

통합 망관리를 위한 Proxy 설계에 관한 연구

정회원 박주희*, 박승균*, 오영환*

A Study on the Design of Proxy for Integrated Network Management

Ju-hee Park*, Seung-kyun Park*, Young-hwan Oh** *Regular Members*

요 약

컴퓨터 통신망이 보다 복잡하게 되고 사용자가 원하는 서비스도 다양화됨으로 컴퓨터 통신망을 효율적으로 관리하는 기능이 필요하게 되었다. 이러한 이유로 각 통신망 관련 기업, 연구기관 및 표준화 기구에서는 보다 효율적인 망 관리 프로토콜을 제안하게 되었다. CMIP과 SNMP는 대상이 되는 통신망이 서로 다르기 때문에 특성, 구조 및 기능에서 상당한 차이점을 보이고 있다. 그러므로, 가장 널리 사용되고 있고, 또한 추후 계속적으로 사용될 두 통신망 관리 프로토콜을 위한 망 관리 프로토콜의 서로 다른 특성들을 연동할 수 있도록 하는 시스템의 구현은 두 통신망의 완전한 연동을 위하여 필수적이다. 본 논문에서는 이러한 상호 연동을 위해서 프록시를 통한 통합 망 관리를 제시하였다. 프록시 설계에 있어서 내부에 캐쉬를 이용하여 기존 설계의 비효율적인 관리 조작을 개선하였으며, CMIP의 스코핑 기능을 사용하여 하위 레벨에 있는 정보를 한꺼번에 가져오는 방법을 사용하였다. 그러므로, 불필요한 관리 조작을 상당히 감소시키는 물론 관리 시스템 전체 응답 시간을 줄이는 효과를 가져왔다.

ABSTRACT

The technology to efficiently manage computer communication network is in need as the computer communication network becomes more complicated and users want services more diversified. This has brought industries, research institutes and standardization organizations to consider an effective network management protocol. CMIP and the SNMP have considerable differences in structure and capacity. Therefore, it is essential to have enables differences between network management protocol to gear for the two communication management system which is the most widely used and will be continuously used. In this thesis we suggest an interworking function Integrated Network Management for the Design of Proxy. According to the suggest algorithm, by using scoping, it stores the management information, so that it may pass the related information at the inner cash efficiently. Therefore, we can not only reduce some unnecessary management operation considerably but also bring some efficiency of saving the total management system response time.

1. 서론

정보화 사회로 발전하면서 통신 서비스 이용자들은 보다 신속하고 다양한 서비스를 요구하게 되었고, 이에 부응하여 통신망은 빠르게 진화되어 가고

있다. 특히, 통신망의 고도화, 지능화에 따라 수동 및 개별 통신망 운용 관리 체계에서 집중화, 자동화된 관리 체제로의 전환과 단위 시설의 운용 보전 개념에서 전체적인 통신망 및 사업 관리 방향으로 발전되는 등 통신망의 관리 방식과 개념도 변화되

* 권운대학교 전자통신공학과 통신망연구실(y2k@kwangwoon.ac.kr)
 논문번호 : 010142-0615, 접수일자 : 2001년 6월 15일

고 있다. 또한, 통신망의 비대화, 복잡화에 따라 운용 능률 제고와 경쟁력 강화를 위해 효율적인 운용 기반 구축이 필요하게 되었다. 이를 기반으로 통신망 운용 관리 업무의 생산성 제고, 경제적인 통신망 운용 관리, 통신 시설의 가용도 향상 및 신속한 통신 서비스 제공이 요구되었다. 다수의 공급자로부터 제공되고, 서로 상이한 특성을 갖는 통신 설비들로 구성된 통신망을 효율적으로 운용 및 유지 보수하는 것은 상당히 어려운 문제이다. 즉, 통신망 구성 요소들은 공급 사마다 특정한 정보 모형과 인터페이스를 사용하는 경향이 있기 때문에 상호 운용성을 유지하면서 통신망 전체차원에서 통합하여 관리하기가 쉽지 않다. TMN에 걸친 프로토콜이나 관리 정보 모델은 OSI관리의 표준에 기초하고 있다. 한편, LAN기기등의 관리는 SNMP(Simple Network Management Protocol)가 업계의 표준으로 정착해가고 있다^[1]. 이러한 배경의 기본 CMIP와 SNMP에 각각 따르는 장치·기기가 혼재하는 네트워크로 통합적인 관리를 행할 수 있도록, OSI관리의 관리자로부터 SNMP의 관리대상을 감시/제어하는 형태의 방식이 지금까지 몇 개 보고되어 있다. 하지만, 현재 SNMP기기 및 그것들을 대상으로 한 관리 어플리케이션이 수없이 존재하는 것에 대해 TMN장치의 도입은 아직 극히 적은 상황이다. 이 때문에 기존의 관리 어플리케이션의 계속성을 꾀하기에는 상기의 형태보다도, CMIP를 SNMP로 변환해서 수용하는 형태가 현실적인 방법이라고 생각할 수 있다. 이 형태를 실현하는 방식으로서, NMF(Network Management Forum) IIMC(ISO/ITU-T & Internet Management Coexistence)가 있다. 여기에서는 기존의 CMIP를 SNMP의 관리대상으로서도 사용할 수 있도록 SNMP의 프로토콜과 관리정보정의에서 인터페이스도 동시에 갖게 하는 방식을 취하고 있고, 그 때문에 필요해지는 OSI관리와 SNMP의 MIB(Management Information Base)와의 대응 철부를 규정하고 있다. 하지만 이 방식에서는 CMIP 장치에 SNMP 관리대상의 인터페이스의 기능추가가 곤란한 경우에는 적용할 수 없다. 또 가령 적용할 수 있는 경우라 할지라도, 그 인터페이스를 제공하기 위한 처리 부하가 증대하여, 본래 제공해야 할 서비스에 악영향을 줄 가능성이 있다는 문제점이 있다. 거기에서 본 논문은 기존의 CMIP에 조금도 변경을 가하지 않고, 다른 워크스테이션 등의 장치 후에, SNMP와 OSI 관리 사이에 관리조작이나 관리정보의 변경을 행함에 의해, SNMP 관리자로부터

CMIP 장치의 감시/제어를 가능하게 하는 SNMP/CMIP 관리 프로시저를 설계했다. 본 논문의 구성은 다음과 같다. 제2장에서는 SNMP와 CMIS/CMIP의 개념과 기존 연구에 대한 고찰을 알아보고, 3장에서는 Proxy설계 그리고 4장에서는 제안한 알고리즘을 살펴본 후 5장에서는 예상효과를 알아본다. 제6장에서 본 논문의 결론을 맺는다.

II. 관계 이론

1. SNMP의 개요

SNMP는 TCP/IP 프로토콜 수행자의 일부분인 응용 레벨 프로토콜이 되도록 설계되었다. 이것은 UDP(User Datagram Protocol)상에서 운영되도록 하였다. 독자적인 관리 스테이션을 위해 관리자 프로세스는 관리 스테이션에 있는 중앙 MIB에 대한 액세스를 제어하고 네트워크 관리자에 대한 인터페이스를 제공한다. 관리 프로세스는 UDP/IP와 관련 네트워크-중속 프로토콜의 위에서 실행되는 SNMP를 사용함으로써 네트워크 관리를 얻을 수 있다. 각 관리대상은 또 SNMP, UDP 그리고 IP도 실행해야 한다. 덧붙여 SNMP의 메시지를 해석하고 관리대상의 MIB를 제어하는 관리대상 프로세스가 있다. 다른 응용들을 지원하는 관리대상 장치는 FTP, TCP 뿐만 아니라 UDP가 요구된다. 그림 1은 SNMP의 프로토콜 환경을 나타내고 있다. 관리 스테이션으로부터 관리 응용의 동작을 나타내는 세 가지 유형의 SNMP 메시지가 발행되는데, 이는 GetRequest, GetNextRequest 와 SetRequest이다. 처음 두 개는 Get기능의 변화이다. 이 세가지 메시지는 전부 관리 응용까지 전해지는 GetResponse 메시지 형태로 프

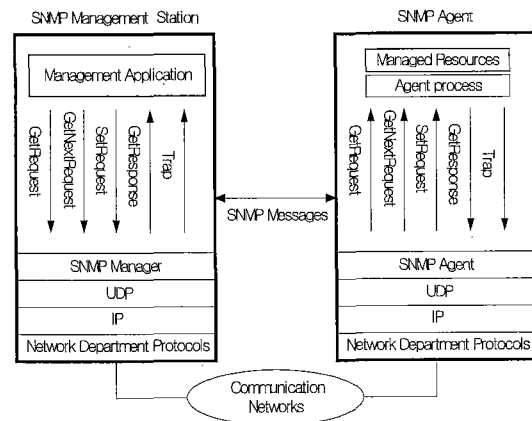


그림 1. SNMP 역할

록시에 의해 알려지게 된다. 덧붙여 프록시는 MIB와 현재 관리되고 있는 자원에 영향을 준 사건에 대하여 Trap 메시지를 내 보낼 수 있다.

SNMP가 비연결형 프로토콜인 UDP에 의존하기 때문에 SNMP 그 자체는 비 연결형이다. 관리 스테이션과 그것의 관리대상간에 진행중인 연결이 유지된 것은 없다. 그 대신에 각 교환은 관리 스테이션과 관리대상간에 분리된 트랜잭션이다.

2. CMIS/CMIP의 개요

OSI 7계층 프로토콜을 기반으로한 모든 망자원들의 관리를 목표로 ISO/IEC와 ITU-T에 의해 개발되고 있는 표준안이다. 객체들 사이의 통신은 OSI 표준 망관리 서비스 및 프로토콜인 CMIS/CMIP을 이용하여 수행된다.

OSI 망관리의 기본적인 기능은 망 관리 프로토콜을 사용하여 두 개체(관리자, 프록시)사이에서 망 관리 정보를 교환하는 것이다. 이러한 기능들을 CMISE(Common Management Information Service Element)라고 하는데, 사용자와의 인터페이스로서 제공되는 서비스를 기술하고 있는 부분인 CMIS와 망 관리 프로토콜로서 PDU의 형식과 절차를 기술하고 있는 부분인 CMIP 두 부분으로 나뉜다.

3. SNMP와 CMIP의 비교

TCP/IP의 망 관리 프로토콜인 SNMP와 OSI 망 관리 프로토콜인 CMIP은 상호 다른 기능을 갖고 있다. SNMP MIB는 ASN.1표기법을 사용하여 간단한 값들을 표현하며, 스칼라 형태의 이차원 배열을 이루고 있으며, 객체 정의의 재사용이 불가능하다. 반면 CMIP MIB는 GDMO를 사용하여 일관성과 객체 지향 개념으로 표현하므로 복잡하고 적용하기 어려운 단점이 있으나, 클래스의 재사용성이 뛰어나다. 뿐만 아니라 SNMP MIB는 단위/원자적 데이터에 대해서만 캡슐화가 가능하고, CMIP MIB는 속성, 행위, 통지 등의 템플릿 단위로 캡슐화가 가능하다^[2].

SNMP와 CMIP에 대한 비교를 다음 표 1에 정리하였다.

4. 기존 연구에 대한 고찰

4.1 관리정보(MIB) 변환을 위한 연구

IIMC에서는 관리정보 변환을 위해서 SNMP MIB를 CMIP MIB로 변환하는 데 필요한 IIMCIMBTRANS와 CMIP MIB를 SNMP MIB

표 1. SNMP와 CMIP의 비교

특징	SNMP	CMIP
망관리구조	관리자-대상구조	관리자-대상구조
관리객체 정의방법	ASN.1	ASN.1
정보의 표현방법	스칼라형태의 이차원 배열	Template
MIB 언어	SMI	GDMO
상속	불가능	가능
CLASS	사용안함	사용
명칭의 사용범위	단일시스템에서 유일	전역적으로 유일
PDU	Get-Request Get-Next Get-Bulk-Request Inform-Request Set-Request Response Trap	M-GET M-SET M-Action M-Create M-Delete M-Cancel-Get M-Event-Report

로 변환하는 데 필요한 IIMCOMIBTRANS라는 문서를 제시하였다^{[3][4]}.

4.2 프록시에 관한 연구

IIMC에서는 프록시에 대한 문서로서 IIMCPROXY를 제시하였다. 여기에서는 CMIP 관리자와 SNMP 관리대상 사이의 프로토콜 변환을 위한 역할을 하는 프록시의 구성 방법을 제시하였다. 본 논문에서 제시하는 SNMP 관리자와 CMIP 관리대상 사이의 프로토콜 변환을 위한 역할을 하는 프록시 구성 방법에 대한 문서는 없다^[5].

4.3 관리정보 모델의 매핑

관리프로토콜의 매핑을 완수하기 위해서는 연합된 정보 모델들이 그려져야만 한다. IIMC는 MIB 번역 문제에 대하여 두가지 가능한 접근 방법들을 제시했다. 명칭 부착 계층에 있어서 각 GDMO class가 이러한 번역이 가능하도록 기계화시킨 새로운 SNMP 그룹에 매핑되도록 하는 직접 번역 방법과 GDMO MIB의 의미론적 내용들을 하나의 SNMP MIB에 매핑되도록 한 추상 번역 방법이 있다. 직접 MIB 번역 방법은 번역된 MIB에 거의 모든 original GDMO MIB의 관점을 표현하도록 시도한다. 그것은 독점 관리 정보(즉, equipment objects)를 위해 제기될 것이다. 추상 번역 방법은

표준화된 SNMP MIB를 지원할 때는 언제든지 필요하다. 예를 들면 RFC1695는 인터페이스 M2에 사용된 ATM인터페이스들의 관리를 위해서 하나의 SNMP MIB 모델을 정의한다. 본 논문에서는 직접 번역 방법을 통한 프록시 설계를 구성하였다^{[6][7]}.

(1) 직접 번역 방법

GDMO 템플릿들로부터 인터넷 MIB 매크로들의 직접 번역은 IIMC 문서 IIMCOMIBTRANS에 이미 조사되어졌다. 여기서는 주로 dual-stack 구현을 쉽게 하도록 설계되었다. 이 메카니즘들의 목적들 중의 하나는 번역에 있어서 가능하면 기능성을 최대한 작게 잃어버리게 하는 것이다. 그 다음에 그것은 하나의 single MO가 동시에 다중 프로토콜들을 경유하여 관리되는 것이 가능해야 한다. 심지어 몇몇 인터넷 SMI 모델의 위반들조차도 GDMO의 복잡한 문법과 의미들을 수행하기 위하여 받아들여진다^{[8][9]}. 결론적으로 IIMC 접근 방법은 하나의 SNMP 관리자가 관습적인 인터넷 사용과 일치한 MIB를 보아야 함에 따라 프록시 설계를 위해 전체적으로 채택할 수 없다. IIMC는 그럼에도 불구하고 그 프록시를 위해 필요한 번역 메카니즘의 대부분을 적용한다.

(2) 일반적인 MIB 번역 방법

그림 2는 일반적인 번역 방법을 가시화하였다. GDMO에 정의된 각 MO class는 하나의 SNMP class table에 번역되어진다. 하나의 MO class의 각 instance는 하나의 SNMP class table에 하나의 행에 포함될 것이다.

열 객체 a,b,c,d는 MO class에 관계된 속성들의 열거이다. 부가적으로 각 행은 보통 SNMPv2 RowStatus textual convention에 의한 하나의 INTEGER 구문과 하나의 row status 변수를 포함한 하나의 table index를 가진다^[10]. 그러므로 하나의 CMIP agent's MIB에 동적으로 하나의 MO instance 생성과 삭제에 평등한 이 두 개의 객체들은 table에 있는 하나의 개념적인 행의 생성과 삭제를 허락한다.

행 생성과 삭제를 위한 SNMPv2 메카니즘은 SNMP에 그러한 절차가 없기 때문에 채택되어졌다^[11]. 이것은 또한 그 MIB에 필요한 변화들을 감하는 데 있어서 SNMP를 거울로 삼은 하나의 프록시로부터 SNMPv2를 거울로 삼는 프록시까지의 이동을 단순화할 것이다. containment relationship에

대하여 개념적인 행들의 생성과 삭제를 위한 허락을 위해 각 행에 있어서 하나의 추가적인 객체가 필요하다.

Parant는 직접적으로 상위 객체에 대하여 하나의 class table 안에 개념적인 행을 가리킨다. CMIP MIB에 MO instances의 Distinguished name(DN) 이름을 짓는 매핑을 위해서 SNMPv2의 Instance Pointer convention에 의해 대체한다. DN는 보통 꽤 길다. 그러므로 SNMP에 명명하여 사용된 OID 형에 넣기가 적합하지 않다. 결론적으로, CMIP의 각 MO class와 관계된 naming attribute는 하나의 SNMP 객체에 매핑될 수 없다. 왜냐하면, CMIP의 MO class ID는 다중 속성 ID로 구성되기 때문이다.

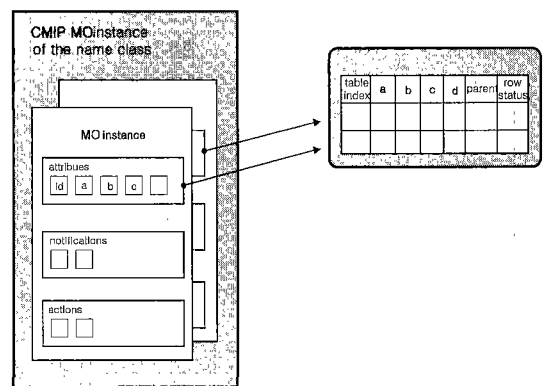


그림 2. 일반적인 MIB 번역 방법

(3) 복잡한 데이터 구조형의 매핑

SNMP가 오직 스칼라 구문에 대해서만 columnar object들을 허락하기 때문에 데이터 구조형들을 위해 조금 더 많이 복잡하게 된다. 구조형 속성 구문을 위한 절차는 IIMCOMIBTRANS에서 복잡한 테이블 종속성인 면을 제안하였다. 그림 3은 복잡한 GDMO 속성을 해결하기 위한 일반적인 접근 방법을 나타낸다. 여기에서는 복잡한 속성마다 하나의 side table이 있도록 제한되어져 있다. 이것은 하나의 side table에서 MIB tree를 통한 검색과 보다 더 복잡하게 된 객체의 생성과 삭제가 모두 유리하기 때문이다.

하나의 side table은 관계된 class table로 인해 같은 인덱스 변수에 의해서 색인이 붙여진다. 구체적으로 다음 번역 메카니즘이 적용되어진다.

- SEQUENCE 와 SET 구문은 하나의 side table

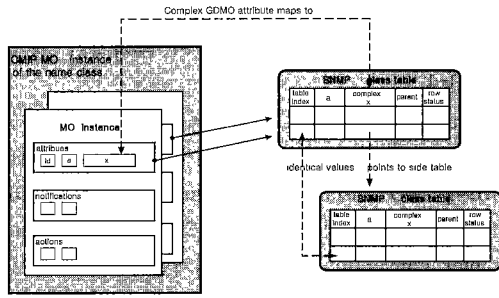


그림 3. 복잡한 데이터 구조형의 변환

에 그러진다. 따라서 그 SET 구문은 요소들의 하나의 차례 목록을 변환한다.

• SEQUENCE 와 SET OF 구문은 변수 길이의 문제를 제기한다. 그래서 하나의 side table에 그려질 수 없다. 대신에 완전한 구조물은 하나의 OCTET STRING에 놓여져야 한다. 이 OCTET STRING을 분해해야만 SNMP 관리자에게 관리 정보를 이동시킬 수 있다. 더 나아가서 이것은 SNMP PDU의 최대 크기에 따라서 OCTET STRING의 길이를 제한해야 한다. 다음 표 2는 간단한 구문 매핑을 위한 몇 가지 경우를 요약하였고, SNMP와 CMIP의 일반적인 매핑은 표 3과 같다.

표 2. 단순 구문 매핑

OSI construct	SNMP SMI syntax
BIT STRING	OCTET STRING
BOOLEAN	TruthValue convention
ENUMERATED	INTEGER with named vlaues
GeneralizedTime	DateAndTime convention
GraphicString	GraphicString convention
INTEGER	INTEGER
REAL	OCTET STRING
ObjectInstance, Distinguished Name	InstancePointer convention

표 3. SNMP와 CMIP 동작 매핑

SNMP Operation	CMIP Operation
Get,GetNext	M-GET
Set	M-SET,M-CREATE, M-DELETE,M-ACTION
Trap	M-EVENT-REPORT

일반적인 Operation Mapping에 있어서의 기존 연구의 문제점은 하나의 SNMP의 Get/GetNext operation은 하나 또는 그 이상의 CMIP M-GET으로 매핑된다. 특히, GetNext operation의 경우, 원하는 관리정보를 모두 얻을 때까지 프록시에서 관리 대상으로 M-GET operation을 반복 수행한다. 이러한 반복 수행 절차는 프록시와 관리대상간의 관리 트래픽을 증가시키는 요소가 된다. 본 논문에서는 이러한 문제점을 해결하기 위하여 다음 절에서 캐쉬를 이용한 프록시를 제안하였다.

III. Proxy 설계

제안한 프록시 설계에 관한 구성은 다음 그림 4와 같다.

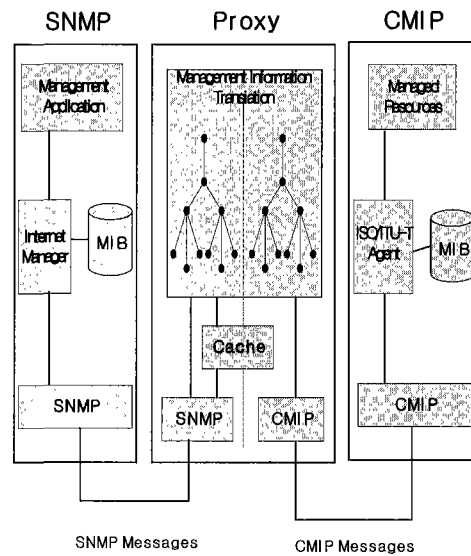


그림 4. 프록시 구성도

프록시는 SNMP 관리자에게 SNMP 인터페이스를 제공하고, CMIP 관리대상에게 CMIP 인터페이스를 제공하여야 한다. 또한 프록시는 MIB 구조 정보(SMI+GDMO)를 가지고 있어야 한다. 각 SNMP 변수는 하나의 MO instance 와 관계된 하나의 속성에 매핑되어야 한다. 다음은 프록시가 해결해야할 3개의 대응 테이블을 제시한다.

- SNMP와 CMIP class subtrees 의 대응 테이블
- SNMP columnar object OIDs를 위한 테이블과 CMIP 속성 등록을 위한 대응 테이블
- CMIP containment hierachy 와 관계된 각

RDN을 하나의 SNMP 테이블 인덱스 값에 복사

본 논문에서는 프록시 내부에 관리 정보를 저장할 수 있는 캐쉬를 두었다. 이 내부 캐쉬는 SNMP의 GetNext request가 왔을 때 해당 object name을 검색하여 곧바로 그 속성값을 응답할 수 있게 해준다.

IV. 제안한 알고리즘

본 논문에서 제안한 알고리즘은 SNMP 매니저에서 어떠한 요청이 왔을 때 프록시에서 우선적으로 그 메시지의 PDU type을 확인한다. 만약에 PDU type이 GetNext가 아니면 기존의 방식을 그대로 따르고, 그렇지 않은 경우 즉, GetNext인 경우는 제안한 알고리즘을 따른다. 제안한 알고리즘은 기존의 방식에서 GetNext인 경우에 불필요한 관리 조작과 트래픽을 최대한 줄이는 데 그 초점을 맞추었다. 그래서, 제안한 알고리즘에서는 GetNext인 경우는 프록시 내부에서는 캐쉬를 이용하여 반복적인 관리 변환 조작을 감소시켰으며, 프록시와 관리대상간에서는 스코핑을 사용하여 한꺼번에 해당 속성값을 가져옴으로 인해 불필요한 관리 트래픽을 줄였다. 제안한 알고리즘 흐름도는 그림5와 같다.

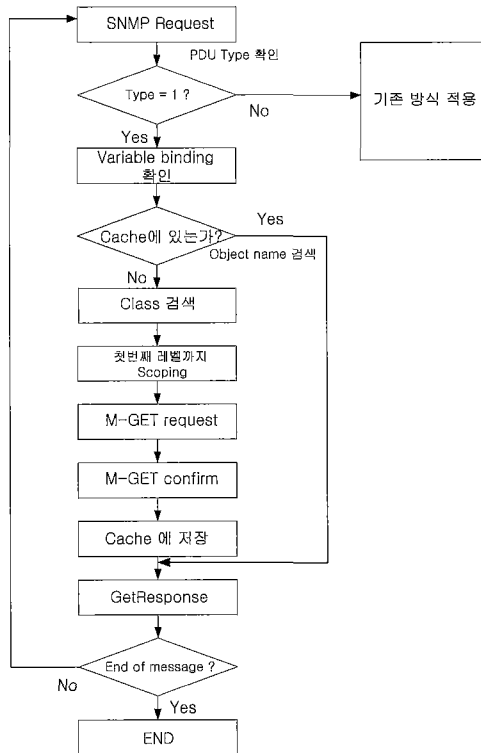


그림 5. 제안한 알고리즘 흐름도

V. 예상효과

제안한 방식을 도입하면 기존의 방식에서 SNMP GetNext 메시지에 대하여 해당 속성값들을 모두 가져올 때까지 프록시에서 CMIP 에이전트로 반복적인 관리조작을 단 한번의 스코핑을 통한 M-GET을 통해서 그 값들을 모두 가져오므로 불필요한 관리 조작을 줄일 수 있다. 또한, 제안한 방식에서는 그 속성값들을 내부 캐쉬에 모두 저장하고 있으므로, 두 번째 SNMP GetNext 요청에 대해서는 곧바로 캐쉬에 있는 객체명을 검색하여 그 속성값을 가져오면 되므로 프록시내에서도 불필요한 관리조작을 줄일 수 있다. 그러므로, 프록시에서 CMIP 에이전트 사이의 관리조작 처리 시간의 단축과 프록시 내에서의 불필요한 반복적인 변환처리를 줄여줌으로써 전체 응답시간이 한층 더 향상되도록 하였다.

관리 트래픽 측면에서도 CMIP는 연결 지향형 프로토콜이므로 각각의 관리조작에 대한 연결 설정, 데이터 전송, 연결 해제등의 복잡한 절차를 갖게 되어 이 절차에 따라 M-GET 관리 트래픽을 반복적으로 발생시킨다. 여기서 제안한 알고리즘을 이용하여 불필요한 반복적 M-GET 관리 트래픽 발생을 억제할 수 있다. 또한, 인터넷을 통하여 관리자가 어디에서든지 장소와 시간에 구애를 받지 않고 관리대상을 관리할 수 있다는 큰 장점이 있다.

VI. 결론

본 논문에서는 SNMP 관리자를 통하여 CMIP 관리대상 시스템 관리를 위한 프록시 설계에 있어서 불필요한 관리 조작을 줄이기 위해 SNMP/CMIP 대응 트리와 캐쉬 그리고 스코핑 절차를 이용하는 알고리즘을 제안하였다. 제안한 알고리즘은 프록시로 들어오는 관리 조작의 PDU type이 GetNext일 경우 스코핑 기능을 사용하여 관리대상으로부터 가져온 관리 정보를 프록시 내부 캐쉬에 저장하는 방식을 사용하였고, GetNext가 아닐 경우는 기존의 방식을 사용하였다. 제안한 알고리즘을 이용하여 기존의 방식에서 발생했던 프록시와 관리대상 간의 불필요한 관리 조작 반복 수와 이로 인해 발생하는 네트워크 관리 트래픽을 감소시켰으며, 이에 따른 프록시와 관리대상 간의 관리 조작 및 관리 정보 처리 시간을 단축시켰다. 또한 제안한 알고리즘은 SNMP 매니저의 폴링주기와 관리 정보 테이블 크

기 및 CMIP 관리대상 수, 즉 관리 대상 수가 증가할수록 제안한 알고리즘의 기대효과가 커질 것으로 예상되어 진다. 본 논문은 프록시 설계를 위한 기본 구조와 관리 조작 처리 효율을 높이기 위한 알고리즘을 제시하였고 앞으로 프록시의 구현과 시뮬레이션을 통한 프록시 성능 평가가 필요하다.

참 고 문 헌

- [1] J. Case, M. Fedor, M. Schoffstall and J. Davin, "A simple network management protocol(SNMP)," Internet RFC-1157, 1990.
- [2] M. Gering, "CMIP versus SNMP in: Proceedings of the 3rd International Symposium on Integrated Network Management," San Francisco, April 1993, pp347-359.
- [3] Network Management Forum, "Translation of Internet MIBs to ISO/ITU-T GDMO MIBs(II MCIMIBTRANS)," Issue 1.0, 1993.
- [4] Network Management Forum, "Translation of ISO/ITU-T GDMO MIBs to Internet MIBs (II MCOMIBTRANS)," Issue 1.0, 1993.
- [5] Network Management Forum, "ISO/ITU-T to Internet Management Proxy(IIMCPROXY)," Issue 1.0, 1993.
- [6] M. Ahmad and K. Tesink, "Definition of managed objects for ATM management version 8.0 using SMIV2," Internet RFC-1695, 1994.
- [7] E. Koerner, "Design of a proxy for managing CMIP agents via SNMP," Computer Communications, V.20, 1997.
- [8] M. Rose and K. McClogbrie, "Structure and identification of management information for TCP/IP-based internets," Internet RFC-1155, 1990.
- [9] M. Rose and K. McClogbrie(Eds.), "Concise MIB Definitions," Internet RFC-1212, 1991.
- [10] K. McClogrie, "Textual conventions for version 2 of the simple network management protocol(SNMPv2)," Internet Draft, 1995.
- [11] K. McCloghrie, "Protocol operations for version2 of the simple network management protocol(SNMPv2)," Internet Draft, 1995.

박 주 희(Ju-hee Park)

정회원



1997년 2월 : 관동대학교
전자공학과 졸업
2000년 8월 : 광운대학교
전자통신공학과 석사
2001년 3월~현재 : 광운대학교
전자통신공학과
박사과정

<주관심 분야> 통신공학, 망관리, 의료정보시스템,

박 승 균(Seung-kyun Park)

1993년 2월 : 광운대학교 전자통신공학과 졸업
1995년 2월 : 광운대학교 전자통신공학과 석사
1999년 2월~현재 : 광운대학교 전자통신공학과
박사과정

오 영 환(Young-hwan Oh)

정회원

한국 통신학회 논문지 제 26권 제4A호 참조