

XML 기반 네트워크 관리를 위한 SNMP-XML 변환기 및 게이트웨이

(SNMP-XML Translator and Gateway for XML-based Network Management)

주홍택⁺, 윤정혁^{*}, 홍원기⁺
⁺포항공과대학교, ^{*}SK Telecom,
{juht, infobank, jwkhong}@postech.ac.kr

요 약

이 논문은 SNMP MIB 을 XML 로 변환하는 구체적인 알고리즘을 제시하고 이 알고리즘을 적용한 SNMP 와 XML 간의 게이트웨이 개발 결과를 제시한다. 개발한 게이트웨이는 XML 에 기반한 네트워크 관리자 시스템과 SNMP 에 기반한 네트워크 대리자 시스템간의 네트워크 관리 정보 교환을 가능하게하는 시스템이다. SNMP 는 인터넷 관리에서 널리 사용되었지만은 효율성과 확장성에 제약 때문에 날로 증대되는 대규모의 네트워크 전체를 SNMP 만으로 관리하는 것은 비효율적이다. 이에 대한 대안 중의 하나가 XML 에 기초한 XML 기반의 네트워크 관리 방안이다. 우리는 기존의 SNMP 대리자 시스템과 XML 관리자 시스템을 통합하는 방법으로 SNMP 와 XML 간의 게이트웨이를 개발하였다. 우리는 이와 같은 게이트웨이 개발을 위한 Specification Translation 과 Interaction Translation 을 정의하였다. SNMP MIB 을 XML 로 변환하여 주는 변환기는 정의한 Specification Translation 을 자동으로 수행하며 SNMP 와 XML 간 게이트웨이는 정의한 Interaction Translation 을 지원하는 시스템이다. 우리는 이 논문에서 XML 관리자 시스템이 SNMP 대리자 시스템을 통하여 네트워크를 관리하는 방안을 제시한다.

1. 서론

1980년대 후반부터 월드 와이드 웹 (WWW: World Wide Web) [1]의 발달과 함께 크게 증가하기 시작한 인터넷은 여러 벤더들이 만들어 낸 다양한 네트워크 장치들로 이루어진 복잡한 네트워크를 형성하게 되었고 자연스럽게 네트워크에 대한 관리의 중요성도 증가하였다. SNMP (Simple Network Management Protocol) [2]는 인터넷을 효율적으로 쉽게 관리하기 위한 방안으로 만들어진 것인데 초창기에는 과도

기적 표준으로 단기간의 사용을 목적으로 시작되었으나 구현이 쉽고 호환이 뛰어난 장점으로 현재까지도 대부분의 인터넷 관리에 널리 사용되고 있다.

SNMP는 많은 장점에도 불구하고 확장성과 효율성에서 단점을 가지고 있다 [3]. 1990년대를 통해서 인터넷이 폭발적으로 증가하면서 SNMP 기반 네트워크 관리 시스템으로 날로 증가하는 거대한 네트워크를 관리하기에는 한계점이 나타났다. SNMP 기반 네트워크 관

리 시스템의 단점을 보완하여 거대한 네트워크에서 생산되는 많은 관리 데이터를 효율적으로 전달하고 처리하기 위한 방안으로 XML [6]을 네트워크 관리에 이용하고자 하는 연구들이 나타나고 있다. 이러한 연구에서는 웹 기반 네트워크 관리 시스템에서 데이터의 전달이나 처리 및 저장 등의 방법으로 XML을 사용한다. 관리정보를 XML 문서로 표현하고 이 XML 문서를 HTTP 통신을 이용하여 전달하며 [3, 4], 데이터베이스에 저장하거나 사용자 어플리케이션으로 데이터처리를 하는 방법으로 표준화된 XML 처리 방법을 사용한다 [5]. 관리정보를 HTTP 상에서 XML로 전달하고 처리하는 방식은 데이터 전달의 신뢰성과 전달 지연의 감소 등 많은 장점을 가지고 있고, 플랫폼에 관계없이 이기종의 시스템 간에도 쉬운 데이터 교환이 가능하다는 장점이 있다.

XML 기반의 네트워크 관리 시스템은 확장성에서 장점을 가지고 있지만 이미 세계적으로 널리 사용되는 SNMP 에이전트가 장착된 장치를 수용하는 방법을 제시하지 못하면 효용성이 충분하지 못하다. 인터넷 네트워크 관리의 대부분이 SNMP를 기반으로 개발되어 왔기 때문에 대부분의 인터넷 장치는 SNMP 에이전트를 장착하고 이를 통해 네트워크 관리가 이루어지고 있다. 따라서 XML 기반 네트워크 관리의 장점을 이용하면서도 기존의 SNMP 에이전트를 통합해서 관리할 수 있는 방안이 필요하다.

XML 기반 네트워크 관리 시스템에서 기존의 SNMP 에이전트를 변경 없이 통합 관리하기 위해서는 SNMP와 HTTP/XML 간의 프로토콜 변환을 위한 게이트웨이가 필수적이다. 게이트웨이는 매니저가 이해하는 언어인 XML과 에이전트가 이해하는 언어인 SNMP 메시지를 중간에서 상호 번역해 주는 프로그램이다. 게이트웨이의 규격변환 (Specification

Translation)을 위해서는 SNMP MIB (Management Information Base) [17]으로 표현된 SNMP 에이전트의 관리정보를 XML로 변환하는 방법을 제공해야 하고, 상호동작 변환 (Interaction Translation) 방법도 고려해야 한다. 본 논문에서는 SNMP MIB을 정보의 손실이 없이 XML로 변환하는 알고리즘을 제시하고, 이에 따른 자동변환기를 구현한다. 또한, 데이터 변환을 위한 게이트웨이를 구현하여 XML기반의 통합 네트워크 관리 시스템에 적용한다.

서론에 이어 2장에서 관련 연구를 살펴보고, 게이트웨이 구현과 관계된 이전의 연구들과 이들의 한계점을 분석한 다음 이를 해결하기 위한 방안을 설명한다. 3장에서는 게이트웨이 구현을 위한 규격 변환 과정으로 SNMP MIB의 XML 변환 알고리즘을 제시하고 자동 변환기의 구현에 관해 설명한다. 4장에서는 SNMP-XML 변환 게이트웨이의 구현과 동작에 관해 설명한다. 마지막으로 5장에서 결론과 향후 연구에 대하여 기술한다.

2. 관련연구 및 문제해결방법

이 장에서는 먼저 SNMP 기반 네트워크 관리와 XML 기반 네트워크 관리에 대해 알아보고 이들이 가지는 문제점을 분석한 뒤 이를 해결하려는 기존의 연구들과 한계점을 분석한다. 그리고 본 논문에서의 해결 방향을 제시한다.

2.1 SNMP의 문제점

1980년대 후반부터 WWW의 확산에 힘입어 인터넷이 급속히 증가하기 시작했다. 그러나 인터넷 네트워크 관리를 위해 사용된 방법은 ICMP (Internet Control Message Protocol) [9]를 이용하여 종단간 네트워크 연결상태를 파악하는 것이 고작이었다. 네트워크가 복잡해짐에 따라 단순한 동작상태만을 파악하는 것을

넘어서 더 많은 관리를 위한 정보를 필요로 하게 되었다. 또한, 각기 다른 업체에서 만드는 각종 인터넷 장비를 공통적인 방법으로 관리하기 위해서는 표준적인 네트워크 관리 방안이 시급히 필요하게 되었다.

SNMP는 1990년 IETF (Internet Engineering Task Force) [32]에서 일차로 표준화되었고, 1995년에는 단점을 보완하고 기능을 확장한 SNMPv2 (SNMP version 2) [21]가 표준 안으로 제시되었다. 그리고, 1998년에는 이전 버전을 더욱 확장하고 가장 큰 문제점으로 인식되던 보안 문제를 해결하기 위한 새로운 버전으로 SNMPv3 (SNMP version 3) [22]이 제안되었다. 현재 가장 많이 쓰이고 있는 것은 처음 표준화된 SNMP인 SNMPv1이고 일부 SNMPv2가 사용되고 있다.

SNMP의 가장 큰 문제점은 확장성과 효율성이다. 확장성이란 얼마나 많은 에이전트를 하나의 시스템에서 관리할 수 있는가 하는 것이고, 효율성이란 데이터의 전달과 처리 등 시스템의 동작에서 얼마나 빨리 효과적으로 처리가 되는가 하는 것이다. 확장성과 효율성이 문제가 되는 이유는 관리 데이터의 증가에 있다. 과거에 관리할 데이터가 적을 경우에는 간단한 SNMP의 특성이 큰 장점이 되었지만 시간이 지날수록 네트워크는 커짐에 따라서 네트워크로 전달되어야 하고, 매니저와 에이전트에서 처리되어야 하는 관리 데이터의 전체 양이 지속적으로 증가하면서 SNMP 기반으로는 거대한 네트워크를 관리하기에 문제가 있다.

관리 데이터가 증가하는 이유는 인터넷의 성장과 함께 전체적인 관리 대상 시스템이 크게 증가하고, 개별 에이전트에서 조회할 관리 데이터의 양이 크게 증가하고 있기 때문이다. 에이전트가 지원하는 MIB이 증가하고 있으며 네트워크 규모가 증가하면서 MIB 테이블(IP routing table, TCP connection table, accounting

table 등)의 크기가 증가하고 있다. 또한, 단순 모니터링이 아니라 네트워크 계획을 위해서 미리 조회하는 데이터의 양이 증가하고 있다. SNMP기반 관리 구조는 이처럼 무한정 증가하는 관리 데이터의 전달과 처리를 효율적으로 할 수 없다. 그 이유로는 다음과 같은 네 가지의 원인이 있다 [4].

첫째, 네트워크 오버헤드의 증가이다. 네트워크 관리 시스템과 관련된 트래픽은 실제 서비스를 위한 트래픽이 아니므로 사실상 관리를 위한 전체 트래픽이 오버헤드이다. 따라서 네트워크 관리를 위한 트래픽을 최소로 유지하도록 해야 한다. 그러나 SNMP 메커니즘에서는 관리할 데이터량이 증가하면 이에 비례해서 오버헤드도 계속 증가하게 된다. 이는 SNMP의 동작에서 get 방식은 한번의 동작에서 하나의 값만을 조회할 수 있고 많은 양의 데이터를 한꺼번에 조회하는 기능이 없기 때문이다. Get-bulk 동작에서도 'max-repetition'을 지정하여 조회할 데이터의 범위를 지정하는데 이것은 예측을 하는 것이므로 정확하지 못해서 재 시도의 문제를 가진다. 오버헤드 증가는 실제 서비스를 위한 네트워크 트래픽에 영향을 주게 된다.

둘째, 지연시간의 증가이다. 데이터의 전달과 처리에 걸리는 지연시간은 최소한으로 하는 것이 좋다. 지연시간이 증가하면 매니저는 문제를 파악하는데 시간이 많이 걸리고 관리 시스템으로서의 기능을 제대로 수행하지 못할 수도 있다. SNMP에서는 데이터 수집 방법이 폴링을 기반으로 하기 때문에 많은 에이전트를 관리하기 위해서는 에이전트의 수에 비례하여 전체 폴링 시간이 증가하고 또한 많은 데이터를 처리하기 위해서는 에이전트와 서버에서 데이터를 인코딩하고 디코딩하는데 시간이 많이 걸린다.

셋째, 매니저의 처리용량의 제한이다. 처리용량은 하드웨어 자원과 분산처리의 여부에 달

려 있다. CPU, 메모리 등의 하드웨어 자원은 비용과 하드웨어 자체의 성능 제한으로 인해 무한정 늘어날 수 없기 때문에 하드웨어 용량 증가로 매니저 처리 용량을 증가하는 데는 한계가 있다. 또한 SNMP는 매니저 간의 통신이나 계층구조의 매니저 모델을 이용한 분산처리 구조를 명확히 제시하지 못하기 때문에 매니저의 처리 부하를 분산하지 못하므로 매니저의 처리부하를 줄이는데 한계가 있다.

넷째, 매니저의 관리 범위의 한계이다. SNMP는 집중화 된 관리 체계로 인해서 하나의 매니저가 관리할 수 있는 범위가 제한적이다. 모든 에이전트가 보내는 데이터는 매니저가 있는 세그먼트의 한 점에 모이게 되고 이는 병목현상을 일으킨다. 때문에 오버헤드를 가능한 한 감소시키는 것이 필요하고 특히 통신 프로토콜의 효율성을 개선하는 것이 필요하다.

결론적으로, 네트워크의 장비는 계속 증가하고 있으나 비용의 문제와 하드웨어의 제한으로 인해 링크 용량을 무한정 증가하거나, 매니저의 처리 용량을 무한정 증가하는 것은 불가능하다. SNMP기반의 관리 구조에서는 이러한 문제를 해결하는 메커니즘을 가지고 있지 않다. 이는 SNMP의 출발이 간단하고 구현이 쉬운 작은 크기의 에이전트를 만드는 것에서 출발하였고, 임시적으로 사용하고자 했던 표준이 계속 사용되면서 네트워크 증가를 따라가지 못하고 있기 때문이다. SNMP의 초기 버전의 문제점을 해결하기 위해 계속해서 보완이 이루어지고 있지만 진행이 너무 느리게 일어나고 있어서 실제 적용해서 사용하기에는 표준화가 시장을 따라가지 못하는 문제점을 안고 있다.

이러한 SNMP의 문제를 해결하고자 하는 시도로서 XML을 이용한 네트워크 관리 방법이 연구되고 있다.

2.2 네트워크 관리를 위한 XML의 장점

XML [6]은 W3C (World Wide Web Consortium) [1]에서 1998년 웹상에서 문서교환을 위한 표준으로 제정한 메타 마크업 언어로서, 너무 복잡해서 사용이 제한적이던 SGML (Standard Generalized Markup Language) [8]을 간략하게 축소시킨 것이다. 따라서 XML은 HTML [7]처럼 HTTP를 이용한 쉬운 전달의 장점과 SGML처럼 확장성을 갖는 장점을 동시에 가지고 있어서 웹 상에서 구조화된 문서 전송이 가능하도록 설계되었다. XML은 광범위한 문서와 데이터 표현능력을 가지고 있으며 이런 특성으로 인해 XML은 관리정보와 인터페이스의 표준으로서의 커다란 잠재력을 가지고 있다.

기존의 웹 문서 전달 양식으로 널리 사용되는 HTML에 비하여 XML이 가지는 장점을 정리하면 다음과 같다.

첫째, XML은 HTML과는 달리 문서의 내용과 표현방식을 분리한다. 표현방식은 XSL (eXtensible Stylesheet Language) [24]을 이용하여 정의하고 이 스타일시트의 변경을 통해서 동일한 XML문서를 가지고 여러 가지로 다르게 표현을 할 수 있다. 또한 XSL의 데이터 정렬과 필터링 기능을 이용하면 원하는 데이터만 보여줄 수도 있고 다른 형식의 XML문서로 변환이 가능하다.

둘째, 구조화된 문서 정의를 지원한다. XML 문서에서 데이터 구조를 정의하는 것은 DTD와 XML 스키마에 의해서 가능하다.

셋째, 확장된 링크 기능이 있다. XML의 표준 셋은 HTML의 링크를 확장한 Xlink [33]와 문서의 전체 또는 일부분을 지정하기 위한 Xpointer [34]를 제공한다. 이것은 양방향 링크와 다중링크 등 확장된 문서간 링크방법을 제공한다.

넷째, 문서 처리를 위한 표준 API를 제공한다. XML은 SGML과는 달리 XML 문서 자체와 문서 모델을 정의한 DTD에 대한 표준

인터페이스를 지원한다. DOM (Document Object Model) [25]은 W3C 에서 XML 문서와 HTML 문서에 대한 API 를 제정한 것으로 DOM 을 지원하는 모든 프로그램들은 XML 문서에 대한 접근, 추출, 저장이 가능하게 되며 문서를 문서 전체가 아닌 객체들로 나누어 처리하게 된다. SAX (Simple API for XML) [26]는 이벤트 기반의 처리를 지원하는데 XML 문서를 읽으면서 문서의 시작과 끝, 엘리먼트의 시작과 끝에서 이벤트를 발생하여 지정하는 작업을 수행한다.

XML은 특히 네트워크 관리를 위한 정보전달과 처리를 위해서 사용될 경우에도 많은 장점을 가지고 있다. 정보 모델링의 측면에서는 XML의 막강한 모델링 기능을 이용하면 모든 정보를 쉽게 모델링 할 수 있다. 특히 SNMP MIB의 트리 구조 정보는 XML로 변환이 쉽게 이루어 질 수 있다. 정보의 전달 측면에서 일반 웹 문서의 전달과 같이 HTTP를 이용한 전달이 가능하기 때문에 이미 널리 퍼져 있는 HTTP를 지원하는 인터넷 장비와 컴퓨터에서 그대로 사용이 가능하다. 호환성 측면에서 XML은 모든 하드웨어와 소프트웨어 플랫폼에서 지원이 가능하기 때문에 어떤 장비 간에도 정보를 주고 받을 수 있도록 지원한다. 이러한 장점은 미들웨어로서의 기능을 할 수 있다는 것을 말한다. 비용 측면에서 XML은 표준 API를 제공하고 이미 구현된 많은 툴들이 있기 때문에 적은 비용으로 시스템을 개발할 수 있다는 장점이 있다. 또한, 네트워크 관리를 위한 많은 데이터 전달에서 XML은 문서 크기의 제한이 없이 많은 양의 데이터를 한꺼번에 전달할 수 있는 장점이 있다. 이러한 XML의 장점들은 기존의 SNMP 기반 네트워크 관리 시스템이 가지는 확장성과 효율성의 문제를 해결하는 기술로 연구되고 있다.

2.3 XML 기반 네트워크 관리

XML 기반 네트워크 관리란 관리정보를 XML로 정의하고 관리를 위한 데이터의 전달을 XML 문서의 형태로 주고 받으며, 데이터의 처리를 위해서 XML 문서 처리 표준 방법을 사용하는 네트워크 관리 방법을 말한다.

가. 내장형 웹 서버를 이용한 방법

네트워크 장치를 관리하기 위해 많은 장치들은 내장된 웹 서버를 이용하고 있다. 이러한 내장 웹 서버를 장치관리에서 네트워크 관리로 확장해서 이용하려는 연구가 있었다 [3]. 이 연구에서 XML을 이용해서 데이터를 전달하는 방법이 연구되었다. XML 기반의 네트워크관리는 웹 서버를 내장하고 있는 에이전트인 WBM agent를 사용하여 네트워크 장치관리를 한다. 매니저와 에이전트의 통신 모델을 그림 1 [3]과 같이 정의하고 있다.

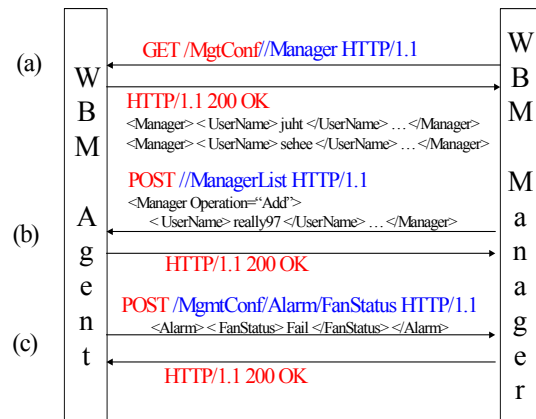


그림 1. XML 기반 관리 통신 모델

그림 1의 (a)는 매니저가 에이전트의 정보를 조회하는 방법으로 매니저의 HTTP GET 요청에 대하여 에이전트가 응답하는 과정으로 이루어진다. (b)는 매니저가 에이전트를 제어하기 위한 방법으로 매니저는 에이전트에게 HTTP POST 메시지 방식으로 필요한 정보를 보내면 에이전트는 이 값을 받아서 그에 해당하는 동작을 한다. (c)는 에이전트에서 발생하는 이벤트가 있을 때 에이전트가 매니저로

보고하는 방식의 통신 방법을 말한다. 이는 각각 SNMP에서 get, set, trap의 동작방법과 유사한 기능을 한다. 여기서 제안하는 통신 모델은 관리정보의 전달 방법으로 HTTP를 사용하여 XML로 된 문서를 전달하는 방안을 제시하고 있다. 이 방법은 이미 인터넷상의 문서전송을 표준 프로토콜을 사용함으로써 네트워크 관리를 위한 새로운 프로토콜을 개발하지 않고도 쉽게 구현할 수 있다는 장점과, 관리정보를 XML로 정의함으로써 앞 절에서 살펴본 데이터 교환의 도구로써 XML이 가지는 장점을 모두 활용할 수 있다는 이점이 있다.

이러한 XML 기반의 네트워크관리 시스템은 기존의 SNMP기반 네트워크 관리 시스템이 가지는 확장성과 효율성의 문제를 해결하는 방법이 되기도 한다.

XML 기반 네트워크 관리 시스템은 많은 장점을 가지고 있다. 그러나 이러한 시스템은 기존의 SNMP 에이전트만이 장착된 장비를 관리하는 방안을 제공하지 않고 있다. SNMP 에이전트는 인터넷 전반에 걸쳐 널리 사용되고 있다. 모든 에이전트를 XML을 지원하는 에이전트인 WBM 에이전트로 변경하는 것은 현실적으로 불가능하다. 따라서 XML 기반 매니저의 장점을 이용하면서 기존의 SNMP 에이전트를 변경하지 않고도 통합해서 관리하는 방법이 제시되어야 한다.

나. XML을 이용한 통합 네트워크 관리

J.P. Martin-Flatin은 웹 기반의 통합 네트워크 관리구조에 관한 그의 연구에서, 데이터 통합을 위한 방법으로 XML을 네트워크 관리에 사용하는 방법을 제시하고 있다 [4]. 더불어, SNMP 에이전트를 관리하기 위해서 SNMP MIB을 XML로 변환하기 위한 모델을 제안하였는데 Model-level mapping과 Metamodel-level mapping을 정의하였다.

Model-level mapping은 DTD가 SNMP MIB에

특정하게 작성되는 방법이다. SNMP 변수명을 그대로 사용하여 XML DTD에서 Element와 Attribute를 작성한다. 이 모델의 작성 예는 다음과 같다.

```
<interface>
<bandwidth>
<type="string">100Mbit/s</bandwidth>
</interface>
```

이 모델로 작성한 XML 문서의 장점은 사람이 읽고 직관적으로 이해하기 좋다는 것이다. 파서(parser)의 작성이 쉽고 문서 검증(validation)의 장점도 있다. 이 모델의 단점은 MIB별로 DTD를 작성해야 하기 때문에 많은 DTD를 필요로 한다는 것이다.

Metamodel-level mapping은 일반화된 하나의 DTD를 정의하여 모든 MIB에 적용하는 방법이다. 이 모델에서는 MIB에 정의된 변수를 사용하여 Element를 명명하지 않고 일반적인 키워드를 사용한다. Model-level mapping에서 보인 간단한 예를 다시 Metamodel-level mapping으로 변환하면 다음과 같이 표현할 수 있다.

```
<class name="interface">
<property name="bandwidth" type="string">
<value>100 Mbit/s</value>
</property>
</class>
```

Martin-Flatin의 연구는 XML을 이용하여 IP 네트워크와 시스템의 웹 기반 통합 관리에 관한 연구로써, HTTP/XML기반 통신의 장점을 서술하고, SNMP MIB의 XML변환에 관한 기본개념을 제시하고 있다.

그러나, SNMP MIB에서 MIB II의 interface 그룹의 XML 변환에 관한 예를 보여주는데 그치고 있어서 SMI의 전체 변환에 관한 구체적인 알고리즘을 제시하지 못하고 있다.

다. XML 기반의 SNMP 데이터 수집

XML을 이용하는 네트워크 관리 데이터 수집

모듈을 개발함으로써 네트워크 관리 시스템의 개발을 쉽게 하고 웹 기반의 네트워크 관리 등에 적용할 수 있도록 하기 위한 연구가 있었다 [5]. 이 연구에서는 SNMP 기반 네트워크 관리에서 기종이 다양한 네트워크장비를 관리해야 할 때 각각의 사설 MIB를 분석하고 서버 구조를 따로 가져가는 개발의 부담을 줄이고자 SNMP의 다양한 형태의 사설 MIB을 XML로 변환시켜주는 변환기를 설계하고 구현하였다.

이 연구에서는 XML DOM을 이용해서 수집한 데이터를 이용해서 DOM을 수정하고 수정된 데이터는 XML 표준 패키지가 제공하는 기능을 이용해서 저장 및 표현을 쉽게 할 수 있는 방법을 제시하였다.

이 연구에서는 SNMP MIB을 XML로 정의하기 위해 DTD를 사용하였다. DTD를 사용하는 방법은 문서의 검증과 관리정보의 정의에서 MIB에 정의된 모든 정보를 표현하는데 한계가 있다. 또한, 구체적인 변환 알고리즘을 기술하지 않고 있고, SNMP의 트랩(trap) 동작에 관한 관리 방법을 포함하지 않고 있다.

라. SNMP MIB의 XML 변환 연구들

SNMP MIB을 XML로 변환하는 방법에 대한 기존의 연구로는 첫째, J.P. Martin-Flatin이 제시한 SNMP MIB을 XML DTD로 매핑하는 예 [4]가 있다. 여기서서는 전체 MIB의 변환에 대한 방법을 제시하지는 않고 단순히 MIB II의 system그룹과 interface그룹에 관한 변환 결과를 예로 제시하고 있다. 둘째, Bell 연구소의 CORBA/SNMP 게이트웨이 개발 프로젝트 [11]에서 연구한 SNMP MIB의 XML 표현에 관한 연구가 있다. 하지만 이 연구에서도 구체적인 알고리즘을 정리하지 않았고 단지 MIB II의 전체 노드에 대한 DTD 정의와 실제 값을 이용한 XML 문서의 예를 보여주고 있다. 셋째, Frank Strauss의 MIB 처리를 위한 라이브러리 'libsmi' [12]가 있다. 이것은 MIB

을 XML, JAVA, C, IDL 등 다른 언어로 변환하도록 하는 라이브러리이다. 이 연구에서 변환 방법은 DTD를 사용한 변환 방법으로 전체 MIB정보를 유지하지 않고 노드이름과 OID만을 저장하여 사용함으로써 실제 표준 변환 방법과는 거리가 있다. 마지막으로 IBM 연구소에서 ASN.1을 XML로 변환하는 연구도 있었다 [13]. 이 연구에서는 ASN.1을 XML로 변환하는 방법을 제공하고 있다. 그러나 SNMP SMI에 정의된 매크로의 변환을 지원하지 않아서 SNMP MIB을 변환하는데 사용하기에는 한계가 있고, 변환방법도 DTD를 이용한 방법이기에 때문에 MIB의 정보가 손실이 된다.

이전의 변환 연구들은 모두 구체적인 변환 알고리즘을 제공하지 못하고 있고, 변환 후에 MIB의 정보가 그대로 XML 표현에 유지되지 않고 정보의 손실이 발생한다. 정보의 손실에서 가장 큰 부분은 노드의 데이터 타입에 관한 것으로 MIB에는 다양한 데이터 타입이 있는데 이것을 지원하지 못하고 단지 모든 값을 문자열로 나타내고 있어서 MIB을 XML로 변환하는 표준적인 방법으로는 한계를 가지고 있다.

2.4 기존 연구의 문제점과 해결방법

기존의 SNMP 기반 네트워크 관리 시스템은 널리 사용은 되고 있지만 관리할 데이터가 많은 거대한 네트워크를 관리하기 위해서는 문제가 있음을 살펴보았다. 이것을 해결하려는 시도로 XML을 이용한 네트워크 관리 시스템이 등장하고 있다.

XML 기반 네트워크 관리 시스템에서 매니저에 해당하는 시스템을 "XML 기반 매니저"라고 정의한다. XML 기반 매니저는 확장성을 가지고 있어서 거대한 네트워크를 관리할 수 있는 방법을 제시하고는 있지만 기존에 널리 활용되는 SNMP 기반 에이전트를 수용해서 관리하는 방안을 제시하지 않고 있다.

본 논문에서 해결하고자 하는 것은 XML 기

반 매니저가 SNMP 에이전트를 변경하지 않고도 관리할 수 있는 방안을 제시하는 것이다. 이를 위해서 SNMP-XML 간 메시지 전달을 중계하는 게이트웨이를 설계하고 구현한다. XML 매니저는 HTTP 상에서 XML 문서로 데이터를 전달하는 방법을 사용하고, SNMP 에이전트는 SNMP 상에서 SNMP 메시지로 데이터를 전달한다. 게이트웨이는 이 사이에서 데이터를 변환하고 중계하는 역할을 한다.

게이트웨이의 설계를 위해서 관리정보의 정의 방법에 대한 변환인 규격변환 (Specification Translation)과 실제 관리 데이터의 전달 변환인 상호동작변환 (Interaction Translation)이 필요하다.

규격변환은 XML 매니저가 SNMP 에이전트로 부터 제공되는 관리 정보를 알기 위한 과정이다. 따라서 SNMP 에이전트가 제공하는 모든 MIB 정보는 게이트웨이에서 모두 변환이 되어야 한다. 규격변환을 위해서는 SNMP 에이전트가 사용하는 관리정보 정의인 SNMP MIB을 XML로 변환해야 한다. SNMP MIB을 XML로 변환하는 기존의 연구들은 구체적인 알고리즘을 정리하여 제시하지 못하고 있고 변환 시에 정보의 손실이 발생하는 문제점을 가지고 있다. 본 논문에서는 SNMP MIB을 XML로 손실 없이 변환하는 알고리즘을 설계하고 자동변환기를 구현한다. 정보 손실이 없는 변환을 위해 DTD를 사용하지 않고 강력한 모델링을 지원하는 XML Schema를 사용하여 정의하도록 하고 기존의 데이터 타입을 모두 지원하도록 고려한다.

상호동작변환을 위해서는 SNMP 메시지와 XML 문서간에 상호 데이터 표현을 변경해주는 게이트웨이의 동작을 규정하고 HTTP와 SNMP 간의 매핑 방법을 설계하고 이를 구현한다.

현재까지는 SNMP와 XML간의 게이트웨이에 관한 연구는 없었으며 이 게이트웨이에 필수

적인 규격변환에 관하여만 다른 필요성으로 연구가 있어왔다. 따라서 SNMP와 XML간의 게이트웨이측면에서 기존연구와의 비교는 불가능하고 규격변화에 대하여는 앞절에서 비교검토하였다.

3. SNMP MIB의 XML 변환

이 장에서는 SNMP MIB의 모든 정보를 손실이 없이 그대로 XML로 변환하는 알고리즘에 대해 알아 보고 이를 이용한 자동 변환기를 구현한다. 이 결과는 게이트웨이의 구현에 필요한 정보 모델 변환을 위해 필요한 과정이다.

3.1 변환 알고리즘

네트워크 관리를 위한 정보 모델의 변환에서 우선적으로 고려할 사항은 변환 후에도 변환전의 모델이 가지는 정보를 그대로 유지하는 것이다. 이를 위해서 변환 전·후의 정보 모델의 구조를 동일하게 유지하도록 하는 것이 중요하다. SNMP MIB의 관리대상 정의는 트리 구조로 구성이 되어 있고, XML 문서도 트리 구조로 정의 될 수 있으므로 문서의 구조를 바꾸지 않고 그대로 트리 구조로 변환을 하는 것이 정보를 쉽게 효율적으로 변환하는 방법이다. 표 1에는 문서 구조의 매핑 방법을 정리하고 있다. SNMP MIB의 각 노드를 XML의 엘리먼트(Element)로 변환하고 엘리먼트의 이름을 노드명으로 한다. MIB 노드의 내부 절(Clause)들은 XML 엘리먼트의 내부에 속성(Attribute)으로 변환한다.

표 1. 문서 구조 변환

SNMP SMI	XML
노드(매크로 정의)	엘리먼트(Element)
노드명	엘리먼트명
노드내의 절	속성(Attribute)

가. 데이터 타입 정의

SNMP MIB에 정의된 데이터 타입을 그대로 XML로 정의 하기 위해서 DTD로는 표현이 불가능하기 때문에 W3C 표준으로 제정된 XML Schema를 이용하였다.

XML Schema를 이용하여 변환을 하면 정수형, 문자열형, 특수한 정의형 등 모든 데이터 타입을 그대로 나타낼 수 있기 때문에 데이터 교환에서 생성된 XML 문서의 검증(Validation)을 강력하게 지원할 수 있고, 이로 인해 이기종의 시스템간에 관리 데이터를 주고 받을 때도 자동으로 XML 파서에서 데이터의 검증이 이루어 지므로 데이터의 무결성을 검사하는 프로그래밍은 추가로 하지 않아도 되는 장점이 있다. SMIV1에서 정의하는 주요 데이터 타입을 XML Schema로 변환은 다음과 같다. 'IpAddress' 타입에서 IP Address 범위는 '[0-255].[0-255].[0-255].[0-255]'으로 정의하였다. 그 외에 숫자형이나 문자열형 등 다양한 타입들은 XML Schema에서 사용할 수는 기본 타입을 그대로 사용하면 정보의 손실이 없이 변환하여 정의할 수 있다. SMIV2에서 도 마찬가지로 이다. 즉 타입들은 XML Schema에서 정의하는 기본 데이터 타입과 일치하는 것이다. 사용자 정의 데이터 타입 변환을 XML Schema로 변환 정의하면 다음과 같다. SNMP MIB에서 정의해서 사용하는 데이터 타입의 경우는 'DisplayString ::= OCTET STRING'과 같이 문자열 대치를 이용해서 정의한다. 이런 경우 DTD로는 정보를 저장해서 XML 파서가 이해하도록 하는 것이 불가능 하지만 XML Schema를 사용하면 새로운 데이터 타입을 정의해서 사용할 수 있다. 사용자 정의형 데이터 타입에서는 열거형, 문자열 길이 지정형, 데이터의 범위 지정형 등 다양한 응용 타입의 정의 방법을 기술하고 있다. 변환은 XML Schema의 문법을 따르고 있다.

나. 매크로 변환

SNMP MIB은 SMI에 정의 된 매크로를 이용하여 정의 되므로 이 매크로를 XML로 변환하는 방법을 정의하여야 한다. 먼저 SMIV1에 정의된 두가지 매크로인 OBJECT-TYPE 매크로와 TRAP-TYPE 매크로가 있다.

OBJECT-TYPE 매크로는 MIB의 테이블 노드나 데이터 노드를 정의하는 대표적인 매크로이다. 이 매크로의 변환 방법을 통해 MIB의 노드를 XML의 엘리먼트로 변환하는 방법을 알 수 있다. OBJECT-TYPE 매크로를 이용한 MIB 노드 표현의 일반형을 기술하면 다음과 같다.

노드명에 해당하는 'nodeName'을 XML에서는 엘리먼트 이름으로 하는 엘리먼트로 변경하고, 'ACCESS', 'STATUS' 등의 내부 절들은 속성으로 정의한다. 단, 'SYNTAX'절은 해당 노드의 데이터 타입에 대한 정보를 가지고 있으므로 XML로 변환 시에 속성으로 나타나지 않고 절의 내용인 'syntaxType'에 따라서 적당한 데이터 타입을 나타내는 '<xsd:restriction base = "xsd:string">'로 지정하거나 해당 엘리먼트의 'type'속성에 나타낸다. 그리고, 'oid'라는 새로운 속성을 정의하고 이것의 값은 해당 노드의 OID값을 가지도록 하였다. 이 값은 나중에 SNMP메시지와 XML문서간의 데이터 변환에서 키로 사용된다. 변환 후에 새로 만들어진 속성인 "oid"는 해당 노드의 OID 경로를 명시하는 것으로 RFC1213에 나타난 MIB II의 경우 system 노드의 oid 값은 "1.3.6.1.2.1.1"이다.

TRAP-TYPE 매크로는 SNMP 에이전트에서 발생하는 이벤트를 매니저로 보고하기 위한 트랩(trap)을 정의하는 이 매크로를 사용한 관리정보 정의의 일반형은 다음과 같이 나타낼 수 있다. 노드명을 엘리먼트명으로 변환을 하고 내부 절들은 엘리먼트의 속성으로 표현을 하였다.

SNMPv2 SMI에는 이전의 매크로를 확장 정의

한 OBJECT-TYPE 매크로 이외에도 여러 가지 매크로가 있다. MODULE-IDENTITY 매크로는 모듈에 관한 정보를 가지고 있다. 이 매크로는 그룹 노드로서의 기능을 하면서 모듈 정보를 가지고 있다. 그룹 노드로서의 기능은 앞서의 OBJECT IDENTIFIER와 같이 oid만을 속성으로 가지는 엘리먼트로 변환을 한다. 그리고 내부 절들이 가지는 정보는 실제 데이터 처리와는 무관한 정보이므로 XML Schema에서 문서에 대한 설명 정보를 저장하는 <annotation> 엘리먼트 부분에 저장을 한다.

SMIv2의 OBJECT-TYPE 매크로를 SMIv1의 OBJECT-TYPE 매크로표현과 유사한 방법으로 일반적인 노드 표현을 XML Schema로 변환한다. 단, 'MAX-ACCESS'절의 내용은 속성으로 변환 시에 속성의 이름을 SMIv1과 같은 이름을 사용하여 'access'로 정의한다. 추가된 속성을 고려하여 SMIv1의 변환 방법과 동일하게 XML Schema로 표현한다

SMIv2의 NOTIFICATION-TYPE 매크로는 노드는 값을 가지지 않고 단지 이벤트가 발생할 경우 해당하는 OID만을 알려주는 역할을 한다. 노드명을 엘리먼트명으로 사용하고 노드내부의 각 절은 엘리먼트의 속성으로 정의한다. 그리고 'oid'속성을 추가하여 노드를 구분하는 키로 저장하여 둔다.

다. 모듈명 및 그룹 노드

SNMP MIB 모듈은 ASN.1의 규칙에 따라서 'DEFINITION' 키워드를 사용하여 다음과 같이 'BEGIN'과 'END' 사이에 관리정보를 정의한다. XML문법에서는 문서 전체를 포함하는 하나의 루트 엘리먼트를 가져야 하므로 모듈명을 XML로 변환할 때 최상위 엘리먼트(루트 엘리먼트)로 변환을 하고, 각 모듈을 하나의 문서로 변환을 하면 정보의 손실이 없이 변환이 가능하다.

'OBJECT IDENTIFIER'는 그룹 노드를 나타내

는 방법으로 내부 절을 갖지 않고 자식 노드만을 가진다. 이 노드는 트리 구조에서 중간 노드에 해당하며 자식노드는 복수 개를 가질 수 있다. SNMP MIB에서는 중간노드는 값을 가지지 않고 트리의 마지막 노드에 해당하는 노드들만 값을 가지므로 이러한 그룹 노드는 값을 가지지 않는다.

이제까지 설명한 일반적인 변환 방법을 요약하면, 기본적으로 해당 노드의 이름을 엘리먼트 이름으로 변환을 하고 그 하부에 있는 노드들을 하부 엘리먼트로 지정을 한다. 내부 절들은 각각 이 엘리먼트의 속성으로 변환을 하고 그 값들은 고정값으로 지정을 한다. 모든 엘리먼트는 oid 속성을 가지는데 이 값은 해당 MIB 노드의 OID를 절대경로로 표시한다.

라. 규격변환의 예

이절에서는 규격변환의 예를 제시한다. 다음은 일반적인 형태의 SNMP MIB을 보여주고 있다.

```

nodeName OBJECT-TYPE
    SYNTAX "syntaxType"
    ACCESS "accessType"
    STATUS "statusType"
    DESCRIPTION "descriptionText"
    REFERENCE "referenceType"
    INDEX "indexList"
    DEFVAL "defaultValue"
 ::= { parentNodeName nodeNumber }

```

위와 같은 SNMP MIB을 앞에서 제시한 알고리즘으로 XML Schema로 변환을 하면 다음과 같다.

```

<xsd:element name="nodeName">
<xsd:complexType>
<xsd:simpleContent>
<xsd:restriction base="xsd:string">
    <xsd:attribute name="oid" type="xsd:string"
        use="fixed" value="oidValue"/>
    <xsd:attribute name="access" type="xsd:string"
        use="fixed" value="accessType"/>
    <xsd:attribute name="status" type="xsd:string"
        use="fixed" value="statusType"/>

```

```

<xsd:attribute name="description" type="xsd:string"
    use="fixed" value="descriptionText"/>
<xsd:attribute name="reference" type="xsd:string"
    use="fixed" value="referenceType"/>
<xsd:attribute name="index" type="xsd:string"
    use="fixed" value="indexList"/>
<xsd:attribute name="defval" type="xsd:string"
    use="fixed" value="defaultValue"/>
</xsd:restriction>
</xsd:simpleContent>
</xsd:complexType>
</xsd:element>

```

위의 변환 결과에서 노드명에 해당하는 'nodeName'을 XML에서는 엘리먼트명으로 사용하고, 'ACCESS', 'STATUS' 등의 내부 절들은 속성으로 정의한다. 단, 'SYNTAX' 절의 내용인 'syntaxType'에 따라서 이 노드의 데이터형이 결정되므로 이 절은 속성으로 지정하지 않고 데이터형을 나타내는데 사용한다. 그리고 'oid'라는 새로운 속성을 정의하고 이것의 값은 해당 노드의 OID값을 가지도록 하였다. 이 값은 나중에 SNMP메시지와 XML문서 간의 데이터 변환에서 키로 사용된다.

3.2 변환기 구현

변환기는 제시한 알고리즘을 자동 변환 프로그램으로 구현하여 SNMP MIB을 입력으로 받아서 XML로 변환해서 출력하는 기능을 가진다. 변환기의 입력은 SNMP MIB 정의가 된 파일이고, 입력은 로컬 파일이거나 URL을 입력하여 네트워크로부터 MIB을 입력할 수 있도록 한다.

변환기의 출력은 관리정보를 정의한 XML Schema 파일과 이 관리정보의 구조를 가지는 XML DOM 트리다. 스키마 파일은 네트워크 관리를 위해 전달되는 XML 문서의 구조를 나타내며 문서의 검증에 사용된다. XML DOM은 템플릿 문서를 생성하거나 관리 데이터의 처리를 위해 사용된다. 작성된 DOM은 런타임에 변경이 가능하여야 하며 여러 개의 MIB을 사용하여 하나의 DOM으로 작성하는 MIB 통합 기능을 가지고 있어야 한다. 표준

MIB과 사설 MIB을 동시에 사용하는 SNMP 에이전트를 가지는 장치를 관리하기 위해서는 복수의 MIB 모듈을 하나의 DOM으로 통합해서 작성하는 기능이 필요하다. 이것은 여러 종류의 SNMP 에이전트에서 데이터 수집을 할 때에 DOM을 수시로 변경하지 않고도 사용할 수 있는 방법을 동시에 제공한다.

이 변환기의 구조를 그림으로 나타내면 그림 2와 같다. MIB 파일을 읽어 들여 일련의 토큰으로 나누고 구문을 분석하는 구문 분석 모듈, 하나의 노드 정보를 하나의 자바 클래스로 매핑하여 저장하는 노드 테이블 생성 모듈, 저장된 노드 테이블에서 정보를 읽어서 XML Schema를 출력하는 XML Schema 생성 모듈, 노드 테이블에서 정보를 읽어서 XML DOM 트리를 생성하는 DOM 생성기로 이루어져 있다.

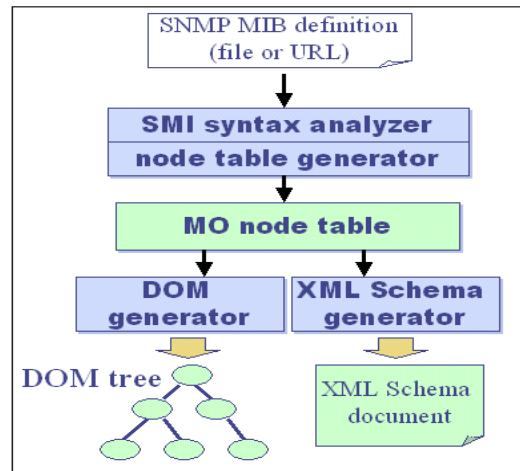


그림 2. SNMP MIB to XML 변환기 구조

구문 분석 모듈(SMI syntax analyzer)은 SNMP MIB을 입력으로 받아서 주석을 삭제하고 일련의 토큰으로 나눈다. 토큰을 분석하여 ASN.1의 부분 집합인 SMI의 구문에 맞는지 검사한다. 구문에 맞지 않는 MIB 정의가 있을 경우 에러를 발생하고 해당 라인을 알려준다.

노드 테이블 생성 모듈(node table generator)에서는 구문 분석된 토큰에서 각각의 노드를 자바클래스로 매핑하고 노드 클래스를 데이

블로 저장한다. 저장된 테이블에서 노드간의 상관관계를 분석하여 존재하지 않는 노드를 부모노드로 가지는 노드가 있을 경우 에러를 발생하고 해당 노드를 알려준다. 새로운 사용자 데이터 타입을 선언한 경우 이를 데이터 타입 테이블에 저장한다. 이것은 XML Schema 출력시에 데이터 타입 선언에 사용된다.

XML Schema 출력 모듈(XML Schema generator)은 노드 테이블에 저장된 관리 정보를 이용해서 W3C 표준의 XML Schema로 변환된 관리정보 정의를 출력하는 기능을 한다. 이 때 생성된 파일은 XML 문서를 검증하는데 사용되며 관리대상을 정의한 정보를 가지는 기능을 한다.

DOM 생성 모듈(DOM generator)은 노드테이블에 저장된 관리 정보를 이용해서 SNMP MIB의 트리 구조와 같은 구조의 DOM 트리를 생성한다. 생성된 DOM 트리는 SNMP 에이전트에 폴링할 때에 기준으로 사용이 되며, XML 문서의 파싱과 생성 등에 사용되는 것으로 이때 표준화된 DOM API를 사용하게 된다. 생성된 DOM 트리의 크기는 SNMP 에이전트 수와 관계없으며 단지 지원하는 SNMP의 다소에 따라서 DOM 트리의 크기가 좌우된다.

SNMP MIB에서 IMPORT절을 사용하여 다른 MIB을 조회하는 경우에는 조회 대상 MIB을 먼저 파싱하여 본래의 MIB에서 사용하는 노드나 데이터 타입을 먼저 노드 테이블과 데이터 타입 테이블에 저장하여 둔다.

변환기 구현에는 자바 프로그래밍 언어를 사용했다. JDK1.3을 사용하여 구현하였으며, DOM 생성을 위해서 Apache에서 제공하는 자바 패키지인 Xerces [28]를 이용하였다.

4. SNMP-XML 게이트웨이

이 장에서는 XML 기반의 네트워크 관리 시스템에서 기존에 널리 사용되고 있는 SNMP 에이전트를 관리하기 위한 방법인 SNMP-

XML 게이트웨이의 구현을 설명한다.

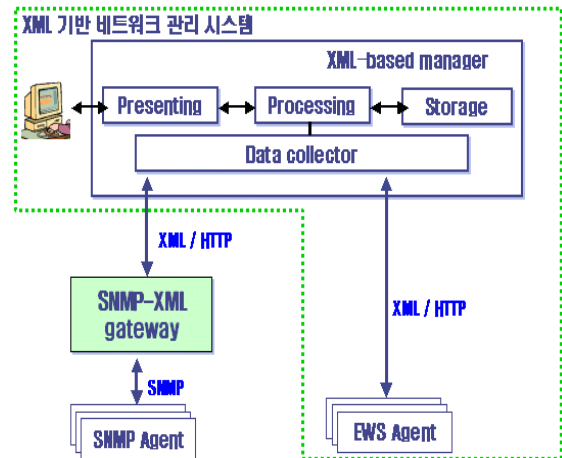


그림 3. SNMP-XML 게이트웨이 적용

XML 기반 네트워크 관리 시스템의 매니저를 이용해서 기존에 널리 사용되는 SNMP 에이전트를 통합해서 관리하는 방안을 제공하기 위해서는 SNMP 메시지를 XML로 표현해 주는 게이트웨이가 필수적이다. 이는 데이터 처리를 XML 기반으로 하는 매니저가 SNMP 에이전트를 장착한 장비를 관리하기 위해서는 SNMP를 이용하여 데이터를 요청하고, 수집된 데이터(SNMP 메시지)를 XML 문서로 변환하여 표현하는 기능을 하는 중간 변환 장치가 필요하기 때문이다. 이러한 기능을 하는 SNMP-XML 게이트웨이를 이용하여 네트워크 관리 시스템을 구성한 구조를 그림 3에 나타내고 있다.

4.1 설계

게이트웨이의 동작을 정의하기 위해서는 매니저와 에이전트 양방향의 규격변환과 상호 동작변환을 고려해야 한다. 본 게이트웨이에서는 SNMP 에이전트가 사용하는 관리 정보인 MIB을 XML Schema로 변환하는 규격변환을 위해서는 4장에서 구현한 변환기가 사용되었다.

상호동작변환이란 실제로 전달되는 데이터의 표현과 전달 방법을 변환하는 것이다. SNMP 프로토콜에서 동작은 get, set, trap의 세가지 유

형으로 나뉘어 진다. get 동작은 SNMP 매니저가 에이전트로부터 데이터를 가져오는 동작으로 이때는 에이전트의 컴퓨터 주소, 커뮤니티명, OID를 필요로 한다. set 동작은 SNMP 매니저가 에이전트에게 데이터를 보내어 특정 OID의 값을 변경하는 동작으로 이때는 컴퓨터 주소, 커뮤니티명, OID, 보낼 값이 필요하다. trap은 에이전트에서 특정한 이벤트가 발생하면 이를 자동으로 매니저로 보고하는 동작으로 이때는 컴퓨터 주소, OID, 값이 전달되며 매니저는 수신만 하게 된다. 각각의 SNMP 동작을 HTTP 동작으로 매핑한 방법을 표 2에 나타내고 있다.

매니저에서 HTTP로 요청할 때 HTTP 메시지는 host, operation, community, xpath, value의 5가지 파라미터로 구성 된다. ‘host’인자는 SNMP 에이전트의 주소를 나타내고 ‘operation’인자는 동작이 get인지 set인지를 구분하고, ‘community’인자는 SNMP 에이전트에서 사용하는 커뮤니티명을 전달한다. ‘xpath’인자는 요청하는 SNMP 메시지의 OID에 해당하는 노드명을 XML에서 사용하는 Xpath를 사용하여 표현한다. ‘value’인자는 SNMP set 명령의 경우에 agent에 전달할 값을 저장한다. Set 명령의 경우에는 HTTP POST방식을 사용하여 데이터를 첨부하는 방법을 사용한다.

표 2. SNMP-HTTP 간 동작 변환

SNMP 동작	HTTP 동작
Get	http://(gateway_address)/SNMP?host=agent_name&community=community_name&operation=get&xpath=node_name
Set	http://(gateway_address)/SNMP?host=agent_name&community=community_name&operation=set&xpath=node_name&value=value_string
Trap	http://(manager_address)/NOTIFICATION/SNMP?host=host_name

게이트웨이는 그림 4과 같은 구조를 가지고 있으며 각 구성 모듈의 기능은 다음과 같다.

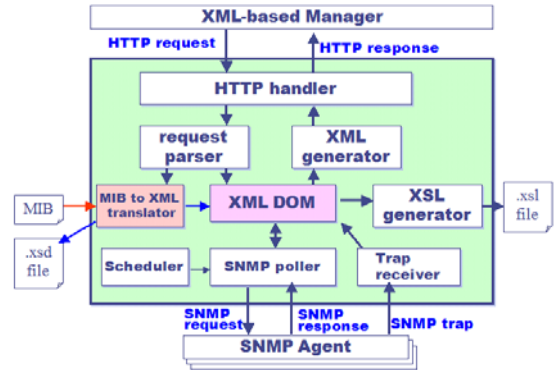


그림 4. 게이트웨이 구조

- HTTP handler : HTTP 연결을 통해 XML 기반 매니저와 통신을 하기 위한 모듈로서 HTTP 요청을 받아서 이를 파서로 전달하거나, 생성된 XML 문서를 XML 매니저로 전달하는 기능을 한다.
- request parser : XML 매니저에서 HTTP GET 혹은 HTTP POST 메시지로 전달되는 요청 메시지를 분석하고 그 요청에 해당하는 동작을 하도록 다음 모듈로 명령을 전달한다.
- XML DOM : MIB 파일을 입력으로 MIB2XML 변환기가 생성한 DOM 트리의 데이터 구조이다.
- XML generator : DOM의 데이터를 기초로 XML 문서를 생성해서 HTTP handler로 전달한다.
- MIB to XML translator : SNMP 에이전트에서 사용하는 MIB 파일을 입력으로 받아서 XML Schema로 변환하고 DOM 트리를 생성한다.
- SNMP poller : SNMP 에이전트에 SNMP 요청 메시지를 보내어 데이터를 수집하고 수집되는 데이터의 해당 OID에 상응하는 DOM 트리에 데이터를 수정

한다.

- Trap receiver : SNMP trap 메시지를 받아서 이에 해당하는 DOM 트리의 데이터를 수정하고 XML 로 변환된 Notification 메시지를 XML 기반 매니저로 전달 하도록 하는 기능을 한다.
- Scheduler : SNMP poller의 폴링 기간을 설정하는 스케줄링 기능을 한다.
- XSL generator : DOM의 데이터를 기초로 XML 문서에 적용할 스타일 시트인 XSL 파일을 생성한다. 이 스타일은 게이트웨이를 MIB 브라우저로 활용할 때 사용된다.

4.2 동작 시나리오

XML 기반 매니저와 SNMP 기반 에이전트의 사이에서 프로토콜 변환을 하는 게이트웨이의 실제 메시지 변환 및 전달 시나리오를 서술하면 다음과 같다.

가. 요청/응답 모델

요청/응답(Request/Response) 모델을 일반적인 관리의 방법으로 매니저에서 에이전트에 필요한 정보를 요구하는 요청 메시지를 보내면, 에이전트는 이에 상응하는 응답 메시지를 매니저로 보내는 방식이다. 이 방식의 동작 과정은 아래의 순서와 같다.

- ① 매니저의 요청이 있을 경우 이는 HTTP handler를 통해 수신되어서 request parser로 전달된다.
- ② request parser에서 요청의 내용을 분석한다. 분석된 요청 결과는 DOM에서 해당 노드를 조회해서 OID 를 구하고 SNMP poller는 매니저의 요청이 get이면 SNMP 에이전트로부터 get 방식으로 해당 OID의 값을 읽어 온다.
- ③ 반환 값은 해당 OID에 상응하는 DOM의

노드에 저장하고, DOM 수정이 완료되면 XML generator에서 XML 문서를 작성한다.

- ④ 작성된 XML 문서는 HTTP handler를 통해서 요청한 매니저로 보내어짐으로써, HTTP 응답 메시지를 보낸다.

나. 통지 모델

통지(Notification) 모델은 에이전트가 미리 정해진 사건이 발생할 경우 이 사실을 매니저에게 자발적으로 통지하는 방식의 전달을 말한다. 이 방식의 동작 과정은 아래와 같다.

- ① SNMP 에이전트에서 이벤트(Event)가 발생하면 이는 trap 메시지로 게이트웨이의 Trap receiver로 전달된다.
- ② Trap receiver는 수신된 trap 메시지를 분석하여 호스트, OID, 값, 발생시간 등의 정보를 구하고 이를 이용해서 DOM 트리를 갱신한다.
- ③ XML generator를 구동하여 이벤트 통지에 해당하는 XML 문서를 작성하고 이를 HTTP handler를 통해서 매니저로 전달한다. 이때는 전달 방식에서 HTTP POST 방식을 사용한다.

4.3 구현

게이트웨이 구현은 자바 프로그래밍 언어를 사용하고, JDK1.3을 기반으로 하였다. SNMP 에이전트가 사용하는 관리정보인 MIB을 XML Schema로 변환하는 규격변환을 위해서는 5장에서 구현한 변환기가 사용되었다. SNMP 에이전트와 통신하기 위한 SNMP Poller 모듈과 Trap receiver 모듈은 자바 기반인 openNMS의 joeSNMP 패키지 [38]를 사용하여 구현하였고 SNMPv1, SNMPv2c를 지원한다. SNMP 에이전트와 통신하는 이 두 모듈은 수신된 데이터를 XML DOM에 저장하는 기능을 포함하도록 구성하였고 DOM을 접근

하기 위한 함수들은 Apache에서 제공하는 자바 패키지인 Xerces [28]를 이용하였다.

XML 기반 매니저와 통신하기 위한 HTTP handler 모듈은 JDK1.3에 제공되는 함수를 이용하여 자바로 구현하였고 HTTP 요청을 분석하는 request parser 모듈과 XML generator 모듈은 DOM을 조회하는 기능을 구현하는데 Xerces를 이용하였다.

게이트웨이의 동작 중에 다른 MIB을 사용하는 SNMP 에이전트를 관리하기 위해 DOM을 변경하여야 하는 기능은 request parser 모듈에 내장된 함수로 이를 구현하였다.

4.4 동작 화면

게이트웨이의 동작을 확인하기 위해서 웹 브라우저에서 직접 게이트웨이로 메시지를 요청을 보내면 게이트웨이는 SNMP 에이전트에서 가져온 데이터를 XML로 변환한다. 게이트웨이의 XML 생성기가 XSL 파일을 자동으로 생성하게 하고 결과 파일에 스타일 시트를 적용하면 그림 5과 같이 사용자가 보기 편한 화면으로 나타낼 수 있다.

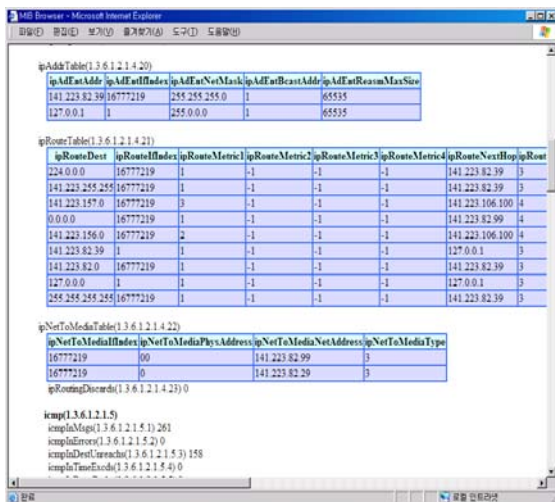


그림 5. 게이트웨이 동작화면

5. 결론 및 향후과제

본 논문에서는 XML 기반의 네트워크 관리

시스템에서 기존의 SNMP 에이전트를 통합 관리하기 위한 방법으로 SNMP MIB을 XML로 변환하는 알고리즘을 제안하고 그에 따른 변환기를 구현하며, 이를 이용하여 SNMP-XML 게이트웨이를 설계하고 구현하였다.

SNMP는 확장성과 효율성에서 제한을 가지고 있어서 날로 증가하는 거대한 네트워크를 관리하기에는 한계점이 나타났다. 이러한 SNMP 기반 네트워크 관리 시스템의 단점을 보완하여 거대한 네트워크에서 생산되는 많은 관리 데이터를 효율적으로 전달하고 처리하기 위한 방안으로 XML을 네트워크 관리에 이용하고자 하는 연구들이 나타나고 있다. XML 기반의 네트워크 관리 시스템은 확장성에서 장점을 가지지만 기존의 SNMP 에이전트가 장착된 장치를 그대로 두고 관리하지는 못하는 문제점이 있다. 인터넷 네트워크 관리는 대부분이 SNMP를 기반으로 개발되어 왔기 때문에 대부분의 인터넷 장치는 SNMP 에이전트를 장착하고 이를 통해 네트워크 관리가 이루어지고 있다. 따라서 XML 기반 네트워크 관리의 장점을 이용하면서도 기존의 SNMP 에이전트를 통합해서 관리할 수 있는 방안을 본 논문에서 제시하였다.

XML 기반 매니저에서 SNMP 에이전트를 변경 없이 통합 관리하기 위해서 다른 프로토콜을 사용하는 매니저와 에이전트의 사이에서 번역을 해주는 게이트웨이를 구현하였다. 규격변환 (Specification Translation)을 위해서는 SNMP 에이전트가 사용하는 관리정보 정의인 SNMP MIB을 XML로 변환해야 하기 때문에 SNMP MIB의 XML 변환 알고리즘을 설계하고 자동변환기를 구현하였다. 상호동작변환 (Interaction Translation)을 위해서는 HTTP 상의 XML 문서와 SNMP 상의 메시지를 변환해주는 매핑방법을 규정하고 구현하였다.

뛰어난 호환성과 사용하기 쉬운 XML의 특성 때문에 XML을 이용한 데이터의 통합은 앞으

로 더욱 가속화 될 전망이다. 특히 네트워크 및 시스템 관리를 위한 정보의 전송에서 이 기종의 시스템간에 정보전달을 위해 XML을 미들웨어로 사용하기 위해서는 SNMP와 XML간의 상호작용이 반드시 필요하며 본 논문에서는 이를 가능케하는 게이트웨이대한 연구결과를 제시하였다.

현재 구현된 게이트웨이는 기능적인면에 중점을 두고 개발되었다. 향후 과제로는 개발된 게이트웨이의 성능분석을 통하여 다수의 SNMP가 설치되어 운영되고 있는 환경에 게이트웨이가 효과적으로 동작할 수 있는 방법을 제시하는 것이다. 변환 알고리즘의 성능평가를 통해 보다 효율적인 변환 알고리즘의 개선이 필요하며 동작변화측면에서도 SNMP 에이전트와 게이트웨이간 그리고 게이트웨이와 XML 매니저간의 관리 동작을 최소화하기 위한 방법이 제시되어야 한다..

참고문헌

- [1] The World Wide Web Consortium,
<http://www.w3c.org>.
- [2] W. Stallings, "SNMP, SNMPv2, SNMPv3, and RMON 1 and 2," Third edition, Addison-Wesley, Reading, MA, USA, 1999.
- [3] Hong-Taek Ju, Sehee Han, Yunjung Oh, Jeong-Hyuk Yoon, Hyojin Lee, James W. Hong, "An Embedded Web Server Architecture for XML-Based Network Management", Proc. of the IEEE/IFIP Network Operations and Management Symposium (NOMS 2002), April, 2002, Florence, Italy, pp. 5-18.
- [4] J.P. Martin-Flatin. "Web-Based Management of IP Networks and Systems", Ph.D. thesis, Swiss Federal Institute of Technology, Lausanne (EPFL), Oct 2000.
- [5] 김영대, 조기용, 허재호, 전재규, 조석형, "SNMP SMI to XML 변환을 이용한 망관리 시스템", Proc. of KNOM 2001 Conference, Taejeonn, May, 2001, pp. 102-106.
- [6] W3C, Extensible Markup Language (XML) 1.0, W3C Recommendation, October 2000,
<http://www.w3.org/TR/REC-xml>.
- [7] W3C, "HTML 4.0 Specification," Internet Draft, HTML Working Group, Apr. 1998.
- [8] International Organization for Standardization, "ISO 8879: Standard Generalized Markup Language (SGML)", 1986.
- [9] Network Sorcery. Inc., "ICMP, Internet Control Message Protocol ,"
<http://www.networksorcery.com/enp/protocol/icmp.htm>.
- [10] ISO/IEC, CCITT, Information Technology- OSI, Common Management Information Protocol (CMIP)-Part 1: Specification ISO/IEC 9596-1, CCITT Recommendation X.711, 1991.
- [11] Subrata Mazumdar, "CORBA/SNMP Gateway", Bell Labs, <http://www.bell-labs.com/project/CorbaSnmpp>.
- [12] Frank Strauss, "A Library to Access SMI MIB Information", <http://www.ibr.cs.tu-bs.de/projects/libsmi/>.
- [13] Imamura, T. and Maruyama, H., "Mapping between ASN.1 and XML", Applications and the Internet, 2001. Proceedings, IEEE, 2001, pp. 57 -64.
- [14] ITU-T, "Specification of Abstract Syntax Notation One(ASN.1)", ITU-T Rec. X.208, 1998.
- [15] M. Rose, K. McCloghrie, "Structure and Identification of Management Information for TCP/IP-based Internets," IETF, RFC1155, May 1990.

- [16] M. Rose, K. McCloghrie, "Concise MIB Definitions", IETF, RFC1212, March 1991.
- [17] K. McCloghrie, M. Rose, "Management Information Base for Network Management of TCP/IP-based Internet: MIB-II", IETF, RFC1213, March 1991.
- [18] K. McCloghrie, et al, "Structure of Management Information Version 2 (SMIv2)", IETF, RFC2578, April 1999.
- [19] K. McCloghrie, et al, "Textual Conventions for SMIv2", IETF, RFC2579, April 1999.
- [20] K. McCloghrie, et al, "Conformance Statements for SMIv2", IETF, RFC2580, April 1999.
- [21] J. Case, et al, "Management Information Base for Version 2 of the Simple Network Management Protocol (SNMPv2)", IETF, RFC1907, January 1996.
- [22] J. Case, "Introduction to Version 3 of the Internet-standard Network Management Framework", IETF, RFC2570, April 1999.
- [23] W3C, "Cascading Style Sheets, level 1", Recommendation, Dec 1996, revised Jan 1999, <http://www.w3.org/TR/REC-CSS1>.
- [24] W3C, "Extensible Stylesheet Language (XSL) Version 1.0", Recommendation, Oct 2001, <http://www.w3c.org/Style/XSL/>.
- [25] W3C, "Document Object Model (DOM) Level 2 Core Specification", Nov 2000, <http://www.w3.org/TR/DOM-Level-2-Core/>.
- [26] Simple API for XML 2.0, 1999, <http://www.saxproject.org/>.
- [27] W3C, "XML Path Language (XPath) Version 1.0", Recommendation, Nov 1999, <http://www.w3.org/TR/xpath>.
- [28] Apache XML project, "Xerces Java parser", <http://xml.apache.org/xerces-j/>.
- [29] Apache XML project, "Xalan - XSLT processor", <http://xml.apache.org/xalan-j/index.html>.
- [30] W3C, Extensible Markup Language (XML) 1.0 - DTD, W3C Recommendation, October 2000, <http://www.w3.org/TR/REC-xml#dt-doctype>.
- [31] W3C, "XML Schema Part 0: Primer", Recommendation, May 2001, <http://www.w3.org/TR/xmlschema-0/>.
- [32] Internet Engineering Task Force, <http://www.ietf.org/>, <http://www.ietf.org/rfc.html>.
- [33] W3C, "XML Linking Language (XLink)", Version 1.0, W3C Proposed Recommendation, Dec 2000, <http://www.w3.org/TR/xlink/>.
- [34] W3C, "XML Pointer Language (XPointer)", Version 1.0, W3C Last Call Working Draft, Jan 2001, <http://www.w3.org/TR/xptr>.
- [35] W3C, "XML-Data", <http://www.w3.org/TR/1998/NOTE-XML-data-0105/>.
- [36] Ora Lassila, "Introduction to RDF Metadata," W3C NOTE, Nov 1997, <http://www.w3.org/TR/NOTE-rdf-simple-intro>.
- [37] W3C, "Document Content Description for XML", <http://www.w3.org/TR/NOTE-dcd>.
- [38] OpenNMS, <http://www.opennms.org/>.