value x0{ v_i (for $1 \leq i \leq n$)}, by Lemma CM-2, there is such a most common value k (for $1 \leq k \leq n$) in each processor and therefore all processors should agree on value x.

Theorem CM -3: The maximum amount of information exchange by $EMAP_{cm}$ is cn.

Proof

In the message exchange phase, we only use one round to exchange messages when source processor sends out at most c*n copies of its initial value to the other processors, where c is a constant and n is the number of processors in the MCN. Therefore, the total number of message exchanges is at most (cn).

Most of the previous results are focus on the fallible component in the network is processor only (Bar-Noy *et al.*, 1987; Barborak *et al.*, 1993; Fisher and Lynch, 1982; Fischer *et al.*, 1985; Lamport *et al.*, 1982; Meyer and Pradhan, 1991; Pease and Lamport, 1980; Siu *et al.*, 1996). Hence, the innocent processors would be treated as faulty due to its faulty communication media. Only few of previous results are focus on the fallible component in the network is communication medium (Wang et al., 2000; Yan et al., 1999). For example, in Wang et al. (2000) protocol, they solve the consensus problem with malicious faulty communication media in the FCN by two round of message exchange. However, in practice, the network structure would not be fully

connected and the failure type of the fallible component is not malicious only, but also the dormant faults.

Therefore, we revisit the BA problem in an MCN with fallible communication media. Since FCN, BCN, GCN are all special cases of the MCN. So that, the proposed protocols $EMAP_{cm}$ and RFC_{cm} can solve the BA problem with fallible communication media in the FCN, BCN, GCN and MCN. Due to the RFC_{cm} can let an un-fully connected network work just like a fully connected network and remove the influence from dormant and malicious faulty components between the sender processor and receiver processor. So that, the number of round of message exchange required by $EMAP_{cm}$ is only one and $EMAP_{cm}$ can tolerate the maximum number of faulty communication media by allowing the dormant fault and malicious faults. We also show the constraints for various protocols on communication media failures in Table 1 and the number of rounds of message exchange for various protocols in Table 2.

Acknowledgments

This work was partially supported by the National Science Council of the Republic of China under Grant No. NSC 90-2213-E-324-017.

References

Bar-Noy, A. *et al.*, 1987. Shifting gears: changing algorithms on the fly to expedite byzantine agreement. in proceedings of the symposium on principles of distributed computing, pp: 42-51.

Barborak M., M. Malek and A. Dahubra, 1993. The consensus problem in fault-tolerant computing, ACM Computing Surveys, 25: 171-220.

Deo, N., 1974. Graph Theory with Applications to Engineering and Computer Science, Englewood Cliffs, N.J.: Prentice-Hall.

Fisher, M. and N. Lynch, 1982. A lower bound for the assure interactive consistency, Information Processing Letters, 14: 183-186.

Fischer, M., M. Paterson and N. Lynch, 1985. Impossibility of distributed consensus with one faulty process, Journal of ACM, 32: 374-382.

Halsall, F., 1995. Data Links, Computer Networks and Open Systems, 4th. Ed., Addison-Wesley Publishers Ltd., Ch., 3: 112-125.

Lamport, L., R. Shostak and M. Pease, 1982. The Byzantine generals problem, ACM Transactions on Programming Languages and Systems, 4: 382-401.

Meyer, F.J. and D.K. Pradhan, 1991. Consensus with dual failure modes, IEEE Trans. Parallel and Distributed Systems, 2: 214-222.

Pease, M., R. Shostak and L. Lamport, 1980. Reaching agreement in the presence of faults, Journal of ACM, 27: 228-234.

Silberschatz, A., P.B. Galvin, G. Gagne, 2002. Operating System Concepts, 6th. Ed., John Wiley & Sons, Inc.

Siu, H.S., Y.H. Chin, W.P. Yang, 1996. A note on consensus on dual failure modes, IEEE Trans. on Parallel and Distributed Systems, 7: 225-229.

Wang, S.C., K.Q. Yan, S.H. Kao and L.Y. Tseng, 2000. Consensus with dual link failure modes on a generalized network, in CY Journal, ISSN 1026-244X, pp: 35-52.

Wang, S.C., Y.H. Chin and K.Q. Yan, 1995. Byzantine agreement in a generalized connected network, IEEE Trans. on Parallel and Distributed System, 6: 420-427.

Yan, K.Q., Y.H. Chin and S.C. Wang, 1992. Optimal agreement protocol in malicious faulty processors and faulty links, IEEE Trans. Knowledge and Data Engineering, 4: 266-280.

Yan, K.Q., S.C. Wang and Y.H. Chin, 1999. Consensus under unreliable transmission, Information Processing Letters, 69: 243-248.