

SNMP를 이용한 네트워크 구성관리 시스템의 설계 및 구현

권교혁*, 안성진**, 정진욱*
*성균관대학교 정보통신공학부
**성균관대학교 컴퓨터 교육과
e-mail:khkwon@songgang.skku.ac.kr

Design and Implementation of Network Configuration Management System using SNMP

Kyo-hyeok Kwon*, Sung-Jin Ahn**, Jin-Wook Chung*
*Dept of Information & Communication Engineering,
SungKyunKwan University
**Dept of Computer Education, SungKyunKwan University

요 약

인터넷의 발전과 더불어 네트워크의 규모가 커지면서 효과적인 네트워크 관리의 중요성이 강조되고 있다. 네트워크 관리 분야 중 하나인 구성 관리를 효율적으로 하기 위해서 관리자는 네트워크의 연결 상태에 대한 정보를 얻는 시스템이 필요하다. 이 경우 단순히 연결 상태 정보만이 아니라 네트워크 장치에 대한 추가적인 정보도 함께 제공되어야 한다. 본 논문에서는 SNMP MIB 값 중에서 각 라우터가 가진 라우팅 테이블과 연관된 값을 이용하여 라우터들의 연결 상태를 가시적인 맵의 형태로 표현하고, 이와 함께 네트워크 장치의 구성 요소 정보를 제공하는 방법을 제안하고자 한다.

1. 서론

1990년대로 들어서면서 컴퓨팅 환경 변화로 데이터 네트워크를 관리하기 위해서 어떠한 체계가 네트워크 관리를 수행하는 것이 효과적인 방법으로 자리잡혔다. 이와 같은 필요에 의해서 네트워크 관리의 개념이 발생하였다. 네트워크 관리는 해당 네트워크로부터 자료를 모으고, 자료를 처리하고, 네트워크를 조작하는 기능 등을 포함한다. 네트워크 관리는 각종 시스템 장비의 성능을 보장하고, 네트워크를 구성하는 자원의 효율적 관리를 지원하여 적은 인원으로 최대의 관리지원을 하는 것을 목표로 한다.

OSI에서는 관리자가 효율적으로 네트워크 관리를 수행하기 위해 사용되는 기본 기능을 다섯 기본 기능으로 분류하였는데 그 중 한가지는 구성 관리이다. 구성관리란 시스템의 논리적 및 물리적인 자원의 배치에 관한 데이터를 정의하고 수집하며 감시하며, 데이터의 일반적인 사용을 제어하는 것을 말한다.

다. 이 기능은 시스템을 동작시키고 종료시키는 것과 일반적 동작에 관한 정보 획득을 포함한다.

본 논문에서는 네트워크 관리 중 구성관리를 편리하고 체계적으로 수행할 수 있도록 도와주는 자동화된 구성관리 프로그램을 설계 및 구현하기 위한 알고리즘과 구현방법에 대해 모색하고자 한다.

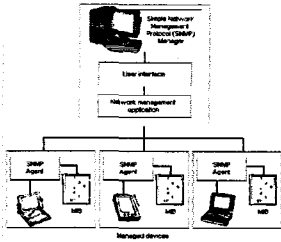
2. 관련 연구

2.1. SNMP

SNMP는 프로토콜, 데이터베이스 정의, 관련된 개념을 포함한 네트워크 관리에 대한 명세서들을 모아 놓은 것을 의미한다. SNMP는 TCP/IP 네트워크를 관리하기 위해 설계되었고, IP 스택 위에 존재하지만 비TCP/IP 장비를 포함한 어떤 네트워크의 관리와 모니터링이 가능하므로 표준이 되었다.

SNMP는 관리국, 에이전트, MIB, 프로토콜로 나누어지며 UDP상에서 작동된다. SNMP를 이용한 네트

워크 관리모델의 구조는 다음 그림과 같다.



[그림 1] SNMP 기반의 네트워크 관리모델의 구조

2.2. MIB

네트워크상에서 관리되는 자원을 오브젝트로 표현한 것을 MIB이라 한다. 각각의 오브젝트는 특별히 관리되는 에이전트 정보의 한 면을 나타내는 데이터 변수이다.

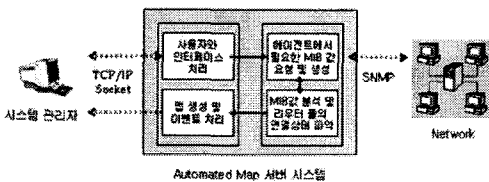
관리국은 MIB 오브젝트의 값을 검색함으로써 감시 기능을 수행하거나, 에이전트에 일어날 수 있는 특정 동작을 일으키고, 특정 변수들의 값을 변경시켜 에이전트의 구성 설정도 변경시킬 수도 있다.

MIB의 오브젝트 형태는 ASN.1의 Object Identifier이다. 여기서 값들은 계층적이기 때문에 이름을 짓는 규약은 오브젝트 형태의 구조를 구별하는 역할을 한다. 오브젝트 식별자는 특정 오브젝트 타입의 유일한 식별자이며, 이 값은 연속적인 정수 형태로 구성된다. 오브젝트 식별자는 트리의 루트로부터 시작하여 그 트리에서 하나의 호 형태로 나타난다.

3. 시스템의 설계

3.1. 전체 시스템의 구성

이 시스템은 크게 사용자와의 인터페이스를 담당하는 웹 인터페이스 부분과 SNMP 에이전트로부터 값을 얻고, 연결상태를 찾아내는 분석 서버 부분으로 나눌 수 있다. 시스템 관리자는 TCP/IP Socket을 통해 분석 서버 시스템의 인터페이스 부분과 통신할 수 있다.



[그림 2] Automated Map 시스템 구조

사용자가 원하는 라우터 주소를 입력하면, 서버 시스템은 해당 라우터에서부터 SNMP를 이용하여 MIB 값을 얻는다. 원하는 모든 MIB값을 얻으면 서버 시스템은 MIB값을 기반으로 라우터간의 연결 상태를 파악하고, 기존 데이터를 맵을 그리기 위한 형태로 변환한다. 마지막으로 요청한 컴퓨터에 맵 구성 정보를 그리는 것으로 작업이 종료된다. 이때 맵 구성 정보뿐만 아니라 장치가 가진 데이터도 함께 전송되어 관리자는 해당 장치를 선택하여 상세 정보를 알 수 있다.

3.2. 시스템에서 사용되는 MIB

이 프로그램에서는 라우터 사이의 연결 정보를 파악하여 네트워크 맵을 형성하고, 각 라우터들에게 필요한 정보를 얻기 위해 SNMP 명령을 이용하여 원하는 라우터의 MIB값을 얻는다.

라우터의 장치 정보를 얻기 위해서 사용되는 MIB 값은 System group의 값을 이용한다. System group 중 이 시스템에서 사용하는 MIB값은 다음과 같다.

해당 MIB	설명
system.sysDescr (1.2)	해당 시스템의 하드웨어나 운영체제와 같은 설정을 나타내는 정보이다.
system.sysName (1.5)	관리자가 설정한 System의 이름을 의미한다.
system.sysService (1.7)	System에서 제공할 수 있는 OSI 계층 서비스를 숫자의 형태로 제공한다.

[표 1] 장치 정보를 알기 위한 MIB값

또한 라우터에서 구성 정보를 파악하기 위해서 각 라우터가 가진 인터페이스 주소와 라우팅 테이블 값을 이용한다. 이러한 MIB값들은 IP group에 있다. IP group 중 이 시스템에서 사용하는 MIB값은 다음과 같다.

해당 MIB	설명
ip.ipAddrTable.ipAdEntntry.ipAdEntfIndex (4.20.1.2)	해당 장치가 가진 인터페이스의 IP를 index의 값과 함께 제공한다.
ip.ipRouteTable.ipRouteEntry.ipRouteNextHop (4.21.1.7)	해당 장치가 가진 라우팅 테이블 중 바로 옆에 인접한 IP 주소를 제공한다.

[표 2] 연결 정보를 알기 위한 MIB값

3.3. 맵 생성 알고리즘

이 논문에서는 맵을 생성하는 방법을 두 가지 방법으로 제안한다. 한 가지 방법은 관리자가 이미 라우터들의 IP 주소 목록을 알고 있어서, 대상 라우터들의 가능한 연결 상태와 라우터 자체의 설정 정보를 알고자 하는 경우이다. 이때 시스템은 모든 라우터들의 주소를 입력받는다. 다른 방법은 관리자가 한 라우터의 IP주소만 알고 있어서 주변 라우터를 검색하는 경우이다. 이때는 시작할 라우터의 IP 주소와 찾기 원하는 라우터의 개수를 입력받는다. 이 경우 관리자에게 찾을 네트워크 장비의 개수를 사용자가 정확하게 하는 이유는, 네트워크 장비들은 연결되어 있기 때문에 계속하여 근접한 장비를 반복하여 찾는 경우 무한한 경우가 발생할 수 있기 때문이다.

첫 번째 경우인 모든 IP주소 목록을 입력받는 경우에는 주변 라우터를 검색할 필요로 하지만, 한 IP로부터 주변을 검색하는 경우에는 프로그램 실행시 MIB값을 이용하여 재귀적으로 주변 라우터를 찾게 된다.

이 방법에서 주위 라우터 검색을 위해 가장 중요한 MIB값은 ipRouteNextHop 값이다. 이 값은 바로 옆에 인접한 라우터의 IP 주소를 의미하는 것이기 때문에, 근접한 라우터에 대한 정보를 얻을 수 있다.

이때 검색 방법은 일반적 그래프 검색 방법 중 하나인 너비 우선 탐색(Breadth-First-Search)을 이용하여 수행한다. 이유는 관리자가 구성 정보를 원하는 경우 인접한 네트워크의 정보를 알고 싶어하기 때문이다. 깊이 우선 탐색을 사용한다면 서버 네트워크 형태가 아닌, 원하지 않은 결과가 나올 가능성이 크다.

이 방법을 구현하려 했을 때, 실행되는 함수 또한 네트워크 장비 검색 방법과 같이 재귀적으로 구성되어야 한다. 이 방법을 구현하기 위해서는 세 가지 종류의 리스트를 필요로 한다. 현재까지 검색 완료 라우터 리스트(A), 테스트 대상 라우터 리스트(B), 테스트 예정 라우터 리스트(C)이다. 검색 완료 라우터 리스트(A)는 모든 탐색이 끝날 때까지 초기화할 필요가 없이 값을 추가한다. 검사 대상 라우터 리스트(B)에 있는 IP주소들에 대한 검색이 끝나면, 다음 순서에 검색을 시도할 리스트(C)가 B의 값으로 변경되고, C는 초기화된다. 이것은 다음과 같은 알고리즘으로 표현할 수 있다.

<재귀 함수전 초기화 부분>

1. 사용자로부터 시작 IP 주소와 생성하려고 하는 네트워크 장비 개수를 입력받는다
2. 시작 IP주소를 테스트 대상 리스트(B)에 넣고, 재귀 함수를 호출한다.

<재귀함수>

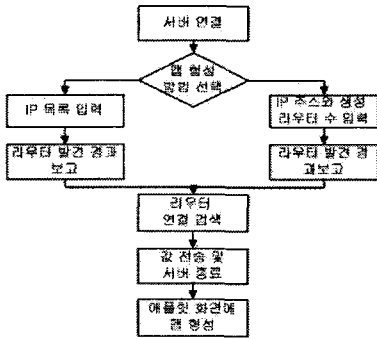
1. 테스트 대상 리스트(B)를 하나씩 2-4번에 실행한다.(i번째)
2. 이번에 검사하는 i번째 IP에 대한 SNMP 명령 실행
3. i번째 IP에 대해 생성된 객체가 SNMP를 지원하는지 검사
 - 3.1. SNMP를 지원하는 경우 : 라우터 정보 장치 객체 생성
 - 3.1.1. 3.1에서 생성된 객체의 인접 IP 주소를 임시 목록(D)으로 만들
 - 3.1.2. 3.1.1에서 만든 임시 목록(D)의 한 IP주소마다, 검색 완료 라우터 리스트(A)의 인터페이스 주소인지 조사한다.(동일 장비 검사)
 - 동일한 장비가 발견되지 않았다면 테스트 예정 라우터 리스트(C)에 추가
 - 3.1.3. 찾은 네트워크 장비 개수에서 한 개 추가
 - 3.1.4. 원하는 개수만큼 장비를 찾았다면 검색 종료
 4. 테스트 대상 리스트(B)에 대해 조사가 끝났으면 검색 종료
 5. 테스트 예정 리스트(C)의 리스트가 비어있는지 조사
 - 5.1. 비어있다면 즉시 라우터 검색 종료
 - 5.2. 비어있지 않다면
 - 5.2.1. 테스트 예정 리스트(C)를 테스트 대상 리스트(B)에 옮긴다.
 - 5.2.2. 테스트 대상 리스트(B)를 초기화한다.
 - 5.2.3. 함수를 다시 재귀적으로 호출한다.

3.4. 연결 정보 탐색

각각의 라우터의 인터페이스 주소(ipAdEntIndex)와 주변 IP주소(ipRouteNextHop)를 통해서 라우터들의 연결 여부를 알 수 있다. 두 라우터 A와 B가 있을 때, A의 ipAdEntIndex 주소 목록 중 하나와 B의 ipRouteNextHop의 주소 목록 중 하나의 값이 동일하다면 두 라우터는 연결되었다고 할 수 있다. 같은 방법으로 A와 B의 ipRouteNextHop, ipAdEntIndex를 비교함으로써, A와 B의 연결 부분의 IP주소를 알 수 있다.

3.5. 프로그램 동작 과정

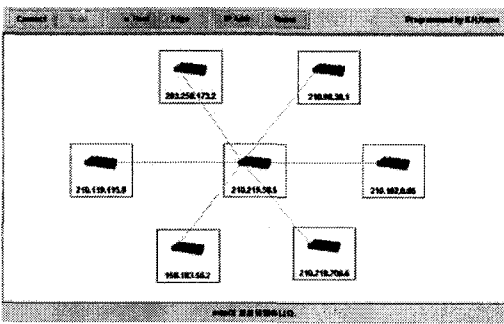
관리자는 자바 애플릿을 통해 시스템 서버 시스템에 접속한다. 관리자는 원하는 맵 형성 방법을 선택하고 정보를 얻기 원하는 라우터 IP주소를 입력하면, 서버는 SNMP 명령을 사용하여 라우터 정보를 구성하면서 관리자에게 진행상태를 계속 보고한다. 원하는 정보 구성이 끝나면 관리자는 라우터 정보를 맵의 형태로 볼 수 있다. 이와 같은 동작 과정을 아래 그림 3과 같은 순서도로 표현 할 수 있다.



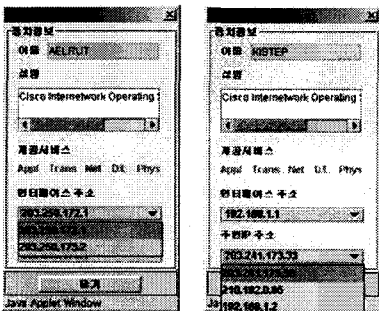
[그림 3] 시스템 동작 과정

4. 구현 결과

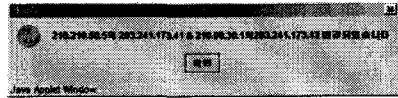
구현 및 테스트 시스템에는 Sun UltraSparc 60 서버의 SunOS 5.7 환경에서 JDK 1.4를 사용하였다. 그림4는 하나의 라우터로부터 시작하여 주위의 다른 라우터를 검색하여 맵을 구성한 경우를 나타낸다. 그림5는 구성된 맵에서 라우터의 상세 정보를 요청했을 때, 보여주는 화면이다. 라우터의 이름과 설명, 세부기능이 표현된다. 그림 6은 두 라우터를 선택하여 라우터 사이의 IP 주소를 나타낸 것이다.



[그림 4] 맵이 구성된 화면



[그림 5] 라우터 정보 보기



[그림 6] 라우터 사이의 연결 정보

5. 결론

보다 효율적인 관리를 위해서 네트워크의 5대 관리 중 구성관리 부분에 초점을 맞추어 개발하였다. 이 시스템은 전체 네트워크의 상황을 체계적으로 파악하게 하는 것으로 관리자에게 네트워크 분석 및 상황 판단을 위한 도구로 사용될 수 있다.

또한 이 시스템은 ipRouteNextHop값을 사용하였기 때문에 SNMP 에이전트가 설치되어 있다면 라우터 뿐만 아니라 일반 호스트와의 연결 상태와 장치 정보도 이 프로그램을 통해서 알아낼 수 있다.

이 프로그램은 이 프로그램 자체로서의 기능보다 다른 네트워크 관리 프로그램의 기반으로 사용될 수 있다. 맵을 형성하는 것으로서 관리하는 네트워크를 이해하는데 도움이 된다. 또한 이러한 이 시스템을 사용함으로써 관리자에게 쉬운 인터페이스 환경을 제공할 수 있다.

향후 과제로는 이미 형성된 맵의 저장과 불러오기 기능과 이 프로그램에서 사용한 두 가지 맵 생성 방법의 장점을 더한 기능을 연구할 필요가 있다. 또한 SNMP를 지원하지 않는 장치라 하더라도 ICMP, Ping, Traceroute, Subnet Mask등을 이용하여 필요한 네트워크 구성 정보를 생성하는 방법에 대한 연구가 필요하다.

참고 문헌

- [1] William Stallings "SNMP, SNMPv2, SNMPv3, and RMON 1 and 2" Addison Wesley
- [2] Mark A. Miller, P.E, "Inside Secrets SNMP Internetworks" 삼각형, 1998
- [3] RFC 1213 (MIB II)
Management Information Base II for network management of TCP/IP-based internets
- [4] RFC 1159 (SNMP)
Simple Network Management Protocol
- [5] Matthew Robinson의 1인 "Swing" 인포북
- [6] Brett Spell "Professional Java Programming" 정보문화사