

# 수중 센서 네트워크를 위한 클러스터 헤드 오류 감지 기법\*

정진만<sup>01</sup> 어상일<sup>1</sup> 민홍<sup>1</sup> 김봉재<sup>1</sup> 조유근<sup>1</sup> 홍지만<sup>2</sup> 허준영<sup>3</sup>

서울대학교 컴퓨터공학부<sup>1</sup>

송실대학교 컴퓨터학부<sup>2</sup>

한성대학교 컴퓨터공학부<sup>3</sup>

{jmjung<sup>0</sup>,sieo,hmin,bjkim,cho}@os.snu.ac.kr<sup>1</sup>, jiman@ssu.ac.kr<sup>2</sup>, jyheo@hansung.ac.kr<sup>3</sup>

## Cluster Head Failure Detection for Underwater Wireless Sensor Networks

Jinman Jung<sup>01</sup> Sangil Eo<sup>1</sup> Hong Min<sup>1</sup> Bongjae Kim<sup>1</sup> Yookun Cho<sup>1</sup> Jiman Hong<sup>2</sup> Junyoung Heo<sup>3</sup>

School of Computer Science and Engineering, Seoul National University<sup>1</sup>

School of Computing, Soongsil University<sup>2</sup>

Department of Computer Engineering, Hansung University<sup>3</sup>

### 1. 서론

헤드 노드의 오류를 정확하고 신속하게 감지하여 불필요한 에너지 소비를 줄일 수 있는 효율적인 기법을 제안한다. 제안 기법은 오류 감지 시간을 더 빠르게 하기 위해 클러스터 헤드와의 거리를 TDMA 타임 슬롯 스케줄링에 고려하였다. 만약 일반 노드들이 클러스터 헤드의 오류 발생 여부를 신속하게 감지하지 못한다면 많은 시간과 에너지를 소모하여 의미 없는 데이터를 전송하게 된다.

### 2. 지연 시간 최소화 오류 감지 알고리즘

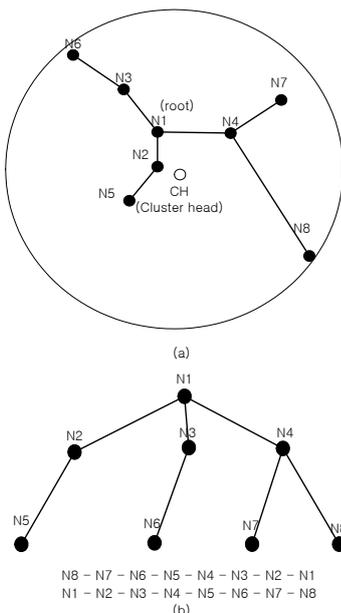


그림 1 기존 기법

전통적인 오류 감지 기법은 각 클러스터의 참여 노드들이 클러스터 헤드의 하트비트의 유무를 통해 독립적으로 결정하였다. 미리 정한 수만큼의 하트비트가 오지 않았다면, 클러스터 헤드가 이상이 있다고 결정하는 방법이다. [1]에서는 클러스터 내 참여 노드들이 독립적으로 판단한 정보를 상호 교환하여 최종 결정하는 방법이다. 이 기법은 클러스터 헤드가 오류가 생길 경우 1-홉 통신이 어렵기 때문에 미리 셋업 단계(setup stage)에서 그림 1 (a)와 같이 브로드캐스트(broadcast) 트리를 만들어 관리한다. 클러스터 멤버들이 독립적으로 판단한 클러스터의 상태 정보는 일반 데이터 송수신 과정에서 포함되어(piggybacked) 전달되며, 트리의 깊이(Depth)가 가장 큰 노드들부터 깊이가 작은 방향으로 전달되며 정보가 병합된다. 최종적으로 루트 노드에게 전달되면, 트리에 속한 멤버들로부터 수집된 정보를 통해 헤드의 오류 여부가 다수결로 결정된다. 이 결과는 모든 노드가 알아야 하기 때문에 같은 방법으로 다시 깊이가 큰 방향으로 전달되며, 이 시간은 두 프레임 정도가 걸린다. 그림 1의 (b)에서와 같이 TDMA의 타임슬롯 스케줄링을 너비 우선 탐색(Breadth first search)으로 하여 두 프레임에 감지 시간을 보장하려고 하였다. 하지만, 이 기법은 TDMA 스케줄링에서 수중 지연 시간을 고려하지 않은 스케줄링 기법을

\* 이 연구를 위해 연구장비를 지원하고 공간을 제공한 서울대학교 컴퓨터연구소에 감사 드립니다

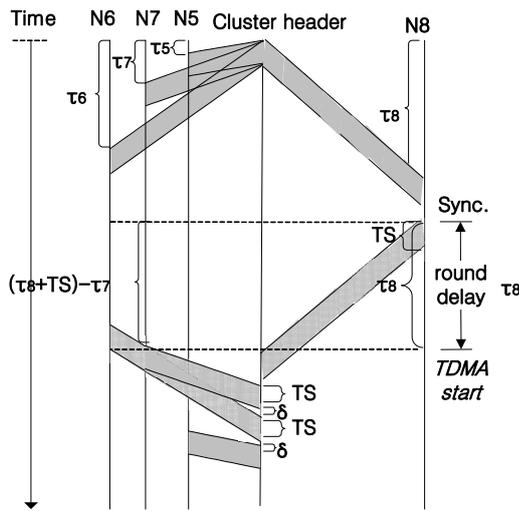


그림 2. 라운드 지연 시간

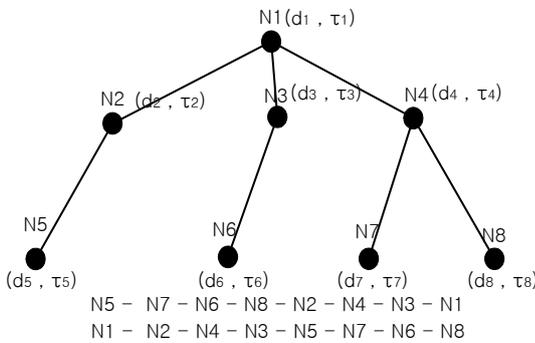


그림 3 제안 기법

사용하여 불필요한 지연 시간을 유발 시킨다. 라운드 첫 프레임에서 할당받은 노드는 동기화 된 직후 전송이 시작되므로 그림 2와 같이 지연 시간을 가지며, 매 라운드마다 발생하므로 라운드 지연(Round Delay)이라고 정의한다.

매 라운드마다 첫 타임슬롯에 할당된 노드  $N_x$ 와 클러스터 헤드  $CH$ 와의 지연시간  $\tau_x$  이 누적되어 진행된다.  $\tau_v^k$  를  $k$  번째 라운드에서  $v$ 노드와 클러스터 헤드와의 지연시간 이라고 할 때,  $n$ 번의 라운드에 유발되는 총 지연 시간은

$$\sum_{k=0}^n E_{v \in T_h}(\tau_v^k)$$

이다. 따라서, 그림 1의 (b)와 같은 기존의 스케줄링 방법은 라운드 지연 문제를 유발 시키므로 이 문제를 해결하기 위해 클러스터 헤드와 노드와의 거리를 고려하는 알고리즘을 제안한다. 차수가  $h$ 이고 노드가  $m$ 인 브로드캐스트 트리  $T$ 가 있고,  $T$ 에서 깊이가  $j$ 인 노드들의 집합을  $T_j(j = 0, 1, \dots, h)$ 라 하자( $T_0$ 는 루트노드를 의미한다). 또한, 클러스터 헤드  $CH$ 는 도착시간(Time-of-Arrival)을 이용하여 노드  $N_i(i = 1, 2, \dots, m)$ 와의 지연시간과 거리( $\hat{\tau}_i, \hat{d}_i$ ) 알 수 있다고 하자. 각  $T_j$ 에 속하는 노드들을  $CH$ 와  $\hat{\tau}_i$ 가 작은 순(즉 거리가 가까운 순)으로 정렬하고, 정렬된  $T_j$ 들을 트리 깊이가 작아지는 순으로 스케줄링한다. 그림 3와 같이  $T_2$ 에 속하는 N5, N6, N7, N8 중  $\hat{\tau}_i$ 가 작은 N5, N7, N6, N8 순으로 변경되고, 마찬가지로  $T_1$ 에 속하는 N2, N3, N4는  $CH$ 와 거리가 가까운 N2, N4, N3 순으로 변경된다. 마지막으로  $T_0$ 의 원소인 루트노드 N1가 추가된다. 제안된 스케줄링 기법에서는  $T_h$ 에서 지연 시간이 최소인 노드를 선택하였으므로  $\tau_{s_1} = \min_{v \in T_h} \hat{\tau}_v$  이 된다. 이때 클러스터 내 노드의 수가 증가할수록  $T_h$ 에 속하는  $v$ 가 많아지기 때문에 지연 시간은 더욱 작아진다. 따라서, 제안하는 지연 시간 최소화 알고리즘은  $n$ 번의 라운드가 지나면 총 라운드 지연 시간은

$$\sum_{k=0}^n \min_{v \in T_h} \tau_v^k$$

### 3. 결론

본 논문에서는 클러스터 헤드의 오류 감지를 위해 지연시간을 고려한 트리 기반의 알고리즘을 제안하였다. 노드들의 클러스터 헤드와의 지연시간을 고려한 타임슬롯 스케줄링을 통해 오류 감지 시간을 줄였다. 또한 노드의 수가 증가할수록 최고 깊이의 노드들의 수가 증가하고, 그중에서 최소의 지연 시간을 갖는 노드를 선택하므로 총 라운드 지연 시간이 크게 줄어든다.

### 4. 참고 문헌

[1] P. Wang, J. Zhen and C. Li, "Cooperative fault-detection mechanism with high accuracy and bounded delay for underwater sensor networks", Wireless Communications and Mobile Computing, p.143-153, 2009.