

# Ad-hoc 네트워크를 위한 분산 망 관리 구조 연구

신봉득, 이금모, 이재용  
연세대학교 전기전자공학과  
e-mail : [shinbd@nasla.yonsei.ac.kr](mailto:shinbd@nasla.yonsei.ac.kr)

## A study of Distributed Network Management for Ad-hoc networks

Bong-Deug Shin, Geum-Mo Lee, Jea-Young Lee  
Dept. of Electrical and Electronic Engineering, Yonsei University

### 요 약

이동 Ad-hoc 네트워크는 기존의 인프라스트럭처 네트워크가 없이 이동 노드들이 자율적으로 다수의 홉을 통해 통신하는 네트워크다. 이동 Ad-hoc 네트워크의 이동 노드는 스스로 호스트나 라우터가 되며 능동적인 연결 설정이 가능하고 네트워크의 참여 또는 이탈이 자유롭고 배터리로 운영되어 에너지원의 공급이 제한되고 무선 망 특성에 따른 잦은 환경 변화에 의해 네트워크의 토폴로지가 자주 변하는 특성을 가진다. 이러한 Ad-hoc 네트워크의 특성 때문에 기존의 고정적인 유선 망 관리에 비해 Ad-hoc 네트워크에서의 망 관리는 더 힘들고 어려운 연구분야다. 특히, 기존의 중앙 집중식 망 관리와 같은 구조는 Ad-hoc 네트워크의 특성에 적합하지 않다. 이 논문에서는 Ad-hoc 네트워크의 특성에 맞는 자동적이고 적응력 있는 지적 망 관리 구조를 제시한다. 제시한 관리 구조는 JMX(Java Management Extensions)와 Mobile Agent 프레임워크를 이용한 분산 망 관리 구조이다.

### 1. 서론

이동 Ad-hoc 네트워크는 무선 인터페이스를 가진 이동 노드들이 자율적으로 구성하는 망으로 네트워크에 자율성과 융통성을 부여한 네트워크다. 이러한 네트워크는 기반 인프라스트럭처가 갖추어져 있지 않거나 설치가 용이하지 않은 지역 또는 시설의 이용이 불가능한 상황인 환경(재해 및 재난 지역 또는 전장터 등)에 적합한 네트워크다[1],[2].

Ad-hoc 네트워크는 다음과 같은 환경 때문에 기존 유선기반 네트워크와는 기본적으로 다른 특성을 가진다. Ad-hoc 네트워크는 기존의 인프라스트럭처의 도움 없이 이동 노드들이 자율적으로 다수의 홉 수를 통해 구성하는 네트워크이며, 네트워크를 구성하는 노드들은 무선 인터페이스를 가지며, 이동 컴퓨팅 기능을 가진 호스트와 라우터 기능을 가진 장비이다. 또한

이러한 이동 노드들은 단순한 센서부터 완전한 기능을 가진 랩탑(laptop)이나 라우터 등 다양한 장비로 구성된다. 따라서 네트워크가 수행하는 임무는 노드들 간의 상호 협력하는 그룹에 의해 분산되고 수행된다. 또한 노드들이 배터리로 운영되기 때문에 네트워크는 에너지를 보존을 위해 효율적으로 에너지를 사용토록 하여야 하며, 노드에 의해 제공되는 기능은 이용할 수 있는 에너지 수준과 능력에 따라 결정된다. 더욱이 노드의 자율적인 이동과 무선 인터페이스 특성에 의해 네트워크의 토폴로지가 빈번히 변하는 아주 동적인 환경 특성을 가진다. 마지막으로 한정된 에너지, 제한된 무선 인터페이스의 대역폭과 품질, 그리고 노드의 이동성으로 인해 잦은 네트워크의 분리를 초래한다.

이러한 특성에 의해 Ad-hoc 네트워크의 망 관리는 다음과 같은 요구사항을 가져야 한다. 첫번째는 다양한 기종의 노드와 여러 가지의 네트워크로 구성되고 언제 어디서나 사용자가 네트워크에 참여 서비스를 요청하므로 확장성을 가져야 한다.

본 연구는 한국과학재단 목적기초연구(R01-2002-000-00531-0) 지원으로 수행되었음.

두 번째는 네트워크의 다양한 토폴로지와 사용자와 서비스의 형태 그리고 사용자의 위치 등에 따라 다양한 관리 통제기능이 필요하므로 지적 관리 능력을 가져야 한다.

세 번째는 무선 링크 및 노드 이동성에 의한 긴급 상황에 대처하도록 견고함을 가져야 한다.

마지막으로 노드가 제한된 자원(대역폭, 품질 등)과 에너지를 가지므로 에너지 효율적이고 관리 트래픽이 적어야 한다. 이러한 요구사항을 고려할 때 Ad-hoc 네트워크에서의 망 관리는 기존 유선 망 기반의 관리에 비해 힘들고 어려운 작업이다. 따라서 일반적인 관리자/대행자 구조는 이러한 Ad-hoc 네트워크의 특성을 만족시키기에는 부적합하다[1].

이 논문에서는 Ad-hoc 네트워크 특성에 맞는 확장성 있고 동적이며, 견고하고 에너지 효율적인 망 관리구조를 제안한다. 제안하는 망 관리구조는 Ad-hoc 네트워크의 전개와 운영 및 효율적 자원 이용을 용이하도록 한다.

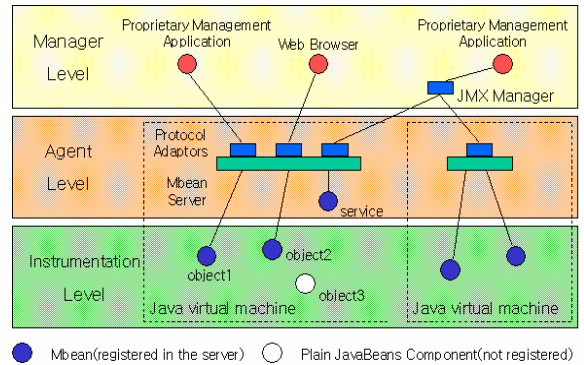
## 2. 관련 연구

기존 유선 망에서의 망 관리는 다양한 연구를 통해 이루어져 왔고 이미 여러 가지 망 관리 기법을 통해 보급되어 운영되고 있다. 그러나 Ad-hoc 네트워크가 재난 및 재해 지역이나 전쟁터 등 일부 군사분야에서 연구가 이루어져 왔고 최근에는 상업분야에서 활발히 연구되는 분야이다. 따라서 Ad-hoc 네트워크 특성에 맞는 망 관리 기법에 대한 연구가 미흡한 실정이다.

기존 SNMP(Simple Network Management Protocol)와 같은 중앙 집중식 망 관리 구조는 모든 관리 노드에 대한 관리를 하나의 관리 시스템이 담당하였는데, 관리 노드의 수가 많아지고 동적인 환경을 가지는 망에서는 하나의 관리 시스템으로는 관리할 수 없게 되었고(lack of scalability), 관리 대상에 설치된 관리 대행자가 고정된 기능만 수행할 수 있었으며(lack of dynamic/flexibility), 네트워크에 긴급상황이 발생하는 경우 대행자는 긴급상황을 스스로 해결하지 못하며(lack of robustness), 주기적 질의/응답에 의한 많은 관리 트래픽과 효율적인 에너지 관리 기능이 없다[3]. 따라서 중앙 집중식 망 관리는 Ad-hoc 네트워크의 망 관리에 부적합하다.

Management by Delegation(MbD)를 통해 망 관리를 효율적으로 수행하면서 Mobile Agent 라는 새로운 관리 구조도 연구 되고 있다[6]. 이는 프로세스와 코드가 함께 원격지의 대행자로 전달되어 관리 작업을 대행자로 분산시킨다. Mobile Agent 는 지적능력을 갖고 새로운 프로세스를 만들어 낼 수 있으며, 서버간에 이동이 가능하다. 또한 분산된 일을 스스로 수행 가능하며, 비동기적인 망 관리 수행이 가능하다[4],[7].

IETF Script MIB 은 기존의 관리 구조와 이동 코드의 개념을 통합한 분산 망 관리구조로 관리 기능을 원격지에 전달하고 실행시키기 위해서, SNMP 와 호환되는 MIB(Management Information Base)을 정의한다. Script MIB 은 분산된 지역에 관리 스크립트를 전달하고 원격지에 전달된 스크립트를 시작하거나 중단/종료 시



<그림 1> JMX 관리 구조

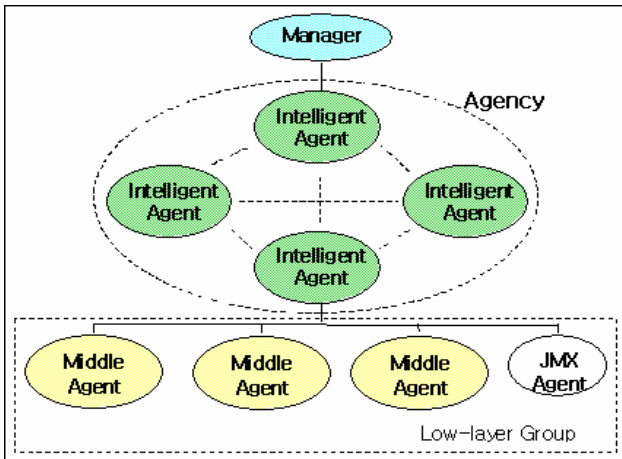
킬 수 있으며 실행되는 관리 스크립트에 필요한 매개 변수 전달과 실행중인 스크립트에 대한 모니터링 및 제어하고 스크립트의 실행 결과를 관리자에게 전달할 수 있다[3],[4].

JMX(Java Management Extensions)는 JMAPI(Java Management Application Programming Interface)에서 얻은 경험을 바탕으로 표준 자바 기술을 이용하여 쉽게 네트워크 관리 구성요소를 개발할 수 있는 방법을 제시하고 있다. 또한 JMX 는 기존의 관리 구조에 독립적이며 클라이언트-서버 모델을 사용하고 어떠한 관리 환경에도 통합되어질 수 있는 관리 프로토콜 API 를 갖고 있다. JMX 의 구조는 <그림 1>과 같이 Instrumentation Level, Agent Level, Manager Level 로 나뉘어진다. Instrumentation Level 은 관리 데이터를 모으거나 Agent Level 로 이 값을 돌려주는 역할을 하고 대행자가 지원하는 프로토콜로 관리 데이터를 일반화한다. 대행자 개발자는 이 레벨에서 관리 정보를 Mbean 라고 불리는 Java 구성요소로 구현한다. Agent Level 은 관리자의 요구를 받고 처리하고 서로 다른 프로토콜을 Mbean 에 의해 지원되는 일반적인 프로토콜로 관리자 요구를 바꾼다. Agent Level 은 이 특별한 프로토콜을 통해서 Mbean 을 갖고 제어하고 동작하는 Common Mbean Server 와 인터페이스를 갖는다. Manager Level 은 아직 완성되지 못하였으며, 기본적으로 JMX Agent 를 관리하기 위한 관리 응용프로그램을 제공하는 역할을 한다[8].

## 3. 제안하는 Ad-hoc 분산 망 관리 구조

Ad-hoc 네트워크의 크기와 토폴로지가 예측할 수 없는 패턴으로 변화함에 따라, 중앙 집중식 관리 패턴에서의 각 개별노드는 상호연동이 불가능하게 된다. 따라서, 기존의 일반적인 분산 망 관리 기법은 Ad-hoc 네트워크의 특성에 적합하지 않다.

기존 Script MIB 은 기존 관리 프로토콜과 통합적인 방법으로 관리 대행자에 관리 기능을 추가하기가 용이하며 다양한 언어를 이용하여 관리 스크립트의 개발이 가능하지만 관리 동작이 실행 가능한 코드나 스크립트의 형태로 제공되고 결과를 갖고 오는데 관리자의 추가적인 작업이 필요하며 기존 SNMP 의 단점을 유지하고 있다. 또한 Mobile Agent 는 Ad-hoc 네트워크의 특성에 적합하도록 기능을 수행할 수 있으나



<그림 2> Manager/Agency 모델

관리자/대행자의 구현을 위한 별도의 플랫폼이 필요하다[3]. 그리고 JMX는 표준 자바 기술을 이용한 망 관리 방법으로 구현이 쉽고 기존 망 관리 기술의 수용에 용이하나 프로세스와 코드의 이동성을 지원하지 않아 동적이고 지적인 환경을 갖는 Ad-hoc 네트워크 망 관리를 지원하기 어렵다.

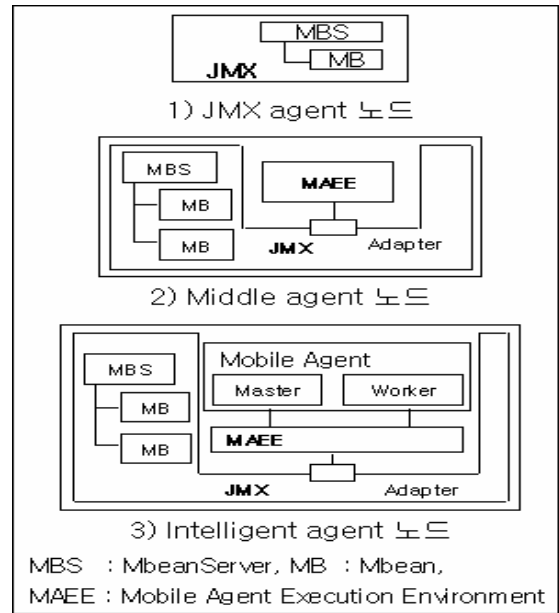
따라서 이 논문에서는 이동성, 자동성, 지적능력을 가진 Mobile Agent와 JMX의 구현의 용이 및 기존 망 관리와의 호환성을 통합하여 Ad-hoc 네트워크 관리 특성에 맞는 망 관리 모델과 메커니즘을 제안한다. 제안한 망 관리 구조는 관리 임무의 분산과 관리 트래픽 및 자원 사용을 줄이기 위한 동적 대행자의 연결과 상호 협력적인 관리 기능을 지원하기 위해 <그림 2>와 같은 Manager/Agency 모델을 제시한다.

Mobile Agent는 Grasshopper, Voyager 등 각 벤더마다 특징을 가지는 다양한 플랫폼이 존재하지만 IKV++사의 자바기반의 Grasshopper와 Java Virtual Machine 1.2.2를 사용하였다.

### Ad-hoc 망 관리 구조

Manager는 관리 임무를 상호 협력적으로 수행하는 Agency를 구성하는 Intelligent Agent 그룹에 관리정책을 분배 및 조정 통제한다. 각각의 Intelligent Agent는 원격지 환경에 적합하도록 네트워크의 동적인 환경에 적절히 대응한다. Agency내의 Intelligent Agent들은 event-control 알고리즘을 통해 상호 협력적인 관리 임무 수행과 견고함을 유지하므로 manager와의 신뢰성 있는 관리 기능의 수행을 보장한다.

Ad-hoc 네트워크의 동적인 환경과 능동적 관리를 위해 Mobile Agent 기술이 필요하다. Intelligent Agent의 지적관리 모듈은 필요 시 Mobile Agent를 통해 이동하거나 이식된다. 또한 Intelligent Agent노드는 관리 기능을 원격지로 전송하여 임무를 수행시키므로 노드나 링크의 failure에 적절히 대응하고 에너지와 대역폭 사용을 감소시킨다. 더욱이 지적인 관리 기능을 유지하기 위하여 네트워크 상황에 따른 이동-결심 알고리즘을 사용한다. 이동-결심 알고리즘은 관리노드의 정보와 Mobile Agent 기술을 통합하여 관리 기능을 지원한다.



<그림 3> 망 관리구조의 구성요소 분류

### Ad-hoc 망 관리 구조 구성 요소

제한한 구조에서, Ad-hoc 네트워크는 이동 노드의 상호 협력에 의해 스스로 그룹을 만들고 관리한다. Ad hoc 네트워크에서 노드는 그 기능과 능력에 따라 단순한 센서 노드부터 완전한 기능을 가진 랩탑이나 라우터로 구성된다. 따라서, 각 노드에서 수행되는 관리 기능은 서로 상이하다. 본 논문에서는 망 관리 구조 구성요소를 <그림 3>과 같이 3가지 형태로 분류한다. 모든 구성요소는 기본적으로 JMX 플랫폼이 실행될 수 있는 환경을 가져야 한다.

**JMX Agent 노드** : 이 노드는 기본적인 JMX 플랫폼만 실행될 수 있는 능력을 가진다.

**Middle Agent 노드** : JMX와 Mobile Agent 플랫폼 둘 다 실행시킬 수 있는 능력을 가지나, Mobile Agent 코드만을 실행가능하고, 에너지 및 노드의 능력이 향상될 경우 Intelligent Agent 역할 수행이 가능하다.

**Intelligent Agent 노드** : 이 노드는 Middle Agent 노드의 능력에 추가하여 하위 수준의 노드를 관리하는 기능, 지적 관리 모듈/상태정보와 Mobile Agent를 가진다. Mobile Agent는 Master와 Worker로 구성되는데 Master Agent는 지적관리 모듈을 관리하고 Agent의 스케줄링과 이동 경로 설정을 지원한다. Worker Agent는 manager의 관리정책이나 관리 임무(성능 감시 등)에 맞는 Agent Code를 생성하여 해당 원격지 노드로 정보를 전달하고 관리하는 역할을 한다.

지적 관리 모듈은 멤버 관리 기능, 이벤트 관리 기능, Mobile Agent 관리 기능을 가진다. 멤버 관리 기능은 각 노드의 JMX Agent에서 수집된 정보를 기반으로 그룹 내의 정보를 갱신하거나 필요 시 결과를 통해 Mobile Agent에게 지적 관리 모듈을 이식하거나 이동, 그리고 그룹을 분리하거나 통합하는 기능을 수행하게 한다. 이벤트 관리 기능은 Intelligent Agent 간 또는 자신의 환경 변화에 대한 적절한 대응을 위한 모

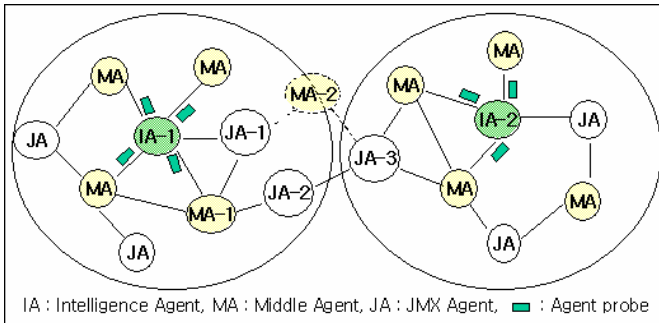


들이며, Mobile Agent 관리 기능은 agent의 생성 및 스케줄링 기능을 수행한다.

Node ID	Host ID	Hop 수	Power level	Capacity
---------	---------	-------	-------------	----------

<그림 4> Hello 메시지 형식

이 구조에서 모든 노드는 하나 이상의 Intelligent Agent를 중심으로 그룹을 형성한다. Intelligent Agent의 초기 설정은 관리자에 의해 수동으로 설정하거나 WCA(A Weighted Clustering Algorithm)을 통하여 자동적으로 선택된다[5]. 그룹에 속해있는 노드들은 자신의 위치와 능력을 이웃노드로 알리기 위하여 <그림 4>와 같은 hello 메시지를 주기적으로 전송한다. 이 메시지를 수신한 노드는 이웃노드와 링크를 인지하며 자신의 Mbean을 주기적 갱신하고 Intelligent Agent는 Agent Probe라는 Mobile Agent를 통해 정보를 수집하여 하위 수준의 노드를 관리한다. Node ID는 hello 메시지를 송신하는 자신의 ID이며 host ID는 자신을 관리하고 있는 Intelligent Agent ID, hop 수는 host에서 자신까지의 hop의 수이고 power level 및 capacity는 노드 자신의 용량을 의미한다.



<그림 5> Intelligent Agent 관리 토폴로지

#### 운용 시나리오

제안한 구조의 대표적인 실제 동작 방식은 Intelligent Agent 선정 후 토폴로지 정보의 수집과 관리, 노드의 추가 및 삭제, 관리 영역의 선정 등을 들 수 있다. 먼저, <그림 5>와 같은 환경이 초기 설정된 후 각 노드들은 이웃노드로 hello 메시지를 전송하여 이웃노드의 정보를 수집하고, Intelligent Agent는 Agent Probe를 관리노드로 전송하여 그룹 내의 모든 노드의 정보와 링크 상태 정보를 수집한다. Intelligent Agent는 hello 메시지 전송 시 노드 ID와 host ID를 동일하게 하고 hop 수는 '0'으로 설정하여 전송한다. Intelligent Agent로부터 hello 메시지를 수신한 노드들은 host ID와 hop 수를 갱신한 후 이웃노드로 전송한다. 각 노드에 구축된 정보는 Agent Probe를 통해 Intelligence Agent로 전달되어 그룹 내 노드들의 정보를 구축한다.

MA(Middle Agent)-1 노드가 기능을 상실하거나 그룹 외부로 이동하는 경우 이웃 노드에서 event 메시지를 IA(Intelligence Agent)-1로 전달하여 정보를 갱신하고 JA(JMX Agent)-2 노드는 기존 그룹과 연결이 해제되거나 JA-3로부터 메시지를 수신하고 있으므로 IA-2를 host로 선정하여 자신의 정보를 갱신한다. MA-2 노드와

같이 그룹에 참여하는 경우 MA-2는 JA-1과 JA-3에서 hello 메시지를 수신, 각 메시지의 hop 수를 비교하여 JA-1의 그룹에 가입하고 event 메시지를 JA-1로 전송하여 정보를 갱신하고 전체적인 토폴로지를 유지한다.

#### 4. 결론 및 향후 연구과제

Ad-hoc 네트워크의 동적인 토폴로지 변화, 제한된 전력, 한정된 무선 대역폭 및 품질 그리고 많은 이기종의 노드와 같은 특징은 망 관리를 더욱 어려운 연구로 만든다. 본 논문에서는 Ad-hoc 네트워크의 특성에 적합한 적응력 있고 자동적인 관리를 위하여 Mobile Agent와 JMX 플랫폼을 이용한 지적 망 관리 구조를 제안하였다. Agency 그룹은 지적관리 능력을 가지고, 동적 네트워크 환경에 적합하도록 상호 협력적으로 운영되고, 관리 영역 내 노드의 관리자 역할을 수행한다. 각각의 노드들은 주기적인 메시지 정보를 통해 이웃노드와의 토폴로지를 구성하고 Agent Probe를 통해 그룹 내 노드 관리 정보를 구축한다. 또한 Intelligent Agent는 노드와 네트워크 상황에 따라 자동적으로 이동하거나 이식되므로 동적인 관리 구조를 유지한다. 이 지적 관리 구조는 적응력 있고 동적이며 견고함과 효율적인 자원 활용을 지원한다.

본 논문의 지적 망 관리 구조를 이용하여 토폴로지 관리, 성능관리, 결함관리, 구성관리 및 QoS 관리 등의 구체적인 적용 방안과 Mobile Agent의 이동 경로 설정에 대한 알고리즘 등의 연구가 필요하다.

#### 참고문헌

- [1] Chien chung shen, An Adaptive Management Architecture for Ad Hoc Networks, IEEE Communications Magazine, February 2003
- [2] Wenli Chen, ANMP: Ad Hoc Network Management Protocol, IEEE, August 1999
- [3] 송봉규, 분산객체를 이용한 관리 응용 프로그램 개발 및 QoS 관리 방안, 연세대학교, 2001
- [4] Romit Roy Choudhury, An Agent-Based Connection Management Protocol for Ad Hoc Wireless Networks, Journal of Network and System Management, December 2002
- [5] Mainak Chatterjee, An On-Demand Weighted Clustering Algorithm(WCA) for Ad hoc networks, IEEE Globecom, 2000
- [6] Timon C. Du, Mobile Agents in Distributed Network Management, ACM, July 2003
- [7] C. Bohoris, Mobile Agent-Based Performance Management for the Virtual Home Environment, Journal of Network and System Management, June 2003
- [8] Addison Wesley, Java™ and JMX: Building Manageable Systems, December 2002