

I. 서론

분산 무선 네트워크에서 빠른 컨센서스를 위한 커버리지 제어

최 현 호[°]

Coverage Control for Fast Consensus in Distributed Wireless Networks

Hyun-Ho Choi[°]

요 약

컨센서스 알고리즘은 노드의 커버리지가 넓어질수록 협력하는 이웃 노드 수가 증가하여 빠르게 수렴하지만, 분산 무선 네트워크에서는 커버리지가 넓어지면 증가한 협력 이웃 노드 간에 접속 충돌로 인하여 정보 공유 지연이 증가한다. 따라서 노드 커버리지의 크기에 따라 컨센서스 알고리즘의 수렴 속도와 네트워크상의 정보 공유 지연 간에 트레이드오프가 존재한다. 본 논문에서는 이러한 트레이드오프 관계를 고려하여 네트워크 규모에 따라 컨센서스 달성 시간을 최소화하는 커버리지 제어 방법을 제시한다.

Key Words : Consensus, Coverage control, Medium access control, Distributed network

ABSTRACT

The consensus algorithm has a faster convergence speed as the number of cooperating neighbors increases, but the information sharing delay in the wireless network increases due to access collisions as the number of cooperating neighbors increases. Therefore, there exists a tradeoff between these two performances according to node's coverage. In this paper, we present a method of coverage control that minimizes consensus time according to network scale.

컨센서스(consensus)란 각 노드의 상태에 의존하는 어떠한 값이 서로 일치하게 됨을 의미하며, 컨센서스 알고리즘은 컨센서스를 위해 각 노드가 네트워크상의 이웃 노드들과 관련 정보를 공유하고 처리하는 상호 규칙을 말한다¹⁾. 컨센서스 알고리즘은 분산 시간 및 주파수 동기화, 분산 공평 자원 할당, 빠른 분산 데이터 공유, 센싱 정보의 분산 퓨전 등 컨센서스를 요구하는 다양한 통신 네트워크 분야에 접목되어 사용되어 왔다²⁾.

컨센서스 알고리즘의 주요 성능 지표는 얼마나 빨리 컨센서스를 달성할 수 있는냐에 대한 것으로, 이는 컨센서스 알고리즘의 제어 파라미터와 네트워크 환경에 영향을 받는다. 지금까지의 컨센서스 관련 연구는 주로 알고리즘 자체의 수렴 여부와 수렴 속도에 중점을 두었지, 실제 네트워크 환경에서 노드 간 상태 정보 교환에 의해 발생하는 전송 지연에 대해서는 고려하지 않았다^{1,3)}. 컨센서스 알고리즘의 수렴 속도는 노드 간 정보 공유가 많을수록, 즉 노드 당 이웃 노드 수가 많을수록 빨라진다. 따라서 컨센서스를 이루고하는 네트워크상의 모든 노드들이 다 같이 협력하는 것이 컨센서스 달성 시간을 최소화하는 이상적인 방법이 된다. 하지만 컨센서스 알고리즘이 수행되는 실제 네트워크 환경을 고려하면 상호 정보 교환을 수행하는 이웃 노드 수가 증가할수록 접속 충돌로 인한 전송 지연이 증가하여 전체 컨센서스 달성 시간이 오히려 증가한다^{2,3)}. 따라서 실제 환경에서 컨센서스 알고리즘은 협력 이웃 노드 수(즉, 노드의 커버리지)에 따라 트레이드오프(tradeoff) 성능을 갖는다.

본 논문에서는 slotted ALOHA 프로토콜을 사용하는 분산 무선 네트워크 환경에서 노드 커버리지에 따른 컨센서스 알고리즘의 트레이드오프 성능을 분석한다. 이를 기반으로 컨센서스 달성 시간을 최소화하는 최적의 이웃 노드 수를 도출하고, 네트워크 규모에 따른 커버리지 제어 방법을 제시한다.

II. 컨센서스 알고리즘

컨센서스 알고리즘은 연속/이산시간에 따라, 수신

※ 이 논문은 2013년도 정부(교육과학기술부)의 재원으로 한국연구재단의 기초연구사업 지원을 받아 수행된 것임(2011-0025424)

•° First Author and Corresponding Author : Hankyong National University, Department of Electrical, Electronic and Control Engineering, and Institute for Information Technology Convergence, hhchoi@hknu.ac.kr, 정회원

논문번호 : KICS2014-04-128, Received April 9, 2014; Revised May 9, 2014; Accepted May 12, 2014

한 노드 상태 값의 처리 방식에 따라 몇가지 형태가 존재하는데, 본 논문에서는 다음 식으로 표현되는 이산시간 기반의 컨센서스 알고리즘을 고려한다^[1].

$$x_i(k+1) = \frac{1}{1+N_i} \left(x_i(k) + \sum_{j \in N_i} x_j(k) \right) \quad (1)$$

여기에서 $x_i(k)$ 는 k 번째 시간에 노드 i 의 상태 값, N_i 는 노드 i 와 통신 가능한 이웃 노드의 집합, N_i 는 노드 i 의 이웃 노드 개수를 나타낸다. 이 알고리즘에 따르면 각 노드는 현재 자신과 이웃 노드들의 상태 값의 평균으로 다음 상태 값을 결정한다. 즉, 각 노드는 주변 이웃 노드의 상태 정보만을 가지고 분산적으로 지역적인 평균(local average)을 취한다. 이러한 반복 동작은 k 가 증가함에 따라 모든 노드의 상태 값이 동일한 값으로 수렴하게 되는(즉, $x_i \approx x_j \quad \forall i, j$) 컨센서스를 이루게 된다.

본 논문에서는 식 (1)의 컨센서스 알고리즘의 수렴 여부와 수렴 속도에 대한 이론적 결과를 바탕으로, 노드 간 연결 토폴로지와 노드별 상태 값 분포 등의 환경 변수를 반영한 시뮬레이션을 통하여 컨센서스 달성 시간을 구한다.

III. 분산 무선 네트워크에서 컨센서스 달성 시간

컨센서스 알고리즘은 노드 간 상태 정보의 교환을 필요로 하므로 이를 네트워크상에서 수행할 때 노드 간 전송 지연이 발생한다. 고려하는 분산 무선 네트워크에서는 고정된 작은 크기의 상태 정보를 전송하기에 효율적이며 간단한 매체 접속 프로토콜로써 slotted ALOHA의 사용을 가정한다. 이를 기반으로 컨센서스 알고리즘 동작시 발생하는 전송 지연을 도출하고 컨센서스 달성 시간에 반영한다.

고려하는 slotted ALOHA 프로토콜은 프레임당 m 개의 타임 슬롯(time slot)을 제공하며, 한 타임 슬롯의 길이는 노드의 상태 정보를 전송하는데 필요한 시간 길이가 된다. 프레임에 동기화된 각 노드는 컨센서스 알고리즘의 반복 동작 마다 m 개의 타임 슬롯 중 하나를 랜덤하게 선택하여 자신의 상태 정보(x_i)를 전송한다. 만약 한 타임 슬롯에 두 개 이상의 노드가 동시에 전송을 한다면 충돌로 간주되어 전송에 실패하고 다음 프레임에서 재전송을 시도한다. 한 노드의 커버리지에 N_i 개의 이웃 노드가 존재하여 서로 경쟁할 때, x 개의 노드가 전송에 성공할 확률은 다음과 같이 계산된다.

$$p(x) = \begin{cases} \frac{N_i! \binom{m}{x} \pi(x)}{m^{N_i}} & \text{if } 0 \leq x \leq \min\{N_i - 2, m - 1\} \\ \frac{N_i! \binom{m}{x}}{m^{N_i}} & \text{if } x = N_i \text{ and } m \geq N_i \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

여기에서 $\pi(x)$ 는 경쟁하는 N_i 개의 노드 중 x 개의 노드가 x 개의 타임 슬롯에 각각 하나씩 접속하여 성공하면서 나머지 $(N_i - x)$ 노드는 i 개의 슬롯에 동시 접속하여 실패하는 모든 경우의 수를 나타내며, 아래와 같이 주어진다.

$$\pi(x) = \sum_{i=1}^{\min\{\lfloor \frac{N_i-x}{2} \rfloor, m-x\}} \left\{ \binom{m-x}{i} \sum_{a_1, a_2, \dots, a_i \geq 2}^{a_1 + a_2 + \dots + a_i = N_i - x} \frac{1}{\prod_{j=1}^i a_j!} \right\} \quad (3)$$

식 (2)을 이용하여 평균 전송 성공 확률은 다음과 같이 구해진다.

$$P_s = \frac{1}{N_i} \sum_{x=0}^{N_i} xp(x) \quad (4)$$

따라서 N_i 개의 노드가 경쟁할 때 전송 성공에 걸리는 평균 지연은 다음과 같이 계산된다.

$$D = mT_s \sum_{x=1}^{\infty} x(1-P_s)^{x-1} P_s = \frac{mT_s}{P_s} \quad (5)$$

여기에서 T_s 는 타임 슬롯의 길이를 나타낸다.

컨센서스 알고리즘에서는 네트워크의 모든 노드들이 자신의 주변 노드들과 경쟁하면서 상태 정보를 전송해야 한다. 커버리지 내에 있는 N_i 개의 모든 노드가 이를 수행할 때 정보 교환에 걸리는 평균 시간은 식 (5)의 평균 지연에 곱하여 $N_i D$ 가 된다. 따라서 최종적으로 컨센서스 달성에 걸리는 시간($T_{consensus}$)은 시뮬레이션을 통해 구한 수렴에 필요한 알고리즘의 반복 횟수(N_{iter})와 정보 공유 지연 값($N_i D$)의 곱으로 다음과 같이 결정된다.

$$T_{consensus} = N_{iter} N_i D \quad (6)$$

IV. 시뮬레이션 결과 및 고찰

컨센서스 알고리즘의 동작을 위해 정규(regular) 네트워크 토폴로지를 가정하며, 각 노드의 커버리지에 해당하는 협력 이웃 노드 수를 변화시키면서 컨센서스 달성 시간을 구한다. 초기에 각 노드의 상태 값은 0~100 사이에서 균일(uniform)하게 결정되며, 컨센서

스 알고리즘의 수렴조건은 모든 노드의 현재 상태 값과 이론적인 수렴 값의 차이 벡터의 2-norm 값이 10^{-2} 보다 작은 경우로 설정된다^[2].

그림 1은 전체 컨센서스 참여 노드 수(N)가 100이고 제공되는 타임 슬롯의 수(m)가 15일 때 협력 이웃 노드 수에 따른 컨센서스 알고리즘의 반복 횟수(N_{iter}), 평균 정보 공유 지연, 컨센서스 달성 시간을 보여준다. 이웃 노드 수가 증가할수록 알고리즘의 반복 횟수는 기하급수적으로 줄어들지만 노드 간 정보 공유 지연은 기하급수적으로 증가한다. 이러한 트레이드오프로 관계로 인하여 컨센서스 달성은 아래로 볼록한 (convex) 형태가 되며, 이 경우 이웃 노드 수가 12일 때 컨센서스 달성 시간이 최소화됨을 보여준다.

그림 2(a)는 컨센서스에 참여하는 노드 수에 따른 최적 협력 이웃 노드 수(최적 노드 커버리지)를 보여준다. 참여 노드 수가 적을 때는 최적 이웃 노드 수가 선형적으로 증가하다가 어느 이상이 되면 일정한 값으로 수렴하게 된다. 즉, 네트워크 규모가 작을 때에는 전송 지연이 발생하더라도 전체 노드가 협력하여 한 번에 컨센서스를 달성하는 것이 성능에 좋지만, 네트워크 규모가 어느 이상으로 커지면 정보 공유 지연

이 성능에 더 나쁜 영향을 미치므로 이웃 노드 수를 제한하는 것이 좋을 수 있다. 따라서 네트워크 규모에 따라 일정 참여 노드 수까지는 전체 노드 간에 협력하도록 전체 커버리지(full coverage)를 유지하다가, 이 한계 노드 수를 넘어 더 많은 노드가 참여하면 커버리지를 줄여 일정 이웃 노드만 포함되도록 해야 한다. 이때 노드 커버리지는 참여 단말 수에 상관없이 일정 값으로 수렴하고, 오직 할당 슬롯 개수에만 의존한다.

그림 2(b)는 할당된 타임 슬롯 개수에 따른 전체 커버리지를 유지하는 한계 노드 수와 최적 이웃 노드 수를 보여준다. 슬롯 개수가 증가할수록 정보 공유 지연이 줄어들므로 전체 커버리지를 유지하는 한계 노드 수 및 최적 이웃 노드 수가 같이 증가한다. 따라서 컨센서스 달성을 최소화하기 위해서는 참여 노드 수와 제공되는 슬롯 개수에 따라 한계 노드 수까지는 전체 커버리지를 유지하다가 그 이상으로 참여 노드 수가 증가하면 최적 이웃 노드 수만 포함되도록 노드별 커버리지를 제어해야 한다.

V. 결론

본 논문에서는 무선 네트워크 환경에서 협력 이웃 노드 수에 따른 컨센서스 알고리즘의 트레이드오프 성능을 분석하여 컨센서스 달성 시간을 최소화하는 최적 이웃 노드 수를 도출하였다. 이를 통해 컨센서스 참여 노드 수가 적을 때에는 전체 커버리지를 유지하다가 노드 수가 어느 이상으로 증가하면 일정 수의 이웃 노드만 포함되도록 커버리지를 줄이는 것이 최적 커버리지 제어 방법임을 알 수 있었다. 추후에는 다양한 네트워크 환경에서 다른 매체 접속 프로토콜을 적용하여 컨센서스 알고리즘의 성능을 고찰할 계획이다.

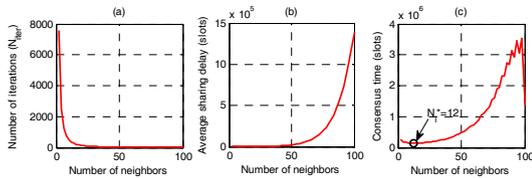


그림 1. 협력 이웃 노드 수에 따른 (a) 수렴을 위한 컨센서스 알고리즘의 반복 횟수, (b) 평균 정보 공유 지연, (c) 컨센서스 달성 시간 ($N=100, m=15$ 일 때)
Fig. 1. (a) number of iterations for convergence, (b) average information sharing delay, and (c) consensus time versus number of neighbors (when $N=100, m=15$)

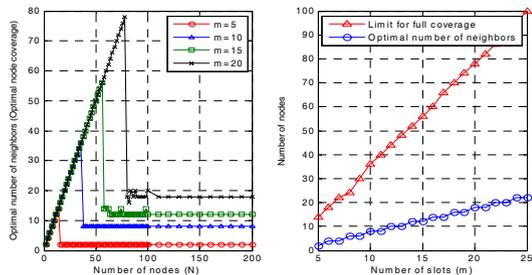


그림 2. (a) 참여 노드 수에 따른 최적 이웃 노드 수 (최적 노드 커버리지)와 (b) 할당된 타임 슬롯 개수에 따른 전체 커버리지를 유지하는 한계 노드 수 및 최적 이웃 노드 수
Fig. 2. (a) optimal number of neighbors versus number of participating nodes and (b) limit number of nodes for full coverage and optimal number of neighbors versus number of time slots allocated

References

- [1] R. Olfati-Saber, J. Fax, and R. Murray, "Consensus and Cooperation in Networked Multi-Agent Systems," in *Proc. IEEE*, vol. 95, no. 1, pp. 215-233, Jan. 2007.
- [2] H.-H. Choi, "Understanding of synchronization in nature and its engineering applications," *KICS Inf. Commun. Mag.*, vol. 31, no. 1, pp. 71-81, Jan. 2014.
- [3] R. Olfati-Saber and R. M. Murray, "Consensus problems in networks of agents with switching topology and time-delays," *IEEE Trans. Automatic Control*, vol. 49, no. 9, pp. 1520-1533, Sept. 2004.