

Optimal Agreement in a Scale-Free Network Environment

Shu-Ching WANG

*Department of Information Management, Chaoyang University of Technology
168, Jifong E. Rd., Wufong Township, Taichung County 41349, Taiwan, R.O.C.
e-mail: scwang@cyut.edu.tw*

Kuo-Qin YAN

*Department of Business Administration, Chaoyang University of Technology
168, Jifong E. Rd., Wufong Township, Taichung County 41349, Taiwan, R.O.C.
e-mail: kqyan@cyut.edu.tw (corresponding author)*

Mao-Lun CHIANG

*Department of Computer Science, National Chung-Hsing University
250, Kuo Kuang Road, Taichung, Taiwan 400, R.O.C.
e-mail: phd9208@cs.nchu.edu.tw*

Received: November 2003

Abstract. Generally, the task in a distributed system must achieve an agreement. It requires a set of processors to agree on a common value even if some components are corrupted. There are significant studies on this agreement problem in a regularized network environment, such as the Fully Connected, BroadCast and MultiCast Networks. Recently, many large complex networks have emerged and displayed a scale-free feature, which influences the system to reach a common value differently. Unfortunately, existing agreement protocols and results cannot cope with the new network environment and the agreement problem thus needs to be revisited. In this paper, we propose a new agreement protocol to adapt to the scale-free network environment and derive its bound of allowable faulty TMs with two rounds of message exchange. We have proved the correctness of this protocol and analyzed its complexity. It is observed that the scale-free network with the proposed agreement protocol can tolerate more faulty TMs than the networks based on previous studies.

Key words: Byzantine agreement, consensus, fault-tolerance, scale-free network, complex network and random network.

1. Introduction

In a distributed computing system, processors allocated in different places or units are connected together to create greater power and ability. Each processor in the system exchanges information with each other. Therefore, the task in a distributed system must achieve an agreement. Notable examples include the two-phase commitment in a dis-

tributed database system, the whereabouts of a replicated file in a distributed environment, and a landing task controlled by a flight path finding system.

To maintain its reliability, the system needs to check periodically and inquire about its exactitude, since components may fail and the failure is unpredictable. The faulty components may disturb or affect others to achieve an agreement among processors. Such a unanimity problem is called the agreement problem (Babaoglu and Drummond, 1985; Barborak *et al.*, 1993; Lamport *et al.*, 1982). The Byzantine Agreement (BA) problem, for example, is an agreement problem in the distributed fault tolerant environment (Dolev, 1982; Dolev and Reischuk, 1985; Lamport *et al.*, 1982; Molina, 1986). The BA problem first studied by Lamport in 1982 (Shima *et al.*, 1996) was solved to make a distributed system run and agree on a common value even if certain processors in the system fail. The main model of the problem describes the system contains n communicating processors, at most t of which are corrupted ($t \leq \lfloor (n-1)/3 \rfloor$) (Erdos and Renyi, 1960). Each processor will agree on a common value if the number of faulty components is less than the fault tolerant boundary t . The goal is achieved if each healthy processor reaches a common value.

A closely related sub-problem, the *consensus* problem, has been extensively studied (Barborak *et al.*, 1993; Bar-Noy and Dolev, 1991; Yan *et al.*, 1999) as well. The solutions are defined as protocols, which achieve an agreement and hope to use the minimum number of rounds of message exchange to achieve the maximum number of allowable faulty capability. The main steps of the *consensus* protocol operate as follows: each processor P_j has its own initial value v_j before executing the agreement protocol, $1 \leq j \leq n$. To start the protocol, each processor broadcasts its initial value to all processors in the first round (the round each processor exchanges messages with others), and subsequently each processor broadcasts the messages received in the first round and makes the decision. The *consensus* problem is solved if the following constraints are satisfied:

(Agreement): All healthy processors agree on a common value.

(Validity): If the initial value of each healthy processor i is v_i then all healthy processors shall agree on the value v_i .

The previous protocols (Erdos and Renyi, 1960; Lamport *et al.*, 1982; Pease *et al.*, 1980) allow processors to reach an agreement and work correctly in those FCN (Fully Connected Network), BCN (BroadCast Network), GCN (Generalized Connected Network) and MCN (MultiCast Network) environments. However, since all the networks show a regularized network structure, these protocols may not work for the network of different structure and connections; for example, some processors immigrate into or move away from the network at any time. Recent research (Barabási, 2000) indicates that the real network environment has shown the property of growth with preferential attachment. One significant discovery observed is that the connectivity distributions of many large-scale complex networks (Barabási and Albert, 1999; Barabási, 1999; Barabási, 2000) including the WWW, Internet and metabolic network, show the power-law form $P(k) \sim k^{-\gamma}$ (Barabási, 2000), where $P(k)$ is the probability that a processor in the network is connected to k other processors, and γ is a positive real number. Namely, the processor in the network has k transmission media, but its connection degree does not

decay exponentially for large k (Erdos and Renyi, 1960). Since the power-law is free of scale, such networks are called “Scale-Free Networks (SFNets)” (Barabási and Albert, 1999; Barabási, 1999; Barabási, 2000; Wang *et al.*, 2000). Generally, a SFNet is inhomogeneous in nature that most processors have a few connections but a small number of particular processors have a huge number of connections. On this ground, the different network topology will seriously influence the system to achieve agreement and possess different capability of fault tolerance. Since the previous protocols are unsuitable for the scale-free network environment, it is necessary to develop new protocols. Based on this reason, this paper proposes a new protocol SFNP (SFNet protocol) to cope with the SFNet environment and finds the new fault tolerance bound for this new type of network.

The rest of this paper is organized as follows. Section 2 describes the scale-free network and the related work on agreement problem. Section 3 presents in detail a new protocol SFNP for the scale-free network and gives an example of executing SFNP. Subsequently, Section 4 proves the correctness and demonstrates the complexity of the new protocol. Finally, Section 5 concludes this paper.

2. Related Work

2.1. Scale-Free Network

The origin of the power-law degree distribution in networks was first addressed by Barabási and Albert-László (Barabási and Albert, 1999; Barabási, 1999; Barabási, 2000; Wang and Chen, 2002). They argued that the two ingredients of self-organization of a network in a scale-free structure are the growth and preferential attachment. These refer to that the network structure continuously grows by the addition of new processors at every time step and preferential attachment to existing processors with a high number of connections. The basic SFNet structure is constructed as follows:

Growth: Starting with a small number (n_0) of processors, at every time step a new processor is introduced and connected to n already-existing processors. Fig. 1 shows there exist four processors P1, P2, P3 and P4 in the network at time t and the new processor P5 immigrates into the network at time $t + 1$.

Preferential attachment: The probability Π that the new processor (P5) is connected to processor P_i depends on the degree k_i of processor P_i , such that

$$\Pi(k_i) = \frac{k_i}{\sum_j k_j}. \quad (1)$$

Fig. 1 shows a scale-free network structure where the network continually grows by adding a new processor P5. The newly introduced processor P5 chooses to connect a processor with k Transmission Media (TMs) following the equation (1), that is, favoring highly connected processors. This phenomenon is called *preferential attachment*. By preferential attachment, the network topology does not follow the Erdős-Rényi (ER) (Erdos and Renyi, 1960) model of random graphs that have n processors and connect every

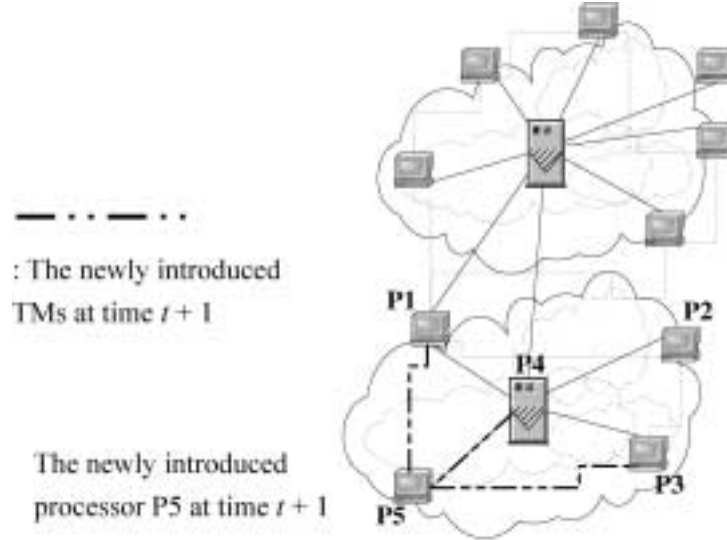


Fig. 1. A general SFNet architecture.

pair of processors with probability α , creating a graph with approximately $\alpha n(n-1)/2$ edges distributed randomly. According to this new observation, each processor in the scale-free network has different connectivity and follows the power law, rather than the Poisson distribution with a peak.

2.2. Previous Results on the Consensus Problem

According to prior literatures, the maximum number of all allowable faulty TMs is $(\lfloor n/2 \rfloor - 1)$ in a $(n-1)$ -connected environment (Dasgupta, 1998; Dolev and Reischuk, 1985; Yan *et al.*, 1992; Yan *et al.*, 1999). Based on the results of Wang (Wang *et al.*, 1995), only two rounds are required in the message exchange phase for each healthy processor to reach an agreement if the number of faulty TMs is less than $(\lfloor n/2 \rfloor - 1)$. Namely, the exchanged messages among the processors are influenced by the faulty TM in the first round, but this fault would be cleared in the second round. Prior literatures also mentioned the c -connectivity protocol (c represents the connectivity state) (Meyer and Pradhan, 1991; Siu *et al.*, 1996), which allows the c -connectivity network to tolerate the maximum number $\lfloor (c+1)/2 \rfloor - 1$ of faulty TMs. Similarly, the system can reach an agreement if the number of faulty TMs does not exceed this limit. However, their assumptions do not cope with the real network (scale-free network), because their number of faulty TMs spreads equally. Thus, the previous research is not appropriate for the SFNet. In the next section, we propose a new protocol SFNP which can achieve the maximum number of fault tolerance within the bound $\lfloor (c_\delta + 1)/2 \rfloor - 1 \leq f_t \leq \lfloor \sum_{i=1}^n (\lfloor (c_i + 1)/2 \rfloor - 1) / 2 \rfloor$ in a scale-free network environment.

3. The Agreement Protocol for SFNet

3.1. SFNP Protocol

This paper proposes a new protocol, called SFNP, to solve the *consensus* problem due to faulty TMs which may send wrong messages to influence the system to achieve an agreement in a SFNet environment. As indicated in the previous studies (Wang *et al.*, 1995; Yan *et al.*, 1988; Yan *et al.*, 1992), the SFNP protocol, like other agreement protocols, consists of two phases and needs two rounds of message exchange to solve the *consensus* problem. In the first round of the message exchange phase, each processor P_i multicasts its initial value v_i through known connected TMs and then receives the initial value of other processors as well. In SFNP, each processor may receive at most c_i (c_i is the connection degree of P_i) messages and constructs a vector from the received messages. In the second round, each processor P_i acts as the sender, sending the vector received in the first round, and constructs a matrix, called the MAT_i , $1 \leq i \leq n$. Finally, the decision making phase will reach an agreement among the processors. The proposed protocol SFNP is presented in Fig. 2.

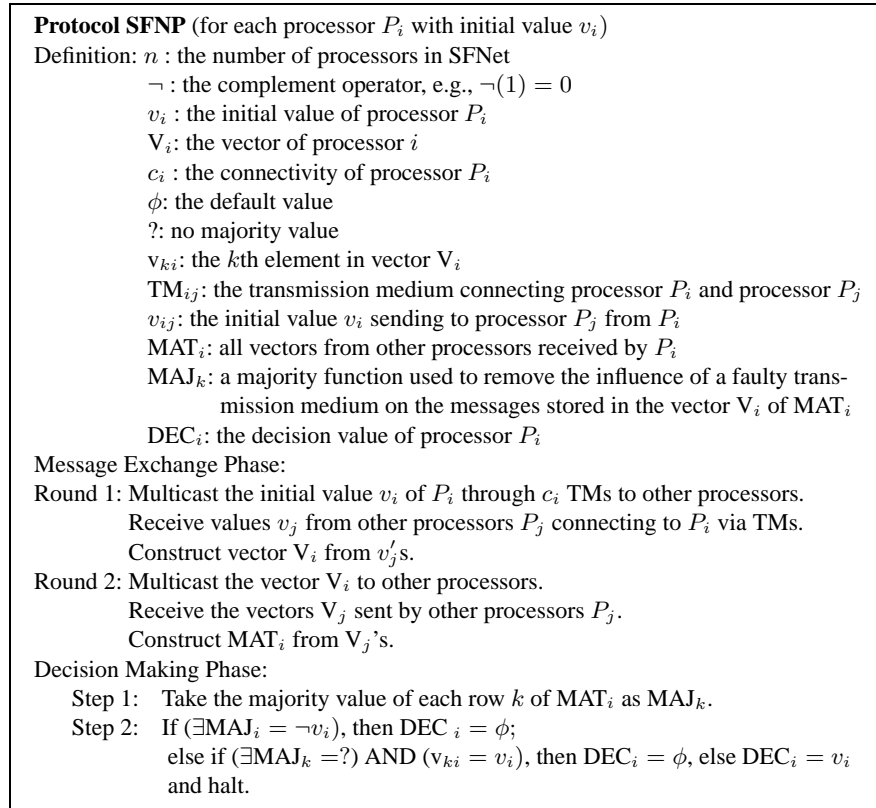


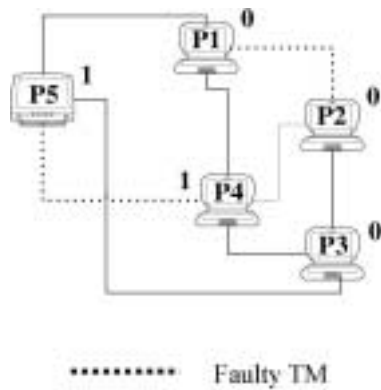
Fig. 2. The proposed protocol SFNP to solve the *consensus* problem.

Subsequently, an example of executing the SFNP protocol based on the network configuration shown in Fig. 3(a) (cutting from Fig. 1) is illustrated as follows. In the first round of message exchange, each processor P_i multicasts its initial value v_i through connected TMs to all other processors, where $1 \leq i \leq n$, and receives the initial value of other processors as well. Then, each processor uses the received message to construct vector V_i as shown in Fig. 3(b). If there is no connection, the value is indicated by λ . In the second round of message exchange, each processor multicasts its vector V_i and receives the vectors from other processors to construct the matrix MAT_i , as shown in Fig. 3(c). Finally, the decision making phase takes the majority value of MAT_i to construct the matrix MAJ_i , as shown in Fig. 3(d), and achieves the common value by MAJ_i .

3.2. Fault Tolerance Capability Analysis

According to literatures (Meyer and Pradhan, 1991; Siu *et al.*, 1996), we may obtain a protocol which can tolerate the transmission media faults in a system provided that $n > 3m + d$ and $c > 2m + d$, where n is the total number of processors in the network, c is the system connectivity, m is the number of malicious faults, and d is the number of dormant faults (Meyer and Pradhan, 1991; Siu *et al.*, 1996; Wang *et al.*, 1995). However, the results are not appropriate for the SFNet environment, as the SFNet grows continuously by preferential attachment and each processor in the network may not have the same number of edges. Therefore, the upper bound $\lfloor (c + 1)/2 \rfloor - 1$ of allowable faulty TMs in the previous research cannot correctly indicate the fault tolerance capability for the SFNet.

To cope with the SFNet environment, the total number of malicious faults m and that of dormant faults d are computed among processors by equations (2) and (3), respectively. Let m_i be the number of malicious faults connecting to processor P_i , and d_i be the



V_1	V_2	V_3	V_4	V_5
0	1	λ	0	0
1	0	0	0	λ
λ	0	0	0	0
1	1	1	1	0
1	λ	1	0	1

Fig. 3(a) A SFNet

Fig. 3(b) The vector received in the first round

Fig. 3. (a), (b). An example of reaching a common agreement in a SFNet (Cont'd).

Messages received by processor $P1$

1	2	3	4	5			
0	0	λ	0	0		0	
1	0	λ	0	λ		0	
λ	1	λ	0	0	\implies	0	\implies 0
1	0	λ	1	0		?	
1	λ	λ	0	1		1	

Messages received by processor $P2$

1	2	3	4	5			
1	1	λ	0	λ		1	
0	0	0	0	λ		0	
λ	0	0	0	λ	\implies	0	\implies 0
0	1	1	1	λ		1	
0	λ	1	0	λ		0	

Messages received by processor $P3$

1	2	3	4	5			
λ	1	λ	0	0		0	
λ	0	0	0	λ		0	
λ	0	0	0	0	\implies	0	\implies 0
λ	1	1	1	0		1	
λ	λ	1	0	1		1	

Messages received by processor $P4$

1	2	3	4	5			
0	1	λ	0	1		?	
1	0	0	0	λ		0	
λ	0	0	0	1	\implies	0	\implies 0
1	1	1	1	1		1	
1	λ	1	0	0		?	

Messages received by processor $P5$

1	2	3	4	5			
0	λ	λ	1	0		0	
1	λ	0	0	λ		0	
λ	λ	0	1	0	\implies	0	\implies 0
1	λ	1	0	0		?	
1	λ	1	1	1		1	

Fig. 3(c)
Construct the matrix MAT_i
Fig. 3(d)
Take the majority value

Fig. 3. (c), (d). An example of reaching a common agreement in a SFNet.

number of dormant faults connecting to processor P_i . Then,

$$m = \sum_{i=1}^n m_i/2 \dots \text{ (for } 1 \leq i \leq n), \quad (2)$$

$$d = \sum_{i=1}^n d_i/2 \dots \text{ (for } 1 \leq i \leq n). \quad (3)$$

According to the above equations, the previous result of $c > 2m + d$ is adjusted to $c_i > 2m_i + d_i$, and the number of allowable fault transmission media for each processor P_i can be analyzed according to the results for c -connectivity protocols.

In what follows, we use the example in Fig. 3 to illustrate how to compute the number of allowable faulty TMs in our protocol. In this example, $n = 5$ because there are five processors in the network. According to the network configuration, $c_1 = 3, c_2 = 3, c_3 = 3, c_4 = 4$, and $c_5 = 3$. Meanwhile, there are $\sum_{i=1}^5 c_i/2 = 8$ TMs in this case. In general, it is assumed that each processor only knows the connection state to its neighbors and we merely use this information for the computation. We analyze its worst-case and best-case scenarios as follows.

Worst Case:

The worst case occurs when faulty TMs focus on those processors that have the minimum connectivity, such as processors P1 and P3 in Fig. 3(a). According to the c -connectivity protocol (Meyer and Pradhan, 1991; Siu *et al.*, 1996), the c -connectivity network allows to tolerate at least $\lfloor (c + 1)/2 \rfloor - 1$ faulty TMs. We assume that f_t is the total number of allowable faulty TMs in a SFNet environment, and c_δ is the smallest connectivity of the system. Because $c_\delta = 3$, the number of allowable faulty TMs in the worst case is $f_t = \lfloor (c_\delta + 1)/2 \rfloor - 1 = 1$. That is, the system is unable to agree on a common value as the number of faulty TMs is larger than 1.

Best Case:

The best case occurs when faulty TMs concentrate on those processors that have the maximum connectivity, such as processor P4. According to the c -connectivity protocol and the Theorem 4 in this paper, the maximum allowable number of faulty TMs is computed by the following equation:

$$\left\lfloor \sum_{i=1}^n (\lfloor (c_i + 1)/2 \rfloor - 1)/2 \right\rfloor = \lfloor (1 + 1 + 1 + 1 + 1)/2 \rfloor = 2.$$

The result shows the allowable number of faulty TMs in the best case is 2, which is different from that in the worst case. While the worst case result is the same as that of prior research (Meyer and Pradhan, 1991; Siu *et al.*, 1996), the best case result does increase the allowable faulty TMs and enhance the fault tolerance capability.

4. Correctness and Complexity

According to the constraints on agreement problem, the protocol SFNP must satisfy the agreement and validity condition if the system reaches a common value. Thus, the following lemmas and theorems are used to prove the correctness and complexity of protocol SFNP.

Lemma 1. *If there is a $MAJ_j = \neg v_i$ in MAT_i , then there is at least one processor with an initial value which disagrees with v_i in the SFNet environment.*

Proof. The majority value in the k th row $= \neg v_i$ means that there are at least $\lceil (n+1)/2 \rceil v_i$'s in the k th row (n is the number of processors). Since the number of faulty TMs is at most $\lfloor n/2 \rfloor - 1$, there exists at least one value $\neg v_i$ received from a healthy transmission medium. In other words, a processor has a disagreeable initial value.

Lemma 2. *Let the initial value of processor P_i be v_i and the TM_{ij} is healthy (not faulty), then the majority value at the i th row in MAT_j should be v_i .*

Proof. Since TM_{ij} is healthy, the processor P_j will receive v_i from processor P_i in the first round and $v_{ij} = v_i$ in MAT_j . Meanwhile, the value v_i of processor P_i will be broadcast to the others. There are at most $\lfloor n/2 \rfloor - 1$ malicious faulty TMs in the system. In the second round, processor P_j receives at least $(n-1) - (\lfloor n/2 \rfloor - 1) = \lceil n/2 \rceil v_i$'s in the i th row of MAT_j . Hence, there are at least $\lceil n/2 \rceil + 1 v_i$'s in the i th row, and the majority value in the i th row should be equal to v_i .

Lemma 3. *If the initial value of processor P_i is v_i , whether the TM_{ij} is healthy or not, the majority value at the i th row of MAT_j , $1 \leq j \leq n$, should either be v_i or not be able to be determined with $v_{ij} = \neg v_i$.*

Proof. By Lemma 2, when TM_{ij} is healthy, the majority value of the i th row in processor P_j is v_i , for $1 \leq j \leq n$. When TM_{ij} is under the influence of malicious fault, we consider the following two cases after running the first round.

Case 1: $v_{ij} = v_i$

Since there are at most $\lfloor n/2 \rfloor - 1$ malicious faulty TMs connected with processor P_j , at most $\lfloor n/2 \rfloor - 1$ values that may be $\neg v_i$'s in the second round. The number of v_i 's is $[(n-1) - (\lfloor n/2 \rfloor - 1)] + 1 = \lceil n/2 \rceil + 1$ in the i th row; therefore, the majority of the i th row in MAT_i is v_i .

Case 2: $v_{ij} = \neg v_i$

There are at most $\lfloor n/2 \rfloor - 1$ malicious faulty TMs. Therefore, in the second round, the total number of $\neg v_i$'s does not exceed $(\lfloor n/2 \rfloor - 1) + 1 = \lfloor n/2 \rfloor$ and the number of v_i 's is at least $[(n-1) - (\lfloor n/2 \rfloor - 1)] = \lceil n/2 \rceil$. If n is an even number, then $\lfloor n/2 \rfloor = \lceil n/2 \rceil$, the majority of the i th row in MAT_j cannot be determined. If n is an odd number, then $\lfloor n/2 \rfloor < \lceil n/2 \rceil$. Hence, the majority of the i th row in MAT_j is v_i .

Lemma 4. *If $(\neg\exists\text{MAJ}_k = \neg v_i)$ and $\{(\exists\text{MAJ}_k = ?)$ and $(v_{ki} = v_i)\}$ in MAT_i , then $\text{DEC}_i = \phi$ is healthy.*

Proof. If MAJ_k does not exist or cannot be determined, there are exactly $n/2v'$'s and $n/2\neg v'$'s in the k th row. Let $v_{ki} = v$ in MAT_i , then, all $n/2\neg v'$'s should be received in the second round. Notably, $\lfloor n/2 \rfloor - 1$ malicious faulty TMs are in the system. Therefore, in the second round, processor P_i at least receives a value from processor P_k without any disturbance. The initial value of processor P_k should disagree with the initial value of processor P_i ; hence it is healthy to choose $\text{DEC}_i = \phi$.

If $v_{ki} = \neg v_i$, we claim that $\neg v_i$ ought to be passed by a faulty TM from processor P_k , and the initial value of processor be $\neg v_{ki} = v_i$.

The next is to prove that if TM_{ki} is healthy, then the initial value of processor P_k should be $\neg v_i$. By Lemma 2, the majority value of the k th row in MAT_i is $\neg v_i$. This is contradictory to the condition of $(\neg\exists\text{MAJ}_k = \neg v_i)$.

If the initial value of processor P_k was $\neg v_i$, then, by Lemma 3, MAJ_k either should be $\neg v_i$ or cannot be determined for $v_{ki} = v_i$. It is a contradiction.

Theorem 1. *SFNP is valid.*

Proof. According to Lemmas 1, 2, 3, and 4, the validity of SFNP is confirmed.

Theorem 2. *Protocol SFNP can reach a consensus.*

Proof. **(1) Agreement:**

Part 1: If a healthy processor agrees on ϕ , then all healthy processors should agree on ϕ .

If the healthy processor P_p with initial value v_i agrees on ϕ , by Theorem 1, there is at least a healthy processor P_k with initial value $\neg v_i$ in the environment. By Lemma 4, the majority value in the k th row of MAT_j , $1 \leq j \leq n$, should be either $\neg v_i$ or ? (cannot be determined) for $v_{kj} = v_i$. All healthy processors with initial value v_i agree on. Similarly, for the healthy processor P_p with initial value v_i , the majority value of the p th row in MAT_j , $1 \leq j \leq n$, either should be v_i or cannot be determined with $\neg v_{ij} = v_i$. All healthy processors with initial value $\neg v_i$ agree on ϕ , too.

Part 2: If a healthy processor agrees on v_i , then all healthy processors should agree on v_i .

If the healthy processor P_i with initial value v_i and $\text{DEC}_i = v_i$, but there exists some healthy processor P_j , $j \neq i$, which has $\text{DEC}_j \neq v_i$. Therefore, such a situation is impossible. To demonstrate this impossibility, if $\text{DEC}_j = \phi$, by Part 1, then $\text{DEC}_i = \phi$. This contradicts the above assumption.

If $\text{DEC}_j = \neg v_i$, then the initial value of processor P_j is $\neg v_i$. Otherwise, it is impossible according to the definition of *consensus* problem. However, if the initial value of processor P_j is $\neg v_i$, by Lemma 4, MAJ_j equals to $\neg v_i$ or cannot be determined with $v_{ji} = v_i$ in MAT_i . Then, $\text{DEC}_j = \phi$, which is a contradiction. Hence, all healthy processors should agree on the same value by the definition of *consensus* problem.

(2) Validity:

To prove this case, the initial value of all processors should be the same. If there is a value $\neg v_i$ in MAT_j , $1 \leq j \leq n$, then the value must be attributed to a malicious faulty transmission medium. There are at most $\lfloor n/2 \rfloor - 1$ malicious faulty transmission medias; hence, there is at most $\lfloor n/2 \rfloor - 1 \neg v_i$'s in each row. Since the value received in the first round may be $\neg v_i$, the majority of each row for all MAT_j should be v_i . Therefore, all healthy processors should agree on v_i .

Theorem 3. *The most amount of information exchange by SFNP is $O(n^2)$.*

Proof. In the first round, each processor sends out at most $(n - 1)$ copies of its initial value to the other processors. In the second round, an n -element vector is sent to the other at most $n - 1$ processors in the environment; therefore, the total number of message exchanges is at most $(n - 1) + (n * (n - 1))$. This finding implies that the complexity of information exchange is $O(n^2)$.

Theorem 4. *The number of allowable faulty components of SFNP is $\lfloor (c_\delta + 1)/2 \rfloor - 1 \leq f_t \leq \lfloor \sum_{i=1}^n (\lfloor (c_i + 1)/2 \rfloor - 1)/2 \rfloor$ in the SFNet configuration.*

Proof. According to the prior literatures (Meyer and Pradhan, 1991; Siu *et al.*, 1996) on c -connectivity protocol, the c -connectivity network allows to tolerate at least $\lfloor (c + 1)/2 \rfloor - 1$ faulty TMs. To cope with the SFNet configuration, we need to accumulate different connectivity of each processor and compute the maximum number of faulty TMs by equations (2) and (3). Thus, the allowable faulty TMs is within $\lfloor (c_\delta + 1)/2 \rfloor - 1 \leq f_t \leq \lfloor \sum_{i=1}^n (\lfloor (c_i + 1)/2 \rfloor - 1)/2 \rfloor$.

5. Conclusion

The agreement problem is a fundamental problem in the distributed environment. The problem has been studied by various kinds of network model in the past. According to previous studies, the network topology plays an important role in this problem. Traditionally, complex networks have been studied in a branch of mathematics known as graph theory. However, the network topology developed in recent years (Barabási and Albert, 1999; Barabási, 1999; Barabási, 2000) shows a scale-free feature such that the previous protocols cannot adapt to it. Therefore, this paper proposes a new protocol SFNP for the scale-free network and finds its new bound on allowable faulty TMs. The SFNP protocol redefines the *consensus* problem in a real SFNet environment and can achieve a common value if the condition $c_i > 2m_i + d_i$ is satisfied. In addition, the maximum number of allowable faulty TMs is improved from $\lfloor (c + 1)/2 \rfloor - 1$ to $\lfloor \sum_{i=1}^n (\lfloor (c_i + 1)/2 \rfloor - 1)/2 \rfloor$. According to our analysis, the new protocol SFNP is feasible to reach the agreement with better fault tolerant capability in the real SFNet environment.

References

- Babaoglu, O., and R. Drummond (1985). Streets of Byzantium: network architectures for fast reliable broadcasts. *IEEE Trans. on Data and Knowledge Engineering*, **SE-11**(6), 546–554.
- Barabási, A.-L., and R. Albert (1999). Emergence of scaling in random networks. *Science*, **286**, 509–512.
- Barabási, A.-L., R. Albert and H. Jeong (1999). Mean-field theory for scale-free random networks. *Physica A*, **272**, 173–187.
- Barabási, A.-L., R. Albert and H. Jeong (2000). Scale-free characteristics of random networks: the topology of the world wide web. *Physica A*, **281**, 69–77.
- Barborak, M., M. Malek and A. Dahbura (1993). The consensus problem in fault-tolerant computing. *ACM Computing Surveys*, **25**(2), 171–220.
- Bar-Noy, A., and D. Dolev (1991). Consensus algorithms with one-bit messages. *Distributed Computing*, **4**, 105–110.
- Dasgupta, P. (1998). Agreement under faulty interfaces. *Information Processing Letters*, **65**, 125–129.
- Dolev, D. (1982). The Byzantine generals strike again. *Journal of Algorithm*, **3**, 14–30.
- Dolev, D., and R. Reischuk (1985). Bounds on information exchange for Byzantine agreement. *Journal of ACM*, **32**(1), 191–204.
- ErdosP., and A. Renyi (1960). On the evolution of random graphs. *Publ. Math. Inst. Hung. Acad. Sci.*, **5**, 17–60.
- Fischer, M. (1983). The consensus problem in unreliable distributed systems (A brief survey). *Lecture Notes in Computer Science*, In *Proceedings of the 1983 International Fault Tolerance Computer Conference*, Borgholm, Sweden. pp. 127–140.
- Lamport, L., R. Shostak and M. Pease (1982). The Byzantine generals problem. *ACM Tran. on Programming Languages and System*, **4**, 384–401.
- Meyer, F.J., and D.K. Pradhan (1991). Consensus with dual failure modes. *IEEE Trans. on Parallel and Distributed Systems*, **2**(2), 214–222.
- Molina, M. (1986). Application of Byzantine agreement in database systems. *ACM Trans. on Database Systems*, **11**(1), 27–47.
- Pease, M., R. Shostak and L. Lamport (1980). Reaching agreement in presence of faults. *Journal of ACM*, **27**(2), 228–234.
- Shima, K., H. Higaki and M. Takizawa (1996). Fault-tolerant causal delivery in group communication. In *Proceedings of International Conference of Parallel and Distributed Systems*. pp. 302–309.
- Siu, H.S., Y.H. Chin and W.P. Yang (1996). A note on consensus on dual failure modes. *IEEE Trans. on Parallel and Distributed Systems*, **7**(3), 335–345.
- Wang, S.C., Y.H. Chin and K.Q. Yan (1995). Byzantine agreement in a generalized connected environment model. In *IEEE Trans. on Parallel and Distributed System*, **6**(4), pp. 420–427.
- Wang, S.C., K.Q. Yan, S.H. Kao and L.Y. Tseng (2000). Consensus with dual link failure modes on generalized network. *CY Journal*, ISSN 1026–244X, 35–52.
- Wang, X.F., and G. Chen (2002). Synchronization in scale-free dynamical networks: robustness and fragility. *IEEE Trans. on Circuits and Systems*, **49**(1), 54–62.
- Yan, K.Q., Y.H. Chin and S.C. Wang (1988). An optimal solution for consensus problem in an unreliable communication system. In *Proceedings of International Conference on Parallel Processing*. pp. 388–391.
- Yan, K.Q., Y.H. Chin and S.C. Wang (1992). Optimal agreement algorithm in malicious faulty processors and faulty links. *IEEE Trans. on Data and Knowledge Engineering*, **4**(3), 266–280.
- Yan, K.Q., S.C. Wang and Y.H. Chin (1999). Consensus under unreliable communication. *Information Processing Letters*, **69**(5), 243–248.

S.C. Wang received the BS degree in computer science from Feng-Chia University, the MS degree in electrical engineering from National Chen-Kung University, and PhD degree in information engineering from National Chiao-Tung University, Taiwan. Currently, she is a professor with the Department of Information Management and dean of the College of Informatics, Chaoyang University of Technology, Taichung County, Taiwan. Her current research interests include distributed data processing, parallel processing, algorithm analysis & design and fault tolerant computing.

K.Q. Yan received the BS and MS degrees in electrical engineering from Chung Cheng Institute of Technology and the PhD degree in computer sciences from National Tsing-Hua University, Taiwan. Currently, he is a professor and chairman of the Department of Business Administration, Chaoyang University of Technology, Taichung County, Taiwan. His current research interests include distributed data processing, parallel processing, fault tolerant computing and electronic commerce.

M.L. Chiang received the MS degree in information management from Chaoyang University of Technology, Taiwan. Currently, he is pursuing his PhD degree in Department of Computer Science from National Chung-Hsing University, Taiwan. His current research interests include distributed data processing, and fault tolerant computing.

Optimalus sutarimas bemačių tinklų aplinkoje

Shu-Cing WANG, Kuo-Qin YAN, Mao-Lun CHIANG

Paskirstytose sistemose procesoriai turi sutarti bendras reikšmes, net jei kai kurie tinklo komponentai yra sugedę. Yra reikšmingų sutarimo problemos tyrimų reguliarių tinklų aplinkoje, pavyzdžiui pilno sujungimo, globalaus paskleidimo tinkluose. Pastaruoju metu atsiranda daug didelių sudėtingų tinklų, deja esantys sutarimo protokolai nepajėgia susidoroti su nauja tinklų aplinka ir dėl to sutarimo problema turi būti peržiūreta. Šiame straipsnyje pasiūlytas naujas sutarimo protokolas, pritaikytas bemačių tinklų aplinkai, ir išvestas jo leidžiamų klaidų režis perdavimo terpėms su dviem pranešimų pasikeitimo ciklais. Straipsnyje įrodytas šio protokolo teisingumas ir išanalizuotas sudėtingumas. Nustatyta, kad bematis tinklas su pasiūlytu sutarimo protokolu yra patikimesnis perdavimo terpėms su klaidomis negu ankstesniais tyrimais pagrįsti tinklai.