

Embedded System상에서의 효율적 제어를 위한 SNMP 기능 확장

임동화^o, 이성준, 금유환, 안광선
경북대학교 컴퓨터공학과

{dh7107, ggaibi, ch3cooh, gsahn}@knight.ce.knu.ac.kr

An Extension of SNMP function for efficient control in Embedded System

Dong-Hwa Lim^o, Seong-Joon Lee, Yu-Hwan Keum, Kwang-Sun Ahn
Dept of Computer Engineering, Kyung-pook University

요약

인터넷을 기반으로 하는 전산망은 관련기술의 발달로 다양한 기능과 서비스를 제공한다. 이러한 전산망 구성 요소를 효과적으로 감시하고 통제할 수 있는 전산망 관리의 필요성은 높아지고 있다. 본 논문에서는 단순한 네트워크 관리 프로토콜인 SNMP를 임베디드 리눅스 시스템상에서 구현하여 효율적인 네트워크 관리를 위한 추가적인 기능을 제안한다. 본 시스템은 SNMP 에이전트와 SNMP 매니저간의 상호작용을 통해 네트워크 장비들을 효율적으로 관리하고 제어할 수 있어서 효과적인 네트워크 관리가 예상된다.

1. 서론

컴퓨터와 통신 기술의 발전에 따라, 인터넷을 통해 전 세계의 컴퓨터들이 네트워크에 연결되면서 기존의 중앙 집중형 시스템에서 네트워크 중심의 분산처리 시스템으로 전환되고 있다. 현재 대부분의 네트워크 장비들은 컴퓨터와 연결되어 있어서 시스템 관리자들의 역할이 점점 더 중요해지고 있다. 즉, 라우터, 서버, 허브, 그 외 기타장비를 운영하는 것이 중요한 일이 되었다. 이런 상황에서 점차 다양해지고 있는 네트워크를 유지, 통제하기 위해서 효과적으로 네트워크를 관리할 수 있는 시스템의 사용이 필요하게 되었고 연구되어 왔다.

대표적인 네트워크 관리 프로토콜은 SNMP(Simple Network Management Protocol)[1, 2, 3]와 CMIP(Common Management Information Protocol)[4]가 있는데, 단순 네트워크 관리 프로토콜(SNMP)은 전산망 구성 요소에 관한 관리 정보가 논리적으로 원격지 사용자에게 의해 수집되거나 변경될 수 있게 하는 간단한 프로토콜로서, 간단하고 구현이 용이해 많은 업체와 연구소에 의해 실험, 채택되어 네트워크 관리 시스템의 표준이 되었다. 그러나, 그 기능은 제한적이라는 단점을 가지고 있다. 매니저/에이전트 모델을 기반으로 하는 SNMP는 에이전트가 최소의 소프트웨어를 요구하기 때문에 단순한 네트워크 관리 시스템이며 보조적인 기타 기능들은 관리되는 시스템에 있는데 반해, 대부분의 처리능력과 저장 매체는 관리하는 시스템에 있다.

본 논문에서는 위의 문제점을 해결하기 위해 임베디드 리눅스상에서 SNMP를 구현하여 네트워크 관리를 효과적으로 수행할 수 있는 SNMP 기능을 추가하여

제안한다. SNMP 매니저/에이전트 간의 상호작용을 통해 네트워크 장비들을 효율적으로 관리, 제어한다. 본 논문의 구성은 다음과 같다. 2장에서는 임베디드 리눅스와 SNMP 구조를 살펴보고, 3장에서는 임베디드 리눅스 시스템상에서 구현하여 SNMP의 추가 기능을 제안하고, 끝으로 4장에서 결론을 맺고 향후 연구과제를 제시한다.

2. 관련 연구

2.1 임베디드 리눅스

컴퓨터가 대중화되면서 노트북 PC 시장 동반 성장과 더불어 개인 휴대 단말기(PDA) 등의 발전으로 이어져, PDA는 개인 데이터베이스 관리, 인터넷워킹, 음성 및 영상 통신, 정보통신 등의 서비스를 편하고 손쉽게 제공해 준다. 또한, 컴퓨터 기술은 디지털 카메라, 의료 및 산업 원격 조종기 등에 널리 응용되고 있는데, PDA를 포함한 이러한 시스템들이 바로 임베디드 시스템(Embedded System)[5, 6]에 속하며 컴퓨터의 하드웨어와 소프트웨어, 기타 추가적인 기계 장치 등으로 구성되어 특정기능을 수행하도록 설계된 시스템이다.

리눅스는 Free O/S시스템[7]으로서, 그 동안 PC환경 즉, IBM PC상에서 발전되어 왔으나 근래에 이르러 시스템의 핵심이라 할 수 있는 커널을 이기종의 작은 CPU에 이식하는데 성공함으로써 임베디드 시스템의 영역이 확대되고 있다. 리눅스는 유닉스 표준인 POSIX[8]에 따라 구현됐으며, 완전한 멀티태스킹, 공유 라이브러리, 효율적인 메모리관리, 가상 메모리, TCP/IP 네트워킹 등 유닉스 운영체제로 적합하며 리눅스 소스 코드는 GNU General Public License[9]에 따

라 누구나 아무런 대가 없이 이용 가능하며, 안정성과 가격 우월성, 개발용이성 등과 같은 장점을 갖고 있다.

임베디드 리눅스[10]는 낮은 성능의 프로세서와 작은 크기의 메모리를 가진 내장형 시스템용으로 개발된 리눅스이다. 따라서, 다음과 같은 조건을 갖는다.

첫째, 임베디드 장치가 작은 크기의 메모리 밖에 장착할 수 없다는 제약으로 리눅스 자체의 크기와 기능을 경량화, 최소화, 맞춤화되어야 한다. 둘째, 낮은 성능의 프로세서를 사용하는 제약을 극복하기 위해 성능이 최적화 되어야 한다. 일반적인 임베디드 리눅스 개발 환경 구성은 그림 1과 같다.

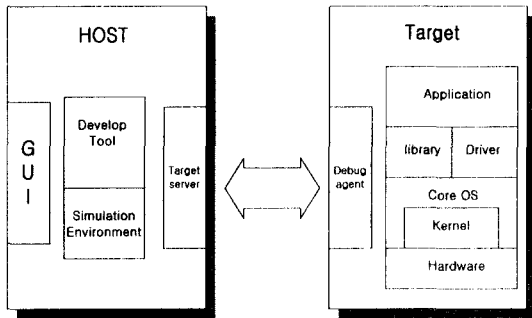


그림 1. 일반적인 임베디드 리눅스 개발 환경 구성

2.2 SNMP

SNMP는 TCP/IP 네트워크에서의 이기종 네트워크를 위한 인터넷 프로토콜이며, 전산망 구성 요소에 관한 관리 정보를 논리적으로 원격지 사용자들에 의해 조사, 변경될 수 있도록 지원하는 간단한 프로토콜이다. SNMP의 구조적인 모델은 그림 2와 같으며, 네트워크 관리 시스템(NMS), SNMP 에이전트, 그리고 전산망 요소(managed Device)들로 이뤄진다. SNMP 에이전트는 전산망 요소들을 감시, 통제하는 네트워크 관리 소프트웨어모듈이며 전산망 요소들의 상태 정보를 갖는다. 또한, SNMP에 적합한 형태로 정보를 바꾸어 준다. 네트워크 관리시스템은 전산망 요소들을 제어하고 모니터링할 수 있는 애플리케이션이며 네트워크 관리에 요구되는 메모리 자원들을 제공한다. 전산망 요소들은 라우터(routers), 브리지(bridges), 허브(hubs), 호스트(hosts) 등과 같은 장비들이 있다.

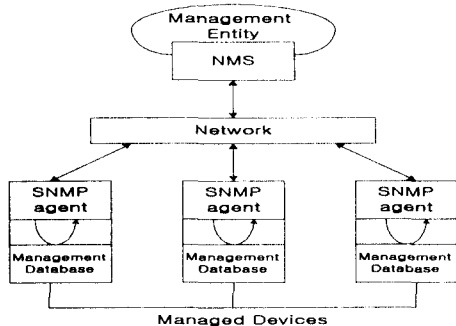


그림 2. SNMP의 구조적 모델

일반적인 SNMP 시스템 구조는 그림 3과 같다. 네트워크 상의 여러 장치들에 대해 비교적 간단한 관리 기능을 수행하는 SNMP는 구조가 단순하여 구현이 용이하다. SNMP의 관리 체계는 관리자가 원격 호스트에 있는 에이전트에게 네트워크 상태에 대해서 물어보고, 그 응답에 대해 원격 호스트의 동작을 제어하도록 한다. 그리고, 관리자와 에이전트 사이에 메시지 전달 형식을 정의하고 예러가 발생하면, 관리자가 그 문제점을 찾아서 해결하는데 사용된다. 명령 체계를 살펴보면, 특정 정보를 요구하는 GetRequest, 특정 정보 요구에 대한 응답으로 GetResponse, 하나 이상의 객체와 그 객체의 값을 요구하는 GetNextRequest, 객체에 값을 할당하기 위한 SetRequest 그리고 예외 상황의 발생을 알려주는 Trap으로 구성된다. UDP/IP를 통해 메시지를 주고받고, 관리자는 통계자료를 결정하기 위해서 MIB 변수로부터 값을 읽는다. 그리고, 관리자는 SNMP 에이전트를 통제하기 위한 값을 MIB 변수에 저장한다. SNMP 메시지는 SNMP 에이전트가 MIB 변수의 어떠한 값을 읽어야 하는지, 저장해야 하는지에 대한 정보를 담고 있다. 에이전트는 관리자의 SNMP 메시지에 지정된 명령을 사용하여 지역(local) 데이터에 맞는 관리 동작을 수행한다. 기본적으로 MIB(Management Information Base)[11, 12]라는 관리 정보 기반을 사용하게 된다. MIB는 가상 정보 저장소라고 볼 수 있는데, 해당 피관리 시스템에 대한 정보 수집의 중요한 자료가 되는 것으로 이를 기반으로 하여 모든 네트워크 관리 시스템들이 개발되고 있다.

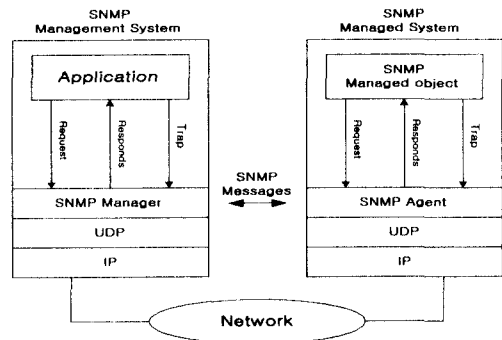


그림 3. SNMP 시스템 구조

3. 추가된 기능을 갖는 SNMP 구조 설계

기존의 Linux기반의 SNMP 에이전트는 관리자의 요구에 따라 응답을 보내는 수동적인 방식으로 동작한다. SNMP는 기본적인 네트워크 하드웨어는 무시할 수 있다. 이것은 운영 소프트웨어가 IP 주소를 사용하므로, 물리적으로 연결되어 있는 네트워크 장비들을 조정할 수 있다. 하지만, 2개 이상의 장비사이에 IP 라우팅이 동작하지 않는다면 상태를 모니터링하거나 재구성하기가 어렵고 접근할 수 없는 단점이 있다.

이 문제를 해결하기 위해 본 논문에서는 임베디드 리눅스 시스템에 SNMP 에이전트를 구현하여 효율적인 네트워크 관리를 위한 SNMP의 추가 기능을 제안한다.

3.1 시스템 구성

본 논문에서 제안한 시스템 구성은 그림 4와 같다. 크게 Embedded Linux System과 Management System으로 구성된다. 세부적으로 살펴보면, 클라이언트 역할을 수행하는 Embedded Linux는 전산망 요소들을 감시하고 통제하여 주기적으로 상태 정보를 Management로 전송하는 SNMP 에이전트와 정보를 저장하는 MIB, 데이터를 전송하는 UDP로 구성된다. 서버 역할을 수행하는 Management는 원격지에 있는 SNMP 에이전트로부터 상태 정보를 물어보고 상태를 감시할 수 있는 SNMP 매니저와 정보를 저장하는 MIB, 클라이언트를 제어하는 Application으로 구성된다. Management는 SNMP 에이전트가 주기적으로 전산망 요소들을 감시, 통제하면서 상태 정보를 보내주기 때문에 클라이언트의 네트워크 상태 정보를 항상 감지할 수 있으며, 필요시에는 클라이언트에게 재전송을 요구하며 부분 제어한다.

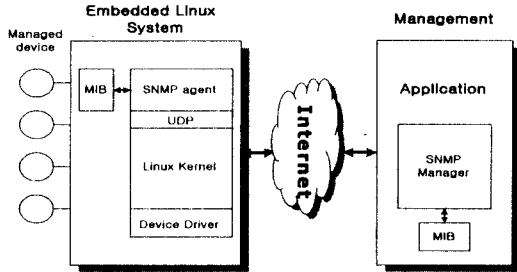


그림 4. 시스템 구성

3.2 동작 과정

본 논문에서 제안한 SNMP 매니저와 SNMP 에이전트 사이의 동작과정을 시퀀스 다이어그램으로 표현하면 그림 5와 같다. SNMP 에이전트는 전산망 요소들의 상태를 감시, 통제하면서 상태 정보를 주기적으로 관리자에게 전송한다. 관리자는 SNMP 에이전트로부터 상태 정보를 받아 클라이언트쪽의 네트워크 상태를 관리하게 되며, 상태 정보 전송이 없을 시에는 SNMP 에이전트에게 상태 정보를 요구하게 되며, 상태 정보가 전송될 때까지 요구한다. 그래도, 전송이 없을 시에는 클라이언트쪽에서 이상이 있는 것으로 판단하여 제어 전송을 명령하게 된다. 제어 명령을 받은 클라이언트는 명령에 따라 실행한다.

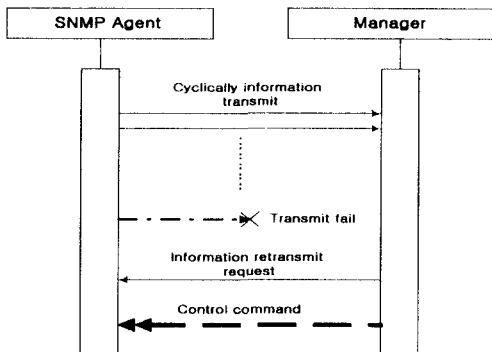


그림 5. 에이전트/관리자 동작과정

4. 결론

네트워크를 이용한 서비스가 다양해지고, 규모가 커짐에 따라 이들 전산망을 효율적으로 관리하기 위한 네트워크 관리 시스템의 필요성이 높아지고 있다. 기존의 Linux기반의 SNMP 에이전트 기능은 관리자의 요청에 따라 응답을 보내는 수동적 방식으로 이뤄짐으로서 효율적인 네트워크 관리측면에서 제한이 있었다.

본 논문에서는 임베디드 리눅스 시스템에 SNMP 에이전트를 구현하여 효율적인 네트워크 관리를 위한 추가적인 SNMP 기능을 제안한다. 안정성과 오픈 소스, 초기 투자 비용이 적은 운영체제인 리눅스를 최적화한 임베디드 리눅스에 SNMP 에이전트를 구성한다. 그리고, SNMP 에이전트는 전산망 요소들의 상태 정보를 Management로 보내고 부분 제어를 받는다. Management는 SNMP 에이전트에게 받은 상태 정보를 보고 전산망 상태를 식별하게 되고 재전송 및 부분 제어를 요구하고 명령한다.

향후 연구 과제는 본 시스템을 구현, 실험하여 실제 구현상의 문제점을 살펴보고 효율적인 네트워크 관리를 위해 추가할 수 있는 제어 기능들에 대해서 연구가 이뤄져야 한다.

참고문헌

- [1] RFC 1157, *A Simple Network Management Protocol(SNMP)*, May. 1990.
- [2] W. Stallings, *SNMP, SNMPv2, SNMPv3 and RMON 1 and 2*, Addison Wesley, 1999.
- [3] NET-SNMP <http://www.net-snmp.org>
- [4] ISO 9596/CCITT X.711, *Common Management Information Protocol - CMIP*, 1991.
- [5] palmpalm <http://www.palmpalm.co.kr>
- [6] Carnegie Mellon University, *Embedded Computer Systems*, <http://www.cs.cmu.edu/~koopan/embedded.html>
- [7] T. Jack and B. Steve, *Using Linux - Fifth Edition*, Macmillan Computer Publishing USA, January 1999.
- [8] R. Card, E. Dumas and F. Mevel, *The Linux Kernel book*, John Willey & Sons Publishers, New York(1998)
- [9] R. Stallman, *Using and Porting GNU CC*, Technical Report, Documentation GNU, November 1995.
- [10] Kelp-Korea Embedded Linux Project <http://www.kelp.or.kr>
- [11] MIB RFC 1213. *An enhancement to MIB 1 for use with network management protocols in TCP/IP based networks*, 1991.
- [12] D.Perkins and E.McGinnis, *Understanding SNMP MIBs*. Prentice Hall, 1997.