무선네트워크 상에서의 침입탐지 에이전트 설계★

윤동식*

요 약

무선 네트워크(Wireless Network) 기술의 급속한 발전과 함께, 안전한 무선 통신을 위한 보안문제가 중요한 이슈로 대두되고 있다. 무선 네트워크에서 침입탐지 시스템을 운영하기 위해서는 탐지 에이전트가 각 무선 노드에 설치되어야 한다. Ad-hoc 네트워크 구조는 무선 네트워크상에서 AP가 없이 흩어져 있는 노드들에게 통신이 가능하도록 연결시키는 구조이다. 침입탐지 에이전트를 노드에 설치 할 경우 이에 해당하는 에너지 소모가 발생하여 생존기간이 줄어들게 된다. 또한 침입탐지 효과의 증대를 위해서는 많은 트래픽을 감시할 수 있는 노드에 침입탐지 에이전트가 배치되어야 한다. 따라서 본 논문에서는 Ad-hoc구조를 활용하여 무선 네트워크에서 네트워크의 생존기간을 최대로 하면서 침입탐지의 효과성을 동시에 고려한 침입탐지 에이전트 설치를 위한 방안을 제안하고자 한다. 또한 각 네트워크상에서 데이터 집계 시스템을 설계하여 데이터 중복을 줄이고 네트워크 에너지 소모량을 줄여 네트워크의 부하를 줄여 시스템 성능을 향상 시키고자한다.

Intrusion detection agents on the wireless network design

Yun Dong Sic*

ABSTRACT

Along with the rapid development of the wireless network (Wireless Network) technology for secure wireless communications, security problems have emerged as an important issue. In order to operate the wireless network intrusion detection system detects the agent installed on each wireless node should be. Ad-hoc network structures scattered in the AP over a wireless network without the node is a structure that makes it possible to communicate to connect. Intrusion detection agent to be installed on the node, and the corresponding energy consumption occurs when the survival time is reduced. On a node that can monitor a lot of traffic in order to increase the effect of intrusion detection, an intrusion detection agent should be placed. Therefore, in this paper, by taking advantage of the structure of Ad-hoc wireless network, considering the maximum living time of the network, while at the same time, the effectiveness of intrusion detection and intrusion detection by proposing a plan for installing the agent. Also improve the system performance by reducing the network load on each network, a system designed for data aggregation to reduce data redundancy, network energy consumption by reducing.

Key words : 무선네트워크, 침입탐지

접수일(2013년 1월 14일), 수정일(1차: 2013년 1월 22일), 게재확정일(2013년 1월 23일)

★ 본 논문은 안동과학대학교 교육역량강화연구지원에 의하여 연구됨.

* 안동과학대학교 의무부사관과

1. 서 론

특정 분야에서만 사용되던 휴대용 이동 단말기가 대중화 되고, 기기의 소형화와 사용 시간의 증가 및 사용 가능 지역의 확대와 같은 기술의 발달에 따라 소비자들은 점차 다양한 분야에서 다양한 목적을 충족시킬 수 있는 네트워크 환경을 요구하기 시작했다. 즉, 시간과 장소의 제약을 뛰어넘어 언제 어디서든지 인터넷을 비롯한 네트워크에 접속하여 다양한 작업을 수행하기를 희망하게 된 것이다. 이와 같은 대중의 요구사항을 충족시키기 위해 등장한 것이 바로 무선 네트워크 이다.

무선 네트워크는 크게 기지국이나 AP(Access Pointer)와 같은 기반 시설을 이용하는 Infra structured 네트워크와 이동 단말들로만 네트워크를 구성하는 Infra structureless 네트워크로 구분할 수 있다.

Infra structured 네트워크는 이동 전화망이나 무선 랜과 같이 유선 네트워크에 구축된 시설의 지원을 받아 무선 네트워크 내의 통신을 수행한다. 그러므로 기반 시설이 재해나 고장 등으로 인해 정상적으로 작동하지 않는 경우에는 무선 네트워크도 올바른 기능을 수행할 수 없다.

이와는 달리 순수하게 이동 단말들 간의 상호 작용으로 네트워크를 구성하는 Infra structureless 네트워크는 기반 시설을 이용할 수 없는 상황에서도 네트워크를 구축하여 통신을 수행할 수 있다. 네트워크를 구성하는 이동 단말들은 단말로서의 역할 뿐 아니라 통신 인프라의 역할도 수행 한다.

Ad hoc 네트워크는 Infra structureless 네트워크의 한 종류로서, 기존유선 네트워크 환경에서 제공하는 통신 인프라의 지원을 받을 수 없는 곳에서 이동 단말기 간의 라우팅 절차만으로 데이터의 송수신을 수행하는 무선 네트워크를 뜻한다. 자연 재해, 전시 상황과 같이 기반 시설이 없는 환경이나 기지국, AP 등의 고장으로 인해 유선 네트워크와의 단절이 발생한 경우에도 이동단말기 자신이 단말의 기능 뿐 아니라 라우터, 서버의 역할도 수행한다.

네트워크 내의 단말 각각의 동적 상태 및 위상 변화를 실시간으로 반영하여 신뢰성 있는 통신을 가능하게 한다. ad-hoc네트워크도 사물의 인식 정보 및 주변의 환경정보를 수집, 집계하여 사용자에게 실시간으로 제공 등 다양한 분야에서 활용되고 있지만, 제한된 에너지양과 메모리를 가지기 때문에 노드에서 에너지 효율성 및 네트워크의 수명 연장을 지원하는 다수의 데이터 집계 기법(Data aggregation technique)이 제안하였다.[1][2][3][4]

데이터 집계 기법은 하위 노드로 부터 받은 데이터와 자신의 데이터의 대표값 (예를 들면, 최대값(max), 최소값(min), 평균값(average), 데이터 수(count), 데이터합(sum))만을 집계하여 상위 노드로 전송한다. 이는 사용자가 원하는 값만을 전송하기 때문에 에너지 측면에서 효율적이며, 많은 응용 분야에서 활용되고 있다.

본 논문에서는 Ad-hoc구조를 활용하여 무선 네트워크에서 네트워크의 생존기간을 최대로 하면서 침입탐지의 효과성을 동시에 고려한 침입탐지 에이전트 설계를 위한 방안을 제시하고, 또한 각 네트워크상에서 데이터 집계를 위한 시스템을 설계하여 데이터 중복을 줄이고 네트워크 에너지 소모량을 줄여 네트워크의 부하를 줄여 시스템 성능을 향상 시키고자한다.

2. Ad hoc 네트워크

ad-hoc 네트워크의 구조에서는 중간에서 제어하는 노드가 없으므로 각 노드들은 자신이 가질 수가 있는 정보를 최대한 활용하여 네트워크에서 통신이 가능하도록 라우팅을 해야 한다. 이러한 통신 특성 때문에 센서 네트워크에서의 노드들 간의 무선 통신은 ad-hoc 통신에 의해 이루어지게 된다. 즉 ad-hoc 네트워크의 경우는 이동 단말들이 유선 환경에 기반을 둔 기지국이나 AP을 중심으로 구성되는 인프라가 있는 Intra structure네트워크와 달리 기지국이나 AP의 도움 없이 순수하게 이동 단말들로 구성된 인프라가 없는 (ad-hoc)네트워크로 무선센서 네트워크 시스템에서는 센서 필드내의 센서 노드들이 네트워크를 구성할 때 사용되는 네트워크 방식이다.

2.1. Ad hoc 네트워크의 라우팅 프로토콜

Ad hoc 네트워크에서 사용되는 라우팅 프로토콜은

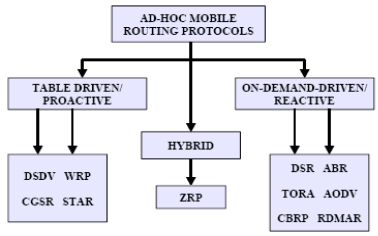
유선 네트워크에서의 라우팅 프로토콜과는 몇 가지 차이가 있다. 이는 Ad hoc 네트워크라는 환경이 단말의 이동성을 지원하고, 이를 관리하는 것을 목적으로 하고 있기 때문이다.

Ad hoc 네트워크에서 사용되는 라우팅 프로토콜은 제한된 대역을 최대로 활용하기 위하여 라우팅 오버헤드는 작게, 제한된 배터리를 낭비하지 않도록 하기 위해 과도한 플러딩(Flooding) 사용과 주기적인 메시지의 발생은 최소화하며, 동적인 위상변화를 빠르게 반영할 수 있어야한다. 그러므로 Ad hoc 네트워크에서 사용하는 라우팅 프로토콜을 설계할 때는 다음과 같은 사항을 고려해야 한다.

첫 번째는 플러딩의 제한적 사용에 관한 것이다. 플러딩이란 데이터를 수신한 패킷이 자신을 제외한 네트워크 내의 모든 노드에게 자신이 수신한 데이터를 전송하는 것을 말한다. 플러딩은 네트워크내의 모든 노드에게 전송해야 할 데이터가 있을 경우에는 유용하나 그 외의 경우, 특히 네트워크의 규모가 커질수록 자원의 낭비를 초래하게 된다. Ad hoc 네트워크는 사용할 수 있는 자원이 한정되어 있으므로 아무런 제한 없이 플러딩을 적용하면 대역폭이나 배터리 등의 자원을 불필요하게 소모하게 된다. 그러므로 TTL(Time To Live)이나 전송 범위를 설정하여 플러딩의 과다 사용으로 인한 자원낭비를 방지한다.

또 한 가지 고려해야 할 사항은 정확한 라우팅 정보의 반영에 관한 사항이다. 네트워크에서는 위상의 Ad hoc 변화가 자주 발생하기 때문에 라우팅 정보도 자주 변경된다. 그러나 변경된 라우팅 정보를 제대로 전달하거나 반영하지 못할 경우에는 무한 카운트 문제나 라우팅 루프가 발생할 가능성이 있다. 이는 비단 무선 네트워크 뿐 아니라 유선 네트워크에서도 발생해서는 안되는 문제이다. 무한 카운트와 라우팅 루프의 발생을 방지하기 위해 DSDV와 AODV 라는 라우팅 프로토콜에서는 목적지 시퀀스 번호를 이용하여 정보의 유효성을 검증하는 방법을 사용 한다.

위와 같은 특성을 고려한 Ad hoc 네트워크의 라우팅 프로토콜은 크게 table-driven (proactive) 방식과 on-demand(reactive) 방식으로 구분할 수 있다.



(그림 1) ad_hop 라우팅 프로토콜

Table-driven 방식은 네트워크 내의 모든 이동 노드들이 라우팅 정보를 주기적으로 혹은 변경사항이 발생할 때마다 네트워크 전체로 전파하게 한다. 이 과정을 통해 각 노드들은 자신의 라우팅 정보를 변경하여 항상 최신의 라우팅 정보를 유지하며, 데이터 전송 시에는 경로 탐색을 위한 지연 없이 유지하고 있는 라우팅 정보를 바탕으로 최적의 경로를 찾아낼 수 있다. 그러나 네트워크 위상의 변화에 따라 변경된 라우팅 정보도 방송되고 갱신되어야 하므로 위상 변화가 자주 발생할 경우에는 오버헤드가 발생한다.

Table-driven 방식의 대표적인 라우팅 프로토콜로는 DSDV(Destination Sequenced Distance Vector)가 있다.

DSDV는 Bellman-Ford 방식에 기초한 라우팅 프로토콜로서 목적지 시퀀스 번호를 사용하여 네트워크 위상 변화 시 발생할 수 있는 라우팅 루프를 방지한다 각 노드는 다른 . 모든 노드로의 경로 정보를 각각의 라우팅 테이블에 유지하고 있으며 갱신 시에는 노드 자신이 가진 모든 라우팅 정보를 다른 노드로 브로드캐스팅 하는 Full dump 방식과 라우팅 정보의 변경이 있을 경우에 새로 변경된 라우팅 정보만을 브로드캐스팅 하는 Incremental dump 방식이 있다.

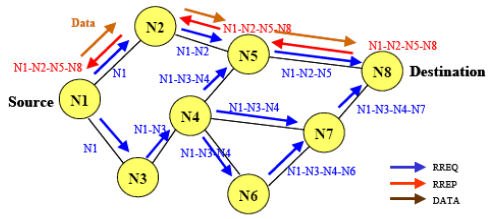
DSDV에서 사용하는 목적지 순차 번호는 새로운 라우팅 정보의 생성이나 갱신 시에 순차적으로 증가되므로 해당 정보가 최신 정보임을 구별할 수 있게 하고, 유효한 정보만을 반영하여 라우팅 루프의 발생을 미연에 방지 한다.

On-demand 방식은 데이터 전송을 위해 경로가 필요할 경우에만 경로를 탐색하는 방법으로서 주기적인 라우팅 정보의 전파와 갱신, 저장에 의해 발생하는 오버헤드를 감소시킬 수 있다. 반면에 데이터 전송 시점

에서 경로를 탐색하기 때문에 경로탐색으로 인해 전송되기까지 지연이 발생하게 된다.

On-demand 방식의 대표적인 라우팅 프로토콜로는 DSR과 AODV 방식이 있다.

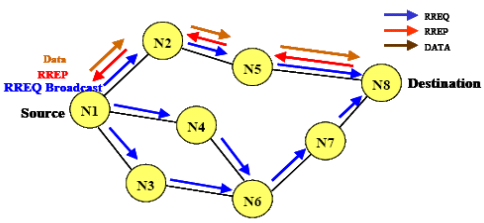
DSR (Dynamic Source Routing)은 이름에서도 나타나듯 소스 라우팅 방식을 기반으로 하며 네트워크 내의 노드들은 라우팅 테이블 대신 라우트 캐시를 유지하고 있다. DSR 방식은 경로 탐색 절차와 경로 관리 절차로 이루어진다. 데이터 전송을 위한 목적 노드로의 경로 정보가 존재하지 않을 경우 경로 정보 획득을 위해 RREQ (Route Request) 메시지를 이웃 노드로 브로드캐스팅 한다. RREQ 메시지를 수신한 중간 노드가 목적 노드로의 루트 정보를 라우트 캐시에 가지고 있지 않을 경우 자신의 주소를 RREQ에 추가하여 이웃 노드로 다시 브로드캐스팅 한다.



(그림 2) DSR 경로탐색

RREQ 메시지를 수신한 중간 노드가 목적 노드로의 경로 정보를 라우트 캐시에 저장하고 있을 경우, 목적 노드로의 경로 정보를 RREP(Route Request) 메시지에 추가하여 소스 노드로 전달한다. 링크의 손실이나 오류 발생할 경우에는 RERR(Route Error) 메시지를 생성하여 소스 노드로 전달한다. RERR을 수신한 노드는 자신의 라우트 캐시에서 해당 오류 발생 링크 정보를 삭제하며, 다른 경로가 있을 경우 이를 이용하여 데이터 전달을 계속하고, 그렇지 않을 경우에는 경로의 손실 을 알리기 위해 RERR 메시지를 소스 노드로 전달한다.

AODV(Ad hoc On-demand Distance Vector)는 같은 DSR과 마찬가지로 경로 탐색과 경로 유지의 단계로 나눌 수 있으며 DSDV에서와 같이 목적지 시퀀스 번호를 사용하여 라우팅 루프를 방지한다. 데이터 전송을 위해 경로가 필요한 경우 RREQ메시지를 생성하여 이웃 노드로 브로드캐스팅 하는 방식으로 경로를 탐색한다. RREQ가 목적 노드까지 전파되는 과정에서 RREQ를 수신한 중간 노드가 목적 노드까지의 경로 정보를 가지고 있다면 중간 노드에서 RREP 메시지를 생성하여 RREQ를 생성한 소스 노드까지 유니 캐스트 방식으로 전송한다. 그렇지 않을 경우 즉, 목적 노드까지의 경로 정보를 가지고 있지 않은 중간 노드는 RREQ 메시지를 이웃 노드로 다시 브로드캐스팅 한다. RREP 메시지는 목적지 노드까지의 경로를 탐색하는 RREQ 메시지에 대한 응답으로 RREQ를 수신한 노드가 저장한 역 경로를 이용하여 유니캐스트 된다. 노드의 이동으로 인해 링크가 손실되거나 오류가 발생한 경우에는 링크 손실이 발생한 노드의 주변에서 지역 복구를 수행하거 나 RERR 메시지를 RREQ를 생성한 소스 노드로 전달하여 오류가 발생한 링크에 관련된 라우팅 정보를 삭제하고 경로 재탐색을 수행하도록 한다.



(그림 3) AODV경로탐색

AODV는 다른 Ad hoc 라우팅 프로토콜에 비해 몇 가지 장점을 가지고 있다.

AODV는 table-driven 방식인 DSDV을 on-demand 방식으로 개선한 방식이기 때문에 DSDV의 특징을 일부 포함하고 있다. DSDV에서 라우팅 루프를 방지하기 위해 사용하는 목적지 시퀀스 번호는 AODV에서도 역시 같은 목적으로 사용한다. 또한 라우팅 테이블을 유지하여 경로에 대한 라우팅 정보를 저장한다.

그러나 DSDV가 AODV와 다른 점은 라우팅 테이블에 유지하는 데이터의 양과 기간이다. DSDV는 네트워크내의 모든 노드에 대한 전체 경로를 유지한다. 그러나 이런 방식은 네트워크의 위상이 비번하게 변경되고 대역폭과 같은 자원이 한정되어 있는 Ad hoc 네

트워크에는 적절하지 않다. 그래서 AODV에서는 이 점을 개선하여 경로 획득 절차에 의해 획득한 경로를 일정 시간만 유지하여 오버헤드를 감소시키므로 DSDV에 비해 보다 Ad hoc 네트워크에 적합한 라우팅 프로토콜이라고 할 수 있다. AODV는 같은 on-demand 방식인 DSR에 비해서도 효율적인 라우팅 프로토콜이다. DSR은 기본적으로 Source Routing 방식을 사용하기 때문에 경로 탐색을 통해 획득한 경로를 데이터 패킷헤더에 추가하여 전송한다. 노드의 수가 적은 네트워크에서는 데이터 패킷에 추가되는 데이터의 양이 많지 않으나 노드의 수가 많은 네트워크에서는 데이터의 크기가 커지므로 데이터 의 전송과 시간에 있어서 자원의 낭비를 초래한다.

3. 클러스터링 기법

네트워크의 통신 효율성을 높이고 네트워크의 확장성을 용이하기 위해서는 인접한 노드들을 하나의 그룹으로 묶어서 지역별로 노드들을 구성하게 되는데 이를 클러스터(cluster)라고 한다. 그리고 이 지역에서 대표가 되는 노드를 클러스터 헤드(cluster head)라고 한다.

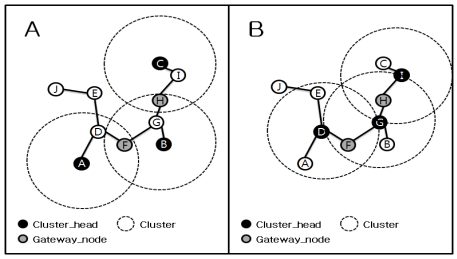
클러스터 헤드는 지역 내에 있는 노드들의 통신을 조정하기 위한 컨트롤러로서의 역할을 수행한다. 클러스터를 구성할 경우에는 싱크로 가는 통신 횟수를 줄일 뿐만 아니라 노드의 에너지를 절약하게 하는 특징이 있다. 지금까지 제안된 클러스터 알고리즘을 분류하면 세 가지 범주로 나눌 수 있다. 첫 번째 노드의 식별자 기반 클러스터 헤드 선정, 두 번째는 노드의 연결성 기반 클러스터 헤드 선정, 세 번째는 노드의 가중치 기반 클러스터 헤드 선정 알고리즘이다. 이들은 무선 애드 혹 네트워크를 기반으로 하여 제안 되었다. 클러스터링 기법으로는 단일 홉 클러스터링 기법과 다중홉 클러스터링기법으로 구분된다.[5][6]

먼저 다중홉 클러스터링의 경우 다중홉 환경 구축이 가능하기 때문에 네트워크 확장이 매우 용이하다. 하지만 넓은 범위의 네트워크를 유지, 관리하기 위한 제어 메시지의 증가가 불가피하기 때문에 오버헤드, 자원 낭비 등의 문제를 야기한다. 반면, 단일 홉 클러스터의 경우 클러스터 헤드와 멤버 노드간의 직접적인 통신이 가능하여, 소규모 통신에 효율적이며 클러스터 구성 및 유지에 안정적이다.

그러나 단일홉 클러스터의 단점으로는 클러스터 영역에서 제외되는 노드가 다중홉 클러스터링에 비해 많으며, 또한 노드의 이동성으로 인하여, 클러스터 헤드의 변경이나 클러스터 헤드의 이동 등으로 인하여 빈번하게 네트워크의 토폴로지가 변형되어 전체 네트워크 성능을 저하시키는 단점을 가지게 된다. 이러한 문제로 인하여 단일홉 클러스터링은 노드의 이동성이 제한되었을 경우 사용하게 된다. 보통 대규모 네트워크에서는 다중홉 클러스터링 기법을 사용하며, 소규모 네트워크일 경우 단일홉 클러스터링 기법을 사용하게 된다. 본 논문에서는 소규모 ad-hoc네트워크가 아닌 대규모 ad-hoc네트워크 상에서 다중홉 클러스터링 기법을 적용하였다.

Lowest ID Clustering 기법에서는 각 노드는 고유의 식별 번호(ID)를 가지고 있으며 정기적으로 이웃 노드들의 리스트를 전파하게 된다. 노드들은 주위에서 보내오는 정보를 수신하고 lowest-ID인 노드를 클러스터 헤드로 선정하는 기법이다. 여기서 lowest-ID란 수신 범위에 있는 노드들 중 가장 낮은 식별자(ID)를 가진 노드를 이르며, 자신이 클러스터 헤드로써의 역할을 포기하기 전까지 주변 노드들의 클러스터 헤드가 되며, 두 개 이상의 클러스터 헤드로부터 정보를 수신하는 노드는 게이트웨이가 된다. 그림 4에서 노드 A,B,C,D는 클러스터 헤드이고 노드 F,H는 게이트웨이다. 노드의 ID는 한번 설정이 되면 변하지 않기 때문에 클러스터 헤드의 전력이 고갈될 때까지 해당 역할을 수행하게 되는 단점이 있다.

Highest Connectivity Clustering기법은 노드의 연결 상태를 고려하여 클러스터 헤드를 선출하기 때문에 연결 기반 클러스터링 기법으로 불린다. 각 노드는 자신의 이웃 노드 정보를 동일한 주기로 브로드 캐스팅하여 가장 많은 이웃 노드(밀도)를 가지는 노드가 클러스터헤드가 된다. 이후 클러스터로 유입되는 다른 노드의 밀도가 현재 클러스터 헤드의 밀도보다 클 경우 해당 노드가 새로운 클러스터 헤드로 선정된다.



(그림 4) Lowest ID Clustering과 Highest Connectivity Clustering의 비교

그림 4는 Lowest ID Clustering과 Highest Connectivity Clustering의 비교한 그림이다. A그림은 Lowest ID Clustering기반 클러스터 형성을 나타낸 것이다. 여기서 노드 A, B, C가 가장 낮은 ID값을 가져 클러스터 헤드로 선정되고 두 개이상의 헤더 전송 반경에 있는 즉 두 개이상의 클러스터내에 포함되어 있는 노드 F, H는 게이트웨이이다. 노드의 ID는 한번 설정이 되면 변하지 않기 때문에 클러스터 헤드의 전력이 고갈될 때까지 해당 역할을 수행하게 되는 단점과 노드 E, J와 같이 클러스터 영역 내에 참여를 못하는 노드들이 생기는 단점을 가지고 있다.

B그림은 Highest Connectivity Clustering기반 클러스터 형성을 나타낸 것이며, 노드의 연결 가중치값이 가장 큰 D, G, I가 클러스터 헤드가 되며, 노드 F, H가 게이트웨이가 된다. 해당 클러스터링 알고리즘은 단순히 노드의 연결 가중치 값을 위주로 클러스터 헤드로 선정하고 노드의 잔여 에너지량을 고려하지 않기 때문에 클러스터 헤드로 선출된 노드는 다수의 다른 노드들을 관리하기 어려운 상태 일수도 있다. 또한 클러스터 헤드가 자주 변경되면서 전체적인 네트워크 토폴로지의 변화에 따른 클러스터의 재구성이 많은 계산량을 요구한다는 단점을 가지고 있다. 또한 단일홉 알고리즘의 특성상 클러스터 영역 내에 참여를 못하는 노드들이 생기게 된다.

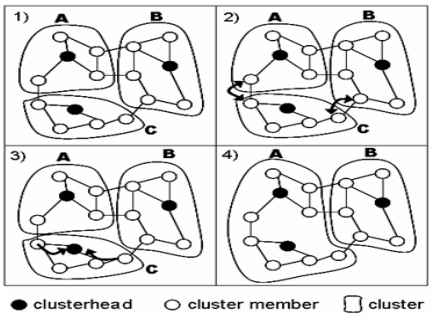
LEACH 프로토콜기법은 센서네트워크의 구조의 특성상 다수의 네트워크 노드에서 싱크로 데이터를 전송하면 싱크에 인접한 노드들은 전송량이 많아져 다른 지역의 노드보다 먼저 에너지 소모가 이루어질 수밖에 없다. 따라서 LEACH 프로토콜에서는 네트워크 노드간의 에너지 소모를 균등하게 하여 네트워크 생존시간을 최대화하기 위해 분산된 환경의 클러스터 기반의 네트워크 구조로 데이터 전송을 수행한다. 네트워크는 임의의 k 개의 클러스터를 구성하고 클러스터 마다 하나의 헤드노드를 선출한다. 클러스터 내부의 일반 노드들은 클러스터 헤드 노드로 데이터를 전송하고 클러스터 헤드 노드는 이를 병합하여 싱크에게 직접 전송한다. 에너지 소모가 균등하게 이루어지도록 일정 시간 마다 클러스터를 재구성하고 헤드 노드를 재 선출한다. 프로토콜은 클러스터 헤드 광고(Advertisement) 단계, 클러스터 설정 단계, 스케줄 생성 단계, 데이터 전송 단계를 거치게 된다. 클러스터 헤드 광고 단계에서는 모든 노드가 다음 식 (1)의 확률함수를 사용하여 값을 구한다.

$$P_i(t)=\begin{cases}\dfrac{k}{N-k*(r \bmod \frac{N}{k})} & : C_i(t)=1 \\ 0 & : C_i(t)=0\end{cases} \qquad (1)$$

위 식에서 i는 노드의 식별자, t는 시각, N은 전체 노드의 수, k는 클러스터의 수, r은 라운드를 나타낸다. $C_i(t)$는 최근 $r \bmod(\frac{N}{k})$라운드 동안 클러스터 헤드였다면 0이고, 아니라면 1이다. 즉, 최근 $r \bmod(\frac{N}{k})$라운드 동안 헤드를 한 번이라도 했다면 다시 뽑힐 확률은 없는 것이다. 이와 같은 확률함수를 통해 자신이 클러스터 헤드라면 이를 주변의 이웃노드에게 브로드캐스트 한다. 이 때 매체 접근 제어 프로토콜은 CSMA 방식을 사용한다.

Adaptive Multihop Clustering(AMC) 알고리즘은 애드혹 네트워크의 클러스터 형성시 멀티홉을 지원하는 방법으로써 각각의 노드들은 5개의 상태별로 분류하여 이상태 정보를 Hello 패킷을 담아 브로드 캐스팅하는 방법으로 토폴로지를 유지한다.

각각의 클러스터는 정해진 범위내의 노드만을 멤버로 가질 수 있고, 따라서 임계치를 벗어나는 수의 멤버를 가지게 된 클러스터는 병합과정을 하거나 불한하게 된다. 그림 5는 AMC에서 두 클러스터가 병합이 되는 예이다.



(그림 5) Adaptive Multihop Clustering

그림 5에서 클러스터 C는 최소 임계치를 벗어나서 인접한 클러스터와 병합을 요청하게 된다. 이중 게이트웨이를 거친 정보수집을 통해 클러스터 A의 멤버수가 적은 것을 판단하고, 클러스터 C는 A와 병합하게 된다.

그림 6는 노드들 사이에 주기적으로 주고받는 hello packet이며, 이러한 hello 메시지를 통해서 다중홉 클러스터링 토폴로지를 유지하게 된다. hello Message Format은 다음과 같다.

Node ID	CH ID	Hop count	State	Weight

(그림 6) Hello Message Format

- Node ID : 노드 ID는 해당 패킷을 송신하는 노드의 ID이다.
- CH ID : 해당 노드가 속해 있는 클러스터의 헤드 ID이다. CH ID를 이용하여 주위에 다른 클러스터의 멤버가 있는지를 판단할 수 있게 된다.
- Hop count : 클러스터헤드로부터의 홉 수를 기록하는 부분이다. 각 노드들은 주변으로부터 받은 Hello Message 중 가장 작은 홉 수에 1을 더함으로써 자신의 홉수를 설정할 수 있다.
- State : 현재 노드의 상태를 의미하는 상태 메시지이다. 예를 들어 새로 클러스터에 합류하게 될 노드가 Join request message를 보내게 되면 각 노드들은 해당 패킷을 헤드에게 전송하게 된다. 클러스터헤드는 새로운 노드의 가입을 알 수 있게 된다.
- Weight : 해당 Hello Message를 보내는 노드의 가중치로, 클러스터 헤드 선출시 Weight가 너무 낮으면 다른 노드로 역할을 넘기는데 사용된다.

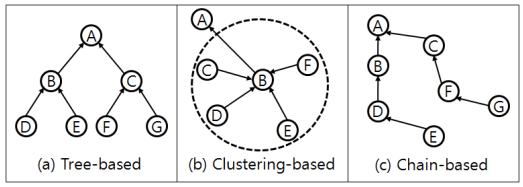
현재 ACM알고리즘을 응용한 여러 다중홉 클러스터링 기법이 있다. 노드의 가중치와 노드의 에너지량 그리고 노드의 평균 홉수를 통계내어 d round 마다 노드의 가중치 값에 의해 클러스터 헤드를 선정하는 AMHC기법이 있으며, AMHC 기법은 기존 ACM알고리즘에서 확장된 개념의 다중홉 클러스터링기법이다.

4. 데이터 집계 기법 설계

센서 네트워크는 기지국, 집계 센서 노드, 단말 센서 노드로 구성된다. 기지국은 큰 용량의 전원을 지니고 있어 많은 연산을 수행하며, 사용자와 센서 네트워크를 연결하는 역할을 수행한다. 집계 센서 노드는 하위 센서 노드로부터 수신된 데이터와 자신의 데이터를 수집할 수 있으며, 이를 이용하여 집계 연산을 수행한다. 단말 센서 노드는 수집한 데이터를 집계 센서 노드로 전송하는 역할을 수행한다. 센서 네트워크는 연결 그래프로 모델링 될 수 있다. 일반적으로, 데이터 집계 연산은 식(2)와 같이 표현된다.

$$y(t) = f(r_1(t), r_2(t), r_3(t), \ldots, r_N(t)) \qquad (2)$$

여기서, $r_i(t)$는 t 시간에 ri의 센서 노드가 수집한 데이터를 의미한다. 데이터 집계 함수는 최대값, 최소값, 평균값, 데이터 수, 그리고 데이터 합 등이 있다. 이러한 데이터 집계 연산은 전송되는 데이터의 양 또는 전송 횟수 등을 줄일 수 있기 때문에, 센서 네트워크 분야에서 라우팅 프로토콜과 더불어 가장 중요시 여겨지는 연구 분야이다. 센서 네트워크에서 데이터 집계기법은 라우팅 방법에 따라, 트리기반 데이터 집계 기법, 클러스터링 기반 데이터 집계 방법, 체인기반 데이터 집계 기법이 존재한다.[7][8]



(그림 7) 데이터집계 처리기법 3가지

본 논문에서는 클러스터링 데이터 집계 처리기법을 이용하며, 다중홉 클러스터링 기반 데이터 집계 연산을 위해 첫째로, 중복되는 데이터를 최대한 줄이기 위해 밀집도 계산을 통해 Hotspot_Zone을 생성하여, 대표노드가 해당 지역을 대표하여, 집계된 데이터를 전송하여 성능을 향상시켰다.

4.1. 클러스터링 헤드 선출 과정

본 논문에서의 클러스터링 헤드 선출 과정은 기본적인 구조는 AMC 알고리즘을 응용하여 설계하였다. 먼저 클러스터 하드를 선출하기 위해서는 R 라운드마다 정보 교환이 실행하여 매 라운드마다 가장 높은 가중치를 가지는 노드를 Cluster_head로 선출하게 된다. LEACH기법에서는 k값과 전체 네트워크의 노드의 개수인 N값을 이용하였지만 네트워크상에 클러스터의 헤드의 개수와 노드의 개수를 일반 노드가 확인할려면 전체 네트워크상에서 추가적으로 검색을 실시하게 된다. 이것은 추가적인 오버헤드를 유발하기 때문에 본 논문에서는 노드가 가지고 있는 정보만을 이용하여 Cluster_head로 선별한다. 노드의 클러스터헤드의 가중치 정보로는 AMC Hello패킷의 정보를 활용하며, hello메시지 패킷에서 활용할 수 있는 정보로는 Hop count와 노드 자신의 Weight 정보이다. 또한 노드의 현재 사용 가능한 에너지 잔여량을 추가적으로 가중치 정보로 활용하게 된다. 만약 노드가 해당 다운드에서 가장 큰 가중치 값을 가지게 된다면 해당 노드는 해당 라운드의 Cluster_head된다. 노드의 가중치는 식 (3)에 의해 결정된다,

$$Node\ Weight = (\frac{E_N}{E_M}\alpha + \frac{N_{My\,W}}{N_{nbW}}\beta) + \frac{IH}{RH} \qquad (3)$$

네트워크의 상황에 따라 α, β값을 지정해준다. 만약 에너지량으로 고려한 클러스터 헤드 선출이면 α값을 증가 시키고, 연결성을 고려한 클러스터 헤드선출이면 β을 증가시키게 된다. 이러한 값은 각각의 클러스터 헤드가 상황에 맞춰 값을 지정하거나 혹은 변경을 원하지 않을 경우 1로 지정한다.

식3-1에서 E_M는 에너지의 최대치 즉 100%상황을 말하며 E_N은 현재 남아있는 잔여량을 의미한다. N_{nbW}는 주변 노드의 가장 큰 연결 가중치 값을 의미하며, N_{MyW}는 현재 자신의 가중치 값을 의미하게 된다. R은 클러스터헤드를 선출하기 위한 라운드이며, R 라운드 주기를 의미하고 I는 현재 라운드를 의미하며, H는 해당 노드의 클러스터와의 Hop수를 의미한다. 이 값은 라운드 값이 증가 하면 증가 할수록 1에 수렴해지게 되는 것을 확인 할수 있다. 이것은 차후 기존의 클러스터 헤드에서 가까운 노드들이 최우선적으로 클러스터 헤드를 선정하겠지만, 나중에 가면 클러스터의 토폴로지가 변화가 이루어지더라도 현 클러스터 헤드와 떨어져 있는 노드들에게도 클러스터 헤드의 기회를 주기 위함이다. 클러스터 헤드의 비율은 네트워크의 노드배치에 따라 헤더 비율의 효율성이 변하지만, 기존연구에서는 적절한 클러스터 헤드의 비율 값을 5%로 추천하고 있으며, 논문에서 헤드의 비율 값 역시 5%를 지향한다.

4.2. 클러스터 헤드 광고 및 클러스터 가입

현재 라운드에서 스스로 클러스터로 선출된 노드들은 다른 모든 노드들 에게 클러스터 헤드 광고메시지(ADV)를 브로드캐스트 한다.

클러스터 헤드가 아닌 노드들은 이 단계 동안에 다른 모든 클러스터 헤드로부터의 광고 메시지를 수신한다. 이 메시지는 노드의 ID와 광고 메시지임을 구별할 수 있는 헤더만으로 구성된 짧은 메시지이다. 이 광고 메시지의 길이는 짧기 때문에 네트워크의 모든 노드들에게 이 메시지를 전달하기 위해 증가되는 에너지의 양은 큰 부담이 되지 않는다. 그래서 네트워크의 모든 노드들이 광고 메시지를 수신할 수 있을 만큼 송신 출력을 크게 설정한다. 그림 8은 광고메시지의 형식을 보여준다.

ADVERTISE	Node ID

(그림 8) 광고메시지 형식

클러스터 헤드가 아닌 노드들은 클러스터 헤드 광고 메시지를 수신하게 되면, 가입 요청 메시지를 Cluster_head에게 전송을 하게 되며, 가입 요청 메시지의 형식은 그림 9와 같다.

Join-REQ	Node ID	Cluster Head ID

(그림 9) 가입 요청 메시지 형식

광고 메시지를 수신후 가입 요청을 한 노드는 주변 노드를 1홉 단위로 탐색을 하여 해당 광고 메시지를 브로드캐스트 하게 되며, 주변 노드가 기존에 클러스터 가입 메시지를 받고 이미 Cluster_head에게 가입 승인 메시지를 보낸 노드들은 해당 메시지를 무시 하게 된다. 이렇게 순차적으로 주변노드가 탐색이 되지 않거나 혹은 주변 노드가 이미 자신의 클러스터 영역이나 혹은 다른 클러스터 영역에 포함되었을 경우 클러스터 형성 알고리즘을 끝내게 된다.

4.3. 클러스터 형성 완료

클러스터링 형성시 주의할 사항은 네트워크 영역 내에 Cluster_Zone의 크기와 수량이다. 이것은 네트워크 성능에 큰 영향을 끼친다. 만약 매우 큰 클러스터가 생성되면, 클러스터 헤드가 관찰해야 할 노드 수가 증가하고 이는 오버헤드를 유발하는 원인이 된다. 또한 크기가 작은 클러스터가 많으면 클러스터 헤드들 사이에서 발생하는 데이터의 교환이 증가하여 네트워크의 트래픽을 증가시킨다. 반면 멀티홉 클러스터 형성시 주의할 사항은 Cluster_head와 단말 노드의 거리이다. 예를들어 Cluster_Zone내에 Cluster_head에서 단말노드의 홉거리가 6이상이면 그만큼 Cluster_head와 거리가 먼 곳에 배치된것을 의미한다. 그럼 해당 단말 노드는 Cluster_head와의 통신비용이 그만큼 늘어나는 단점을 가진다. 기존연구에서는 적절한 클러스터 헤드의 비율 값을 5%로 추천하고 있으므로, 본 논문에서 마찬가지로 5%를 지향하며, 홉 거리는 k값이 5%이고 전체 노드의 개수가 100이면 하나의 클러스터 영역의 평균 노드의 개수가 $N_{average}$가 20개가 될 것이다. 또한 Cluster_head에 가중치가 높다면 노드가 밀집되어 있을 확률이 높기 때문에 식(4)을 통해 최대 홉 수를 지정을 하여 클러스터를 형성한다.

$$Hop_{\max} = \frac{N_{average}}{CH_{Weight}} \qquad (4)$$

만약 최대 임계치를 넘어서면 클러스터 분할 작업을 하며, 클러스터가 최소 임계치에 미치지 못한다면 해당 클러스터는 합병연산을 통해 전체적인 네트워크 분할 작업을 하여 균등 하게 나누어준다. 클러스터 헤드는 클러스터 내의 데이터 전송을 조정하는 중앙제어센터의 역할을 한다. 클러스터 헤드는 자신의 클러스터의 멤버가 되고자 하는 모든 노드들로부터 메시지를 수신하고, 멤버 노드의 수를 기준으로 각 노드들이 완료 전송할 시점을 지정 생성하여, 클러스터 내의 모든 노드들에게 브로드캐스트 한다.

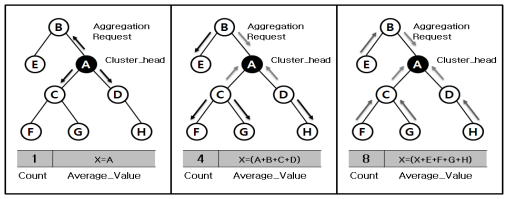
4.4. ad-hoc 네트워크의 효율적인 데이터 집계 처리

Hotspot_Zone영역을 노드 밀집도 계산을 통해서 Hotspot_Zone을 선정하며, Hotspot_Zone내에 에너지 잔여량이 높은 노드를 대표 노드로 선정하여 데이터 집계에 관한 요청이 들어오면 대표 노드의 집계된 데이터값만을 전송하여, 중복되는 정보의 수를 최대한 줄여 노드의 불필요한 통신정보를 줄여 노드의 에너지 소비량을 줄였다.

데이터 집계 방법으로는 라우팅 방법에 따라, 트리 기반 데이터 집계 기법, 클러스터링 기반 데이터 집계 방법, 체인기반 데이터 집계 기법이 존재한다. 보통 데이터 집계 연산은 각각의 노드들의 정보를 최단 라우팅 경로를 통해 Root_node에게 데이터를 전송하여 Root_node에서는 average값이나 SUM, MAX, MIN 연산을 수행하게 된다. 이것은 모든 노드에게 집계 요청 메시지를 일괄적으로 보낸 후에 모든 노드가 동시에 집계된 데이터를 전송하여 Root_node노드에서는 데이터 집계 연산을 하는 과정에서 클러스터영역 내에 많은 노드가 배치된다면 집계의 속도는 노드의 수에 비례하여 집계 결과값이 늦게 반환되는 문제점을 가지고 있다.

클러스터링 기반 데이터 집계 방법을 이용하여 데이터를 집계하되, 클러스터링의 구조는 멀티홉 클러스터링 기반으로 하였으며, 집계를 위하여 Cluster_head에서 데이터 집계 연산이 요청 메시지가 오면, 해당 노드는 주변에 다른 노드들에게 집계 요청 메시지를 전송함과 동시에 Cluster_head에게 자신의 집계데이터를 전송하여 즉시 집계 연산을 할 수 있도록 설계 하였다.

그림 10 은 average값을 집계하는 방법을 본 논문에서 제안하는 트리구조를 이용한 데이터 집계를 설명한 내용이다.



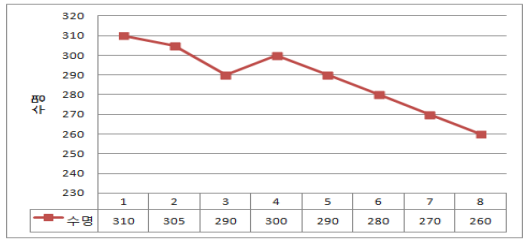
(그림 10) 트리구조를 이용한 데이터집계

먼저 Cluster_head내에는 데이터 집계 연산시 필요한 정보를 저장하는 간단한 데이터저장 공간을 가지고 있으며, 그림에서는 Count가 의미하는 것은 하위 노드로부터 요청한 데이터집계 응답 및 데이터를 전송 받으면 Count값을 1 증가 시키며, value는 집계된 데이터를 임시로 저장하는 공간이다. 그림에서는 average값을 value공간에 저장하는 것을 나타내며, average연산은 모든 노드로부터 데이터 집계한 정보를 이용하여 합계를 구한 후 count값을 이용하여 나눈 후 average값을 가지게 된다.

데이터 구조에서 표현 한 것처럼 평균값을 구함과 동시에 합계 그리고 MAX값, MIN값을 바로 확인 할 수 있으며, 기존의 데이터 집계 연산보다 더 많은 정보를 더욱 빠르게 확인 할 수 있도록 설계 하였다.

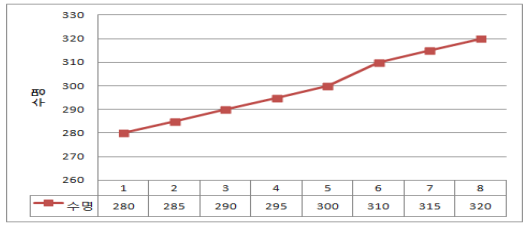
5. 시뮬레이션 및 성능 평가

본 논문에서 제안한 방법의 성능을 검증하기 위해, 시뮬레이션을 실시하였다. 제안하는 다중홉 클러스터링 기법과 데이터집계기법은 노드의 밀집도 변화에 따른 결과를 얻기 위해, 밀집도를 20%부터 10%씩 증가시켜 90%까지 변화시켜며, 또한 제안하는 역추적 기법에서는 기존 연구들을 바탕으로 비교분석 하였다. 먼저, 노드들이 불균형적으로 분포된 상황에서, AMCH기법의 경우 밀집도 변화에 따른 네트워크의 수명 변화를 확인하기 위하여 시뮬레이션을 통해 노드들의 평균 수명을 확인하여 보았다.



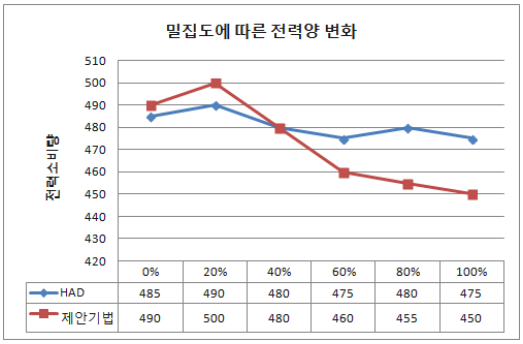
(그림 11) 밀집도 변화에 따른 노드 평균 수명

그림 11에서와 같이 AMCH기법에서는 노드들의 밀집도가 증가함에 따라 노드들의 평균 수명이 근소하게 감소를 확인 할 수 있었다. 이와 비교하기 위해 본 논문에서 제안하는 방법에서도 노드의 평균 수명을 측정하여 보았는데, 그림 12에서는 노드의 밀집도가 증가함에 따라 노드의 평균 수명이 길어짐을 확인 할 수 있었다.



(그림 12) 밀집도 변화에 따른 노드 평균 수명

노드가 균등분포가 되어있었을 경우 AMCH기법보다 노드의 평균 수명을 낮은걸 확인할 수 있는데 이것은 클러스터 헤드를 선정시에 Hotspot_Zone을 설정하는 통신 비용으로 인하여 평균 수명이 더 낮은걸 확인할 수 있다. 하지만 밀집도가 점차적으로 증가되면서 AMCH기법보다 본 논문에서 제안하는 기법의 노드 평균수명이 더 증가 되는 것을 확인할 수 있다. 이것은 밀집된 지역에 분포되어 있는 대기 노드들의 에너지 소비를 감소 시켜 노드들의 평균 수명을 증가 시키는 것을 확인할 수가 있다. 데이터집계 성능 평가 부분에서는 노드의 밀집도에 따른 클러스터헤드에서의 전력 소비량을 측정하였으며, 집계를 하고자하는 클러스터 헤드의 전력 소비량을 확인 하였을때 그림 13과 같이 제안된 기법이 기존의 HDA에 비해서 일집도가 높을 수록 전력소비량이 급속하게 감소하는 것으로 나타나고 있다.



(그림 13) 노드밀집도에 따른 전력양 변화

6. 결 론

Ad-hoc네트워크는 사물의 인식 정보 및 주변의 환경정보를 수집, 집계하여 사용자에게 실시간으로 제공 등 다양한 분야에서 활용되고 있지만, 제한된 에너지 양과 메모리를 가지기 때문에 노드에서 에너지 효율성 및 네트워크의 수명 연장을 지원하는 다수의 데이터 집계 기법이 제안되었으며, 데이터 집계를 위한 여러 데이터 보호 기법 등이 연구되어지고 있다.

본 논문에서는 효율적인 데이터 집계 기법을 위한 멀티홉 기반 클러스터링을 설계하였다. 멀티홉 기반 클러스터링 설계 부분에서는 기존의 논문에서는 단말 노드의 Hop수 즉 단말노드와의 거리와 최대 Hop수를 따로 선정하였으며, Cluster_head 선정시 네트워크의 현재 상황에 고려하여 노드의 잔여량과 노드의 연결 가중치를 추가적인 가중치 값을 활용하여 상황에 맞춰서 헤드 선정할 수 있도록 설계하였다. 네트워크상에서 데이터의 중복성을 최대한 줄여주기 위해서 노드의 밀집도를 계산하여, 노드의 밀집 영역을 Hotspot_Zone으로 설정하였다. Hotspot_Zone내에 있는 대기노드와 대표노드를 선정하여, 데이터 집계나 통신 경로를 사용할 때 대표노드가 대기노드를 대표하여 모든 데이터 집계자료나 통신 경로 역할을 하게 된다. 대기노드는 에너지를 보존함으로써, 차후 Cluster_head의 재선정시 Cluster_head로 선정될 가능성이 높아지기 때문에 전체적인 네트워크 토폴로지의 변화를 막을 수 있게 된다.

둘째로, 데이터 집계 설계 부분에서는 노드의 밀집도를 고려한 Hotspot_Zone을 이용하여 대표노드만이 데이터 집계를 하여 중복되는 데이터량을 줄여 네트워크 에너지 소모량을 줄였으며, 트리구조를 이용하여 데이터 집계 속도를 향상 시켜 노드들의 수명을 연장시키기 위해 빈번하게 재선출되는 Cluster_head호출을 줄여주어 Cluster_head의 수명을 증가 시키는 기법을 제안하였다.

본 논문에서는 Ad-hoc구조를 활용하여 무선 네트워크에서 네트워크의 생존기간을 최대로 하면서 침입탐지의 효과성을 동시에 고려한 침입탐지 에이전트 설계를 위한 방안을 제시하였으며 또한 각 네트워크상에서 데이터 집계를 위한 시스템을 설계하여 데이터 중복을 줄이고 네트워크 에너지 소모량을 줄여 네트워크의 부하를 줄여 시스템 성능을 향상 시켰다.

참고문헌

[1] K. Du, J. Wu, and D. Zhou, "Chain-based Protocols for Data Broadcasting and Gathering in Sensor Networks", Int'l. Parallel and Distributed Processing Symp., 2003.

[2] W. R. Heinzelman, "Application-Specific Protocol Architectures for Wireless Network", Ph.D. thesis, Massachusetts Institute of Technology, 2000.

[3] O. Younis and S. Fahmy, "HEED: a Hybrid, Energy-Efficient, Distributed Clustering Approach for Ad Hoc Sensor network", IEEE Trans. Mobile Computing, Vol. 3, No. 4, pp. 366~379, 2004.

[4] S. Madden, M. J. Franklin, and J. M. Hellerstein, W. Hong, "TAG: A Tiny Aggregation Service for Ad hoc Sensor Network", ACM SIGOPS Operating System Review, Vol. 36, pp. 131~146, 2002.

[5] Y. Wang, H. Chen, X. Yang, and D. Zhang, "WACHM : Weight based adaptive Clustering for large scale heterogeneous MANET", InProc.

of Communications and Information Technologys, pp. 936-941, Oct. 2007.

[6] Y. Wu, and W. Wang, "MEACA : Mobility and Energy Aware Clustering Algorithm for Constructing Stable MANETs", In Proc. of IEEE MILCOM, pp. 1-7, Oct.2006.

[7] H. Choi. S. Zhu, T. F. La Forta, "SET: Detecting Node Clones in Sensor Networks", In Proc. of IEEE 3rd IntelConf. on Security and Privacy in communication Networks. 2007

[8] Jeong Sam Kim. "Density Based Clustering Scheme for Wireless Sensor Network", KyungBuk National Univwrsity. 2010.

[저 자 소 개]

윤 동 식(Yun Dong sic)

1992년 관동대학교 정보처리학과 (공학사)
1994년 관동대학교 전자계산공학과 (공학석사)
2000년 관동대학교 전자계산공학부 (공학박사)
1999년~2008년 안동과학대학교 사이버테러대응과 교수
2008년~현제 안동과학대학교 의무부사관과 교수

e-mail : yundos@asc.ac.kr