

# OSGi 프레임워크기반 확장형 SNMP 에이전트 번들 설계

A Design of Scalable SNMP Agent Bundle Based on OSGi Framework

황태인, 박광로 • 한국전자통신연구원 네트워크연구소 홈네트워크팀

Taein Hwang, Kwangroh Park • Home Network Team, Network Technology Laboratory, ETRI

## 요약

기존의 SNMP 에이전트는 컴포넌트화 되어있지 않고 코드 이동성을 제공하기 위한 프레임워크가 없기 때문에 원격에서 온라인으로 에이전트의 MIB 코드를 새롭게 추가한다거나 에이전트 자체를 업그레이드하기 어려웠다. 이런 문제를 해결하기 위해서 OSGi 프레임워크 상으로 다운로드 가능한 SNMP 에이전트 번들을 SNMP 메시지 처리를 위한 SNMP 에이전트 기본 번들과 관리 정보 수집을 위한 MIB 번들로 나누어 설계하였으며 번들 서버가 인증을 거친 후 MIB 번들을 동적으로 업로드 할 수 있도록 하기 위하여 다운로드 서버 번들과 사용자 인증 번들을 추가하였다. 또한 네트워크 관리자가 장치에 탑재되어 있는 MIB 번들의 상태 정보를 모니터링할 수 있도록 하기 위하여 MB-MIB를 설계하였다.

본 논문에서 설계한 OSGi 기반 SNMP 에이전트 번들을 사용함으로써, 관리자는 원격에서 온라인으로 관리하고자 하는 MIB 모듈과 SNMP 에이전트 기본 모듈을 업그레이드시킬 수 있으므로 다변화하는 대규모 네트워크를 관리하는 시간과 비용을 줄일 수 있다.

## Abstract

A traditional agent was difficult to upgrade and to add MIB implementation code to it because most SNMP agent designs had MIB that was not a self-contained module. To solve this problem, we design the SNMP agent bundle composed of remotely uploadable SNMP agent core bundle and MIB Bundle. Also, downloadable server bundle and user administration bundle is added to SNMP agent bundle to download MIB bundle from bundle server. By designing MB-MIB, we enable the network manager to monitor status of MIB bundle on managed devices. The network manager is allowed to install and upgrade the SNMP module or MIB remotely on line. This is the major benefit of designing the SNMP agent bundle based on OSGi. Finally, we expect the network manager to reduce time and expenses in managing enormous networks by using the SNMP agent bundle.

## I. 서론

인터넷 사용자의 폭발적인 증가, 기술의 발전, 두 대 이상의 컴퓨터를 소유한 가정의 증가, 그리고 인터넷 정보 가진 기기의 등장은 네트워크의 성장을 가속화 시키고 있으며 인터넷이 없으면 생활하는데 불편함을 느낄 정도로 사회의 깊은 곳까지 인터넷의 물결이 파급되어 가고 있다. 또한, 인터넷에 접속하는 단말의 수가 급속도로 증가하고 있으며 이로 인하여 네트워크가 갈수록 복잡해질뿐만 아니라 다양해지고 있다.

이런 네트워크 환경하에서 실시간 정보 중에서의 의미 있는 정보를 추출하여 분석하고 그 정보를 바탕으로 시스템을 구성해야 하는 네트워크 관리자들은 상당한 부담을 안게 되었다.

네트워크 관리자가 겪는 관리의 어려움 중에 하나는 피관리 시스템에 탑재되는 Simple Network Management Protocol(SNMP) 에이전트가 하드웨어 의존적이기 때문에 관리자가 원하는 형태로 수정하기가 힘들다는 것이다. 네트워크 장치에 탑재되는 SNMP 에이전트는 관리자로부터의 SNMP 요구 메시지에 응답하기 위한 소프트웨어로서 장치를 관리하는데 필요한 정보를 Management Information Base(MIB)로 구성하여 관리자의 요구에 응답한다.

MIB는 장치를 개발한 업체에서 제공하기 때문에 장치 출고 이후 업체 직원이 직접 방문하여 소프트웨어를 업그레이드 하지 않으면 수정하는 것이 힘들다. 왜냐하면 SNMP 에이전트 모듈을 수정하기 위해서는 업그레이드할 소프트웨어 모듈을 컴파일하고 장치를 켜다온 시킨 후 재시동해야 하기 때문이다[1].

이와 같은 작업은 힘들고 번거로운 작업이며 장치를 납품한 벤더들이나 관리자가 변경하고자 하는 MIB 변수 및 실행코드를 새로운 관리 정보

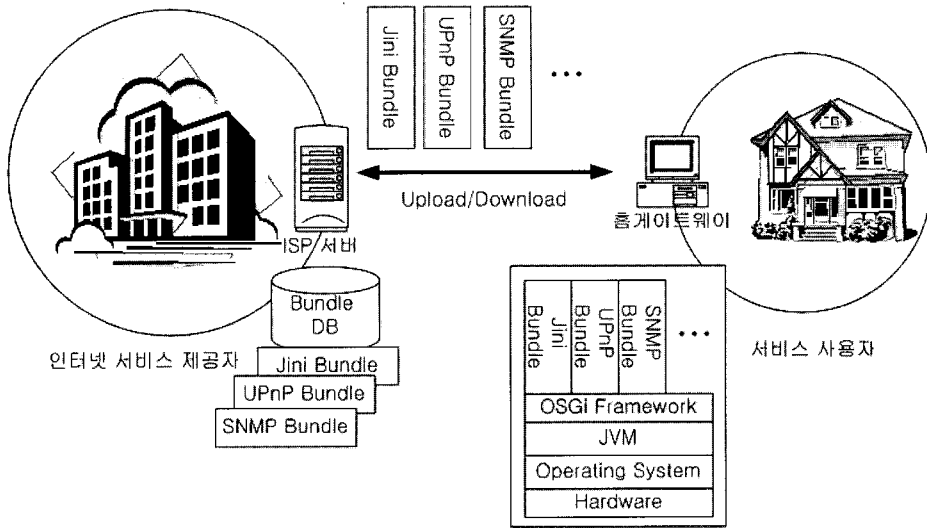
로 변경하기 어렵게 만드는 요인이다. 대부분의 SNMP 에이전트에 의해 노출된 또 다른 문제는 코드가 컴포넌트화 되어있지 않을 뿐만 아니라 이동성을 가진 실행 모듈이 동작할 수 있는 환경을 제공해주지 못하기 때문에 새로운 MIB 모듈을 원격에 위치한 장치에서 사용할 수 있도록 업로드 한다거나, 다른 어플리케이션으로의 통합이 힘들다는 것이다.

앞으로는 더욱더 복잡하고 다양한 네트워크 관리 어플리케이션이 등장하고 이런 형태의 어플리케이션으로 발전해갈 것이기 때문에 SNMP 에이전트의 일부 실행 모듈의 이동을 위한 프레임워크와 컴포넌트화가 필요하다.

본 논문에서는 번들로 알려진 확장 및 다운로드 가능한 서비스 어플리케이션을 위한 실행 환경인 Open Service Gateway initiative(OSGi) 프레임워크를 이용한다. [그림 1]과 같이 OSGi 프레임워크를 탑재한 장치는 전체 시스템을 재시동할 필요없이 켜져 있는 상태에서 번들을 다운로드받아 설치 및 제거시킬 수 있다[2]. 이와 같은 OSGi의 특징을 이용하여 SNMP 프로토콜을 OSGi 프레임워크기반 위에 적용하여 번들화시킴으로써 벤더들이 네트워크 장치를 관리하기 위한 소프트웨어를 온라인상에서 원격으로 확장할 수 있기 때문에 유지 보수가 편리하며 관리하는 시간과 비용을 절감할 수 있다.

본 논문의 2장에서는 SNMP 에이전트에서 MIB 구현 부분을 분리하기 위한 선행 연구와 본 논문을 비교한다. 3장에서는 네트워크 관리 시스템의 기본 모델에 번들 서버를 추가하여 OSGi 프레임워크에 기반한 네트워크 관리 시스템의 구조를 제안하며, 제안한 시스템을 기반으로 MIB 번들을 원격에서 온라인으로 다운로드할 수 있는 SNMP 에이전트 번들의 구조를 제안한다.

4장에서는 3장에서 제안한 MIB 번들의 상태



(그림 1) OSGi 프레임워크 개념도

정보를 관리하기 위한 MIB Bundle-MIB(MB-MIB)를 정의하고, 5장에서 결론을 맺고자 한다.

## II. 관련 연구

일반적으로 SNMP 에이전트를 구현함에 있어 MIB를 구현한 코드는 에이전트 코드안에서 함께 구현된다. 즉, MIB 코드와 에이전트 코드가 완전히 서로 독립적인 컴포넌트로 구현되는 것이 아니라 함께 구현된다.

MIB에서 관리 정보를 얻어오기 위해서 호출되는 함수의 구현이 MIB 모듈에만 영향을 미치는 것이 아니라 전체 에이전트 코드에 영향을 주기 때문에 SNMP 에이전트의 MIB 모듈을 업그레이드하기 위해서는 SNMP 에이전트 전체 코드를 장치에 설치하고 컴파일한 후 재시동하여야 한다[1]. 이와 같은 문제를 해결하기 위해서 여러 선행 연구들이 수행되었다.

Kalyanasundaram은 SNMP 에이전트에서

MIB의 구현 부분을 분리하기 위하여 Spreadsheet 개념을 도입하여 이차원의 셀 테이블로 구성된 MIB 모듈을 내장하는 에이전트를 제안하였다[3]. 각 셀은 데이터를 가지고 있는 셀과 그 셀로부터 결과 값을 유도하기 위한 스크립트로 구성된다.

Spreadsheet 에이전트의 매니저는 셀의 값을 얻어오기 위한 스크립트를 SNMP SET 요구 메시지의 데이터 필드에 포함시켜 에이전트로 전달하며 이를 통해 새로운 셀을 만들 수 있다. 제안한 방법은 에이전트에서 실행 가능한 스크립트 언어로 구성되어 있기 때문에 언제든지 SNMP SET 요구 메시지에 의해서 스크립트를 MIB에 할당해 주어 MIB 값을 얻어오기 위한 실행 함수를 동적으로 변경할 수 있지만, 전체적인 MIB의 구성을 바꾸지 못하며 Spreadsheet로 구성되어 있는 일부 MIB의 수정만이 가능하다는 단점이 있다.

Ajita John은 동적으로 MIB 객체를 추가하거나 삭제할 수 있는 XNAMEI 에이전트를 제안하였

다[1, 4]. 이 에이전트는 SNMP SET 메시지의 데이터 필드에 스크립트 문자열을 바이트 코드 형태로 포함하여 전달하기 때문에 전달된 스크립트를 직접 실행시켜서 원하는 정보를 얻을 수 있으며 XNAMI MIB를 정의하여 실행 스크립트를 포함하는 객체를 추가하거나 삭제하는 것이 가능하다. 하지만, SNMP 에이전트의 메시지 처리 모듈, 보안 모듈과 같은 기능 모듈을 새로운 버전으로 업그레이드 한다거나 장치에 의존적인 정보들, 예를 들어, 디바이스 드라이브에 의존적인 정보, 운영체제와 관련된 정보는 얻어올 수가 없다.

본 논문에서 제안하는 SNMP 에이전트 번들은 MIB를 구현한 코드 자체를 자바 번들 소프트웨어로 구현함으로써 언제든지 관리자는 MIB의 전체 구성과 실행 코드를 변경하여 번들 서버를 이용하여 원격에서 온라인으로 서비스 번들을 업로드할 수 있기 때문에 에이전트를 컴파일하고 재시동할 필요가 없다.

SNMP 에이전트 번들은 MIB의 정보를 실제로 수집하기 위한 모듈이 컴포넌트로 따로 존재하며 이 번들은 자바로 구현되어 있어서 Java Native Interface(JNI)를 통하여 직접 디바이스 드라이버나 운영체제로부터 정보를 가져올 수 있다. [표 1]은 본 논문을 다섯 가지 항목으로 나누어 선행 연구와 비교한 표이다.

### III. SNMP 에이전트 번들 설계

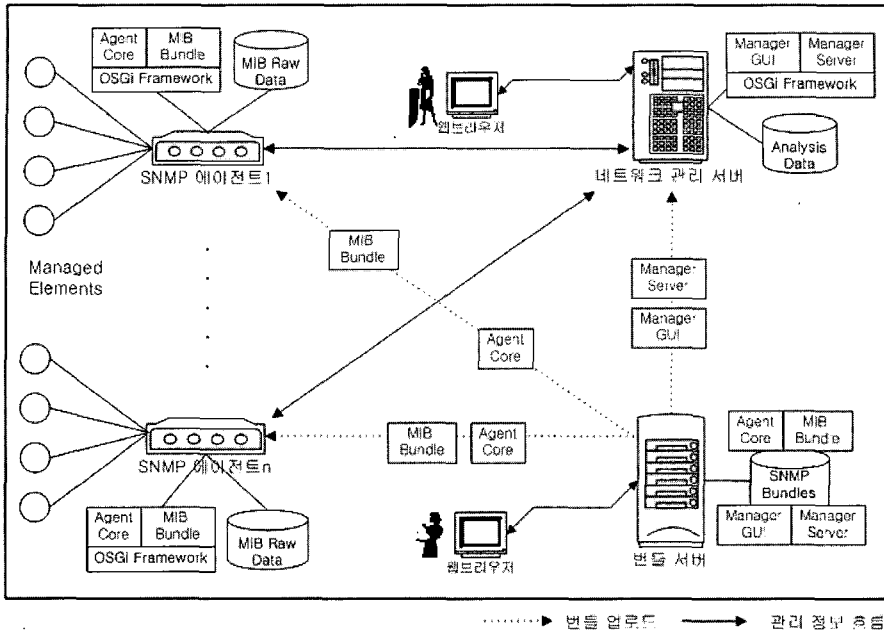
SNMPv3가 제안되면서 SNMP 코드 구현에 있어서 상당 부분 모듈화가 진행되어 기존 코드에서 새로운 기능 모듈이나 MIB 모듈이 분리되었다[5-7]. 그러나, SNMPv3 에이전트를 구현하여 온라인상에서 에이전트 모듈을 추가하거나 삭제하는 작업은 OSGi 개념이 출현하기 전에는 힘든 일이었다. 왜냐하면, 코드 자체는 모듈화가 되어 있지만, 각 모듈로 구성된 실행 코드의 이동성을 지원해 주기 위한 프레임워크가 없었기 때문이다. 하지만, OSGi 프레임워크의 출현으로 각 모듈을 번들화 할 수 있게 됨으로써 온라인 업그레이드가 가능하게 되었다[8-10]. 이 장에서는 OSGi 프레임워크 상에서 이동성을 갖춘 SNMP 에이전트 번들 개발을 위한 OSGi 기반 네트워크 관리 시스템의 구조와 SNMP 에이전트 번들의 구조를 제안한다. 번들간 서비스 호출을 위한 서비스 인터페이스를 정의하고, 번들간 서비스 호출 시나리오를 제시한다.

#### 1. OSGi 기반 네트워크 관리 시스템

OSGi 기반 네트워크 관리 시스템의 구조는 [그림 2]와 같이 SNMP 에이전트, 네트워크 관리 서버, 그리고, 번들 서버로 구성된다. SNMP

(표 1) 선행 연구와의 비교

	Kalyanasundaram	Ajita John	Proposed Scheme
적용 기반 기술	Spreadsheet	XML	OSGi
에이전트 코드 컴포넌트화	X	X	O
MIB 객체 동적 추가, 삭제	O	O	O
에이전트 모듈 업그레이드	X	X	O
네이티브 코드 구현 가능	X	X	O



(그림 2) OSGi 기반 네트워크 관리 시스템의 구조

에이전트는 관리 시스템의 가장 하부에 위치하며 네트워크 관리 서버로부터 관리를 받는 장치로서, 관리 서버가 그 장치에 특정 정보를 요구할 경우 이에 대한 응답 메시지를 제공하며, OSGi 프레임워크가 설치되어 있어서 번들 서버가 제공하는 번들을 제공 받아 설치 및 업그레이드가 가능하다.

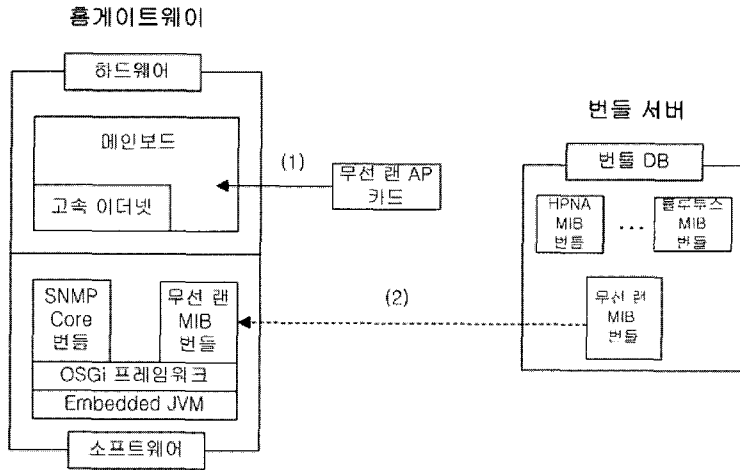
네트워크 관리 서버는 SNMP 에이전트로부터 받은 관리 정보를 이용하여 하부의 피관리 시스템을 관리하기 위한 서버이며, Manager Server 번들이 OSGi 프레임워크 위에 설치되어 있어서 SNMP 에이전트에게 관리하고자 하는 MIB 정보를 요구하는 메시지를 전송한다. 또한, 네트워크 관리자는 웹 브라우저를 통하여 네트워크 관리 서버에 있는 Manager GUI를 이용하여 네트워크 정보를 원격에서 모니터링하고 분석할 수 있다.

번들 서버는 다수의 번들을 데이터베이스에 보

유하고 있으며, 서비스 제공자는 웹 브라우저를 통하여 이 서버에 접근하여 원격에서 SNMP 에이전트나 네트워크 관리 서버의 프레임워크 상으로 번들을 전달하고 설치할 수 있다.

[그림 2]의 시스템이 목표로하는 피관리 장치인 홈게이트웨이는 맥내망과 액세스망을 연결시켜주기 위한 네트워크 종단 접속 장치로서 다양한 유무선 인터페이스를 제공한다. 홈게이트웨이의 액세스망 인터페이스로는 구리선을 이용한 xDSL, 동축을 이용한 케이블, 그리고 위성 및 광 케이블 등이 있다. 또한 홈게이트웨이의 맥내망 인터페이스로는 IEEE1394, 고속 이더넷, HomePNA, 전력선, USB 등의 유선 인터페이스와 무선 랜, 블루투스, HomeRF, WPAN 및 IrDA 등의 무선 인터페이스가 있다.

일반적으로 홈게이트웨이는 다양한 인터페이스로의 확장성을 고려하여 PCI(Peripheral Component Interconnect) 버스 인터페이스를



[그림 3] 홈게이트웨이상에서의 MIB 번들 전송

가지도록 설계한다. 따라서 홈게이트웨이에서 쉽게 다양한 네트워크 인터페이스를 추가 및 삭제할 수 있으나 변경된 인터페이스 관리 정보를 수집할 수 있는 MIB 모듈의 즉각적인 추가가 힘들기 때문에 새로운 인터페이스의 실시간 동적 관리가 어렵다. 그러나, 제안한 시스템 구조는 홈게이트웨이의 인터페이스가 동적으로 변경되더라도 확장 인터페이스에 대한 관리 정보를 수집할 수 있는 MIB 번들을 원격에서 설치하고 업그레이드할 수 있기 때문에 피관리 장치에 대한 관리 기능을 수행할 수 있다.

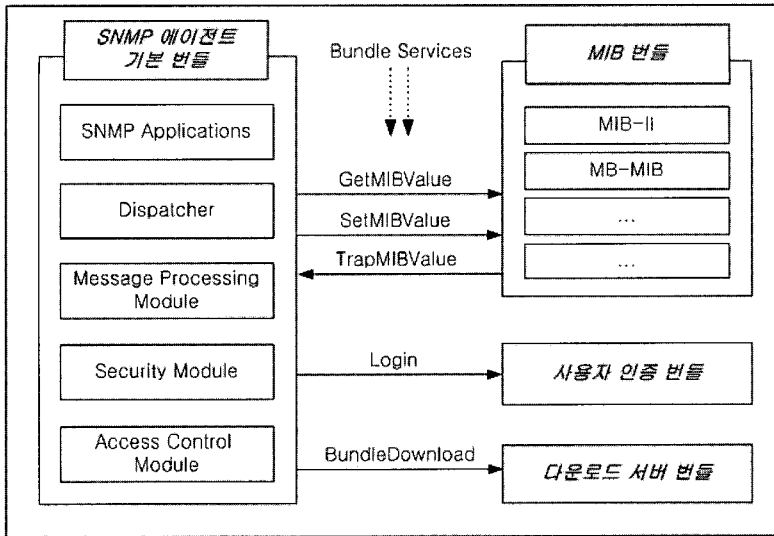
SNMP Core 번들이 설치된 홈게이트웨이상으로 MIB 번들을 전송하는 과정은 [그림 3]과 같다. OSGi 프레임워크와 임베디드 자바 가상머신이 탑재된 홈게이트웨이에서 무선 랜 Access Point(AP) 인터페이스를 추가하여 AP에 대한 성능 및 장애 정보를 원격에서 관리하고자 할 경우, 무선 랜 AP 기능을 수행하는 카드를 PCI 버스 인터페이스에 장착하고(1) 번들 서버에서 홈게이트웨이로 무선 랜 AP를 관리하기 위한 무선 랜 MIB 번들을 전송함으로써(2) 관리

기능을 수행한다.

## 2. SNMP 에이전트 번들 구조

OSGi 기반 네트워크 관리 시스템의 번들 서버에서 SNMP 에이전트로 업로드하는 SNMP 에이전트 번들은 [그림 4]와 같이 SNMP 에이전트 기본 번들, MIB 번들, 사용자 인증 번들, 그리고 다운로드 서버 번들로 구성되어 있다. 그중에서 SNMP 에이전트 기본 번들은 RFC2271에서 정의한 SNMP 관리 구조를 토대로 하여 구성하였다.

서비스 제공자가 번들 서버를 이용하여 번들 업로드를 위한 메시지를 보내면, SNMP 에이전트 기본 번들은 사용자 인증 번들의 login 서비스를 호출하여 인증 기능을 수행한 후 MIB 번들의 GetMIBValue, SetMIBValue 서비스를 호출하여 관리자가 필요로 하는 MIB 정보를 수집 및 설정한다. MIB 번들은 SNMP 에이전트 기본 번들의 TrapMIBValue 서비스를 호출하여 네트워크 관리자에게 장치의 상태 정보를 전달할 수 있



(그림 4) SNMP 에이전트 번들의 구조

다. SNMP 에이전트에서 서비스 제공자가 전달하는 번들을 다운로드 하기 위해서는 다운로드 서버 번들이 파일을 받을 수 있도록 대기해야 하며 이를 위해서 에이전트 기본 번들은 다운로드 서버의 BundleDownload 서비스를 호출하여 다운로드를 위한 준비를 한다.

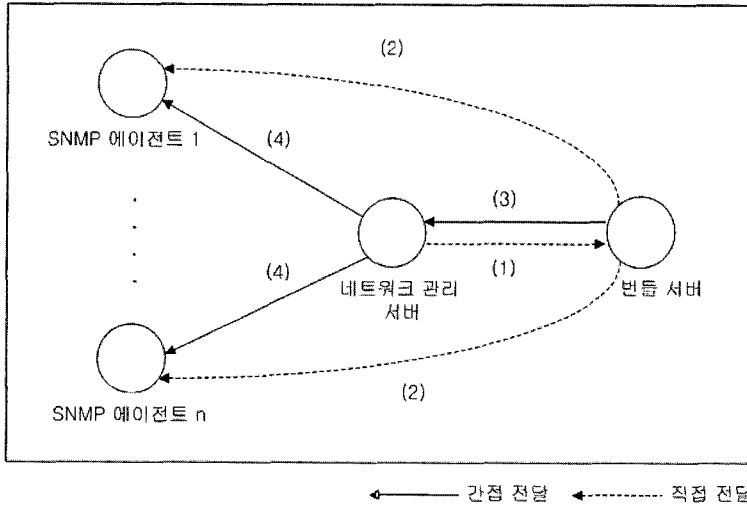
### 3. 번들간 서비스 호출 시나리오

다운로드 서버 번들을 통해 전달되는 MIB 번들은 (그림 5)와 같이 직접 전달 방식과 간접 전달 방식으로 SNMP 에이전트에 전달될 수 있다. 직접 전달 방식은 번들 서버에서 직접 SNMP 에이전트가 탑재된 장치로 번들을 업로드 하는 방식이며, 이 방식은 업그레이드해야 할 장치가 몇 대 되지 않아 업로드하는 데 크게 부하가 걸리지 않을 경우에 가능한 방법이며 업로드하기에 앞서 네트워크 관리 서버로부터 업그레이드해야 할 관리 대상 장치에 대한 정보를 얻어온 후(1) 그 정보를 이용하여 해당 장치의 번들을 업그레이드

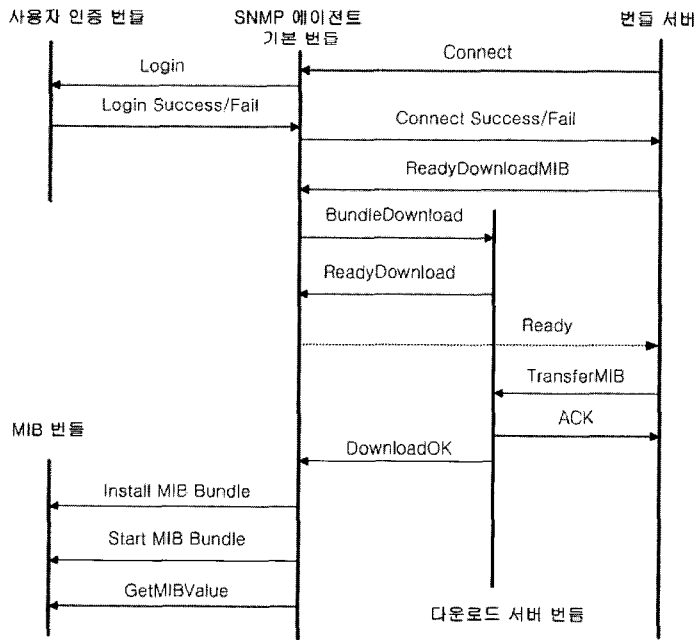
한다(2).

간접 전달 방식은 네트워크 관리 서버가 번들 서버로의 역할을 하는 경우이며 네트워크 관리 서버가 번들을 업로드할 수 있는 기능을 수행할 수 있는 모듈을 가지고 있어야 한다. 번들 서버가 네트워크 관리 서버에게 MIB 번들을 전달하고 (3) 네트워크 관리 서버가 피관리 장치의 SNMP 에이전트에게 MIB 번들을 전달하는 방식이다 (4). 이와 같은 방법은 대규모 네트워크에서 관리하고자 하는 대상 번들이 많아서 분산하여 MIB 번들을 업로드해야 할 경우에 적합하며 직접 전달 방식에 비해 번들 업로드를 위한 트래픽을 분산시킬 수 있는 이점이 있다.

번들 서버로부터 MIB 번들을 업로드하는 과정에서 SNMP 에이전트 기본 번들, MIB 번들, 다운로드 서버 번들, 사용자 인증 번들간 서비스 호출 시나리오를 (그림 6)에 나타내었다. 번들 서버는 MIB 번들을 업로드하기 위해서 우선 SNMP 에이전트 기본 번들로의 Connect 메시지 전송을 통하여 호를 설정하며 사용자 인증 번



(그림 5) MIB 번들 전달 방식

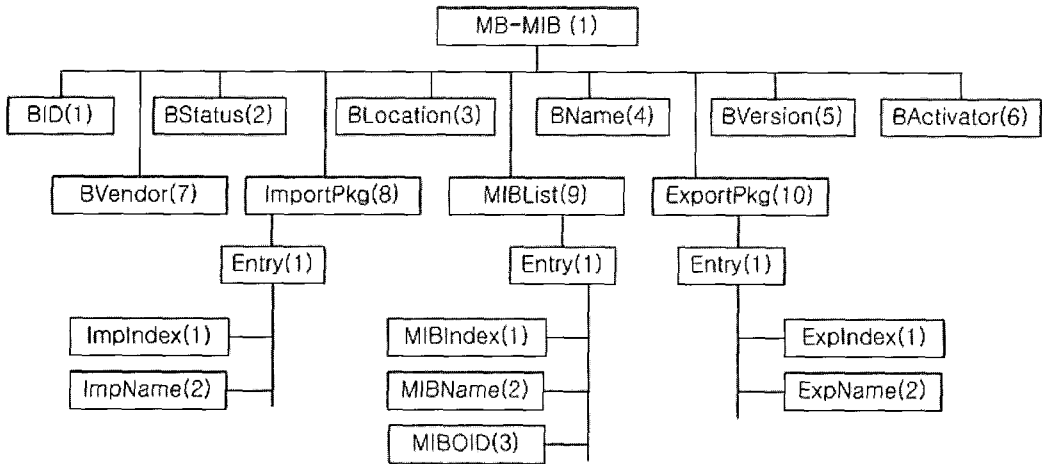


(그림 6) 번들간 서비스 호출 시나리오

들로부터 사용 권한을 인증받는다. 인증이 끝나면 번들 서버는 ReadyDownloadMIB 메시지를 전송하여 다운로드 서버 번들이 번들을 다운로드 받을 수 있는 준비를 하게 한다.

번들을 다운로드 받을 수 있는 준비가 끝나면, 다운로드 서버는 번들 서버로부터 번들을 다운로드 받은 후 ACK 메시지를 보내어 정상적으로 다운로드가 끝났다는 것을 알려준다. SNMP 에이





(그림 7) MB-MIB 트리

전트는 MIB 번들의 다운로드가 정상적으로 완료되었는지 확인하기 위하여 MIB 번들의 GetMIBValue 서비스를 호출하여 MIB 값을 정상적으로 수집하는지 확인한다.

#### IV. MB-MIB(MIB Bundle-MIB) 구조

MIB 번들이 정상적인 상태로 동작하고 있는지 또는 현재 SNMP 에이전트가 제공하는 MIB 정보들에는 어떤 것들이 있는지를 알기 위해서는 그런 정보를 수집할 수 있는 MIB가 필요하며 이를 위하여 (그림 7)과 같은 MB-MIB를 정의한다.

MB-MIB는 [표 2]와 같이 번들의 ID, 번들의 현재 상태 및 번들이 설치되어 있는 파일 시스템의 위치, 번들 이름, 버전, 번들을 활성화 시키기 위한 초기 코드를 담고 있는 번들 Activator, 제조회사, Import 패키지, Export 패키지, MIB 리스트를 포함한다. MIB 리스트에는 현재 MIB 번들에서 제공해 주는 MIB에 대한 정보를 제공하며, MIB의 이름과 해당 MIB의 루트 객체 식별

자를 제공해 주기 때문에 이 객체 식별자를 통하여 요구한 MIB 객체가 어떤 MIB에 속하는지를 유추할 수 있다. MB-MIB는 MIB 번들에 포함되어 3장에서 정의한 시나리오에 따라 다운로드 서버 번들에 의해 다운로드되어 추가될 수 있다. MIB 번들은 SNMP 에이전트 기본 번들의 GetMIBValue 서비스를 통하여 네트워크 관리 서버가 요구하는 관리 객체에 대한 식별자를 인자로 받게 되며, 이 인자에 해당하는 MIB를 MIB 트리로부터 찾게 된다. MIB 번들은 최상위 루트 객체 하부에 개별 MIB에 대한 서브 트리를 연결하여 하나의 트리를 형성함으로써 일관성 있는 MIB 객체의 조회가 가능하다.

#### V. 결론

기존의 SNMP 에이전트는 하드웨어 의존적이기 때문에 관리자가 원하는 형태로 수정하기가 힘들었으며, 컴포넌트화 되어 있지 않고 코드 이동성을 제공해 주기 위한 프레임워크가 없었기

[표 2] MB-MIB 객체 정의

MIB	Object Name	Type	Identifier	Description
MB-MIB	BID	Integer	1.1	번들 고유 ID 번호
	BStatus	Integer	1.2	번들 상태
	BLocation	DisplayString	1.3	번들이 설치된 위치
	BName	DisplayString	1.4	번들 이름
	BVersion	DisplayString	1.5	번들 버전
	BActivator	DisplayString	1.6	번들 Activator
	BVendor	DisplayString	1.7	번들 제조사명
	ImportPkg	Sequence	1.8	Import 패키지
	<i>ImpIndex</i>	<i>Integer</i>	<i>1.8.1</i>	<i>Import 패키지 인덱스 번호</i>
	<i>ImpName</i>	<i>DisplayString</i>	<i>1.8.2</i>	<i>Import 패키지명</i>
	MIBList	Sequence	1.9	MIB 리스트
	<i>MIBIndex</i>	<i>Integer</i>	<i>1.9.1</i>	<i>MIB 인덱스 번호</i>
	<i>MIBName</i>	<i>DisplayString</i>	<i>1.9.2</i>	<i>MIB 이름</i>
	<i>MIBOID</i>	<i>DisplayString</i>	<i>1.9.3</i>	<i>MIB 루트 객체 식별자</i>
	ExportPkg	Sequence	1.10	Export 패키지
	<i>ExpIndex</i>	<i>Integer</i>	<i>1.10.1</i>	<i>Export 인덱스 번호</i>
	<i>ExpName</i>	<i>DisplayString</i>	<i>1.10.2</i>	<i>Export 패키지명</i>

때문에 MIB 코드를 새롭게 추가한다거나 에이전트를 업그레이드시키기 어려웠다. 본 논문에서는 OSGi 개념을 이용하여 장치를 동적으로 관리해 줄 수 있는 확장 가능한 SNMP 에이전트 번들을 설계하였다.

SNMP 에이전트 번들 구조를 제안하기 위하여 번들 서버, 네트워크 관리 서버, SNMP 에이전트로 구성된 OSGi 기반 네트워크 관리 구조를 제안하였으며 이 구조를 바탕으로 번들 서버로부터 관리되어야 할 장치로 전달 가능한 SNMP 에이전트 번들을 설계하였다. SNMP 에이전트 번들을 SNMP 메시지를 처리하기 위한 SNMP 에이전트 기본 번들, 관리 정보를 수집하기 위한 MIB 번들, 번들 서버가 제공해 주는 번들을 받

기 위한 다운로드 서버 번들, 그리고 번들 서버의 접근 권한을 인증하는 사용자 인증 번들로 구분하였으며 각 번들간 서비스 호출 시나리오를 제시하였다.

SNMP 에이전트 번들의 설계 외에도 MIB 번들을 관리하기 위한 MB-MIB를 정의하였다. 이 MIB는 MIB 번들에 대한 여러 가지 상태 정보를 제공함으로써 서비스 제공자나 네트워크 관리자가 현재 SNMP 에이전트에 설치되어 있는 MIB 번들의 상태를 실시간으로 모니터링할 수 있으며, 현재 관리하고 있는 도메인에 MB-MIB를 가지고 있는 SNMP 에이전트가 새로 추가 되었을 경우 그 에이전트가 보유하고 있는 MIB 번들에 대한 상세 정보를 제공해 주기 때문에 관리의 효

을성을 증대시킬 수 있다.

제안한 SNMP 에이전트 번들은 일반적인 SNMP 에이전트와는 달리 관리자가 필요로 하는 기능을 온라인상에서 원격으로 업그레이드할 수 있다. SNMP 에이전트 번들을 하나의 번들이 아니라 다른 기능을 수행하는 여러 개의 서브 번들로 분리함으로써 네트워크 관리자가 관리하고자 하는 장치의 SNMP 에이전트 모듈을 새로운 모듈로 업그레이드하고자 할 때 교체하기를 원하는 일부 번들만을 교체할 수 있기 때문에 각 관리 장치로의 관리 번들 업로드 시간을 단축시킬 수 있다.

본 논문에서 설계한 OSGi 기반 SNMP 에이전트 번들을 사용함으로써 관리자는 SNMP 에이전트 모듈을 원격에서 편리하게 업그레이드시킬 수 있으므로 대규모 네트워크를 관리하는 관리자에게 드는 시간과 비용을 절감할 수 있다.

저자 소개



황 태 인

1999년 성균관대학교 정보공학과  
 2001년 성균관대학교 대학원  
 2001년~현재 ETRI 네트워크연구소  
 홈네트워크팀  
 관심 분야 : 홈네트워크 기술, 네트워크  
 관리, 제어미들웨어기술



박 광 로

1982년 경북대학교 전자공학과(학사)  
 1985년 경북대학교 대학원(석사)  
 2002년 충북대학교 대학원(박사)  
 1984년~현재 ETRI 네트워크연구소 홈  
 네트워크팀장(책임연구원)

관심 분야 : 홈네트워크 기술, 무선LAN 기술, VoIP 기술,  
 L-Biz

■ 참고문헌

- [1] A. John, K. Vanderveen, and B. Sugla, "XNAMI-An eXtensible-based paradigm for Network and Application Management Instrumentation," *IEEE International Conference on Networks*, pp. 115-124, 1999.
- [2] M. Condry, U. Gall and P. Delisle, "Open Service Gateway Architecture Overview," *IECON' 99*, pp. 735-742, 1999.
- [3] P. Kalyanasundaram et al., "A Spreadsheet-Based Scripting Environment for SNMP," *in Integrated Network Management V*, pp. 752-765, 1997.
- [4] A. John, "A Java-Based SNMP Agent for Dynamic MIBs," *Global Communication Conference*, pp. 396-400, 1999.
- [5] W. Stallings, *SNMP, SNMPv2, SNMPv3, and RMON1 and 2*, Addison-Wesley, 1999.
- [6] D. Perkins, and E. McGinnis, *Understanding SNMP MIBs*, Prentice Hall, 1997.
- [7] D. Zeltserman, *A Practical Guide to SNMP v3 and Network Management*, Prentice Hall, 1999.
- [8] D. Marples, "The Open Services Gateway Initiative: An Introductory Overview," *IEEE Communication Magazine*, pp. 110-114, 2001.
- [9] K. Chen and L. Gong, *Programming Open Service Gateways with Java Embedded Server Technology*, Sun Microsystems, 2001.
- [10] D. Valtchev and I. Frankov, "Service Gateway Architecture for a Smart Home," *IEEE Communications Magazine*, pp. 126-132, Apr. 2002.