네트워크 감시 프로토콜 기능 향상에 관한 연구

윤천균*
*호남대학교 정보기술학부
e-mail : chqyoun@honam.ac.kr

A Study on Function Improvement of a Network Monitoring Protocol

Chun-Kyun Youn*
*Division of Information Technology, Honam University

요 약

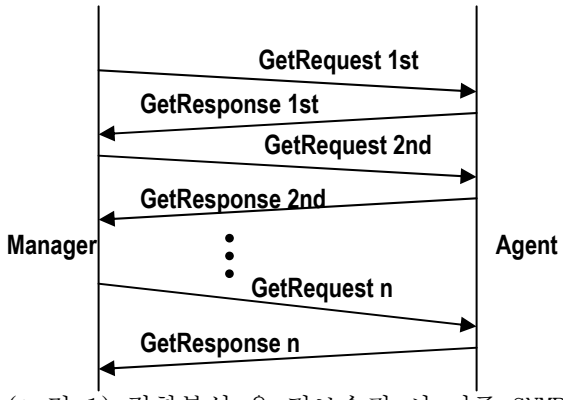
단순 네트워크 관리 프로토콜(SNMP)에서는 메니저가 송신한 polling 에 대해 에이전트가 이에 응답을 하는 방식으로 네트워크 관리 정보를 송수신하기 때문에 비교적 많은 네트워크 트래픽이 발생한다. 특히, 심화분석을 위해 특정 관리항목에 대하여 장시간 일정 주기로 반복적인 정보 수집이 요구되는 경향분석을 실시할 때 발생되는 관리용 트래픽은 네트워크 부하를 가중시킨다.

본 논문에서는 기존 SNMP 의 정보 수집 및 통신 방식을 개선하여 경향분석 용 정보 수집 시 메니저와 에이전트간 불필요한 네트워크 관리용 트래픽 발생을 최소화할 수 있는 효율적인 방안을 제시하고 구현하였다. 시험 분석 결과 기존 방식과 호환성을 유지하면서 네트워크 관리용 트래픽이 크게 감소하였다.

1. 서론

네트워크 관리를 위해서는 다양한 분석 항목들을 사용하는데 그 중 심화분석 항목들은 실시간 분석과는 달리 특정 기간 동안의 네트워크 상태를 알아보기 위하여 특정 객체에 대하여 일정기간의 이력정보, 통계정보 등을 수집하여 경향을 감시해야 한다[1][2][3]. 이 경우 기존 SNMP 에서는 (그림 1)과 같이 메니저가 해당 MIB 객체에 대하여 일정 주기로 “GetRequest” PDU 를 에이전트에 반복하여 Polling 하고 에이전트로부터 “GetResponse” PDU 들을 반복 수신한다. 이와 같은 반복적인 요청과 응답으로 인하여 네트워크 트래픽의 증가, 메니저 시스템의 부하 가중, 이에 따른 응답시간 지연 등의 문제를 일으킨다[4][5][6][7].

본 논문에서는 위와 같이 네트워크 관리를 위해 일정기간 동안 반복적인 측정이 요구되는 경향분석 용 정보 수집 시 발생하는 문제를 효율적으로 처리하여 네트워크 트래픽을 감소시킬 수 있는 개선방법에 대하여 제안하고, Prototype 을 구현하여 시험한 후 그 결과를 분석하고자 한다.

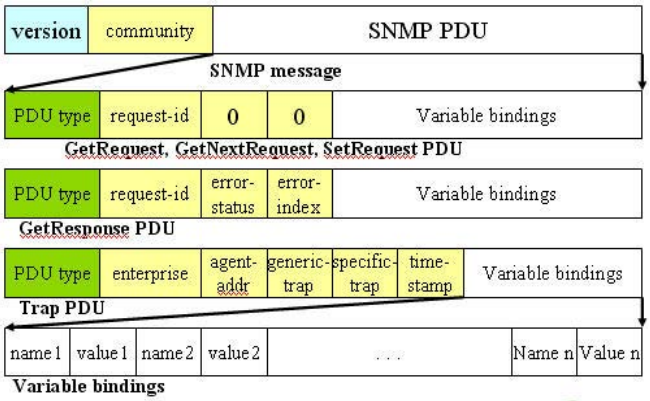


(그림 1) 경향분석 용 정보수집 시 기존 SNMP 의 PDU 송수신 방법

2. 기존 SNMP PDU 및 개선 모델

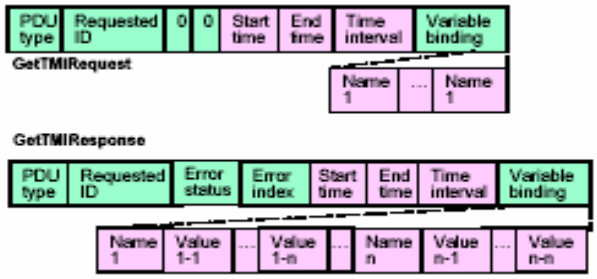
기존 SNMP 에서 네트워크 관리를 위한 경향분석 용 정보를 수집하기 위해 (그림 2)와 같은 PDU 들을 (그림 1)과 같이 송수신 하기 때문에 1 절에서 언급한 문

제가 발생한다[4][5][6][7].

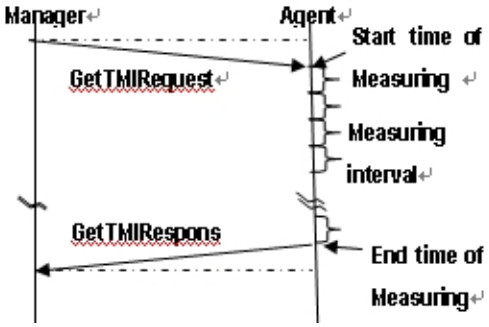


(그림 2) 기존 SNMP PDU Format

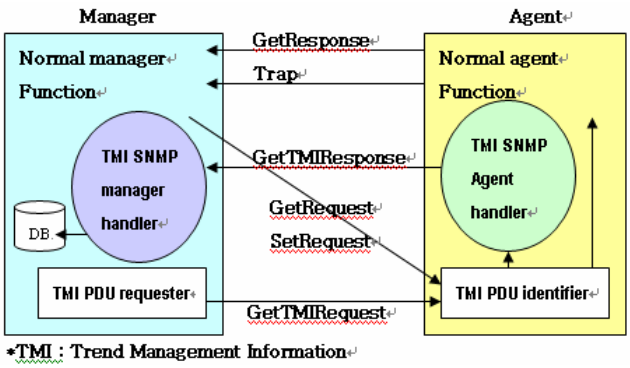
본 논문에서는 이러한 문제점을 효과적으로 해결하기 위해 기존 SNMP PDU 이외에 시각 정보를 포함한 (그림 3)과 같은 PDU 들을 추가하고, 메니저와 에이전트 간의 데이터 송수신 과정을 (그림 4)와 같이 개선한 (그림 5) 형태의 모델을 제안한다.



(그림 3) 시간정보를 추가한 SNMP PDU



(그림 4) 개선된 PDU 송수신 방법



(그림 5) 개선된 SNMP 모델

2.1 추가된 SNMP PDU의 구성 및 기능

추가된 PDU 들은 통상 관리자가 네트워크 관리 시스템을 이용하여 네트워크의 상태를 감시하고자 할 때 경향분석 용 정보를 선택하면 사용될 수 있도록 하고, 추가된 GetTMIRequest 와 GetTMIResponse PDU 내의 항목들 중 기존 SNMP PUD 들과 비교하여 새로이 포함된 항목들을 살펴보면, 시작시간(Start Time), 종료시간(End Time), 시간간격(Time Interval) 그리고 Variable binding 이 있다. 이 4 가지 항목들이 경향분석 용 정보들을 효과적으로 수집하기 위한 시각 정보들을 포함하고 있으며, 각 항목들의 의미는 다음과 같다.

- Start Time: 요구한 MIB 변수에 대한 자료수집 시작시간을 나타낸다. Octet string 형으로 표시(예: 12:30)
- End Time: 요구한 MIB 변수에 대한 자료수집 종료시간을 나타낸다. Start Time 과 마찬가지로 Octet string 형으로 표시.
- Time Interval: 요구한 MIB 변수에 대한 자료수집 주기를 나타낸다. Integer 형으로 표시되며, 값은 초(sec) 단위를 지정하게 된다. 이 값의 간격으로 SNMP 에이전트에서 해당 변수의 MIB 값을 수집하여 저장한다.
- Variable binding for GetTMIRequest: name 1, name 2…… name n 형태로 여러 개의 객체를 동시에 요청할 수 있도록 변수명 표기.
- Variable binding for GetTMIResponse: GetTMIRequest PDU 에서 요청한 변수 name 1, name 2, …… name n 각각에 대하여 자료 수집 요청 시간 동안 주기에 따라 수집된 값을 변수 명과 함께 표기.

2.2 개선 SNMP 모델의 동작 방법

(그림 5)와 같이 개선된 SNMP 모델의 PDU 송수신 수순을 살펴보면 메니저는 관리자가 입력한 자료 수집시작 시간(Start Time), 종료시간(End Time), 수집간격(Time Interval), Variable binding 필드를 이용하여 GetTMIRequest PDU 를 구성하여 에이전트에게 보낸다.

이를 수신한 에이전트는 TMI PDU Identifier 모듈에 전달하고, TMI PDU Identifier 모듈은 PDU 형식을 분석한다. PDU 형식이 기존 SNMP PDU 인 경우는 이를 Normal Agent 모듈로 전달하여 기존 SNMP 에서와 같이 처리 되도록 하고, 경향분석 용 정보일 경우는 이를 TMI SNMP Agent handler 에게 전달하게 된다.

TMI SNMP Agent handler 는 수신한 GetTMIRequest PDU 의 header 로부터 request-id, 자료수집 시작시간, 자료수집 종료시간, 자료수집 주기, variable-bindings 필드를 분리하여 필드 정보와 IP 패킷 header 에서 추출한 메니저 시스템의 IP 주소를 이용하여 지정된 시간 간격으로 data 를 종료시각까지 수집하여 GetTMIRequest PDU 를 생성하여 메니저에게 보낸다.

메니저는 에이전트의 TMI SNMP Agent Handler 모듈로부터 수신한 GetTMIResponse PDU header 에서 PDU 형식을 분석하여, 기존 SNMP 인 경우는 Normal manager 모듈로 전달하여 기존 SNMP 방법과 같이 처리되도록 하고, 경향분석 용 GetTMIResponse PDU 일 경우는 TMI SNMP manager handler 에게 전달하여 처리하게 한다.

TMI SNMP manager handler 는 GetTMIResponse PDU header 에서 PDU type, request ID, 자료수집 시작시간, 종료시간, 수집주기, variable-bindings 을 추출하여 해당 객체의 경향분석 용 정보를 관리시스템의 데이터베이스에 저장한다.

상기와 같이 경향분석 용 정보를 메니저와 에이전트간에 1 회씩 만 송수신함으로써 반복적인 polling 과 응답으로 인해 발생되는 불필요한 네트워크 트래픽의 최소화, 시스템의 부하감소 등의 효과를 얻을 수 있다.

3. Prototype 구현 및 테스트 베드 구성

제안한 SNMP 모델은 관리자와 에이전트로 구분되며 프로그램은 SNMP v2 를 지원하는 UCD SNMP v3.4 의 Source 모듈들을 C 언어로 수정하고 추가하였다.

구현된 SNMP 소프트웨어 모듈 구성은 크게 관리자와 에이전트 모듈로 구분되며, 관리자 시스템에는 인터넷으로 접속하여 실험할 수 있도록 관리자 시스템에 아파치 웹 서버를 구축하고, 관리자 기동용 관리화면을 Shell 기반 CGI 를 이용하여 기존 SNMP 기능과 개선된 기능을 수용할 수 있도록 각 모듈을 분리하여 작성하였다. 에이전트에는 수신된 SNMP 패킷을 구분하는 모듈과 구분된 PDU 형식에 따라 이를 처리하는 SNMP Handler 모듈들을 분리하여 구성하였다.

제안한 SNMP 모델과 기존 SNMP 모델에서의 네트워크 트래픽을 비교 시험하기 위하여 (그림 6)과 같은 테스트 베드를 구축하였다. 테스트 베드 시스템의 구성 환경은 다음과 같다.

- 각 시스템의 운영 체제:
  - Manager: UNIX (Solaris2.8)
  - Agent: Linux (Redhat6.2)
  - 측정시스템: Windows XP
- 사용된 SNMP version: SNMP version 2
- 사용된 SNMP 프로그램: UCD SNMP v3.4
- 구현에 사용된 언어: C, Shell 기반 CGI
- C 컴파일러: GCC v3.1
- 측정 툴: EtherPeek, AppDancer FA™

구현한 Prototype 모듈들은 해당 시스템에 설치하여 메니저와 에이전트 간에는 SNMP 에 의하여 정보를 송수신하고, 트래픽 측정 시스템에서는 웹을 이용하여 메니저 시스템을 기동시키며, 네트워크 관리 툴들을 이용하여 네트워크 트래픽을 측정할 수 있도록 하였다.

(그림 6) 제안 모델의 테스트 베드 구성도

테스트 베드를 이용한 시험 조건은 기존 SNMP 와 개선된 SNMP 각 경우에 대하여 메니저에서 MIB 객체 "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 5 초 주기로 30 분간 전송하여 측정된 네트워크 트래픽을 비교하였다.

4. 시험 결과 및 분석

4.1 기존 SNMP 에 의한 네트워크 트래픽

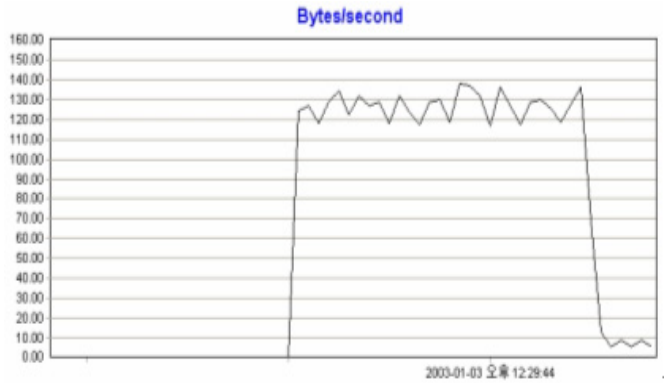
기존 SNMP 에서 메니저와 에이전트 간의 네트워크 트래픽 발생 과정은 (그림 1)과 같으며, 메니저가 발생시킨 GetRequest PDU 에 의한 트래픽을 Tm 라 하고 이에 대한 에이전트의 응답인 GetResponse PDU 에 의한 트래픽을 Ta 라 하면, 한 개의 정보 수집을 위해 발생되는 트래픽(To)은 식 1)과 같으며, 정보 수집기간이 D 이며 수집 주기가 I 일 경우 발생되는 총 트래픽(Tt)은 식 2)와 같다.

To = Tm + Ta -------- 식 1)

Tt = D/I * To -------- 식 2)

식 2)에서 알 수 있듯이 정보 수집기간이 길고 수집 주기가 짧을수록 발생되는 트래픽이 증가한다.

시험조건에 따라 테스트 베드를 이용하여 측정된 기존 SNMP 의 트래픽은 (그림 7)과 같으며, 평균 약 127 Bytes/초의 트래픽이 발생되었다. 트래픽이 일정하지 않은 이유는 구축된 테스트 베드에 SNMP 이외에도 측정 시스템에서 웹을 이용하여 매니저 기동 시 발생되는 HTTP 와 Windows Xp 에서 발생되는 NetBIOS 등의 프로토콜들이 추가되었기 때문이다.



(그림 7) 기존 SNMP 에서 측정된 네트워크 트래픽

4.2 개선된 SNMP에 의한 네트워크 트래픽

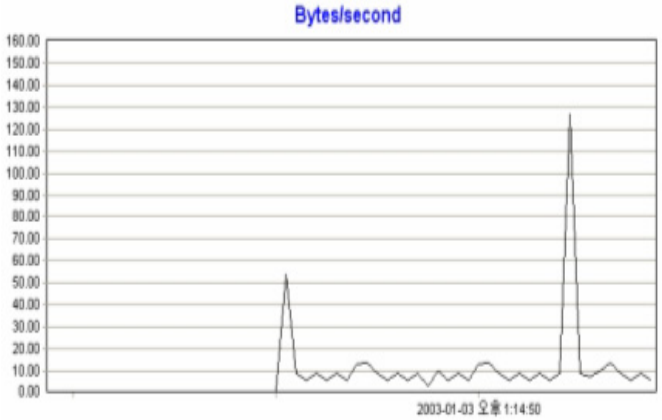
경향분석 용 정보 수집을 위해 개선된 SNMP 에서의 메니저와 에이전트 간 네트워크 트래픽 발생 과정은 (그림 4)와 같다. 여기서 메니저가 발생시킨 GetTMIRequest PDU 에 의한 트래픽을 Tmi 라 하고, 이에 대한 에이전트의 응답인 GetTMIResponse PDU 에 의한 트래픽을 Tai 라 하면, 한 개의 정보수집 시 발생되는 트래픽(Toi)은 식 3)과 같으며, 정보 수집기간이 D 이며 수집간격이 I 일 경우의 발생되는 총 트래픽(Tti)은 식 4)와 같이 표현할 수 있다.

Toi = Tmi + Tai -------- 식 3)

Tti = Tmi + Tai = Toi -------- 식 4)

식 3)과 4)에서 알 수 있듯이 정보 수집기간과 수집 주기와 상관없이 발생되는 트래픽이 동일하다.

기존 SNMP 와 동일한 시험조건에서 테스트 베드를 이용하여 측정된 개선된 SNMP 에서의 트래픽은 (그림 8)과 같이 평균 약 21bytes/초의 트래픽이 발생되었으며, 순수 SNMP 트래픽은 메니저가 에이전트에 GetTMIRequest PDU 를 전송하는 초반과 정보수집이 완료되어 에이전트가 GetTMIResponse PDU 를 메니저에 전송하는 마지막 부분에서 발생되었다.



(그림 8) 개선된 SNMP 에서 측정된 네트워크 트래픽

4.3 비교분석

기존 SNMP 와 개선된 SNMP 에서 발생된 네트워크 트래픽을 비교해 보면 단일전송의 경우 (그림 2)와 (그림 3)의 PDU 구성에서 알 수 있듯이 GetTMIRequest 와 GetTMIResponse 가 GetRequest 와 GetResponse 보다 크기 때문에 Tmi≥Tm, Tai≥Ta 의 관계가 성립되어 식 1)과 3)에서 Toi≥To 관계가 된다. 이들 간의 차이는 요청하는 변수에 따라 약간 차이가 발생하지만 시험기준에 의하면 기존 SNMP 에서 21bits (108-87), 61bits (152-91)씩 적게 발생하였다.

반면에 본 논문에서 중점을 둔 경향분석 용 정보의 경우 식 2)와 4), (그림 7)과 (그림 8)에서 알 수 있듯이 전체 트래픽은 개선된 SNMP 에서 6 배 감소하였고, 순수 SNMP 에 의해 발생된 트래픽 만을 비교해 보면 기존 SNMP 는 식 1)과 식 2)에 의해 64,080bytes ((87B+91B)*60/5 초*30 분)가 발생되었고, 개선된 SNMP 의 경우에는 식 3)과 식 4)에 의하면 1,622 bytes (108B+1,514B)가 발생하여 개선된 SNMP 의 네트워크 트래픽이 약 39.5 배 감소하였다.

결론적으로, 기존 SNMP 의 경우 데이터 송수신 횟수에 정비례하여 네트워크 트래픽이 증가하는 반면, 개선된 SNMP 의 경우는 측정횟수에 따라 다소 증가하나 큰 차이가 없기 때문에 측정기간이 길고 측정 주기가 짧아질수록 개선된 SNMP 의 네트워크 트래픽이 상대적으로 크게 감소하는 효과를 나타낸다

5. 결론

시험 결과를 바탕으로 제안한 SNMP 의 특징을 정리해보면 첫째, 경향분석 용 정보 수집 시 기존 SNMP 에 비해 적은 횟수의 요구와 응답 메시지를 송수신함으로써 네트워크 트래픽를 크게 줄일 수 있다.

둘째, 기존의 SNMP 와 호환성을 유지한다. 개선된 SNMP 모델은 기존 SNMP 의 기능과 완전하게 호환성을 유지하면서 네트워크 성능관리에 필수적인 경향분석 용 정보 수집 시 트래픽 감소 특성이 있다.

셋째, 네트워크 관리 시스템이 관리해야 할 대상 장비가 다수인 대형 네트워크 경우 네트워크 관리 시스템의 부하를 경감시킬 수 있다.

참고문헌

[1] 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.

[2] Sang-chul Shin, Seong-jin Ahn, jin-Wook Chung, “A new approach to gather network management data periodically”, ITC-CSCC ‘97, 1997.

[3] 유승근, 안성진, 정진욱, “SNMP MIB-II 를 이용한 인터넷 관리 시스템의 웹 인터페이스 설계 및 구현”, 한국정보처리학회논문지, 제 6 권 제 3 호, pp. 699-709, 1999.

[4] Mani Subramanian, “Network Management: Principles and Practice”, Pearson Addison Wesley, 2000.

[5] William Stallings, “SNMP, SNMPv2, SNMPv3, and RMON1 and 2, Third edition”, Pearson Addison Wesley, 1996.

[6] 천진영, 정진하, 윤완오, 최상방, “SNMP 기반 네트워크 관리를 위한 적응형 네트워크 모니터링 방법”, 한국통신학회논문지, 제 27 권 제 12C 호, pp. 1265-1275, 2002.

[7] 김민우, 박승균, 오영환, “SNMP 트래픽 최적화를 위한 폴링 방식에 관한 연구”, 한국통신학회논문지, 제 26 권 제 6A 호, pp. 1051-1058, 2001.