The Bounds of Faulty Components on Consensus with Dual Failure Modes

Kuo-Qin Yan[1]   Shu-Ching Wang[2]

[1]Department of Business Administration, [2]Department of Information Management,
Chaoyang University of Technology
Email: scwang@mail.cyut.edu.tw

Abstract

The consensus is an important topic in the reliable distributed system because the system can cope with the influences from faulty components when the agreement is achieved. Siu, Chin and Yang have indicated that the bound on the number of allowable faulty processors in Meyer and Pradhan's scheme is an overestimated one, and they have given a new fixed bound to the allowable faulty processors. However, we find out that the new bound for the allowable faulty processors by Siu et al. still seems to give the scheme too much credit. In this study, we shall give a more accurate estimation to the bound of the allowable faulty processors.

Keywords: Consensus, Distributed System, Fault Tolerance, Dual Failure Mode, Interconnection Networks.

1. INTRODUCTION

In order to ensure the reliability of the fault-tolerant distributed system, reaching a common agreement at the presence of malfunctioning components is the central issue. Such an agreement problem was first studied by Lamport [1], who named it the Byzantine Agreement (BA). In the BA problem, there is a transmitter that transmits the messages at the first round. After the message exchange, each healthy processor should agree on the same value. The BA problem in the distributed system and the assumptions of the BA problem by brought up Lamport [1] are as follows:

Lamport 1. There are *n* processors in a distributed system, where *n* is a constant.

Lamport 2. The processors communicate with each other through reliable fully connected network.

Lamport 3. One or more of the processors may be failed, so a faulty processor may transmit incorrect messages to other processors.

Lamport 4. The number of the arbitrary faulty processors *m* is less than one-third of the total number of processors in the network ($n > 3m$).

Lamport 5. After message exchange, all fault-free processors can reach a common agreement.

The symptoms of processor failure can be classified into three categories [2]. They are crashes, omissions, and arbitrary faults. A crash fault happens when a processor is broken; an omission fault takes place when a processor fails to transmit or receive a message on time or at all; finally, the third kind of processor failure, the thorniest kind of all, is called the arbitrary fault. The behavior of an arbitrarily faulty processor is unpredictable and thus called malicious. Fault-free processors can detect both crash faults and omission faults, so we can call both kinds by still another name “dormant faults.” However, it is not so easy to locate arbitrarily faulty processors.

Due to the facts that different symptoms of processor failure exist and that many practical, popularly used network systems are not of the fully connected type, Meyer et al. [2] revisited the BA problem under quite different assumptions from Lamport’s. Meyer et al. [2] raised the assumptions as follows.

Meyer 1. There are *n* processors in a distributed system, where *n* is a constant.

Meyer 2. All the processors communicate with each other through the network, which is not completely connected.

Meyer 3. One or more of the processors may be failed, and a faulty processor may transmit incorrect messages to other processors, and the failure types of the faulty processors can be either arbitrary or dormant. Fault-free processors can detect the dormant faulty processors.

Meyer 4. The constraint on the number of arbitrary faulty processors *m* and dormant faulty processors *b* is “$n > 3m+b$” and “$c > 2m+b$,” where *c* is the connectivity of the network.

Meyer 5. After message exchange, all fault-free processors can reach a common agreement.

Siu, Chin and Yang [4] have indicated that the bound on the number of allowable faulty processors in Meyer and Pradhan’s scheme [2] is an overestimated one, and they have given a new fixed bound to the allowable faulty processors. However, we find out that the new bound for the allowable faulty processors by Siu et al. [4] still seems to give the scheme too much credit. In this study, we shall give a more accurate estimation to the bound of the allowable faulty processors.

However, Siu et al. [4] indicated that the bound on the number of allowable faulty processors in the scheme proposed by Meyer et al. [2] was overestimated (the assumption Meyer 4). They argued that the correct bound on the number of allowable faulty processors should be $n>\lfloor (n-1)/3 \rfloor+2m+b$.

Nevertheless, we find that the constraint $n>\lfloor(n-1)/3\rfloor+2m+b$ brought up by Siu et al. [4] is also an overestimation in some special situations. Therefore, in this study, we shall give an additional constraint to the bound on the number of allowable faulty processors to make the estimation more credible.

The rest of this paper is organized as follows. Section 2 gives the problem with the bound on the number of faulty processors allowed. Finally, Section 3 shows the application domains of various bounds and gives the conclusion.

2. THE PROBLEM WITH THE BOUND ON THE NUMBER OF FAULTY PROCESSORS ALLOWED

The problem with the figure brought up by Siu et al. [4] is that the bound on the number of allowable faulty processors is overestimated. The constraint on the connectivity, $c>2m+b$, is indeed a necessary condition for reaching an agreement under dual failure modes. That is, the constraint on the connectivity is correct. However, the constraint on the number of processors required, $n>\lfloor(n-1)/3\rfloor+2m+b$, is overestimated. The following example will demonstrate our point:

In Figure 1, there is a fully connected network system with six processors, and the connectivity is five. Suppose that processors P_s and P_1 are subject to arbitrary faults. That is, we have n=6, c=5, m=2, and b=0 in this case. According to the constraints on the failures, namely $n>\lfloor(n-1)/3\rfloor+2m+b$, $c>2m+b$, the bound holds, because 6>1+4+0 and 5>4+0; however, the fault-free processors P_2, P_3, P_4, and P_5 cannot reach an agreement when the BA algorithm [5] is applied.

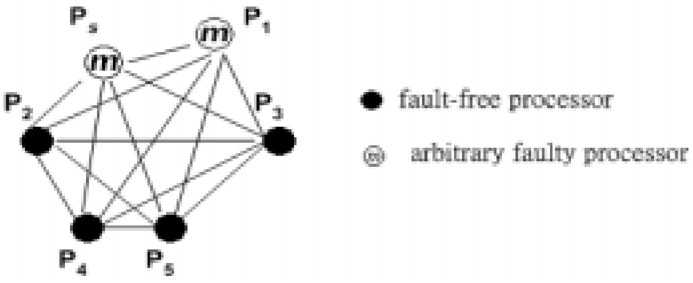


Figure 1. An example network

Figure 2 explains why the results are inaccurate as follows. In order to reach a common agreement, the first step to execute in the BA algorithm [5] is to calculate the number of rounds of message exchange. The number of rounds of message exchange for executing the BA algorithm [5] is $\lfloor(n-1)/3\rfloor+1$. That is, if we want to reach a common agreement in the network shown in Figure 1, we need two rounds of message exchange.

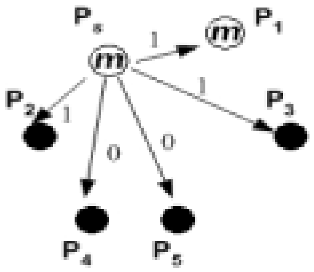
In the first round, the source processor P_s broadcasts its initial value to all the other n-1 processors. Since the source processor is an arbitrary faulty processor, it can broadcast different values to different processors. Figure 2 (a) shows that the message is sent by the source processor. Figure 2(b) shows the messages received by all the fault-free processors in the 1st round. In the second round, each processor broadcasts the message it received in the first round to all the other processors, and then each processor receives messages from other processors as Figure 2(c) shows. After the majority voting in Figure 2(d), we can find that the agreement among all the fault-free processors is not reached; in other words, each fault-free processor's agreement value is not the same. The reason for that is the overestimated bound on the number of arbitrary faulty processors.

Siu et al. [4] have indicated that the number "$n > \lfloor (n -1)/3 \rfloor + 2m + b$" is the maximum when the network has dormant faulty processors only or arbitrary faulty processors only, or when the network has both dormant faulty processors and arbitrary faulty processors. However, we find that the bound $n > \lfloor (n -1)/3 \rfloor + 2m + b$ is an overestimation when (n mod 3)= 0 and b=0. Lamport et al. [1] have indicated that the bound $n>3m$ is the maximum when the network only has arbitrary faulty processors, the detail proof is in Pease et al. [3]. That is, if the network has only arbitrary faulty processors while the total number of processors in the network n mode three equals zero, then the bound offered by Siu et al. [4] is overestimated.

In this study, we shall also give our estimation as to the bound. At the same time, we shall bring in another constraint: If (n mod 3)=0 and b=0, then the bound would be $n>3m$; otherwise, the bound would be $n > \lfloor(n-1)/3\rfloor+2m+b$. Table 1 gives the examples as to the number of faulty processors allowed under the bound by Siu et al. [4] and the correct bound by our estimation.

3. Conclusion

Table 2 shows the application domains of various bounds by Lamport et al. [1], Siu et al. [4] and us. The bound ($n > 3m$) by Lamport [1] is applicable when the network has only arbitrary faulty processors, so it is not applicable when the network has dormant faulty processors or when the network is in the dual failure mode (has both dormant faulty processors and arbitrary faulty processors). The bound ($n > \lfloor (n -1)/3 \rfloor + 2m + b$) by Siu et al. [4] is applicable when the network has only dormant faulty processors or when the network is in the dual failure mode; it is not applicable when the network has only arbitrary faulty processors, for the bound would be overestimated when (n mod 3)= 0 and b=0. Our improved bound is the optimal bound for the case where the network has arbitrary faulty processors only as well as the case where the network has dormant faulty processors, and the case where the network has both arbitrary faulty processors and dormant faulty processors.



(a) The message is sent by the source processor

	level1 root
fault-free processor P_2	1
fault-free processor P_3	1
fault-free processor P_4	0
fault-free processor P_5	0

(mg-tree)

(b) The messages received by each fault-free processor in the 1st round

level 1 root	level 2	
s 0	s1 0	← from P_1
	s2 0	← from P_2
	s3 0	← from P_3
	s4 1	← from P_4
	s5 1	← from P_5

The messages received by fault-free processor P_2 in the 2nd round

level 1 root	level 2	
s 0	s1 1	← from P_1
	s2 0	← from P_2
	s3 0	← from P_3
	s4 1	← from P_4
	s5 1	← from P_5

The messages received by fault-free processor P_3 in the 2nd round

level 1 root	level 2	
s 0	s1 1	← from P_1
	s2 0	← from P_2
	s3 0	← from P_3
	s4 1	← from P_4
	s5 1	← from P_5

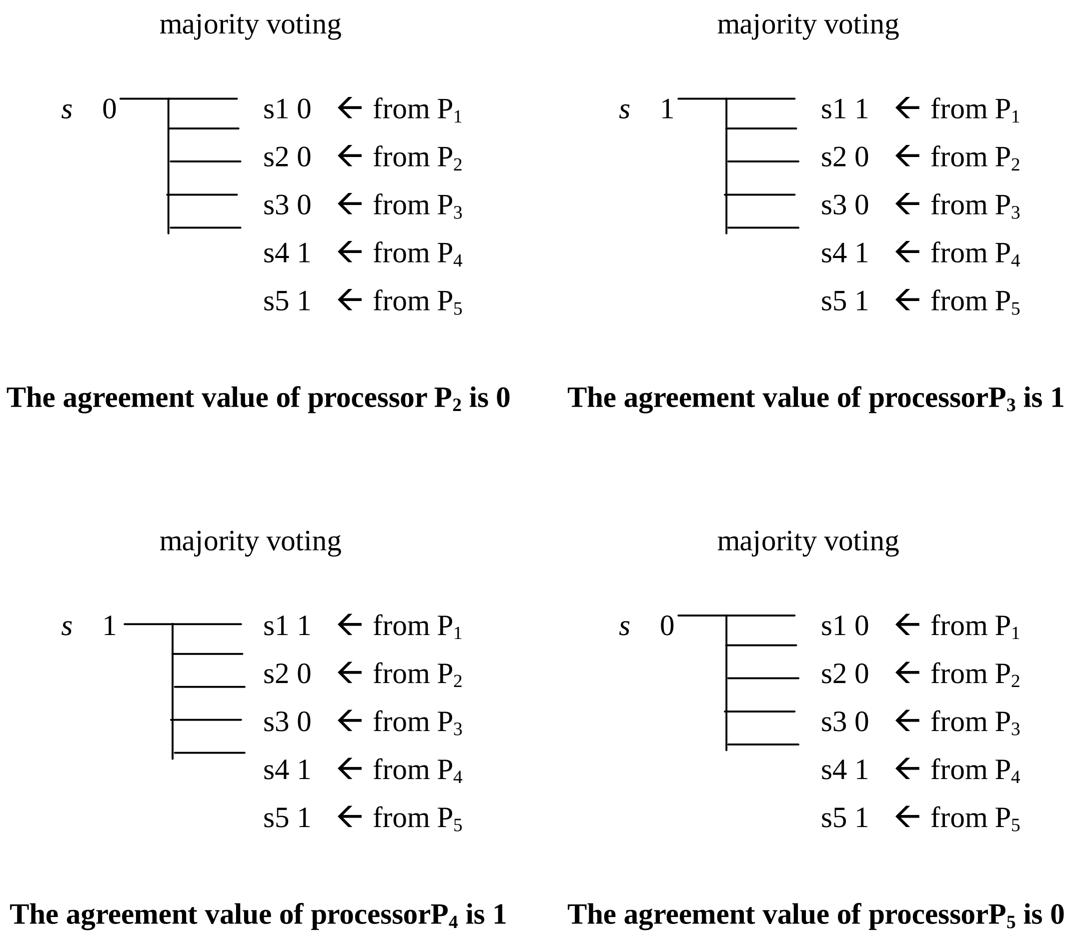
The messages received by fault-free processor P_4 in the 2nd round

level 1 root	level 2	
s 0	s1 0	← from P_1
	s2 0	← from P_2
	s3 0	← from P_3
	s4 1	← from P_4
	s5 1	← from P_5

The messages received by fault-free processor P_5 in the 2nd round

(c) The messages received by each fault-free processor in the 2nd round

Figure 2. The step-by-step execution of the Byzantine Agreement algorithm with $n > \lfloor (n-1)/3 \rfloor + 2m + b$, where n=6, m=2,b=0



The agreement value of processor P_2 is 0

The agreement value of processorP_3 is 1

The agreement value of processorP_4 is 1

The agreement value of processorP_5 is 0

(d) After majority voting

Figure 2. The step-by-step execution of the Byzantine Agreement algorithm with $n > \lfloor (n -1)/3 \rfloor + 2m + b$, where n=6, m=2,b=0

Table 1. Examples of the number of faulty processors allowed under the bound by Siu et al. and the correct bound by our estimation

	The original bund by Siu et al.		Corrected bound by our estimation	
	$\underline{n > \lfloor (n-1)/3 \rfloor + 2m + b}$		if (n mod 3)= 0 and b=0 then $\underline{n > 3m}$ else $\underline{n > \lfloor (n-1)/3 \rfloor + 2m + b}$	
n	*m*	*b*	*m*	*b*
6	**2*** 1	**0** 2	**1** 1	**0** 2
7	2 1	0 2	2 1	0 2
8	2 1	1 3	2 1	1 3
9	**3*** 2 1	**0** 2 4	**2** 2 1	**0** 2 4
10	3 2 1	0 2 4	3 2 1	0 2 4
11	3 2 1	1 3 5	3 2 1	1 3 5
12	4* 3 2 1	0 2 4 6	3 3 2 1	0 2 4 6

***Exceeds the tolerable bound**

Table 2. The application domains of various bounds

Bounds	Arbitrary fault only	Dormant fault only	Dual failure mode
Lamport et al. [1] ($n > 3\ m$)	V		
Siu et al. [4] ($n > \lfloor (n -1)/3 \rfloor + 2\ m + b$)		V	V

Our improvement
if (n mod 3)= 0 and b=0 then $n>3m$
else $n> \lfloor (n-1)/3 \rfloor +2m+b$)
	V	V	V

REFERENCES

[1] L. Lamport, R. Shostak, and M. Pease, “The Byzantine Generals Problem,” *ACM Transactions on Programming Languages and Systems*, 4(3), 1982, pp. 382-401.

[2] F.J. Meyer and D.K. Pradhan, “Consensus with Dual Failure Modes,” *IEEE Transactions on Parallel and Distributed Systems*, 2(2), 1991, pp. 214-222.

[3] M. Pease, R. Shostak, and L. Lamport, “Reaching Agreement in the Presence of Faults,” Journal of ACM, 27(2), 1980, pp. 228-234.

[4] H.S. Siu, Y.H. Chin, and W.P. Yang, “A Note on Consensus on Dual Failure Modes,” *IEEE Transactions on Parallel and Distributed System*, 7(3), 1996, pp. 225-230.

[5] H.S. Siu, Y.H. Chin, W.P. Yang, “Byzantine Agreement in the Presence of Mixed Faults on Processors and Links,” *IEEE Transactions on Parallel and Distributed Systems*, 9(4), 1998, pp. 980-986.