SNMP PDU의 시간변수 추가를 통한 네트워크 모니터링 성능 향상에 관한 연구

윤천균[†] · 정일용[††]

요 약

인터넷 환경에서는 일반 정보에 비해 수십 배 또는 수백 배가 큰 음성과 영상을 포함한 멀티미디어 정보가 전송된다. 네트워크 관리를 위한 분석 유형들은 실시간 분석, 기본분석과 심화분석으로 구성되며, 심화분석은 특정 object들에 대해 일정기간 주기적으로 경향정보를 수집하여 네트워크 상태를 분석하는데 유용하다. 심화분석 용 경향정보 수집을 위해 SNMP 적용 시 관리자의 폴링에 대해 에이전트가 매번 응답해야 하기 때문에 네트워크 부하 증가, 응답시간 지연, 데이터 수집의 정확성 감소를 초래한다. 본 논문에서는 기존 SNMP PDU에 시간변수를 추가하여 심화분석 시에 관리자와 에이전트간 불필요한 트래픽 발생을 최소화하고, 보다 정확하게 경향정보를 수집할 수 있는 효율적인 방안을 제안하고 구현하였다. 시험 분석 결과 기존 SNMP와 호환성을 유지하면서 네트워크 트래픽 부하가 감소하였으며, 정보수집의 정확도가 증가하였다.

A Study on an Improvement of Network Monitoring Performance by Adding Time Variables in SNMP PDU

Chun-Kyun Youn[†] and Il-Yong Chung[††]

ABSTRACT

Multimedia information containing voice and image is transmitted on Internet, which is ten times or hundred times larger than ordinary information. Analysis types for network management in this environment consist of a real time analysis, a basic analysis and an intensive analysis. The intensive analysis is useful for gathering the trend information of specific objects periodically for certain period in order to monitor network status. When SNMP is applied to collect the trend information of intensive analysis, it brings on the increase of network load, the delay of response time and the decrease of data collection accuracy since an agent responds to manager's every polling. In this paper, an efficient SNMP is proposed and implemented to add time variables in the existing SNMP PDU. It minimizes unnecessary traffic in the intensive analysis between manager and agent, and collects trend information more accurately. The results of experiments show that it has compatibility with the existing SNMP, decreases the amount of network traffic greatly and increases the accuracy of data collection.

Key words: SNMP, 트래픽 모니터링

1. 서 론

파일 용량이 다른 정보에 비해 수십 배 또는 수백 배 큰 음성, 영상 등의 멀티미디어 정보들이 대부분인 오늘날의 인터넷 환경에서 네트워크 대역폭은 수요에 비해 공급이 매우 부족한 상태이다[1]. 이러한 환경에서 효율적으로 네트워크를 운영하기 위해서는 네트워크 관리가 필수적이나 이를 위해 추가되는 트래픽 역시 부족한 대역폭에 부담으로 작용하기 때

접수일 : 2003년 4월 9일, 완료일 : 2003년 6월 4일
[†] 정회원, 호남대학교 정보기술학부 교수
[††] 종신회원, 조선대학교 정보통신공학부 교수

문에 이에 대한 감소 방안에 대한 많은 연구가 진행되고 있다[2,3].

인터넷 환경에서 효과적인 네트워크 관리를 위해서는 다양한 분석 파라미터들을 분류하여 분석항목을 구성하고 각 분석 요청에 따른 분석항목들의 범주를 조합해야 한다.

표 1은 각 분석항목과 분석항목들의 범주를 분류한 것으로 실시간 분석, 기본분석, 심화분석 항목으로 나누고 있으며, 그 중 심화분석 항목은 네트워크 장비의 특정 기간의 현황을 알아보기 위한 항목으로 관리범위에 따라 종합분석, 비교분석 등으로 분류하며 각 분석범위별 다양한 분석항목들이 있다[4-6].

위와 같은 심화분석 항목들은 실시간 분석과는 달리 특정 기간 동안의 현황을 알아보기 위하여 특정 object에 대하여 일정기간의 이력정보, 통계정보 등을 수집하여 경향을 감시해야 한다.

표 1. 인터넷 분석항목

분석 유형	관리범위	분 석 항 목
실시간 분석	성능 분석	이용율, 가용성, 인터페이스 패킷 송수신율, 입출력 트래픽 비율, 시스템 패킷 입출력율, 패킷 전달율
	장애 분석	에러 수신율, 인터페이스 패킷 손실율, 시스템 메모리 부하율, 시스템 패킷 손실율, 패킷 전달 실패율, 라우팅 에러율
	관리 트래픽 분석	관리 트래픽 이용량
기본분석	성능 분석	이용율, 가동율, 시스템 장애 횟수
심화분석	종합 분석	전체분석, 선로분석, 장비관련 분석, 장비 활용 분석
	시간대별 분석 월간/주간/일간 분석	에러 수신율, 인터페이스 패킷 송수신율, 인터페이스 패킷 손실율, 입력 트래픽 분석, 출력 트래픽 분석, 방송형 트래픽 분석, 시스템 메모리 부하율, 패킷 전달율, 라우팅 에러율
	비교 분석	유출입 바이트 량, 유출입 패킷량, 방송형 트래픽 분석(방송/ 비방송), 패킷당 바이트 량(송/수신), 정상/비정상 패킷 분석, 시스템 패킷 손실율, 시스템 패킷 입출력 분석
	분석 기간 중 비교 분석	유출입 바이트 량, 유출입 패킷량, 방송형 트래픽 분석(방송/ 비방송), 패킷당 바이트 량(송/수신), 정상/비정상 패킷 분석, 시스템 패킷 손실율, 시스템 패킷 입출력 분석

네트워크 관리방법으로 가장 많이 사용하고 있는 TCP/IP기반의 SNMP(Simple Network Management Protocol)의 메시지 송수신 방법은 UDP 상에서 관리자가 측정하고자 하는 MIB object에 대하여 에이전트에 정보 송신을 요청하고, 에이전트가 이에 응답하는 방식으로 단일 object에 대한 요청에 대하여 단일 응답이 이루어지는 형태이다[7]. 이러한 SNMP를 이용하여 심화분석 용 네트워크 관리 정보를 수집할 경우 그림 1과 같이 관리자가 해당 MIB object에 대하여 "GetRequest"를 에이전트에 반복적으로 폴링하고, 에이전트는 "GetResponse"를 매 폴링에 응답하는 데이터 송수신이 이루어진다[8,9]. 이와 같은 반복적인 요청과 응답으로 인하여 발생되는 데이터 송수신은 네트워크 트래픽의 증가, 관리자 시스템의 부하 가중 그리고 응답시간 지연 등의 문제를 일으킨다[10-12]. 뿐만 아니라 네트워크의 부하상태에 따라 응답시간이 불균일하여 에이전트에서 수집되는 정보가 일정한 시간 간격으로 측정되지 않는 문제점이 있다.

본 논문에서는 이와 같이 일정기간 동안 반복수집이 필요한 심화분석 항목들에 대한 정보를 "경향정보(Trend Information)"라 정의하고, 이러한 경향정보 수집을 위해 SNMP 적용 시 발생하는 문제를 효율적으로 처리하여 네트워크 트래픽 부하 감소와 정확한 정보수집이 가능한 개선방법에 대하여 제안하고, prototype을 구현하여 시험한 후 그 결과를 분석하고자 한다. 2장에서는 SNMP의 기본적인 개념에 대하여 이해하고, 제안하고자 하는 SNMP 모델과 그 동작 원리를 설명한다. 3장에서는 이를 구현하여 기

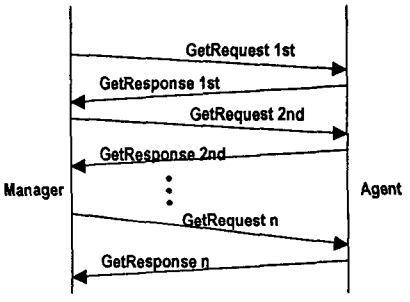


그림 1. 기존 SNMP의 동작 과정

존 SNMP와 비교 실험을 하기 위한 prototype 시험 bed 구축 및 이를 이용하여 개선된 SNMP의 성능을 검증하기로 한다. 마지막으로 4장에서 이 논문을 결론짓고자 한다.

2. 관련연구

2.1 기존 SNMP의 기능 및 PDU 구성

SNMP의 구성은 그림 2와 같이 관리 시스템(Manager), SNMP 에이전트(Agent), MIB (Management Information Base), 네트워크 관리 프로토콜인 SNMP로 이루어진다[9].

SNMP는 UDP상에서 동작하는 비동기식 요청/응답 메시지 프로토콜로서 관리 시스템의 관리자와 관리대상 에이전트 간에는 그림 3에서와 같이 관리자가 에이전트에게 GetRequest, Set-Request, GetNextRequest 등의 PDU를 보내면 에이전트가 이와 연관되는 결과를 GetResponse PDU를 이용하여 관리자에게 전송하는 방식으로 MIB 정보를 수집한다. 또 관리자의 요구 없이 에이전트가 관리자에게 정보 전달을 필요로 하는 경우 에이전트는 trap을 발생시켜 장애나 오류 등의 필요한 정보를 관리자에게 전달한다[8,9]. SNMP는 바로 이 둘 사이의 통신을 위해 사용되는 표준화된 프로토콜로서 구현이 쉽고 간편해서 네트워크 관리를 요구하는 대부분의 네트워크 장치에 보편적으로 사용되고 있다[7].

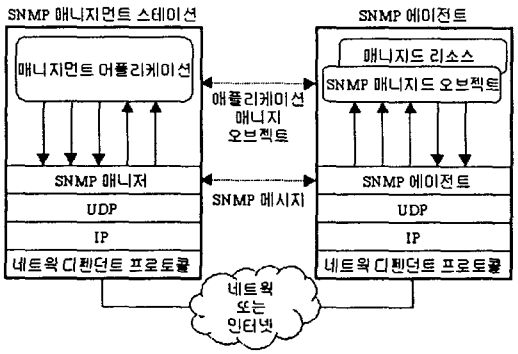


그림 2. SNMP 프로토콜 아키텍처

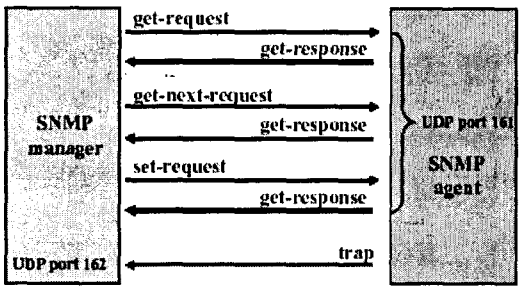


그림 3. SNMP 메시지 동작 과정

SNMP의 명령은 GetRequest, GetNextRe-quest, GetResponse, SetRequest, Trap으로 구성되어 있으며, 다음과 같은 명령어를 이용하여 관리자와 에이전트간 정보를 송수신 한다[9,13]. GetRequest는 관리자에서 피 관리 시스템인 에이전트에게 관리정보 수집을 요청하는 명령이고, GetNextRequest는 MIB가 계층적 구조를 유지하고 있으므로 관리자가 에이전트에게 해당 tree의 다음 하위계층 MIB를 읽도록 요청하는 명령이다. GetResponse는 피 관리 시스템인 에이전트에서 관리자에게 GetRequest, GetNextRequest에 의해 요청된 결과를 전송하는 명령이고, SetRequest는 관리자의 지시로 피 관리 시스템인 에이전트의 MIB를 설정하여 장비를 제어하기 위한 명령이다. Trap은 에이전트가 관리자에게 보고하는 Event로 시스템에 변화가 있을 경우 에이전트는 이를 감지해서 MIB에 미리 정의된 보고 유형에 따라 보고 한다.

SNMP PDU는 그림 4와 같은 포맷으로 구성되어 있으며, Version은 관리자와 에이전트 간에 사용 중인 SNMP 버전이다. v1, v2, v3가 있으며 메시지 포맷은 서로 약간 다르다. 만일 관리자와 에이전트간의 버전이 일치하지 않으면 에러가 발생한다. Community는 관리자와 에이전트 간에 간단한 인증을 위한 것으로 미리 정의된 값이 서로 일치해야 한다. 기본적으로 Get에는 'public', Set에는 'private'를 사용한다. SNMP-PDU는 관리자와 에이전트 사이의 요청과 응답에 대한 실제 정보가 저장되는 필드이다.

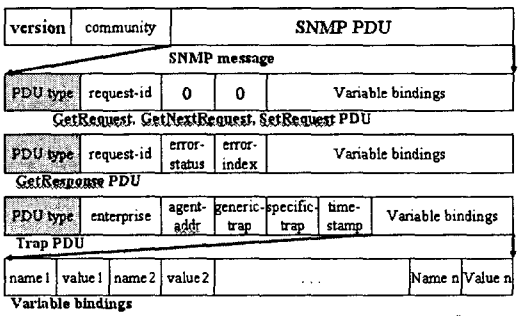


그림 4. SNMP PDU Format

PDU-type은 관리자가 에이전트에게 요구하거나 에이전트로부터 관리자로 결과가 올 때 어떤 요청인지를 알려주기 위한 것으로 GetRequest(0)/GetNext-Request(1)/SetRequest(2)/SetResponse(3)/Trap(4) 등이 있다. RequestID는 관리자가 에이전트에 요구를 보낼 때 부여하는 ID로 그 결과값은 부여한 ID와 동일한 ID를 보고 판단 할 수 있다. Error-status는 에이전트가 관리자의 요구를 해석하고 처리하는데 어떤 문제가 발행 했을 때 표시한다. noError(0), tooBig(1), noSuchName(2), badValue(3), readOnly(4), genError(5) 등이 있다. Error-index는 variable-bindings에서 특정 위치의 값이 오류가 발생했을 때 몇 번째 필드인지를 표시한다. variable-bindings는 관리자는 한번에 1개의 MIB object에 명령을 내리는 것이 아니라 여러 가지의 MIB object에 동시에 내릴 수 있다. 따라서 원하는MIB object를 1~N개까지 (OID+VALUE).....(OID+VALUE) 형태로 묶어서 요청할 수 있다.

2.2 기존 SNMP 문제점 및 개선 SNMP 모델 제안

본 논문의 연구 대상은 1절에서 언급한대로 일정기간 동안 반복수집이 필요한 경향정보로, 기존 SNMP에서 이러한 경향정보를 일정기간 감시하기 위해서는 그림 1과 같이 관리자가 수집하고자 하는 MIB object에 대하여 그림 4와 같은 형태의 "GetRequest"를 일정한 주기를 갖고 반복적으로 에이전트에 폴링하고, 에이전트는 응답으로 "GetResponse"들을 해당 횟수만큼 송신한다. 이와 같은 형태의 반복적인 정보 송수신으로 인하여 네트워크 트래픽을 과다하게 발생시키는 문제가 있다. 또한 일정 주기의 정보수집이 요구되는 경우 수시로 변화하는 네트워크의 부하에 따라 각 응답시간이 달라질 수 있기 때문에 에이전트의 정보수집 시간 간격이 일정하지 않게 되어 수집된 데이터의 정확성이 떨어질 수 있다.

이와 같은 문제점들을 개선하기 위하여 기존 SNMP PDU에 효율적인 경향정보 수집을 위하여 그림 5와 같이 GetTiRequest와 GetTiRes-ponse PDU를 추가하여 그림 6과 같이 동작하는 개선된 SNMP 모델을 제안한다.

개선된 SNMP 모델에서는 관리자에서 기존의 "GetRequest" PDU와 자료수집 시작시간(start time), 자료수집 종료시간(end time), 자료수집 주기(time interval), 전송 요구시간(sending time) 필드를 이용하여 그림 5와 같은 "GetTiRequest" PDU를 에이전트에게 보낸다. 이를 수신한 에이전트는 수신한 PDU의 PDU 형식을 분석하여 기존의 "GetRequest" PDU면 기존 SNMP 방식과 같이 처리되도록 한다. 경향정보용 "GetTiRequest" PDU일 경우는 PDU header로부터 Request-id, 자료수집 시작시간, 자료수집 종료시간, 자료수집 주기, 전송 요구시간, variable-bindings의 부분을 인식하여 그림 6에서와 같이 지정된 시작 시간부터 주어진 주기로 자료수집 종료시각까지 데이터를 수집하여 임시 보관하고 있다가 전송요구 시간이 되면 한 개의 "GetTiResponse" PDU를 생성하여 관리자에게 보낸다.

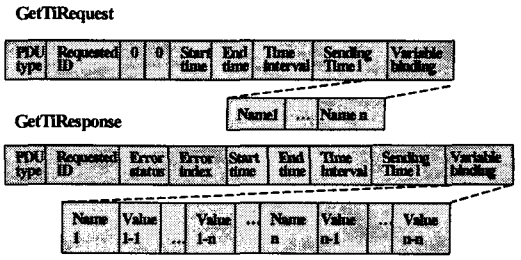


그림 5. 경향정보 용 추가 SNMP PDU

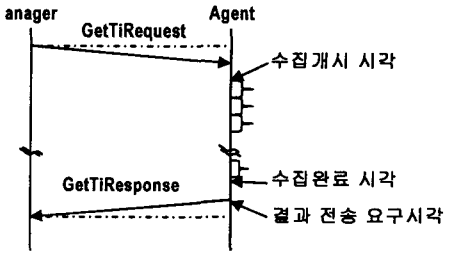


그림 6. 개선된 경향정보 용 PDU 송수신 수순

개선안에서는 위와 같이 네트워크 상태를 측정하고자 하는 시간대와 측정 주기, 그 결과를 수신하고자 하는 시간을 사전에 지정할 수 있어 관리자에게 많은 유연성을 제공할 수 있으며, 관리자와 에이전트 간에 1회씩만 메시지를 송수신 함으로써 기존 SNMP에서의 반복적인 폴링과 응답으로 인한 불필요한 네트워크 트래픽을 감소시키고, 관리자 시스템의 부하감소 및 에이전트 시스템에서의 정확한 정보수집 등의 효과를 낼 수 있다.

2.3 개선된 SNMP의 PDU 구성

그림 5와 같이 추가된 "GetTiRequest"와

"GetTiResponse" PDU의 구성을 살펴보면, start time, end time, time interval, sending time, variable binding 필드로 구성된다. 이 5가지 필드들이 경향정보들을 주기적으로 수집하는 관련 정보들을 포함하고 있으며, start time은 요구하는 MIB 변수에 대한 자료수집 시작시간을 나타낸다. end time은 요구하는 MIB 변수에 대한 자료수집 종료시간을 나타내고, time interval은 요구하는 MIB 변수에 대한 자료수집 주기로 초(sec) 단위를 지정하게 된다. 이 값 간격으로 SNMP 에이전트에서 해당 변수의 MIB값을 수집하여 저장한다. sending time은 관리자에서 에이전트로부터 수집된 결과를 수신하고자 하는 시각을 나타내며, variable binding for GetTiRe- quest는 name 1, name 2, ... name n 형태로 여러 개의 object를 동시에 요청할 수 있도록 변수명을 표기할 수 있다. variable binding for GetTiResponse는 GetTiRequest PDU에서 요청한 변수 name 1, name 2 ..... name n 각각에 대하여 자료수집 요청시간 동안 주기에 따라 수집된 값을 변수명과 함께 표기한다.

2.4 개선된 SNMP 모델의 모듈 구성 및 기능

제안한 SNMP 모델은 그림 7과 같이 관리자와 에이전트로 구분되며, 관리자는 웹 모듈과 SNMP Get 모듈로 이루어지고, 에이전트는 수신된 SNMP 패킷을 구분하는 모듈과 이를 처리하는 SNMP Handler 모듈로 구성된다.

관리자의 각 모듈 기능을 살펴보면, 웹 모듈은 웹을 통하여 관리자에게 Interface를 제공하는 기능으로 측정하고자 하는 MIB object를 선정하고 자료수집 시작시간, 종료시간, 수집주기, 전송시간을 설정할 수 있도록 기존 SNMP와 제안한 경향정보용 모듈로 구성되어 있다. SNMP Get 모듈은 웹 화면에서 입력된 정보를 이용하여 기존 SNMP 형태의 정보수집 요구와 개선된 경향정보 수집 요구에 따라 GetRequest와 GetTiRequest PDU를 생성하고 이를 에이전트에 전송하는 기능을 수행하며, 에이전트로부터 수신한 측정결과 데이터들을 처리하는 기능을 수행한다.

에이전트의 각 모듈 기능을 살펴보면 SNMP Packet Reader/Identifier 모듈은 관리자로부터 수신한 PDU를 읽고 분석하여 기존의 GetRequest와 GetTiRequest PDU를 구분하여 해당되는 handler

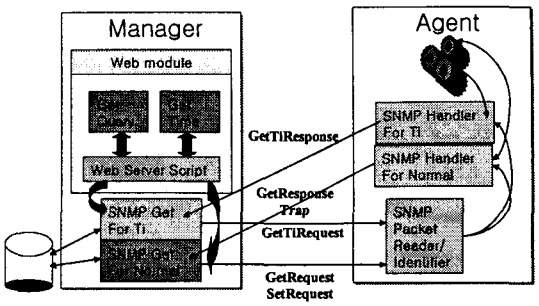


그림 7. 개선된 SNMP 모델의 구성도

모듈을 기동한다. SNMP handler 모듈은 기존 SNMP용과 개선된 Ti용으로 구분되어 요청된 변수와 시간정보들에 따라 Resource들로부터 정보를 수집하여 관리자에게 전달하는 기능을 수행한다.

개선된 SNMP에서는 이러한 모듈들로 구성되었기 때문에 기존의 SNMP 기능들을 그대로 사용할 수 있으면서 경향정보를 효율적으로 처리할 수 있다.

3. 시험 환경 및 결과 분석

3.1 시험 환경

제안한 SNMP를 구현하고 시험을 위하여 네트워크 트래픽 부하를 측정하는 시험과 응답시간을 측정하는 시험으로 분리하였다. 관리자로는 썬의 Blade 100 Unix (Solaris2.8) 시스템을 사용하였고, 에이전트는 일반 PC에 Linux(Redhat 6.2)를 설치하였으며, 측정 시스템으로 Windows XP를 설치한 일반 PC로 시험 bed를 구축하였다. 관리자와 에이전트용 SNMP 프로그램은 SNMP version 2를 지원하는 UCD SNMP v3.4를 C언어를 이용하여 모듈을 수정하고 추가하였으며, 외부에서 인터넷으로 접속하여 관리자 시스템을 이용하여 시험을 할 수 있도록 관리자 시스템에 아파치 웹 서버를 구축하고 관리자 기동용 관리화면을 shell 기반 CGI를 이용하여 기존 SNMP용과 개선된 SNMP용으로 분리하여 작성하였다. 측정 시스템에는 네트워크 측정 Tool인 EtherPeek (WildPackets, inc.)와 App-Dancer FA (AppDancer Networks, inc.)를 이용하여 트래픽과 응답시간, 프레임의 크기 등을 측정하였다. 응답시간의 정확한 측정을 위해 각 시스템 간 시간 동기화 문제를 해결하기 위하여 NTP(Network Time Protocol)을 관리자 시스템과 에이전트 시스템에 설치하고, 측정 시스템

은 OS에서 제공하는 동기화 기능을 이용하였다.

3.1.1 네트워크 트래픽 부하 측정을 위한 시험 Bed

경향정보 수집 시 네트워크 트래픽 부하를 시험하기 위하여 그림 8과 같이 타 네트워크에 연결되지 않은 독립된 형태의 시험 bed를 구축하고 기존의 SNMP 기능 수행 시와 개선된 SNMP 수행시의 트래픽 부하를 측정 비교한다. 시험 bed의 각 시스템들은 shared media인 단순 허브를 공유하는 이더넷 LAN 형태이다.

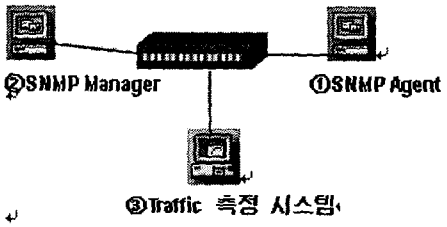


그림 8. 트래픽 부하 측정을 위한 시험bed 구성도

3.1.2 응답시간 측정을 위한 시험 Bed

시험을 위하여 그림 9와 같이 호남대학교에 위치한 시스템(211.227.240.155)에 관리자를 설치하고, 에이전트는 같은 네트워크에 있는 시스템(211.227.238.251)에 설치하였으며, WAN 환경의 에이전트는 같은 광주광역시이지만 지리적으로 서로 떨어진 조선대 시스템(220.67.202.129)에 설치하였다. 본 시험에서 LAN 환경에서의 홉(hop) 수는 1이며, WAN 환경의 홉 수는 7이다.

응답시간 측정 시 관리자와 에이전트 간의 시간 동기는 매우 중요하므로 NTP를 반듯이 기동시켜야 한다.

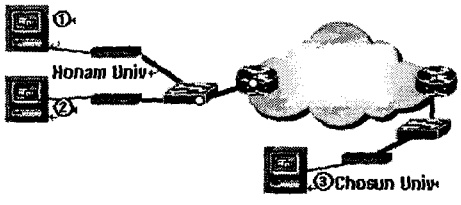


그림 9. 응답시간 시험을 위한 시험 bed 구성도

3.2 시험 결과 및 분석

3.2.1 네트워크 트래픽

그림 8의 트래픽 측정 시스템에서 웹을 이용하여 원격으로 관리자 시스템을 기동시켜 SNMP 관리자와 에이전트 간의 SNMP 송수신을 발생시키고, 네트워크 관리 도구들로 네트워크 상의 트래픽을 측정한다. 시험 편의를 위하여 개선된 SNMP 경우 수집 종료시간과 전송시간을 동일하게 설정하여 에이전트에서 수집완료 시 바로 관리자에 전송토록 한다.

시험 bed을 이용한 시험조건은 기존 SNMP와 개선된 SNMP 경우에 대하여 관리자에서 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)를 에이전트에 5초 주기로 30분간 전송하여 측정한 네트워크 트래픽을 1분 주기의 평균값으로 표기하여 비교하였다.

1) 기존 SNMP에 의한 네트워크 트래픽

기존 SNMP에서 관리자와 에이전트간의 네트워크 트래픽 발생은 그림 10과 같으며, 그래프 상의 각 시간은 다음과 같은 의미를 갖는다.

- T1 : Manager에서 Agent에 전송 시 소요되는 전송지연 시간
- T2 : Agent에서 Manager에 전송 시 소요되는 전송지연 시간
- T3 : Manager에서 Agent의 응답 수신 후 소요되는 처리시간 및 대기 시간
- T4 : Agent에서 Manager의 요청 수신 후 소요되는 처리시간으로 CPU에서 처리하는데 걸리는 시간으로 다른 시간에 비하여 매우 짧기 때문에 거의 0ms라고 해도 무리가 없다.

그림 10에서 관리자가 발생시킨 GetRe-quest PDU에 의한 트래픽 양을 Tgrq라 하고, 에이전트의 응답인 GetResponse PDU에 의한 트래픽 양을 Tgrp라 하면, 한 개의 정보수집 시 발생되는 트래픽 양(Tf)은 식(1)과 같으며, 정보 수집기간이 Dt이며 수집 간격이 It일 경우의 발생되는 총 트래픽 양(Ttf)은 식(2)와 같다.

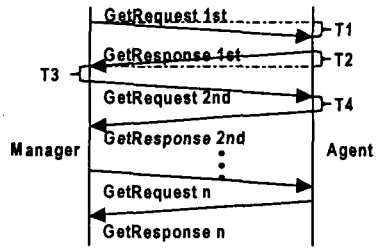


그림 10. 경향정보수집 시 기존 SNMP의 동작 과정

$$Tf = Tgrq + Tgrp \quad (1)$$

$$Ttf = Dt/It * Tf \quad (2)$$

식(2)에서 알 수 있듯이 정보 수집기간이 길고, 주기가 짧을수록 발생되는 트래픽 양이 증가한다.

기존 SNMP를 이용하여 측정된 트래픽은 그림 11과 같으며, 평균 약 127Bytes/초의 트래픽이 발생되었으며 트래픽이 일정하지 않은 이유는 구축된 시험 bed에 SNMP이외에도 측정 시스템에서 웹을 이용하여 관리자 기동 시 발생되는 HTTP와 Windows Xp에서 발생되는 NetBIOS 등의 프로토콜들이 추가되었기 때문임.

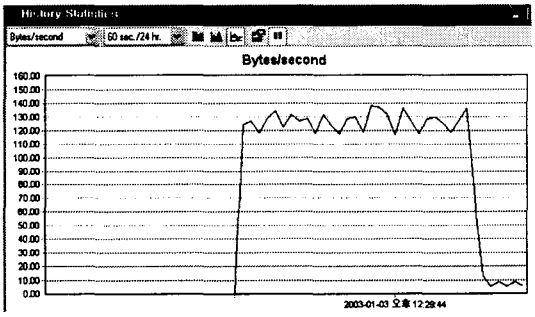


그림 11. 기존 SNMP를 이용하여 측정된 트래픽 부하

2) 개선된 SNMP에 의한 네트워크 트래픽

개선된 SNMP에서 경향정보 수집 시 관리자와 에이전트간의 네트워크 트래픽 발생 과정은 그림 6과 같다. 이 과정에서 관리자가 발생시킨 Get-TiRequest PDU에 의한 트래픽 양을 Tegrq라 하고, 이에 대한 에이전트의 응답인 GetTi- Response PDU에 의한 트래픽 양을 Tegrp라 하면, 한 개의 정보수집 시 발생되는 트래픽 양(Tef)은 식(3)과 같으며, 정보 수집기간이 Dt이며 수집 간격이 It일 경우의 발생되는 총 트래픽 양(Tetf)은 식(4)와 같이 표현할 수 있다.

$$Tef = Tegrq + Tegrp \quad (3)$$

$$Tetf = Tegrq + Tegrp = Tef \quad (4)$$

개선된 SNMP에서 그림 12와 같이 평균 약 21bytes/초의 트래픽이 발생되었으며, 순수 SNMP 트래픽은 관리자가 에이전트에 GetTi-Request PDU를 전송하는 초반과 정보수집이 완료되어 에이전트가 GetTiResponse PDU를 관리자에 전송하는 마지막 부분에서 발생 되었다.

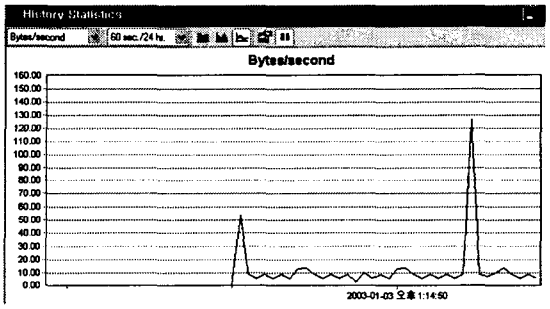


그림 12. 개선된 SNMP를 이용하여 측정된 트래픽 부하

3) 비교 분석

기존의 SNMP와 개선된 SNMP를 비교해 보면 단일전송의 경우 그림 4와 그림 5의 PDU구성에서 알 수 있듯이 GetTiRequest와 GetTiR-esponse가 GetRequest와 GetResponse보다 크기 때문에 $Tegrq \geq Tgrq$, $Tegrp \geq Tgrp$의 관계가 성립되어 식(1)과 식(3)에서 $Tef \geq Tf$ 관계가 된다. 이들 간의 차이는 요청하는 변수에 따라 약간 차이가 발생하지만 시험 기준에 의하면 기존 SNMP에서 ***21bits (108-87), 61bits (152-91)*** 씩 적게 발생한다.

반면에 본 논문에서 중점을 둔 경향정보의 경우 그림 11과 그림 12에서 알 수 있듯이 전체 트래픽은 개선된 SNMP에서 6배 감소하였고, 순수 SNMP에 의하여 발생된 트래픽만을 비교해 보면 기존 SNMP는 식(1)과 식(2)에 따라 ***64,080 bytes((87B+ 91B) *60/5초*30분)***가 발생되었고, 개선된 SNMP의 경우에는 식(3)과 식(4)에 의하면 ***1,622 bytes (108B+ 1,514B)***가 발생하여 개선된 SNMP의 네트워크 트래픽이 약 ***39.5***배 감소하였다.

결론적으로, 기존 SNMP의 경우 데이터 송수신 횟수에 정비례하여 네트워크 트래픽이 증가하는 반면, 개선된 SNMP의 트래픽은 측정횟수에 따라 다소 증가하나 큰 차이가 없기 때문에 측정기간이 길고

측정 주기가 짧아질수록 개선된 SNMP의 네트워크 트래픽이 상대적으로 크게 감소효과를 나타낸다.

3.2.2 응답시간

심화분석 항목 중에서 일정한 주기의 측정값이 요구되는 경우 수집된 정보의 정확성은 응답시간에 따라 달라지므로 응답시간의 균일화 정도를 측정하여 데이터 수집의 정확성을 비교할 수 있다. 응답시간은 여러 가지 여건에 따라 차이가 발생하나 근거리와 원거리일 경우 물리적 거리가 달라지기 때문에 라우팅 경로상의 중간 Node 수의 차가 발생하고, 이로 인하여 달라지는 기존 SNMP와 개선된 SNMP에서의 응답시간의 변화를 분석하고자 한다. 따라서 관리자 시스템의 위치는 고정하고, 에이전트의 설치 위치를 그림 9와 같이 LAN 환경과 WAN 환경으로 구분하여 응답시간을 측정, 비교하였다.

응답시간의 균일화 정도 비교를 위해 기존SNMP와 개선된 SNMP에서 관리자가 각 에이전트에 대하여 동일한 MIB object "ifInOctets (1.3.6.1.2.1.2.2.1.10)"을 5초 주기로 30분간 폴링하여 그 응답 수신 시까지 소요되는 시간을 1분 주기의 평균값으로 표기하여 비교한다.

그림 10과 같이 동작하는 기존 SNMP에서 관리자가 특정 object에 대하여 1회의 정보수신을 위하여 소요되는 응답시간(Rt)은 식(5)와 같다.

$$Rt = T1 + T2 + T4 \fallingdotseq T1 + T2 \quad (5)$$

(여기서, T1은 관리자에서 에이전트에 전송 시 소요되는 전송지연 시간이고, T2는 에이전트에서 관리자에 전송 시 소요되는 전송지연 시간이다. T3는 관리자에서 에이전트의 응답 수신 후 소요되는 처리시간 및 대기 시간이며, T4는 에이전트에서 관리자의 요청 수신 후 소요되는 처리시간으로 CPU에서 처리하는데 걸리는 시간으로 다른 시간에 비하여 매우 짧기 때문에 거의 0ms 라고 해도 무리가 없다.

정보 수집기간이 Dt이며 수집주기가 It일 경우인 경향정보를 수집 시 소요되는 총 응답시간(Rtt)은 식(6)과 같다.

$$Rtt=Dt/It*Rt=Dt/It*(T1+T2) \quad (6)$$

식(6)에서 알 수 있듯이 기존 SNMP에서의 총 응답시간은 경향정보 수집 기간이 길고 주기가 짧을수록 증가한다.

그림 6과 같이 동작하는 개선된 SNMP의 경우 관리자의 GetTiRequest에 대하여 에이전트에서는 그림 13과 같은 처리가 이루어진다.

여기서, T5는 요청한 object의 값을 획득한 후 다음 주기까지 대기하는 시간으로 기존 SNMP에서는 결과를 관리자에 전송하고 관리자로부터 다시 GetRequest를 수신할 때까지 대기하는 T3에 해당한다. T6은 데이터 수집 완료 후 전송시각까지 대기하는 시간으로 이 시간도 응답시간 계산에서는 제외한다. T7은 관리자로부터 미리 수신한 GetRequest를 수집 개시시간까지 대기하는 시간으로 응답시간에는 무관하므로 제외시킨다. 엄격한 응답시간의 정의에 준하여 개선된 SNMP와 기존 SNMP의 응답시간 비교는 곤란하지만 성능비교를 위하여 그림 13과 같이 처리되는 개선된 SNMP의 응답시간은 다음과 같이 3가지 형태로 분류하여 비교가 가능하다.

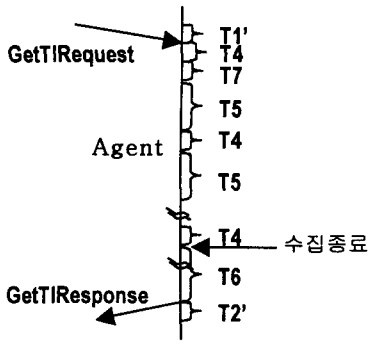


그림 13. 개선된 SNMP에서 에이전트의 내부 처리 과정

처음 부분의 응답시간(Rte1)은

$$Rte1=T1'+T4 \fallingdotseq T1' \quad (7)$$

Rte1~T6 전까지의 응답시간(Rte2)은

$$Rte2 = T4 \quad (8)$$

마지막 부분의 응답시간(Rte3)은

$$Rte3 = T2' \quad (9)$$

로 표기할 수 있으며, Rte2 구간의 누적 응답시간은 $Dt/It*T4$ 이므로 개선된 SNMP에서의 총 응답시간(Rtte)은 $Rtte = Rte1+Rte2+Rte3 \fallingdotseq Dt/It* T4+T1'+T2' \ (10)$과 같이 표기할 수 있다.

T1, T2와 T1', T2'의 차이는 그림 4와 그림 5에서 알 수 있듯이 GetRequest, GetResponse PDU와

GetTiRequest, GetTiResponse PDU의 길이 차로 $T1' \geq T1$, $T2' \geq T2$의 관계가 성립되나, GetRequest와 GetTiRequest PDU 길이 차이는 21 bytes로 크지 않기 때문에 전송 소요시간 차가 거의 없다고 볼 수 있기 때문에 T1≒T1′ 이라고 할 수 있으나, T2와 T2′는 정보 수집시간이 길 경우 그 차가 커지므로 $T2' \geq T2$가 유지된다. 따라서 식(10)은

$$Rtte \doteqdot Dt/It*T4+T1+T2' \quad (11)$$

로 표기할 수 있다.

1) LAN 환경에서의 응답시간 실험 결과

그림 14에서 평균 응답시간이 기존 SNMP **3.1msec**, 개선된 SNMP **0.8msec**로 개선된 경우가 더 빠르고 안정적임을 알 수 있고, 관리자가 에이전트에게 GetTiRequest를 전송하는 시점의 응답시간은 기존 SNMP와 유사하나 마지막 수집결과 전송 시 상대적으로 패킷 크기가 매우 크므로 더 많은 시간이 소요됨을 알 수 있다. 누적 응답시간은 기존 SNMP가 **94msec**, 개선된 SNMP에서 **24msec**로 약 4배가 개선되었음을 알 수 있다.

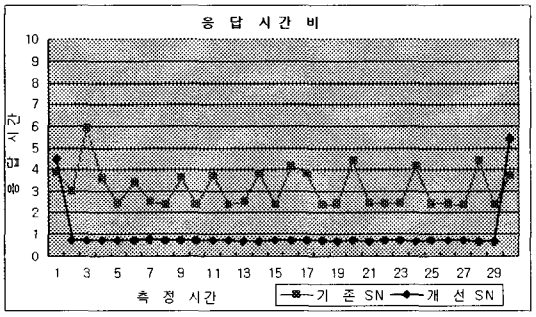


그림 14. LAN 환경에서의 응답시간 측정치

2) WAN 환경에서의 응답시간 실험 결과

그림 15의 WAN 환경에서 평균 응답시간은 기존 SNMP가 **118msec**, 개선된 SNMP에서 **7msec**로 약 17배의 차이를 보이고, 누적 응답시간은 각각 **3,539 msec**와 **209msec**를 나타내고 있다. 기존 SNMP의 경우 네트워크의 부하에 민감하므로 응답시간이 매우 불규칙하게 나타나고 개선된SNMP에서는 네트워크의 부하에 영향을 받는 초기와 마지막을 제외하면 매우 안정적임을 알 수 있다.

3) 비교 분석

기존 SNMP의 경우 식(5)와 식(6)에서 알 수 있듯이 응답시간이 네트워크의 부하 변동에 따라 T1과 T2가 변하므로 응답시간의 균일화는 이루어지기 어렵다. 시험 결과인 그림 14와 그림 15의 기존 SNMP 그래프에서도 응답시간의 불규칙성을볼 수 있으며, LAN 환경 보다 WAN 환경에서 훨씬 불규칙함을 알 수 있다.

반면에 개선된 SNMP의 경우 식(7-9)에서 알 수 있듯이 초기와 말기의 식(7)과 식(9)의 경우에만 네트워크의 부하 변동에 영향을 받고, 에이전트 내부에서 처리 중에는 에이전트 시스템의 부하에 따라 T4가 변하므로 응답시간이 비교적 균일하다. 그림 14와 그림 15의 개선된 SNMP 그래프에서도 처음과 마지막을 제외하고는 LAN 및 WAN 환경 모두 균일한 응답시간을 잘 보여주고 있다. 개선된 SNMP에서 측정된 응답시간의 평균 값 중 초기와 마지막 부분을 제외한 부분에서 LAN 환경 **0.7msec**, WAN 환경 **1.1msec**로 서로 차이가 있는 것은 그림 9에서 LAN 에이전트 시스템(②)은 Pentium-4, 1.70GHz이고, WAN 에이전트 시스템(③)은 Pentium-3, 866MHZ으로 두 시스템의 성능 차 때문에 발생하였다.

누적 응답시간도 기존 SNMP의 경우인 식(6)과 개선된 경우의 식(11)을 비교해 보면 관리자와 에이전트간의 전송시간인 T1과 T2가 에이전트 내부 처리 시간인 T4에 비해서 매우 크므로 개선된 SNMP 경우가 짧으며, 시험결과 그림 14와 그림 15에서도 기존 SNMP (**LAN: 94msec, WAN: 3,539msec**)와 개선된 SNMP (**LAN: 24msec, WAN: 209msec**)를 비교하면 기존 SNMP가 LAN환경에서 약 ***4배***, WAN 환경에서 약 ***17배*** 크다는 것을 알 수 있다.

결과적으로 상대적으로 개선된 SNMP의 응답시간이 짧고 균일하여 정보수집 정확성이 더 높다고 할 수 있다.

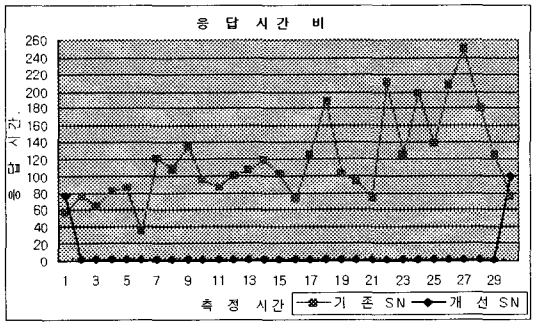


그림 15. WAN 환경에서의 응답시간 측정치

4. 결 론

시험 bed에서의 시험결과를 바탕으로 개선된 SNMP의 특징을 정리해보면 첫째, 기존 SNMP에 비해 적은 횟수의 요구와 응답 메시지를 송수신 함으로써 네트워크 트래픽 부하를 크게 줄일 수 있다.

둘째, 응답시간의 균일성을 확보하여 일정한 정보수집 간격이 요구되는 경우 수집된 데이터의 정확성을 향상시킬 수 있다. 이러한 효과는 근거리보다 원거리에서 크게 나타나므로 점점 복잡해지고 관리범위가 확장되어 가는 Global 환경에서의 네트워크 관리를 효율적으로 할 수 있다.

셋째, 기존의 SNMP에 대하여 호환성을 유지한다. 개선된 SNMP 에이전트는 기존 SNMP 에이전트의 기능과 완전하게 호환성을 유지하면서 네트워크 심화분석 분야의 경향정보 수집 시 트래픽 감소 장점을 갖고 있다.

넷째, 개선된 SNMP의 PDU에 시간변수를 추가하여 네트워크 상태를 측정하기 원하는 시간대와 측정주기, 그 결과를 전송 받기를 원하는 시간을 사전에 지정할 수 있어 관리자에게 관리의 유연성을 제공할 수 있다.

참 고 문 헌

[1] Nalin Sharda, Multimedia networks: fundamentals and future directions, Association for Information Systems, 1999.

[2] Nathan J. Muller, "Improving and managing multimedia performance over TCP IP nets", International Journal of Network Management, Volume 8, Issue 6. pp. 356-367, December 1998.

[3] Jacobus van der Merwe, Ramn Cceres, Yang-hua Chu, Cormac Sreenan, "mmdump: a tool for monitoring internet multimedia traffic", ACM SIGCOMM Computer Communication Review, Volume 30 Issue 5, pp. 48-59, October 2000.

[4] Sang-chul Shin, Seong-jin Ahn, jin-Wook Chung, "A new approach to gather network management data periodically," ITC-CSCC '97, 1997.

[5] Sang-chul Shin, Seong-jin Ahn, jin-Wook Chung, "Design and Impelementation of SNMP based performance parameter extraction system", Asia-pacific network operations and Management Symposium, 1997.

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

[7] Amatzia Ben-Artzi, Asheem Chandna, Unni Warrier, "Network Management of TCP/IP Networks: presents and future", IEEE Network Magazine, pp. 35-43, July 1990.

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

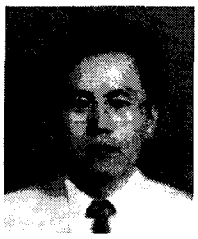
[9] William Stallings, SNMP, SNMPv2, SNMPv3, and RMON1 and 2, Third edition, Addison-Weseley, 1996.

[10] 천진영, 정진하, 윤완오, 최상방, "SNMP기반 네트워크 관리를 위한 적응형 네트워크 모니터링 방법", 한국통신학회논문지, 제27권 제12C호, pp. 1265-1275, 2002.

[11] 김민우, 박승균, 오영환, "SNMP 트래픽 최적화를 위한 폴링 방식에 관한 연구", 한국통신학회논문지, 제26권 제6A호, pp. 1051-1058, 2001.

[12] M. checkhrouhou and J, Labetoulle, "An Efficient polling Layer for SNMP", proceddings of the 2000 IEEE/IFIP Network operations and Management System. pp. 447-490, 2000.

[13] K. McCloghrie, M. Rose. "Management Information Base for Network Management of TCP/IP based Internets: MIB-II." Internet RFC 1213, March 1991.



윤 천 균

1982년 인하대학교 전자공학과 졸업
1997년 포항공과대학 정보통신학과 석사
2003년 조선대학교 전자계산학과 이학박사

1982년~1998년 포항종합제철㈜ 과장
1998년~현재 호남대학교 정보기술학부 조교수
관심분야 : Network, 정보가전, 공장자동화

정 일 용

1983년 한양대학교 공과대학 졸업
1987년 City University of New York 전산학 석사
1991년 City University of New York 전산학 박사
1991년~1994년 한국전자통신연구소 선임연구원
1994년~현재 조선대학교 컴퓨터공학부 부교수
1999년~2000년 조선대학교 정보전산원장
관심분야 : 네트워크 보안, 전자상거래, 분산 시스템 관리, 코딩이론, 병렬 알고리즘

교 신 저 자

윤 천 균 506-714 광주광역시 광산구 서봉동 59-1번지
호남대학교 정보기술원