

저부하의 네트워크 트래픽을 갖는 효율적인 SNMP 설계

정회원 윤천균*, 이찬민**, 탁동길***, 정일용***

The Design of an Efficient SNMP with Low Load of Network Traffic

Chun-Kyun Youn*, Chan-Min Lee**, Dong-Kil Tak***,
Il-Yong Chung*** *Regular Members*

요 약

SNMP에서는 관리정책에 따라 Manager가 특정 Object에 대하여 MIB 값을 Agent에 요청하고 Agent가 이에 응답하는 식으로 정보를 송수신하므로 많은 네트워크 traffic이 발생한다. 특히, Traffic의 경향 분석 기능 수행시 동일 Object에 대한 반복적인 정보 송수신 발생은 매우 비효율적이다.

본 논문에서는 기존의 SNMP에 시간 기능을 수용할 수 있는 새로운 PDU를 추가하고, 이를 이용하여 Manager와 Agent간에 불필요한 정보 송수신을 최소화하면서 경향 관리 정보를 수집할 수 있는 효율적인 방안을 제시하였다. 시험 결과 구현된 SNMP는 기존 SNMP와 호환성을 유지하면서 네트워크 Traffic을 크게 감소시켰다.

키워드: SNMP, Traffic

ABSTRACT

An SNMP manager requests a value in MIB of a designated object to an agent by management policy defined in the SNMP, and then the agent responds to this manager. So, it causes a lot of network traffic between them. Specially, repetitious occurrence of sending-receiving information related to the same object is very inefficient when the trend analysis of traffic is performed.

In this paper, an efficient SNMP is proposed to add new PDUs to the existing SNMP in order to accept time function. Utilizing this PDU, it minimizes unnecessary sending-receiving information and collects information on trend management appropriately. The implemented SNMP is tested for compatibility with the existing SNMP and decreases the amount of network traffic greatly.

I. 서 론

네트워크 관리 방법에는 UDP/IP기반의 관리 프로토콜로 IETF의 SNMP (Simple Network Management Protocol)와 ISO 네트워크 관리 표준인 Common Management Information Protocol(CMIP)/ Common Management Information Service(CMIS)가 있으나 SNMP가 가장 많이 사용하고 있다[1][2].

SNMP의 Message 송수신 방법은 UDP(User Datagram Protocol) 상에서 Manager가 관리대상 MIB Object에 대하여 Agent에 정보 송수신을 요청하고, Agent가 이에 응답하는 비동기식 서버/클라이언트 방식을 통하여 이루어진다. SNMP v2에서 "GetBulk Request" PDU를 이용하여 동시에 여러 MIB 정보를 요청하고 그 결과를 Agent로부터 수신하는 경우를 제외하고는, 일반적으로

* 호남대학교 정보기술원(chqyoun@itc.honam.ac.kr), ** 광주기독교간호대학, *** 조선대학교 컴퓨터공학부

* 논문번호 : 020530-1209, 접수일자 : 2002.12.09

단일 Object에 대한 단일 응답 형태이다[3][4].

네트워크 관리를 위해서는 다양한 분석항목들을 사용하는데 그 중 심화분석 항목들은 실시간 분석과는 달리 특정기간 동안의 현황을 알아보기 위하여 특정 object에 대하여 일정기간의 History 정보, 통계정보 등을 수집하여 Trend를 Monitoring해야 한다[5] [6][7]. 이를 위하여 SNMP에서는 Manager가 해당 MIB Object에 대하여 일정 주기로 "GetRequest" PDU를 Agent에 반복하여 Polling하고, Agent로부터 "GetResponse" PDU들을 반복 수신하여, 이를 Database에 저장하고 Trend 정보로 표시한다. 이와 같은 반복적인 요청과 응답으로 인하여 발생하는 Message들은 네트워크 Traffic load 증가, Manager 시스템의 부하 가중 그리고 응답시간 지연 등의 문제를 일으킨다[8].

본 논문에서는 이와 같이 일정기간 동안 반복수집이 필요한 정보를 "경향관리 정보(Trend Management Information)"라 정의하고, 이러한 경향관리 정보 수집 시 기존 SNMP에서 발생하는 문제를 효율적으로 처리하여 네트워크 Traffic load를 감소시킬 수 있는 개선방법에 대하여 제안하고, Prototype을 구현하여 Test한 후 그 결과를 분석하고자 한다.

2장에서는 기존 SNMP의 기본적인 개념에 대하여 알아보고, 제안하고자 하는 개선된 SNMP 모델과 동작원리를 설명한다. 3장에서는 Test bed에 구현된 Prototype을 이용하여 기존 SNMP와 비교 실험을 실시하여 개선된 SNMP의 성능을 검증하기로 한다. 마지막으로 결론과 향후 계획을 제시한다.

II. 관련연구

1. SNMP의 기본 개념

TCP/IP 네트워크를 관리하기 위한 4가지 모델은 관리 시스템(Manager), 관리대상 에이전트(Agent), MIB(Management Information Base), 네트워크관리 프로토콜인 SNMP로 구성된다[4]. 관리 시스템의 Manager와 관리대상 Agent 간의 동작은 서버와 클라이언트 개념으로 설명될 수 있으며, 그림1에서와 같이 관리 시스템의 Manager가 Agent에게 GetRequest, SetRequest, GetNextRequest 등의 PDU를 보내면 Agent가 이를 받아서 이와 연관되는 결과를 GetResponse PDU를 이용하여 Manager에게 전송하는 방식으로 MIB 정보를 수집한다. 또 Manager의 요구 없이 Agent가 Manager에게 정보 전달을 필요로 하는 경우 Agent는 Trap을 발생시킴으로 장애나 오류 등

의 필요한 정보를 Manager에게 전달한다[3][4]. 네트워크 관리 프로토콜인 SNMP는 바로 이 둘 사이의 통신을 위해 사용되는 프로토콜이다. SNMP는 표준화된 프로토콜로서 구현이 쉽고 간편해서 네트워크 관리를 요구하는 대부분의 네트워크 장치에 보편적으로 사용되고 있다[2].

SNMP는 UDP(User Datagram Protocol)상에서 동작하는 비동기식 요청/응답 메시지 프로토콜로서 다음의 4가지 기능을 수행한다[4][9].

Get은 Agent의 상태 및 가동시간 등의 MIB를 읽어 들이고, Get Next는 MIB가 계층적 구조를 유지하고 있으므로 Manager가 Agent에게 해당 tree의 다음 하위계층 MIB를 읽는 기능을 한다. Set은 Agent의 MIB를 설정하여 장비를 제어하고, Trap은 Agent의 중요 또는 긴급 사황에 대하여 Manager에게 보고하는 Threshold나 Event 기능을 수행한다.

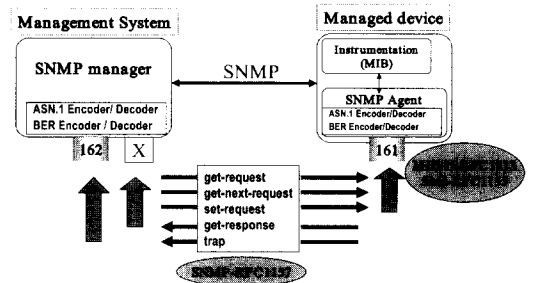


그림 1. SNMP의 동작 구성도

2. 경향관리 정보를 위한 SNMP 개선 모델 설계

기존 SNMP에서 경향관리 정보를 일정기간 monitoring하기 위해서는 그림 2와 같이 Manager가 동일 MIB object에 대하여 그림 3과 같은 형태의 "GetRequest" PDU를 반복적으로 Agent에 Polling하고 Agent는 응답으로 "GetResponse" PDU들을 해당 횟수만큼 송신한다. 이와 같은 반복 송수신에 의하여 발생하는 네트워크 Traffic 문제를 개선하기 위하여 그림 4와 같은 SNMP 개선 Model을 설계한다.

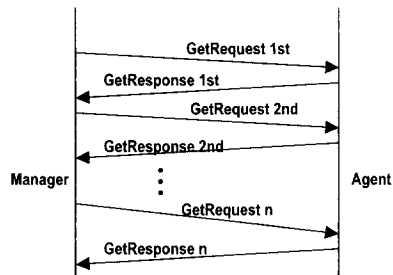


그림 2. 경향정보 수집 시 SNMP의 동작 process

개선된 SNMP Model에서는 기존 SNMP PDU에 경향관리 정보용의 PDU로 "GetTMIRequest"와 "GetTMIResponse"를 그림 5와 같이 정의하여 추가하고, 그림 6과 같이 송수신 Sequence를 처리함으로써 기존 SNMP에서 반복되는 "GetRequest"와 "GetResponse" 전송을 최소화하였다.

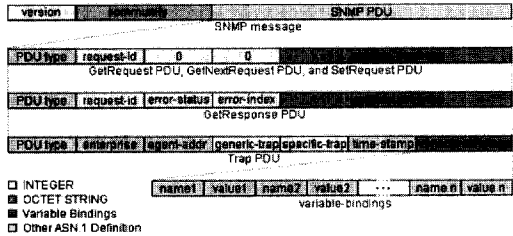


그림 3. SNMP V1에서의 PDU 구성

면 다음과 같다.

- TMI PDU Requester : Manager의 화면을 통하여 사용자가 지정한 MIB 변수에 대한 자료수집 시작시간, 자료수집 종료시간, 수집주기를 이용하여 GetTMIRequest PDU를 생성하고 이를 Agent의 TMI PDU Identifier에 전송하는 기능 수행.
- TMI PDU Identifier ; Manager로부터 전달된 GetTMIRequest PDU의 PDU type field를 분석하여 기존 SNMP PDU인지 경향관리 정보를 위한 PDU인지 판단한다. 그리고 기존의 SNMP PDU인 경우는 기존 SNMP의 기능을 수행하도록 하고, 경향관리 정보를 위한 PDU인 경우는 메시지를 TMI SNMP Agent handler에게 전달하는 기능을 수행한다.

- TMI SNMP Agent Handler : 개선된 SNMP Model 구성 Module 중 핵심 기능을 수행하는 부분으로, Manager로부터 수신한 GetTMIRequest PDU를 처리한다. 이 Module은 수신된 GetTMIRequest PDU에서 Header를 분리하여 Request ID, 자료수집 시작시간, 자료수집 종료시간, 수집주기를 추출하고 UPD header로부터 요청한 Manager 시스템의 IP주소를 추출하여 이를 바탕으로 해당 Object의 MIB 정보를 수집하여 GetTMIResponse PDU를 생성한 후 이를 Manager의 TMI SNMP Manager Handler에게 송신하는 기능을 수행한다. 즉, 수집시작 시간이 되면 이 모듈은 추출한 수집주기에 따라 주기적으로 해당 Object의 MIB 정보를 수집하여 요청한 Manager 시스템의 Request ID에 따라 자료를 버퍼에 저장하고, 종료시간이 되면 수집을 중단하고 GetTMIResponse PDU를 생성하여 Manager의 TMI SNMP Manager Handler에게 송신한다.

- TMI SNMP Manager Handler : Agent의 TMI SNMP Agent Handler Module로부터 수신한 GetTMIResponse PDU에서 Header를 분리하여 자료수집 시작시간, 자료수집 종료시간, 수집주기를 추출하고 이를 바탕으로 해당 Object의 MIB 정보를 관리 시스템에 있는 Data base에 저장한다.

개선된 SNMP Model에서는 기존의 SNMP 기능들을 그대로 사용할 수 있으면서 추가된 경향관리 정보를 위한 기능도 수행할 수 있다.

3. 추가 SNMP PDU와 송수신 Sequence

기존의 SNMP v1의 PDU에 그림 5와 같이 추가한 "GetTMIRequest"와 "GetTMIResponse" PDU의 구성을 살펴보면, 시작시간(Start Time), 종료시간(End Time), 수집주기(Time Interval) 그리고

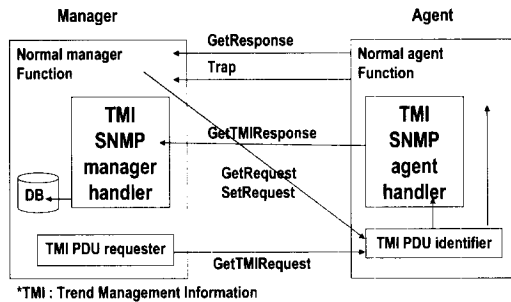


그림 4. 경향관리 정보를 위한 개선 SNMP model

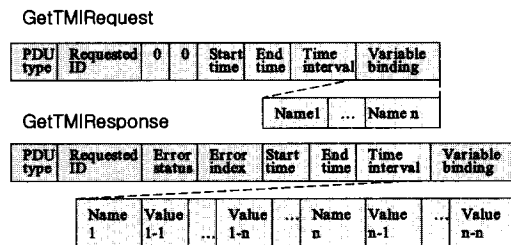


그림 5. 경향관리 정보용 추가 SNMP PDU

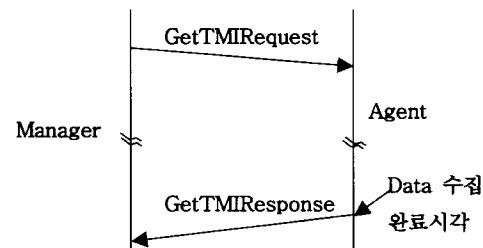


그림 6. 경향관리 정보용 PDU 송수신 Sequence

개선된 SNMP Model의 구성과 그 기능을 살펴보

Variable binding 필드로 구성된다. 이 4가지 필드에는 경향관리 정보들을 주기적으로 수집하는데 필요한 정보들을 포함되어 있으며, 각 필드들의 내용은 다음과 같다.

- Start Time: 요구하는 MIB 변수에 대한 자료수집 시작시간을 나타낸다. Octet string 형으로 표시된다.
- End Time: 요구하는 MIB 변수에 대한 자료수집 종료시간을 나타낸다. Start-Time과 마찬가지로 Octet string 형으로 표시된다.
- Time Interval: 요구하는 MIB 변수에 대한 자료수집 주기를 나타낸다. Integer 형으로 표시되며, 값은 초(sec) 단위를 지정하게 된다. 이 값의 간격으로 SNMP Agent에서 해당 변수의 MIB값이 저장된다.
- Variable binding for GetTMIRRequest : name 1, name 2... name n 형태로 여러 개의 Object를 동시에 요청할 수 있도록 변수명을 표기한다.
- Variable binding for GetTMIRResponse : GetTMIR-Request PDU에서 요청한 변수 name 1, name 2 ... name n에 대하여 자료수집 요청 시간 동안 요청된 수집주기에 따라 수집된 값들을 변수명과 함께 표기한다.

개선된 경향관리 정보용 SNMP PDU의 송수신 Sequence를 살펴보면, 그림 6과 같이 Manager는 자료수집 시작시간(Start Time), 종료시간(End Time), 수집주기(Time Interval) 그리고 Variable binding 필드를 이용하여 "GetTMIRRequest" PDU를 구성하여 agent에 전송하고, agent에서는 요청된 조건에 따라 자료를 수집하고 수집이 완료되면 "GetTMIR-Response" PDU를 구성하여 즉시 전송함으로써 반복적인 Polling과 응답으로 인한 네트워크 Traffic Load의 증가 문제를 크게 개선할 수 있다

4. 개선된 SNMP Model의 동작 Process

경향관리 정보 수집 기능을 수행하기 위해 그림 4에서와 같이 Manager는 PDU Type, Request-ID, 자료수집 시작시간, 자료수집 종료시간, 수집주기, Variable-bindings Field들로 SNMP GetTMIRRequest PDU를 구성하여 Agent에게 보낸다. 개선된 SNMP Agent는 수신한 GetTMIRRequest PDU를 TMI PDU Identifier Module에 전달한다. TMI PDU Identifier Module은 수신된 GetTMIRRequest PDU의 PDU Type을 분석하여, 기존 SNMP PDU의 경우는 이를 Normal Agent Module로 전달하여 기존 SNMP에서와 같이 처리되도록 하고, 경향관리 정보용 GetTMIRRequest PDU일 경우는 이를 TMI SNMP Agent

handler에게 전달하게 된다.

TMI SNMP Agent handler는 수신한 GetTMIR-Request PDU의 Header로부터 Request-ID, 자료수집 시작시간, 종료시간, 수집주기, Variable-Bindings의 Field를 분리하여 각 필드들의 정보와 IP packet header에서 추출한 Manager 시스템의 IP주소를 이용하여 주어진 시간 동안 지정한 간격으로 data를 수집하여 자료수집 종료 시각에 GetTMIRRequest PDU를 생성하여 manager에게 보낸다.

Manager는 Agent의 TMI SNMP Agent Handler Module로부터 수신한 GetTMIRResponse PDU Header에서 PDU Type을 분석, 기존 SNMP의 경우는 이를 Normal Manager Module로 전달하여 기존 SNMP 방법과 같이 처리되도록 하고, 경향관리 정보용 GetTMIRResponse PDU일 경우는 이를 TMI SNMP Manager Handler에게 전달하게 된다.

TMI SNMP Manager Handler는 GetTMIR-Response PDU Header에서 PDU Type, Request ID, 자료수집 시작시간, 종료시간, 수집주기, Variable-Bindings를 추출하여 해당 Object의 경향관리 정보를 관리시스템의 Data base에 저장한다.

III. 실험 환경 및 결과 분석

1. 실험 환경

효율적인 경향관리 정보 수집을 위하여 개선된 SNMP의 구현 및 기존 SNMP와의 성능비교를 위한 시험 환경조건은 다음과 같다.

- 운영체제 : Manager : Unix(Solaris2.8), Agent : Linux(Redhat6.2), 측정시스템 : Windows XP
- 사용된 SNMP version : SNMP version 2
- 사용된 SNMP toolkit : UCD SNMP v3.4
- 구현에 사용된 언어 : C, Shell 기반 CGI
- C 컴파일러 : GCC v3.1
- Traffic 측정 Tool : LANdecoder32 (Triticom), AppDancer FA (AppDancer Networks)

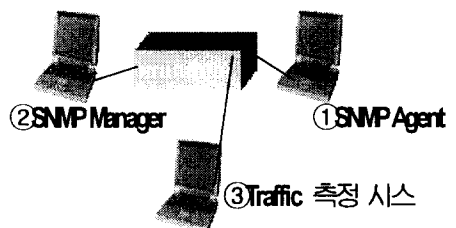


그림 7. Traffic load 측정을 위한 Test bed 구성도

개선된 SNMP에 의한 Traffic 감소를 시험하기 위하여 그림 7과 같이 타 네트워크에 연결되지 않는 독립된 형태의 Test Bed를 구축하고 기존 SNMP 기능 수행 시와 개선된 SNMP 수행 시의 Traffic load를 측정 비교한다. Test Bed의 각 시스템들은 Shared media인 단순 허브를 공유하는 Ethernet LAN이다. Manager(IP 211.227.240.56)와 agent(IP 211.227.240.155) 간에는 SNMP에 의하여 정보를 송수신하고, Traffic 측정 시스템(IP 211.227.240.35)에서는 Web을 이용하여 Manager 시스템을 기동시키며 네트워크 관리 Tool들을 이용하여 네트워크 Traffic을 측정한다.

기존 SNMP와 개선된 SNMP를 이용하여 수집한 경향관리 정보의 경우 처리하는 방법은 서로 다르지만 동일한 결과를 나타내는 것과 네트워크 Traffic load 감소를 실험하기 위하여 Manager가 agent에 대하여 동일한 MIB object "iso(1).org(3).dod(6).internet(1).mgmt(2).mib-2(1).Interfaces(2).ifTable(2).IfEntry(1).ifInOctets(10)"(total number of octets received on the interface, including framing characters)의 값을 5초 주기로 30분간 측정한다. 측정된 결과를 1분 주기의 평균값으로 그래프 상에 표기하여 비교한다.

2. 실험 결과 분석

2.1 네트워크 Traffic Load

1) 기존 SNMP에 의한 네트워크 Traffic

그림 2에서와 같이 특정 object에 대하여 경향관리 정보 수집 시 Manager가 발생시킨 GetRequest에 의한 Traffic량을 Tgrq라하고, 이에 대한 Agent의 응답인 GetResponse에 의해 발생된 Traffic량을 Tgrp라하면, 한 개의 정보 수집 시 발생되는 Traffic량(Tf)은 식(1)과 같다. 그리고, 정보 수집기간이 Dt이고 수집주기가 It일 경우 발생되는 총 Traffic량(Ttf)은 식(2)와 같다.

$$Tf = Tgrq + Tgrp \tag{1}$$

$$Ttf = Dt/It * Tf \tag{2}$$

식(2)에서 알 수 있듯이 경향관리 정보 수집기간이 길고, 주기가 짧을수록 발생되는 총 Traffic양이 증가한다.

기존 SNMP를 이용하여 30분간 측정된 Traffic은 그림 8과 같으며, 분당 약 12.9KBytes의 평균 Traffic이 발생되었으며, Traffic이 계속 일정하지 않

은 이유는 구축된 Test Bed에는 그림 9에서 알 수 있듯이 SNMP 외에도 측정 시스템에서 Web을 이용하여 Manager 기동 시 발생하는 HTTP와 Windows Xp에서 발생하는 NetBIOS 등의 프로토콜들이 존재하기 때문이다

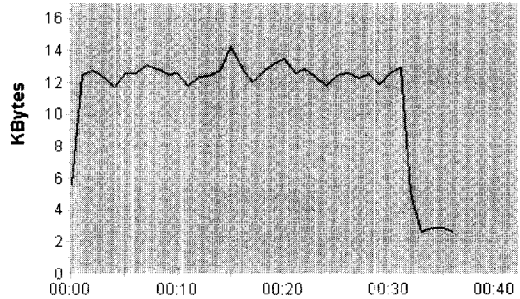


그림 8. 기존 SNMP에 의해 측정된 Traffic load

그림 9는 Manager(211.227.240.56)에서 주기적으로 GetRequest를 Agent(211.227.240.155)로 보내고, Agent에서는 요구 정보를 수집하여 이에 응답하는 Sequence를 네트워크 Tool로 측정된 화면을 2등분한 것이다. Source/Destination 주소와 1번째 SNMP 송수신의 경우 응답시간이 1.957msec 소요되었으며, GetRequest Frame의 크기가 87Bytes이고 Get-Response Frame의 크기가 90Bytes임을 알 수 있다.

	Src. Addr	Dst. Addr	Protocol	Errors
1	211.227.240.35/1456	211.227.240.56/80	HTTP	0
2	211.227.240.56/35717	211.227.240.155/161	SNMP	0
3	211.227.240.56/35716	211.227.240.155/161	SNMP	0
4	211.227.240.56/35715	211.227.240.155/161	SNMP	0
5	211.227.240.56/35714	211.227.240.155/161	SNMP	0
6	211.227.240.56/35713	211.227.240.155/161	SNMP	0

Frame #	Direction	Size
4573	GET[Short Frame]	87 Bytes
4574	RESPONSE[Short Frame]	90 Bytes

Avg. Resp. Time	Total Frames	Update Time
0.000785	2326	11/20/2002 19:17:17.254.010
0.001953	2	11/20/2002 19:16:55.931.102
0.001952	2	11/20/2002 19:16:50.811.277
0.001957	2	11/20/2002 19:16:45.691.985
0.001960	2	11/20/2002 19:16:40.572.027
0.002032	2	11/20/2002 19:16:35.453.091

Delta Time	Rel. Time	Len
[]	[]	[]
[]	[]	[]

그림 9. 기존 SNMP의 Traffic Sequence

2) 개선된 SNMP에 의한 네트워크 Traffic

그림 6과 같이 개선된 SNMP을 이용하여 경향관리 정보 수집 시 Manager가 발생시킨 GetTMI-Request에 의한 Traffic량을 Tegrq라하고, 이에 대한 Agent의 응답인 GetTMIResponse에 의한 Traffic량을 Tegrp라 하면, 한 개의 정보 수집 시 발생하는 Traffic량(Tef)은 식(3)과 같으며, 정보 수집 기간이 Dt이고 수집주기가 It일 경우 발생하는 총 Traffic 량(Tetf)은 식(4)와 같다.

$$Tef = Tegrq + Tegrp \tag{3}$$

$$Tetf = Tegrq + Tegrp = Tef \tag{4}$$

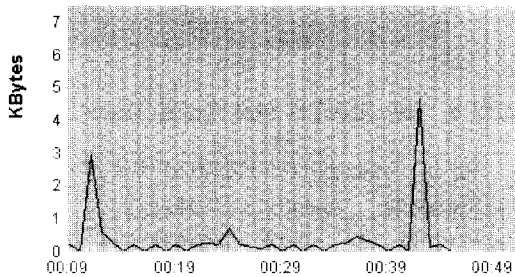


그림 10. 개선된 SNMP를 이용하여 측정된 Traffic load

개선된 SNMP를 이용하여 30분간 측정된 Traffic은 그림 10에서와 같이 분당 약 0.4KBytes의 평균 Traffic이 발생되었으며, 순수 SNMP Traffic은 Manager가 Agent에 GetTMIRequest를 전송하는 초반과 정보수집이 완료되어 Agent가 GetTMI-Response를 Manager에 전송하는 마지막 부분에서 발생되었다. 그 외 Traffic들은 기존 SNMP에서와 같이 측정 시스템에서 Web을 이용하여 Manager 기동 시 발생한 HTTP와 Windows Xp에서 발생하는 NetBIOS 등의 프로토콜들이다.

그림 11은 기존 SNMP에서와 같이 Manager(211.227.240.56)에서 Agent(211.227.240.155)로 GetTMI Request를 보내고 Agent에서는 약 30분간 요구 정보를 수집하여 Manager에 보내는 개선된 SNMP의 Sequence를 네트워크 Tool로 측정 한 화면을 2등분 한 것이다. 그림 9의 기존 SNMP에서와 달리 SNMP Total Frame 수가 2회 발생하고 응답시간이 1863.8 초(약 30분)인 것은 그림 6의 Logic이 정상 작동한다는 것이다. GetTMIRequest와 GetTMIResponse Frame의 크기가 각각 108Bytes와 1,514Bytes로 개선된 SNMP에서는 그림 10에서와 같이 후반부의 Traffic Load가 상대적으로 높다는 것을 알 수 있다.

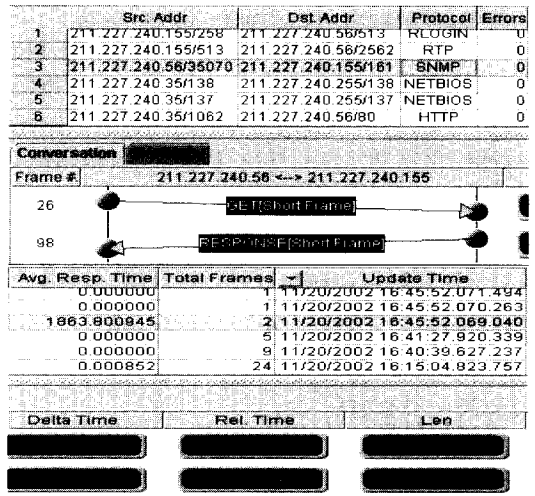


그림 11. 개선된 SNMP의 Traffic Sequence

3) 결과 비교 분석

기존의 SNMP와 개선된 SNMP 방식을 비교 분석해 보면 단일전송의 경우 그림 3과 그림 5의 PDU구성에서 알 수 있듯이 GetTMIRequest와 GetTMIResponse가 GetRequest와 GetResponse보다 크기 때문에 $Tegrq > Tgrq$, $Tegrp \geq Tgrp$ 의 관계가 성립되어 식(1)과 식(3)에서 $Tef \geq Tf$ 가 성립한다. 이들 간의 Frame 크기 차이는 요청하는 변수와 측정기간에 따라 차이가 있으며, 단일 전송의 경우는 기존 SNMP가 각각 21bytes(108/87), 62bytes(152/90)씩 다소 적은 Traffic 양을 나타낸다.

반면에 본 논문에서 중점을 둔 경향관리 정보의 경우 순수 SNMP에 의하여 발생된 총 Traffic을 비교해 보면 기존 SNMP는 그림 9와 같이 총 63,720 bits((87B+90B)*60/5초*30분)의 Frame Traffic이 발생되었고, 개선된 경우에는 그림 11과 같이 1,622bits (108B+1,514B)가 발생되어 개선된 SNMP에서의 Traffic이 약 39.3배 감소하였음을 알 수 있다. 다른 프로토콜들을 포함한 총 Traffic은 평균값 기준으로 비교하여 약 32배(12.9/0.4KB)가 감소하였다.

정리하면 기존 SNMP의 경우 Data 송수신 횟수에 정비례하여Network Traffic이 증가하는 반면, 개선된 SNMP의 Traffic은 측정횟수에 따라 다소 증가하나 큰 차이가 없기 때문에 측정기간이 길고 측정 주기가 짧아질수록 네트워크 Traffic Load가 크게 감소함을 알 수 있다.

2.2 측정된 경향관리 정보일치 실험

개선된 SNMP에서 측정된 경향관리 Data가 기존

SNMP에 의해 측정된 Data와 동일한 결과를 나타내는지 알아보는 실험으로 측정시스템에서 Web Browser로 Manager 시스템을 기동하여 기존 SNMP와 개선된 SNMP가 거의 동시에 Agent에 동일한 MIB object "ifInOctets(1.3.6.1.2.1.2.1.10)" (total number of octets received on the interface, including framing characters)의 값을 30초 주기로 30분간 측정하여 이를 그래프로 나타내어 비교한다.

동일한 시간을 기준으로 기존의 SNMP와 개선된 SNMP가 동일 MIB 값을 읽어야하므로 Manager와 Agent, 측정 시스템간에 시간동기가 필요하다. 이를 위하여 Manager와 Agent에는 NTP(Network Time protocol)[10]를 설치하여 "gps.bora.net" 서버의 표준 시간에 일치시키고, 측정시스템은 Windows XP에서 제공하는 기능을 이용하여 인터넷 시간 서버와 동기화 되도록 하였다.

그림 12. Data 정합성 실험 측정값 예

	A	B	C	D
1	절대시간(초)	Time(sec)	Normal SNMP	Improved SNMP
2	1038319989	0	400971788	400971281
3	1038320019	30	400973465	400972598
4	1038320049	60	400974332	400973825
5	1038320079	90	400975769	400975142
6	1038320109	120	400976376	400976369
7	1038320139	150	400978193	400977686
8	1038320169	180	400979300	400978913
9	1038320200	210	400980497	400979720
10	1038320230	240	400981784	400981337
11	1038320260	270	400982861	400982084
12	1038320290	300	400984088	400983641
13	1038320320	330	400984335	400984388
14	1038320350	360	400986392	400985945
15	1038320380	390	400987139	400986692
16	1038320410	420	400988696	400988249

그림 12에서 기존 SNMP와 개선된 SNMP에 의하여 측정된 값이 약간 차이를 보이는 이유는 측정시스템의 모니터에 각각의 Web화면을 Display해 놓은 상태에서 거의 동시에 이들을 기동하였으나 마우스 조작 시간과 프로그램들 간의 실행시간 차 등이 발생하였기 때문이며, 측정 MIB Object의 먼저 기동된 개선된 SNMP에 의해 측정된 값이 다소 적게 나타나고 있다. 수집시간 차에 따른 측정값들의 차는 당연하며 그 차가 매우 근소하여 그림 13과 같이 2개의 그래프가 거의 겹치는 것을 알 수 있다. 이는 개선된 SNMP가 기능상에 문제가 없음이 증명된다.

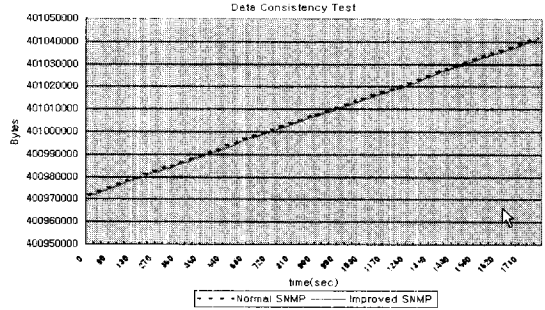


그림 13. Data 정합성 실험 측정 그래프

결론적으로 개선된 SNMP는 경향관리 정보의 수집에 있어서 그림 13에서와 같이 기존 SNMP와 동일한 결과를 나타내므로 기존 SNMP와 호환성을 유지하면서 네트워크 Traffic Load를 크게 감소시킨다.

IV. 결론 및 향후 계획

실험 결과를 바탕으로 개선된 SNMP의 특징을 정리해보면, 첫째, 기존 SNMP와 동일한 기능을 수행하면서 매우 적은 횟수의 요구와 응답 메시지를 송수신함으로써 네트워크 Traffic Load를 크게 감소시켰다. 둘째, 기존 SNMP 기능과 완전하게 호환성을 유지하면서 네트워크 성능관리에 필수적인 경향관리 정보를 위한 확장 기능을 성공적으로 수행한다. 따라서 개선된 SNMP를 대형 네트워크를 관리하는 네트워크 관리시스템(NMS)에 적용시킬 경우 그 효과는 매우 클 것으로 사료된다.

추가적으로 연구가 필요한 부분은 Agent Device 종류에 따라 하드디스크가 없거나 메모리가 작은 경우 경향관리 정보의 일시 보관 문제 해결 방안, Embedded OS일 경우 적용 방안, Agent로부터 수집된 정보가 한꺼번에 전송되므로 GetTMIResponse PDU 누락 시 전체 Data의 손실 발생 가능성에 대한 대책이 필요하고, 경향관리 정보를 네트워크 Traffic이 혼잡한 시간을 피해 요구된 시간에 수신될 수 있도록 하는 연구가 필요하다.

참고 문헌

[1] Aiko Pras, "Network Management Architectures," CTIT Ph. D-thesis series, *ISSN 1381-3617*; no.95-02, Feb. 1995
 [2] Amatzia Ben-Artzi, Asheem Chandna, Unni

Warrier, "Network Management of TCP/IP Networks: presents and future", *IEEE Network Magazine*, pp. 35-43, July 1990

[3] Uyless Black, *Network Management Standards SNMP, CMIP, TMN, MIBs and Object Libraries, Second Edition*, 1993

[4] William Stallings, *SNMP, SNMP v2, SNMP v3, and RMON 1 and 2, Third edition*, Addison-Weseley, 1996

[5] 유승근, 안성진, 정진욱, "SNMP MIB-II를 이용한 인터넷 관리 시스템의 웹 인터페이스 설계 및 구현", *한국정보처리학회논문지*, 제6권 제3호, pp. 699-709, 1999

[6] Sang-chul Shin, Seong jin Ahn, jin Wook Chung, "Design and Impelementation of S NMP based performance parameter extrac tion system", *Asia-pacific network operati ons and Management Symposium*, 1997

[7] Sang chul Shin, Seong-jin Ahn, jin-Wook Chung, "A new approach to gather netwo rk management data periodically", *ITC-C SCC '97*, 1997

[8] 김 동수, 정 태명, "실시간 네트워크 관리를 위 한 SNMP의 확장에 관한 연구", *한국정보처리 학회논문지*, 제6권 제2호, pp. 449-458, 1999

[9] K. McCloghrie, M. Rose. "Management Inf ormation Base for Network Manage- ment of TCP/IP based Internets: MIB -II." *In ternet RFC 1213*, March 1991.

[10] <http://time.ewha.net/#DJClock>

윤 천 균(Chun-Kyun Youn) 정회원



1982년 : 인하대학교 전자공학과 졸업
 1997년 : 포항공과대학 정보통신학과 석사
 1993년 : 조선대학교 전자계산학과 박사과정 수료
 1982년~1998년 : 포항종합제철

주 과 장
 1998년~현재 : 호남대학교 정보기술원 조교수
 <주관심분야> Network, 정보가전, 생산관리시스템, 공장자동화 시스템

이 찬 민(Chan-Min Lee) 정회원



1980년 : 전남대학교 공과대학 졸업
 1983년 : 전남대학교 공과대학 석사
 1997년 : 전남대학교 공과대학 공학박사
 2003년 : 조선대학교 전자계산학

과 박사과정수료
 1997년~1999년 : 전남대학교 사범대학 강사
 2001년~2002년 : 조선대학교 컴퓨터공학부 겸임조 교수
 1999년~현재 : 기독간호대학 전산교양 교수

<주관심분야> 전산시스템 관리, 네트워크 보안, 웹 프로그래밍

탁 동 길(Dong-Kil Tak) 학생회원



1998년 : 조선대학교 교육대학원 석사
 2000년 : 조선대학교 전자계산학과 박사과정
 2001년~현재 : 서강정보대학 인터넷정보과 겸임교수

<주관심분야> 정보보안, 전자상거래

정 일용(Il-Yong Chung) 정회원



1983년 : 한양대학교 공과대학 졸업
 1987년 : City University of New York 전산학 석사
 1991년 : City University of New York 전산학 박사
 1991년~1994년 : 한국전자통신

연구소 선임연구원
 1994년~현재 : 조선대학교 컴퓨터공학부 부교수
 1999년~2000년 : 조선대학교 정보전산원장

<주관심분야> 네트워크 보안, 전자상거래, 분산 시스템 관리, 코딩이론, 병렬 알고리즘