

# 무선 센서 네트워크에서 에너지 효율성을 위한 클러스터 기반의 연속 객체 예측 기법

정회원 이완섭\*, 홍형섭\*\*, 종신회원 김상하\*\*\*°

## Cluster-based Continuous Object Prediction Algorithm for Energy Efficiency in Wireless Sensor Networks

Wanseop Lee\*, Hyungseop Hong\*\* *Regular Members*, Sang-Ha Kim\*\*\*° *Lifelong Member*

### 요약

무선 센서 네트워크에서 넓은 지역의 현상 및 움직임을 지속적으로 추적, 관찰하기 위해 에너지 효율성은 가장 중요한 요소이다. 이러한 망에서 에너지를 절약하기 위해 센서노드들에 대한 선택적 활성화 기법은 효과적인 방법이다. 그러나 이러한 선택적 활성화 기법을 적용한 기존의 대부분의 연구들은 침입자, 탱크와 같은 개별객체에 대해서만 고려했기 때문에 산불, 유독가스와 같은 연속객체에 대해서는 기존 기법을 적용하기 어렵다. 이는 연속객체들은 주변 환경의 영향에 상당히 민감하기 때문에 매우 유동적이고 불안정하여 이동할 다음 위치를 단순히 시공간 기법만을 가지고 예상할 수 없기 때문이다. 따라서 본 논문에서는 망의 센서노드들이 충분히 밀집한 경우보다 정밀한 예측을 위해 객체가 퍼지거나 수축될 것으로 예상되는 지역의 센서들을 클러스터 단위로 활성화 시키는 클러스터 기반의 선택적 활성화 기법을 제안한다. 또한, 본 방안은 예측을 위한 계산이 동시에 수행될 필요가 없기 때문에 예측을 함에 있어 비동기식 기법을 적용한다.

**Key Words** : Sensor Networks, Continuous Object, Energy Efficiency, Selective Wakeup, Grid Cluster

### ABSTRACT

Energy efficiency in wireless sensor networks is a principal issue to prolong applications to track the movement of the large-scale phenomena. It is a selective wakeup approach that is an effective way to save energy in the networks. However, most previous studies with the selective wakeup scheme are concentrated on individual objects such as intruders and tanks, and thus cannot be applied for tracking continuous objects such as wild fire and poison gas. This is because the continuous object is pretty flexible and volatile due to its sensitiveness to surrounding circumferences so that movable area cannot be estimated by the just spatiotemporal mechanism. Therefore, we propose a cluster-based algorithm for applying the efficient and more accurate technique to the continuous object tracking in enough dense sensor networks. Proposed algorithm wakes up the sensors in unit cluster where target objects may be diffused or shrunken. Moreover, our scheme is asynchronous because it does not need to calculate the next area at the same time.

\* 한국조폐공사 기술연구원 (wslee@komsco.com), \*\* 한전KDN (hshong@kdn.com)

\*\*\* 충남대학교 컴퓨터공학과 컴퓨터네트워크 연구실 (shkim@cnu.ac.kr) (° : 교신저자)

논문번호 : KICS2011-04-206, 접수일자 : 2011년 4월 29일, 최종논문접수일자 : 2011년 7월 19일

## I. 서 론

무선 통신 기술 및 마이크로 센서 기술 등의 발달은 대량의 센서들이 광범위한 지역에 흩어져 무선 센서 네트워크를 구성하는 것을 가능하게 하였다<sup>[1]</sup>. 이러한 무선 센서 네트워크에서 가장 중요한 서비스 중 하나가 바로 적 탱크, 산불, 생화학물질 등 관심 있는 특정 물체에 대한 탐지 및 모니터링이다<sup>[2,3]</sup>. 그러나 이러한 서비스를 지속적으로 유지하기 위해서는 센서 노드 자체의 자원 제약으로 인하여 에너지 효율성이 가장 중요한 요소이다. 따라서 에너지 소비를 최소화하기 위해 추적 객체 주변의 센서노드들만 활성화 상태에 있고 그 나머지는 비활성화 상태에 있는 선택적 활성화 접근 방법은 무선 센서 네트워크에서 가장 효과적인 방법 중 하나로 알려져 있다. 많은 연구들은 이러한 방법을 적용하기 위해 복잡한 센서 상태 스케줄링 알고리즘을 통해 전체적인 에너지 소비를 대폭 줄인 것을 보여 왔다<sup>[4]</sup>.

선택적 활성화 기술을 효과적으로 적용하기 위해서, 해당 객체의 다음 위치를 예상하는 것은 중요한 과제이다. 이것은 에너지를 절약하기 위해 현재 비활성화 상태인 센서로 추적 객체가 근접하는 경우 해당 객체를 즉시 감지하기 위해 미리 활성화 상태에 있어야 하기 때문이다. 따라서 해당 객체의 움직임을 예측하기 위해 기존의 대부분의 연구들은 점-대-점 이동으로써 일정 속도 및 가속도로 다음 위치를 계산하는 시-공간(spatiotemporal) 기법을 적용하였다. 그러나 이러한 기법은 동물, 탱크 및 적군과 같은 개별 객체에 적용하는 것은 가능하지만 산불, 독가스 및 방사능 물질과 같은 연속 객체에 대해서는 적용이 어렵다. 이는 연속 객체는 그 특성상 광범위한 지역에 퍼질 뿐만 아니라 그 모양과 크기가 동적으로 변하기 때문이다<sup>[5-7]</sup>.

최근 연속 객체 추적에서 예측을 위해 PRECO<sup>[8]</sup>가 소개되었다. 그러나 이 기법 또한 연속 객체 추적에 적용이 어려운 시-공간 관련 기법을 적용되었을 뿐만 아니라 다음과 같은 이유로 받아들이기 어렵다. 첫째, PRECO는 현재 경계 센서 노드들이 단지 이웃 센서 노드들이 보내온 짧은 이력 정보만으로 다음 경계가 될 지점을 수학적으로 계산하는 것은 상당히 부정확한 결과를 초래할 수밖에 없다. 왜냐하면 연속 객체는 점-대-점으로 움직임이 표현될 수 있는 개별 객체와는 달리 2차원적으로 광범위한 지역에 걸쳐 퍼질 뿐만 아니라 움직임에 영향을 줄 수 있는 바람, 지형 및 장애물과 같은 다양한 변수들이 주변에 존재하기 때문이다. 즉, 연속 객체의 다음 위치를 정확하게 예측하

는 것은 일기 예보와 같이 수많은 데이터를 입력 값으로 하여 슈퍼컴퓨팅과 같은 연산을 필요로 하기 때문에 이러한 데이터 수집 및 연산을 센서 노드들에게 기대하는 것은 상당히 비현실적이다. 둘째, PRECO에서는 다음 이동 지점을 예상하기 위한 연산과 예상되는 센서 노드들을 활성화시키기 위한 일련의 작업이 현재 경계 센서 노드들끼리 동시에 수행되어야만 하는 제약사항이 있다. 이는 다음 예상 지점이 점들의 집합인 선으로써 계산되어야 하기 때문에 중간에 끊길 수가 없기 때문이다. 그러나 광역에 걸친 무선 센서 네트워크에서 연속 객체를 추적하기 위해 센서 노드들끼리 동기화를 갖는 것은 대단히 많은 시그널링 메시지를 요구하기 때문에 비현실적이다.

따라서 본 논문에서는 망에 센서 노드들이 충분히 분포함을 가정하고, 연속 객체 추적에 있어 에너지 효율을 극대화시키기 위해 선택적 활성화 기법을 적용하되 보다 정밀한 시간과 위치에 센서 노드들을 활성화시키기 위하여 구조화된 클러스터를 기반으로 하는 새로운 예측 기법을 제안한다. 이러한 기법을 구현하기 위해, 무선 센서 네트워크를 구조화된 그리드 클러스터로 가상화시키고 각 클러스터 마다 클러스터 대표를 운영한다. 본 기법을 통해, 연속 객체의 예상 이동 위치는 이차원 적인 특성을 반영하여 점-대-점 매핑이 아닌 면-대-면 매핑을 통해 예상되어지고, 해당 면으로 계산된 해당 지역의 센서 노드들이 클러스터 단위로 활성화된다. 또한, 본 기법은 다음 활성화 혹은 비활성화 되는 지역의 예측이 동시에 계산될 필요가 없으므로 각 센서 노드가 활성화 및 비활성화 판단을 위해 동기화를 가질 필요가 없다.

본 논문의 나머지는 다음과 같이 구성되어있다. 2장에서는 연속 객체 추적에 있어 본 기법을 적용하기 위해 필요한 기본적인 프레임워크를 소개하고, 3장에서는 본 기법에 대한 자세한 설명을 적시한다. 그리고 4장에서는 시뮬레이션 결과를 통해 제안된 방법의 성능을 평가하고 마지막으로 5장에서 결론을 정리한다.

## II. 클러스터 기반 추적을 위한 프레임워크

### 2.1 구조화된 그리드 클러스터 구성

선택적 활성화 기반의 연속 객체 추적 기법의 기본 동기는 한정된 자원을 절약하기 위해 해당 객체가 접근하기 전까지는 가능한 센서 노드들을 에너지 소비 최소 상태인 비활성화 상태로 유지시키는 것이다. 이러한 기법을 효과적으로 적용하기 위해서는 비활성화 되어 있는 센서노드 중 해당되는 센서노드들을 알맞

은 시간에 활성화 시키는 것이 중요하다. 따라서 해당 객체의 움직임을 예측하는 것이 필수적이다. 그러나 연속 객체의 다음 위치를 정확하게 예측하는 것은 비현실적이기 때문에 본 기법에서는 해당 면단위로 센서 노드를 활성화 시키며 이를 위해 그리드 클러스터 구조를 이용한다.

점이 아닌 지역으로써 객체가 움직일 다음 지역을 계산하기 위해 본 논문에서는 무선 센서네트워크를 구조화된 클러스터로 가상화 한다. 해당 클러스터 구조는 기준점(Reference Point)과 정해진 클러스터 크기를 이용하여 간단하게 구성될 수 있다. 또한, 각 센서노드들은 자신의 위치정보를 GPS 신호를 받거나 혹은 지역화 기법 등을 사용하여 아는 것을 가정한다<sup>[9]</sup>. 본 네트워크에서 클러스터의 크기는 모두 동일하다. 그러나 추적하고자 하는 객체의 특성에 따라 그 크기가 미리 조정될 수 있다. 그림 1은 기준점과 클러스터 크기  $\alpha$ 에 의해 간단하게 구성되는 클러스터링 예를 보이고 있다. 또한, 클러스터 크기  $\alpha$ 는 네트워크 구성 초기시 하나의 시스템 파라미터로써 정해질 수 있다. 각 클러스터는 자신의 좌표 값을 가질 수 있으며 이를  $C_{cluster}$ 로 표시할 수 있다. 센서 노드 자신의 위치 값(x, y)로부터 자신이 속한 클러스터의 좌표 값인  $C_{cluster}(a, b)$ 를  $(x_0, y_0)$ 가 기준점의 좌표라고 가정했을 경우 수식(1)과 같이 계산할 수 있다. 단, 문제를 간단하게 하기 위해 본 논문에서는  $C_{cluster}$ 가 항상 양수임을 가정한다.

$$a = \lfloor (x - x_0) / \alpha \rfloor, b = \lfloor (y - y_0) / \alpha \rfloor. \quad (1)$$

각 클러스터에는 하나의 클러스터 헤더를 운영하고 클러스터 헤더는  $CH(a, b)$ 로 표시한다. 단, a와 b는  $C_{cluster}(a, b)$ 의 좌표 값이다. 클러스터 헤더를 선출하기 위해  $C_{cluster}(a, b)$  클러스터 안의 (x, y)에 위치한 센서 노드는 수식 (2)에 의해 유도된 타이머를 이용한다.

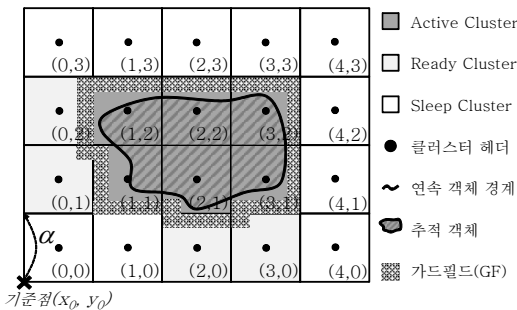


그림 1. 크기  $\alpha$ 와 기준점을 이용한 클러스터 구성

$$\left\lfloor \frac{2R}{\alpha \times E_{residual}} \right\rfloor, \quad (2)$$

단,  $E_{residual}$ 은 센서 노드의 잔여 전력의 양을 의미하며  $R$ 은  $C_{cluster}(a, b)$ 의 중앙지점인  $(x_0 + \alpha(a + 1/2), y_0 + \alpha(b + 1/2))$  와의 거리를 의미하며 다음과 같이 계산된다.

$$R = \sqrt{(x - (x_0 + \alpha(a + \frac{1}{2})))^2 + (y - (y_0 + \alpha(b + \frac{1}{2})))^2}.$$

즉,  $R$ 이 작을수록 클러스터의 헤더로써 되도록 해당 클러스터의 중앙지점의 인근에 위치하며 그러한 센서 노드들 중에 되도록 많은 양의 배터리 전력을 갖는 센서 노드가 선정되도록 하였다. 각 센서 노드들은 시간이 경과한 후에 자신의 위치 정보 (x, y)와 함께 ‘헤더 선언’ 메시지를 각 클러스터 안에 제한된 플러딩을 통해 자신이 속한 클러스터인  $C_{cluster}(a, b)$ 에 송신한다. 각 노드들은 다른 노드들로부터 헤더 선언 메시지를 이미 받은 경우, 자신의 헤더 선언 메시지를 제한하고 해당 헤더의 위치 정보를 저장한다. 만일, 중복되는 다른 노드의 헤더 선언 메시지가 수신된 경우, 해당 클러스터의 모든 센서 노드들은  $R$ 값이 작은 노드를 헤더로 선언한다.

### 2.2 센서 상태 정의 및 전이

에너지 소비에 따라 각 센서 노드들은 간단히 활성화 상태, 비활성화 상태로 구별할 수 있다. 그러나 연속 객체 추적은 해당 객체를 감지한 모든 센서들의 정보가 필요한 것이 아니라 그 객체의 모양을 검출하기 위한 경계 정보가 중요하기 때문에 연속 객체 안에 분포된 센서 노드들에 대한 구분을 고려할 필요가 있다. 또한, 경계 노드 상에 있는 센서 노드일지라도 기존에 제안된 경계 노드 검출 방법들에 따라 모든 센서 노드가 자신의 정보를 전송할 필요는 없다. 따라서 이러한 사항들을 고려하여 본 기법에서는 센서 노드의 상태를 다음과 같이 5가지 종류로 구분하여 정의한다.

1) **Sleep-White** : 이 상태는 센서 노드가 필드에 최초 배포된 상태이거나 혹은 아직 추적 객체에 대해 감지하지 못한 상태이다. 이 상태의 센서는 비활성화 상태로 전력 소모를 최소화하되 자신의 감지범위 내에 추적객체가 감지되었는지를 주기적으로 검사한다. 만일 특별한 이벤트가 발생하지 않는다면 해당 센서는 에너지 절약을 위해 즉시 비활성화 상태로 돌아간다.

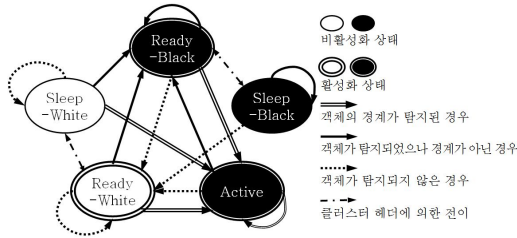


그림 2. 센서노드 상태 전이 다이어그램

2) *Sleep-Black* : 해당 객체가 전체적으로 둘러싸고 있는 상태로써, 객체의 경계선이 해당 클러스터로부터 떨어진 경우 클러스터 헤더에 의해 이 상태로 전이될 수 있다. 즉, 연속 객체의 경우 경계정보가 중요하므로, 해당 센서가 추적하고자 하는 객체의 중앙에 존재하는 경우 연속 객체의 경계선이 떨어진 경우 에너지 절약을 위해 비활성화 상태로 전이된다. 그러나 이상 상태에서 *Sleep-White* 상태와 비슷하게 주기적으로 활성화되어 추적 객체의 존재여부를 검사한다. 이는 연속 객체의 중심부에 불규칙적으로 공백이 발생하는 경우를 대비하기 위해서이다.

3) *Ready-White* : 이 상태에서 센서 노드는 활성화되어 짧은 주기로 추적 객체에 대한 모니터링을 수행한다. 즉, 해당 객체가 접근할 것으로 예상되는 지역의 센서들은 이 상태를 유지하게 되어, 실제 추적 객체의 접근 시에 그 즉시 객체를 감지할 수 있다.

4) *Ready-Black* : 센서 노드들이 객체를 감지했으나 자신이 경계노드가 아닌 경우 이 상태로 진입한다. 이 상태에서 센서노드들은 객체의 경계를 감지하기 위해 감지 작업을 시작한다. 예를 들면, *Ready-Black* 상태의 센서 노드가 경계 노드가 된 경우, 해당 센서 노드는 즉시 *Active* 상태가 되고, 만일 객체를 감지하지 못한 경우에는 *Ready-White* 상태가 된다.

5) *Active* : *Ready-White* 혹은 *Ready-Black* 상태에서 일반적으로 추적 객체의 경계를 탐지하고 해당 센서 노드가 경계 노드인 경우 *Active* 상태로 전이할 수 있다. 이 상태에서 센서 노드는 자신의 클러스터 헤더에게 추적 객체를 감지했음을 알리면서 경계 정보를 보고한다.

그러나, 클러스터 헤더들은 지연시간 없이 즉시 해당 클러스터를 관리하고 다른 클러스터 헤더 및 싱크와 정보를 송·수신하기 위해 항상 활성화 상태임을 가정하여 위와 같은 상태전이에 독립적이다. 또한, 본 논문에서는 해당 객체에 경계 정보를 클러스터 헤더가 집적하여 싱크에게 보고하는 것을 가정하였으며 클러스터 헤더와 싱크간의 통신을 위한 라우팅 경로

상의 센서 노드 또한 위 상태전이와 독립적임을 가정하였다.

각 센서들의 상태에 따라 각 클러스터는 *Sleep Cluster*, *Ready Cluster*, *Active Cluster*로 분류할 수 있다. 즉, *Sleep Cluster*는 해당 클러스터 안의 센서 노드들이 모두 *Sleep-White* 혹은 *Sleep-Black*인 상태이고, *Ready Cluster*는 해당 클러스터 안의 모든 센서 노드들이 *Read-White* 혹은 *Read-Black* 상태이다. 마지막으로 *Active Cluster*는 해당 클러스터 안에 추적 객체의 경계를 가지고 있어서 *Active* 상태인 센서 노드가 존재하는 경우를 의미한다.

### III. 클러스터 기반 선택적 활성화 기법

#### 3.1 선택적 활성화 및 예측을 위한 시그널링

효과적이면서 보다 정밀한 선택적 활성화를 위하여, 각 클러스터 헤더는 제안된 방법에 따라 해당 객체가 움직일 예상지역을 계산하고 이벤트에 따라 요청 혹은 선언 메시지를 송신한다. 센서들은 경계 정보를 알리기 위해 경계 노드가 된 경우 해당 센서는 *Object Boundary Detection (OBD)* 메시지를 자신이 속한 클러스터의 클러스터 헤더에게 송신한다. 해당 객체의 중간에 존재하는 센서들은 그들이 해당 객체가 수축하거나 후퇴한 경우 이를 클러스터 헤더에게 알리기 위해 *Object Not Detection (OND)* 메시지를 보낼 수 있다.

클러스터 헤더는 추적 객체가 진행될 것으로 예상되는 해당 클러스터의 클러스터 헤더에게 *Wakeup Request(WR)* 메시지를 보낼 수 있다. 만일 클러스터 헤더가 *WR* 메시지를 받는다면 그 메시지를 받은 클러스터 헤더는 자신의 클러스터 안에 존재하는 센서 노드들을 활성화시키기 위해 *Cluster Wakeup Request (CWR)* 메시지를 송신한다. 이외는 반대로 클러스터 헤더는 추적 객체의 경계선이 충분히 원거리로 멀어졌다고 판단되는 경우 인접한 클러스터의 클러스터 헤더에게 에너지를 절약하기 위해 센서 노드들을 비활성화 시키라는 *Sleep Request (SR)* 메시지를 송신한다. 만일 클러스터 헤더가 *SR* 메시지를 수신한 경우, 에너지 소모를 최소화하기 위해 해당 클러스터의 센서노드들을 비활성화 시키기 위한 *Cluster Sleep Request (CSR)* 메시지를 송신할 것이다.

#### 3.2 이동지역 예측 및 선택적 활성화

추적 객체의 다음 위치를 계산하기 위하여, 모든 클러스터들은 가드필드(*Guard Field: GF*)를 운영한다.

각 클러스터는 인접한 모든 클러스터들에 대해 8개의 GF를 가지게 되며, 각 GF들은 인접한 클러스터에 각각 매핑될 수 있다. 각 GF들의 영역은 서로 겹칠 수 있으며, 그림 3은 그러한 예를 보이고 있다. 즉,  $C_{cluster}(i-1, j)$ ,  $C_{cluster}(i-1, j+1)$  와  $C_{cluster}(i, j+1)$ 를 위해 매핑되는 GF영역은 서로 겹치게 됨을 알 수 있다. 또한, GF의 크기 및 모양은 추적 객체의 특성 및 응용에 따라 동적으로 정해될 수 있다.

그림 3은 또한 추적 객체의 다음 위치를 예측하는 예를 보이고 있다. 추적 객체의 경계가 그림 3의 (a)와 같이 확산되었을 경우, 해당 GF의 경계 센서들은 정의된 상태 전이 방법에 따라 Ready-White 상태에서 Active 상태가 된다. Active 상태가 된 센서 노드는 OBD 메시지를 이용하여 자신의 클러스터 헤더  $CH(i, j)$ 로 해당 정보를 송신한다. 해당 정보를 수신한 클러스터 헤더는 OBD 메시지를 통해 추적 객체의 경계의 현 위치 뿐만 아니라 진행 방향을 예측할 수 있다. 즉, 객체의 다음 진행 예상 위치는  $C_{cluster}(i-1, j)$ 로 간단히 예상할 수 있다.  $C_{cluster}(i-1, j)$ 를 활성화시키기 위해  $CH(i, j)$ 는 WR 메시지는  $CH(i-1, j)$ 에 송신한다. 그러나 이를 수신한  $CH(i-1, j)$ 는 즉시 자신의 모든 센서 노드들을 활성화 시키지는 않는다. 우선  $CH(i-1, j)$ 는  $C_{cluster}(i, j)$ 에 매핑되는 GF의 센서들을 우선적으로 활성화시킨다. 즉, 에너지 소모를 최소화하기 위해 일부 지역 센서만을 활성화 시킨 후 만일 자신의 클러스터에 객체가 탐지된 경우에 비로소 모든 객체 탐지를 위해 모든 센서들을 활성화 시키게 된다.

클러스터 크기는 추적하고자 하는 객체의 특성에 따라 최적화되어 정해져야만 할 것이다. 그러나 객체가 움직이는 속도가 일반적이지 않은 갑자기 빠르게 변화하는 경우 하나의 클러스터가 아닌 두개 이상의 클러스터가 준비를 위해 필요한 경우가 예외적으로 발생할 수 있다. 그림 4는 그러한 경우의 예를 보이고 있다. 시간 t에  $C_{cluster}(i+2, j)$ 안에 존재하던 객체의 경

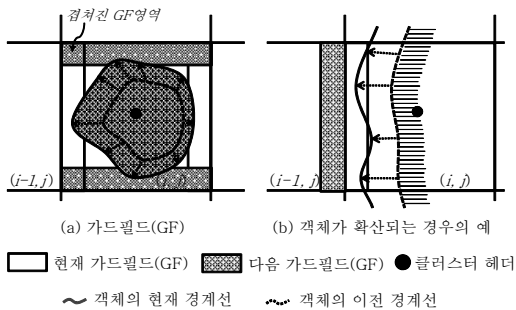


그림 3. 가드필드(GF)에 따른 예측 기법의 예

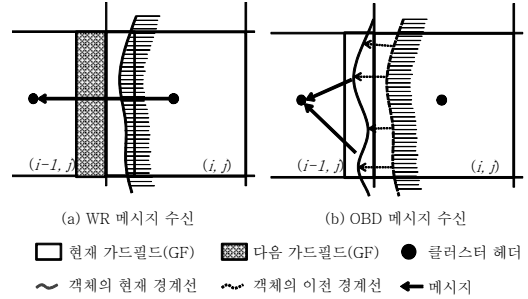


그림 4. 예측 영역이 한 클러스터 크기를 초과하는 예

계선은 예외적으로 빠르게  $C_{cluster}(i+1, j)$ 까지 전진할 수도 있다. 이러한 경우를 대비하여 각 클러스터 헤더는 객체의 다음 예상 위치가 자신의 클러스터 영역만으로 커버할 수 있는지 판단하는 계산을 수행해야만 한다. 그러한 커버리지를 계산하기 위해 다음의 변수가 사용된다.

- $T_{WR}$  : WR 메시지를 수신한 시간
- $T_{OBD}$  : OBD 메시지를 수신한 시간
- $T_{OBD} - T_{WR} = \Delta t$
- $\delta$  : 커버리지 임계치 값

클러스터 헤더는 인접 클러스터의 GF로부터 자신의 클러스터로 객체의 경계가 이동하는데 걸린 시간을 계산한다. 만일  $\Delta t$ 가 자신의 클러스터 영역만으로 커버할 수 없는 시간인  $\delta$ 안에 들어온 경우 클러스터 헤더는 자신의 클러스터크기로는 예측 영역을 커버할 수 없는 것으로 판단하여 그 즉시, 해당되는 인접 클러스터 헤더에게 WR 메시지를 송신한다. 즉, 일반적인 경우에는 GF영역의 센서들만 우선 활성화되지만 이러한 경우 급속한 속도변화에 대비하기 위해 해당 클러스터의 센서 노드 전체가 활성화하여 대기하게 된다.

- $\delta \geq \Delta t$  : CH는 CWR 메시지 송부
- $\delta < \Delta t$  : CH는 CWR 메시지 및 WR 메시지 송부

또한, 임계치 값  $\delta$ 는 객체 및 환경에 대한 통계적인 특성, 클러스터 및 GF의 크기 등과 같은 여러 요소들에 따라 최적화되어 결정되어야 한다.

#### IV. 성능 분석

본 논문은 시뮬레이션을 통해 다양한 파라미터들을

적용하여 성능평가를 수행하고자 한다. 우선 성능 측정을 위한 메트릭 및 시뮬레이션을 위해 사용된 파라미터를 정의한 후 시뮬레이션 결과를 설명한다. 다음은 시뮬레이션에 사용된 메트릭을 정의한다.

- 1) 총 에너지 소비량(*Total energy consumption*) : 시뮬레이션 시간 동안 네트워크에서 소비한 총 에너지
- 2) 예측정확성(*Accuracy*) : 객체의 경계선을 정확하게 예측하는지에 대한 정확성을 평가하기 위해, 본 논문에서는 선택적 활성화 및 예측기법을 사용하지 않고 모든 센서가 항상 활성화인 경우를 *All Active* 모드로 정의한다. 또한, *All Active* 모드에서는 예측기법을 수행하지 않으므로 이 상태에서 측정된 경계선 센서 노드의 수는 정확하다고 가정한다. 따라서 본 논문에서 예측 정확성은 다음과 같이 제한된 기법을 도입한 경우 측정된 경계선 센서 노드의 수와 *All Active* 모드의 경우 측정된 경계선 센서 노드의 수간의 관계로 나타낼 수 있다.

$$Accuracy = \frac{\text{제한된방법에서측정된경계선센서노드수}}{\text{All Active 모드에서측정된경계선센서노드수}}$$

본 시뮬레이션은 쉘넷 ver 4.0<sup>[10]</sup>을 이용하여 PRECO와 제안된 메커니즘을 비교하였다. 시뮬레이션된 네트워크는 2,500개의 센서노드로 구성되며 200m × 200m의 정사각형 영역에 임의로 분포되었다. 단위 클러스터와 GF의 디폴트 크기는 각각 20m, 5m로 설정하였다. Telos<sup>[11]</sup> 센서 노드의 특성에 근거하여 활성화 상태인 경우와 비활성화 상태인 경우 전력소비는 각각 3mW, 15uW이며, 송·수신 전력소비는 각각 42mW, 29mW로 설정하였다. 또한, 연속 객체는 네트워크의 중앙에서 최초 2m의 반지름으로 발생하여 임의의 속도로 확산된다. 또한, 통신 거리는 30m, 센싱거리는 1m로 설정하였다.

본 논문에서는 선택적 활성화 기법이 적용된 경우 네트워크에 어떠한 영향을 있는지에 관하여 예측하였다. 우선 선택적 활성화 기법이 없이 모든 센서 노드가 활성화 되어 있는 경우 비활성화되어 있는 센서 노드가 에너지를 덜 소비하기 때문에 선택적 활성화 기법을 적용한 경우 전체 에너지 소비량은 줄 것으로 예상할 수 있다. 또한, 본 실험의 주요 쟁점은 연속 객체의 경계선을 얼마나 정확하게 예측하느냐이다. 만일 경계선을 예측하는 정도가 받아들일 수 없는 수준인 경우, 제안된 기법을 적용하는 것은 의미가 없을 것이다.

그림 5와 6은 시뮬레이션 시간동안 선택적 활성화 기법을 적용한 효과에 대해 보여준다. *All Active* 상태

인 경우와 비교하여 제한된 기법을 적용한 경우 전체 전력 소모가 현저히 줄어들었음에도 불구하고, PRECO 기법보다 더 좋은 예측 정확성을 보이고 있다. 비록 PRECO에 비해 좀 더 많은 전력을 소비하지만 이러한 결과의 원인 중에 하나로 예측 실패로 인하여 미리 활성화되어 있는 영향이 있을 것으로 판단된다.

그림 7은 클러스터 크기의 변화에 따른 영향을 보이고 있다. 클러스터 크기가 작을수록 활성화를 위해 대기시키는 클러스터를 좀 더 정교하게 선택할 수 있어 그만큼 활성화 되는 영역이 작아 질 수 있으므로 전력소모가 작을 수 있다. 그러나 이러한 경우 미리 활성화 되는 영역이 작아서 예측 성공률이 그만큼 줄게 된다. 그러나 본 시뮬레이션에서 클러스터 크기가 줄어들었음에도 불구하고 예측 정확성은 받아들일만하다. 그 이유는 GF의 크기가 적정수준에서 고정되어 있기 때문으로 판단된다.

그림 8은 GF의 크기 변화에 따른 영향을 보여준다. 그래프에서와 같이 GF영역이 좁아질수록 예측 실패

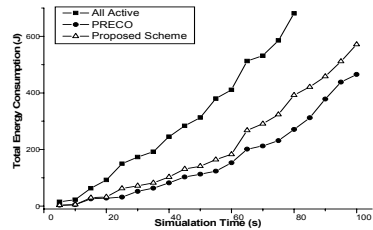


그림 5. 에너지 총 소비량 비교

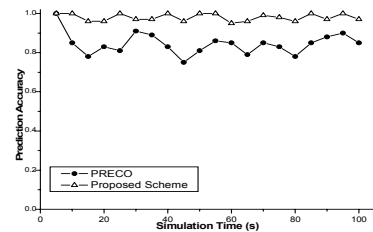


그림 6. 예측 정확성 비교

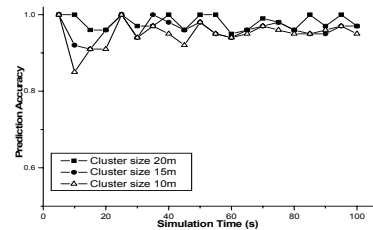


그림 7. 클러스터 크기에 따른 영향

율은 높아진다. 반대로 GF영역이 넓어질수록 예측 실패율은 낮아진다. 이는 GF영역이 넓어질수록 많은 수의 센서들이 다가 오는 객체를 식별할 수 있는 확률이 높아지기 때문으로 판단된다. 따라서 클러스터와 GF의 크기는 추적 객체의 특성에 따라 최적화되어야만 할 것이다.

그림 9와 10은 연속객체가 확산되는 속도와 가속도가 네트워크에 주는 영향을 보이고 있다. 객체의 속도가 빨라질수록 예측 정확성은 떨어질 수 있다. 이는 추적 객체 경계의 다음 위치가 계산되기 전에 객체가 지나칠 수 있기 때문이다. 그러나 제안된 본 방법은 간단한 계산만을 요구하기 때문에 예측 수행 연산이 무척 빠르다. 따라서 객체의 확산 속력 및 가속도에 대한 영향력이 그만큼 작다. 반면 PRECO는 현실적으로 센서노드가 수행하기 어려운 복잡한 계산을 요구하고 오직 일정한 가속도만을 지원하기 때문에 객체의 속도 및 가속도가 일정 값을 넘어선 경우 예측을 전혀 수행하지 못한다.

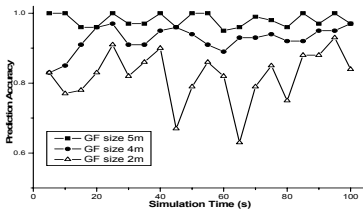


그림 8. GF 크기에 따른 영향

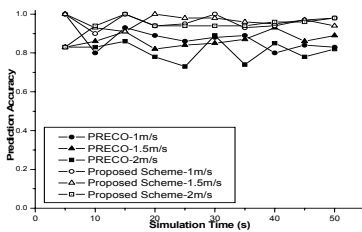


그림 9. 속도에 따른 영향

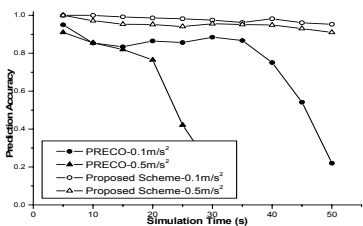


그림 10. 가속도에 따른 영향

## V. 결 론

본 논문은 연속 객체 추적을 위해 에너지 효율성의 중요성과 선택적 활성화 기법 도입 시 예측의 정확성을 지적하였다. 따라서 본 논문에서는 에너지 효율을 높이기 위해 선택적 활성화 기법을 도입하였고 연속 객체의 다음 위치에 대한 예측 정확성을 높이기 위해 다음 위치를 점이 아닌 정해진 클러스터 영역 단위로 센서 노드를 활성화 시키는 기법을 제안하였다. 시뮬레이션 결과는 제안된 기법이 센서 네트워크 전체의 에너지 소모를 극적으로 줄이고 있음을 보이고 있으며, 그림에도 불구하고 유연하게 크기와 모양을 변화시키는 연속 객체의 경계선을 정확하게 예측하고 있음을 알 수 있다. 또한, 활성화되는 클러스터의 크기 및 GF의 크기와 같은 다양한 파라미터들은 추적하고자 하는 연속 객체의 특성에 따라 응용에서 조절 될 수 있을 것이다.

## 참 고 문 헌

- [1] I.F. Akyildiz, S. Weilian, Y. Sankarabramaniam, and E. Cayirci, "A Survey on Seonsor Networks," *IEEE Communications Magazine*, Vol.40, pp.102-114, Aug. 2002.
- [2] W. Zhang and G. Cao, "DCTC: Dynamic Convoy Tree-Based Collaboration for Target Tracking in Sensor Networks," *IEEE Transactions on Wireless Communications*, Vol.3, No.5, Sept. 2004.
- [3] W. P. Chen, J. C. Hou and L. Sha, "Dynamic Clustering for Acoustic Target Tracking in Wireless Sensor Networks," in *Proc. IEEE ICNP 2003*, pp.284-294, Nov. 2003.
- [4] Y. Xu, J. Winter and W.C. Lee, "Prediction-based Strategies for Energy Saving in Object Tracking Sensor Networks," in *Proc. IEEE MDM 2004*, pp.346-357, Jan. 2004.
- [5] W. Chang, H. Lin, and Z. Cheng, "CODA: A Continuous Object Detection and Tracking Algorithm for Wireless Ad Hoc Sensor Networks," in *Proc. IEEE CCNC 2008*, pp. 168-174, Jan. 2008.
- [6] X. Ji, H. Zha, J. J. Metzner, and G. Kesidis, "Dynamic Cluster Structure for Object Detection and Tracking in Wireless Ad-hoc

