XML기반 PIB를 이용한 네트워크 관리구조

정회원 윤 권 섭*, 홍 충 선*

A Network Management Architecture Using XML-based PIB

Kwoun-Sup Youn*, Choong-Seon Hong* *Regular Members*

요 약

정책기반 네트워크 관리 구조는 정책을 전송하기 위해 COPS(Common Open Policy Service)와 이의 모델인 COPS-PR (Policy Provisioning)을 사용한다. COPS-PR은 여러 네트워크 관리 영역에서 효율적으로 활용될 수 있는 매커니즘을 가지고 있다. COPS-PR은 정책을 저장하고 디바이스정보를 저장하기 위해 PIB(Policy Information Base)를 사용한다. PIB는 PRC(Provising Class)와 PRI(Provisiong Instances)로 구성되어 있다. PIB는 정책을 통해 디바이스를 제어하기 위한 기능이 미리 구현되어 있다. PIB에 PRI를 추가시킴으로써 정책을 수행시킬 수 있다. 하지만 미리 구현되어 있는 기능에 한정되어 사용하기 때문에 새로운 기능에 대해선 정책을 적용할 수 없는 단점을 가지고 있다. 본 논문에서 제안하는 구조는 기존 PIB에 단점을 보완하기 위해 XML로 변환된 PIB를 사용한다. XML 기반 PIB는 동적으로 새로운 기능의 추가가 가능하며, 정책을 통해 이를 수행시킬 수 있는 구조이다. 또한 본 구조에서는 정책 기술 시에도 XML을 이용하였다. 정책 기술시 XML을 이용하면 서로 다른 정책기술 언어를 사용하는 이 기종의 관리 시스템간에 정책 교환 시 XSLT(eXtensible Stylesheet Language Transformation)을 이용하여 쉽게 변환될 수 있다. 본 논문에서는 동적 확장을 고려한 XML기반 네트워크 관리 구조를 제안하고, 이를 구현하여 기존 시스템과의 차별성을 평가하였다.

ABSTRACT

XML is being used to describe components and applications in a vendor and language neutral. Therefore it already has a role in distributed system. XML is also being used as a data interchange format between components and applications in loosely coupled large-scale application. Until now, policy is described for specific applications and devices. Its use has been very limited. In current network management system, we can only invoke predefined operations and actions using policy-based network management. The main motivation for the recent interests in policy-based networks is to support dynamic adaptability of behavior by changing policy without recoding or stopping system. For these reasons we present the use of the XML for describing the policy and PIB(Policy Information Base) in COPS-PR. It improves flexibility and interoperability among heterogeneous network systems. It also can add new functionality into network components. In this paper, we propose a *dynamically extensible network management architecture using XML-based PIB.*

I. 서 론

최근 수년간, 네트워크를 효율적으로 관리하기 위하여 여러 가지 연구 개발이 추진되어 왔다. 초기에는 네트워크 전체보다는 각각의 네트워크 요소를 관리하는 것에 초점을 맞추었지만, 네트워크의 증가 및 복잡성으로 인하여 각각의 네트워크 요소의 관리 방법은 현재의 네트워크 관리에 충분한 방법을 제시하지 못한다. 이러한 결과로 인해 나타난 것이 정책기반의 네트워크 관리 구조이다[]. 정책기반 네트워크 관리[1]는 네트워크를 구성하는 각 네트워크 요소에 정책을 변화시킴으로서 시스템의 운용을 동적으로 수정할 수 있는 유연성을 제공할 뿐만 아니라, 적은 비용으로 전체적인 네트워크를 구성 할 수 있다.

* 경희대학교 전자정보학부 (cshong@khu.ac.kr)
※ 본 연구는 한국과학재단 목적기초연구(R05-000-00976-0)지원으로 수행되었음

현재 정책기반 관리는 특별한 어플리케이션을 위한 정책 구현 및 그 정책을 적용하려는 엔티티를 위한 특별한 정보모델 지원에 한정되어 있다. 또한 정책을 네트워크 요소에 적용할 때 특정한 관리 프로토콜을 이용해야 한다. 이것은 규모가 큰 이 기종의 네트워크 관리에 많은 문제점을 야기하고 있다. 따라서 네트워크를 구성하는 각각의 네트워크 요소의 타입이나 서로 다른 업체의 장비의 종류에 상관없이 정책을 적용할 수 있는 방법 및 정책을 기술하고 표현, 관리하는 방법이 필요하다. 또한 좀 더 이상적인 관리 시스템은 관리정보의 교환을 위한 구현 및 운영체제에 중립적인 정보모델을 가져야 하며, 다른 네트워크 관리 솔루션과의 상호 작용 및 운용성이 있어야 한다.

이러한 점을 해결하기 위해 DMTF(Distributed Management Task Force)는 정책을 기술하기 위한 공통적인 모델로 CIM(Common Information Model)[6,7]을 사용하고, 이와 관련하여 DEN (Directory Enabled Network)[26]을 제시하고 있다. DEN[26]은 각각의 네트워크 요소를 관리하기보다는 전체적인 네트워크를 관리하는 방법을 제공하며, 서로 다른 어플리케이션간에 네트워크 정보의 재사용 및 공유를 위한 확장방법을 사용한다. 또한 각 디바이스의 관리프로토콜에 상관없이 LDAP (Lightweight Directory Access Proto -col)[10, 11]을 이용하여 디바이스들을 관리한다.

IETF에서는 COPS(Common Open Policy Servi -ce)[2]와 그것의 확장인 COPS-PR(Policy Provis -ioning)[3]을 개발하였다. COPS는 PDP(Policy Decision Point)[12]와 PEP(Policy Enforcement Point)[12]사이에 정책과 관련된 정보를 주고받기 위해 디자인되었다. 그림 1은 PDP와 PEP 간의 관계 및 기본적인 COPS 오퍼레이션을 보여주고 있다. PEP는 필요한 정책을 PDP에 요청하고, PDP는 네트워크 장비들에게 반영할 정책 정보를 수립 PEP에게 전송한다. PEP는 해당 정책 정보를 받아 네트워크 장비에 반영한 후 PDP에게 결과를 보고한다.

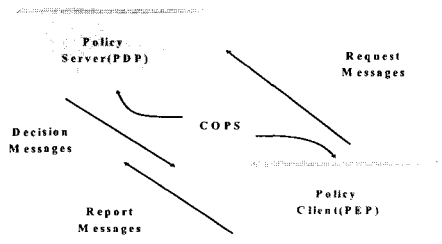

그림 1. PDP-PEP 관계 및 COPS 오퍼레이션 모델

COPS-PR[3]은 네트워크 장비를 제어하고 PEP에 정책정보를 저장하기 위한 PIB(Policy Informat -ion Base)[3,5]라는 구조를 사용한다. 앞서 언급한 바와 같이 PIB는 정책을 통해 디바이스를 제어하기 위한 기능이 미리 구현되어 있다. PIB에 있는 PRC(Provisioning Class)[3]에 PRI(Provisioning Instances)[3]를 추가시킴으로써 정책을 수행시킬 수 있다. 하지만 미리 구현되어 있는 기능에 한정되어 사용하기 때문에 새로운 기능에 대해선 정책을 적용할 수 없는 단점을 가지고 있다. 본 논문에서 제안하는 구조는 기존 PIB에 단점을 보완하기 위해 XML로 변환된 PIB를 사용한다. XML로 변환된 PIB을 사용하면 기존 PIB의 단점을 극복할 수 있다. XML 기반 PIB는 동적으로 새로운 기능의 추가가 가능하며, 정책을 통해 이를 수행시킬 수 있는 구조이다. 또한 본 구조에서는 정책 기술 시에도 XML을 이용하였다. 정책 기술시 XML을 이용하면 서로 다른 정책기술 언어를 사용하는 이 기종의 관리 시스템간에 정책 교환 시 XSLT(eXten -sible Stylesheet Language Transformation)을 이용하여 쉽게 변환될 수 있다.

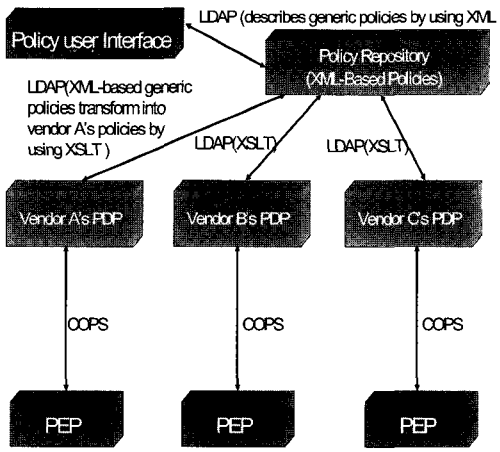

그림 2. XML 기반 정책관리 시스템 구조

그림 2는 하나의 XML로 기술된 정책을 디렉토리 서버에서 XSLT을 이용하여 각각의 관리 시스템에 맞는 정책 포맷으로 변환되어 이 기종의 관리 시스템들의 정책서버로 정책이 보내어 지는 것을 보여주고 있다. 따라서 관리시스템마다 따로 정책을 기술하는 것이 아니라 하나의 정책서버만을 사용함으로서 관리시스템간 상호 운용성 및 정책을 일관성 있게 네트워크에 적용할 수 있다는 장

점을 가진다.

본 논문의 2장에서는 관련연구로서 COPS-PR, SNMP MIB이나 관리정보의 XML변환 그리고 이전 정책기반 네트워크 관리 구조에 대해 살펴보고, 3장에서는 제안하는 시스템 구조 및 각각의 컴포넌트에 대해 설명한다. 4장에서는 XML로 만들어진 템플릿과 PIB에 대해 언급하고, 5장에서는 제안구조의 구현사항을 설명토록 한다. 6장에서는 본 시스템의 운용시나리오에 대해 살펴보고, 7, 8장에서는 평가와 본 논문의 결론 및 향후 연구과제에 대해여 각각 논한다.

Ⅱ. 관련연구

2.1 COPS-PR(Policy Provisioning)

IETF의 RAP(Resource Allocation Protocol) WG에 의해 COPS의 확장으로서 개발되었다. COPS-PR은 초기에 DiffServ policy provisioning을 목적으로 하였으나, 여러 가지 다른 관리영역에서도 사용할 수 있는 규격으로 완성되었다. COPS-PR의 이름에서도 알 수 있듯이, COPS-PR은 provisioning 모드에서 동작한다. 클라이언트가 적절한 PDP에 접속한 후, 자신의 능력 및 제한점을 리포트 한 후 PDP에게 정책을 요청한다. PDP는 각 클라이언트의 요청을 처리한 후 PEP에 적절한 구성정보를 전송한다. 만약 네트워크 상태나 정책이 바뀌면 PDP는 관리 시스템을 일관되게 제어하기 위하여 구성정보를 업데이트 하게 된다. COPS-PR에서 각 클라이언트는 PDP로부터 받은 구성정보를 저장하기 위한 PIB(Policy Information Base)라는 특별한 데이터베이스를 유지해야 한다. 그림 3은 PIB의 구조를 나타내고 있다. PIB의 구조는 MIB의 구조와 비슷하며, PRC(*Policy Rule Class or* Provisionig Cl -ass)[3]와 PRI(Policy Rule Instance or Provisio -ning Instance)[3]로 구성된 개념적인 트리 네이밍공간으로 기술되어 질 수 있다. COPS-PR에 의해 정의된 PIB들은 단지 추상적인 구조이며, 각 PIB의 구조는 독립적인 표준문서에 의해 정의되어진다. PIB들은 하드웨어의 복잡성을 감추기 위하여 높은 추상레벨로서 정의되어진다.

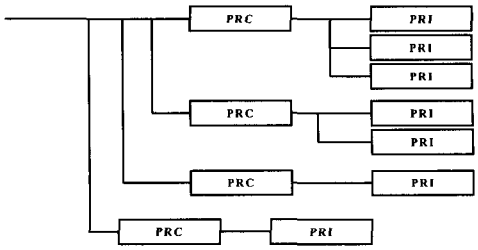

그림 3. PIB의 구조

PRI는 PIB내에서 PRI identifier(PRID)[3]로서 구별되어진다. 정책은 PIB의 PRI들로서 형성되어진다. PRI들을 추가하고 삭제함으로써, PDP들은 디바이스에 적용되어질 수 있는 정책들을 구현한다.
각 PIB의 정책은 미리 정의되어 구현되어진다. 네트워크 디바이스를 제어하기 위해, PDP는 하이 레벨 정책을 PEP의 PIB에 구현된 정책에 맵핑시켜야 한다. PIB는 SPPI(Structure of Policy Provisioning Information)[13]을 사용하여 정의되어진다.

2.2 관리정보, SNMP MIB의 XML 변환

DMTF에서는 enterprise 컴퓨팅 환경에서 통합적인 웹기반 네트워크 관리를 하기 위한 표준으로 WBEM(Web-based Enterprise Management)[24]를 제시하고 있다. WBEM은 표준 정보모델로 정의된 CIM (Common Information Model)[6]를 사용하고 있으며, 이를 XML로 변환하는 표준으로 xmlCIM을 제시하고 있다. DMTF는 CIM을 XML로 변환하는 알고리즘으로 'schema mapp-ing' 과 metaschema mapping'이라는 두가지 모델을 제시하고 하고 있다[8].
Schema mapping은 CIM 클래스를 표현하기 위해 XML schema를 사용하는 방법이다. 이 모델에서 CIM 인스턴스는 'valid XML 문서'로 맵핑된다. 즉 XML Element 명으로 CIM Element 명을 그대로 사용하는 방식이다.

Metaschema mapping은 CIM metaschema를 표현하기 위해 XML Schema를 사용하는 방법이다. 이 모델에서는 XML Element명으로 CIM 클래스명을 사용하지 않고 일반적인 Element 명을 사용한다. 각 클래스는 공통의 Element명을 사용하여 정의되고, Attribute를 이용하여 구분을 하는 방

식이다.

J.P Martin-Flatin은 웹기반의 통합 관리 아키텍쳐를 제안하고, 데이터 통합을 위한 방법으로 SNMP MIB을 XML로 변환하기 위한 모델을 제안하였다. 이 모델은 DMTF에서 제시한 모델에 상응하는 것으로, Model-level mapping[17] 과 Meta-model-level mapping[17]이라고 정의하고 있다.

Model-level mapping은 DTD가 SNMP MIB에 특정하게 작성되는 방법이다. SNMP 변수명을 그대로 사용하여 XML DTD에서 element 와 atrribute를 작성한다. 표 1은 Model-level mapping을 보여주고 있다.

표 1. Model-level Mapping 변환 과정

```
// SMI를 이용한 MIB의 정의
sysDescr OBJECT-TYPE
SYNTAX DisplayString(SIZE(0..255))
ACCESS read-only
STATUS mandatory
"a textual..." ::= { system 1}
//DTD를 이용하여 변환된 XML 문서
<system oid = 1.3.6.1.2.1.1">
<sysDescr oid =" 1.3.6.1.2.1.1.1"
SYNTAX="DisplayString" ACCESS = "read
-only" STATUS="mandatory">
</sysDescr></system>
//DTD를 이용하여 XML 기반의 MIB 정의
<!Element sysDescr #PCDATA>
<!ATTLIST sysDescr oid CDATA
#REQUIRED SYNTAX CDATA
#REQUIRED ACCESS CDATA
#REQUIRED STATUS CDATA
#REQUIRED>
```

이 모델로 작성한 XML 문서의 장점은 사람이 읽고 직관적으로 이해하기 좋다는 것이다. 이 모델의 단점은 MIB별로 DTD를 작성해야 하기 때문에 많은 DTD를 필요로 한다는 것이다 . 또 다른 단점은 XML 문서에 계층적인 포함구조가 MIB와 직접적으로 일치하지 않기 때문에 자동 변환 프로그램의 구현이 어렵다는 것이다.

Metamodel-level mapping은 일반화된 하나의 DTD를 정의하여 모든 MIB에 적용하는 방법이다. 이 모델에서는 MIB에 정의된 변수를 사용하여 Element를 명명하지 않고 일반적인 키워드를 사용한다. 표 2는 Model-level Mapping을 Metamodel-level mapping으로 바꾼 예이다.

표 2. Metamodel-level mapping

```
// SMI를 이용한 MIB의 정의
sysDescr OBJECT-TYPE
SYNTAX DisplayString(SIZE(0..255))
        ACCESS read-only
        STATUS mandatory
        "a textual..."
        ::= { system 1}
//DTD를 이용하여 변환된 XML 문서
<mibs name="system" oid ="1.3.6.1.1">
<objects name="sysDescr" oid="1.3.6.1.1.1"
access="read-only" status="mandatory">
<syntax>DisplayString</syntax>
<description>.....</description>
</objects></mibs>
//DTD를 이용하여 XML 기반의 MIB 정의
<!ELEMENT mibs objects*>
<!ELEMENT objects (syntax, access,
status, description)>
<!ATTLIST objects oid CDATA #REQUIRED
access CDATA #REQUIRED
status CDATA #REQUIRED>
<!ELEMENT syntax #PCDATA>
<!ELEMENT description #PCDATA>
```

이 방법은 가독성이 떨어지는 단점이 있지만, 하나의 DTD를 정의하여 모든 SNMP MIB에 적용이 가능하므로 변환 프로그램이 용이한 장점이 있다.

본 시스템에서 PIB(Policy Information Base)을 XML로 변환할 때 metamodel-level mapping 방법과 유사한 방법으로 변환하였다. 새로운 PRC (Policy Class)을 동적으로 추가할 때 Model-level mapping 방법을 사용하게 되면 매 PRC 추가시 DTD를 같이 추가시키거나 업데이트를 해주어야 하는 단점이 생긴다. 따라서 metamodel-level mapping 방법과 유사한 방법을 제공함으로서 이런 단점을 배제 시켰다.

2.3 정책기반 네트워크 관리 구조

정책기반 네트워크 관리 구조는 네트워크 관리 및 운용를 위한 새로운 기술이다. 본구조는 관리자가 원하는 네트워크 운용을 하이 레벨 정책으로 작

성하면 자동적으로 디바이스 레벨의 구성정보로 맵핑되어 네트워크에 반영되는 것이다. 정책은 어떻게 네트워크를 운용 할 것인지 방법보다는 무엇을 할 것인지를 기술하는 것이다. IETF와 DMTF에서 정의한 정책기반 네트워크 관리 구조는 그림4와 같으며 다음 4개의 컴포넌트로 구성되어진다.

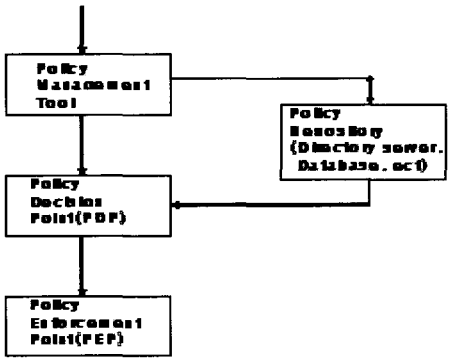

그림 4. IETF/DMTF 의 정책기반 네트워크 프레임워크

Policy Management Tool은 네트워크 관리자와 시스템간의 인터페이스이다. 이것은 정책을 입력 및 수정, 네트워크의 상태를 모니터링 하는데 사용되어진다. Policy Management Tool은 만든 정책이 문법에 맞는지, 정책간에 중복은 없는지 체크하며, 정책들은 Policy Repository의 저장 형태에 맞게 변환되어 저장되게 된다. Policy Repository 는 일반적인 디렉토리 서버로서 Policy Management Tool에서 기술된 정책을 저장하고 있다. Policy Decision Point는 Policy Repository 에 접속하여 저장되어 있는 정책들을 기본으로 정책을 결정한다. Policy Decision Point는 정책의 변화 및 정책의 중복을 탐지하며, 네트워크 디바이스로부터 네트워크의 상태에 대한 이벤트를 기록하며, 네트워크의 사용량을 모니터링 하게 된다. Policy Enforcement Point는 정책을 적용할 수 있는 실제 네트워크 디바이스들이다. 예를 들어 PEP는 라우터, 방화벽 , 스위치 같은 실제 장치들이 될 수 있다. Policy Enforcement Point는 처음으로 부팅될 때 Policy Decision Point로부터 정책 정보를 모으며, 이런 정보를 캐시에 저장하게 된다. Policy Enforcement Point 는 네트워크나 디바이스 상태가 변화되면 PDP에게 알리고 필요한 구성정보를 요청하고 받아오게 된다. Policy Management Tool 과 Policy Repository 는 보통 LDAP(Lightweight Directory Access Protocol)로 통신하게 되며, Policy Decision Point 와 Policy Reposity는 COPS 나 LDAP를 사용하게 된다. Policy Decision Point 와 Policy Enforcement Point는 COPS를 사용하여 정책정보를 주고받게 된다. COPS는 TCP을 기반으로 하고 있으며, 타이머를 통하여 연결을 영속적으로 유지하게 된다.

III. 제안한 시스템 구조

본 논문에서 제안하는 정책기반 네트워크 관리 시스템은 정책 및 PIB의 구조를 XML로 기술함으로써 유연성과 확장성을 고려하였으며, 표준 프로토콜인 COPS을 사용함으로써 각 디바이스에 정책을 적용시킬 수 있도록 고려하였다.

XML은 데이터의 구조를 명시하고, 이 기종의 컴포넌트간에 데이터를 이동하기 위한 표준방법을 제공함으로써, 정책의 구조를 명시하고, 다른 이 기종의 컴포넌트간에 정책을 교환하기에 적합하다. PIB를 XML로 구성할 경우, 동적으로 PRC을 추가 할 수 있으므로, 미리 정의되지 않은 PRC을 구성할 수 있다. 또한 새로운 프로토콜을 생성하지 않고 표준 프로토콜을 사용함으로써 범용성을 높일 수 있다.

시스템의 전체적인 구성은 그림 5와 같다. 각 컴포넌트는 6개로 구성되어진다.

그림 5에서 GUI(Graphical User Interface)는 XML 기반 정책 템플릿 저장소로부터 가져온 XML 기반 템플릿을 이용하여 새로운 정책을 수립하거나 기존 정책을 수정할 수 있는 환경을 제공해준다. 또한 네트워크 관리자와 본 시스템간의 기본적인 인터페이스이다. GUI는 정책에디터, PDP 뷰어, PEP 뷰어로 구성되어진다.

정책에디터는 정책 제너레이션 엔진의 정책 제너레이션 관리자을 이용하여 필요한 XML기반 템플릿을 가져온 후 새로운 정책을 수립하거나, 수정할 수 있는 기능을 제공하며, 그 외에 XML 템플릿 자체를 수정하거나 추가할 수 있는 기능도 포함하고 있다.

PDP 뷰어는 현재 정책서버의 상태를 모니터링 하여 관리자에게 보여주는 역할을 수행한다.

PEP 뷰어은 XML 기반 PIB의 현재 값 및 PEP의 상태정보를 조회할 수 있는 기능을 제공하고 있으며, PIB 컨트롤러를 통하여 새로운 PRC를 추가 시킬 수 있는 기능도 포함하고 있다.

정책 제너레이션 엔진은 정책 제너레이션 관리자

와 데이터베이스 관리자로 구성되어있다. 정책 제너레이션 관리자는 정책에디터를 통해 만들어진 정책을 PDP로 전달하는 역할을 수행하며, 데이터베이스 관리자를 통해 템플릿 과 정책정보를 조회하여 GUI를 통해 관리자에게 보여주는 역할을 수행한다.

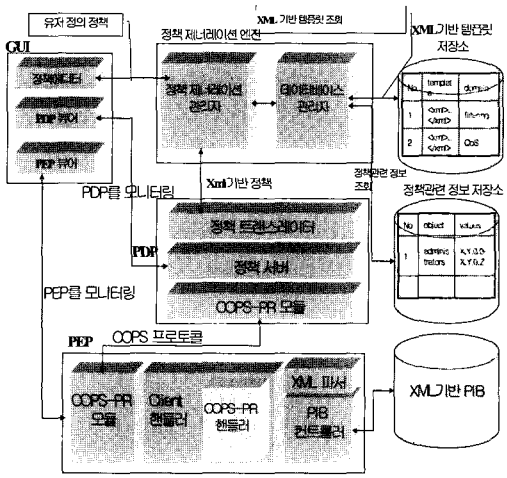

그림 5. XML 기반 PIB을 사용하는 네트워크 관리 구조

데이터베이스 관리자는 JDBC를 통해 데이터베이스로부터 템플릿과 정책정보를 삽입, 수정 ,삭제, 조회 등의 기능을 수행한다.

XML기반 템플릿저장소는 세부 카테고리 별로 정책을 기술할 수 있는 템플릿을 구분하여 저장하고 있는 데이터베이스의 테이블이다.

정책관련정보저장소는 정책과 관련된 정보들을 저장하고 있다. 이것은 관리되는 디바이스와 맵핑 될 수 있는 엔티티를 정의하고 있다.

PDP(Policy Decision Point)는 전형적인 정책기반 네트워크 관리 구조의 컴포넌트중 하나이다. 이 컴포넌트는 정책트랜스레이터, 정책서버 그리고 COPS-PR 모듈로 구성되어 있다.

정책트랜스레이터는 하이 레벨 정책을 디바이스 레벨의 정책으로 변환하는 기능을 수행하며 각 정책에 대한 유효성 검사를 해서 잘못 기술된 정책에 대해 오류 메시지를 관리자에게 보내게 된다.

정책서버는 각각의 PEP들에게 알맞은 정책을 전달하는 역할을 수행한다. COPS-PR 모듈은 PEP에게 보낼 COPS-PR 메시지를 만들거나. PEP로부터 메시지를 받아 그 메시지를 분석하는 역할을 한다. PEP(Policy Enforcement Point) 또한 전형적인 정책기반 네트워크 관리 구조를 구성하는 컴포넌트 중 하나이며, COPS-PR 모듈, 클라이언트핸들러, COPS-PR 핸들러 그리고 PIB 컨트롤러를 포함하고 있다.

COPS-PR 모듈은 PDP의 COPS-PR 모듈과 같은 기능을 수행하며, 클라이언트핸들러는 서로 다른 목적을 가진 클라이언트를 제어한다. COPS-PR 핸들러는 각 COPS-PR 클라이언트를 제어한다.

PIB 컨트롤러는 PEP의 XML 기반의 PIB의 PRI를 추가하거나 삭제하는 기능을 수행하며, 또한 PIB에 새로운 PRC를 추가하는 기능을 포함하고 있다.

Ⅳ. XML 기반의 정책 템플릿과 PIB(Policy Information Base)

4.1 XML 기반 정책 템플릿

본 논문에서는 정책을 기술하기 위해 XML의 사용을 제안하였다.

표 8. XML 스키마를 이용한 템플릿의 예

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">
<xs:element name="message_template">
<xs:complexType>
<xs:sequence>
<xs:element ref="people"/>
<xs:element ref="operation"/>
<xs:element ref="start_time"/>
<xs:element ref="end_time"/>
<xs:element ref="where"/>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="operation" type="xs:string"/>
<xs:element name="people" type="xs:string"/>
<xs:element name="start_time" type="xs:time"/>
<xs:element name="end_time" type="xs:time"/>
<xs:element name="where" type="xs:string"/>
</xs:schema>
```

정책을 XML로 기술함으로써 서로 다른 관리 시스템간 관리 정보의 공유 및 기존 정책을 쉽게 확장할 수 있다. XML 기반 정책 템플릿은 일반적으로 자주 사용되는 정책을 기술함으로써 특정한

정책을 기술할 때 이 템플릿을 이용하여 쉽게 작성할 수 있다. 표 3 은 XML Schema을 이용하여 만든 정책 템플릿의 예를 보여주고 있다.

본 XML 스키마는 다섯 개의 일반적인 엘리먼트들을 포함하고 있다.

1) people : 일반적으로 작업자를 기술한다.

2) operation : 컴퓨터 상에서 어떤 작업을 할 것인지를 기술한다.

3) start_time : 작업을 시작하는 시간을 기술한다.

4) end_time : 작업을 끝내는 시간을 기술한다.

5) where : 작업을 하는 장소를 기술한다.

각 엘리먼트는 관리 도메인 상에서 오브젝트를 표현하고 있다.

표 4. 템플릿을 이용해 기술한 정책

```
<?xml version="1.0" encoding="UTF-8"?>
<message_template xmlns:xsi="http://www.
w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="C:\XMLP
olicyBasedSystem\policschema.xsd">
<people>administrator</people>
<operation>Internet</operation>
<start_time>08:00</start_time>
<end_time>17:00</end_time
<where>every<where></message_template>
```

표 4는 위의 템플릿을 이용한 특정 정책을 기술하고 있다. 정책의 의미는 다음과 같다.

"관리자는 모든 장소에서 로그인할 수 있으며, 작업시간(08:00 ~ 17:00)동안 인터넷을 사용할 수 있다."

4.2 XML을 사용한 PIB 변환

앞서 언급한 바와 같이, 본 논문에서는 XML 기반의 PIB를 사용하고 있다. 본 논문에서 SPPI 기반의 PIB를 XML 기반의 PIB로 변환하기 위해서 메타모델레벨(Metamodel-level) 맵핑 방법과 유사하게 하였다. 메타모델레벨 맵핑 방법은 앞에서 언급한 바와 같이 일반화 된 하나의 DTD를 정의하여 모든 PIB에 적용하는 방법이다. 이 모델을 PIB에 적용시킬 경우 PIB에 정의된 변수 대신 일반적인 키워드를 사용하여 정의하게 된다. 일반적인 PIB는 SPPI을 이용하여 기술되어 있다. SPPI는 RFC 3159[13]에 정의되어 있으며, PIB를 정의하기 위한 수많은 데이터 타입 및 구조체를 정의하고 있다.

표 5. PIB의 구조를 정의하기 위한 DTD

```
<?xml version="1.0" encoding="UTF-8"?>
<!ELEMENT PIB (Tables+)>
<!ATTLIST PIB oid CDATA #FIXED
"1.3.6.1"><!ELEMENT Tables (Entrys)>
<!ATTLIST Tables name CDATA #REQUIRED
          oid CDATA #REQUIRED
          method CDATA #REQUIRED>
<!ELEMENT Entrys (Prid+)>
<!ATTLIST Entrys name CDATA #REQUIRED
        oid CDATA #REQUIRED>
<!ELEMENT Prid (Objects+)>
<!ATTLIST Prid SYNTAX CDATA #REQUIRED
ACCESSMODE CDATA #REQUIRED
          index CDATA #REQUIRED
          oid CDATA #REQUIRED>
<!ELEMENT Objects (#PCDATA)>
<!ATTLIST Objects
name CDATA #REQUIRED
SYNTAX CDATA #REQUIRED
ACCESSMODE CDATA #REQUIRED
oid CDATA #REQUIRED>
```

본 논문에서는 SPPI 기반의 PIB을 DTD(Document Type Definition)을 사용하여 XML 기반의 PIB로 변환시켰다. DTD는 XML문서의 컨텐츠의 허용값을 포함하여 문법 및 구조를 정의하기 위해 사용된다.

표 5, 6은 본 논문에서 사용하고 있는 DTD 와 XML 기반 PIB를 보여주고 있다.

표 5의 DTD는 PIB, Table, Entrys, Prid 와 Objects 의 5개의 대표적인 엘리먼트로 구성되어 진다. PIB는 적어도 1개 이상의 테이블로 구성되어지며, 테이블은 한 개의 엔트리로 구성되어 진다. 엔트리는 적어도 1개 이상의 Prid로 구성되어지며, 한 개의 Prid는 여러 개의 오브젝트로 구성되어진다. 여기서 오브젝트란 표 6에서와 같이 송신자의 아이피 주소, 넷 마스크 등과 같은 Prid을 구성하는 속성 값이라 할 수 있다.

표 6. DTD를 이용해 정의한 PIB

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE PIB SYSTEM "pib.dtd">
<PIB oid="1.3.6.1">
<Tables name="ipv4PriorityTable" oid="1.3.6.1.1"
method ="priorityClass">
<Entrys name ="Ipv4Priority" oid ="1.3.6.1.1.1">
<Prid SYNTAX="Unsigned32" ACCESSMODE
="read-write" index= "1" oid="1.3.6.1.1.1.1">
<Objects name="SourceIp" SYNTAX="IpAddress"
ACCESS MODE="read-write" oid="1.3.6.1.1.1.2">
163.180.118.99</Objects>
<Objects name="SourceMask" SYNTAX="IpAddress"
ACCESS MODE="read-write" oid="1.3.6.1.1.1.3">
255.255.255.0</Objects><Objects
name="SourcePort" SYNTAX="Integer32" ACCESS
MODE="read-write" oid="1.3.6.1.1.1.4">80</Objects>
<Objects name="DestinationIp" SYNTAX="IpAd
dress" ACCESS MODE="read-write" oid="1.3.6.1.
1.1.5">255.255.255.255</Objects>
<Objects name="DestinationMask" SYNTAX="IpA
ddress" ACCESS MODE="read-write" oid="1.3.6.1.
1.1.6">255.255.255.255</Objects>
<Objects name="DestinationPort" SYNTAX="Inte
ger32" ACCESS MODE="read-write" oid="1.3.6.1
.1.1.7">0</Objects>
</Prid></Entrys></Tables></PIB>
```

본 논문에서 메타모델과 유사한 방법을 사용함으로써 PRC의 동적인 추가를 쉽게 구현할 수 있었으며, PIB Controller의 구현도 쉽게 할 수 있었다. 하지만 가독성에서 많은 단점을 가지고 있다. DTD로는 PIB의 구조외에 어떤 종류의 PRC인지 구별할 수 없으며, PRI가 몇 개의 속성(오브젝트)로 이루어지는지 알 수 없다.

V. 구현

본 장에서는 앞에서 언급한 제안구조의 구현 사항에 설명한다. 본 구조의 구현은 JDK1.3.1 및 JBuilder 7을 이용했으며, 운영체제는 Windows XP 기반으로 하였다. Database는 오라클 9i를 사용하였다. 본 구조는 캐나다 워터루 대학 과 스페인 발렌시아 대학이 공동으로 구현한 MetaPolicies을 이용한 정책기반 네트워크 프레임워크[25]을 기반으로 작성되었다.

구현의 이해를 돕기 위해 정책을 기술하는 부분, PDP와 PEP 부분으로 나누어서 설명토록 한다.

5.1 정책을 기술하는 부분

정책을 기술하는 부분은 앞서 제안한 구조에서 Policy Generation Engine, Policy Editor, JDBC을 포함한 DatabaseManger, Policy Temp -late와 Policy Information Repository로 구성되어진다. GUI는 XML 에디터와 위저드 방식을 지원한다. XML 에디터는 XML에 대해 잘 다룰 수 있는 고급사용자를 위해 만들어졌으며, 위저드 방식은 망 및 XML에 대해 알지 못하는 일반사용자를 위해 만들었다. 즉 사용자에게 따라 다양한 뷰를 제공함으로써 일반사용자들도 쉽게 정책을 기술할 수 있도록 구성하였다. 그림 6은 정책을 기술하는 부분에 대한 클래스 다이어그램이다.

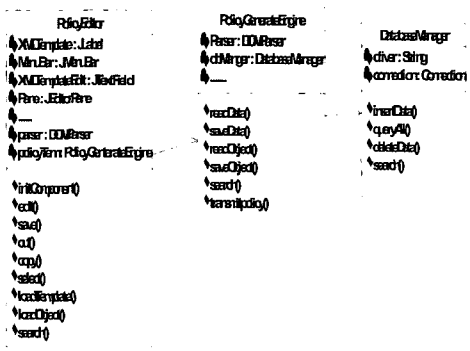

그림 6. 정책을 기술하기 위한 클래스 다이어그램

Policy Editor는 템플릿 및 정책관련 정보의 수정, 삭제, 저장의 기능을 포함하고 있으며, 위저드 방식은 각 템플릿 목록 및 정책정보를 단계별로 선택함으로서 쉽게 작성할 수 있다.

그림 7은 Policy Editor의 실행화면이다. 정책 작성 위저드의 실행화면은 그림 8와 같다.

5.2 PDP(Policy Decision Point)

PDP는 앞서 언급한 바와 같이 정책 기반 네트워크의 한 요소로서 PEP에게 구성정보를 보내주고 필요할 때 구성정보를 업데이트 하는 기능을 수행한다.

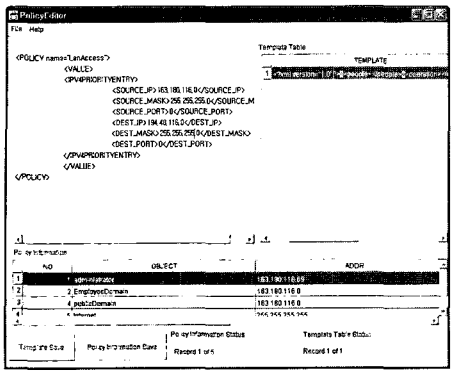

그림 7. Policy Editor의 실행화면

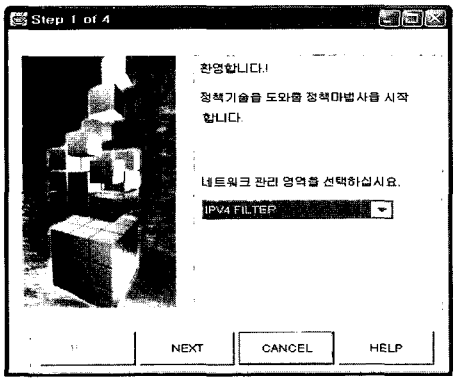

그림 8. 정책 작성 위저드의 실행화면

PDP의 기능의 핵심은 Policy Server Class 이다. 이 클래스는 PEP와 통신을 수행하기 위해 필요한 클래스의 오브젝트들을 참조하고 있다. 예를 들자면 Policy Server 클래스는 들어오는 요청을 위해 ServerSocket을 가지고 있으며, 들어오는 COPS 메시지와 오브젝트들을 처리하고 디코딩하기 위해 CopsMessageReceiver 객체를 참조하고 있다. 그리고 CopsMessageReceiver에서 디코드된 객체를 위해 COPSMessage 객체를 참조한다.

정책기반 네트워크 구조에서 PDP는 현재 네트워크의 상태를 따라 지능적으로 정책을 결정하며 각 PEP 상태 및 요청에 따라 정책을 업데이트 해주는 역할을 수행함으로 매우 복잡하게 구성되어 있다. 이런 완전한 PDP의 구현은 본 논문의 범위를 벗어나므로, 본 제안구조에서 꼭 필요한 기능을 테스트하기 위한 사항에 대해서만 구현하고 있다. 구현된 PDP는 PEP와 통신하기 위한 COPS 와 COPS-PR의 메시지 및 오브젝트들을 포함하고 있으며, Policy Translator로부터 구성정보를 받아 PEP에게 전달하는 기능을 가지고 있다. 앞서 언급한 바와 같이 구현의 복잡성을 줄이기 위해 Policy Translator의 기능도 본 논문에서 기술된 정책만을 파싱할 수 있도록 파서를 제작하였다.

그림 9는 PDP의 클래스 다이어그램을 보여주고 있다.

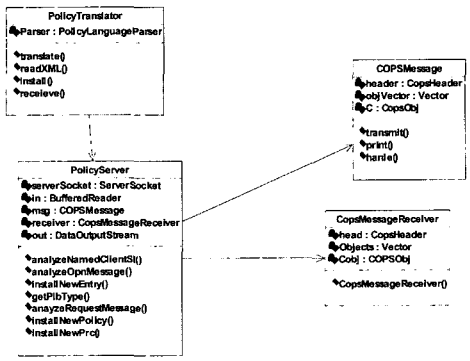

그림 9. PDP의 클래스 다이어그램

5.3 PEP(Policy Enforcement Point)

PEP는 PDP와 통신 세션을 열고, PDP로부터 구성정보 및 업데이트 메시지를 기다린다. PEP는 PDP로부터 받은 메시지의 형태에 따라 PIB에 구성정보를 저장, 업데이트 그리고 없애는 역할을 수행한다. 하나의 PEP에는 하나 이상의 클라이언트들이 존재할 수 있으며, 각 클라이언트 관리 목적에 따라 COPS-PR 및 COPS-RSVP 등이 존재한다. PEP에 구현에서 가장 중요한 클래스는 PEP 이다. PEP 클래스는 PEP 기능에 필요한 관련된 클래스를 참조하고 있다. COPSCom 클래스는 COPS 프로토콜과 관련된 기본적인 통신 기능을 가지고 있는 클래스이며 PEP 클래스에 의해 참조되어 진다. 하나의 PEP에는 다수의 클라이언트가 존재할 수 있다. 따라서 PEP는 다수의 클라이언트를 다루기 위해 ClientHandler 객체의 배열을 가지고 각 클라이언트들을 참조하고 있다. 또한 ClientHandler 클래스는 서로 다른 목적의 클라이언트를 다룬다. 본 논문에서는 COPS-PR에 국한되어 구현하고 있다. 따라서 ClientHandler는 COPS-PR Client 클래스를 참조하고 있다.

COPS-PR Client 클래스는 COPS-PR 프로토콜과 코드 등을 포함하고 있으며, 프로토콜의 메시지,

오브젝트 등을 다룬다. COPS-PR Client 클래스는 PIBController 클래스를 참조하고 있다. PIBController 클래스는 받은 메시지에 따라 PIB의 구성정보를 설치, 삭제, 업데이트 기능 등을 포함하고 있다. 따라서 PIBController 클래스는 PIB 클래스를 참조하고 있다. 즉 PIB Controller 클래스는 PIB 클래스의 기능을 통해 PIB을 다루게 된다. 본 논문에서는 XML 기반의 PIB를 사용하기 때문에 XML을 다룰 수 있는 파서를 내장하고 있다. 그림 10은 PEP 클래스의 다이어그램을 보여주고 있다.

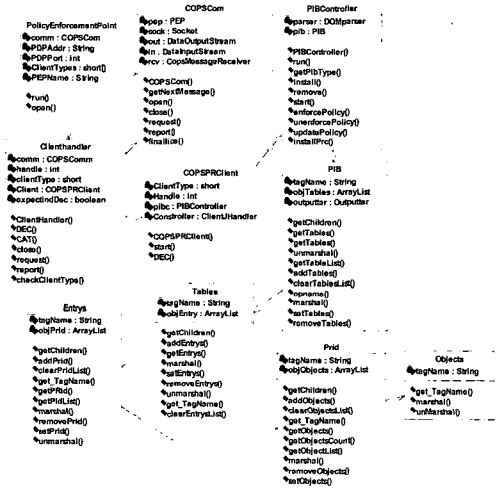

그림 10. Policy Enforcement Point의 클래스 다이어그램

VI. 운용 시나리오

본 논문에서 제안된 구조의 동작과정 및 새로운 PRC(Provisiong Class)의 추가를 보여주기 위한 간단한 실험을 하였다. 현재 PEP의 PIB에는 필터링을 하기 위한 PRC만 존재하고 있다. PDP 정책은 표 7과 같이 3개의 정책을 PEP에 전송하게 된다.

1) 모든 내부 트래픽은 허용한다.

2) 작업시간동안은 인터넷을 허용한다.

3) 작업시간동안 관리자는 데이터베이스서버에 접속할 수 있다.

그림 11은 SPPI(Structure of Policy Provisio -ning Information)-based PIB을 사용한 정책기반 네트워크 관리 시스템동작과정의 순서를 보여주고 있다.

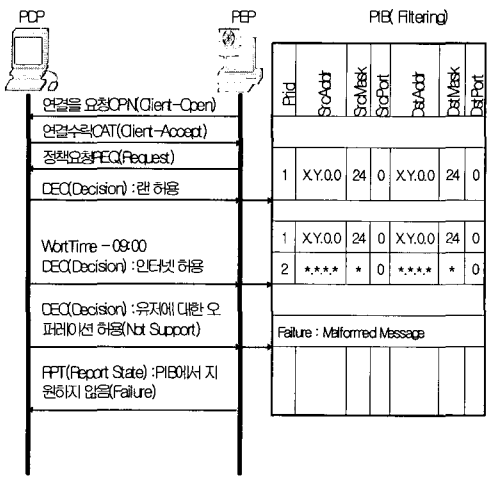

그림 11. SPPI-based PIB을 사용한 정책기반 관리 시스템의 동작과정

그림 11에 보여지는 것과 같이 기존의 PIB을 사용하는 정책기반 관리 시스템은 PIB에 미리 정의되어져 있는 PRC에 대해서만 정책을 적용할 수 있으므로 새로운 PRC의 추가 및 새로운 정책을 적용할 수 가 없었다. 따라서 PIB에서 지원하지 않은 정책에 대해선 Report 메시지를 통해 에러를 PDP에게 보고하게 된다.

본 제안구조는 XML 기반의 PIB를 사용하고 있으므로, 동적으로 새로운 PRC의 추가 및 추가된 PRC에 새로운 정책을 적용할 수가 있다. 그림 12는 제안구조의 동작 과정을 보여주고 있다.

그림 12와 같이 본 구조에서는 PDP에서 PIB에서 지원하지 않는 정책을 보내 PEP로부터 RPT (Report) 메시지를 통해 에러 메시지를 받으면 새로운 PRC 추가에 대한 Decision 메시지를 보내 현재 PIB에 새로운 PRC를 추가하게 된다. 그리고 추가된 PRC에 추가할 정책을 다시 보내 적용하게 된다.

그림 13은 그림 12의 (1)번 과정에서 PDP가 보낸 Decision 메시지의 모습이며, 그림 14는 그림 12의 (2)번 과정에서 PDP가 Report 메시지를 통해 에러 메시지를 받은 모습이다.

그림 15는 그림 12의 (3)번 과정에서 PEP가 새로 추가된 PRC에 PRI를 추가하고 난 후 PIB의 내용 및 PDP에게 RPT 메시지를 보내는 모습이다. 그림 16은 PEP로부터 PIB에 PRI가 성공적으로 인스톨되었다는 RPT(Report) 메시지를 받은 모습이다.

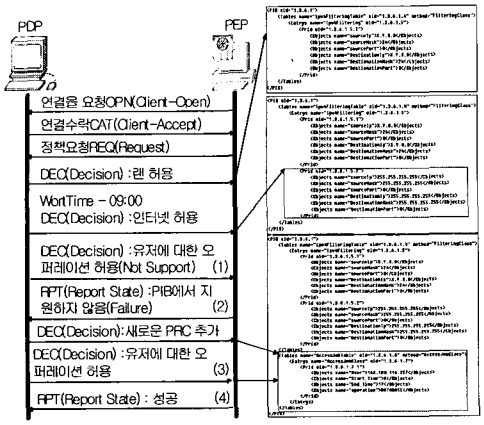

그림 12. XML-based PIB를 사용하는 제안구조의 동작 과정

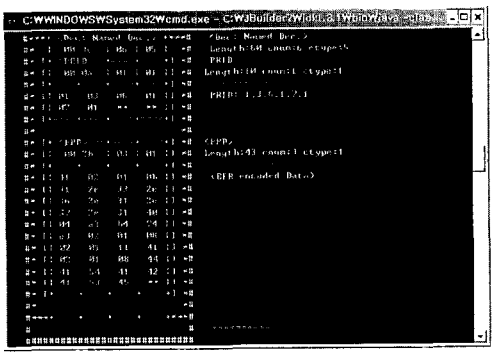

그림 13. 그림 12. (1)번 과정(PDP의 화면)

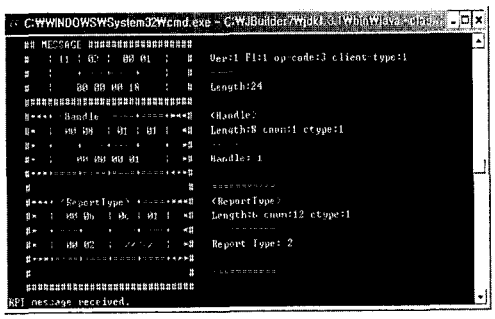

그림 14. 그림 12. (2)번 과정의 Report 오류 메시지

Ⅶ. 평 가

기존 SPPI 기반 PIB을 사용하며 XML이 아닌 어플리케이션에 종속적인 정책으로 기술된 네트워크 관리 시스템은 다른 관리 시스템과 정책을 교환할 수 없으며, 미리 정의되어진 오퍼레이션만을 정책을 통해 수행함으로써 유연성과 확장성에서 많은 문제가 야기된다. 기존 시스템에서 새로운 오퍼레이션을 추가하기 위해서는 시스템을 정지시킨 후 PIB를 패치 및 업데이트를 한후 시스템을 재 가동 시켜야 한다.

그림 15. 그림 12. (3)번 과정
(PIB의 값 및 PDP에게 RPT 메시지 전송)

그림 16. 그림 12. (4)번 과정
PEP로부터 받은 RPT 메세지

본 논문에서 제안하고 있는 XML 기반 PIB을 사용한 정책기반 관리 구조는 미리 정의된 오퍼레이션 외에 동적으로 새로운 오퍼레이션을 추가할 수 있으므로 시스템에 정지 없이 확장이 가능하다. 또한 본 구조에서 정책 기술시 XML을 이용함으로써 서로 다른 정책기술 언어를 사용하는 이 기종의 관리 시스템간에 정책 교환 시 XSLT(eXtensible Stylesheet Language Transformation)을 이용하여 쉽게 변환될 수 있다. 즉 하나의 XML로 기술된 정책 디렉토리 서버에서 XSLT을 이용하여 각

각의 관리 시스템에 맞는 정책 포맷으로 변환되어 이 기종의 관리 시스템들의 정책서버로 정책이 보내어 지는 것이다. XSLT는 각각의 관리시스템의 정책포맷에 따라 미리 작성되어 있어야 한다.
표 7은 기존 정책기반 관리 시스템과 본 논문에서 제안하는 시스템과의 기능상 차이를 비교 분석한 것이다.

표 7. 기존 시스템과 제안구조 비교표

관리 구조 / 기능		기존 정책기반 관리 구조 (SPPI based PIB)	제안 구조 (XML based PIB)
PIB 관련 사항	PRI 추가	가능	가능
	PRI 삭제	가능	가능
	PRC 추가	불가능(시스템 정지 후 가능)	가능(동적 확장)
	PRC 삭제	불가능(PRI를 삭제함으로서 사용안함)	가능
프레임 워크 관련 사항	처리 시간	동일	동일
	정책 교환	불가능	가능(XSLT를 이용 변환)
	메시지 교환 횟수	동일	동일

Ⅷ. 결 론

XML은 모든 하드웨어와 소프트웨어 플랫폼에서 지원이 가능하기 때문에 어떤 장비에 구애받지 않고 정보교환을 할 수 있다. 이런 여러 가지 장점으로 인하여 XML은 기존 네트워크 관리 시스템에서 문제점으로 지적되던 확장성, 유연성 그리고 효율성 문제를 해결할 수 있는 대안으로 대두되고 있다.

본 논문에서는 XML을 통해 정책의 기술 및 PIB의 구조를 정의함으로써 유연성과 확장성을 개선한 네트워크 관리 구조를 제안하였다. 또한 일반적인 정책을 템플릿으로 미리 정의함으로써 네트워크의 사전지식을 가진 유저가 아니라도 쉽게 정책을 수립할 수 있으며, 에디터 방식과 위저드 방식을 제공함으로써 유저에 따라 편리하게 정책을 작성할 수 있게 하였다.

기존 SPPI 기반 PIB에서는 미리 정의되어 있는 PRC에 대해서만 PRI를 추가함으로써 정책을 PEP에게 반영할 수 있었다. 본 구조에서는 XML 기반 PIB를 이용함으로써 동적으로 새로운 PRC를 추가했으며, 추가된 PRC에 PRI를 추가함으로써 새로운 기능을 정책을 통해 수행할 수 있었으며, 간단한 테스트를 통해 본 제안구조의 개선점을 입증하였다.
본 논문에서는 네트워크 관리 구조에 XML을 적용함으로써 얻을 수 있는 장점을 설명하고 있으며, 정책기반 네트워크 관리 시스템에 구현사항에 대해 언급하였다. 향후 연구과제로는 추상레벨의 정책을 쉽게 관리 디바이스에게 적용될 수 있는 정책으로 맵핑될 수 있는 효율적 방안 및 PEP의 기능성을 높이기 위해 제안된 Meta-Policy PIB[14]를 이용하여 PEP의 기능성을 향상시킬 수 있도록 연구할 예정이다. 본 논문에서는 COPS프로토콜을 지원하는 PEP에 대해서만 기능을 구현하고 있으며, 앞으로 COPS 외에 SNMP, CLI 등 다양한 관리 프로토콜을 지원하도록 확장할 계획이다.

참 고 문 헌

[1] Mark L. Stevens, Watler J. Weiss, "Policy-based Management for IP Networks",*Bell Labs Technical Journal*,1999

[2] D.Durham et al.,"The COPS(Common Open Policy Service) Protocol", IETF, *RFC2748* ,Jan.2000, http://www.ietf.org/rfc/rfc2748.txt

[3] K.Chan et al.,"COPS Usage for Policy Provisioning," IETF, *RFC3084* , Mar.2001, http://www.ietf.org/rfc/rfc3084

[4] W3C, "Extensible Markup Language", http://www.w3.org/XML/

[5] M.Fine et al.,"Framework Policy Information Base", IETF, Internet- Draft, *draft-ietf-rap-frameworkpib-04.txt*, Nov. 2000, http://www.ietf.org/internet-drafts/draft-ietf-rap-frameworkpib-04.txt

[6] DMTF, "Common Information Model(CIM) Specification Version 2.2", June 14, 1999

[7] DMTF " CIM Schema Version 2.6 " 7, 2001

[8] DMTF, "Specification for the Representation of CIM in XML", Version 2.0, July 1999

[9] DMTF, "XML as a Representation for management Information", *A White Paper Version 1.0*,

September 1998
[10] "Understanding LDAP",http://www.redbooks.ibm.com/pubs/pdfs/redbooks/
[11] M.Wahl ," A Summary of the X.500(96) User Schema for use with LDAPv3" December 1997.
[12] A.Westerinen et al.,"Terminology", IETF, Internet draft, *draft-ietf-policy-terminology -02.txt*, Nov.2000, http://www.ietf.org/inter net-drafts/draft-ietf-policy-terminology-02.txt
[13] K. McCloghire et al.,"Structure of Policy Provisioning Information [SPPI]", IETF, Internet draft,*draft-ietf-rap-frameworkpib-06 .txt,* Feb. 2001,http://www.ietf.org/internet-drafts/draft-ietf-rap-frameworkpib-06.txt
[14] R. Boutaba et al.,"The Meta-Policy Inform ation Base(M-PIB)", IETF, Internet draft
[15] M.Casassa Mont et al.,"POWER Prototype : Towards Integrated Policy-Based Manag -ement", NOMS 2000, April 2000,
[16] Natarajan, R et al.,"A XML based policy-driven management information ser -vice ",*IM* 2001
[17] J. P Martin-Flatin, "Web-based management of IP Networks and Systems", Ph.D thesis, *Swiss Federal Institute of Technology, Lausanne (EPEL),* Oct. 2000
[18] International Organization for Standardization, "ISO 8879 : Standard Generalized Markup Language(SGML), 1986
[19] W3C, "Extensible Stylesheet Language(XSL) Version 1.0", Recom -mendation, Oct 2001, http://www.w3c.org/Style/XSL/
[20] W3C, Extensible Markup Language(XML) 1.0 -DTD, W3C Recommendation, October 2000, http://www.w3.org/TR/REC-xml#dt-doctype
[21] W3C, "XML Schema Part 0: Primer", Reco -mmendation, May 2001, http://www.w3.org/TR/xmlschema-0/
[22] 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
[23] 윤정혁, 주홍택, 홍원기, "XML 기반 네트워크 관리를 위한 SNMP- XML 변환기 및 게이트 웨이", *KNOM Review* 제5권 1호, June 2002
[24] DMTF, "WBEM Initiative", http://www.dmtf.org/wbem
[25] R. Boutaba et al.,"Implementation of a policy based network frame work using metapolicies", December 2001
[26] Ritu Chadha, "Directory-Enabled Networking", *NOMS 2000 tutorials* 8 10-14 April 2000
[27] Apache XML project, "Xalan-XSLT processor", http://xml.apache.org /xalan-j/index.html

윤 권 섭(Kwoun-Sup Youn) 정회원

2001년 2월 : 경희대학교 전자계산공학과 졸업
2003년 2월 : 경희대학교 컴퓨터공학과 석사과정졸업

<주관심분야> 망 관리 시스템, XML기반 데이터베이스, 분산시스템

홍 충 선(Choong-Seon Hong) 종신회원

1983년 2월 : 경희대학교 전자공학과 졸업
1985년 2월 : 경희대학교 전자공학과 석사
1997년 2월 : Keio Univer sity, Department of Information and Computer Science 박사
1988년 ~1999 년 : 한국통신 통신망 연구소 선임연구원/ 네트워킹연구실장
1999년 ~현재 : 경희대학교 전자정보학부 조교수

<주관심분야> 인터넷 서비스 및 망 관리 구조, 네트워크보안, Mobile Computing