

# 양방향 선호도에 기반을 둔 분산 리더 선거 전략의 견고성

진 기 범<sup>†</sup> 최 경 희<sup>\*\*</sup> 이 형<sup>\*\*\*</sup>

## 요 약

리더를 선출하는 기존의 선거 전략은 노드의 고유 번호만을 고려하게 되어 노드의 성능을 고려하지 않아서 좋은 리더를 선출할 수 없다. 이러한 문제점을 해결하기 위하여 성능을 고려하여 리더를 선출하는 방법에 대한 연구가 진행 되고 있다. 리더 선출 도중에 발생할 수 있는 링크의 고장은 리더의 성능에 영향을 주게 된다. 본 논문에서는 성능을 양방향 지연값을 선호도로 할때, 링크의 고장이 이러한 선거 전략의 견고성에 미치는 영향에 대하여 연구하였다. 특히 선호도에 기반을 둔 선거전략 중 링크고장이 발생했을때 견고성이 가장 좋은 전략이 어느 전략인지를 모의 실험을 통하여 연구하였다.

## Robustness in the Distributed Election Strategies Based on the bidirectional preference of a Node

Ki-Bum Chin<sup>†</sup>, Kyung-Hee Choi<sup>\*\*</sup> and Hyung Lee<sup>\*\*\*</sup>

## ABSTRACT

Most of traditional approaches to leader election have been based on a node ID number. As performances were not considered in these approaches, they were not very practical. So there has been a study on leader election strategies based on performances. This paper proposes new preference based on performances. While link failures occur during the election period, the elected leader's performance may not be the expected one. We simulate these effects of link failures on the robustness at election strategies. In particular, we tried in this study to find the best of the distributed election strategies based on the bidirectional preference from the standpoint of the robustness through simulation.

### 1. 서 론

분산 컴퓨터 시스템은 어떤 공동 목적을 위하여 자치적인 컴퓨터, 노드 또는 프로세스들이 협동 한다. 공유 자원을 액세스 할때 작업의 제어를 분산 제어로 할 경우에는 모든 프로세스가 참여하게 되는데 중앙 집중식으로 제어 할 경우에는 한 리더를 선출하여 조정 역할을 맡도록 한다. 기존의 리더 선출 알고리즘에서는 각 노드에 게 고유 번호를 주고 가장 큰 번호 또는 가장 작은 번호를 갖는 노드를 뽑는 것이었다[1, 2, 6, 7,

8, 9, 17]. 작업을 중앙 집중 식으로 제어 할때는 여러 노드 중에서 한 노드를 뽑아서 조정 역할을 맡아서 처리 하도록 한다. 그러나 리더가 고장이 났다면 시스템 전체의 고장으로 이어지는 결점이 있어 이 점을 극복하기 위해 고장나지 않은 프로세스 중에서 새로운 프로세스를 선출해서 고장난 리더의 역할을 하도록 한다[3, 4, 15, 16].

리더를 선출하는 도중에 노드의 고장은 없고 링크의 고장은 발생할 수 있다고 가정 할 때에, 노드의 고유번호만을 고려하여 리더를 선출하면 결국 고유 번호가 가장 큰 노드가 리더로 선출되므로 링크의 고장이 리더 선출에 미치는 영향은 단순히 리더 선거 알고리즘 자체의 문제로 국한되며, 많은 연구는 링크의 고장이 발생하더라도

<sup>†</sup> 정 회 원 : 숭실대학교 전자계산원 전임강사  
<sup>\*\*</sup> 정 회 원 : 아주대학교 컴퓨터공학과 교수  
<sup>\*\*\*</sup> 종신회원 : 대전대학교 정보공학과 교수  
논문접수 : 1994년 4월 1일, 심사완료 : 1994년 7월 20일

어떻게 리더를 안정적으로 선출하는가라는 문제를 취급하고 있다[10].

성능에 기반을 둔 연구[16]에서도 노드고장이라는 것은 노드는 기능을 하는데 잘못된 투표만을 던지는 것으로 정의하고서 견고성을 정의 한 다음 노드 고장이 생기는 경우에 대한 견고성을 연구 하였다. 그 연구에서는 링크고장에 대해서는 전혀 고려하지 않았다.

본 연구에서는 각 선거전략별로 링크고장 확률 몇%까지를 견디는가 즉 링크고장에도 불구하고 최적리더를 선출하는 확률이 얼마나 변하는지와 상대오차에 대하여 모의 실험을 통하여 단방향 지연값을 선호도로 하는 선거전략과 양방향 지연값을 선호도로 하는 선거전략에 대한 견고성을 비교 연구하고, 여러 선거전략 중 견고성이 좋은 선거전략이 어떤 전략인지를 알아보려고 한다. 모의실험을 위하여 균등분포를 이루는 난수를 발생하여 랜덤 가중치 행렬을 만들어 각 선거전략에 의한 리더를 선출하고 난수를 발생시켜 링크고장을 발생시켜서 모의실험을 통하여 나온 결과를 분석한다.

선거 알고리즘을 실행하는 동안에 노드가 고장나는 경우는 없다고 가정하였고, 본 논문에서는 분산 시스템을 논문[16]에서와 같이 지연값에 기반을 두어 가중치 그래프로 모델링 하였다. 지연값과 링크가 고장날 확률은 난수를 발생시켜 부여 하였다.

확률별로 링크가 고장날 수 있는 경우를 나누어서 기존의 선거전략[16]과 본 논문에서 제안한 양방향 지연값을 선호도로 하는 선거전략의 견고성에 대하여 몬테칼로(Monte Carlo)기법을 이용하여 실험결과를 비교하면서 실험결과를 분석하였다.

2장에서 시스템 모델(model)을 간략히 기술하고, 3장에서는 견고성의 측도에 대하여 4장에서는 링크의 고장과 견고성에 대하여 기술하고, 5장에서는 링크 고장과 리더의 견고성에 대하여, 6장에서는 시뮬레이션과 결과에 대해서 기술하고 있고, 7장에서는 결론을 내리고 있다.

## 2. 시스템 모델

분산 시스템은 그래프  $G(V, E)$  ( $V$ :노드들의 집합,  $E$ 는 간선(edge)들의 집합)로 나타낼 수 있고 노드 사이의 통신채널에 해당하는 링크를 나타내는 모든 간선은 양방향이고 가중치(weight)가 주어 진다고 가정한다. 가중치는 노드 사이의 통신에 소요되는 시간, 즉 지연값을 나타낸다고 하자.  $(i, j)$ 가 간선이면  $(j, i)$ 도 역시 간선이고,  $(i, j)$ 위에서의 지연값은  $(j, i)$ 에서의 지연값과는 서로 독립적이다.

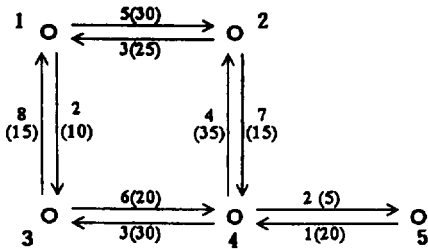
링크 위에서의 지연값은 전송 노드에서의 전송 지연과(delay) 큐잉(Queueing)지연의 합으로 정의 된다. 논문[16]에서의 지연 모델과는 달리, 본 논문에서는 이러한 지연값과 함께 각 간선에 링크가 고장날 확률을 부여 하였다. 이러한 확률은 논문[16]에서 언급하고 있는 링크의 신뢰도에 해당한다.

리더 노드는 다른 노드들과 계속적으로 통신채널을 통하여 메시지를 교환하게 되므로 통신에 소요되는 지연값이 적을수록 리더는 다른 노드에게 좋은 서비스를 제공할 수 있게된다. 이에 본 논문에서는 리더노드의 성능의 측도로써 노드 사이의 지연값을 택하였다. 기존의 방법[16]에서는 노드  $j$ 의 시스템의 전반적인 성능은 모든 노드  $i$ 로부터  $j$ 로 가는 최단 경로의 합으로 정의 하였으나, 본 논문에서는 다른 모든 노드로부터 오는 경로와 가는 경로를 모두 고려한 최단 경로들의 합으로 정의한다. 즉, 노드  $j$ 의 시스템 전반의 성능  $D(j)$ 는  $\sum_{i=1 \neq j}^n d(i, j) + \sum_{k=1 \neq j}^n d(k, j)$ 으로 정의 한다. 단  $d(i, j)$ 는 노드  $i$ 에서  $j$ 까지의 최단 경로이다. 따라서  $D(j)$ 는 모든 노드에서  $j$ 까지 최단 경로길이의 합과 노드  $j$ 에서 모든 노드까지의 최단 경로의 합이다.

$D(k) \leq D(j)$  ( $1 \leq j \leq n, j \neq k$ )를 성립하는 노드  $k$ 를 최적리더라 정의한다.

논문 [16]에서는 각 노드에게 후보노드까지의 지연값을 선호도로 하여 각 선거전략별로 리더를

선출하였는데 본 논문에서는 각 노드에서 후보 노드까지의 지연값과 후보노드에서 그 노드까지의 지연까지의 지연값의 합을 선호도로 하여 노드  $i$ 가 본 노드  $j$ 의 양방향 선호도  $P_i(j)=d(i, j)+d(i, j)$ 로 정의하여 다음 주어진 시스템 모델을 가지고 각 노드에서 본 양방향 선호도를 구하여 선거전략 중 가장 선호하는 노드에게 한표를 던져서 최다 득표한 노드를 리더로 선출하는 다수결 전략의 예를 들면 다음과 같다.



단, 간선위의 가중치는 지연값을, 괄호안의 값은 링크 고장확률을 나타낸다.

노드 1이 본 다른 노드들의 양방향 선호도는 다음과 같다.

$$P_1(2)=d(1, 2)+d(2, 1)=8$$

$$P_1(3)=d(1, 3)+d(3, 1)=10$$

$$P_1(4)=d(1, 4)+d(4, 1)=15$$

$$P_1(5)=d(1, 5)+d(5, 1)=18$$

그러므로 노드 1이 가장 선호하는 노드는 2번 노드이다.

노드 2가 본 다른 노드들의 양방향 선호도는 다음과 같다.

$$P_2(1)=d(2, 1)+d(1, 2)=8$$

$$P_2(3)=d(2, 3)+d(3, 2)=15$$

$$P_2(4)=d(2, 4)+d(4, 2)=11$$

$$P_2(5)=d(2, 5)+d(5, 2)=14$$

그러므로 노드 2가 가장 선호하는 노드는 1번 노드이다.

노드 3이 본 다른 노드들의 양방향 선호도는 다음과 같다.

$$P_3(1)=d(3, 1)+d(1, 3)=10$$

$$P_3(2)=d(3, 2)+d(2, 3)=15$$

$$P_3(4)=d(3, 4)+d(4, 3)=9$$

$$P_3(5)=d(3, 5)+d(5, 3)=12$$

그러므로 노드 3이 가장 선호하는 노드는 4번 노드이다.

노드 4가 본 다른 노드들의 양방향 선호도는 다음과 같다.

$$P_4(1)=d(4, 1)+d(1, 4)=15$$

$$P_4(2)=d(4, 2)+d(2, 4)=11$$

$$P_4(3)=d(4, 3)+d(3, 4)=9$$

$$P_4(5)=d(4, 5)+d(5, 4)=3$$

그러므로 노드 4가 가장 선호하는 노드는 5번 노드이다.

노드 5가 본 다른 노드들의 양방향 선호도는 다음과 같다.

$$P_5(1)=d(5, 1)+d(1, 5)=18$$

$$P_5(2)=d(5, 2)+d(2, 5)=14$$

$$P_5(3)=d(5, 3)+d(3, 5)=12$$

$$P_5(4)=d(5, 4)+d(4, 5)=3$$

그러므로 노드 5가 가장 선호하는 노드는 4번 노드이다.

다수결 전략에 의하면 노드 1은 1표, 노드 2도 1표, 노드 3은 0표, 노드 4는 2표, 노드 5는 1표를 얻어 최다득표한 노드 4가 리더로 선출 된다.

논문 [16]에서 우수한 선거전략이라고 밝힌 플루애플(Plu-app)전략과 보다(Borda)전략을 기술하면 플루애플전략은 각 노드에서 가장 선호하는 노드에게 2표를, 평균 이상인 선호도를 갖는 후보 노드에게는 1표를 던져 최다 득표한 노드가 리더가 되고, 보다 전략은 각 노드에서 후보노드에 대한 선호도 순으로 1표씩 차등으로 투표하여 최다 득표한 노드가 리더로 선출 되는 방식이다.

여러개의 후보중에 하나를 랜덤(random)하게 선출하는 전략을 랜덤 선거전략이라 한다. 노드 번호를 기반으로 리더를 선출하는 방법은 선호도를 고려하지 않았다는 점에서 이 전략과 동일하며 다른 선거전략들과 비교하기 위한 벤치마크 알고리즘의 역할을 하게한다[10].

그러나 위에서 기술한 선거전략들이 항상 최적 리더를 뽑는다고는 볼 수 없으므로 선출된 리더

에 대한 질(quality)을 평가하기 위하여 선거전략에 의해 최적리더가 선출될 확률이나 선출된 리더와 최적리더가 시스템에 제공하는 성능의 상대적인 차의 기대치 즉, 상대오차중에서 본 논문에서는 상대오차를 기준으로 선출된 리더의 성능 비교하였다.

$$\text{상대오차} = \frac{D(0) - D(i)}{D(0) - D(k)} \times 100 \text{으로 정의한다.}$$

단, 0는 선거 알고리즘에 관계없는 시스템의 최적리더이며, k는 최악의 리더이고, i는 선거전략에 의하여 선출된 리더의 번호이다.

### 3. 견고성(robustness)의 측도

선거전략의 견고성을 링크 고장이 발생 되었을 때 선출되는 리더의 성능을 상대오차를 이용하는 것과 최적리더 선출 확률을 이용하여 다음과 같이 정의한다.

① 상대오차를 이용한 견고성 측도

$$\frac{|E(\text{상대오차} | n\% \text{ 링크고장}) - E(\text{상대오차} | 0\% \text{ 링크고장})|}{E(\text{상대오차} | 0\% \text{ 링크고장})} \times 100$$

② 최적 리더 선출 확률을 이용한 견고성 측도

$$\frac{|P(\text{최적리더를 뽑는다} | n\% \text{ 링크고장}) - P(\text{최적리더를 뽑는다} | 0\% \text{ 링크고장})|}{P(\text{최적리더를 뽑는다} | 0\% \text{ 링크고장})}$$

### 4. 링크고장과 견고성

전송된 메시지가 목적하는 노드에 끝내 도달되지 않는 것을 링크 고장이라고 정의한다. 성능에 기반을 둔 분산선거 연구에서 노드는 기능을 하지만 잘못된 투표를 던지는 것을 노드고장이라고 하였는데 일부 고장난 노드가 잘못 투표를 하더라도 각 선거전략에 의해 뽑힌 리더의 성능은 얼마나 되는가를 연구하였다. 본 논문에서 링크고장이 발생하면 각 노드의 선호도가 변화하는 데도 불구하고 뽑히는 리더의 성능은 과연 얼마나 되는지 즉 얼마나 최적리더의 성능에 근접하는지와 링크고장이 발생하였을때 각 선거전략이 최적리더를 뽑을 확률이 얼마인가를 모의실험을 통하

여 구하여서 3절에서 정의한 견고성이 가장 좋은 선거전략은 어느 것인지를 연구한다.

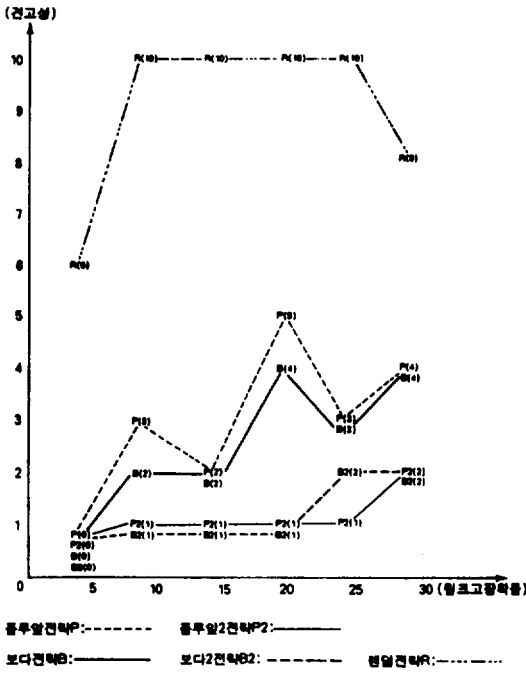
### 5. 시뮬레이션

확률론적으로 일반화 할 수 없는 다른전략은 랜덤가중치 행렬위에서 심도있는 시뮬레이션을 수행시켰다. 시뮬레이션을 위하여 랜덤숫자를 이용하여 가상적인 네트워크를 나타내는 가중치 그래프를 구성하였다. 노드의 갯수는 6개, 9개, 12개의 경우로 나누어서 구성하였으며, 각 경우마다 노드와 노드사이에 링크가 존재할 확률을 30%, 60%, 90%로 역시 랜덤 숫자를 부여하였다. 가중치 분포로 지수분포, 균등분포등 여러가지 경우로 줄 수 있지만 가중치에 대한 난수의 확률 분포가 성능에 별 영향을 주지 않으므로 [16], 본 논문에서는 각 링크의 가중치는 임의의 난수를 발생시켜서 양방향으로 서로 다른 값을 부여하였다. 링크가 고장날 확률은 5%에서 30%까지 5%씩 간격으로 주어지는 경우를 고려하였다.

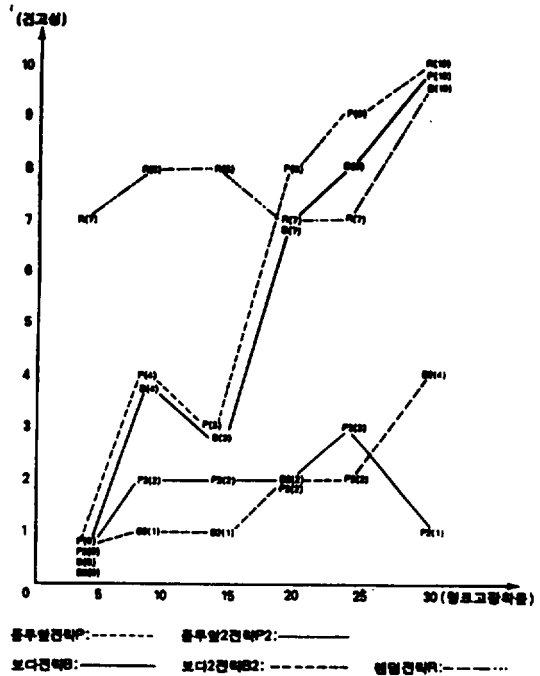
링크의 고장이 없는 경우를 시뮬레이션 하기 위하여 노드당 10,000개의 가중치 그래프를 구성하고 이들로 부터 각 선거전략에 의하여 구해지는 리더의 상대오차를 평균하였으며, 링크의 고장을 부여한 경우에도 노드당 10,000개의 가중치 그래프를 구성하고, 이들로 얻어지는 결과를 평균하였다. 구성된 가중치그래프에서 2장에서 기술한 선거전략으로 리더를 선출한다. 시스템의 최악의 리더, 그리고 최적의 리더를 구하고 이들을 이용하여 2장에서 설정한 리더의 상대오차를 구하고 앞에서 정의된 견고성 값중에서 상대오차를 이용한 견고성 값을 각 선거전략별, 노드 갯수별(6,9,12), 링크 고장 확률별로 구하였다. 링크의 고장이 있는 경우에 링크가 고장날 확률 분포에 따라 발생된 랜덤 숫자를 이용하여 링크가 고장난 것으로 결정하여 가중치 그래프를 변경하였다.

#### 5.1 시뮬레이션 및 결과

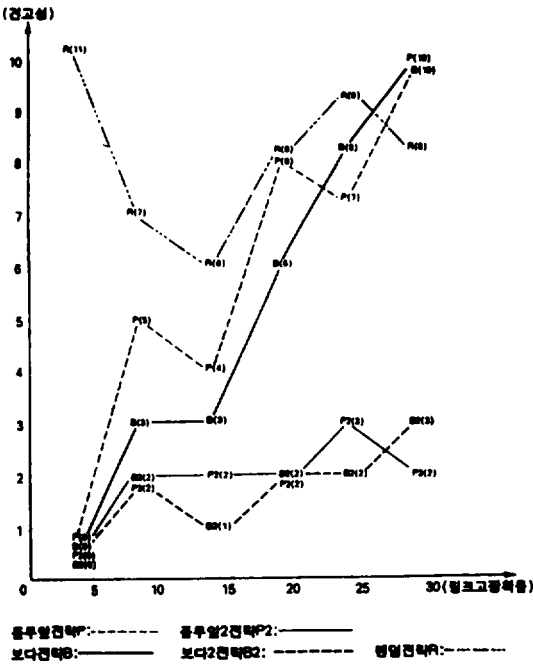
노드별(6, 9, 12) 및 연결확률별(30, 60, 90) 링



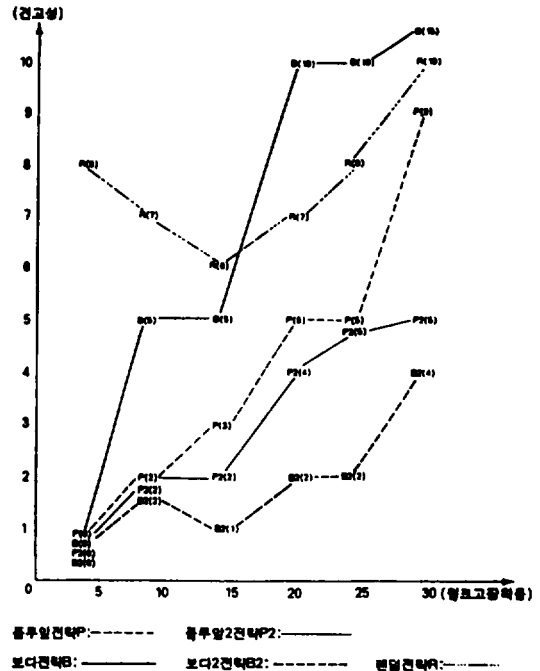
(그림 1) 노드 6개 연결확률 30%  
(Fig. 1) # {node}:6 connection-prob:30%



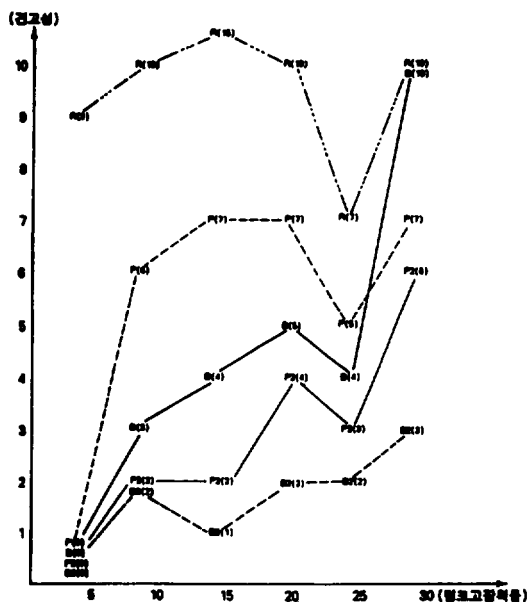
(그림 2) 노드 6개 연결확률 60%  
(Fig. 2) # {node}:6 connection-prob:60%



(그림 3) 노드 6개 연결확률 90%  
(Fig. 3) # {node}:6 connection-prob:90%

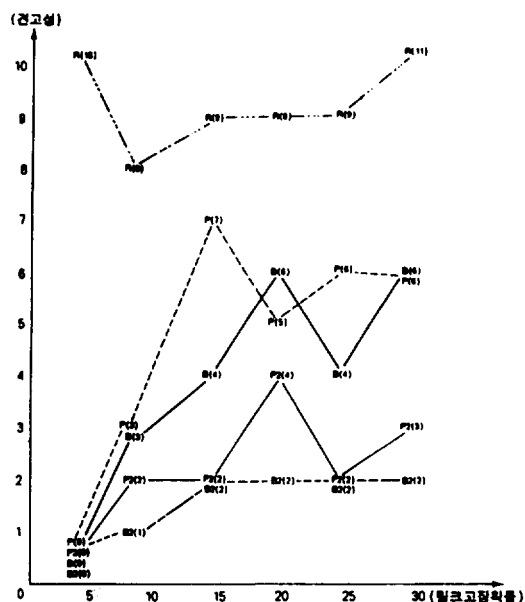


(그림 4) 노드 9개 연결확률 30%  
(Fig. 4) # {node}:9 connection-prob:30%



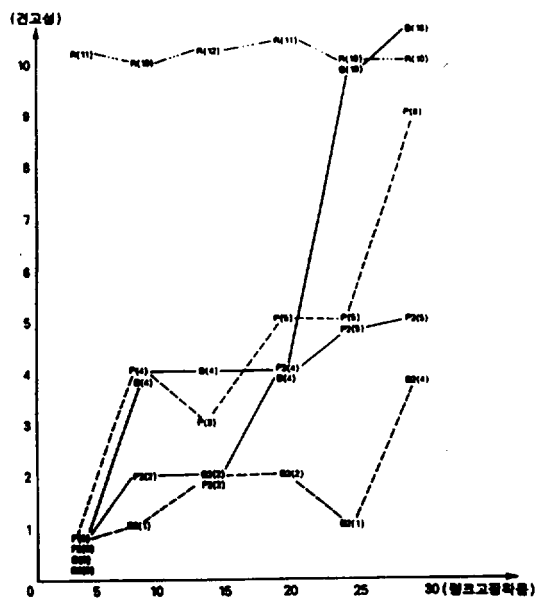
플루일전략P: ————— 플루일2전략P2: —————  
 보다전략B: ————— 보다2전략B2: - - - - - 연립전략R: ······

(그림 5) 노드 9개 연결확률 60%  
(Fig. 5) # {node}:9 connection-prob:60%



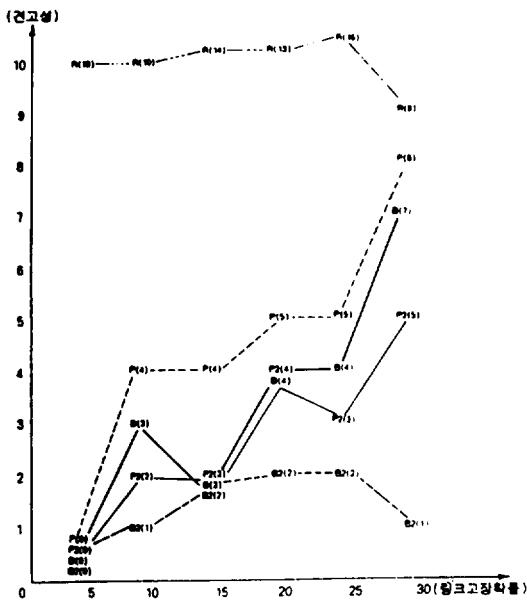
플루일전략P: - - - - - 플루일2전략P2: —————  
 보다전략B: ————— 보다2전략B2: - - - - - 연립전략R: ······

(그림 6) 노드 9개 연결확률 90%  
(Fig. 6) # {node}:9 connection-prob:90%



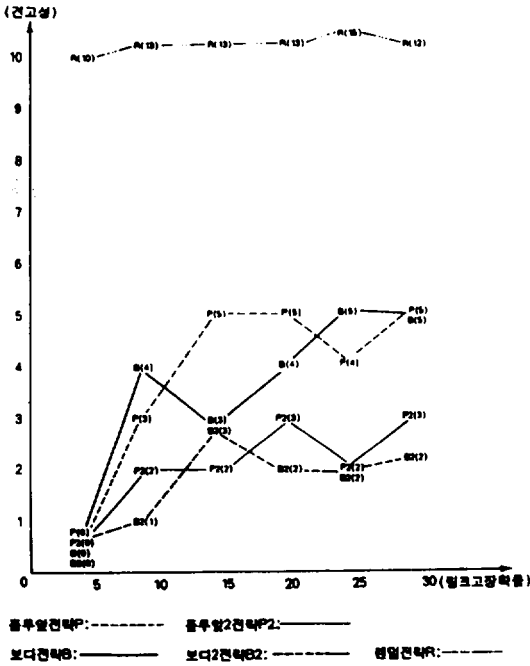
플루일전략P: ————— 플루일2전략P2: - - - - -  
 보다전략B: ————— 보다2전략B2: - - - - - 연립전략R: ······

(그림 7) 노드 12개 연결확률 30%  
(Fig. 7) # {node}:12 connection-prob:30%



플루일전략P: - - - - - 플루일2전략P2: —————  
 보다전략B: ————— 보다2전략B2: - - - - - 연립전략R: ······

(그림 8) 노드 12개 연결확률 60%  
(Fig. 8) # {node}:12 connection-prob:60%



(그림 9) 노드 12개 연결확률 90%  
(Fig. 9) # {node}:12 connection - prob:90%

크고장확률(5%, 15%, 20%, 25%, 30%)별로 랜덤 가중치 행렬을 균등난수를 발생시켜서 만든다. 단, 난수를 발생시켜 링크고장이 발생을 모의 실험을 하였다.

링크고장 확률은 0~30%까지 5%씩 증가시키면서 각 선거전략의 상대오차를 이용한 견고성을 구한 결과는 다음과 같다.

(그림 1), (그림 2), (그림 3)은 노드 6개, 연결확률 30%일때 링크고장확률을 5%에서 30%까지 5% 간격으로 하였을때 앞에서 정의된 견고성을 나타내고 있다. 링크 고장확률이 15% 정도까지는 거의같은 견고성을 유지하다가 20%부터는 플루앞전략, 보다전략이 견고성 값이 커지는데 보다2전략은 25%까지도 견고성을 그대로 유지하고 있고 30%에 가서 조금 커지는 경향을 보이고 있다.

(그림 3)에서 연결확률이 90%로 커질때는 링크 고장확률 30%인 경우에서는 보다 전략과 플루앞 전략의 견고성 값이 크게 올라가게 된다.

(그림 4)와 (그림 5)에서 노드 6개일때와 거의 같은 경향을 보이고 있는데 연결확률이 높아지면 플루앞전략과 보다전략의 견고성 값은 커지고 있고 보다2전략은 여전히 좋은 견고성 값을 유지하고 있다.

(그림 6)에서 연결확률이 90%로 올라갈때 링크 고장확률이 30%일때 더 견고성값이 작아지고 있음을 보이고 있다.

(그림 7), (그림 8)에서는 (그림 4), (그림 5)와 거의 같은 결과를 보이고 있다. 링크고장 확률 25% 이상일때에는 보다전략의 견고성 값이 랜덤전략보다 커지고 있고 링크 고장확률이 30%일때는 보다전략과 플루앞전략이 견고성 값이 랜덤전략과 거의 같게 된다. 그러나 보다 2전략은 견고성을 그대로 유지하고 있다.

(그림 9)는 링크 고장확률 25%일때까지는 견고성이 (그림 7), (그림 8)과 거의 같은 결과를 보이고 있고 링크 고장확률이 커져도 플루앞전략, 보다전략이 (그림 7), (그림 8)보다는 견고성 값이 더 작아지고 있다.

## 6. 결 론

이 논문에서 성능의 정의를 논문[16]과는 달리 양방향 지연값을 선호도로 하여 여러 링크 고장 확률을 일정하게 하는 경우에 본 논문에서 제안한 양방향 지연값을 선호도로 한 보다2선거전략이 논문[16]에서 견고성이 우수한 선거전략이라고 밝혀진 플루앞, 보다전략보다 견고성이 더 좋음을 보여 주었다.

향후 연구에서는 확률론적인 측면에서 링크의 고장과 선출되는 리더와 관계를 연구하고자 하며, 이론적인 결과와 본 논문의 시뮬레이션의 결과를 비교하여 입증하고자 한다. 또한 노드 수를 증가하고 연결확률을 더 세분해서 실험하여 링크의 고장과 선거전략의 견고성과의 관계를 보다 더 분명히 하고자 한다.

## 참 고 문 헌

[1] Hosame Abu-amara, "Fault-Tolerant Distrib-

uted Algorithm for Election in Complete Networks”, IEEE Trans. on computer Vol. 37, No. 4. pp. 449-453, April, 1988.

[2] B. Awerbuch, “Optimal Distributed Algorithms for Minimum weight Spanning Tree, Counting, Leader Election and related problems”, ACM STOC, pp. 230-240, 1987.

[3] K. Birman. et al, “Implementing Fault Tolerant Distributed objects”, IEEE Trans. on Software Engineering., Vol. SE-11, No. 6 pp. 502-508, Jun, 1985.

[4] P. H.Enslow, “What is a distributed data processing system?,” computer, pp. 13-21, Jan, 1978.

[5] Michael J. Fisher and Nancy A.Lynch and Michael S.PATERSON “Impossibility of Distributed Consensus with one Faulty” Journal of ACM, Vol,32, No. 2, pp. 374-382, April, 1985.

[6] G. Fredrickson and N. Lynch, “Electing a Leader in a Synchronous Ring”, JACM, Vol. 34, No. 1, pp. 98-115, Jan, 1987.

[7] E. Gafni, “Improvements in the Time complexity of Two Message-Optimal Algorithms”, ACM STOC, pp. 175-207, 1985.

[8] H. Garcia-Molina, “Election in a Distributed computer system”, IEEE Trans. on computer, Vol, c-31, No. 1, pp. 48-59, Jan, 1982.

[9] Oded Goldreich and Liuba shrira, “On the complexity of computation in the presence of link failures the cass of a ring;8, Distributed Computing 5:121-131, 1991.

[10] H. Moulin, “The strategy of Social choice”, North Holland publishing Company, Advanced Textbooks in Economics, Amsterdam, 1983.

[11] Alon Itai, Shay Kutten, Yaron Wolf Stahl, and Shmuel Zaks “Optimal Distributed t-resilient Election in complete networks”, IEEE Trans. On S. E, Vol. 16, No. 4, pp. 415-420 April, 1990.

[12] Shay kutten and Yaron Wolfsthal, “Finding a leader in a distributed system where elements may fail”, proc. 17th IEEE Anrou. Electronics and Aerospace Conf., Washington, DC, pp. 101-105, Sept, 1984.

[13] D. Menasce, R. Muntz and J. Popek, “A Locking protocol for resource Coordination in Distributed Databases”, ACM TODS, Vol. 5, No. 2, pp. 103-138, June, 1980.

[14] L. Shrira and O. Goldreich, “Electing a leader in a ring with link failures,” Acta Inform., Vol 24, pp. 79-91, 1987.

[15] R. G. Smith, “The contract net protocol: High level communication and control in a distributed problem solver,” in proc. 1st INT. conf. Distributed comput. syst, Huntsville, AL, pp. 185-192, Oct, 1979.

[16] Singh, Suresh Prakash, “Preference-Based Leader Election in Distributed systems”, ph. D. Dissertation, Dept. of Computer and Information Science, University of Massachusets, Sept, 1990.

[17] E. Korach, S. Kutten, and S. Moran, “A Modular Technique for the Design of Efficient Distributed Leader Finding Algorithms”, ACM Transactions on Programming Languages and Systems, Vol. 12, No. 1, pp. 84-101, January, 1990.



진 기 병

1974년 서울대학교 사범대학 수학교육과 졸업(이학사)  
 1986년 숭실대학교 산업대학원 전자계산학과 졸업(공학석사)  
 1994년 아주대학교 대학원 컴퓨터 공학과 졸업(공학박사)  
 1980년~현재 숭실대학교 전자계산원 전임강사

관심분야 : 분산 시스템, 운영 체제, 프로그래밍 언어





**최 경 회**

1976년 서울대학교 사범대학  
수학교육과 졸업(이학사)  
1979년 블란서 ENSEIHT 그랑  
데폴 정보공학과(공학석사)  
1982년 블란서 폴자바티에대  
정보공학과(공학박사)  
1982년~현재 아주대학교 공과  
대학 컴퓨터공학과 교수

관심분야: 운영 체제, 분산 시스템, 실시간 시스템



**이 형**

1964년 서울대학교 사범대학  
수학교육과 졸업(이학사)  
1971년 성균관대학교 경영대학  
원(전자계산학 전공)  
1992년 조선대학교대학원 전산  
전공(공학박사)  
현재: 대전대학교 공과대학 정  
보공학과 교수

관심분야: 컴퓨터 통신, 그래픽스