무선 센서 네트워크에서 안전하고 에너지 효율적인 클러스터 기반 프로토콜

Secure and Energy Efficient Protocol based on Cluster for Wireless Sensor Networks

김진수*, 이정현**
인하대학교 정보공학과*, 인하대학교 컴퓨터공학부**

Jin-Su Kim(kjspace@inha.ac.kr)*, Jung-Hyun Lee(jhlee@inha.ac.kr)**

요약

무선 센서 네트워크에서는 센서 노드의 제한된 자원으로 동작하기 때문에 클러스터 기반의 라우팅 방법을 통해 네트워크의 수명을 향상시키고 있다. 본 논문에서는 클러스터 기반의 라우팅 기법이 가진 에너지 효율을 최대화하기 위해 방향, 거리, 밀도, 그리고 잔여 에너지량을 사용하고자 한다. 상위 클러스터 헤드의 방향 정보를 이용하여 새로운 클러스터 헤드를 자율적으로 선출할 때 고립 노드의 발생 빈도를 최소화하고, 거리 정보를 이용하여 셋업 단계에서 새로운 클러스터와 이전의 클러스터 모두에 포함되어 정보를 갱신할 필요가 없는 센서 노드들을 슬립 모드로 전환하여 에너지 소모를 줄이며, 단지 새로 가입되는 센서 노드만 정보 교환함으로써 불필요한 에너지 소비를 줄이고자 한다. 뿐만 아니라, 클러스터 기반의 라우팅 기법에서 내부나 외부의 공격에 강한 키 관리 기법을 통해 안전하고 에너지 효율적인 통신이 가능하도록 하여 전체적인 네트워크 효율을 높이고자 한다. 따라서 전체 네트워크내의 멤버들에게 클러스터 헤드가 될 수 있는 안전하고 균등한 기회를 주고자 하는 클러스터 헤드 선출 기법을 제안한다.

■ 중심어 : | 무선 센서 네트워크 | 클러스터 기반 라우팅 | 클러스터 헤드 선출 | 키 관리 |

Abstract

Because WSNs operate with limited resources of sensor nodes, its life is extended by cluster-based routing methods. In this study, we use data on direction, distance, density and residual energy in order to maximize the energy efficiency of cluster-based routing methods. Through this study, we expect to minimize the frequency of isolated nodes when selecting a new cluster head autonomously using information on the direction of the upper cluster head, and to reduce energy consumption by switching sensor nodes, which are included in both of the new cluster and the previous cluster and thus do not need to update information, into the sleep mode and updating information only for newly included sensor nodes at the setup phase using distance data. Furthermore, we enhance overall network efficiency by implementing secure and energy-efficient communication through key management robust against internal and external attacks in cluster-based routing techniques. This study suggests the modified cluster head selection scheme which uses the conserved energy in the steady-state phase by reducing unnecessary communications of unchanged nodes between selected cluster head and previous cluster head in the setup phase, and thus prolongs the network lifetime and provides secure and equal opportunity for being cluster head.

■ keyword : | Wireless Sensor Network | Cluster based Routing | Cluster Head Selection | Key Management |

* 본 논문은 인하대학교 지원에 의하여 연구되었습니다.
접수번호 : #091126-002
접수일자 : 2009년 11월 26일
심사완료일 : 2010년 01월 06일
교신저자 : 김진수, e-mail : kjspace@inha.ac.kr

I. 서 론

무선 센서 네트워크(Wireless Sensor Networks)는 주위 환경을 감시하고 데이터를 수집하는 용도로 다양한 응용이 가능한 기술로써, 군사 지역이나 보안 지역에서의 침입 탐지, 자연 환경에서 야생 동물의 서식지 모니터링, 온도와 습도 같은 환경 모니터링 등에 적용이 가능하다. 센서 노드들은 주변 현상을 인식하고, 측정된 데이터를 기지국(Base Station, BS)에 전송하며 기지국에서는 수집된 데이터를 분석한다. 이러한 무선 센서 네트워크에서의 가장 큰 제약사항은 제한된 자원을 들 수 있으며, 에너지를 효율적으로 사용하기 위한 대표적인 방법으로 데이터 통합을 통한 중복된 데이터들을 압축하여 통신비용을 줄이는 LEACH[1], LEACH-C[2], HEED[3]와 같은 클러스터 기반의 라우팅 방법이 있다. 이러한 대표적인 클러스터 기반의 라우팅 방법들 중 LEACH는 확률 함수를 이용하여 매 라운드마다 효율적인 클러스터 헤드를 선정할 수 있지만, 매 라운드마다 동일한 클러스터 헤드의 수를 보장할 수 없다는 단점이 있다. 이에 비해 LEACH-C는 LEACH와 동일한 클러스터 기반이지만 기지국의 통제 하에 클러스터 헤드를 선정함으로써 클러스터 헤드의 수를 보장 받을 수 있는 장점이 있다. 그러나 센서 네트워크에 참여하는 각 센서 노드들의 정보를 기지국에 전송해야 하고, 또한 노드들의 거리를 계산하여 평균 거리가 가장 짧은 센서 노드를 찾아야 하는 NP-Hard Problem에 속한다. 이러한 문제를 해결하기 위해 Simulated Annealing Algorithm[4]을 사용하여 효율적으로 클러스터 헤드를 선정하지만, 최적의 값을 보장할 수 없고, 위치를 파악하기 위한 오버헤드가 많이 필요로 한다. 또한, 매 라운드마다 CH를 선출하여 일정한 시간동안 라우팅의 역할을 하기 때문에, 악의적인 목적을 갖는 공격자는 라우팅 요소를 인증하기 어렵게 만들거나 손상시키려는 주요한 공격의 목표가 된다. 따라서 클러스터 기반의 프로토콜에 안전한 통신을 강화하기 위해 동적으로 또는 일정 시간 후에 분배된 키를 재배치하여 노드들 간의 링크를 변경하게 되면 수반되는 오버헤드가 상당히 크기 때문에 꼭 필요한 경우가 아니면 부적절하다.

본 논문에서는 클러스터 기반의 라우팅 기법이 가진 에너지 효율을 최대화하기 위해 방향, 거리, 밀도, 그리고 잔여 에너지량을 사용하고자 한다. 상위 클러스터 헤드의 방향 정보를 이용하여 새로운 클러스터 헤드를 자율적으로 선출할 때 고립 노드의 발생 빈도를 최소화하고, 거리 정보를 이용하여 셋업 단계에서 새로운 클러스터와 이전의 클러스터 모두에 포함되어 정보를 갱신할 필요가 없는 센서 노드들을 슬립 모드로 전환하여 에너지 소모를 줄이며, 단지 새로 가입되는 센서 노드만 정보 교환함으로써 불필요한 에너지 소비를 줄이고자 한다. 밀도와 잔여 에너지량은 에너지 효율적인 최적의 클러스터 구성시 필요한 인자이다. 또한 클러스터 기반의 라우팅에서 안전한 정보 교환을 위한 키 관리를 통해 전체 네트워크의 에너지 효율과 안전한 키 관리 프로토콜을 제안한다.

II. 기존 연구

클러스터 기반의 라우팅 프로토콜의 대표적인 연구는 LEACH, LEACH-C, HEED등이 있다.

LEACH는 네트워크 노드들 간의 에너지 소모를 균등하게 함으로써 네트워크의 생존시간을 최대화하려 하였고, 클러스터 헤드 선출은 식 (1)의 확률함수에 의해 결정된다.

$$P_i(t)=\begin{cases}\dfrac{k}{N_{total}-k\times(r \bmod \frac{N_{total}}{k})} & : C_i(t)=1 \\ 0 & : C_i(t)=0\end{cases} \quad (1)$$

식 (1)에서 i는 노드의 식별자, t는 시간, N_{total}은 전체 노드의 수, k는 클러스터의 수, r은 라운드를 나타낸다. 그러나 클러스터 헤드를 선출하는 셋업 단계에서 멤버 노드들과의 빈번한 정보의 교환으로 인해 에너지 소비가 많이 발생하는 오버헤드를 가진다. 비록 LEACH-C에서는 이러한 문제점을 해결하기 위해 Simulated Annealing Algorithm을 사용하여 효율적으로 클러스터 헤드를 선정하였지만 최적의 값을 보장할

수 없고, 노드 전체의 에너지와 자신의 현재 에너지를 고려하여 클러스터 헤드를 결정하기에는 추가적인 오버헤드가 필요하다는 단점을 가지고 있다.

HEED 프로토콜은 클러스터 헤드를 선출하기 위해 개별 노드 자신의 인자만을 이용한다는 점에서 선출 기법이 우수하다. 클러스터 헤드 선출 시 사용되는 식은 멤버 노드들이 가진 잔여 에너지를 이용하며, 식 (2)와 같은 확률 함수를 이용한다.

$$CH_{prob} = C_{prob} \times \frac{E_{residual}}{E_{\max}} \quad (2)$$

$E_{\max}$는 멤버 노드가 가진 초기 에너지, $E_{residual}$은 멤버 노드의 현재 잔여 에너지, C_{prob}는 전체 네트워크 노드 중 클러스터 헤드의 비율(5%)을 의미한다. HEED 프로토콜 알고리즘은 초기의 C_{prob}와 $P_{\min}$ 값 중에서 큰 값으로 시작하여 CH_{prob}이 1이 될 때까지 CH_{prob}에 2를 곱하거나 1이 된 이웃 노드들로부터 메시지를 수신할 때까지 반복하여 클러스터 헤드를 선출하도록 한다. 이것은 모든 노드들의 에너지를 알 필요 없이 오직 노드 자신의 인자만을 이용하여 클러스터 헤드를 선출할 수 있으며, 일정 반복 횟수를 수행한 후 종료되는 장점을 가지고 있다. 그러나 선출 확률 값에 2배씩 반복 수행함으로써 노드들이 가진 에너지의 양이 유사할 경우 클러스터 헤드의 수를 보장하지 않으며, 반복해서 ADV 메시지들을 전송하기 때문에 노드들의 송수신에서 발생하는 에너지 소모에 커다란 오버헤드를 갖는 문제점을 가지고 있다.

WSN에서 제안된 키 관리 기법들은 노드 배치 이전에 키 분배의 유무, 마스터 키의 사용 여부에 따라 사전 키 분배 방식[5], 마스터 키 기반 방식[6], 베이스 스테이션 기반 방식[7]으로 구분할 수 있다.

사전키 분배 방식(RPK)[5]은 Random Key Predistribution, Shared Key Discovery, Path Key Establishment의 3단계의 과정을 거쳐 노드들 간의 안전한 인증을 보장한다. 사전키 분배 방식은 연결 정도가 확률적으로 구성되기 때문에 WSN를 나타내는 전체 그래프가 완전하게 연결되지 않을 수 있으며, 센서 노드의 배치가 불규칙적이거나 배치된 환경에 물리적으로 통신을 방해하는 요소가 있는 경우 더욱 심해진다.

하나의 키를 사용하는 메커니즘으로는 대량의 센서가 흩어져 있는 센서 네트워크에서는 안전한 키 메커니즘의 설계가 어렵다는 판단으로, 4개의 암호키와 키 설정 프로토콜을 가진 LEAP 프로토콜[6]가 제시되었다. 4개의 암호키는 BS와 공유하는 개인키, BS가 네트워크에 있는 모든 노드와 공유하는 브로드캐스팅 키인 그룹키, 다른 센서 노드와 공유하는 Pairwise 키, 그리고 몇 개의 이웃 노드와 공유하는 클러스터 키이다. LEAP은 공격 노드는 개인키를 알 수 없으며, pair-wise 키와 클러스터 키는 주위의 이웃 노드를 인증하기 위해서만 사용되고 그룹키는 방송되는 메시지를 복호화하기 위해서만 사용되므로, 위협 노드를 가진 센서 네트워크의 생존성을 극대화 할 수 있는 방법이다. 개인키와 그룹키는 센서 노드가 배치되기 전에 탑재되기 때문에 악의적인 공격자에게 센서 노드가 탈취될 수 있다.

HIKES[7]는 BS가 신뢰된 인증기관(TA)의 역할을 담당하면서 그 기능 중 일부를 CH에게 위임하는 방법으로, 모든 노드에 partial key escrow 테이블을 가지고 키를 생성하며 CH로 선출될 수 있고, 데이터 통합 후 CH들 간의 메시지 교환을 통해 BS에 정보를 전송한다. 그러나 센서 노드의 인증이 BS을 통해 이루어지고, 모든 노드에 partial key escrow 테이블을 저장하고 있어야 하기 때문에 추가적인 저장 공간이 필요하다. 또한 악의적인 공격자가 노드 탈취를 통해 partial key escrow 테이블을 획득한 경우 이를 이용하여 다른 지역에 위치한 CH와 센서 노드간의 pairwise 키를 유추할 수 있고, 클러스터에 속한 노드들의 수가 증가함에 따라 CH는 노드 인증을 위해 전송해야 하는 메시지의 크기가 증대되기 때문에 전체 네트워크 생존 시간을 줄어들 수 있다.

III. 안전하고 에너지 효율적인 클러스터 헤드 프로토콜

1. 에너지 효율적인 셋업 단계

본 논문에서 제안하는 셋업 단계는 다음과 같다.

① 클러스터 내 멤버 노드들에게 기존의 클러스터 헤드(CH_{old})가 새로운 클러스터 헤드(CH_{new})가 선출되었음을 멤버들에게 알리고, ② 이를 수신한 멤버 노드들은 그 정보를 이용하여, 자신의 상태를 결정한다. 즉, 클러스터가 변경되지 않는 멤버 노드들은 셋업 단계가 종료될 때까지의 일정 시간 동안을 슬립 상태가 되며, 삭제되는 멤버 노드들은 새로운 클러스터 헤드를 선택할 준비를 하고, 선출된 클러스터 헤드는 멤버 노드들에게 ADV 메시지를 전송한다. ③ 새로 추가된 멤버 노드들은 Join-REQ 메시지를 전송하고, ④ 수신된 정보를 이용하여 기존의 방법과 유사하게 TDMA 슬롯을 결정하고, 멤버 노드들의 목록을 작성하며, 그 결과를 모든 멤버 노드들에게 브로드캐스팅하고, 새로 삽입된 멤버 노드들은 이 정보를 수신하여, 다음 단계를 준비함으로써 셋업 단계를 종료한다. 여기서 제안된 방법은 셋업 단계에서 변경되지 않는 멤버 노드들이 셋업 단계가 종료되기까지 슬립 상태가 되고, 정보 교환에 참여하는 송수신 노드들의 수 차이만이 있을 뿐 동일하게 적용된다. 또한 변경되지 않는 노드들은 기존 클러스터에서 사용했던 정보들(TDMA slot과 CDMA spreading 등)을 사용하며, 이 정보는 상속을 통해 전달되고, 삭제된 멤버 노드들의 정보를 새로 삽입된 멤버 노드들이 대체함으로써 재사용을 통한 효율을 증가시키고자 한다.

표 1. 셋업 단계에서 소모되는 에너지 비교

단계	기존의 셋업 단계	제안된 셋업 단계
1	E_{Tx} - $(N-1)E_{Rx}$	E_{Tx} - $(N-1)E_{Rx}$
2	$(N-1)E_{Tx}$ - $(N-1)E_{Rx}$	$(N-n-1)E_{sleep}+E_{Tx}^{'}$ $+E_{Tx}+nE_{Rx}$
3	E_{Tx} - $(N-1)E_{Rx}$	nE_{Tx} - nE_{Rx}
4	-	E_{Tx} - nE_{Rx}

[표 1]에서 N은 클러스터내의 전체 노드 수, n은 클러스터 내에 새로 삽입되는 멤버 노드 수를 나타내고, $E_{Tx}^{'}$는 CH_{old}에서 CH_{new}로 상속에 필요한 정보를 전송하는 것이다. [표 1]에 보인 것과 같이, 기존의 방법과 제안된 방법 사이의 셋업 단계상의 에너지 소모를 줄이기 위한 최적화된 방법을 찾기 위해, 클러스터 헤드의 변동 거리와 변동 거리에 따른 변경되지 않는 멤버 노드들의 수를 이용하여 에너지 효율을 높이려고 한다. 제안된 셋업 단계에서의 전체 에너지 소모량이 기존의 셋업 단계에 사용된 전체 에너지 소모량보다 작거나 같은 경우를 다시 정리하면 식 (3)과 같다.

$$\frac{(N-n-3)E_{Tx}}{(3n-2N+2)E_{Rx}}+\frac{(n-N+1)E_{Sleep}}{(3n-2N+2)E_{Rx}} \geq 1$$
$$\therefore n \leq \left\lfloor \frac{3N-5}{4} \right\rfloor \quad (3)$$

여기서, E_{Tx}와 E_{Rx} 값의 차는 $l\epsilon_{fx}d^2$으로, 같은 클러스터 내의 짧은 거리($d < d_0$)이기 때문에 송수신하는 메시지의 비트수가 크지 않고, 새로 삽입되는 노드들의 수가 정수이기 때문에, $\frac{E_{Tx}}{E_{Rx}} \approx 1$로 치환한 식으로 정리하였다. 또한 E_{Rx}에 비해 E_{Sleep}에 사용되는 에너지의 소비가 아주 작기 때문에($E_{Rx} \gg E_{Sleep}$), 즉 $\frac{E_{Sleep}}{E_{Rx}} \approx 0$을 적용하여 간소화된 식 (4)를 얻을 수 있다. LEACH와 LEACH-C의 경우에 프로그램에 세팅된 E_{Sleep}의 값은 0으로 적용하여 실험하였다. 이 식 (3)을 통해 기존의 방법보다 에너지 효율을 향상시킬 수 있는 변경되지 않는 멤버 노드들의 수(m; $m = N-n$)는 식 (4)와 같다.

$$m \geq \left\lceil \frac{N+5}{4} \right\rceil \quad (4)$$

[그림 1]은 기존의 클러스터 헤드(CH_{old})가 새로운 클러스터 헤드(CH_{new})로 변동되었을 때, 변경되지 않는 중첩된 부분을 보여준다. 중심 CH_{old}와 중심 CH_{new}가 서로 다른 클러스터 반경(r,R)을 가졌다고 가정했을 때, 겹치는 부분이 다른 클러스터로 변경되지 않는 멤버 노드들의 영역을 나타내며, 이러한 중첩된 영역($Area(r,R,d)$)을 계산하면, 다음 식 (5)와 같다.

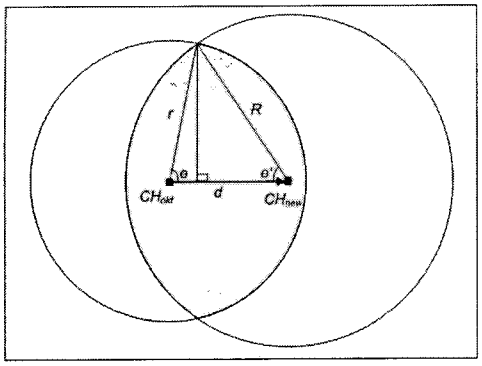


그림 1. 클러스터 헤드 변동에 따른 변경되지 않는 영역

$$Area(r,R,d) = \left(acos(\frac{r^2-R^2+d^2}{2dr})r^2 - \frac{(R^2-r^2-d^2)sin(\theta)r}{2d}\right) + \left(acos(\frac{3d^2-R^2+r^2}{2dR})R^2 - \frac{(3d^2-R^2+r^2)sin(\theta')R}{2d}\right) \quad (5)$$

$$sin(\theta) = sin(acos(\frac{R^2-r^2-d^2}{2dr}))$$
$$sin(\theta') = sin(acos(\frac{3d^2-R^2+r^2}{2dR})) \quad (6)$$

여기서, θ는 각도(degree)이고, d는 클러스터 헤드의 이동 거리이고, R은 클러스터 헤드의 반경으로 새로운 클러스터 헤드의 반경과 기존의 클러스터 헤드의 반경은 같다고 가정하였다. 전체 네트워크의 멤버 노드들의 분포가 균등 분포라고 가정하였을 때, 식 (5)로부터 계산된 중첩된 영역을 하나의 노드가 분포될 수 있는 평균 영역($M^2/Node_{total}$)으로 나누면, 중첩된 영역에 분포될 수 있는 멤버 노드 수($Node_{overlap}$)를 구할 수 있고, 식 (7)과 같다.

$$Node_{overlap} = \frac{Node_{total} \times Area(r,R,d)}{M^2} \quad (7)$$

이때, M^2은 센서 네트워크의 전체 영역이며, $Node_{total}$ 은 센서 네트워크 내에 분포한 전체 노드수를 의미하고, 계산된 $Node_{overlap}$은 식 (4)의 m과 같다.

본 논문에서는 클러스터 헤드 결정에 영향을 끼치는 요소인 Cost를 결정하기 위해 밀도(Density)의 개념을 사용하였다. 밀도는 한 클러스터 내에서 각 멤버 노드와 클러스터 헤드 사이의 거리에 따라 다르며, 각 멤버 노드로부터 일정 거리, 즉 클러스터 헤드가 포함할 수 있는 반경(R)에 속한 동일한 클러스터 내의 이웃 노드 수($Node_{neighbor}$)와 다른 클러스터에 포함된 외래 노드 수($Node_{foreign}$)의 비율을 말하며, 식 (8)과 같이 정의하였다.

$$Density_i = \frac{Node_{neighbor}}{Node_{neighbor} + Node_{foreign}} \quad (8)$$

여기서, 멤버 노드 i가 자신이 속한 클러스터 헤드에 가까우면 가까울수록 $Density_i$의 값은 커지고, 멀어지면 멀어질수록 작아지며, 이 값의 범위는 0과 1 사이로 정규화 하였다. 시간이 경과함에 따라 멤버 노드들은 가지고 있던 모든 에너지를 소모하고, 생명 주기를 마치게 된다. 따라서 네트워크의 유지 시간이 경과함에 따라 생존 노드 수($Node_{alive}$)는 작아지고, 불능 노드 수($Node_{dead}$)는 커지기 때문에 밀도에 영향을 미친다. 그러나 기존의 클러스터링 방법[1][2][4]에서와 같이 시간에 따른 생존 노드 수는 불능 노드가 발생한 후 일정 한계를 넘어서는 시점부터 급격히 감소하여 전체 네트워크의 수명 기간을 마치게 된다. 따라서 효율적인 에너지 관리를 위한 밀도 계산은 클러스터내의 생존 노드 수와 잔여 에너지가 각각 한계값 이상일 경우에 각 멤버 노드의 정보와 현재의 클러스터 헤드(CH_{new}) 정보를 식 (5)와 식 (7)에 적용하면 빠르게 계산된다. 한 라운드의 셋업 단계에서 클러스터 헤드 선출을 위해서는 전체 네트워크 내의 모든 멤버 노드들이 공평한 기회를 주고, 에너지 소모를 최소화 할 수 있는 멤버 노드들을 찾기 위해서 계산되어야 하는 것이 Cost이며, 각 멤버 노드들로부터 클러스터 헤드로 전송 받는 정보이다.

본 논문에서는 Cost에 영향을 주는 주요 인자들로 밀도와 잔여 에너지량을 이용하여 결정하였다. CH_{old}가 획득한 정보들 중, 먼저 평균 에너지 이상을 가진 클러스터 내 노드들을 선별하여 리스트를 생성하고, 선별된 노드 리스트들 중에서 $Density_i$의 값이 한계값 이상인 노드들만 선별하여 최종 후보 리스트를 생성한다. Cost는 식 (9)에 의해 계산된다.

$$Cost_i = \frac{1}{\alpha \cdot Density_i + (1-\alpha) \cdot \frac{E_{residual}}{E_{max}}} \quad (9)$$

여기서, α 는 밀도와 잔여 에너지량 사이의 가중치를 나타낸다. 적절한 가중치 α 를 적용함으로써 밀도와 잔여 에너지량의 중요한 요인 중 하나에 치우치지 않고 효율적으로 Cost를 설정하고자 한다.

2. 에너지 효율적인 클러스터 헤드 선출 알고리즘 (3DE_var algorithm)

본 논문에서는 클러스터 헤드와 멤버 노드들로부터 발생하는 정보들($Cost$, $NodeID$, 잔여 에너지량 등)과, 거리, 그리고 밀도를 이용하여 후보 클러스터 헤드 집합(S_{CH})을 구성한다.

[그림 2]는 클러스터 헤드 선출을 위한 3DE_var 알고리즘을 보여준다. 고립 노드들의 수를 최소화하기 위해 방향 정보를 이용하고, 이러한 정보는 이전의 상위 클러스터 헤드가 다음 상위 클러스터 헤드를 선출하게 되었을 때SimDirection() 함수에 의해 결정되며 하위의 클러스터 헤드들(CH_{old})에게 참 조된다. 여기서, k 는 k 번째 클러스터의 번호를 의미하고, num_Cluster(k) 함수는 k 번째 클러스터에 속한 멤버 노드들의 수를 의미한다. 그리고, Vector() 함수는 클러스터에 속한 노드와 현재 클러스터 헤드(CH_{old})를 가지고 벡터로 변환하고, 또한 이전 상위 클러스터 헤드(SCH_{old})와 선출된 상위 클러스터 헤드(SCH_{new})로부터 벡터를 만들어, SimDirection() 함수의 두 매개변수로 사용함으로써 두 벡터 사이의 사이각(θ)의 유사도를 계산할 수 있고, 두 벡터 사이의 사이각은 $cos(\theta) = A \cdot B / |A| \ |B|$ 을 이용한다. 두 벡터 간의 사이각의 한계값으로 ±90°(즉, $0 \le cos(\theta) \le 1$)를 설정하였다. AvgEnergy(k) 함수는 k 번째 클러스터의 평균 에너지를 구하는 함수이고, Energy() 함수는 각 멤버 노드가 갖고 있는 잔여 에너지량를 구하는 함수이다. numOverlap() 함수는 클러스터 헤드와 후보 클러스터 헤드들과의 중첩된 영역에 속한 멤버 노드의 수를 구하는 함수이고, m 은 식 6에서 계산된 값이다. 만일 CostList가 공집합이면, 기존의 전통적인 셋업 단계(PreviousSetup()함수)를 수행하고, 공집합이 아니고 Cost() 함수의 반환값이 한계값(δ)이하일 경우, 난수(rand() 함수)를 발생시켜 클러스터 헤드를 선출한 후, 제안한 셋업 단계(ProposedSetup() 함수)를 수행하는 알고리즘을 나타낸다. Cost() 함수는 식 (9)를 이용하여 클러스터 헤드 후보 노드들의 Cost 값을 계산하여 반환하는 함수이다.

```
//num_Candidate: The number of Candidates(CH)
//CandidateList: List of the selected CH candidates using Direction
information and Average residual energy
//CostList: List of the selected CH candidates above threshold(m)
Input: Direction Information (Node_i, CH_old, SCH_old,
SCH_new), Residual Energy, Distance between Nodes and CH_old
Output: new Cluster Head(CH_new)

// Extract CandidateList
for(i=0; i<num_Cluster(k); i++){
 vecA = Vector(Node_i, CH_old)
 vecB = Vector(SCH_old, SCH_new);
if(SimDirection(vecA,vecB)) >= 0)
if( Energy(Node_i) >= AvgEnergy(k))
      CandidateList[num_Candidate++]+= Node_i
}
// Extract CostList
for(i=0; i<num_Candidate; i++)
if(numOverlap(CH_old ,CandidateList[i])>=m                    &&
Cost(CandidateList[i])<= δ)
CostList[cnt++] += CandidateList[i];
// Determine Proposed Setup phase or Previous Setup phase
if( cnt > 0){
 CH_new = CostList[ rand() % cnt ];
  if(cnt > γ)    // threshold γ
 R = R - α   // Radius R
ProposedSetup();
}
else {
R = R + α   // Radius R
PreviousSetup();
}
```

그림 2. 클러스터 헤드 선출을 위한 3DE_var 알고리즘

3. 에너지 효율적인 키 관리 기법(3DE_sec)

사전키 분배 단계는 k개의 랜덤 키를 분배하는 과정과 BS와 유일하게 통신 가능하도록 공유하는 개인키 설정 과정의 두 과정으로 이루어진다. 사전키 분배 과정에서는 모든 노드들이 WSN에 배치되기 전에 P개의 키를 가진 커다란 풀과 그들의 키 식별자를 생성하고, 각 센서 노드의 메모리에 P개의 키 중 k($k \ll P$)개의 키를 무작위로 가져온다. 이 때, 전체 P개의 키를 유일하게 생성하기 위한 각 키의 크기가 N 비트일 때, 2^N개의 서로 다른 키를 생성할 수 있기 때문에($P \leq 2^N$), 풀에 생성될 키의 크기는 $log_2 P$ 비트 이상이면 키의 유일성은 보장된다.

공유키 탐색 과정에서는 CH로부터 무선 통신 범위 내의 멤버 노드들과의 공유키를 탐색하는 단계로, CH는 자신의 키 ID를 브로드캐스트하여 멤버 노드들과 공유하는 키를 가지고 있는지를 알 수 있다. 이러한 공유된 키를 통해 노드들과의 안전한 링크를 설정하며 안전한 통신을 보장할 수 있다. 각 노드에 부여된 키 링은 이웃 노드들과 하나 이상 공유할 수 있는 확률은 식 (10)과 같이 P와 k의 수에 의해 계산한다.

$$sp' = 1 - \frac{((P-k)!)^2}{(P-2k)!P!}, 0 \leq sp' \leq 1 \quad (10)$$

스털링 공식(스털링 근사)을 적용하면, 이웃 노드들과 하나 이상 공유할 수 있는 확률은 다음 식 (11)과 같이 간략화된다.

$$sp' = 1 - \frac{(1-\frac{k}{P})^{2(P-k+\frac{1}{2})}}{(1-\frac{2k}{P})^{(P-2k+\frac{1}{2})}} \quad (11)$$

사전키 분배 단계를 거친 후, 각 노드에는 k개의 키 링과 BS와 안전한 통신을 위한 유일한 개인키를 가진다. 이때 각 노드에 부여된 k개의 키들을 이용하여 노드들 간의 공유키 설정을 통한 인증이고, 이를 1차 인증이라 하며 안전한 통신로가 수립된다

공유키를 통해 인증할 수 없는 경우, 이전 클러스터 형성 단계에서 BS로부터 신뢰된 인증 노드들로부터 인증을 받는다. 즉, 새로 선출된 CH와 클러스터 내의 노드사이에 공유키가 존재하지 않는 경우, BS를 통해 인증을 획득하면 BS의 오버헤드가 가중되며 전체 네트워크 트래픽을 증대할 수 있다. 따라서 기존의 CH로부터 획득한 클러스터 내의 노드들의 정보를 재활용하여 인증을 빠르게 수행할 수 있어 전체 지연시간을 최소화할 수 있다. 3DE_var와 같은 경우에서는 기존의 CH가 다양한 정보를 취합하여 최적의 CH를 선출할 수 있기 때문에 신뢰할 수 있는 노드로부터 인증 받을 수 있다. 또한 상위 클러스터 헤드를 통해 이전에 속했던 노드인지를 클러스터 키를 통해 인증하며, 이를 2차 인증이라 한다.

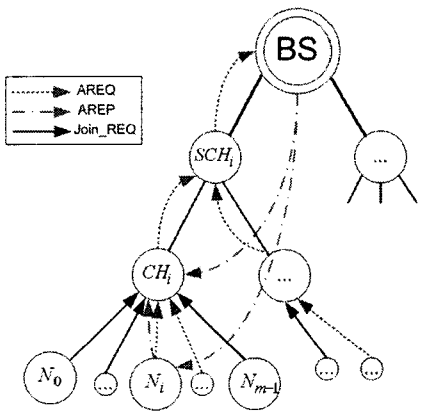


그림 3. 공유키가 없는 노드의 BS를 통한 인증 요청 및 응답 처리 과정

1차, 2차 인증 단계를 통해 인증을 받지 못한 노드인 경우, [그림 3]과 같이 각 노드에게 부여된 키 링의 키들을 랜덤한 위치의 값들을 선택하여 BS에게 인증 요청 패킷(Authentication Request Packet, AREQ)을 전송하여 BS의 인증 절차를 거친다. 즉 BS에게 인증받기 위해 자신의 키 링의 일부 값들을 무작위로 추출하고 개인키를 이용하여 암호화한 메시지를 선출된 CH에 전송한다. CH는 공유키를 갖지 않은 노드들로부터 획득한 암호화된 메시지를 BS에 전송하며, BS는 암호화된 메시지를 복호화하여 송신한 노드 ID의 인증 샘플 코드인 $< idx, BA, length, val >$ 쌍들의 집합과 일치하는지를 확인한다. 이 때 적용된 idx는 노드의 키 링에 존재하는 순서를 의미하고, BA는 기준 주소(Base

Address), *length*는 길이, *val*은 값(value)을 의미하며, 이러한 값들은 랜덤 함수를 이용하여 선택하고, *val*은 *BA*로부터 *length* 만큼 떨어진 위치까지의 키 값들을 나타낸다. 일치하는 경우에 인증 응답 패킷(Authentication Reply Packet, AREP)을 전송하며, 일치하지 않고 통신 방해를 위한 악의적인 노드들의 ID를 검출하여 각 CH에 전송하여 모든 노드들의 키 링으로부터 제거하도록 한다.

IV. 실험 및 성능평가

본 논문에서 제안된 클러스터 헤드 선출을 위한 3DE_var 알고리즘을 기존의 클러스터링 방법들과 성능을 평가하기 위해, NS-2[8] 시뮬레이터를 사용하여 구현하였고, 시뮬레이션 환경에 사용된 매개변수와 그에 대응하는 값은 [표 2]에서 보여주고 있다.

표 2. 시뮬레이션에 사용된 매개변수와 해당 값

매개변수	값
Network grid	(0,0) ~ (100,100)
Base Station	(50, 175)
Threshold distance	75m
Cluster radius	25m
E_{elec}	$50nJ/bit$
εfs	$10pJ/bit/m^2$
εmp	$0.0013pJ/bit/m^4$
εaggregation	$5nJ/bit/signal$
Data packet size	500 bytes
Broadcast packet size	25 bytes
Packet header size	25 bytes
Initial energy	2J/battery
Number of nodes	100
Number of clusters	5

또한, 키 관리를 위한 성능을 평가하는 부분에서 기존의 연구에는 네트워크 내의 안전한 통신을 위한 노드 수에만 언급하고, 네트워크의 크기($M \times M$)에 대한 언급이 없이 단순한 전체 노드수를 확장하여 실험하였다. 그러나 네트워크의 밀도는 전체 클러스터 형성 시간의 지연시간과 밀접한 관계를 가지고 있으며, 전체 네트워크의 크기 또한 무선 센서에게는 큰 오버헤드로 작용한다. 따라서 실험에서는 네트워크의 크기($100m \times 100m$)를 제한하고, BS로부터 네트워크 영역까지의 거리에 따른 최적의 클러스터 수(k_{opt})[2]를 이용하여 클러스터를 형성하였다.

$$k_{opt} = \frac{\sqrt{P}}{\sqrt{2\pi}} \sqrt{\frac{\epsilon_{fs}}{\epsilon_{mp}}} \frac{M}{d_{toBS}^2} \tag{12}$$

P은 WSN을 구성하는 노드들의 수이고, d_{toBS}는 BS까지의 거리이다.

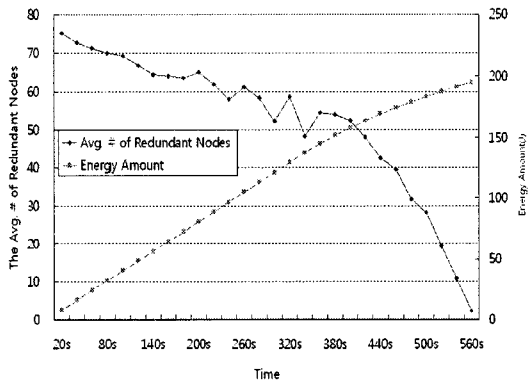


그림 4. 시간에 따라 변경되지 않는 평균 중복 노드 수와 전체 평균 에너지 소모량

[그림 4]는 무작위로 생성한 100개의 멤버노드들을 가진 100개의 테스트 집합을 이용하여 클러스터 헤드를 선출하는 과정에서 일정 시간동안 변경되지 않는 평균 멤버 노드의 수가 큰 차이가 없다가 불능 노드의 수가 급증하는 약 420초 지점부터 급격히 감소하며, 이때의 평균 에너지 소모량이 약 1.66J이었다. 따라서 본 논문에서는 불능노드와 생존 노드의 관계를 이용한 밀도 계산에서 한계값으로 0.83(1.66J/2J)을 설정하고 수행하였다. 실험에 사용된 100개의 샘플들 중 하나의 결과는 전체 28,366개의 변경되지 않은 멤버 노드가 있었으며, 이 중에서 유효한 멤버 노드의 수는 21,219개 추출되었다. [표 2]의 매개변수를 적용하였을 때, 이러한 변경되

지 않은 멤버 노드들의 불필요한 송수신의 제거로 상당한 양의 에너지가 절약됨을 알 수 있다.

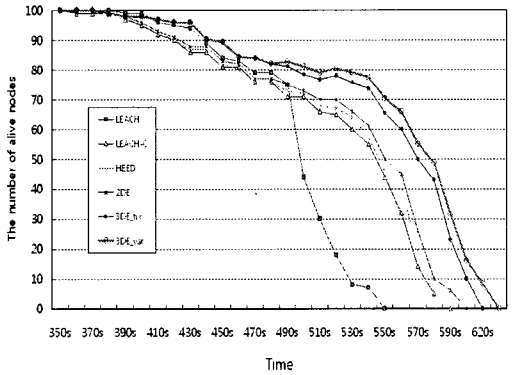


그림 5. 시간에 따라 변경되는 생존 노드 수의 비교

[그림 5]는 시간에 따라 변경되는 생존 노드의 수를 비교한 그래프이다. [그림 5]와 같이 LEACH, LEACH-C, HEED, 2DE[9], 3DE_fix[10], 그리고 3DE_var의 불능 노드가 처음 발생하는 시점은 각각 390초, 360초, 380초, 380초, 381초, 그리고 385초였고, 마지막생존 노드가 발생하는 시점은 각각 547초, 580초, 590초, 614초, 620초, 그리고 629초에 발생하였다. 이것은 전체 센서 네트워크의 생존 시간을 3DE_var가 LEACH, LEACH-C, HEED, 2DE, 그리고 3DE_fix 와 비교하여, 각각 15.0%, 8.4%, 6.6%, 2.6%, 그리고 1.6% 정도 증가되었다. 기존의 LEACH나 LEACH-C에 비해 전체 멤버 노드들의 셋업 단계에서 사용되는 불필요한 에너지를 감소시킴으로써, 전체 에너지 효율을 높일 수 있었다. 또한 HEED는 클러스터 헤드를 선출하기 위해 이웃노드들과의 잦은 메시지 교환을 통해 정보를 송수신하여 불필요한 에너지를 낭비하였고, 본 논문에서 제안한 3DE_var는 방향 정보를 이용하여 전체 센서 네트워크가 균형을 맞춰 고립되는 센서 노드들을 최소로 하고, 또한 HEED와 다르게 거리에 따라 이웃 노드들의 개수를 파악함으로써 에너지를 효율적으로 사용할 수 있었다. 비록 LEACH에 비해 3DE_var는 기지국과의 정보 교환 또는 클러스터 내의 정보 교환으로 인한 부담으로 먼저 불능 노드가 발생하지만 시간이 흐름에 따라 생존 노드가 많이 분포되어 중첩된 멤버 노드들의 누적된 에너지로 안정 단계에서 효율적으로 사용됨을 알 수 있다.

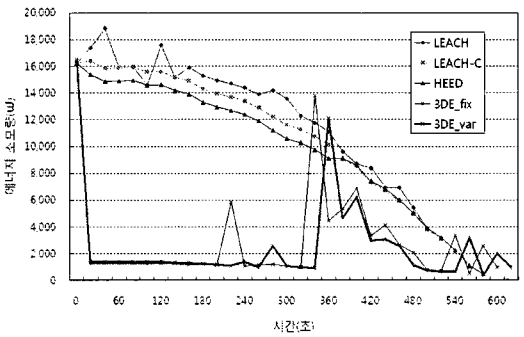


그림 6. 각 라운드마다 사용된 Setup 단계에서의 에너지 소모량 비교

[그림 6]은 각 라운드마다 셋업 단계에서 소모된 에너지량을 비교한 것이다. 3DE_fix와 3DE_var의 경우, 이전 클러스터와 새로운 클러스터에 모두 포함되는 센서 노드들은 매 라운드 초반의 Setup 단계에 참여하여 불필요한 에너지를 낭비할 필요가 없기 때문에 평균 2.749mJ의 에너지 소모량을 보였다. 비록 유효 중복수의 한계값을 넘지 못한 경우는 LEACH-C처럼 자신의 에너지 정보를 기지국에 전송하여 기지국에의해 제어되는 경우 기존의 방법들처럼 많은 에너지 소모를 보였다. 특히, 불능 노드들이 많이 발생하는 라운드 이후에는 기존의 방법들과 차이가 나지 않았다. 그러나 3DE 계열이 LEACH, LEACH-C, HEED와 비교하여 약 4.7배, 4.1배, 3.3배 이상 에너지가 적게 소모되었다.

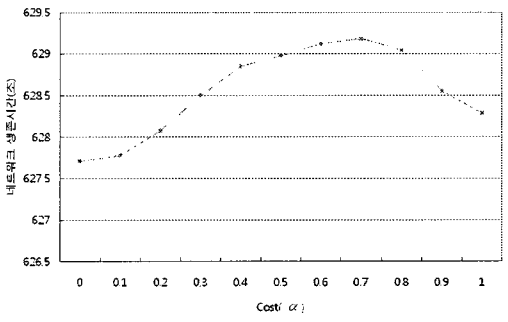


그림 7. Cost의 가중치에 따른 네트워크 생존시간

[그림 7]은 Cost를 계산하기 위해 사용된 가중치(α)에 따른 네트워크 생존 시간을 비교한 것이다. α의 값이 0.6에서 0.8일 때, 적절한 클러스터 헤드를 선출하여 효율적인 에너지 사용됨을 볼 수 있다. 그러나 잔여 에너지량을 고려하지 않고 밀도만 고려하는 0.9나 1.0과 같은 가중치를 주어졌을 때, 오히려 성능이 떨어진 이유는 클러스터 헤드 선출시 클러스터 내의 노드들의 평균 에너지량 이상의 노드들 중에서 밀도와 잔여 에너지량을 고려하였기 때문에, 잔여 에너지량에 비해 밀도가 더욱 의존함을 보였다.

3DE_var에 안전한 통신을 위한 키 관리 기법을 병합한 3DE_sec에서는 매 라운드마다 미인증 노드의 인증을 위해 BS까지의 전송 에너지량은 $n \times E_{TX} \times Length'$이고, HIKES에서의 BS까지의 전송 에너지량은 $N \times E_{TX} \times Length''$이라고 했을 때, 본 논문이 HIKES에 비해 효율적인 에너지 성능을 위해서는 총 에너지 사용량이 적어야 하며 다음 식 (13)과 같다.

$$n \times E_{TX} \times Length' < N \times E_{TX} \times Length''$$
$$Length' < \frac{N \times Length''}{n} \quad (13)$$

여기서, $Length''$는 각 키의 크기를 나타내며, $Length'$는 BS와의 인증을 위한 인증 샘플 코드의 크기, $\{< idx, BA, length, val >\}^*$를 나타낸다.

표 3. 에너지 효율을 위한 키 링의 크기에 따른 인증 샘플 코드의 크기(P=10,000)

k	sp'	비공유키수	인증 샘플 코드 크기의 최대값	Length'
50	0.2222	7,778	〈 21 bit	38 bit
94	0.5901	4,099	〈 40 bit	39 bit
95	0.5979	4,021	〈 40 bit	39 bit
100	0.6358	3,642	〈 44 bit	39 bit
162	0.9306	694	〈 231 bit	40 bit
200	0.9831	169	〈 947 bit	40 bit
220	0.9929	71	〈 2254 bit	40 bit
250	0.9984	16	〈 10000 bit	40 bit

[표 3]은 기존의 키 관리 기법과 본 논문에서 제안한 3DE_sec와의 필요한 저장 용량[7]을 비교한 것이다. 전체 노드의 수가 10,000개일 때, 키 링의 크기 k의 값을 94로 가정하면, 각 노드가 갖는 공간 사용량이 $100 \times |Key|$으로 LEAP, RPK, HIKES에 비해 필요한 공간 성능이 각각 19.4%, 60%, 그리고 17.4% 적게 사용됨을 알 수 있다. 이때 $|Key|$는 키의 크기를 나타내며, 에너지 효율보다 WSN 내의 안정성을 높이기 위해서는 식 (11)의 k의 값을 높이면 된다.

V. 결 론

본 논문에서는 클러스터링 방법에서 각 라운드마다 수행해야 하는 셋업 단계를 상황에 맞게 선택하고, 전체 네트워크의 멤버 노드들이 고립되지 않고, 각 클러스터에 균등하게 분포시키기 위해 방향성을 고려하여 클러스터 헤드를 선출하는 변형된 셋업 방법과, 클러스터 기반에 적합한 안전한 통신을 위한 키 관리 기법을 제안하였다. 클러스터 헤드가 선출되었을 때, 대부분의 라운드에서 이전의 클러스터 헤드와 선출된 클러스터 헤드 사이에 중복된 멤버 노드들의 불필요한 정보교환을 최소화하고, 가변적으로 클러스터의 반경을 조절함으로써 에너지 효율성을 극대화하였다. 뿐만 아니라, 키 관리 부분을 기존의 3DE_var에 병합하여 기존의 키 관리 기법과 비교한 결과 우수한 성능을 보였다.

향후 연구 과제로 클러스터에서 삭제될 노드들을 이전 클러스터 헤드가 정보를 획득하여 선출된 클러스터 헤드에게 전송함으로써 지연 시간이 추가되어 셋업 단계의 구축 시간을 증가시켰다. 따라서 셋업 단계의 지연 시간을 최소화하고, GPS의 사용과 같은 제약사항을 극복할 수 있는 연구가 필요하며, 센서 네트워크 내의 멤버 노드들 간의 이동성까지 고려한 더욱 안전한 클러스터 헤드 선출 기법에 관한 연구가 필요하다.

참 고 문 헌

[1] W. R. Heinzelman, A. P. Chandrakasan, and H. Balakrishnan, "Energy-Efficient Communication

Protocol for Wireless Microsensor Networks," Proc. 33rd Hawaii Int'l. Conf. Sys. Sci., 2000(1).

[2] W. B. Heinzelman, A. P. Chandrakasan, and H. Balakrishnan, "An Application-Specific Protocol Architecture for Wireless Microsensor Networks," IEEE Trans. Wireless Commun., Vol.1, No.4, pp. 660-670, 2002(10).

[3] O. Younis and S. Fahmy, "Distributed Clustering in Ad-hoc Sensor Networks: A Hybrid, Energy-Efficient Approach," IEEE INFOCOM 2004, 2004(3).

[4] T. Murata and H. Ishibuchi, "Performance evaluation of genetic algorithms for flowshop scheduling problems," Proc. 1st IEEE Conf. Evolutionary Computation, Vol.2, pp.812-817, 1994(1).

[5] L. Eschenauer and V. D. Gligor, "A key management scheme for distributed sensor networks," In 9th ACM conference on Computer and communications security, pp.41-47, 2002.

[6] S. Zhu, S. Setia, and S. Jajodia, "LEAP: efficient security mechanisms for large-scale distributed sensor networks," In 10th ACM conference on Computer and communication security, pp.62-72, 2003.

[7] J. Ibriq and Imad Mahgoub, "A Hierarchical Key Establishment Scheme or Wireless Sensor Networks," Proceedings of 21st International Conference on Advanced Networking and Applications(AINA'07), pp.210-219, 2007.

[8] http://www.isi.edu/nsnam/ns

[9] 김진수, 최성용, 한승진, 최준혁, 임기욱, 이정현, " 변형된 셋업 단계를 이용한 클러스터 헤드 선출 프로토콜", 한국콘텐츠학회 논문지, 제9권, 제1호, pp.167-175, 2009.

[10] J. S. Kim, J. H. Lee, and K. W. Rim, "3DE: Selective Cluster Head Selection scheme for Energy Efficiency in Wireless Sensor Networks," In 2nd ACM International Conference on Pervasive Technologies Related to Assistive Environments (PETRA 2009), 2009(6).

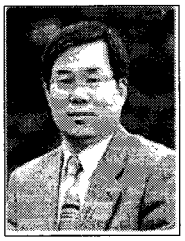
저 자 소 개

김 진 수(Jin-Su Kim) 정회원

- 1998년 2월 : 인천대학교 전자계산공학과(공학사)
- 2001년 8월 : 인하대학교 컴퓨터공학과(공학석사)
- 2001년 9월 ~ 현재 : 인하대학교 컴퓨터정보공학과 박사과정

<관심분야> : 유비쿼터스 컴퓨팅, 무선 센서 네트워크, 데이터마이닝, 정보검색

이 정 현(Jung-Hyun Lee) 정회원



- 1977년 2월 : 인하대학교 전자과(공학사)
- 1980년 9월 : 인하대학교 전자공학과(공학석사)
- 1988년 2월 : 인하대학교 전자공학과(공학박사)
- 1979년 ~ 1981년 : 한국전자기술 연구소 시스템 연구원
- 1984년 ~ 1989년 : 경기대학교 전자계산학과 교수
- 1989년 1월 ~ 현재 : 인하대학교 컴퓨터공학부 교수

<관심분야> : 자연어처리, HCI, 음성인식, 정보검색, 고성능 컴퓨터구조