A Leader Election Algorithm in a Distributed Computing System

Tai Woo Kim[0], Eui Hong Kim, Joong Kwon Kim
System Engineering Research Institute
Korea Institute of Science and Technology
P.O. Box 1, Yoosung-Gu, Taejon, 305-600, KOREA
s024ktw@crayc90.seri.re.kr, keh@globalx.seri.re.kr

Tai Yun Kim
Department of Computer Science, Korea University
1-Ga, Anam-Dong, Seoul, 136-701, KOREA

Abstract

In this paper, a leader election algorithm based on the performance and the operation rate of nodes and links is proposed. The performance of the existing leader election algorithms should be improved because a distributed system pauses until a node becomes a unique leader among all nodes in the system. The pre-election algorithm that elects a provisional leader while the leader is executing is also introduced in this paper.

1 Introduction

In a distributed system, every node should operate accurately to coorperate with each other. It is difficult to coorperate with each other because of the failure of nodes and links. To get coorperation among nodes, it is needed to control the operation of the entire system. Such mechanisms or entities should be reliable and must have higher performance. There are mechanisms or entities such as file server, time server, and central lock coordinator in the distributed system. In generally, these servers are called leaders. Algorithms which select a leader are called the leader election algorithms.

There are two basic strategies in the leader election algorithms. One of strategy is to make the system temporarily halt normal operation and take a time to reorganize the system. During the reorganization period, the components of the system can be evaluated and compete with each other. The other strategy is to let the system contain software which can operate continuously and correctly when failure occurs and can recover the system. The latter is more complex than the former.

In this paper, the proposed algorithm may make the system operate faster than the former approach, but not continuously. The proposed algorithm is more complex than the former, but it is simpler than the latter.

In the existing leader election algorithms[1, 2, 5, 12], nodes generally elect the leader when the leader fails. In the proposed algorithm, they elect a provisional leader while the leader is under executing. This pre-election algorithm is not to elect the leader, but to replace the leader with the provisional leader when leader fails. This approach can prevent the system from the performance degradation in the distributed system.

The failure of nodes and links can occur during or after the election in the distributed system. In this paper, a leader election algorithm based on the performance and the operation rate of nodes and links is modeled in order to predict the appropriate leader node[10].

Section 2 reviews the existing works relevant to this study and discusses the issues in the election algorithm. The assumptions and definitions of the proposed algorithm are given in section 3. System model is sketched in section 4. Section 5 focuses on the description of the pre-election algorithm. Performance analysis of the algorithm is discussed in section 6. Finally, the evaluation of this work is discussed.

2 Background

The leader election algorithms considering the performance and operation rate of nodes have been proposed by Singh and Kuros[6]; and Singh[7]. The former is based on the delay model, and the latter is based on the expected connectivity in the distributed system.

These ideas are considered to be excellent with the performance and operation rate of nodes and links. On the other hand, these approaches have some disadvantages as follows. First, in the algorithm by Singh and Kuros[6],

0-8186-7125-4/95 $04.00

since the $g_i(j)$ has been assumed to be the estimated relative performance in the link delay model, it seems to be difficult to model the estimated relative performance at node i for node j. Second, the performance of each node is derived from the expected connectivity[7]. The performance of nodes and links means the amount of bytes processed per unit time. In the algorithm by Singh[7], if a node becomes the best leader with the reason that the node is only operated without nodes and links failure, it might be possible to be modeled unreasonably. Finally, it is believed that the node's own performance should have been considered in their works[6, 7]. In this paper, the leader election algorithm which is based on the performance and operation rate of nodes and links is proposed to improve their algorithms. Also, pre-election algorithm which elects a provisional leader while leader is running is introduced in this paper.

3 The assumptions and definitions

3.1 Assumptions

Existing algorithms require some difficult conditions to work in practice, such as atomically writable storage that is preserved across node crashes and reliable transmission mechanism. Most of assumptions in this paper follow the details suggested in the Morina[2]. The assumptions for the proposed algorithm which differ from Morina[2] are as follows.

- The contents in the core memory is completely removed when a node crashs.
- Messages are the process-to-process datagram which can be lost, delayed, duplicated, or received in different order[11].
- All tasks at a node can be performed continuously even if the leader is replaced with the provisional leader.
- There is no specific server which provides a global time in the distributed system[3].
- The performance of nodes and links differs from each other.

3.2 Definitions

The definitions of link, node, and failure rate are discussed in this section. The proposed algorithm is based on these definitions.

Definition 1 The performance of links

It is assumed that a system contains n nodes (labeled S = {1,2,$\cdots$, n}). The shortest path between nodes i and j is denoted as l_{ij}. The weighted value d_{ij} of l_{ij} means transmission delay time in the system. Although d_{ij} is related to the message length and the performance of a node, it is assumed that d_{ij} is given for simplicity. The weighted value d_{ij} of a link has the following characteristics.

a) asymmetricity : $d_{ij} \neq d_{ji}$

b) non-negativity : $d_{ij} \geq 0$, iff $i = j$, d_{ij}=0

c) transitivity : $d_{ik} \leq d_{ij} + d_{jk}$

Definition 2 The performance of nodes

a) Absolute performance of nodes

The absolute performance of node i is the amount of bytes processed per unit time and is denoted as f_i. For the arbitrary node i in the system, the weigthed value of the node denotes a_i which is defined as the absolute response delay time.

When $\mathcal{R}_i$ is a function which is represented by real value, the relationship between the aboslute response delay time a_i and the absolute performance f_i of node i is as follows.

$$f_i = \mathcal{R}_i(\frac{1}{a_i}) \tag{1}$$

b) Relative performance of nodes

The $U_i(j)$ is the sum of weighted value at node i for nodes j in the system. The $U_i(j)$ is defined as the relative response delay time. The $g_i(j)$ represents the relative performance of node i for nodes j. When $\mathcal{R}_i$ is a function which is represented by real value, the relationship between the relative response delay time $U_i(j)$ and the relative performance $g_i(j)$ of a node is as follows.

$$U_i(j) = \mathcal{R}_i(\frac{1}{g_i(j)}) \tag{2}$$

Definition 3 Operation rate of nodes and links

The operation rate of node i is defined as p_i. The operation rate of link l_{ij} is defined as p_{ij}. Considering the operation rate of node i and link l_{ij}, the estimated performance, $\hat{f}_i$ and $\hat{d}_{ij}$ are defined as follows, respectively.

$$\hat{f}_i = f_i \cdot p_i \tag{3}$$

$$\hat{d}_{ij} = d_{ij} \cdot p_{ij} \tag{4}$$

4 System model

The optimal leader is the nearest node from all nodes and has the superior performance to other nodes in a system. The method to find the leader is similar to the network location algorithm[8]. It is assumed that S is the set of nodes and r is an arbitrary point on the link. When h is an arbitrary node in the system, the node

which minimizes the sum of the weighted distance from all nodes is formulated as the equation (5).

$$\sum_{i \in S} a_i \cdot d_{ih} \leq \sum_{i \in S} a_i \cdot d_{ir} \tag{5}$$

The detailed proof of the equation (5) refers to the [8]. If $U_i(j) = \frac{1}{g_i(j)}$, then $U_i(j)$ is the equation (6) using the equation (5).

$$U_i(j) = \sum_{j \in S}^{n} a_j \cdot d_{ij} \tag{6}$$

Replacing a_j of the equation (6) with the equation (1) and considering the equation (2) becomes,

$$g_i(j) = \sum_{j \in S}^{n} \frac{f_j}{d_{ij}} \tag{7}$$

When w_i denotes the total performance at node i for the entire system, the total performance at node i can be expressed as $w_i = f_i + g_i(j)$.

Let's consider the failure rate of nodes and links. If the equation (7) is substituted by the equation (3) and (4), then $\hat{g}_i(j) = \sum_{j \in S}^{n} \frac{\hat{f}_j}{\hat{d}_{ij}} = \sum_{j \in S}^{n} \frac{f_j \cdot p_j}{d_{ij} \cdot p_{ij}}$. The total estimated performance $\hat{w}_i$ of node i is $\hat{w}_i = \hat{f}_i + \hat{g}_i(j)$. The node which has $\hat{W}_i$ is elected as a leader node. Note that $\hat{W}_i = Max\{\hat{w}_1, \hat{w}_2, \cdots, \hat{w}_n\}$.

When the time T goes by, after the provisional leader is elected, the predictability to become a leader is calculated as follows. The relationship between the failure rate $\bar{p}$ and the operation rate p of node i is $p = 1 - \bar{p}$. Since the failure rate of node i follows an exponential distribution[4], the failure rate of node i can be defined as $\bar{p} = \lambda$. If and only if at least one of nodes in the system is working, the probability of the operation rate for node i is calculated as $P^i = p \cdot q^{n-1}$. The node which has the largest P^i is elected according to the algorithm. The expected value for node i has a functional relationship between the total performance w_i and the operation rate P^i of node i.

$$\mathbf{E}^i = w_i \cdot P^i \prod_{j=1}^{n-1} \lambda_j, \quad i \neq j, \quad 1 \leq i, j \leq n$$

The time to recover from the node failure is represented as $\frac{1}{\mu}$. When a node is operated at the initial time t_0, the operation rate of node i at time t is calculated as follows.

$$P^i_{00}(t) = \frac{\lambda_i}{\mu + \lambda_i} \cdot e^{-(\mu+\lambda_i)t} + \frac{\mu}{\mu + \lambda_i} \tag{8}$$

The equation (8) is derived from the Kolmogorov's equation $P'_{0j}(t) = \lambda_0 \cdot P_{1j}(t) - \lambda_0 \cdot P_{0j}(t)$ [4]. Using equation (8), the expected value of the operation rate for node i at time $t > 0$, $\mathbf{E}^i_{00}(t)$ is expressed as follows.

$$\mathbf{E}^i_{00}(t) = w_i\left(\frac{\mu + \lambda_i e^{-(\mu+\lambda_i)t}}{\mu + \lambda_i}\right) \prod_{j=1}^{n-1} \left(1 - \frac{\mu + \lambda_j e^{-(\mu+\lambda_j)t}}{\mu + \lambda_j}\right)$$

where $i \neq j$ and $t > 0$. If the recovery time of node i is $\frac{1}{\mu} \ll 1$, $\mathbf{E}^i(0) \approx \mathbf{E}^i_{00}(t)$. This means the leader elected at time 0 approximates the one at time t. The detailed proof of this equation refers to the [9].

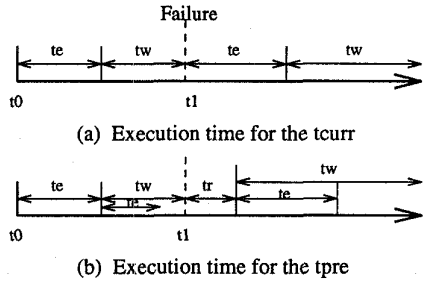

Figure 1: Execution time for the t_{curr} and the t_{pre}

5 Description of the algorithm

When a current leader fails for any reason, each node detects this and replaces the leader with the provisional leader. The node which firstly detects the leader failure elects the optimal provisional leader based on the performance and the operation rate of the nodes and the links. It broadcasts the elected provisional leader node ID to all nodes in the system. If any node becomes the new leader, all the nodes resume normal operation and only one node elects the provisional leader. Figure 1 describes this process. The t_{curr} means the processing time for the existing algorithm. The t_{pre} refers to the processing time to process the pre-election algorithm. The t_e is the election time to elect the leader or the provisional leader. The t_w is the normal working time to perform it's tasks at node i. From Figure 1(a), since the k nodes of which the system consists participate in the election to elect the leader node, the system is under idle state during t_e. After the election, the system is under working state during t_w.

At Figure 1(b), the pre-election algorithm operates like Figure 1(a) at time t_0. The k nodes participate in the election at time t_0. When the leader node crashes at time t_1, k nodes replace the leader with the provisional leader and $(k-1)$ nodes resume their tasks, while only the node which firstly detects the leader failure elects the next provisional leader. When the node that firstly detects the leader failure finishs the election, it resumes normal operation to perform it's tasks.

6 Performance Analysis

The time complexity and the message complexity are discussed in this section.

It is assumed that the system contains the n nodes, and the k nodes are on working($k \leq n$). The D denotes the diameter of the distributed system network. The time complexity of an election algorithm which elects the leader as a leader failure and that of the pre-election algorithm are $t_{curr} = k(3.5t_b + t_s + k)$, $t_{pre} = k(2.5t_b + t_s + 1)$, respectively[9].

The t_b is denoted as the required time to broadcast the messages and the t_s is represented the required time to send the message between two nodes. The t_{pre} is faster than the t_{curr} as $k(t_b+t_s-1) = k(k-1)(2D+1)$. The time complexity of the proposed algorithm is the $O(k(k-D))$ and figures out in Figure 2. The time complexity of the algorithm depends upon the network topology and the number of nodes in the system. The time complexity of the proposed algorithm is examined on the bounded degree network, on the complete network, and on the ring network. According to the network topologies, the time complexity of the proposed algorithm is measured as Figure 3 shows. When the performance and operation rate of a nodes are given, the predictability to become leader for node i is depicted at Figure 4. The message complexity of the proposed algorithm is the O(n + k) and is same as the existing algorithms.

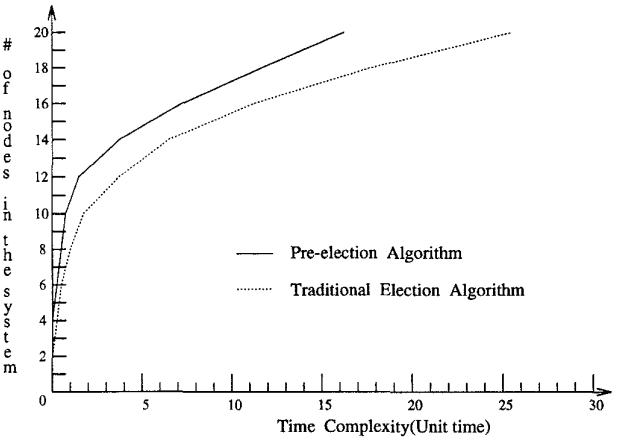

Figure 2: Time complexity

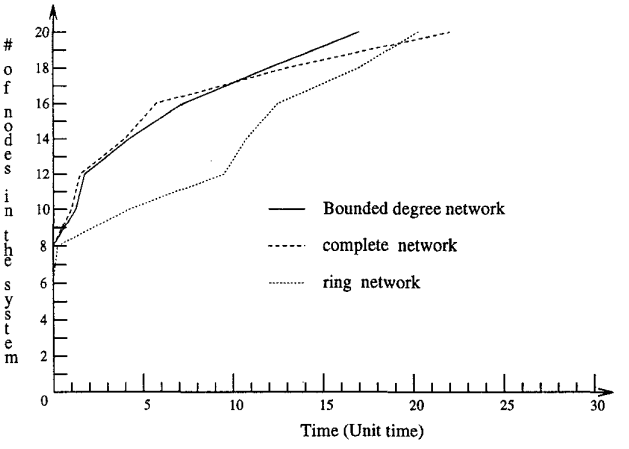

Figure 3: Time complexity according to network topologies

7 Conclusion

The pre-election algorithm which elects the provisional leader while leader is running is proposed and modeled in this paper. However, the performance of the pre-election algorithm is influenced by the predictability of an algorithm. To improve the predictability the pre-election algorithm adapts the election scheme which is considered as the performance and the operation rate of nodes and links. The proposed algorithm has a following characteristics.

- While the existing algorithms elect the node simply having the largest ID number, the proposed algorithm elects the node considering the performance and operation rate of nodes and links.
- The pre-election algorithm which elects the provisional leader while the leader is running, but the existing election scheme elects the leader when the leader fails.
- While the existing algorithms have been developed with the specific network topology, the proposed algorithm can be applied at the any kind of network topologies.

A further study is to develop the efficient algorithm based on the proposed algorithm when the number of nodes is increased in the distributed computing network. The effects of the traffic pattern and the amount of the messages among nodes also will be studied.

References

[1] Gurdip Singh, "Real-time leader election," *Information Processing Letters*, vol 49, 59-61, 1994.

[2] Hector G. Morina, "Election in a distributed computing system," *IEEE TX on computers*, vol C-31, no 1, 1982.

[3] Riccardo Gusella and Stefano Zatti, "The Berkeley UNIX 4.3 BSD time synchronization protocol : pro-

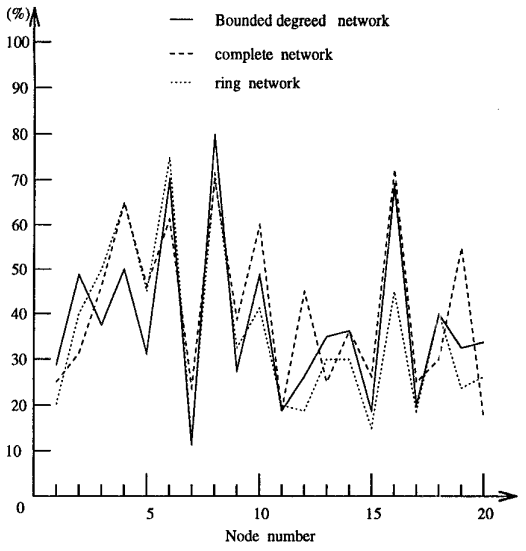

Figure 4: The predictability of nodes

tocol specification," *Technical Report*, CSD85-250, *U. of California*, Jun 1985.

[4] Sheldon M. Ross, *Introduction to probability models : Ch 6. Exponential models and continuous time markov chains*, Academic Press Inc., 1982.

[5] Shing-Tsaan, "Leader election in uniform rings," *ACM TX on PL and systems*, vol 15(3), 563-573, 1993.

[6] Suresh Singh and James F. Kuros, "Electing leader based upon performance:the delay model," *Proceedings of the 11th international conference on DCS*, Arlington, 464-471, 1991.

[7] Suresh Singh, "Expected connectivity and leader election in unreliable networks," *Information Processing Letters*, vol 42, 282-285, 1992.

[8] T. B. Boffey, *Graph theory in operations research : Ch 5. Location problem*, Macmillan Press Ltd., 1982.

[9] Taiwoo Kim and Taiyun Kim, "A pre-election algorithm in the distributed computing system," Submitted to *Journal of the Korean Institute of Communication Sciences*, 1995.

[10] Taiwoo Kim, Joongkwon Kim and Taiyun Kim, "A study on the leader election algorithm in distributed system," *Proceedings of the 21st KISS Fall conference*, Seoul, 201-204, Oct 1994.

[11] Uyless Black, *TCP/IP and related protocols*, McGraw-Hill Inc., 1992.

[12] Yuan-Chieh, Luo K., and Newman R., "An optimal distributed algorithm for failure-driven leader election in bounded-degree networks," *Proceedings for 3rd IEEE workshop on FTDCS*, Taipei, 136-141, 1992.