DOI: 10.3745/KIPSTC.2009.16-C.2.151

센서 네트워크에서 mHEED를 이용한 에너지 효율적인 분산 침입탐지 구조

김 미 희[†] · 김 지 선[††] · 채 기 준[†††]

요 약

센서 네트워크는 유비쿼터스 컴퓨팅 구현을 위한 기반 네트워크 중의 하나로 그 중요성이 점차 부각되고 있으며, 네트워크 특성상 보안 기술 또한 기반 기술과 함께 중요하게 인식되고 있다. 현재까지 진행된 센서 네트워크 보안 기술은 암호화에 의존하는 인증 구조나 키 관리 구조에 대한 연구가 주를 이루었다. 그러나 센서 노드는 쉽게 포획이 가능하고 암호화 기술을 사용하는 환경에서도 키가 외부에 노출되기 쉽다. 공격자는 이를 이용하여 합법적인 노드로 가장하여 내부에서 네트워크를 공격할 수 있다. 따라서 네트워크의 보안을 보장하기 위해서는 한정된 자원의 많은 센서로 구성된 센서 네트워크 특성에 맞는 효율적인 침입탐지 구조가 개발되어야 한다. 본 논문에서는 센서 네트워크에서 에너지 효율성과 침입탐지 기능의 효율성을 함께 고려하여 침입탐지 기능을 분산적이고 동적으로 변화시킬 수 있는 분산 침입탐지 구조를 제안한다. 클러스터링 알고리즘인 HEED 알고리즘을 수정 (modified HEED, mHEED라 칭함)하여 각 라운드에서 노드의 에너지 잔량과 이웃 노드 수에 따라 분산 침입탐지노드가 선택되고, 침입탐지를 위한 코드와 이전 감시 결과가 이동 에이전트를 통해 전달이 되어 연속적인 감시 기능을 수행한다. 감시된 결과는 일반 센싱 정보에 첨부되어 전달되거나 긴급한 데이터의 경우 높은 우선순위 전달을 통해 중앙 침입탐지 시스템에 전달이 된다. 시뮬레이션을 통해 기존 연구인 적응적 침입탐지 구조와 성능 비교를 수행하였고, 그 결과 에너지 효율성 및 오버헤드, 탐지가능성과 그 성능 측면에서 뛰어난 성능 향상을 입증할 수 있었다.

키워드 : 센서 네트워크, 분산 침입탐지 구조, 에너지 효율성

Energy Efficient Distributed Intrusion Detection Architecture using mHEED on Sensor Networks

Mihui Kim[†] · Jisun Kim[††] · Kijoon Chae[†††]

ABSTRACT

The importance of sensor networks as a base of ubiquitous computing realization is being highlighted, and espicially the security is recognized as an important research isuue, because of their characteristics.Several efforts are underway to provide security services in sensor networks, but most of them are preventive approaches based on cryptography. However, sensor nodes are extremely vulnerable to capture or key compromise. To ensure the security of the network, it is critical to develop security Intrusion Detection System (IDS) that can survive malicious attacks from "insiders" who have access to keying materials or the full control of some nodes, taking their charateristics into consideration. In this perper, we design a distributed and adaptive IDS architecture on sensor networks, respecting both of energy efficiency and IDS efficiency. Utilizing a modified HEED algorithm, a clustering algorithm, distributed IDS nodes (dIDS) are selected according to node's residual energy and degree. Then the monitoring results of dIDSswith detection codes are transferred to dIDSs in next round, in order to perform consecutive and integrated IDS process and urgent report are sent through high priority messages. With the simulation we show that the superiorities of our architecture in the the efficiency, overhead, and detection capability view, in comparison with a recent existent research, adaptive IDS.

Keywords : Sensor Network, Distributed Intrusion Detection Architecture, Energy Efficiency

1. 서 론

무선 네트워크에서의 기밀성, 무결성, 인증, 부인방지를 위한 전통적인 보안 메커니즘으로는 인증 프로토콜, 전자서명, 암호화가 있다. 무선 네트워크와 마찬가지로 센서 네트워크에서도 악의적인 노드의 위협에 맞서기 위한 보안 메커

※ 본 연구는 정보통신부 및 정보통신연구진흥원의 대학 IT연구센터 지원사업의 연구결과로 수행되었음(IITA-2008-C1090-0801-0028).
† 정 회 원 : 미국 North Carolina State University 컴퓨터공학과 방문연구원
†† 준 회 원 : 이화여자대학교 컴퓨터공학과 석사과정
††† 종신회원 : 이화여자대학교 컴퓨터학과 교수
논문접수 : 2008년 9월 4일
수 정 일 : 1차 2008년 10월 13일
심사완료 : 2008년 10월 23일

니즘은 필수적이다. 하나의 예로써 보안 메커니즘 중의 하나인 암호화 및 인증 기술은 외부 노드의 공격에 맞서기 위한 보안성을 제공하지만 이미 암호화 키를 가지고 있는 내부의 악의적인 노드의 공격에 대한 보안은 제공하지 않는다. 특히, 전쟁터와 같이 군사용 센서 네트워크에서의 배치 환경은 쉽게 노드들이 위협(compromise) 받을 수 있는 환경이며, 다른 일반 센서 네트워크에서도 많은 노드들이 세심한 관리 밖으로 노출될 수 있기 때문에 내부 노드의 위협에 의한 공격이 용이하게 된다. 그러므로 이러한 내부 악의적인 노드에 대비하여 네트워크나 시스템을 조사 및 감시하고 필요한 조치를 취하는 시스템인 침입탐지 시스템 (Intrusion Detection System, IDS)은 필수적이다.

그러나 기존 무선 네트워크에서의 IDS 연구 결과는 센서 네트워크에서 특성상 그대로 적용하기 부적합하므로 이에 맞는 연구가 필수불가결 하다. 센서 네트워크의 IDS 주요 연구 이슈로는 에너지 효율성 및 많은 센서 노드 지원을 위한 범위성(scalability)를 고려하여 IDS 구조에 대한 연구[3,4,6-10]와 공격에 대한 대응 기술 연구[14-21]가 있다. IDS 구조에 대한 연구는 크게 각 노드에서 수행하는 Stand-alone 방식[3]과 계층적 방식[4,6,9], 분산 협업 방식[7,10]으로 나뉘어 연구되고 있고, 처리나 저장 가용성(capability)의 한계를 갖고 있는 센서 네트워크 특성상 랜덤 혹은 그리드 배치의 평평한(Flat) 구조의 센서 네트워크에서는 분산 협업 방식이, 계층적인 센서 네트워크에서는 계층적 IDS 방식이 적합하다. 센서 네트워크에서 주로 연구되는 공격으로서 분산 서비스거부 공격 (Distributed Denial of Service Attacks, DDoS), 슬립거부 공격(Denial of Sleep Attack), 재밍 공격 (Jamming Attacks), Sybil 공격, 프라이버시에 대한 공격들이 있고 이에 대한 대응 방안들이 연구되고 있다[14-20].

본 논문에서는 센서 네트워크에서 에너지 효율성과 IDS 기능의 효율성을 함께 고려하여 IDS 기능을 분산적이고 동적으로 변화시킬 수 있는 분산 IDS 구조를 제안한다. 에너지 및 IDS 기능 효율성을 고려하여 IDS 기능을 수행하는 적절한 노드 선정 시, 기존 클러스터 헤드를 선정하기 위한 알고리즘 중의 하나인 HEED 알고리즘을 적은 오버헤드를 요구하도록 수정하여 사용하고, IDS 수행노드 교체 주기인 라운드 마다 IDS 수행노드가 변경되면 이전 라운드의 IDS 수행노드에서 새 라운드의 IDS 수행노드로 해당 모니터링 결과와 탐지 코드를 에이전트화하여 전달하여 IDS 기능의 연속성 및 협업을 가능하게 한다. 본 분산 IDS 구조의 주요 요소로서 초기 IDS 구동을 초기화하고 탐지 결과를 통합하여 최종 판단을 주관하는 중앙 침입탐지 노드(cIDS), 매 라운드 마다 최적의 노드로 선정되어 지역적인 침입탐지를 수행하는 분산 침입탐지 노드(dIDS), 선정된 dIDS에서 침입탐지를 수행할 수 있도록 침입탐지 모델을 담고 있고 새로운 정상 트래픽 모델 생성을 위한 트래픽 수집 역할의 이동 에이전트(MA), 일반적인 센싱정보에 첨부된 dIDS로부터의 주기적 보고를 cIDS에 포워딩해주는 SINK가 있다. 각 dIDS는 주기적 보고 내용을 센싱정보에 첨부하여 SINK로 전송하면, 이를 모은 SINK는 주기적으로 cIDS에게 전달한다. 그러나 dIDS에서 긴급한 침입탐지 보고 내용은 높은 우선순위 메시지를 통해 cIDS에게 직접 전달한다.

본 논문의 구성은 다음과 같다. 1장의 서론에 이어 2장에서는 관련 연구로서 센서 네트워크에서 침입탐지 시스템 연구 동향과 본 논문의 비교 연구로서 사용될 적응적 침입탐지 구조[10]에 대해 간단히 기술한다. 3장에서는 본 논문에서 제안하는 에너지 효율적인 분산 침입탐지 구조를 자세히 설명하고, 4장에서는 제안한 구조의 효율성을 다각적인 시뮬레이션 결과를 통해 증명하며, 마지막으로 5장에서 결론을 맺는다.

2. 기존 연구

2.1. 센서 네트워크의 침입탐지 시스템 연구 동향

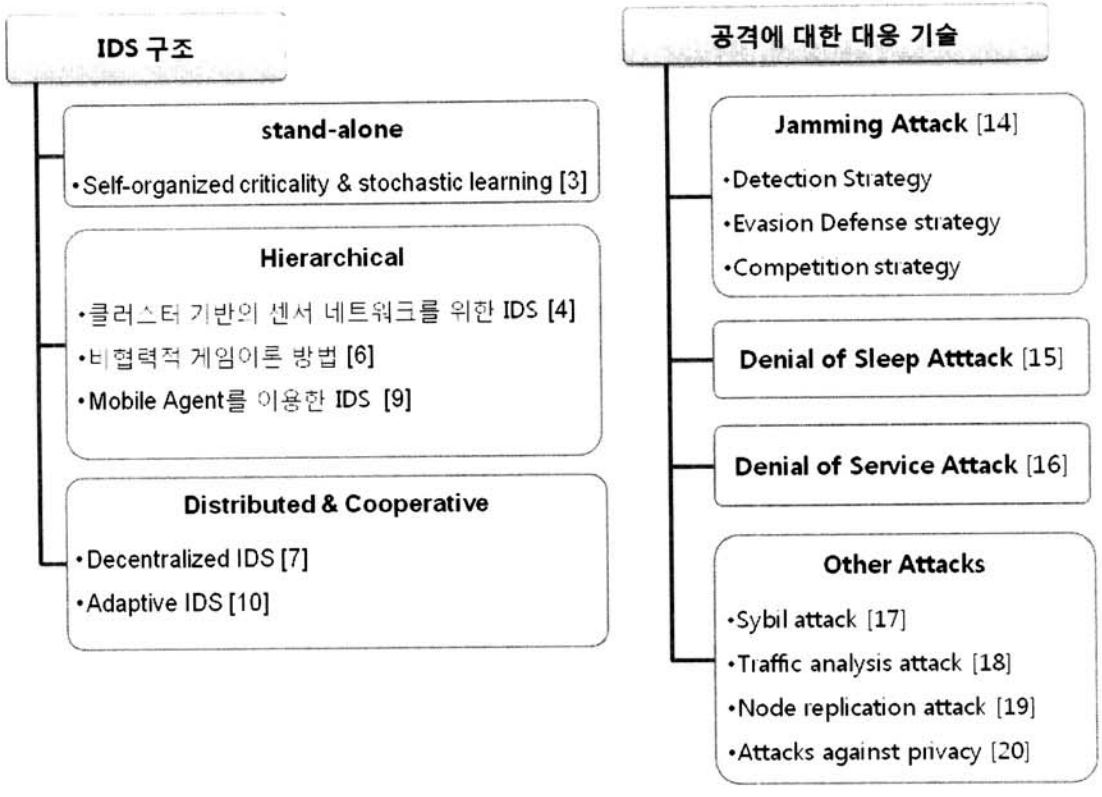
본 절에서는 최근에 진행되고 있는 센서 네트워크에서의 침입탐지 시스템 연구 동향을 기술한다. 관련 연구 동향의 주된 이슈는 (그림 1)에서와 같이 네트워크 특성을 고려한 IDS 구조에 관한 연구와 특정 공격에 대한 대응 기술 연구이다.

2.1.1. 센서 네트워크의 IDS 구조

유선 네트워크에서의 IDS 구조는 크게 호스트기반 IDS와 네트워크기반 IDS로 분류한다. 전자는 운영체제의 감사흔적, 시스템과 어플리케이션 로그, 시스템콜을 모니터링하는 모듈에 의해 생성된 감사 데이터를 가지고 침입탐지를 하는 반면, 후자는 네트워크 트래픽을 가지고 침입탐지를 수행한다. [1]에서는 무선 애드혹 네트워크와 센서 네트워크 특성에 맞추어 연구되고 있는 IDS 구조를 세 가지로 분류하였다. 첫째 Stand-alone IDS 구조에서는 각 노드가 독립적으로 IDS를 운영하고 자신을 위한 공격 탐지 기능을 수행한다. 또한 각 IDS는 어떠한 정보도 공유하지 않고, 다른 시스템과 협력하지도 않는다. 그러므로 이 구조에서는 모든 노드가 IDS를 운영할 능력을 가져야 한다. 둘째 Hierarchical IDS 구조의 경우, 클러스터헤드를 가진 여러 개의 클러스터로 나뉘어진 계층적인 센서 네트워크를 가정한다. 이러한 네트워크 구조에 맞춰 클러스터 안에서 각 노드는 어떠한 악의적인 행동 탐지를 위한 간단한 감시 역할을 수행하고 이를 클러스터헤드가 취합하여 통합하며, 상위에 있는 싱크노드에 전달하는 계층적인 구조를 나타낸다. 셋째 Distributed & Cooperative IDS 구조에서는 모든 노드 혹은 일부 노드가 자신만의 IDS 기능을 수행하고, 전역의 침입탐지 메커니즘을 수행하기 위해 IDS끼리 서로 협력한다. 다음은 각 구조에서의 대표적인 연구 내용을 기술한 것이다.

- *Stand-alone IDS*

Self-Organized Criticality & Stochastic Learning 기반 IDS: Doumit와 Agrawal[3]은 이벤트 발생에 기반한 비정상탐지방법 모델을 제안했다. 이 방법은 이전 데이터와 새로운 데이터를 비교하고 추측하여 비정상적인 행동을 탐지하고 환경



(그림 1) 센서 네트워크에서 침입탐지 시스템의 주요 연구 이슈

변수를 기반으로 비정상 탐지를 위한 임계값을 스스로 정하는 이점을 갖는다. 기존 유선 시스템을 위한 네트워크기반 IDS에서 사용된 Hidden Markov모델을 사용하여 오직 이전 상태만을 가지고 침입의 확률을 구하기 때문에 적은 메모리가 사용된다. 따라서 메모리 제한적인 센서 네트워크에서 효율적이라고 볼 수 있으나, 모든 노드가 계속적으로 IDS 활동을 해야 하기 때문에 배터리 파워가 제한적인 센서 네트워크에서는 오버헤드가 큰 시스템이라고 볼 수 있다.

■ *Hierarchical IDS*

클러스터 기반의 센서 네트워크를 위한 IDS: Su[4]는 클러스터 기반의 센서 네트워크의 보안을 향상시키기 위해 외부 공격자를 막기 위한 두 가지 방법을 제안했다. 첫 번째 방법은 모든 메시지에 메시지 인증 코드(MAC)를 추가하는 인증기반의 모델을 사용하는 방법이다. 각 노드는 역할(클러스터헤드, 멤버노드, 베이스스테이션)에 따라 Pairwise 키나 개인키에 의해 생성된 MAC과 Time Stamp를 추가하여 메시지를 보낸다. 두 번째 방법은 Energy-Saving 스킴으로 클러스터헤드와 멤버노드의 부정행위를 탐지하는데 초점을 맞추어, 클러스터헤드를 모니터링 할때 클러스터헤드의 모든 멤버노드가 다같이 협력적으로 모니터링하고, 멤버노드를 모니터링 할 때에는 멤버 노드들끼리 서로서로 모니터링하는 방식이 아니라 클러스터헤드가 멤버노드를 모니터링하기 때문에 에너지를 절약할 수 있게 된다. 그러나 이 방법은 각 노드가 많은 수의 공유키를 가지고 있어야 하고, 키가 분배된 이후의 키 갱신이 어렵고 이동성을 제공하지 않는다는 단점을 가지고 있다.

비협력적 게임이론 방법: Agah[6]는 클러스터링 기반 센서 네트워크에서 한정된 네트워크자원으로 인해 보안에 가장 취약한 노드를 찾고 그것을 보호하기 위한 게임이론 프레임워크를 제안했다. 방어를 위해 세가지 다른 스킴을 보이고 있는데, 첫번째 스킴에서는 공격자와 센서 네트워크 사이에 비협력적 게임을 정의한다. 게임의 Player로서 공격자는 공격을 할지 후에 더 좋은 공격을 위해 기다릴 지를 결정하고 센서 네트워크는 침입으로부터 센서 노드를 방어하기 위해 IDS를 사용한다. 이러한 게임이론 프레임워크를 사용하여 이 게임이 공격자와 IDS사이에 내쉬균형(상대방의 전략을 예상할 수 있다는 가정 아래 자신의 이익을 최대화시킬 수 있는 전략을 선택해 형성된 균형 상태를 의미)이 이루어짐을 볼 수 있고, 이로써 IDS가 침입에 대해 대처하기 위한 좋은 방어 전략을 이끌어 낼 수 있다. 두번째 스킴은 가장 취약한 센서노드를 예측하기 위해 Markov Decision Process(MDP)를 사용한다. 공격자의 이전 행동과 시스템의 과거 상태를 기반으로 MDP를 사용하여 가장 취약한 클러스터헤드와 공격의 목표가 될 가능성이 높은 목표물을 예측하는 방법이다. 세번째 스킴은 각 타임슬롯에 클러스터의 트래픽 양을 나타내는 Activity load를 이용하여 IDS가 보호하기 위한 클러스터를 결정한다. IDS는 이 값을 기반으로 방어하기 위한 클러스터를 선택한다. 그래서 각 타임슬롯의 트래픽 값이 가장 높은 클러스터가 가장 취약한 클러스터이고 이것을 공격으로부터 방어하는 방법이다.

Mobile Agent를 이용한 IDS: P. Kannadiga와 M. Zulkernine[8]은 Mobile Agent(MA)를 이용한 DIDMA를 제안했다. MA는 네트워크에 있는 모든 호스트를 이동할 수 있는 소프트웨어 개체로서 Static Agent(SA)로부터 받은 침입관련 데이터의 취합(aggregation)과 연관(correlation) 작업을 수행하여 침입을 탐지하는 방식이다. 이러한 방식은 일부 센서 노드에모니터링을 통해 지역적 보안을 담당하는 Nodal Agent(NA)를 탑재하여 센서 네트워크에 적용되었으나[9], 고정 노드에서 NA 수행에 따른 비균형적 에너지 소비의 가능성을 내포하고 있다.

■ *Distributed & Cooperative IDS*

Decentralized IDS: A.P. Silva[7] 등은 센서 네트워크의 요구사항과 제한사항을 고려하여 분산 배치된 모니터노드에서 다음 세 단계를 수행하는 IDS 구조를 제안하였다. 불규칙적으로 수집된 메시지에서 중요한 정보를 저장하는 데이터수집 단계, 저장된 데이터에 규칙을 적용하는 규칙적용처리 단계, 발생된 실패의 수가 예상되는 양 이상이면 침입탐지로 판단하여 경보를 울리는 침입탐지분석 단계가 그것이다. 여러 분산된 모니터노드들이 협력하여 IDS 역할을 수행하는 방식으로 센서 네트워크에 적합하지만, 역시 고정 모니터노드에서 IDS 역할 수행에 따른 비균형적 에너지 소비를 야기시킬 수 있고, 적합한 위치 선정에 과ㄴ해 고려하지 않고 있다.

Adaptive IDS: P. Techateerawat와 A. Jennings[10]는 이웃 노드 수에 기반한 투표와 각 노드의 서로 다른 임계값을 기반으로 각 라운드에서 노드의 IDS 수행 여부를 결정하는 방식을 사용하여 하나의 노드가 계속적으로 IDS 역할을 하지 않고 라운드 마다 돌아가면서 IDS 역할을 하는 Adaptive IDS를 제안하였다. 만일 이번 라운드에 IDS가 되면 임계값을 증가시켜 다음 라운드에 IDS가 되는 확률을 줄이고, 이번 라운드에 IDS가 되지 않으면 임계값을 감소시켜 다음 라운드에 IDS가 되는 확률을 높이는 방식이다. 이 방식은 본 논문에서와 같이 평평한(Flat) 센서 네트워크 구조를 가정하였고, IDS 역할 선정 방식을 제안한 방법이어서 본 논문 성능평가 시 비교 논문으로 사용하였다. 실험 결과, 각 라운드 마다 불균형적인 IDS 수를 나타내었고, 투표 시, 노드의 남은 에너지를 고려하지 않기 때문에 라운드가 진행 될수록 에너지 분포가 고르지 못하여 비효율적인 에너지 사용의 단점을 나타내었다. 관련 결과는 4장에서 자세히 기술한다.

지금까지 살펴본 IDS 구조 중에 자원 제약이라는 특성을 고려하면 Hierarchical IDS와 Distributed & Cooperative IDS 구조가 센서 네트워크에 적합하다고 할 수 있다. 본 논문에서는 랜덤분포나 그리드분포와 같이 평평한 센서 네트워크 구조에서 에너지 효율성을 고려하여 적합한 IDS 위치를 선정하고, 이동 에이전트를 활용하여 매 라운드마다의 감시 결과를 취합할 수 있는 Distributed & Cooperative IDS를 제안하고자 하며, 이는 계층적 센서 네트워크에서도 적용가능하다.

2.1.2 공격에 대한 대응 기술

또 하나의 주요 IDS 연구 이슈로 특수 공격에 대한 대응 기술 연구가 있다. 센서 네트워크에서 주로 연구되는 공격으로는 재밍 공격(Jamming Attacks), 슬립거부 공격(Denial of Sleep Attack), 분산 서비스거부 공격 (Distributed Denial of Service Attacks, DDoS)이 있고, 기타 공격으로는 Sybil 공격, 트래픽 분석 공격, 노드 중복 공격, 프라이버시 침해 공격등이 있다.

연구되고 있는 재밍 공격 대응 기술로는 크게 탐지, 회피, 경쟁전략으로 나뉘어 진다. 탐지전략은 신호의 세기, 캐리어 센싱 타임, 패킷 전송율을 기반으로 공격을 탐지하는 것이고, 회피전략으로는 재밍지역만 채널을 변경해 주는 Channel Surfing 방식과 재밍지역을 철수하여 안전한 지역으로 이동하고 네트워크를 분할하여 공격자로부터 예방하는 방법인 Spatial Retreat 방식이 있다. 마지막으로 경쟁전략으로는 강력한 Error Correcting Code를 사용하여 성공적인 패킷 전송 가능성을 증가시키는 방법과 전송 파워를 증가시켜 합법적인 라디오 장치에 의해 발생한 노이즈보다 높은 신호로 작동하게 하는 방법이 있다[14]. 센서 네트워크에서는 에너지 제약이라는 센서 노드의 취약점을 악용하여 슬립 모드로 들어가지 못하게 하여 빠른 에너지 고갈을 야기시키는 슬립거부 공격이 위협적인 공격의 하나로 대두되고 있고, 이에 관련된 대응 기술이 연구되고 있다[15]. 다른 네트워크와 마찬가지로 제한된 처리용량을 갖는 센서 노드에서도 각 레이어별 서비스거부 공격의 가능성이 존재하므로 이에 대한 대응 기술들이 연구되고 있다[16].

또한 다수의 아이디를 도용해 투표기반 라우팅 메커니즘이나 다른 제어 메커니즘에 참여하여 제어를 공격자가 원하는 방향으로 바꾸게 하는 Sybil 공격, 개인 프라이버시 정보가 중요한 센서 네트워크 응용에서 이에 대한 침해 공격 등, 센서 네트워크 특유의 공격에 대한 대응 기술 또한 활발히 연구되고 있다[17-20].

2.2. 적응적인 침입탐지 구조

본 절에서는 앞서 설명한 센서 네트워크에서의 침입탐지 시스템 연구 중에서 본 논문에서 제안하는 방법과 비교하고자 하는 기존 연구 방안에 대해 간단히 기술하고자 한다. 선정된 "적응적인 침입탐지 구조[10]"(Adaptive_dIDS라 칭함)는 본 논문과 가정하고 있는 센서 네트워크 구조가 가장 비슷하며 최신 연구 결과 중의 하나이다.

Adadptive_dIDS의 특징은 어떤 노드가 IDS 기능을 수행할지 degree값 즉 이웃 노드수에 기반하여 투표(voting)를 하고 투표의 결과를 각 노드가 관리하는 2개의 임계값과 비교하여 자신이 IDS 기능을 수행할지의 여부를 결정하고 임계값을 조정한다. 이 때, 한 노드가 투표에 선출 가능한 이웃 노드와의 거리인 홉 수는 시스템 변수가 될 수 있다.

<표 1>은 Adaptive_dIDS의 처리 과정을 기술한 Pseudocode이다. 초기화 과정으로서, 지정된 홉 카운트 내의 노드를 이웃 노드 리스트(S_{nbr})에 추가한다. 각 노드는 자신의 degree 값

〈표 1〉 Adaptive_dIDS 처리 과정

Adaptive_dIDS 과정	처리 과정
초기화(Initialization)	1. S_{nbr} ← {v: v lies within the specified hop count} 2. Compute degree (degree is the number of one-hop neighbor) and broadcast it to nodes in S_{nbr}
주 처리 (Main Processing)	1. Send_voting_msg(NodeID, Node_WithHighestDegree) 2. While(Not_Timeout) 3. If(Recevied_voting_msg()) 4. vote_num1++ 5. If(vote_num1>threshold1) 6. Send_challenge_msg(NodeID, Nodes in S_{nbr}) 7. While(Not_Timeout) 8. If(Recevied_challenge_msg()) 9. vote_num2++ 10. If(vote_num2<threshold2) 11. Active intrusion detection 12. Increase threshold1&threshold2 13. Else 14. Reduce threshold1&threshold2 15. Completed voting 16. Else 17. Reduce threshold1&threshold2 18. Completed voting

인 한 홉 이웃 노드 수를 계산하여 이웃 노드 리스트의 노드들에게 전송한다. 주 처리 과정으로서 (1) 각 노드는 자신의 이웃 노드 중에서 가장 큰 degree값을 갖고 있는 노드를 IDS로 선출하는 의미에서 해당 노드에게 선출 메시지를 전송(Send_voting_msg)한다. 각 노드는 임의 시간 동안 기다리면서 자신이 선출 받은 메시지를 카운팅한다(vote_num1). (5) 선출 받은 수가 자신의 첫 번째 임계치인 threshold1보다 크면 이웃 노드 중에 많이 선출받은 노드 수를 알아보기 위해 다시 이웃 노드들에게 challenge 메시지를 전송한다. (10) 수신한 challenge 메시지의 수가 자신의 두 번째 임계치인 threshold2보다 작으면, 즉 주변에 어느정도 선출 받은 노드가 적으면 자신이 IDS가 되고, 자신의 두 임계치 값을 증가시켜 다음 라운드에 자신이 IDS가 되는 가능성을 줄이고 주 처리과정을 마친다. (13) 그러나 주변에도 많이 선출받은 노드가 많거나, (16) 아예 자신이 선출 받은 수가 threshold1보다 작다면 다음 라운드에서 자신이 IDS가 되는 가능성을 증가시키기 위해 임계치를 감소시키고, 주 처리과정을 마친다.

3. 에너지 효율적인 분산 침입탐지 구조

3.1 가정한 센서 네트워크

본 논문에서 제안하는 분산 침입탐지 구조에서는 우선 순위 높은 메시지를 우선적으로 전송 및 포워딩할 수 있고 가정한다. 그러나, 노드간 시간 동기화는 가정하지 않는다.[21] 에서 언급된 것처럼 mHEED 알고리즘은 시간 동기화에 상관없이 수행가능하며 효율성을 극대화 하기 위해 RBS[22]와 같은 기법이 적용될 수도 있다. 또한 제안하는 분산 침입탐지 구조를 위해 센서 네트워크에서는 다음과 같은 구성 노드를 갖는다. (그림 2)는 이러한 구성요소로 두 라운드 동안 운영되는 제안한 분산 탐지 구조의 예시를 나타내고, (그림 3)은 각 노드에서 분산 탐지를 위해 전송되는 메시지 흐름 및 운용되는 방식을 도식화 한 것으로 클러스터 헤드인 CH와 dIDS가 동일한 노드에 위치하는 것으로 가정하였다.

- cIDS(Centralized Intrusion Detection System): 네트워크에 하나 존재하는 중앙 IDS 시스템으로 초기 선출된 dIDS들에게 "탐지를 위한 에이전트(MA)" 파견 역할과 MA들이 수집한 모니터링 결과를 취합하여 최종 침입탐지에 대한 판단을 내리는 시스템
- dIDS(Distributed Intrusion Detection System): mHEED에 의해 선출된 침입탐지 시스템들로 cIDS 혹은 이전 dIDS들로부터 파견된 MA를 수행하여 특정 기간동안 자신의 구역에 대해 침입탐지를 위한 모니터링 기능을 수행한다. 주기적으로 모니터링 결과를 일반 센싱 메시지에 첨부(piggybacking)하여 전송하며, 긴급 정보인 경우 전송 우선순위가 높은 메시지를 생성하여 cIDS에게 직접 전송한다. dIDS 수행 기간이 완료되면 mHEED에 의해 선출된 이웃의 다음 dIDS에게 자신이 모니터링한 결과의 요약과 MA를 파견한다.
- MA(Monitoring Agent): 탐지를 위한 에이전트 프로그램으로 분산 위치한 dIDS들에서 수행된다. 오용탐지를 위해 이미 발견된 침입에 대한 모델을 포함하고 있으며, 이상 탐지를 위한 정상 트래픽 패턴 구성을 위한 정보를 수집한다.
- SINK: 일반 센싱 정보를 수집하여 관리하는 시스템으

로 네트워크에 하나 혹은 여러 개 존재 가능하다. 일반 센싱 정보에 첨부된 침입탐지 관련 모니터링 결과는 자신이 cIDS가 아니라면 해당 cIDS에게 특정 기간동안 모아서 전달한다. cIDS와 SINK는 같은 시스템일 수도 있고 아닐 수도 있다.

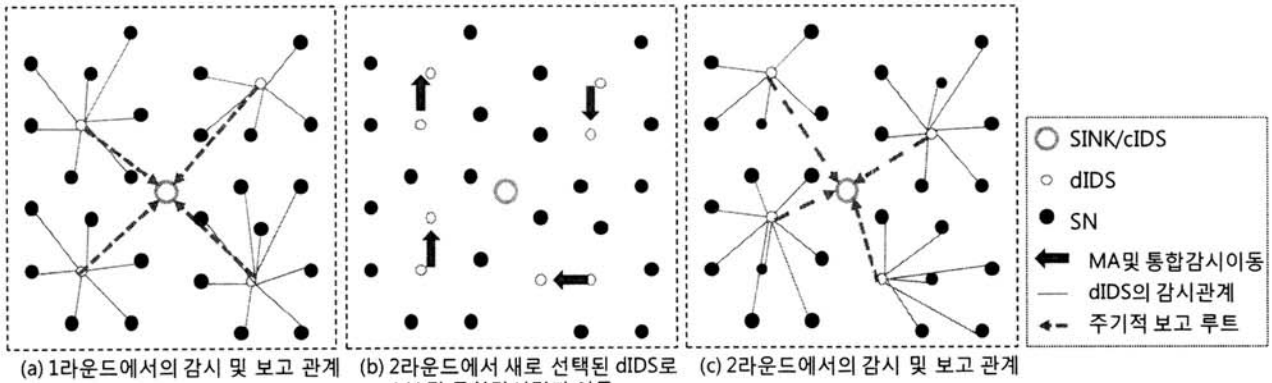
- SN: 일반적으로 센싱 기능을 수행하는 노드로서 mHEED에 의해 dIDS로 결정되면 일정 기간동안 센싱기능과 dIDS 기능을 함께 수행한다.

3.2 dIDS 선출을 위한 mHEED 알고리즘

본 mHEED(modified HEED) 알고리즘은 에너지 효율적인 클러스링 알고리즘인 HEED 알고리즘을 기반으로 하고 있으나 dIDS 선출 목적으로 사용될 수 있도록 전송 메시지를 줄여 그 알고리즘을 수정하였다. 물론 클러스터링 목적을 위해 HEED로서 클러스터 헤드 선출 및 해당 클러스터가 dIDS 수행을 하여도 무관하나 본 논문의 범위가 클러스터링을 포함하고 있지 않으므로 이에 관해서는 더 이상 언급하지 않는다.

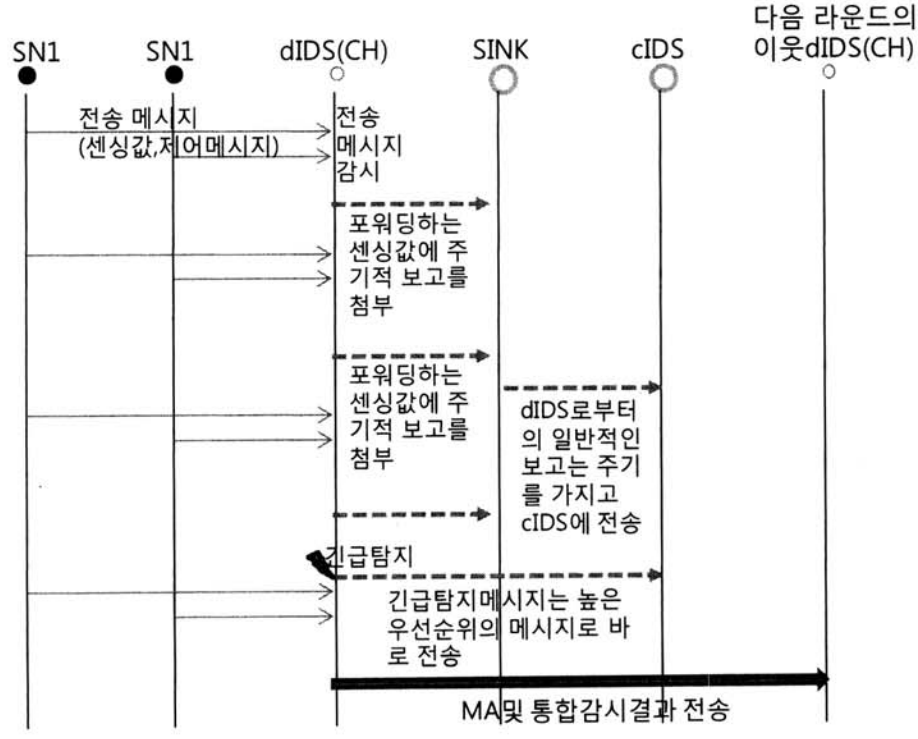
각 노드는 주기적으로 Tsel(dIDS 선출 시간)을 가지며 새로운 dIDS를 선출한다. 단 이 시간에도 일반 센싱 및 센싱결과 전송의 일반 네트워크 운영 기능을 함께 한다. Tsel 시간이 되면, 다음과 같은 알고리즘을 수행하여 dIDS를 선출한다. 선출기준은 남은 에너지 잔량의 최대화, 통신 비용의 최소화를 추구한다.

- 분산된 에너지 소비 제고를 위한 함수로서 각 노드는



(a) 1라운드에서의 감시 및 보고 관계 (b) 2라운드에서 새로 선택된 dIDS로 MA및 통합감시결과 이동 (c) 2라운드에서의 감시 및 보고 관계

(그림 2) 분산 침입탐지 구조를 갖은 센서 네트워크 예시



(그림 3) 제안하는 분산 침입탐지 구조의 운용 예

자신이 dIDS가 될 확률을 다음의 식으로 계산한다:
$dIDS_{prob} = C_{prob} * E_{residua}/E_{max}$
- C_{prob}: 초기 dIDS 공지(announcement)의 수를 제한하기 위해 사용되는 값
- $E_{residual}$: 해당 노드가 현재 남아 있는 에너지 잔량
- E_{max}: 최대 충전된(changed) 에너지

• 통신 비용의 최소화를 위한 함수로서 노드의 전송 파워 레벨 조절 능력 유무에 따라 다음 중 하나를 선택하여 통신 비용 계산 $dIDS_{cost}$ = AMRP 혹은 node degree
- 노드가 다른 노드 각각과 통신을 위한 최소 전송 파워 레벨을 조절할 수 있는 경우: AMRP(Average Minumum Reachability Power) $=\sum_{i=1\sim M} MinPwr_i / M$
  · $MinPwr_i$: 이웃 노드 i와의 통신을 위한 최소 파워 레벨
  · M: 전송 레인지(노드의 최대 파워 레벨을 가지고 통신 가능한 범위) 안의 이웃 노드 수
- 전송 파워 레벨 조절 능력 없는 경우: $dIDS_{cost}$ = 1/node degree 즉, 이웃 노드 수의 역수

<표 2>는 Proposed_dIDS의 처리 과정을 기술한 Pseudocode이다. 초기화 과정으로서, 전송범위 내의 노드를 이웃 노드 리스트(S_{nbr})에 추가한다. 각 노드는 자신의 AMRP 혹은 degree을 이용하여 비용을 계산하고 이웃 노드들에게 전송한다. 자신이 dIDS가 될 확률값인 $dIDS_{prob}$를 남은 에너지에 따라 초기화 한다. 주 처리 과정은 $dIDS_{prob}$가 1이 될 때까지 반복하는데, 1) 가능한 dIDS 리스트인 S_{dIDS}에 노드가 존재하면 그 노드들 중에 최소 비용을 갖는 노드를 자신의 dIDS로 삼는다. 4) 그 때 최소 비용의 노드가 자신이며 $dIDS_{prob}$가 1이 되었다면, 이웃들에게 final 상태값을 알리며 종료한다. 그러나 아직 $dIDS_{prob}$가 1이 되지 않았다면, 이웃들에게 자신도 dIDS가 될 가능성이 있음을 tentative 상태값을 가지고 알리게 된다. 9) $dIDS_{prob}$가 1이 되었다면 마찬가지로 이웃들에게 final 상태값을 알리며 종료한다. 13) 자신의 랜덤값이 현재 자신의 $dIDS_{prob}$보다 작게되면 자신이 dIDS가 될 가능성이 있음을 tentative 상태값으로 알리게 된다. 마지막으로 자신의 확률값을 2배로 증가시켜 주 처리를 반복하게 된다. 종료 과정에서는 주처리에서 종료되지 않은 노드들을 처리하며, S_{dIDS}의 최소 비용의 노드를 자신의 dIDS로 삼거나, S_{dIDS}에 노드가 없는 경우에 자신이 dIDS가 되게 된다.

원래 클러스터링 알고리즘 HEED에서 수정된 부분은, HEED의 종료(Finalization) 과정에서 Final 상태가 되면

〈표 2〉 dIDS 선출을 위한 mHEED 과정

mHEED 과정	처리 과정
초기화(Initialization)	1. $S_{nbr} \leftarrow$ {v: v lies within communication range} 2. Compute $dIDS_{cost}$ and broadcast it to nodes in S_{nbr} 3. $dIDS_{prob} \leftarrow \max(C_{prob} * E_{residua}/E_{max}, p_{min})$ 4. is_final = FALSE
주 처리 (Main Processing)	Repeat 1. If(($S_{dIDS} \leftarrow$ {v: v is a dIDS}) $\neq \Phi$) 2. my_dIDS ← least_cost(S_{dIDS}) 3. If(my_dIDS = NodeID) 4. If($dIDS_{prob}$ = 1) 5. dIDS_Announce_Msg(NodeID,final,cost) 6. Is_final ← TRUE 7. Else 8. dIDS_Announce_Msg(NodeID,tentative,cost) 9. ElseIf($dIDS_{prob}$ = 1) 10. dIDS_Announce_Msg(NodeID,final,cost) 11. My_dIDS ← NodeID 12. Is_final ← TRUE 13. ElseIf (Random(0,1) ≤ $dIDS_{prob}$) 14. dIDS_Announce_Msg(NodeID,tentative,cost) 15. $dIDS_{previous} \leftarrow dIDS_{prob}$ 16. $dIDS_{prob} \leftarrow \min(dIDS_{prob} * 2, 1)$ Until $dIDS_{previous}$ = 1
종료(Finalization)	1. If(is_final = FALSE) 2. If(($S_{dIDS} \leftarrow$ {v: v is a final dIDS}) $\neq \Phi$) 3. my_dIDS ← least_cost(S_{dIDS}) 4. Else 5. dIDS_Announce_Msg(NodeID,final,cost) 6. my_dIDS ← NodeID

(is_final = TRUE) 또 다시 dIDS_Announce_Msg(NodeID, final, cost)를 통해 자신의 Final 상태를 알리는데, 이 부분은 이미 주 처리(Main Processing) 과정 부분에서 Final 상태가 되면서 자신의 상태를 알려주므로 이를 삭제하여 상태에 대한 중복 메시지 송신 부분을 제거 하였고, 이로 인해 자신의 dIDS를 선정하지 않은 경우에 대해 Final 상태가 되면 자신이 dIDS가 되도록 보완하였다. 음영 표시가 된 부분이 HEED에서 수정된 부분이다.

3.3 MA 구현에 대한 고찰

본 논문의 침입탐지 구조에서는 분산 탐지를 위해 선출된 dIDS에 탐지 기능을 수행하는 에이전트인 MA(Monitoring Agent)가 파견된다. 이러한 이동 에이전트 프로그램은 최근 연구되고 있는 "센서 네트워크에서의 코드 재분배 기술[11-13]"을 통해, 구현에 활용 될 수 있다.

관련 코드 재분배 방법은 크게 "전체 수정된 바이너리 코드를 전송하는 방법"과 연속적인 두 버전의 코드를 비교하여 그 차이를 요약한 스크립트만 전송하는 "Diff 기반 코드 재분배 방법"으로 나눌 수 있고, 센서 네트워크에는 에너지 효율성이 중요한 요소이므로 후자에 관련된 연구가 활발히 진행되고 있다. 대표적인 전자의 방법으로서 TinyOS의 코드 분배 기술로서 XNP에 의해 구현된 방법이 있다. 또한 다른 레벨의 코드 재분배 방법으로서 소형 가상 머신에 의한 방법과 동적 링커를 활용하는 방법이 있는데, 이러한 방법은 바이너리 인스트럭션 대신, 가상 머신 프리미티브와 같이 상위 레벨로 표현된 코드를 전송함으로써 전송되는 코드 수를 줄이는 방법도 있으나 수행시간에 큰 오버헤드를 야기시킬 수 있다[11]. 마지막으로 코드 재분배 수행은 보안상 높은 안전성을 요구하므로 센서 네트워크에서의 안전한 코드 분배 방법에 대한 연구도 진행되고 있다[12,13].

본 논문에서의 침입탐지 에이전트는 분배의 안전성을 고려하며, 상황에 따라 위에서 언급한 대표적인 두 가지 분류 방법, 즉 "바이너리 코드 분배 방법"과 "Diff 기반 코드 분배 방법"을 혼용하여 적용하는 것이 적합하다. 즉, 초기에 침입탐지 시스템이 수행되어 각 노드에서 처음으로 dIDS로 선출되는 경우가 많은 경우, 또는 새로운 공격 모델이 추가되는 경우에는 "바이너리 코드 전송 방법"이 적합하며, 침입탐지 시스템이 어느정도 안정적으로 수행되어 dIDS로 재선출되는 확률이 높아지는 경우에는 "Diff 기반 코드 분배 방법"을 통해 코드 분배의 양을 최소화 할 수 있다.

4. 시뮬레이션 및 결과 분석

본 장에서는 본 논문에서 제안하고 있는 "mHEED를 이용한 분산 IDS 구조 (Proposed_dIDS라 칭함)"의 성능을 확인하기 위하여 센서 네트워크에서의 분산 IDS 구조 기존 연구 중에서 본 논문과 가정하고 있는 센서 네트워크 구조가 가장 비슷하며 최신 연구 결과 중의 하나인 "적응적인 침입탐지 구조[10]"(Adaptive_dIDS라 칭함)와 비교하여 다양한 시뮬레이션을 수행하고 그 결과를 분석한다.

4.1 시뮬레이션 환경

본 논문에서의 센서 노드의 분포는 두 가지 분포모델을 가정하였다. 첫 번째는 기본적으로 200개의 노드가 랜덤하게 분포되어 있는 랜덤분포이고, 두 번째는 각 노드 간 거리가 동일한 환경을 가정하여 196개의 노드가 분포되어 있는 그리드분포이다.

Adaptive_dIDS와 비교를 통해 Proposed_dIDS의 성능 확인하기 위해 기존 연구에서 사용된 에너지 소비 모델을 참고하여 이용하였다[21,23]. <표 3>의 에너지 소비 모델 매개변수를 가지고, 분산 침입탐지 역할의 dIDS노드를 선출하는 과정에서 전송되는 제어메시지의 송수신 시 소모 에너지를 계산하는 식은 (1),(2)와 같다.

$$E_{Tx} = LE_{elec} + L\varepsilon_{fs}d^{2} \quad (1)$$

$$E_{Rx} = LE_{elec} \quad (2)$$

일반 센싱 데이터의 전송 에너지 소모를 계산하기 위해서 한 라운드 동안에 클러스터 헤드와 비 클러스터 헤드 노드 사이에서 데이터를 주고 받으면서 소모되는 에너지는 식은 (3)과 같다. 본 시뮬레이션에서는 선출된 dIDS노드가 클러스터 헤드 역할도 수행한다고 가정한다.

$$E_{CH} = (n/k-1)LE_{elec} + (n/k)LE_{DA} + LE_{elec} + L\varepsilon_{fs}d^{2BS}$$
$$E_{nonCH} = LE_{elec} + L\varepsilon_{fs}d^{2CH} \quad (3)$$

본 논문의 시뮬레이션에서 사용된 기본적인 시뮬레이션 매개 변수의 값은 <표 4>에 나타내었다. 제어메시지의 길이는 TinyOS 메시지를 가정하여 기본헤더 12bytes와 메시지 종류나 비용에 대한 값을 전달할 것을 고려하여 15bytes로 결정하였고, 나머지 값들도 기존 논문[21,23]에서 사용된 값을 참조하여 결정하였다. Proposed_dIDS의 비용 계산시 degree의 역수를 사용하였다.

4.2 결과 및 분석

제안한 Proposed_dIDS의 에너지 효율성 및 오버헤드 측면, 탐지가능성과 그 성능 측면의 시뮬레이션 결과를 구했다. 첫째 전체 네트워크의 에너지 소비의 효율성 측면을 분석하기 위하여, 라운드 증가 시 1) dIDS 수, 2) 에너지 총

<표 3> 에너지 소비 모델 및 Proposed_dIDS 처리 시 사용되는 매개 변수

변수	값	변수	값
L	메시지 길이	d^{BS}	CH와 BS 간 거리
E_{elec}	회로 에너지 소모	k	클러스터 헤드 수
E_{DA}	Aggregation 에너지	n	총 노드 수
ε_{fs}	자유공간 손실	C_{prob}	dIDS 선택 확률 조절 값
d^{CH}	노드와 CH 간 거리	p_{min}	주처리 수행시, 반복 수 제한 값

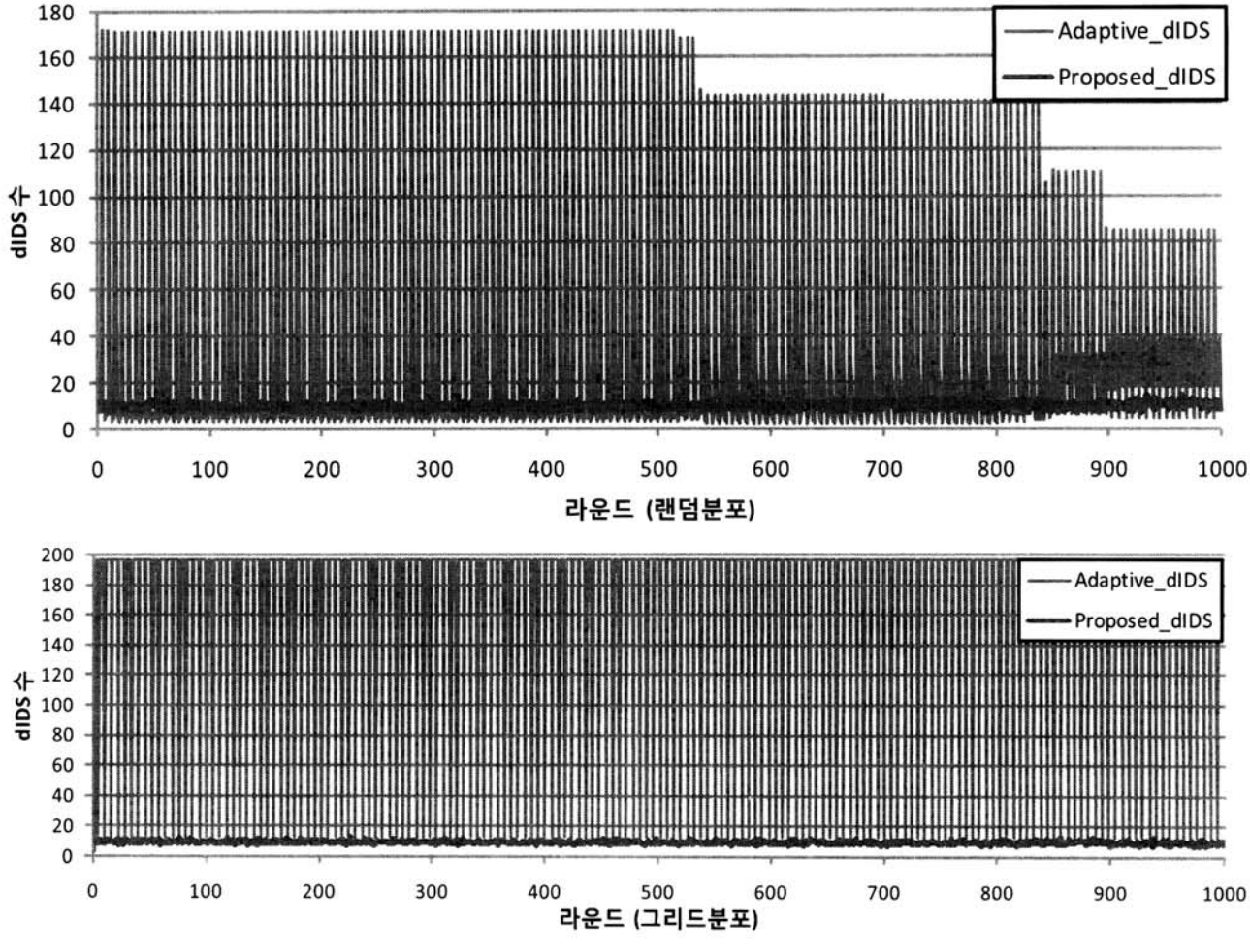
〈표 4〉 시뮬레이션 매개 변수 값

변수	값	변수	값
노드 수	200	제어메시지 길이	15bytes
시뮬레이션 공간	100m x 100m	C_{prob}	0.05
라디오 전송 범위	30m	p_{min}	0.0005
SINK 위치	(50,50)m	ε_{fs}	10pJ/bit/m^2
초기 에너지	0.5J	E_{elec}	50nJ/bit
한 라운드 동안 각 센서에서 전송되는 센싱데이터메시지 총 길이 합	8000bit	E_{DA}	50nJ/bit/report

잔량, 3) 에너지 분포도, 4) 에너지 0이 되는 노드 수와, 센서 노드의 전송 범위 변화 시 5) dDIDS 수를 비교하였다. 둘째 오버헤트 측면을 분석하기 위하여 1) 총 노드 수 변화 시 오버헤드(전송되는 제어메시지 수), 2) 센서 노드의 전송 범위 변화 시 오버헤드를 비교하였다. 셋째 탐지 가능성과 그 성능 측면을 분석하고자 dIDS로 관리되지 않는 노드의 수를 비교하였다.

■ *에너지 효율성 측면*

(그림 4)는 각 라운드의 dIDS의 수를 보여주는 그래프로 랜덤분포와 그리드분포에서 총 1000 라운드를 수행하였다. 그림 에서 보이는 것처럼 분포에 상관없이 Adaptive_dIDS는 dIDS의 수가 라운드가 증가함에 따라 주기적으로 높아졌다 낮아졌다 하는 것을 확인할 수 있다. 이 알고리즘은 각 노드의 서로 다른 임계값을 기반으로 각 라운드에서 노드의 IDS의 여부를 결정하는 방식이다. 만일 이번 라운드에 IDS가 되면 임계값을 증가시켜 다음 라운드에 IDS가 되는 확률을 줄이고, 이번 라운드에 IDS가 되지 않으면 임계값을 감소시켜 다음 라운드에 IDS가 되는 확률을 높이는 방식이다. 그렇지만 이 방법은 degree에 따라 투표를 하여 자신의 임계값과 비교한 뒤 IDS를 결정하기 때문에 처음에는 degree 값이 높은 일부 노드가 IDS 활동을 하다가 그 노드마저 임계값이 높아지면서, IDS 활동을 하는 노드가 하나도 생기지 않게 되는 라운드가 발생한다. 이와는 반대로 degree 값이 낮은 노드들이 한동안 IDS로 선택되어지지 않다가 자신의 임계값이 낮아지면서 많은 노드가 한꺼번에 IDS가 되려고 하여 급격하게 IDS 수가 증가하는 현상이 나타나게 되고, 이러한 현상은 위 그래프에서 보이는 것과 같이 주기적으로 찾아온다. 반면 Proposed_dIDS 방법은 dIDS의 수가



(그림 4) 각 라운드의 dIDS수

라운드가 증가해도 일정하게 유지되는 것을 확인할 수 있다. 이는 남은 에너지와 degree를 둘다 고려하고, dIDS 선택 확률 조절 값 C_{prob}을 통해 자신이 IDS가 될 확률을 결정하기 때문에 각 라운드 마다 일정한 수의 IDS가 선출된다.

(그림 5)는 각 라운드의 에너지 총 잔량을 보여주는 그래프로 랜덤분포와 그리드분포에서 총 3000 라운드를 수행하였다. Adaptive_dIDS방식은 degree만을 고려하여 투표메시지를 보내는 방식이라 에너지를 고려하지 않기 때문에 에너지를 고려하여 IDS가 될 확률을 결정하는 본 논문의 Proposed_dIDS방식이 에너지 잔량 측면에서 우수한 결과를 나타내었다.

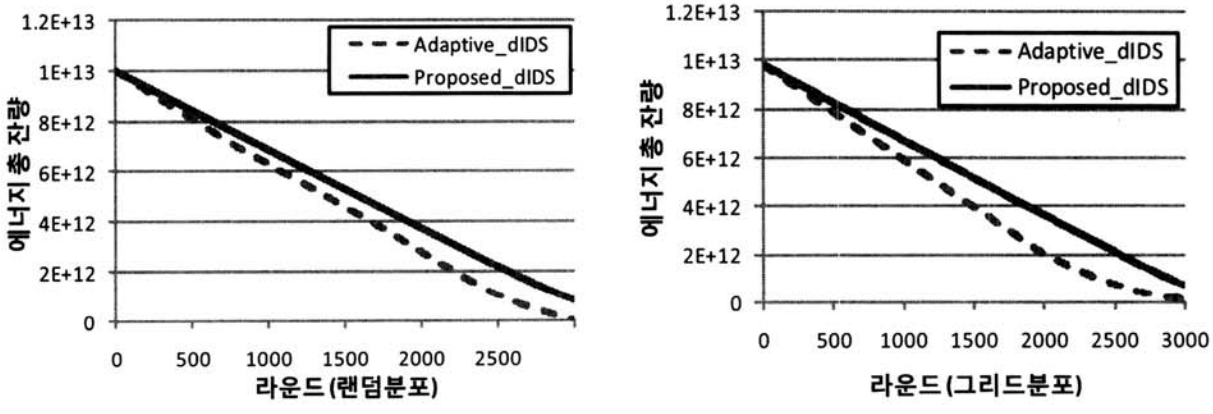
(그림 6)은 각 라운드의 에너지 분포도를 보여주는 그래프로서 분포도 계산에서 많이 사용하는 통계식의 하나인 Entropy 계산식(식 (4))을 이용하여 값을 구하였다. Adaptive_dIDS 방식은 에너지를 고려하지 않고 dIDS를 선출하여 degree가 높은 노드일수록 IDS가 될 확률이 높아지고 에너지도 빠르게 감소한다. 따라서 특정 노드의 에너지가 빠르게 감소하기 때문에 에너지 분포가 고르지 못한 결과를 확인할 수 있다. 반면 Proposed_dIDS방식은 에너지를 고려하여 에너지 잔량이 더 많이 남아있는 노드일수록 IDS가 될 확률이 높아지기 때문에 Adaptive_dIDS방식에 비해 에너지가 고르게 소비되는 결과를 확인할 수 있었다.

$$H = -\sum_{i=1}^{n} P_i \log_2 P_i, \quad P_i = E_{residual} / E_{\max} \qquad (4)$$

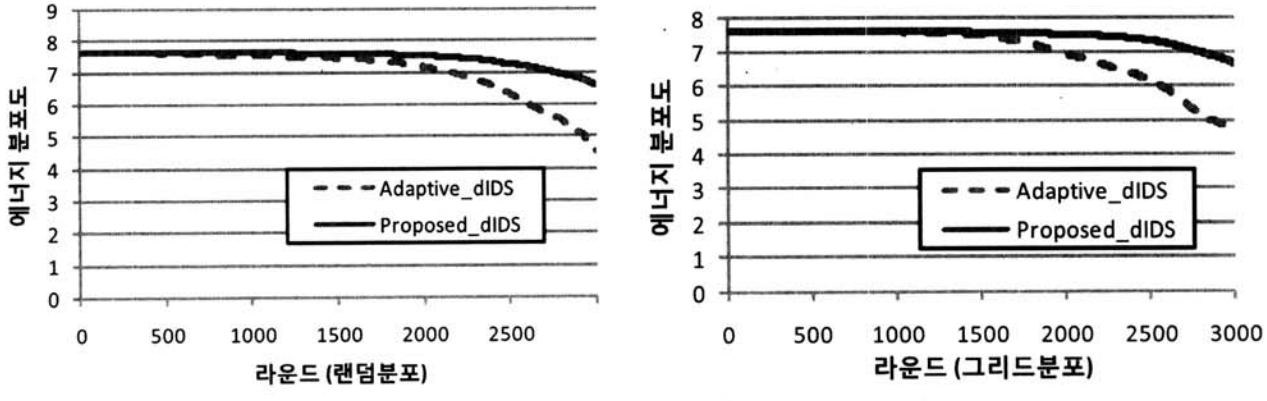
(그림 7)은 각 라운드에서 에너지가 0이 되는 노드 수를 보여주는 그래프로 총 3500번의 라운드를 수행하였다. Adaptive_dIDS방식은 에너지를 고려하지 않고 dIDS를 선출하여 degree가 높은 노드일수록 IDS가 될 확률이 높아져 degree가 높은 특정 노드의 에너지가 빠르게 감소한다. 따라서 그래프를 통해 에너지가 0이 되는 노드가 처음 생기는 라운드를 비교해 보았을 때, 그리드분포 구조에서는 Proposed_dIDS와 약 100라운드 정도의 차이를 보이지만, 랜덤분포 구조에서 약 1000라운드 정도의 차이를 보이는 것을 확인할 수 있다. 또한 에너지가 0이 되는 수도 각 라운드 별로 분석했을 때 Proposed_dIDS의 방식이 더 적음을 확인할 수 있다.

4.3 전송 범위에 따른 dIDS 수

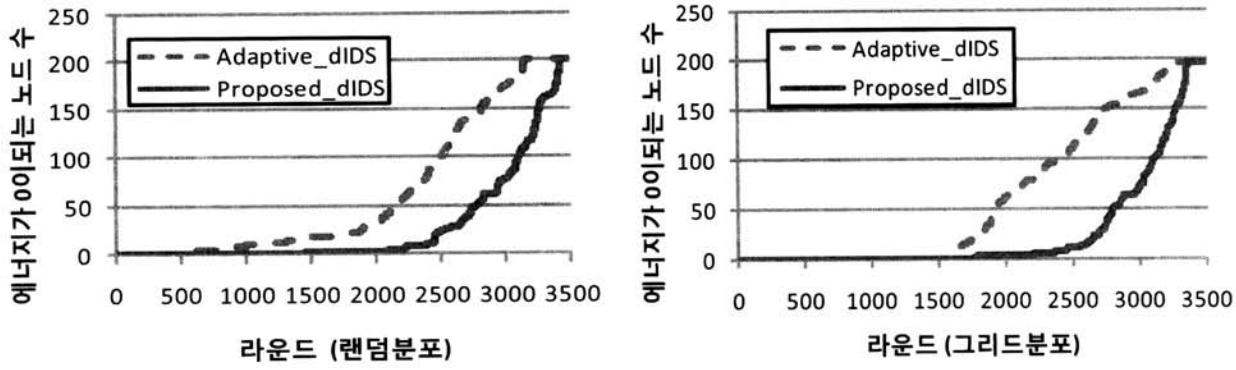
(그림 8)은 전송범위를 변화시켜 각 노드의 이웃 노드 수를 변화시켜 보았을 때, 평균 dIDS 수의 변화를 나타내는 그래프로서 총 1000번 라운드를 수행하여 얻은 그래프이다. Proposed_dIDS방식은 각 노드의 전송범위가 넓어 질수록 커버할 수 있는 노드 수가 많아지기 때문에 평균 dIDS의 수가 더 적어짐을 확인할 수 있다. 반면 Adaptivfe_dIDS방식은 전송범위가 커지게 되면 degree가 큰 소수의 노드가 많은 투표를 받아 dIDS 역할을 수행하다가 그 노드의 임계값이 커지게 되어 IDS 활동을 하는 노드가 하나도 생기지 않게 되면 degree가 낮은 많은 노드들이 한꺼번에 dIDS가



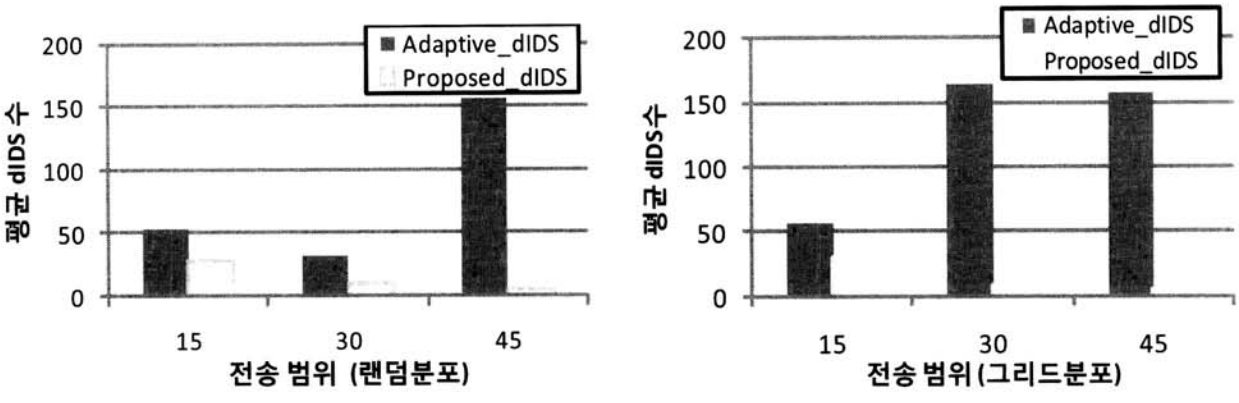
(그림 5) 각 라운드의 에너지 총 잔량



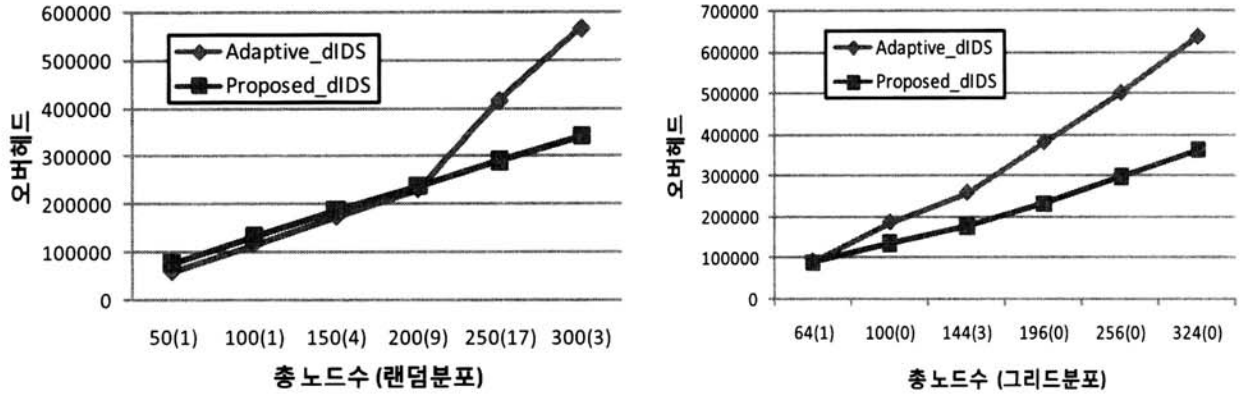
(그림 6) 각 라운드의 에너지 분포도



(그림 7) 각 라운드의 에너지가 0이되는 노드 수



(그림 8) 전송범위에 따른 평균 dIDS 수



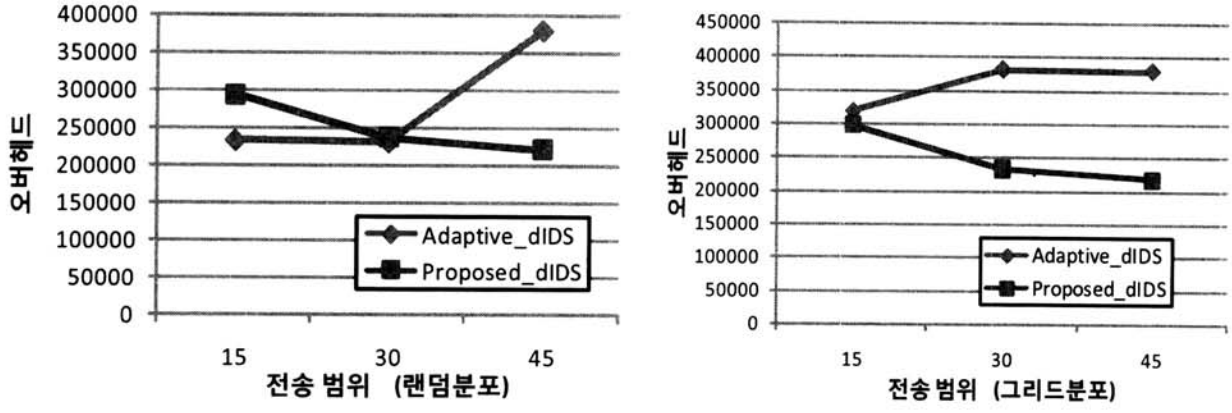
(그림 9) 총 노드수에 따른 오버헤드

되는 정도가 더욱 커져 급격하게 IDS 수가 증가하는 현상으로 인해 평균 dIDS 수가 급증하는 결과를 나타내었다.

■ *오버헤드 측면*

(그림 9)는 노드수별 오버헤드(전송되는 제어메시지수)를 보여주는 그래프로 라운드는 총 1000번 수행하였다. 그래프에서 총 노드수의 괄호안의 값은 Adaptive_dIDS를 사용하였을 때 1000 라운드 수행 후 에너지가 0이 된 노드 수이다. Proposed_dIDS는 1000 라운드 수행 후에도 에너지 0이 되는 노드가 나타나지 않아 따로 표기하지 않았다. 랜덤분포의 경우, 노드 수가 작을 때에는 Proposed_dIDS의 경우가 Adaptive_dIDS방식보다 오버헤드가 약간 많았지만, 노드 수가 많아 질수록 Adaptive_dIDS 방식의 오버헤드가 현저히 증가한 결과를 나타내었다. 반면 그리드분포에서는 노드수에 상관없이 Adaptive_dIDS 방식의 오버헤드가 항상 큰 것을 확인할 수 있다.

(그림 10)은 전송범위에 따른 오버헤드를 보여주는 그래프로 총 1000 라운드를 수행하였다. Proposed_dIDS의 경우에는 전송범위가 커질수록 오버헤드가 줄어들었다. 오버헤드는 dIDS의 수의 변화에 따라 결정되는데, Adaptive_dIDS의 경우에는 (그림 7)의 설명에서처럼 전송범위 증가시 오히려 dIDS 수가 증가하는 현상 때문에 마찬가지로 오버헤



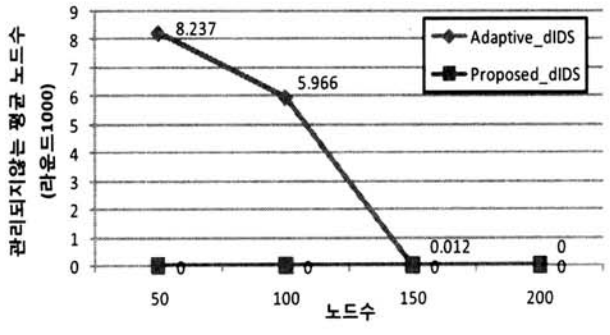
(그림 10) 전송범위에 따른 오버헤드

드도 증가되는 결과를 나타내었다.

■ *탐지 가능성과 그 성능 측면*

본 논문에서는 탐지 모델에 초점을 두지 않았기 때문에 dIDS 노드가 탐지를 수행하는 능력은 제안한 Proposed_dIDS와 Adaptive_dIDS가 동일하다고 가정한다. 단, 각 라운드에서 선택된 dIDS들이 전체 네트워크의 센서 노드를 감시할 수 있도록 잘 선택되어져야 공격 발생 시 탐지 가능성과 그 성능 높일 수 있다. 이 척도로서 dIDS에 의해 감시되지 않는 노드 수를 비교하였다.

(그림 11)은 노드 수가 증가함에 따라 관리되지 않는 노드, 즉 자신을 맡아 줄 dIDS가 없는 평균 노드 수를 보여주는 그래프로 랜덤분포 구조에서 실행하였으며, 총 1000 라운드를 수행하여 평균값을 구하였다. 제안한 Proposed_dIDS에서는 자신을 포함하여 dIDS가 될 가능성이 있는 노드, 즉 $dIDS_{prob}$가 1이 된 노드 중에서 최소의 비용(이웃 노드 수인 degree의 역 수)을 갖는 노드를 dIDS로 선정하므로 관리되지 않는 노드는 발생하지 않았다. 이에 반해 Adaptive_dIDS의 방식의 경우, degree 값에 따라 dIDS로서 투표가 되어 높은 degree를 갖는 노드만 dIDS가 되다가 몇 라운드 지나면 이들 노드 조차 자신의 임계값이 커지게 되어 dIDS가 극소로 줄어들고, 낮은 degree 값을 갖는 노드의 임계값이 높아져 많은 노드가 dIDS가 되는 등, (그림 4)에서처럼 dIDS 수의 급증, 급감 상황이 반복이 되고, 이로 인해 관리되지 않는 노드 수가 많은 라운드가 발생하게 된다. 결과적으로 노드의 수가 50개 일 때는 전체 노드 중 약 16%의 노드가 관리되지 않고, 노드의 수가 100개 일 때는 전체 노드 중 약 6%의 노드가 관리되지 않는 결과를 나타내었다. 따라서 관리되지 않는 노드를 이용한 공격은 탐지할 수 없기 때문에 상당히 공격에 취약하다고 볼 수 있다. 더욱이 이 결과는 Adaptive_dIDS의 시스템 매개변수인 홉수가 2 이상의 값을 가지는 경우에는, 관리받는 노드가 2, 3홉 멀리 떨어져 있을 수 있게 되어 그만큼 탐지 지연이 발생할 수 있게 된다.



(그림 11) 총 노드수당 관리되지 않는 평균 노드 수

5. 결 론

본 논문에서는 평평한 센서 네트워크 특성을 고려하여 에너지 소비가 효율적이며 많은 수의 센서 노드를 감시할 수 있는 분산 침입탐지 구조를 제안하였다. 각 라운드 마다 에너지 잔량 및 이웃 노드 수에 따라 분산 침입탐지 노드들을 결정하고, 선택된 탐지노드는 감시 결과 및 탐지 모델링에 필요한 기반 자료들을 다음 라운드의 새로운 탐지노드에게 탐지 코드와 함께 이동 에이전트를 통해 전달한다. 감시에 대한 주기적 보고는 센싱 정보에 첨부되어 SINK를 통해 중앙 탐지노드에게 전달하고, 긴급 탐지 결과는 높은 우선순위의 메시지를 통해 바로 전달하게 한다. 에너지 효율성, 오버헤드, 탐지 가능성 및 그 성능 측면에서 기존 연구 결과의 하나인 적응적 침입탐지 구조와 비교 시뮬레이션을 수행하였으며, 본 논문에서 제안한 메커니즘이 향상된 성능을 보임을 입증할 수 있었다.

참 고 문 헌

[1] P. Brutch and C. Ko, "Challenges in intrusion detection for wireless ad-hoc networks," Proc. of the 2003 Symposium on Applications and the Internet Workshops (SAINT'03 Workshops), pp.368-373, 2003.

[2] A. A. Strikos, "A full approach for intrusion detection in wireless sensor networks," School of Information and Communication Technology, KTH, March, 2007.

[3] S. Doumit and D.P. Agrawal, "Self-organized criticality &

stochastic learning based intrusion detection system for wireless sensor network," MILCOM 2003 - IEEE Military Communications Conference, Vol.22, No.1, pp.609-614, 2003.

[4] C. Su, K. Chang, Y. Kuo, and M. Horng, "The new intrusion prevention and detection approaches for clustering-based sensor networks," 2005 IEEE Wireless Communications and Networking Conference (WCNC 2005), March, 2005.

[5] S. Zhu, S. Setia, and S. Jajodia, "LEAP: Efficient Security Mechanisms for Large-Scale Distributed Sensor Networks," Proc. of the 10th ACM conference on Computer and communications security, 2003.

[6] A. Agah, S. Das, K. Basu, and M. Asadi, "Intrusion detection in sensor networks: A non-cooperative game approach," 3rd IEEE International Symposium on Network Computing and Applications (NCA 2004), pp.343-346, August, 2004.

[7] A. da Silva, M. Martins, B. Rocha, A. Loureiro, L. Ruiz, and H. Wong, "Decentralized intrusion detection in wireless sensor networks," Proc. of the 1st ACM international workshop on Quality of service & security in wireless and mobile networks, 2005.

[8] P. Kannadiga and M. Zulkernine, "DIDMA: a distributed intrusion detection system using mobile agents," First ACIS International Workshop on Self Assembling Wireless Networks (SNPD/SAWN 2005), pp.238-245, May, 2005.

[9] Ketel, M., "Applying the Mobile Agent Paradigm to Distributed Intrusion Detection in Wireless Sensor networks," 40th Southeastern Symposium on System Theory (SSST 2008), pp.74-78, March, 2008.

[10] P. Techateerawat and A. Jennings, "Adaptive Intrusion Detection in Wireless Sensor Networks," International Conference on Intelligent Pervasive Computing, 2007.

[11] Youtao Zhang, Jun Yang, Weijia Li. "Towards Energy-Efficient Code Dissemination in Wireless. Sensor Networks," International Conference on Information Processing in Sensor Networks (IPSN 2008), April, 2008.

[12] Jing Deng, Richard Han, Shivakant Mishra, "Secure code distribution in dynamically programmable wireless sensor networks," Proc. of the fifth international conference on Information processing in sensor networks 2006, pp.292-300, 2006.

[13] Sangwon Hyun, Peng Ning, An Liu, Wenliang Du, "Seluge: Secure and DoS-Resistant Code Dissemination in Wireless Sensor Networks," Proc. of the 2008 International Conference on Information Processing in Sensor Networks (ipsn 2008), pp.445-456, April, 2008.

[14] Wenyuan Xu, Ke Ma, Trappe, W., Yanyong Zhang, "Jamming sensor networks: attack and defense strategies," Network, IEEE, Vol.20, No.3, pp.41-47, 2006.

[15] D. Raymond et al., "Effects of Denial of Sleep Attacks on Wireless Sensor Network MAC Protocols," Proc. 7th Ann. IEEE Systems, Man, and Cybernetics (SMC) Information Assurance Workshop (IAW), IEEE Press, pp.297-304, 2006.

[16] David R. Raymond, Scott F. Midkiff, "Denial-of-Service in Wireless Sensor Networks: Attacks and Defenses," IEEE Pervasive Computing, Vol.7, No.1, pp.74-81, 2008.

[17] J. Newsome, E. Shi, D. Song, and A. Perrig, "The sybil attack in sensor networks: analysis & defenses," Proc. of the third international symposium on Information processing in sensor networks, ACM Press, pp.259-268. 2004.

[18] J. Deng, R. Han, and S. Mishra. "Countermeasuers against traffic analysis in wireless sensor networks," Technical Report CU-CS-987-04, University of Colorado at Boulder, 2004.

[19] B. Parno, A. Perrig, and V. Gligor, "Distributed detection of node replication attacks in sensor networks," Proc. of IEEE Symposium on Security and Privacy, May, 2005.

[20] H. Chan and A. Perrig, "Security and privacy in sensor networks," IEEE Computer Magazine, pp.103-105, 2003.

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

[22] O. Younis and S. Fahmy, "Distributed Clustering in Ad-Hoc Sensor Networks: A Hybrid, Energy-Efficient Approach," Proc. IEEE INFOCOM, Mar., 2004.

[23] 최경진, 윤명준, 심인보, 이재용, "무선 센서 네트워크에서의 에너지 효율적인 클러스터 헤드 선출 알고리즘", 한국통신학회논문지, Vol.32, No.6, 2007.

김 미 희

e-mail : iceblueee@gmail.com

1997년 이화여자대학교 전자계산학과(학사)

1999년 이화여자대학교 컴퓨터학과(석사)

1999년 (주)인티 연구원

1999년~2003년 한국전자통신연구원 연구원

2007년 이화여자대학교 컴퓨터공학과(박사)

2007년~2009년 이화여자대학교 컴퓨터공학과 전임강사

2009년~현 재 미국 North Carolina State University 컴퓨터공학과 방문연구원

관심분야 : 네트워크 보안, NEMO(NEtwork MObility) 보안, 센서 네트워크 보안, 유비쿼터스 네트워크 보안

김 지 선

e-mail : 272lovelyjs@ewhain.net

2007년 서울여자대학교 정보보호공학과(학사)

2007년~현 재 이화여자대학교 컴퓨터공학과 석사과정

관심분야 : 센서 네트워크 보안, 침입 탐지 및 대응 기술, 센서 네트워크 리프로그래밍 기술

채 기 준

e-mail : kjchae@ewha.ac.kr

1982년 연세대학교 수학과(학사)

1984년 미국 Syracuse University 컴퓨터학과(석사)

1990년 미국 North Carolina State University 컴퓨터공학과(박사)

1990년~1992년 미국 해군사관학교 컴퓨터학과 조교수

1992년~현 재 이화여자대학교 컴퓨터학과 교수

관심분야 : 네트워크 보안, 인터넷/무선통신망/고속통신망 프로토콜 설계 및 성능분석, 센서 네트워크, 유비쿼터스 컴퓨팅