

제어 기능을 갖는 디지털 컨버전스 장비를 위한 SNMP 확장에 관한 연구

허 길[†] · 김 은 회^{††} · 최 재 영^{†††}

요 약

SNMP는 네트워크 장비들을 관리하기 위한 표준 프로토콜로서, 확장성, 정보관리, 인증, 암호화, 접근 제어 등과 같은 우수한 장비 관리기능을 제공한다. 그러나 상황적응형 서비스를 제공하기 위한 제어 기능을 갖는 디지털 컨버전스 장비들을 통합 관리하기에는 제어기능이 취약하며, SNMP 통신 메시지의 길이에 제약이 있다는 단점이 있다. 이러한 문제점을 해결하기 위하여, 본 논문에서는 제어기능을 가진 디지털 컨버전스 장비들을 통합 관리하기 위한 확장된 SNMP를 제안한다. 확장된 SNMP는 다양한 디지털 컨버전스 장비들을 효과적으로 제어하기 위해 SNMP 엔진에 DDS (Device Driver Subsystem)를 추가하고 CADM (Control Agent Driver Model)을 정의하여 그 구조를 개선하였다. 따라서 설정과 제어 구분의 모호성을 해결하였고, 추상화된 인터페이스를 지원함으로써 SNMP 어플리케이션이 장비를 쉽게 제어할 수 있도록 하였다. 또한 SNMP 통신 메시지 길이의 제약을 받지 않는 세 개의 SNMP 명령어를 추가함으로써 용량이 큰 고수준의 데이터도 전송이 가능하도록 하였다.

키워드 : 원격관리, 디지털 컨버전스, SNMP, 장비 제어

An Extended SNMP Scheme for a Digital Convergence Device with Control Functions

Gil Heo[†] · Eunhoe Kim^{††} · Jaeyoung Choi^{†††}

ABSTRACT

SNMP (Simple Network Management Protocol) is a standard protocol for management of network devices, and it provides excellent features such as scalability, information management, authentication, encryption, and access control. However, SNMP has a structural weakness to fully support control functions for integrated management of digital convergence devices, and it has a limitation of message length in SNMP communication. In this paper, we present an extended SNMP scheme for integrated management of digital convergence devices with control functions. We modified the SNMP architecture by adding DDS (Device Driver Subsystem) to SNMP engine for controlling different devices and by defining CADM (Control Agent Driver Model), therefore we solved the ambiguity problem between 'set' and 'control' of SNMP. And the extended SNMP made it easy for SNMP applications to use various control functions. The extended SNMP can transport massive high-level information by adding three new SNMP commands which eliminate the limit of message length.

Keywords : Remote Management, Digital Convergence, SNMP, Device Control

1. 서 론

보다 편리하고 쾌적한 삶을 영위하고자 하는 사람들의 욕구가 더욱 증대됨에 따라서, 환경 보존, 에너지, 건강, 주거 환경, 사회 환경, 교육 등 국가 전반에 걸친 다양한 분야에

서 새로운 서비스들이 활발하게 개발되고 있다. 또한 오늘날에는 서비스를 제공받는 사용자에게 현재 상황을 반영한 더 높은 수준의 서비스를 제공하고자 하는 추세이기 때문에, 상황 정보를 수집하기 위한 관심이 높아지고 있다[1].

특히 사회 환경 분야에서는 교통 정보나 사고, 공기오염도, 날씨 등 수집되는 상황 정보들을 바탕으로 교통 통제 서비스와 같은 공익 성격이 강한 서비스들을 제공한다. 이러한 종류의 서비스들은 보통 관제 센터를 중심으로 광범위한 범위에서 상황 정보를 수집하고 가공하여 필요한 서비스 정보를 생성한다. 따라서 관제 센터에서 상황정보를 수집하는 원격 장비를 효율적으로 통합 관리하기 위한 기술이 중

* 본 연구는 지식경제부 및 정보통신산업진흥원의 대학IT연구센터 지원사업의 연구결과로 수행되었음. NIPA-2009-(C1090-0902-0007)

† 정 회 원 : (주)브레넥스 기술연구소 책임연구원

†† 정 회 원 : 숭실대학교 지능정보기술연구소 전임연구원

††† 총신회원 : 숭실대학교 컴퓨터학부 교수

논문접수 : 2009년 4월 20일

수정일 : 1차 2009년 7월 29일

심사완료 : 2009년 8월 10일

요한 관심사가 되었다.

또한 사용자가 요구하는 서비스의 수준이 높아지면서 정보의 집중화를 방지하고 서비스의 효율성을 높이기 위하여, 중앙 집중식 처리방식에 개별 분산식 처리방식을 결합한 새로운 서비스 시나리오가 활발히 개발되고 있다 [2]. 즉, 상황 정보를 수집하는 장비들은 중앙 관제센터의 개입없이 수집한 상황정보를 기반으로 일정수준의 단순한 상황적응형 서비스들을 개별적으로 처리한다. 오늘날에는 신호등이나 가로등, 광고판, 벽 등과 같이 주변에서 흔히 볼 수 있는 다양한 공공시설물들을 활용하여 상황 정보를 수집하며, 동시에 사용자에게 실제적으로 서비스를 제공하는 매체로도 사용된다. 따라서 이러한 목적을 위한 공공시설물은 상황정보를 수집하는 기능과 함께, 자체적으로 서비스를 제공하기 위한 제어기능을 갖는 새로운 형태의 디지털 컨버전스 장비로 주목받고 있다. 따라서 제어기능을 갖는 디지털 컨버전스 장비들은 특정 목적을 위한 전용 장비라기보다, 주변의 환경에 대한 정보를 수집하고, 적합한 서비스를 제공하기 위해 적절히 판단할 수 있는 능력이 있으며, 적절한 서비스를 제공하는 매체로 활용될 수 있다 [3].

본 논문에서는 먼저 네트워크 장비의 관리와 상태 모니터링을 위해 널리 사용되고 있는 SNMP를 제어기능을 가진 디지털 컨버전스 장비의 관리에 적용할 때의 문제점을 파악한 후, SNMP를 확장하여 그 해결책을 제시하였다. 디지털 컨버전스 장비의 장치 제어를 지원하기 위하여 SNMP 엔진에 DDS(Device Driver Subsystem)를 추가하고 CADM(Control Agent Driver Model)을 정의하여 SNMP의 구조를 확장하며, SNMP set 명령어를 확장한 새로운 제어용 set 명령어를 정의하였다. 또한, 발달하는 기술과 함께 높아지는 사용자의 요구사항에 적합한 고수준의 서비스를 제공하기 위하여 대용량 메시지를 전송할 수 있는 SNMP 기술이 필

요하다. 이러한 요구사항을 해결하기 위하여 새롭게 정의한 제어용 set 명령어의 메시지 규약을 해지하고, 용량이 큰 메시지를 처리할 수 있도록 기존의 Response와 InformRequest 명령어를 보완한 명령어를 새롭게 정의하여 추가하였다.

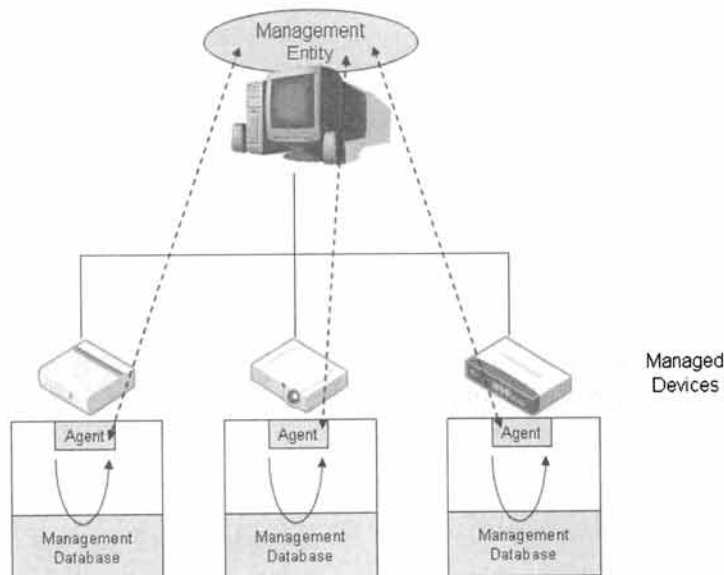
본 논문의 구성은 다음과 같다. 2장에서는 SNMP의 기본적인 구조와 제어기능 지원 방법을 설명한다. 그리고 제어기능을 갖는 디지털 컨버전스 장비를 관리하기 위한 시스템의 일반적인 구조를 설명한다. 3장에서는 제어기능을 갖는 디지털 컨버전스 장비들을 관리하기 위한 관련연구들을 논의한다. 4장에서는 확장 SNMP를 소개하고, 5장에서는 확장 SNMP의 실험결과를 보이고, 성능평가를 한다. 6장에서는 결론과 향후 연구 과제를 논한다.

2. 배 경

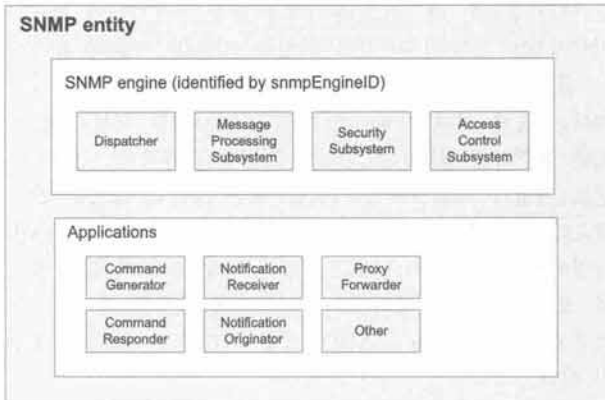
2.1 SNMP 개요

SNMP는 네트워크 장비를 관리 감시하기 위한 목적으로 TCP/IP 상에 정의된 응용 계층 표준 프로토콜이다. SNMP로 관리되는 장비들은 자신의 정보를 MIB (Management Information Base)라고 불리는 트리 형태로 유지하게 되는데, 원격 관리자는 해당 장비의 MIB 노드의 값을 참조하는 방식으로 장비를 관리한다. MIB는 ASN.1 표기법을 따르는 SMI (Structure of Management Information) 형식으로 정의되어 공개된다. 따라서 관리 대상 장비의 MIB 정의만 알면 바로 관리가 가능한 것이 SNMP가 갖는 장점 중의 하나이다. (그림 1)은 SNMP 관리를 개념적으로 보여주고 있다 [4, 5].

MIB의 노드들은 OID (Object Identifier)로 구분되는데, 그 값은 MIB의 루트 노드로부터 해당 노드까지의 경로이다. 각 노드들에 부여되는 숫자는 해당 노드의 형제 노드들



(그림 1) SNMP 관리 개념도



(그림 2) SNMP 엔티티 구조

(sibling)과의 구분을 위한 식별 번호이기 때문에 형제 노드들 간의 크기나 순서를 뜻하지는 않는다. MIB의 구조는 표준으로 정의되어 있으나, OID 1.3.6.1.4.1에 해당하는 enterprises(1) 노드에 장비 고유의 정보를 추가하는 부분이 있다. 이 부분에 각 제조회사별로 부여된 고유의 회사번호와 장비번호를 이용하여 장비 고유의 MIB 서브트리를 정의한다.

SNMP를 사용하여 관리하거나 관리되는 대상을 SNMP 매니저 또는 SNMP 에이전트라고 불렀으나, 가장 최근 버전인 SNMP v3에서는 모두 SNMP 엔티티라고 부른다 [6]. SNMP 엔티티는 프로토콜을 버전별로 해석하고 인증과 접근 제어 처리를 담당하는 SNMP 엔진과 MIB로 접근하여 실제 관리되는 정보들을 불러오거나 수정하는 기능을 수행하는 어플리케이션들로 구성된다. (그림 2)는 SNMP v3에서의 SNMP 엔티티 구조를 보여주고 있다.

2.2 SNMP에서의 제어 처리

장비를 효과적으로 관리하기 위해서는 필요에 따라 장비의 설정을 변경해야 하고, 장비 재시작과 같은 장비 제어가 필요한 경우가 있다. 그러나 SNMP에서는 장비를 제어하기 위한 별도의 명령어를 지원하지 않기 때문에, SNMP의 SetRequest 명령어를 활용하여 장비를 제어하는 방법을 사

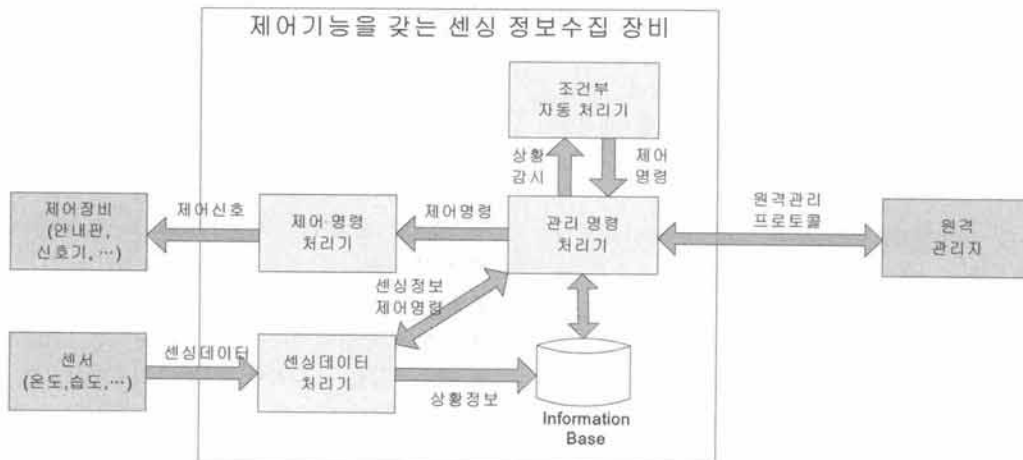
용하고 있다. 명령어 후킹(command hooking) [7], OID 모니터링 [8], 설정 및 실행(Set & Execute) [9] 방법 등이 SNMP의 SetRequest 명령어를 활용하는 대표적인 방법이다.

명령어 후킹 방법은 장비의 SNMP 어플리케이션에 명령어가 도착할 때마다 이를 가로채서 명령어와 OID를 검사하고, OID가 제어를 필요로 하는 OID와 일치하면 그에 따른 제어를 처리하는 방법이다. 명령어 후킹 방법은 가장 많이 사용되고 있는 SNMP SetRequest를 활용하는 장비제어 방법이다. OID 모니터링 방법은 특정 OID를 모니터링하여 변화를 감지했을 때 간접적으로 제어를 처리하는 방법이다. 마지막으로 설정 및 실행 방법은 SNMP 엔티티가 제어 처리를 위한 게이트웨이로서의 역할을 수행하고 제어 처리는 별도의 프로세스를 이용하여 처리하는 방법으로 SNMP로 관리가 가능한 서버장비에서 주로 사용된다. 본 논문에서의 장비 제어는 명령어 후킹 방법을 사용한다.

2.3 제어기능을 갖는 디지털 컨버전스 장비

제어기능을 갖는 디지털 컨버전스 장비는 상황적응형 서비스를 실현하기 위하여 상황을 인식하는데 필요한 정보를 수집하고 이를 원격 관리자로 전달하는 역할을 하는 센싱 정보수집 기능과 원격 관리자로부터 제어 명령을 전달받아 장비 자체 또는 외부에 연결된 작동 장치를 제어하는 역할을 수행할 수 있는 제어 기능을 갖춘 장비이다. 주로 신호등이나 가로등, 광고판, 벽 등과 같은 공공시설에 임베디드되어 있다. (그림 3)은 제어기능을 갖는 디지털 컨버전스 장비를 관리하기 위한 시스템의 일반적인 구조를 나타내고 있다.

제어기능을 갖는 디지털 컨버전스 장비는 일반적으로 센서와 연결되는 센싱 데이터 처리기와 장비로 전달되는 제어 명령을 처리하는 제어명령 처리기, 원격 관리자로부터의 요청을 처리하고 모니터링과 같은 서비스를 제공하는 관리 명령 처리기, 그리고 자료 공유를 위한 정보베이스(Information Base)로 구성된다. 종류에 따라 자동화 처리를 위하여 상황 정보의 조건부 자동 처리기가 추가로 포함되기도 한다. 조



(그림 3) 제어기능을 갖춘 디지털 컨버전스 장비 관리시스템의 일반적인 구조

건부 자동 처리기는 수집되는 센싱 정보를 모니터링하다가 검사하는 조건이 만족되면 그에 따른 적절한 동작 명령을 발생시키는 모듈이다. 본 논문에서는 제어기능을 갖는 디지털 컨버전스 장비의 관리를 위해 센싱 데이터 처리기, 제어 명령 처리기, 관리 명령 처리기, 정보베이스, 원격 관리자 기능을 4장에서 제안하는 확장 SNMP을 이용하여 설계하고 구현한다.

3. 관련 연구

현재 네트워크로 연결된 디지털 컨버전스 장비들을 관리하기 위하여 내장형 웹과 CGI를 이용한 관리 [10], 자체 프로토콜을 사용하는 Java 애플릿 또는 전용 도구를 이용한 관리 [11], SNMP을 이용한 관리 등이 사용되고 있다. 먼저, 웹서버와 CGI를 이용하고 PC에서 웹브라우저로 접속하여 각종 설정을 변경하거나 버튼 클릭으로 장비를 제어하는 방법은 가장 널리 사용되는 방법이지만, 소규모 장비의 설정과 제어에 유리하기 때문에 통합 관제센터와 같은 대규모 관리 서버에는 적합하지 않다. 자체 프로토콜을 사용하는 Java 애플릿이나 전용 도구를 사용하는 관리 기법은 주로 디지털 컨버전스 장비들의 실시간 모니터링에 많이 사용된다. 자체 프로토콜을 사용하므로 장비에서 제공되는 모든 기능을 자유롭게 활용할 수 있다는 장점을 가지지만, 이질적인 장비들을 통합 관리하기에는 부적합하다. SNMP를 이용하면 장비의 유지 보수 및 대규모 시스템 관리, 그리고 이질적인 장비의 통합 관리에 유리한데, 이는 SNMP와 같은 표준 관리 프로토콜이 갖는 장점이다. 또한 SNMP 트랩(trap) 기능은 긴급 상황을 정확하게 관리자에게 통보할 수 있다. 따라서 관리를 위해 SNMP에 대한 사전 지식을 습득해야 하는 어려움은 있지만, 센싱정보 수집 기능 위주의 기존 디지털 컨버전스 장비들을 통합 관제하는 센터에 유용하게 활용될 수 있다.

제어기능을 갖춘 디지털 컨버전스 장비는 상황정보를 수집하기 위한 센서들뿐만 아니라 수집된 상황정보를 기반으로 서비스를 수행하기 위한 작동기(actuator)들을 갖추고 있는 장비이다. 따라서 제어기능을 갖춘 디지털 컨버전스 장비들을 관리하기 위해서는 연결되어 있는 센서들뿐만 아니라 작동기들까지 효율적으로 관리할 수 있어야 한다. SNMP를 기반으로 임베디드 장비에 연결되어 있는 센서들과 작동기들을 효율적으로 관리하기 위한 관련 연구로는 [12-14]가 있다.

[12]는 센서와 제어장치들로 구성된 여러 종류의 필드버스(fieldbus) 통신망을 통합 관리하기 위하여 구조적으로 하나의 마스터 에이전트와 여러 개의 서브 에이전트 구조를 가지는 확장된 SNMP 기술을 사용한다. 즉 각각의 필드버스 통신망에 하나의 SNMP 서브 에이전트를 구동시켜 데이터를 수집하고, 서브 에이전트들을 통합하는 하나의 마스터 에이전트는 SNMP 요청을 처리하고 MIB를 관리한다. [12]

는 여러 종류의 필드버스 통신망을 통합 관리하기에 적합한 기술이지만, SNMP와 함께 확장된 SNMP 기술의 습득이 더 필요하고, SNMP 본연의 기능인 모니터링에 적합하지만 제어 기술에 대한 구체적인 방법은 제시하지 못하고 있다. 또한 연결되는 필드버스 통신망에 적합한 각각의 서브 에이전트 개발이 필요하므로 개발을 위한 시간과 노력이 많이 필요하다. 반면 본 논문에서 제시하는 확장 SNMP는 SNMP 명령어만을 확장하였으므로 별도의 기술 습득이 필요 없으며, 또한 연결되는 센서 또는 작동기에 적합한 드라이버를 개발하면 쉽게 확장 SNMP로 통합 관리할 수 있다는 장점이 있다.

[13]은 센서와 작동기들을 모니터링하고 제어하기 위한 국제표준인 IEEE 1451.2의 STIM (Smart Transducer Interface Module)을 SNMP 기반으로 개발함으로써 SNMP의 장점인 표준 프로토콜로서의 유연성 및 구현의 용이성과 IEEE 1451.2 표준성을 장점으로 갖는 관리 시스템을 구축하였다. IEEE 1451의 NCAP (Network Capable Application Processor)는 하드웨어 리소스의 요구사항이 높고, 사용하는 객체 모델을 실제 구현하기 매우 어렵다는 단점을 가지므로 네트워크에 연결을 제공하는 NCAP을 SNMP 에이전트로 대체한 것이다. 그러나 IEEE 1451.2 STIM의 센싱 데이터를 저장하기 위한 포맷인 TEDS (Transducer Electronic Data Sheet)를 SNMP의 MIB로 바꿀 때 몇몇 데이터 타입에 대하여 정확하게 MIB 데이터 타입으로 변환할 수 없다는 문제점이 발생하며, IEEE 1451.2를 지원하는 센서와 작동기 전용이라는 한계성을 가진다. 반면 본 논문에서 제안하는 확장 SNMP는 데이터 변환이 필요없고, CADM의 드라이버 인터페이스로 간단한 read, write 오퍼레이션을 제공하므로 드라이버 개발이 쉬워 확장성이 뛰어나다. 따라서 CADM을 위한 IEEE 1451.2의 STIM 드라이버를 개발하여 IEEE 1451.2 센서와 작동기를 관리할 수 있다.

[14]는 센서와 제어장치들을 포함하는 분산 시스템을 통합 관리하기 위하여 SNMP 에이전트를 개발하였다. [14]는 센서나 작동기에서 발생하는 데이터를 검사하여 에이전트에 기록되어 있는 알람 조건과 비교하고 알람 상황이 발생하면 SNMP Trap을 이용하여 관리자에게 이를 알린다. [14]는 센서와 제어장치를 효과적으로 관리하기 위하여 새롭게 TestRequest, TrapAck라는 명령어를 정의하여 제공하고 있다. [14]는 새로운 SNMP 명령어를 정의하여 센서와 제어장치를 관리한다는 점에서 본 논문과 접근법이 비슷하지만, 본 논문에서는 SNMP의 취약한 제어기능을 보강하기 위하여 SNMP 명령어를 확장하였으므로 그 목적이 다르다. 또한, [14]는 센서나 제어장치들이 연결되어 있는 통신망 또는 임베디드 장비에 최적화된 SNMP 에이전트들을 각각 개발해야 하는 단점이 있는 반면, 본 연구는 CADM이라는 센서 및 작동기들을 추상화한 모델을 사용함으로써 센서나 작동기에 독립적인 SNMP 에이전트를 개발할 수 있다는 장점을 가진다.

4. 확장 SNMP 설계

4.1 확장 SNMP 개요

사회 인프라 시설에 임베디드되어 있는 제어기능을 갖춘 디지털 컨버전스 장비들을 관리할 때, 서로 다른 플랫폼을 사용하는 장비들을 통합 관리하는 것이 필요하므로, SNMP와 같은 표준화된 프로토콜을 사용하는 것이 유용하다. 그런데 제어 기능의 비중이 더욱 커지면서 다양한 장비를 제어해야 할 필요성이 대두되었지만, SNMP 프로토콜은 장비 제어를 위한 별도의 명령어를 지원하지 않는다. SNMP에 제어 기능을 제공하기 위하여 SNMP SetRequest 명령을 가로채서 장비를 제어하는 방법들을 사용하지만, 여전히 점차 다양해지는 제어 장비들을 효율적으로 제어할 수 있는 방법은 없다. 이러한 문제점들을 해결하기 위하여 본 논문에서는 SNMP 엔티티의 구조를 개선함으로써 SNMP를 확장하였다. 또한 상황 적응형 서비스 제공에 적합한 용량이 큰 메시지 처리를 가능하게 하기 위하여 새로운 명령어를 추가로 정의하였다.

4.1.1 SNMP 엔티티 구조개선

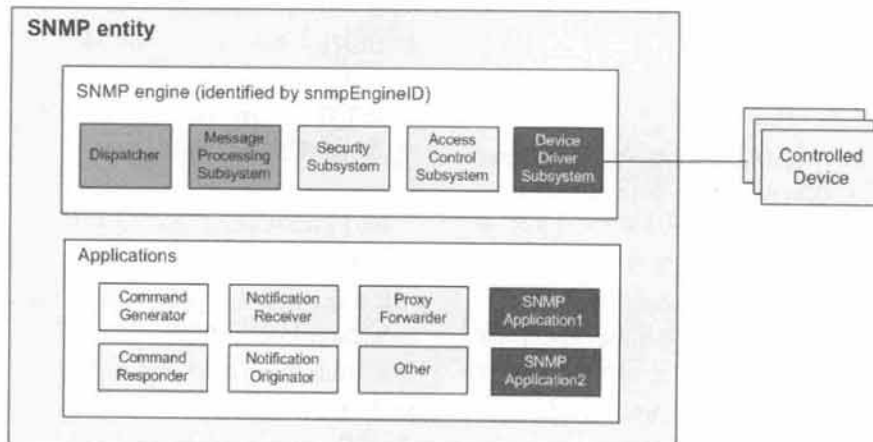
다양한 장치 제어를 위해 관리 프로토콜 차원의 서비스를 제공하도록 SNMP 엔티티의 엔진 부분에 장치 드라이버 서브시스템 개념을 도입하여 제어 장치에 대한 드라이버 계층화와 서비스 추상화를 제공한다. 이와 같은 장치 드라이버 개념의 도입은 제어 장치에 대한 접근을 용이하게 만들어주기 때문에 다양한 형태의 상황 적응형 서비스를 제공할 수 있는 SNMP 어플리케이션을 쉽게 탑재할 수 있게 한다. 따라서 SNMP 엔진 내에 새로운 서브시스템인 DDS (Device Driver Subsystem)를 추가하고 장치 드라이버에 대한 고수준의 서비스 API를 제공하여 장치 계층화와 추상화를 제공한다. SNMP 엔진 내의 다른 서브시스템과 마찬가지로 DDS는 목적별로 다양한 형태의 Model들을 설계할 수 있다. 본 논문에서는 장치 제어와 관련된 모델인 CADM (Control Actuator Driver Model)을 제시한다.

(그림 4)는 SNMP 엔진 내에 DDS를 새로 추가한 SNMP 엔티티 구조도를 나타내고 있다. 새로 추가되는 SNMP 엔티티의 서브시스템인 DDS는 장치로의 접근과 관련된 서브시스템으로써, SNMP 엔티티 내의 모든 어플리케이션은 DDS의 서비스 API를 이용하여 장비를 공유하여 사용할 수 있다. (그림 4)의 SNMP 어플리케이션은 DDS의 서비스 API를 이용하는 상황인지 어플리케이션이다.

4.1.2 SNMP 명령어 추가

SNMP의 명령어는 576 bytes를 넘지 않도록 권장하고 있다 [15, 16]. TCP/IP 통신이 갖는 패킷의 단편화(fragmentation) 현상이 통신의 신뢰성을 떨어뜨리기 때문에, 어떠한 프로토콜을 사용하거나 네트워크 장비를 경유하더라도 단편화가 일어나지 않는 크기인 576 bytes를 넘지 않도록 권장하고 있다. 그러나 GetBulkRequest 명령어와 같이 메시지의 길이가 길어지면서 회사에 따라 임의로 최대 길이를 576 byte보다 크게 설정하여 사용하고 있지만, 표준 권장 사항이 아니므로 장비 간의 큰 메시지는 메시지의 전달이 보장되지 않는다.

그러나 발달하는 기술과 함께 높아지는 사용자의 요구사항에 적합한 고수준의 서비스를 제공하기 위하여 점차 용량이 큰 메시지를 전송할 수 있는 SNMP 명령어의 필요성이 증가하고 있다. 그 한 예로 국내에 설치되어 있는 도로전광표지판(VMS: Variable message Signs) 관리를 들 수 있다. VMS의 경우 일반적으로 한글을 한 줄에 8음절씩 두 줄까지 표시할 수 있는 400*100 해상도를 갖고, 2색 또는 3색 LED를 사용한다. 따라서 한 화면을 표현하는데 10KB (2색의 경우), 20KB (3색의 경우) 정도의 정보가 필요하다. 고품질의 정보표현을 위하여 출력 색상수가 점차로 늘어나는 추세이고, VMS의 해상도 또한 점점 높아지는 추세를 생각할 때, 한 화면을 표현하기 위한 정보의 양은 더욱 증가할 것이다. 따라서 관제센터에서 SNMP 에이전트에 연결되어 있는 VMS의 화면을 관리 및 제어하기 위한 정보의 교환량은 567 bytes의 메시지 제약을 가지는 SNMP 명령어로는 불가



(그림 4) 확장된 SNMPv3 엔티티 구조도

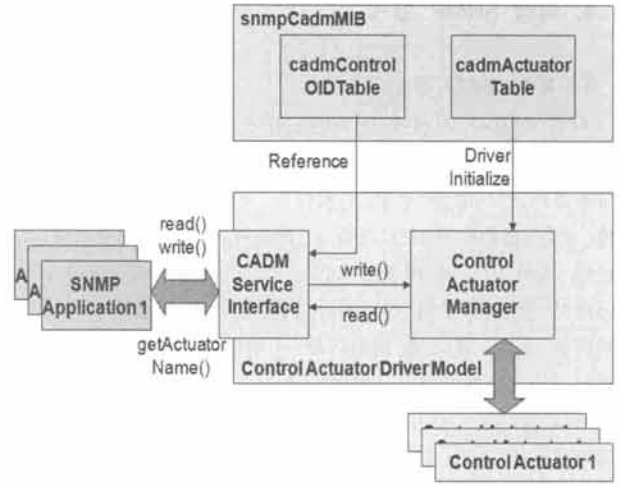
능하다.

대용량 메시지 전달이 필요한 다른 예로 로그정보의 관리를 들 수 있다. 센서나 작동기에 동작내역을 기록한 로그 정보를 저장하고 있을 때 관리자는 필요에 따라 로그정보를 요청하게 된다. 기기의 종류와 동작 내역에 따라 로그정보의 종류는 매우 가변적일 수 있으므로 그 크기 또한 다양하다. 따라서 관제센터에서 576 bytes이상의 로그 정보를 전송받기 위해서는 명령 메시지 크기의 제약이 없는 SNMP 명령어가 필요하다. 또한, 최근에는 데이터의 상호운용성을 높이기 위하여 데이터를 XML 형태로 주고 받는 경우가 많다. 그러나 텍스트 기반의 XML 메시지의 단점은 메시지 크기가 커진다는 것이므로 XML 문서를 포함하는 메시지를 SNMP에서 전송하기 위해서는 메시지 크기의 제약이 없는 SNMP 명령어가 필요하다.

이를 위하여 본 논문에서는 기존의 SetRequest, Response 명령어와 업링크(uplink)형 통지(notification)를 처리하는 InformRequest 명령어를 보완하여 snmpV3e-set-request, snmpV3e-response, snmpV3e-inform-request의 3개의 명령어를 추가로 정의하였다. snmpV3e-set-request는 제어용 set 명령어로서 용량이 큰 메시지를 전송할 수 있는 다운링크형 제어 명령어이다. 따라서 관제센터에서 VMS의 화면에 고품질을 위한 용량이 큰 데이터를 표시하고 싶을 때, VMS 제어 명령어와 표시하고자 하는 데이터를 snmpV3e-set-request 명령어에 담아 VMS를 관리하는 에이전트에 전송하게 된다. snmpV3e-response는 센서 또는 작동기에서 용량이 큰 데이터를 가져올 때 사용한다. 즉 위의 로그 정보의 예처럼 관제센터에서 센서 또는 작동기에 있는 로그정보를 획득하고 싶을 때, SNMP get 명령어를 이용하여 요청하게 되고 에이전트는 snmpV3e-response 명령어에 용량이 큰 로그 정보를 실어 응답메시지를 전송한다. snmpV3e-inform-request는 센서나 작동기에서 발생한 통지사항을 관제센터, 즉 SNMP 매니저로 전송하는 명령어로서 용량이 큰 통지메시지를 전송할 수 있다. 새로 추가되는 명령어는 새로운 명령어 코드를 사용하기 때문에 확장 SNMP를 지원하지 않는 기존의 SNMP 원격 관리자나 SNMP 장비와의 통신에서 알 수 없는 명령어 코드로 인식하므로 기존 장비에서 오동작이 발생되지 않는다.

4.2 CADM 정의

DDS의 드라이버 모델로 CADM (Control Actuator Driver Model)을 정의하였다. CADM은 장치를 제어하기 위하여 장치 추상화를 통한 서비스 API를 SNMP 어플리케이션에 제공한다. (그림 5)는 CADM의 내부 구조와 SNMP 어플리케이션과 CADM을 위해 정의한 snmpCadmMIB와의 관계를 나타내고 있다. CADM은 SNMP 어플리케이션이 제어장비에 접근하고 제어할 수 있도록 CADM 서비스 인터페이스를 제공한다. 또한 제어작동기 매니저(Control Actuator Manager)는 CADM 모듈이 시작될 때 cadmActuatorTable에 설정되어 있는 제어장치에 대한 초기화 설정 정보를 이용하여 드



(그림 5) CADM 내부 구조

```

snmpCadmMIB (snmpv3Modules 1)
+ cadmMIBObjects (snmpCadmMIB 1)
+ cadmControlOIDTable (cadmMIBObjects 1)
| + cadmControlOIDEntry (cadmControlOIDTable 1)
| | + cadmControlName (cadmControlOIDEntry 1)
| | + cadmControlOID (cadmControlOIDEntry 2)
| | + cadmControlActuatorName (cadmControlOIDEntry 3)
|
+ cadmActuatorTable (cadmMIBObjects 2)
+ cadmActuatorEntry (cadmActuatorTable 1)
+ cadmActuatorName (cadmActuatorEntry 1)
+ cadmActuatorType (cadmActuatorEntry 2)
+ cadmActuatorInitStr (cadmActuatorEntry 3)
    
```

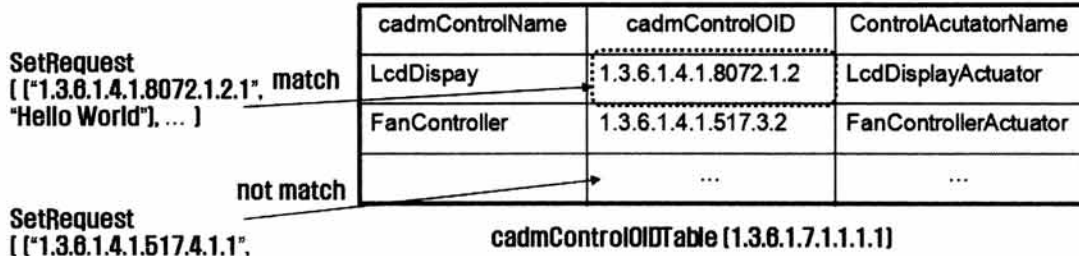
(그림 6) snmpCadmMIB의 자식노드 트리구조

라이버들을 초기화하고 관리하며, SNMP 어플리케이션의 제어장비에 대한 사용 요청을 해당 제어작동기 매니저에 전달한다.

CADM에서는 제어 장치에 대한 서비스를 제공하기 위하여 (그림 6)과 같은 snmpCadmMIB를 정의한다. snmpCadmMIB는 OID가 1.3.6.1.7.1인 snmpv3Modules 노드의 첫 번째 자식노드로 정의되고, 제어 OID 정보, 작동기 드라이버 정보를 포함하는 cadmControlOIDTable과 cadmActuatorTable이 정의되어 있다.

4.2.1 cadmControlOIDTable (OID: 1.3.6.1.7.1.1.1)

CadmControlOIDTable은 장비 제어관련 OID에 대한 정보들을 저장하는 MIB이다. cadmControlOIDTable의 cadmControlName 컬럼은 CADM에서의 장비제어명을 저장하고, cadmControlOID 컬럼은 장비제어명에 대한 실제 제어 관련 OID의 경로를 저장한다. cadmControlActuatorName은 제어 관련 OID에 해당하는 실제 작동기 드라이버 이름을 저장한다. CADM에서 cadmControlOID 컬럼에 제어와



(그림 7) OID의 제어 관련성 확인 방법

관련된 OID 경로들을 설정함으로써 확장 SNMP 에이전트는 getActuatorName() 인터페이스를 이용하여 주어진 OID가 제어에 관련된 경로에 위치하는가를 판단하여 제어와 관련된 처리를 수행한다. 제어 에이전트는 SetRequest가 도착하였을 때, (그림 7)과 같이 CADM 제어 인터페이스의 getActuatorName() 인터페이스를 이용하여 명령어에 포함되어 있는 OID가 제어와 관련된 OID인지를 cadmControlOIDTable에서 검사하고, 제어와 관련된 OID일 경우 제어처리를 담당할 해당 cadmControlAcuatorName을 돌려보내 DDS에서 명령을 처리하게 한다. 따라서 설정을 위한 본래 SNMP set 명령어와 제어를 위한 set 명령어 활용의 모호함을 해결할 수 있다.

4.2.2 cadmActuatorTable (OID: 1.3.6.1.7.1.1.1.2)

cadmActuatorTable은 작동기 드라이버에 대한 정보를 저장하는 MIB로서, 드라이버 타입과 초기 설정 값을 가지고 있는 테이블이다. 제어작동기 매니저는 cadmAcuatorTable 정보를 바탕으로 작동기 드라이버를 초기화하여 작동기로 명령어를 전달하기 위한 준비를 하고, cadmActuatorName을 관리하여 제어관련 OID에 대한 명령어를 실제 작동기 드라이버에 전달한다. (그림 8)는 제어작동기 매니저가 cadmActuatorTable 정보를 바탕으로 제어 장치에 대한 드라이버를 초기화하고 제어 명령이 전달되는 과정을 보여주고 있다. cadmActuatorName으로 식별되는 각각의 드라이

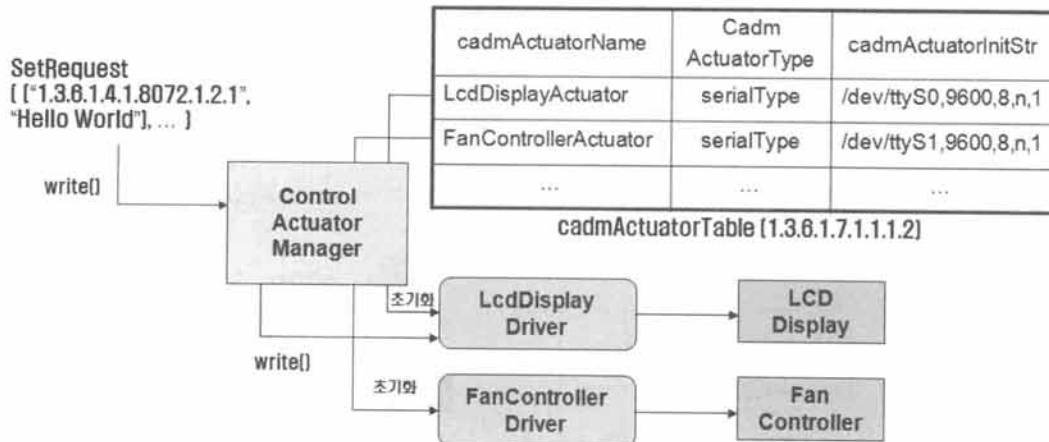
버들은 제어 장치의 종류와 초기화하는 방법에 대한 문자열을 가지고 있는데, 이를 바탕으로 CADM이 처음 작동될 때 제어장치 드라이버들을 초기화하고 드라이버들을 관리하면서 SNMP 어플리케이션으로부터의 요청에 따라 장치로의 입출력과 관련된 처리를 수행한다.

4.2.3 CADM 서비스 인터페이스

CADM에서 SNMP 어플리케이션에 제공하는 서비스 API는 (그림 9)와 같이 getActuatorName(), read(), write()이 있다. getActuatorName() 메소드는 주어진 OID가 제어와 관련된 처리를 해야 하는지를 확인하는 메소드로, OID가 장치 제어를 위한 OID이면 장치로 입출력할 수 있도록 작동기 이름을 보내고, 아니면 null을 보낸다. read() 메소드는 작동기로부터 최대 길이가 size인 데이터를 읽어서 버퍼에 저장한다. write() 메소드는 버퍼의 데이터를 최대길이 size 길이만큼 읽어서 작동기에 기록함으로써 제어처리를 하게 한다.

```
string getActuatorName( OctetString OID )
int read( String actuatorName, char[] data, int size )
int write( String actuatorName, char[] data, int size )
```

(그림 9) CADM 서비스 API 명칭



(그림 8) CADM의 초기화와 처리 흐름도

4.3 SNMP 명령어 추가

기존의 SNMP 명령어와 기능은 동일하면서 대용량 메시지를 처리할 수 있도록 새로 추가한 3개의 SNMP 명령어는 snmpV3e-set-request, snmpV3e-response, snmpV3e-inform-request이다. <표 1>은 새롭게 추가된 명령어에 해당하는 메시지의 식별자를 보여준다.

새로 추가되는 세 개의 명령어에 대한 PDU 정의는 RFC 3416에 근간하여 확장하였다 [17]. (그림 10)은 새로 추가된 snmpV3e-set-request, snmpV3e-response, snmpV3e-inform-request를 위하여 ASN.1 형식으로 기술된 PDU 정의, 즉 SNMPv3e-SetRequest-PDU, SNMPv3e-Response-PDU, SNMPv3e-InformRequest-PDU를 보여주고 있다.

(그림 11)은 추가된 PDU에 대한 실제 PDU 타입 정의를 나타내고 있다. 새롭게 추가된 snmpV3e-set-request, snmpV3e-response, snmpV3e-inform-request 명령어는 그 기능이나 형식이 기존의 set-request, response, inform-request 명령과 동일하기 때문에 동일한 본래의 PDU 타입을 그대로 사용하도록 정의하였다. 그러므로 추가되는 명령어는 (그림 4) SNMP 엔진의 접근제어와 보안 기능을 그대로 활용할 수 있다.

SNMP 엔진의 디스패처는 네트워크에서 전달받은 SNMP 메시지를 메시지 처리 서브시스템, 보안 서브 시스템, 접근 제어 서브 시스템을 사용하여 해석해서 PDU로 만들고, 이를 적절한 SNMP 어플리케이션에 전달하는 역할을 수행한다. 메시지 처리 서브시스템은 메시지를 SNMP 해당 버전에 따라 해석하는 역할을 담당한다. 따라서 메시지 처리 서브시스템에는 SNMPv1, SNMPv2, SNMPv3c, SNMPv3에

<표 1> 추가된 SNMP 메시지와 식별자

SNMP Message	Identifier (Hex)
SNMPv3e-SetRequest-PDU	A9
SNMPv3e-Response-PDU	AA
SNMPv3e-InformRequest-PDU	AB

```

PDU ::= CHOICE {
  get-request          GetRequest-PDU,
  get-next-request    GetNextRequest-PDU,
  get-bulk-request    GetBulkRequest-PDU,
  response            Response-PDU,
  set-request         SetRequest-PDU,
  inform-request     InformRequest-PDU,
  snmpV2-trap        SNMPv2-Trap-PDU,
  report              Report-PDU,

  snmpV3e-set-request  SNMPv3e-SetRequest-PDU,
  snmpV3e-response    SNMPv3e-Response-PDU,
  snmpV3e-inform-request  SNMPv3e-InformRequest-PDU
}
    
```

(그림 10) 추가된 명령어의 PDU를 정의하는 ASN.1 구문

```

GetRequest-PDU ::= [0] IMPLICIT PDU
GetNextRequest-PDU ::= [1] IMPLICIT PDU
Response-PDU ::= [2] IMPLICIT PDU
SetRequest-PDU ::= [3] IMPLICIT PDU
-- [4] is obsolete
GetBulkRequest-PDU ::= [5] IMPLICIT BulkPDU
InformRequest-PDU ::= [6] IMPLICIT PDU
SNMPv2-Trap-PDU ::= [7] IMPLICIT PDU
Report-PDU ::= [8] IMPLICIT PDU
SNMPv3-SetRequest-PDU ::= [9] IMPLICIT PDU
SNMPv3-Response-PDU ::= [10] IMPLICIT PDU
SNMPv3-InformResponse-PDU ::= [11] IMPLICIT PDU
    
```

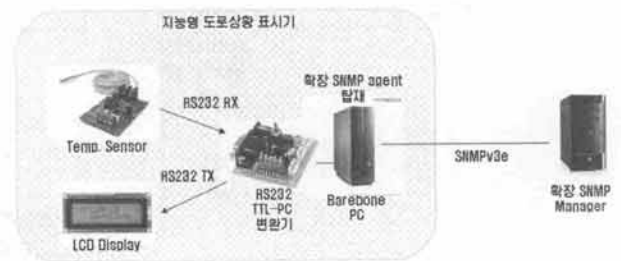
(그림 11) 추가된 명령어의 PDU 타입을 정의하는 ASN.1 구문

해당하는 각각의 메시지 모델이 있다. 추가된 세 개의 명령어는 SNMPv3e라는 새로운 메시지 처리 모델로 새롭게 추가된 후에 디스패처에서 SNMP 메시지 헤더에 기록되어 있는 <표 1>의 명령어 식별자를 분석하여 SNMPv3e 메시지 처리 모델을 호출할 수 있게 된다. 디스패처는 또한 새롭게 추가된 세 개의 명령어에 대하여 메시지 제약을 해제하기 위하여 디스패처내의 transport mapping에서 수행하는 메시지 제약을 해제 하여 대용량 메시지를 전송할 있도록 하였다.

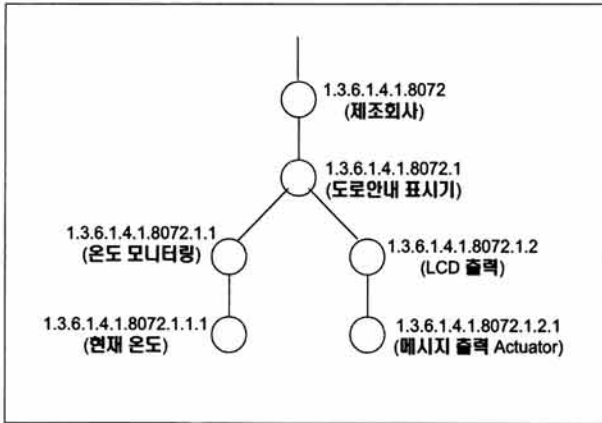
5. 실험 및 성능 평가

4장에서 설계한 확장 SNMP를 구현하여 온도정보 수집기능과 문자 출력 기능을 가지고 있는 도로상황 표시기에 설치하고 제어 및 관리 기능을 실험하였다. (그림 12)와 같이 지능형 도로상황 표시기에는 확장 SNMP 에이전트가 설치되어 있고, 확장 SNMP 매니저를 사용하는 원격 관리자가 제공하는 다양한 정보를 출력하거나 현재의 온도를 모니터링하는 작업을 수행한다. 그러나 온도 센서에 의해서 노면의 온도가 기준 이상으로 올라가면, 도로상황 표시기의 SNMP 에이전트는 그 상황을 인식하여, 원격 관리자의 개입없이 자동으로 LCD 디스플레이에 경고 메시지를 표시하는 상황인지 SNMP 어플리케이션을 탑재하고 있다.

확장 SNMP의 구현을 위하여 공개 소프트웨어인 SNMP4J 1.9.1g를 확장하였다. 확장 SNMP 에이전트와 확장 SNMP 매니저는 Intel P4 프로세서에 Windows Vista 운영체제가 작동되는 PC 환경에서 구동되었으며, JVM은 Sun의 JRE 6 Update 2를 사용하였다. 사용한 온도센서는 원칩 솔루션



(그림 12) 확장 SNMP로 관리되는 지능형 도로상황표시기



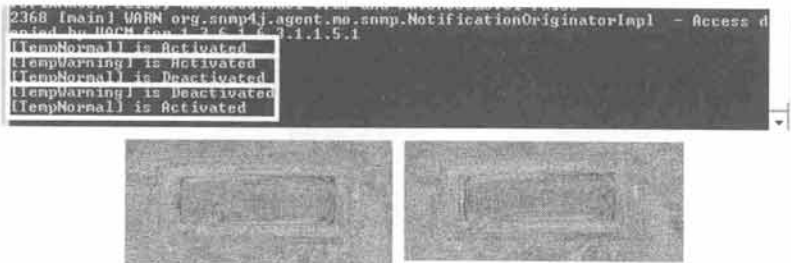
(그림 13) 도로상황 표시기를 위한 MIB 구조

(Onechip Solutions)사의 ST-052TM/RT 온도센서보드를 사용하였고, 제어용 LCD 디스플레이는 콤파일(Comfile)사의 CLCD216-G 보드를 사용하였다. 두 보드를 PC에 연결하기 위하여 콤파일사의 RS232 변환기 보드를 사용하였다. 도로 상황표시기를 위한 MIB는 (그림 13)과 같다.

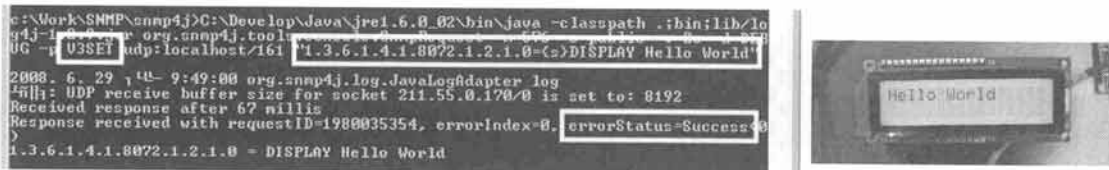
PC에 확장 SNMP 에이전트가 실행될 때, CADM은 cadmActuatorTable에 등록되어 있는 지능형 도로상황표시

기 작동기 드라이버의 cadmActuatorInitStr을 읽어서 작동기 드라이버를 초기화시킨다. 지능형 도로상황표시기를 위한 CADM 작동기 드라이버는 온도센서로부터 온도를 0.5초 간격으로 읽어 도로상황표시기를 위한 MIB의 현재온도 OID(1.3.6.1.4.1.8072.1.1.1) 값을 갱신한다. 한편 확장 SNMP 에이전트는 지능형 도로표시기를 위한 상황인지 SNMP 어플리케이션을 구동시키고, 어플리케이션은 현재온도 OID 값을 모니터링하여, 31°C 미만이면 CADM의 write() 인터페이스를 사용하여 LCDDisplayActuator (OID: 1.3.6.1.4.1.8072.1.2.1)에 "Normal" 문자열 쓰고, 31°C이상이면 "Warning" 문자열을 기록한다. (그림 14)는 노면 온도에 따른 상황을 표시하는 상황인지 SNMP 어플리케이션의 작동 예를 보여준다.

한편, 원격 관리자는 확장 SNMP 매니저를 사용하여 도로상황 표시기를 관리할 수 있다. (그림 15)는 원격관리자가 도로상황 표시기의 LCD 디스플레이를 제어하는 예로써 원격 관리자는 확장된 set 명령어를 사용하여 도로상황 표시기의 LCD 디스플레이에 "Hello World"을 출력한다. 그리고 확장 SNMP는 명령어 길이의 제약사항을 제거하였기 때문에 용량이 큰 XML 문서도 전달할 수 있다. (그림 16)은 get 명령어로 현재 온도 oid에 대하여 그 값을 요청했을 때, 확장된 response 명령어를 이용하여 도로상황 표시기가 수



(그림 14) 도로상황 표시기를 위한 상황인지 SNMP 어플리케이션 작동의 예



(그림 15) 확장된 set 명령어를 사용한 도로상황표시기 제어 예



(그림 16) get, snmpV3e-response 명령어를 사용한 대용량 데이터 가져오기 예

집한 온도를 XML 형태로 응답받는 예를 보여주고 있다.

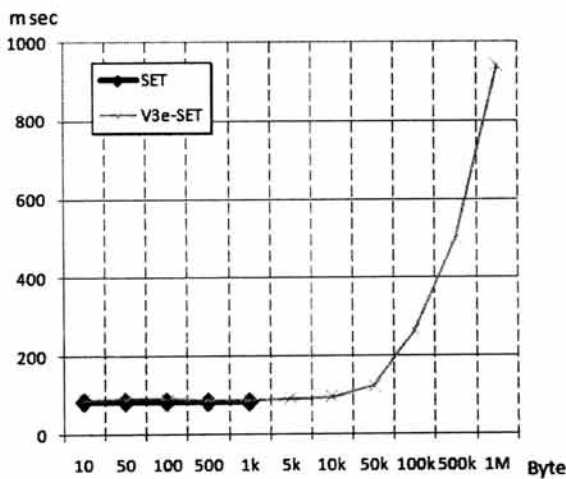
다음의 (그림 17)은 구현한 확장 SNMP의 성능을 나타내고 있다. 성능평가를 위하여 확장하기 전 SNMP4J의 에이전트와 SNMPRequest라는 매니저 프로그램을 이용하여 일반 set 명령어와 get-response 명령어의 응답시간을 측정하였고, 확장 SNMP 에이전트와 확장 매니저의 확장 set 명령어와 get-확장 response 명령어의 응답시간을 측정하여 비교하였다. 실험환경은 위의 도로상황표시기의 실험환경과 동일하고 사용한 SNMP 버전은 SNMPv3이므로 보안과 접근 제어에 대한 처리가 수반된다. UDP의 데이터그램은 메시지 제약이 있으므로 메시지 길이 제약여부에 대한 실험을 위하여 에이전트들은 TCP 모드로 작동시켜 실험하였으며, TCP 모드는 대용량 메시지 전송에 있어서 메시지 순서와 신뢰성을 보장할 수 있다는 장점이 있다. (그림 17)의 a와 <표 2>는 일반 set 명령과 확장 set 명령의 응답시간을 나타내고 있으며, (그림 17)의 b와 <표 3>은 일반 get-response 명령과 확장 SNMP에서 get-확장 response 명령의 응답시간을 나타내고 있다. 모든 실험의 결과 값은 10번씩 측정 한 값의 평균값을 사용하였다.

(그림 17)과 <표 2>, <표 3>을 통하여 확장하기 전과 확장 후 set, get 명령어의 응답시간의 차이는 약 5ms 이내로

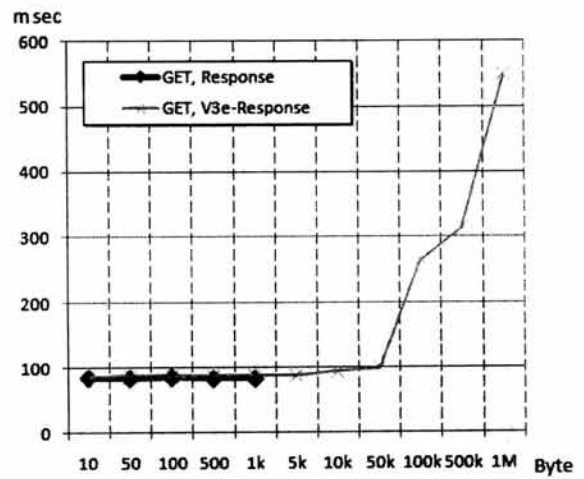
확장 SNMP의 응답시간이 확장 전 SNMP와 거의 유사함을 알 수 있다. 또한 확장 전 SNMP는 메시지 제약이 있으므로 5 Kbytes 이상의 데이터를 전송할 수 없으며, 확장 SNMP는 메시지 제약을 해결한 확장 set, 확장 response 명령을 사용하였으므로 5 Kbytes 이상의 데이터도 전송할 수 있음을 알 수 있다. 전송 메시지의 크기가 50 Kbytes 보다 작을 때는 확장 set 및 get-확장 Response 명령의 응답시간이 메시지 크기에 거의 영향을 받지 않지만, 그 이상되면 수행시간이 증가됨을 알 수 있다. 그 이유는 확장 SNMP의 메시지를 전송할 때 사용하는 버퍼의 크기가 512 Kbytes로 설정되어 있기 때문에 버퍼링 시간의 증가로 응답시간이 증가하게 되는 것이다.

6. 결론 및 향후 과제

네트워크 표준 관리 프로토콜인 SNMP는 확장성, 강력한 정보관리, 인증, 암호화, 그리고 접근 제어와 같은 많은 우수한 장점을 가지고 인터넷 기반의 장비들을 관리할 수 있다. 하지만 SNMP는 장비들을 설정하고 모니터링하는데 적합한 프로토콜이기 때문에, 상황적응형 서비스를 제공하기 위한 디지털 컨버전스 장비를 관리하기에는 제어기능이 취약하



a. SET vs. 확장 SET



b. GET-Response vs. GET-확장Response

(그림 17) 메시지 크기별 응답시간

<표 2> SET vs. 확장 SET 명령어의 응답시간

(단위: msec)

	10	50	100	500	1k	5k	10k	50k	100k	500k	1M
SET	85.2	85.2	85.2	83.9	84.4	-	-	-	-	-	-
V3e-SET	88.1	91.2	90	88.5	88.9	91.7	96.3	123.5	266.1	496.8	939.8

<표 3> GET-Response vs. Get-확장Response 명령어의 응답시간

(단위: msec)

	10	50	100	500	1k	5k	10k	50k	100k	500k	1M
GET, Response	84.6	84.1	85.6	83.3	83.6	-	-	-	-	-	-
GET, V3e-Response	86.1	88.2	86.9	87.5	87.7	88	94.1	100	265.1	314.3	548.1

다. 또한 SNMP는 통신 메시지 크기의 제약으로 용량이 큰 데이터를 전송하기에 적합하지 않다.

본 논문에서 제안하는 확장된 SNMP는 다양한 제어장치를 제어할 수 있도록, 디바이스 드라이버 서브시스템(DDS)을 추가하여 SNMP의 구조적인 단점을 개선하였고, CADM(Control Agent Driver Model)을 정의하여 장치를 제어하기 위한 추상화된 인터페이스를 제공하였다. 따라서 기존의 SNMP 장비가 내재하고 있던 설정과 제어 구분의 모호성을 해결하고, 다양한 SNMP 어플리케이션들이 장비 제어를 쉽게 접근할 수 있도록 하였다. 또한 SNMP 통신메시지의 길이 제약을 제거한 `snmpV3e-set-request`, `snmpV3e-response`, `snmpV3e-inform-request` 명령어들을 추가로 정의하여 XML 문서와 같은 대용량 데이터 처리를 가능하게 함으로써 상향인지 어플리케이션 개발을 용이하게 하였다.

향후에 확장된 SNMP를 사용하는 엔티티가 실제로 장비에서 유용하게 적용되기 위해서는 장비용 MIB를 근거로 한 소스코드 템플릿을 제공하는 것이 필요하다. 따라서 템플릿 안에 장비를 위하여 최적화된 코드를 입력하고, 장비용 개발 도구로 크로스 컴파일하여 장비에 쉽게 포팅되도록 하는 일련의 개발도구를 개발할 계획이다. 특히 새로 추가된 제어 명령어의 경우, 제어 명령이 전달될 작동기별로 소스코드 템플릿을 생성하여 개발자가 쉽게 장비에 확장 SNMP 엔티티를 포팅할 수 있도록 지원할 예정이다.

참 고 문 헌

- [1] "USN기반 핵심 응용 서비스 기술 개발 보고서", ETRI, 정통부, 2006년 2월.
- [2] "USN 서비스 연동기반 규격제시 및 u-NIC 도입방안 연구 보고서", 한국 인터넷 진흥원, 2006년 12월.
- [3] 나경원, 민상원, 김복기, "지능형교통시스템(ITS)의 CCTV 제어용 SNMP 적용 사례연구", 한국ITS학회논문지, 2002, Vol.1 No.1.
- [4] David Zeltserman, *A Practical Guide to SNMPv3 and Network Management*, Prentice Hall, 1999.
- [5] Douglas R. Mauro, Kevin J. Schmidt, *Essential SNMP*, O'Reilly, Jul., 2001.
- [6] D. Harrington, R. Presuhn, B. Wijnen, "An Architecture for Describing Simple Network Management Protocol (SNMP) Management Frameworks", RFC 3411, Dec., 2002.
- [7] "CIAGENT: Concepts and Walkthrough," SNMP Research International Inc., Jan., 2007.
- [8] "Room Alert 4E User's Guide & Reference Manual," AVTECH Software Inc., Feb., 2008.
- [9] "Net-SNMP Release 5.3.2.3: Agent manual pages," Networks Associates Technology Inc., Feb., 2009.
- [10] Tao Lin, Hai Zhao, Jiyong Wang, Guangkie Han, Jindong Wang, "An Embedded Web Server for Equipments",

ISPAN'04, 2004.

- [11] "Alien Reader Protocol with ITR", Alien Technology Corp., Oct., 2008.
- [12] Micael Kunes, Thilo Sauter, "Fieldbus-Internet Connectivity: The SNMP Approach", IEEE Transactions on Industrial Electronics, Vol.48, No.6, Dec., 2001.
- [13] Balazs Scherer, Csaba Tóth, Tamás Kovács házy, Balázs Vargha, "SNMP-based Approach to Scalable Smart Transducer Networks", IMTC 2003, May, 2003.
- [14] Jin-Shyan Lee, Pau-Lo Hsu, "Design and Implementation of the SNMP Agents for Remote Monitoring and Control via UML and Petri Nets", IEEE Transactions on Control Systems Technology, Vol.12, No.2, Mar., 2004.
- [15] Christopher A. Kent, Jeffrey C. Mogul, "Fragmentation Considered Harmful", SIGCOMM '87, Vol.17, No.5, Oct., 1987.
- [16] John Heffner, Matt Mathis, Ben Chandler, "Fragmentation Considered Very Harmful", IETF 66, Jul., 2006.
- [17] R. Presuhn, J. Case, K. McCloghrie, M. Rose, S. Waldbusser, "Version 2 of the Protocol Operations for the Simple Network Management Protocol (SNMP)", RFC 3416, Dec., 2002.



허 길

e-mail : semylab@paran.com

1995년 단국대학교 전자계산학과(학사)
 1997년 숭실대학교 컴퓨터학과(공학석사)
 2008년 숭실대학교 컴퓨터학과(공학박사)
 1996년~1998년 (주)세기통신 개발부 팀장
 2000년~2004년 (주)네오피스 개발팀 연구원

2004년~2006년 아주대학교 유비쿼터스시스템연구센터 책임연구원
 2006년~현 재 (주)브레क्स 기술연구소 책임연구원
 관심분야: 임베디드 시스템, 상황인식, GIS



김 은 회

e-mail : ehkim@ss.ssu.ac.kr

1989년 숭실대학교 전자계산학과(학사)
 1996년 숭실대학교 컴퓨터학과(공학석사)
 2006년 숭실대학교 컴퓨터학과(공학박사)
 2008년~현 재 숭실대학교 지능형로봇연구소 전임연구원

관심분야: 병렬/분산처리, 유비쿼터스 컴퓨팅, RFID



최재영

e-mail : choi@ssu.ac.kr

1984년 서울대학교 제어계측공학과(학사)

1986년 미국 남가주대학교 컴퓨터공학(석사)

1991년 미국 코넬대학교 컴퓨터공학(박사)

1992년~1994년 미국 국립 오크리지연구소
연구원

1994년~1995년 미국 테네시 주립대학교 연구교수

2001년~2002년 미국 국립 슈퍼컴퓨팅 응용센터(NCSA) 초빙연
구원

1995년~현 재 숭실대학교 컴퓨터학부 교수

관심분야 : 병렬/분산처리, 고성능컴퓨팅(HPC), 유비쿼터스컴퓨팅