

XML 기반 SNMP MIB 의 실시간 확장 구조의 설계

°박 용, 차시호, 조국현
광운대학교 컴퓨터과학과 정보통신연구실

Design of XML-based Framework for A Extensible SNMP MIB

°Yong Park, Si-Ho Cha, Kuk-Hyun Cho
Dept. of Computer Science, Kwangwoon Univ.

요약

본 논문에서는 SNMP 기반의 네트워크 관리자가 피관리 요소들에서 MIB 을 구성하는 변수들을 동적으로 선택하기 쉽게 할 수 있는 방법을 제시한다. 즉, 우리는 XML 과 자바 코드를 사용하여 실시간으로 MIB 을 확장함으로써 SNMP 에이전트의 실시간 확장을 용이하게 하는 구조를 설계한다. 우리가 설계한 XML 기반 SNMP MIB 확장 구조는 관리 방법을 채택한다. 실시간으로 MIB 을 기술하기 위해 XML 과 DOM 을 사용 자바 기반의 네트워크와 관리방법을 채택한다. 이 방법은 MIB 이 직렬화 되어 관리자와 에이전트간의 네트워크 상으로 전달될 수 있도록 해준다. 또한 XML 의 사용은 MIB 이 쉽게 탐색 되고 관리 태스크들을 위한 온라인 문서와 자유롭게 통합 될 수 있도록 한다. 더 나아가서 XML 은 다른 정보 모델을 사용하는 네트워크 관리 응용들간의 데이터 교환이 쉽도록 할 것이다.

1. 서론

SNMP 를 사용하는 네트워크 관리 툴과 응용들은 낮은 규모성을 가지며 유연성이 없고 사용이 어려운 경향을 갖는다. 그 이유는 대부분의 SNMP 에이전트를 위한 MIB 이 자체적으로 포함되는 모듈이 에이전트 코드 전반에 흩어져 있기 때문이다. [1,2]종종 네트워크 관리자는 여러 필요한 정보들이 MIB 에서 정의되어 있지 않은 상황에 직면하게 된다. 따라서 필요한 정보를 얻기 위해서는 관리대상 장비로부터 현재의 에이전트를 다운 시키고 필요한 MIB 을 재정의하고 그 MIB 을 위한 새로운 에이전트를 관리대상 장비로 다운 로딩해야 한다는 것이다. 이것이 에이전트의 코드를 다시 컴파일 하지 않고는 이 MIB 을 수정하거나 대체하기 어렵게 만든다. 즉, 실시간으로 필요한 MIB 을 에이전트에 추가하여 에이전트를 확장하기가 어렵다. 우리는 위에서 언급한 MIB 들의 한계를 극복하기 위해 동적으로 재구성 가능하고 확장 가능한 MIB 을 갖는 SNMP 관리를 위한 XML 기반구조를 설계하였다.

2 SNMP

IP 네트워크를 관리하기 위한 표준 네트워크 관리 프로토콜은 SNMP(Simple Network Management Protocol)이다. 이 문에서 의미하듯이 SNMP 는 단순성을 목표로 설계되었기 때문에 쉽게 채택될 수 있다. 또한 SNMP 는 견고성, 피관리 장비의 낮은 오버헤드, 그리고 디버깅의 용이함 등을 위

해 설계되었다.

SNMP 오퍼레이션은 메시지 전송 서비스에 의해 SNMP 메시지를 교환함으로써 발생한다. 메시지 형식은 ASN.1(Abstract Syntax Notation One)의 subset 을 사용하여 정의된다. 이 메시지를 전달하기 위해서는 우선 옥텟 스트림으로 변환해야 한다. 전송 구문(transfer syntax)은 변환된 데이터의 형식을 명세화 한다. SNMP 는 encode 된 메시지들의 형식을 정의하기 위해 BER(Basic Encoding Rules)을 사용한다.

SNMP 에서의 오퍼레이션은 관리 정보 값의 검색, 관리 정보 값의 변경, 그리고 사건(event) 보고로 제한된다. SNMPv1 의 오퍼레이션으로는 Get, Get-Next, Set, Get-Response 와 Trap 이 있으며, SNMPv2 오퍼레이션에는 SNMPv1 오퍼레이션에 GetBulk-Request 와 Inform-Request 가 추가된다.

관리 정보 클래스의 하나의 인스턴스는 SNMP 변수라고 불리며, 각 피관리 노드는 여러 개의 변수들을 가진 것처럼 보여진다. SNMP 오퍼레이션은 이들 변수들의 값을 읽음으로써 피관리 노드를 감시할 수 있고, 이들 변수들의 값을 변경시킴으로써 피관리 노드를 제어할 수 있게 된다.

2.1 MIB(Management Information Base)

SNMP MIB 은 SNMP 에이전트가 액세스를 제공하기 위한 변수들로 알려진 일련의 관리 정보에 대한 정의를 포함 한다[1]. 관리 정보는 피관리 서버 시스템들 내에 존재한다. 몇몇 관리 정보는 관리자에 의해 요청되었을 때만 생성되는 추상적인 개념이다. 다른 관리 정보는 피관리 서버

시스템들의 행위를 초기화하기 위한 인자로서 사용된다. 여기서 서버 시스템이란 한 시스템을 구성하는 프로세스 및 물리적인 구성 요소를 말한다.[6]

SNMP에서는 관리 정보를 변수들과 그 변수들에 대한 값으로 여긴다. 따라서 SNMP 오퍼레이션들은 이들 변수들의 값을 검색하거나 변경함으로써 이루어진다. 이들 관리 정보의 모델, 허용된 데이터 타입, 관리 정보의 클래스 기술 규칙들, 그리고 사건(event)을 위한 모델 및 사건의 클래스 기술 규칙들은 관리 정보 구조(SMI)에 기술되어 있다. 이들 관리 정보들은 ASN.1 언어를 사용하여 정의된다.

각 관리 정보의 클래스는 객체(object) 또는 객체 타입이라 하며, 각 객체 타입은 데이터 타입, 객체에 대한 접근 권한, 할당된 식별자, 인스턴스들의 인덱스, 그리고 그 객체 타입의 의미 등을 포함하여 정의된다.

3 XML

XML은 마크업 언어를 정의하기 위한 언어인 *markup metalanguage* 이다.

DOM이란 HTML과 XML 문서를 위한 API(Application Programming Interface)로써 문서의 논리적 구조와 문서가 접근되고 다루어지는 방법을 정의한다.[4] 이러한 방법으로 DOM은 SGML, XML, HTML 등과 같은 구조적인 문서 내에 정의된 요소들을 각각 하나의 객체로 모델링(modeling)하고 트리(tree)구조의 논리적인 형태로 표현해준다.[3]

XML이 여러 시스템에 저장된 문서라기보다는 데이터로 보여지는 많은 상이한 종류의 정보를 표현하는 방법으로 사용되는 것이 일반화되면서 DOM에서의 문서라는 용어는 광의의 의미로 사용된다. DOM을 사용해 문서를 만들 수 있고, 그 구조를 탐색할 수 있으며, 엘리먼트와 내용을 추가, 수정 또는 삭제할 수 있다. HTML과 XML 문서에서 발견되어지는 어떤 것이든 DOM을 이용해 접근, 변경, 삭제, 추가될 수 있으며 앞으로 확장된 기능을 제공하는 인터페이스를 추가함으로써 보다 풍부한 기능을 정의할 수 있다.

4 XML 기반 네트워크 관리

MIB으로부터 하나의 변수를 제거하기 위해서, 관리자는 에이전트에 그 변수에 대한 삭제 요청을 전달할 수 있어야 한다. 이러한 두 가지 오퍼레이션이 단순성을 위해 SNMP 프로토콜의 구조 내에서 성취되어야 하고, 어떠한 상황에서는 관리자가 다른 방법으로 피관리 요소를 액세스할 수 없기 때문이다. 몇몇 SNMP 에이전트들은 에이전트를 다운 시키고 그 에이전트에 새로운 코드를 컴파일함으로써 MIB에 새로운 변수들이 추가될 수 있도록 해준다 [3]. 이런 해결책은 세련되지 못한 방법이며 모든 환경에서 실행가능하지 않을 것이다. 마스터 에이전트 내에 서버 에이전트들이 동적으로 추가될 수 있는 구조를 갖는 확장 가능한 SNMP 에이전트에 대한 많은 노력들이 있었다 [13]. 이 방법은 새로운 MIB 모듈을 새로운 서버 에이전트로 단

순히 추가함으로써 추가될 수 있도록 해준다. 그러나, 이 방법은 규모성을 갖지 못하는 데 두 가지 이유가 있다. 첫째는 각 서버 에이전트가 별개의 프로세스로서 동작하기 때문이고 둘째는 에이전트 대 에이전트 통신의 오버헤드 때문이다.

XML 기반 SNMP MIB 확장 구조는 아래의 기능을 지원함으로써 네트워크를 관리하는데 세련되고 유연한 방법을 제공한다.

- 피관리 요소의 MIB에 새로운 변수들이 추가될 수 있다. 새로운 변수를 액세스하기 위한 정의와 코드가 관리자에서 에이전트로 다운로드 된다.
- 더 이상 사용되지 않는 변수들은 피관리 요소의 MIB으로부터 제거될 수 있다.
- 각 피관리 요소를 위한 MIB은 관리자가 탐색할 수 있다.
- 피관리 요소의 MIB 내의 변수들은 모니터링을 위해 선택될 수 있다.

4.1 XML 기반 SNMP MIB 확장 구조 MIB

대부분의 SNMP 에이전트와 달리, XML 기반 SNMP MIB 확장 구조 에이전트는 에이전트 자체 코드와 독립적으로 분리된 자신의 MIB에 대한 명확한 실시간 표현을 유지한다. XML 기반 SNMP MIB 확장 구조는 OID 트리의 구조를 반영하는 자바 객체의 트리로써 MIB을 유지한다. XML 기반 SNMP MIB 확장 구조는 자신의 MIB을 OID 트리 구조를 반영하는 XML 문서에 대한 DOM 표현으로써 유지한다. 이 표현은 MIB을 기술하기 위한 XML 문서를 파싱함으로써 시작시에 생성한다. XML 기반 SNMP MIB 확장 구조의 DOM 표현에서의 객체는 자바 객체이다[5].

그림 1은 SNMPv2 MIB을 기술하는 XML 문서를 위한 DTD이다[7]. 피관리 객체는 5개의 필수 요소인 NAME, OID, SYNTAX, MAX-ACCESS, STATUS, DESCRIPTION과 3개의 선택 요소인 REFERENCE, INDEXPART, 그리고 DEFVAL로 구성되는 하나의 SMIV2-OBJECT 요소에 의해 표현된다.

```

<!ELEMENT SMIV2-OBJECT (NAME,
                        OID,
                        SYNTAX,
                        MAX-ACCESS,
                        STATUS,
                        DEFERENCE?,
                        INDEXPART?,
                        DEFVAL?)>
<!ELEMENT NAME (#PCDATA)>
<!ELEMENT OID (#PCDATA)>
<!ELEMENT SYNTAX (#PCDATA)>
<!ELEMENT MAX-ACCESS (not-accessible|accessible-
                        for-notify |read-only|read-
                        write
                        |read-create)>
<!ELEMENT not-accessible EMPTY>
<!ELEMENT accessible-for-notify EMPTY>
    
```

```

<!ELEMENT read-only EMPTY>
<!ELEMENT read-write EMPTY>
<!ELEMENT read-create EMPTY>
<!ELEMENT STATUS (current|deprecated|obsolete)>
<!ELEMENT current EMPTY>
<!ELEMENT deprecated EMPTY>
<!ELEMENT obsolete EMPTY>
<!ELEMENT DESCRIPTION (#PCDATA)>
<!ELEMENT REFERENCE (#PCDATA)>
<!ELEMENT INDEXPART (INDEX|AUGMENTS)>
<!ELEMENT INDEX (IMPLIED?,OID)+>
<!ELEMENT IMPLIED EMPTY>
<!ELEMENT AUGMENTS (OID)>
<!ELEMENT DEFVAL (#PCDATA)>
    
```

그림 1. SMiv2를 위한 DTD

에이전트의 MIB 에 서브 트리를 추가하기 위해, 관리자는 mgmtMIB 와 mgmtMethods 객체에 SET 오퍼레이션을 수행한다. 이 SET 오퍼레이션을 수행하는 SNMP SET PDU의 구조가 그림 2 에 있다. 이 PDU 는 각 프록시 객체를 위한 OID 와 프록시 객체에 설정할 값을 포함하는 스트링을 갖는다.

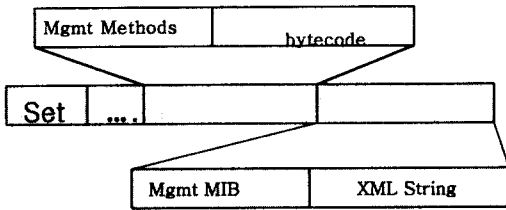


그림 2. SNMP PDU

관리자가 mgmtMIB 에 SET 하는 값은 서브 트리 내의 새로운 객체에 대한 XML 의 표현을 포함하는 스트링이다. 이 스트링은 시작 시에 MIB 을 기술한 XML 문서, 그리고 DOM 표현이 생성되는 것과 같은 방법으로 파싱된다. 그 다음, 새로운 서브 트리의 이 DOM 표현이 MIB 트리에 접목된다.

관리자가 mgmtMethods 객체에 전달하는 값들은 트리에 추가되는 각 종단 노드를 위한 압축된 자바 바이트코드를 포함하는 스트링이다. 이 바이트코드 스트링은 새롭게 추가된 MIB 객체들에 대한 GET 과 SET 오퍼레이션을 수행하기 위한 코드를 포함한다. 새로운 MIB 객체가 생성될 때, 각 노드를 위한 바이트코드는 MIB 객체에 대한 GET 과 SET 을 위한 두 개의 메소드를 포함하는 자바 클래스로써 압축이 해제되고 로딩된다. 그 다음, 이 클래스의 인스턴스는 새로운 MIB 객체를 위해 생성된 노드 내에서 만들어지고 저장된다.

MIB 객체의 삭제는 mgmtMIB 에 대한 SET 오퍼레이션을 수행함으로써 간단히 성취된다. 값으로 전달된 XML 스트링은 간단한 "delete" 명령과 삭제할 노드의 OID 를 포함한다. 하나의 노드에 대한 삭제는 그 노드에서부터 생성된 전체 서브 트리를 제거한 효과를 갖는다.

mgmtMIB 와 mgmtMethods 객체에 대한 GET 과 SET 오퍼레이션을 위한 코드는 MIB 내의 모든 다른 객체를 위한 방법과 같은 방법으로 자체적으로 저장되고 접근된다.

5 결론 및 향후과제

본 논문은 네트워크 관리를 위한 XML 기반의 실시간 확장이 가능한 SNMP 관리 구조에 대한 방법을 제시하였다. XML 기반 SNMP MIB 확장 구조라 불리는 이 구조는 SNMP 프레임워크 내에서의 실시간으로 MIB 을 확장할 수 있다. 즉 XML 의 사용은 MIB 이 네트워크 상에서 전달될 수 있도록 하고, 쉽게 검색되고, 관리 업무를 위한 온라인 문서와 쉽게 통합되도록 해준다. 또한, XML 은 다양한 정보 모델을 사용하는 관리 어플리케이션들 간의 데이터의 상호 교환을 쉽게 해줌으로써 상호 운용성을 유용하게 할 것이다. 따라서 미리 예상치 못한 필요한 관리정보를 실시간으로 확장 시킴으로써 네트워크를 관리하는 데 실시간 확장성과 유연성을 제공 할 수 있다. 관리자는 웹 서버 상에서 동작하는 서블릿으로 구성할 것이며, 새로운 관리자의 서버 기능이 실시간으로 유연하게 추가될 수 있도록 구현할 것이다. 이 XML 기반 네트워크 관리자는 서블릿을 지원하는 어떠한 웹 서버 상에서도 수행될 수 있을 것이다.

향후 과제로는 본 논문에서 설명한 XML 기반의 SNMP 관리 프레임워크를 구현하는 것이며, 구현을 통해 실제로 얼마만큼의 실시간 확장성을 제공하는 지에 대해 조사해야 할 것이다. 또한 네트워크 관리의 관리 영역인 구성 관리, 계정 관리, 보안 관리, 장애 관리, 성능 관리 기능 등에 대한 응용에 대해 계속 연구가 진행되어야 할 것이다.

6 참고문헌

- [1] D.Perkins and E.McGinnis. *Understanding SNMP MIBs*. Prentice Hall, 1997.
- [2] S.Roberts. An Introduction to SNMP MIB Compilers. *The Simple Times*, 2(1), January 1993.
- [3] Frank Boumprey and etc. *PROFESSIONAL XML APPLICATIONS*,1999.
- [4] W3C. The DOM Web page.<http://www.w3c.org/DOM>
- [5] A.Kristensen. Nexus Web Server page.
- [6] William Stallings. *SNMP,SNMPv2,SNMPv3 and RMON 1and 2*, 1999.
- [7] Dan Connolly, W3C. <http://www.w3.org/XML/1998/06/xmlspec-report.htm>,1998
- [8] Jurgen Quittk and Cornelia Kqppler. Remote Service Deployment on Programmable Switches With IETF SNMP Script MIB
- [9] W3C. The XML Web page. <http://www.w3c.org/XML>
- [10] D.Tennenhouse,J.Smith,W.Sincoskie,D.Wetherall and G.Minden .*A survey of Active Network Reseach*.IEEE Communications Magazine 35(1):90-86, January 1997
- [11] D.B.Levi and J. Schonwalder. Definitions of Managed Object for the Delegation of Management Scripts. RFC2592, SNMP Research, TU Braunschweig, May 1999.
- [12] J. Schonwalder and J. Quittek. Secure Management By Delegation within the Internet Management Framework. In Proc. 6th International Symposium on grated Network Management, Boston,May1999.(to appear).
- [13] T. lindholm and F.Yellin. *The Java Virtual Machine Specification*. Addison Wesley,1997.