실시간 네트워크 관리를 위한 SNMP의 확장에 관한 연구

김 동 수† · 정 태 명††

요 약

실시간 시스템에서 정확한 응답 시간과 통신상의 지연이 최소화 될 것이 요구된다. 그러나, 현재의 SNMP 프로토콜은 주기적인 자료 요청을 하는데 많은 통신량을 유발하며, 네트워크 자원을 비효율적으로 사용한다. 본 논문은 현재 사용중인 SNMP에 시간적 제약 조건을 수용할 수 있도록 확장한 실시간 SNMP의 구현에 대한 논문이다. 실시간 SNMP는 기존에 사용하던 SNMP가 주기적 요구(periodic request)를 수행할 때 발생하는 통신상의 비효율성을 개선하였다. 이를 위해서는, 기존의 SNMP의 서버/클라이언트 방식의 기법에 변화를 주지 않고, 관리국(Management Station)이 Real-Time SNMP PDU를 에이전트에게 전송하여, 실시간 SNMP 에이전트 측에서 관리국에서 요청한 MIB에 대한 정보를 주기적으로 관리국으로 정보를 제공할 수 있게 하였다. 이러한 설계를 바탕으로 관리국과 에이전트 사이의 요구에 대한 정확한 응답 시간(response time)을 제공할 수 있게 하여, 네트워크 관리의 효율을 높일 수 있는 방법을 본 논문에서는 제시하고, 실제 구현 환경에서의 실험을 통하여 성능이 개선됨을 보였다.

The Extension of SNMP for Real-Time Network Management

Dong-Soo Kim† · Tai-Myoung Chung††

ABSTRACT

In real-time systems, the accurate response time and minimal communication delay is highly desired. However, a conventional SNMP uses network resources inefficiently and introduces too much network traffic for periodic requests in particular. This paper presents an extended SNMP called Real-Time SNMP. It is designed to support real-time applications with time constraints to provide more accurate response time and less network traffic for periodic SNMP requests. While keeping current Client/Server implementation unchanged, we simply add a periodic SNMP request PDU which is sent from management station to agents. In addition, the module of the Real-Time SNMP agent works between management station and each SNMP agent to periodically generate requests for the associated agent. In this paper we have implemented the proposed Real-Time SNMP agent module and extended SNMP PDU. We also show the experimental results that indicate more punctual response time and reduced communication delay using the proposed Real-Time SNMP.

1. 서 론

최근 많은 응용 프로그램들이 네트워크를 통해 데이터를 교환하고 있으며, 이러한 네트워크에서의 데이터 교환은 통신량의 증가를 가져 왔다. 이러한 환경에서도 네트워크 상에서 데이터의 정확성과 빠른 정보 제공은 사용자들에게 투명성을 보장해 주어야 하는데, 이것을 수행하기 위한 네트워크 관리 프로토콜의 중요성은 더욱 강조되어지고 있으며, 그 방법들 역시 날로 다양해지고 있다. OSI 네트워크 관리 표준으로 CMIP

※ 본 논문은 1997년도 한국학술진흥재단의 공모과제 연구비에 의해 연구되었음.
† 준 회 원 : 성균관대학교 전기전자 및 컴퓨터공학부
†† 종신회원 : 성균관대학교 전기전자 및 컴퓨터공학부 교수
논문접수 : 1998년 10월 19일, 심사완료 : 1999년 2월 2일

과 CMIS, IETF의 SNMP 등, 여러 네트워크 관리 프로토콜들이 있으며, 가장 많이 사용되고 있는 것이 TCP/IP 네트워크 관리의 표준인 SNMP로서 간단하고 확장성이 좋으며, 현재 많은 네트워크 장비 제조업체들이 지원하고 있다는 장점을 가지고 있다[8]. SNMP의 운영 방법으로는 관리자가 네트워크의 현재 상태에 관한 정보를 요구하여 그 요구에 의해 에이전트가 수집한 자료를 분석해서 네트워크를 관리하는 형태를 갖는다[7]. 이러한 패러다임에서, 주기적인 네트워크의 상태를 확인하고자하는 경우 기존의 SNMP를 사용하면 일정 시간 간격마다 관리국이 에이전트를 폴링(polling)하여 원하는 정보를 얻어야만 한다. 하지만, 이러한 방법은 네트워크 상에서 폴링으로 인한 통신량 증가를 발생시키게 되고 때로는 폴링을 위한 요구 횟수가 많아져, 정보 전송 시간이 지연되는 경우도 발생된다. 따라서, 실제 요구된 시간에 에이전트로부터 정보를 얻지 못하는 단점이 존재한다. 이러한 단점을 보완하기 위해서 본 논문에서는 실시간 SNMP를 제안하고 구현하였다. 실시간 SNMP는 기존의 SNMP에 시간적 제약 조건을 표현할 수 있도록 확장한 프로토콜을 말하는데, 기존의 SNMP와 비교해 볼 때, 다음과 같은 장점을 얻을 수 있다.

첫째, 기존의 SNMP에 비해 적은 요구 메시지를 보내므로 네트워크의 통신량을 줄일 수 있다. 즉, 기존의 SNMP에서와 같이 주기적 요구를 함으로써 규칙적인 시간 간격으로 MIB 데이터에 대한 정보를 매 주기마다 에이전트에게 요구하기보다는, 한 번의 요구를 통해서 특정 시간에 MIB 정보를 알아낼 수 있는 기능을 제공한다[2].

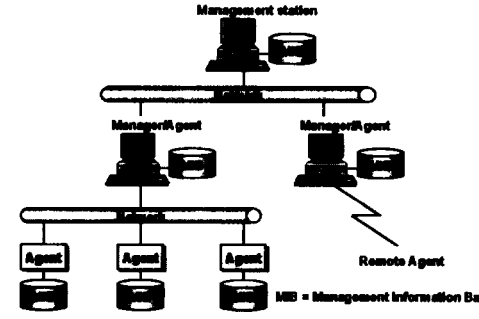
둘째, 데이터 수집 시간의 오차를 줄일 수 있다. 기존의 SNMP에서는 폴링의 결과가 네트워크의 통신시간만큼 지연되는 영향을 받게 되지만, 실시간 SNMP에서는 에이전트 내에서 요구가 이루어지므로 그만큼 정확한 시간의 정보를 수집할 수 있게 된다는 장점이 있다.

셋째, 네트워크 상에서 자료 손실에 대한 확률을 감소시킨다. 특히, 원거리 통신을 통해 네트워크 관리가 이루어지는 경우, 공중망이나 통신량이 많은 구간에서 유실되는 패킷으로 인해 요구 메시지나 응답 메시지가 손실되는 경우가 많다. 그러나, 실시간 SNMP에서는 한 번의 요구로 주기적인 응답의 수신을 가능하게 하므로 요구 메시지의 손실에 의한 자료 손실을 감소시키는 효과를 가져온다.

본 논문에서는 많은 주기적 MIB 정보가 필요한 경우에 대한 효율적인 실시간 SNMP에 대한 모델을 제시하고 구현 및 실험을 통하여 그 효율성을 살펴보도록 한다. 2장에서는 기존의 SNMP에 대한 간략한 고찰을 하고, 3장에서는 실시간 SNMP의 개념과 모델, 동작 원리를 설명하고, 4장에서는 실험을 통해 일반 SNMP와 비교하여 그 결과로 얻어진 장점 및 특성을 검증하도록 한다. 그리고, 마지막으로 결론과 향후 계획을 제시하도록 하겠다.

2. Overview of SNMP

네트워크 관리의 기능에는 네트워크 상에서 잘못된 부분을 찾아 제거하고 복구하는 관리를 수행하는 장애관리(Fault Management)와 시스템의 계정과 이용 과금을 관리하는 계정 관리(Accounting Management), 네트워크 상의 작업을 수행하기 위한 네트워크 환경을 관리하는 구성 관리(Configuration Management), 성능이 갑자기 떨어지지 않고 일정 이상의 성능을 유지시키는 성능 관리(Performance Management), 네트워크의 보안을 유지, 보수하는 보안 관리(Security Management)의 다섯 가지의 기능이 제공되어야 한다. 특히, 네트워크 관리의 기능을 제공하기 위해서는 (그림 1)에서와 같이 관리 시스템을 구성하는 네 가지 요소가 있다[5,8].



(그림 1) 네트워크 관리 기능 구조
(Fig. 1) Network management infra-structure

① Management Station(관리국)
② Management Agent(에이전트)
③ Management Information Base(MIB)
④ Network-management protocol

네트워크 관리 요소 중에서 네트워크 관리 프로토

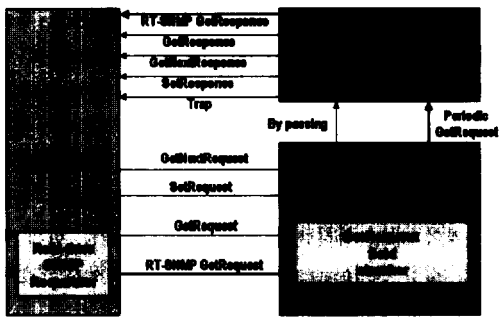
콜은 네트워크 상의 관리국과 에이전트사이에 어떻게 정보를 주고받을 것인지를 표준으로 정한 것이다. 이러한 프로토콜 중 하나가 SNMP이다. SNMP는 GET, SET, TRAP 메시지를 이용하여 관리정보를 요구하고 응답하여 네트워크 관리를 수행한다[12].

3. 실시간 SNMP

SNMP는 일반적으로 널리 알려져 있고 현재 많이 사용되고 있다. 그리고, 대부분의 업체들이 생산하는 네트워크 장비들이 장비의 관리를 위해 지원하고 있는 프로토콜이기도 하다. 이러한 배경으로 효과적인 네트워크 관리를 위해서는 SNMP의 기능 확장을 통해 기존의 SNMP의 장점들을 수용하면서 효율성 향상을 시도하는 형태가 되어야 한다. 따라서, 실시간 SNMP는, SNMP가 가지고 있는 동작들을 모두 수행하면서 주기적인 시간개념을 추가하여 최소의 비용을 가지고 좀더 효율적인 네트워크 관리를 수행하는 것이다. 그러므로, 실시간 SNMP는 효과적인 주기적 요구를 구현함으로써 네트워크 관리에 좀더 나은 효율성과 정확성을 가져올 수 있다.

3.1 배경

관리국이 특정한 MIB 정보에 대한 주기적이고 계속적인 감시(tight monitoring)가 필요하다면, 관리국은 보다 짧은 시간 간격을 가지고 필요한 MIB의 자료를 주기적으로 요청하게 된다. 같은 자료에 대한 반복되는 요구와 응답 시간까지 걸리는 시간을 절약하려는 경우 네트워크 관리자에게 네트워크 관리에 대한 효율적인 방법을 제시할 필요가 있는데, 이러한 요구를 만족시키기 위해서는 실시간 SNMP의 도입이 필요하다.



(그림 2) 실시간 SNMP 모델
(Fig. 2) A real-time NMS model using SNMP

3.2 실시간 SNMP 모델

기존의 SNMP 프로토콜은 관리자가 에이전트에게 GetRequest, SetRequest, GetNextRequest 등의 PDU를 보내면 에이전트가 이를 받아서 이와 연관되는 결과를 GetResponse PDU를 이용하여 관리자에게 돌려주는 형식으로 관리 상태 정보를 수집한다[1,4]. (그림 2)에서는 각각의 응답 메시지를 구분하기 위해 다른 응답 PDU명을 사용하였으나 실제로, 모든 에이전트의 응답은 GetResponse PDU를 사용한다. 또 관리자의 요구 없이 에이전트가 관리자에게 정보 전달을 필요로 하는 경우, 에이전트는 Trap을 발생시킴으로 장애나 오류 등의 필요한 정보를 관리자에게 전달한다. 여기에, 실시간 SNMP GetRequest PDU를 정의하고 이를 처리할 수 있는 요소들을 구성하여 실시간 SNMP의 기능을 구현하게 된다. 실시간 SNMP의 동작을 위한 각 부분의 세부 역할을 보면 다음과 같다.

① 실시간 SNMP Requester
사용자가 지정한 MIB 변수에 대한 자료 수집 시작 시간, 자료 수집 종료 시간, 수집 주기를 이용하여 실시간 SNMP PDU를 생성하고 이를 실시간 SNMP 에이전트에 전송하는 기능을 수행한다.

② GetRequest field checker
관리국으로부터 전달된 GetRequest PDU의 각 field를 분석, 기존의 SNMP 메시지인지 실시간 SNMP 메시지인지를 판단한다. 그리고 기존의 SNMP 메시지인 경우는 이 메시지를 그대로 Normal Requester에게 전달하여 일반 SNMP의 기능을 수행하도록 하고, 실시간 SNMP의 경우는 메시지를 RT invoker에게 전달하는 기능을 수행한다.

③ Normal requester
기존의 일반적인 SNMP 메시지를 처리하는 기능을 담당하며, 수신된 메시지를 그대로 일반 SNMP 에이전트에게 전달하는 기능을 수행한다.

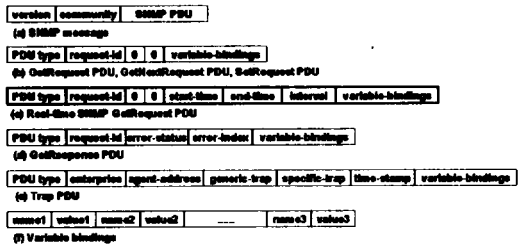
④ RT(Real-Time) invoker
RT invoker는 실시간 SNMP의 기능 수행 요소들 중 가장 핵심적인 부분이다. 즉, 주기적인 요구가 바로 이 요소를 통해 이루어진다. RT invoker는 실시간 SNMP GetRequest PDU를 처리하는 기능을 갖는 요소이다. 이 요소는 수신된 실시간 SNMP GetRequest PDU에서 헤더를 분리하여 자료 수집 시작 시간, 자료 수집 종료 시간, 수집 간격을 추출

하고, 이를 바탕으로 실시간 SNMP의 GetRequest 기능을 수행한다. 즉, 수집 시작 시간이 되면 이 모듈은 추출한 수집 간격에 따라 주기적으로 일반 SNMP 에이전트에게 GetRequest PDU를 전달하게 되며, 수집 종료 시간이 되면 수행을 중단한다.

결과적으로, 실시간 SNMP 에이전트는 기존의 SNMP port(UDP port 161)를 사용하여 관리국이 보내는 메시지들을 수신하게 된다. 실시간 SNMP 에이전트는 수신된 GetRequest PDU가 일반 SNMP의 GetRequest PDU인지 실시간 SNMP의 GetRequest PDU인지를 판단하여 각 PDU에 맞는 모듈에 전달하여 처리를 달리 하게 된다. 이러한 각 모듈의 기능에 의해 실시간 SNMP에서는 관리자가 기존의 SNMP의 명령들을 그대로 사용할 수 있으면서 실시간 SNMP의 기능을 수행할 수 있다.

3.3 실시간 SNMP PDU

실시간 SNMP GetRequest PDU((그림 3)의 (c) Real-Time SNMP GetRequest PDU)의 내용을 살펴보면 시작시간(Start Time), 종료시간(End Time) 그리고 간격(Interval)이라는 필드가 있다. 이 세 가지 필드들이 주기적인 요구를 하기 위한 정보들을 포함한다. 각 필드들의 내용은 다음과 같다.



(그림 3) SNMPv1 PDU와 실시간 SNMP PDU
(Fig. 3) SNMPv1 PDU vs. Real-Time SNMP PDU

① Start-Time : 요구하는 MIB 변수에 대한 자료 수집 시작 시간을 나타낸다. octetstring 형으로 표시된다.(예를 들면, '12:30'의 형태)

② End-Time : 요구하는 MIB 변수에 대한 자료 수집 종료 시간을 나타낸다. Start-Time과 마찬가지로 octetstring 형으로 표시된다.

③ Interval : 요구하는 MIB 변수에 대한 자료 수집 주기를 나타낸다. integer 형으로 표시되며, 값은 초(sec) 단위를 지정하게 된다. 이 값의 간격으로 GetRequest PDU가 SNMP 에이전트에게 전달된다.

3.4 실시간 SNMP의 동작

관리국이 실시간 SNMP GetRequest PDU를 에이전트로 전송함으로써 에이전트로부터 실시간 SNMP의 주기적 요구를 하게된다. 관리국은 request-id, 자료 수집 시작시간, 자료 수집 종료시간, 수집 간격, variable-bindings의 내용을 갖고 있는 실시간 SNMP GetRequest PDU를 구성하여 실시간 SNMP 에이전트에게 보낸다.

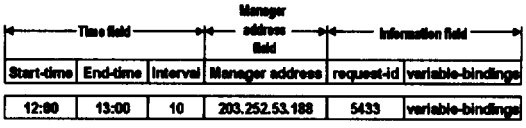
실시간 SNMP 에이전트는 수신한 GetRequest PDU가 일반 SNMP의 GetRequest PDU인지 실시간 SNMP GetRequest PDU인지를 판단하기 위해 이 PDU를 Get-Request field checker로 전달한다. 이는 실시간 SNMP GetRequest PDU가 일반 SNMP의 GetRequest PDU와 같이 PDU type값(0)을 가지므로 이를 구분해 주는 모듈이 필요하기 때문이다.

GetRequest field checker는 수신된 GetRequest PDU의 각 field를 분석, 일반 SNMP의 경우는 이를 그대로 Normal requester로 전달하여 일반 SNMP의 GetRequest PDU와 같이 처리 되도록 하고, 실시간 SNMP의 GetRequest PDU일 경우는 이를 RT invoker에게 전달하게 된다.

RT invoker는 request-id, 자료 수집 시작시간, 자료 수집 종료시간, 자료 수집 주기 필드를 분리하고 각 필드들의 정보와 실시간 SNMP 에이전트의 IP 패킷 수신 부분에서 추출한 매니저의 주소 그리고, PDU내의 variable-bindings의 정보를 이용하여 스케줄 표를 작성한다. 이 스케줄 표의 자료를 근거로 하여 에이전트에게 주어진 시간동안 지정한 간격으로 GetRequest PDU를 보낸다. 즉, 이 정보를 이용하여 RT invoker는 주어진 시간동안 주어진 간격으로 Source 주소값으로 관리국의 주소를 갖고, Destination 주소값을 localhost로 갖는 IP 패킷을 생성하여 GetRequest 신호를 일반 SNMP 에이전트에게 전달하게 된다. 또한, 실시간 SNMP 에이전트는 기존의 SNMP 포트(UDP 포트 161번)를 청취하고 일반 SNMP 에이전트는 다른 포트를 청취하는 형태를 가져야 한다. 이러한 방식을 취하는 이유는, 마치 매니저가 직접 GetRequest PDU를 전송한 것처럼 구성함으로써 기존의 어떠한 SNMP 에이전트라도 수정 없이 실시간 SNMP 에이전트와 연동이 가능하도록 하기 위함이다. 실제로, 이러한 동작은 다른 요구 메시지의 경

우도 마찬가지로 수행된다. Schedule Table의 각 Record의 구성을 구체적으로 보면 다음과 같다.

- 시간 필드(Time field)
  주기적 요구에 대해 요구한 MIB 자료를 보내기 위한 표. 즉, 자료 수집 시작 시간, 자료 수집 종료 시간, 자료 수집 간격을 저장하고 있게 된다.
- 매니저 주소 필드(Manager address field)
  현재 Record내의 시간 정보에 근거하여 자료를 보내어 주어야 할 Manger의 IP 주소를 저장하게 된다.
- 자료 정보 필드(Information data field)
  관리국에서 GetRequest로 에이전트 측에 요구한 MIB 정보에 대한 내용(variable-bindings)과 request-id를 가지고 있다.



(그림 4) RT invoker의 스케줄 표의 항목과 내용
(Fig. 4) Records of RT invoker's schedule table

3.5 실시간 SNMP의 활용

SNMP에서 관리국이 에이전트(Router, Bridge, Host)가 지니고 있는 정보를 폴링 하여 주기적으로 네트워크를 관리한다. 이때 에이전트의 MIB으로부터 얻은 데이터를 가지고 에이전트가 동작하기 시작한 시간, IP address, TCP와 UDP에 대한 정보, 패킷의 error, 등 MIB으로부터 얻은 정보를 비교 분석함으로써 네트워크의 성능관리나 장애관리에 도움을 줄 수 있다[5,8].

실시간 SNMP는 원거리에 있는 에이전트에 대한 네트워크 관리(WAN의 경우)와 웹 서버 관리에도 효과적이다[6]. 원격지의 에이전트에 대한 관리국의 주기적인 요구 같은 경우에는 에이전트에 대한 정보를 요구하는 시간이 관리국으로부터 에이전트까지의 거리가 멀리 떨어져 있을수록, 정보를 요청한 후 그 요청에 대한 응답 시간 또한 지연되어 진다. 만약 이러한 관리국이 원거리에 있는 에이전트를 관리하고자 한다면 앞에서 언급했던 것과 마찬가지로 실시간 SNMP의 주기적 요구를 보다 효율적인 방법으로 구현함으로 보다 좋은 네트워크 관리 작업을 수행 할 수 있다.

4. 실험 결과 및 분석

4.1 분석 항목과 실험 환경

실시간 SNMP와 일반 SNMP의 성능 비교를 위한 실험 환경 및 조건은 다음과 같다.

① 각 호스트의 운영 체제 : Solaris 2.5.1
② 사용된 SNMP version : SNMP version 1
③ 사용된 SNMP toolkit : UCD SNMP v3.4
④ 구현에 사용된 언어 : C
⑤ C 컴파일러 : SunSoft C Compiler v4.0

위 환경 하에서 실시간 SNMP 에이전트를 구현하여 실험한 결과, 실시간 SNMP와 일반 SNMP의 성능 비교를 위한 항목들은 다음과 같이 정의해 볼 수 있다. 이 항목들은 두 프로토콜이 성능 상에서 뚜렷한 차이를 보인 항목들이다.

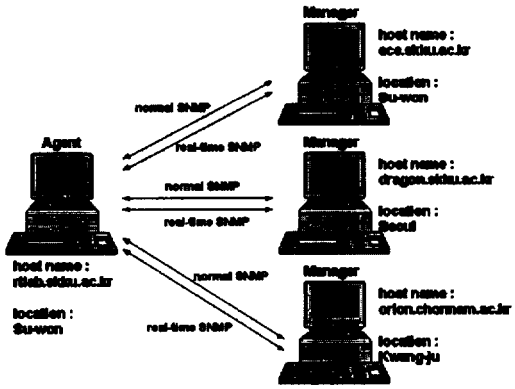
① 응답 시간
② 응답 시간 간격
③ 응답 시간의 변화 정도
④ 네트워크 통신량 점유율
⑤ 메시지의 손실율

이러한 인자들을 설정한 후에, 두 프로토콜의 성능 비교의 다양화를 위해 매니저 호스트를 원거리와 근거리, 그리고 로컬 네트워크상의 호스트의 세 가지 상황으로 나누어서 비교를 하였다. 즉, 에이전트 호스트의 위치는 고정되어 있으며, 각 매니저 호스트의 위치를 로컬 네트워크, 근거리, 원거리로 변화시켜 각각의 매니저 호스트에서 각 항목들의 값을 측정하였다.

① 에이전트 host
   rtlab.skku.ac.kr(203.252.53.165)
② 매니저 host
   원거리 : orion.chonnam.ac.kr(168.131.33.6)
   근거리 : dragon.skku.ac.kr(203.252.32.4)
   로컬 네트워크 : ece.skku.ac.kr(203.252.53.9)

물리적인 위치로 볼 때 로컬 네트워크 상의 관리자 호스트는 같은 대학교 도메인 내에서 거리 상으로 근접한 장소에 설치된 호스트이며, 근거리 호스트는 같은 대학교 도메인(skku.ac.kr)이지만 수원-서울의 거리에 위치한 두 호스트가 되고, 원거리 스테이션은 대한민국의 대학교(ac.kr)라는 같은 도메인을 갖는 수원-광

주의 거리에 위치한 두 호스트가 된다(그림 5). 물리적 거리는 네트워크 상에서 쌍방을 연결하는 물리적 매체의 길이와 라우팅 경로상의 중간 node)의 수를 결정하는 요인이 되며 결과적으로 패킷 전송 시간을 결정하는 주요한 요인이 된다. 본 실험에서 로컬 네트워크 상의 호스트까지의 중간 node의 수는 0이며, 근거리 호스트까지의 중간 node의 수는 3이며, 원거리 호스트까지의 중간 node의 수는 8이다.

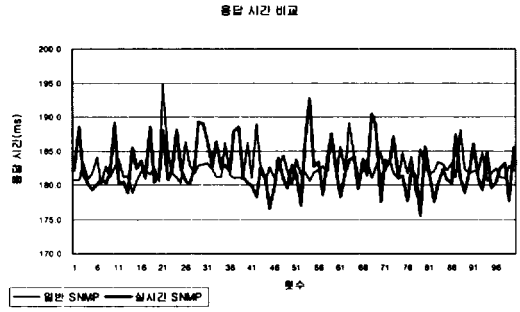


(그림 5) 실험에서 관리국과 에이전트의 관계
(Fig. 5) Relationship between manager and agent for the testbed

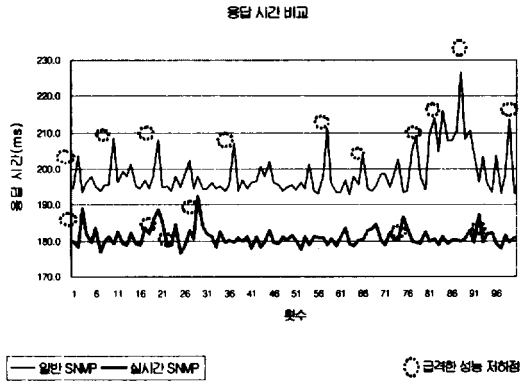
응답 시간과 응답 간격의 비교는 각 프로토콜마다 100회의 폴링을 실시하며 이것을 5회씩 실시한 평균적인 수치를 기록한 것이다.

4.2 응답 시간의 비교

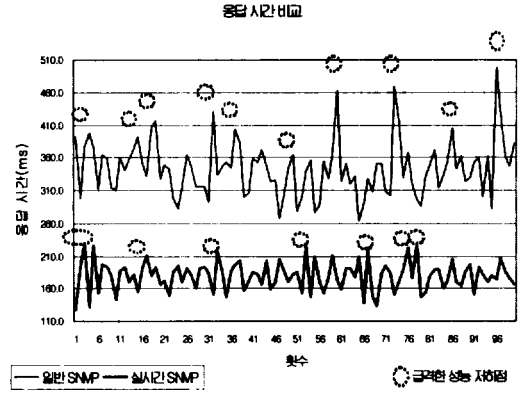
로컬 네트워크상의 호스트를 이용한 실험에서 두 프로토콜은 응답 시간이 평균적으로 거의 차이를 보이지 않았다(그림 6). 이는 두 호스트가 하나의 LAN 구역(segment)에 속하여 네트워크의 통신량이 그리 많지 않고, 매체의 길이에 의한 전송 지연이 상당히 미약한 것이 원인이다. 결론적으로, 로컬 네트워크 상의 호스트 사이에서 두 프로토콜의 성능 차이는 거의 없다고 말할 수 있다. 그리고, 근거리 호스트 사이에서 일반 SNMP와 실시간 SNMP의 평균 응답시간의 차이는 약 10~20ms의 차이를 보였고, 원거리 호스트 사이에서는 일반 SNMP와 실시간 SNMP의 평균 응답시간은 100~150ms의 차이를 보였다(그림 7, 8). 이는 일반 SNMP의 경우 요구 메시지 전송 지연과 응답 메시지 전송 지연 시간이 존재하는 반면, 실시간 SNMP의 경우는 응답 메시지 전송 지연 시간만이 존재하는데서 기인한다.



(그림 6) 로컬 네트워크 호스트의 응답 시간 결과 그래프
(Fig. 6) Response time in the same LAN segment



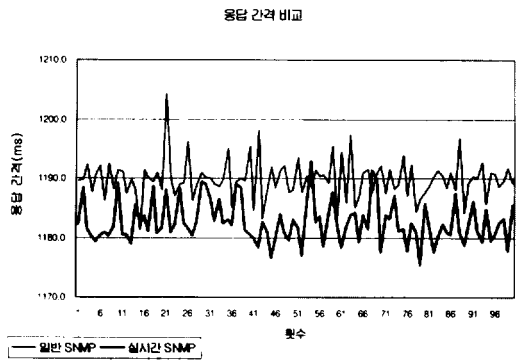
(그림 7) 근거리 호스트의 응답 시간 결과 그래프
(Fig. 7) Response time between two separated LANs



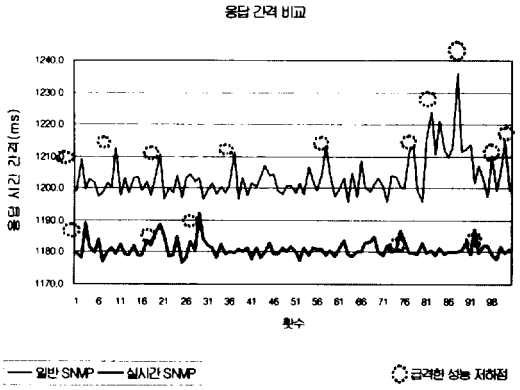
(그림 8) 원거리 호스트의 응답 시간 결과 그래프
(Fig. 8) Response time in WAN environment

4.3 응답 간격의 비교

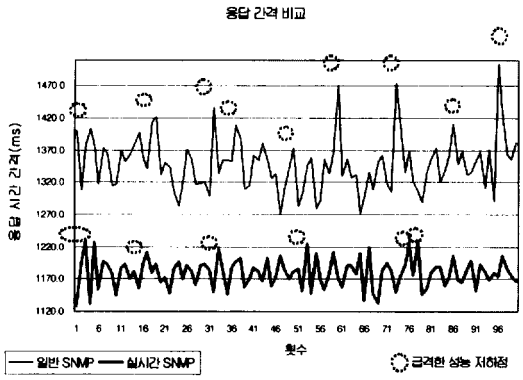
다음으로, 응답 간격의 정확성에 관한 것이다. 즉, 매니저가 요구하여 설정된 수집 주기에 응답 간격이 얼마나 정확한가에 대한 것이다. 이 항목에서도 역시 응답 지연 시간과 유사한 결과를 보였다(그림 9,10,11). 즉, 수집 요구 간격 시간에 응답 지연시간과 약간의 처리시간이 더해진 형태의 시간 결과를 보였다. 즉, 일반 SNMP의 경우는 원래 요구한 수집 간격 시간에 완전한 요구-응답에 걸리는 응답 지연 시간과 에이전트가 메시지를 처리하는 프로세스 시간이 더해진 반면, 실시간 SNMP의 경우에는 응답에 걸리는 전송 지연 시간과 에이전트가 메시지를 처리하는 프로세스 시간이 더해진 형태의 수집 간격 시간이 측정된 것으로, 다음과 같이 요약할 수 있다.



(그림 9) 로컬 네트워크 호스트의 응답 간격 결과 그래프
(Fig. 9) Response time interval in the same LAN segment



(그림 10) 근거리 호스트의 응답 간격 결과 그래프
(Fig. 10) Response time interval between two separated LANs



(그림 11) 원거리 호스트의 응답 간격 결과 그래프
(Fig. 11) Response time interval in WAN environment

① 일반 SNMP

$$I_{rp} = I_{get} + t_{rq} + t_{rp} + t_{mp}$$

② 실시간 SNMP

$$I_{rp} = I_{get} + t_{rp} + t_{mp}$$

I_{rp} : 응답 간격
I_{get} : 요구된 자료 수집 간격
t_{rq} : 요구 전송 지연시간
t_{rp} : 응답 전송 지연시간
t_{mp} : 메시지 처리 지연시간(Processing time)

결론적으로, 실시간 SNMP에 의해 수집되는 자료들이 원하는 시간, 간격에 근접하여 수집될 수 있음을 알 수 있다.

위에서 설명한 두 항목 즉, 응답 시간과 수집 간격 정확도는 응답 시간의 경우와 마찬가지로, 에이전트와 매니저간의 물리적 거리 - 전송 매체의 길이와 라우팅 경로의 홉(HOP)을 결정하는 - 가 멀어질수록 두 프로토콜간의 평균적인 차이는 더욱 커지는 결과를 보였다. 근거리 관리에 있어서 이러한 차이는 무시될 수 있을 정도로 미약하지만 원거리 네트워크 관리를 필요로 하는 곳에서 자료 수집의 정확성, 신속성을 요구하는 네트워크 관리 체계의 경우, 실시간 SNMP가 일반 SNMP에 비해 정확하며 합리적인 자료를 바탕으로 한 네트워크 관리 기능을 수행할 수 있음을 알 수 있다.

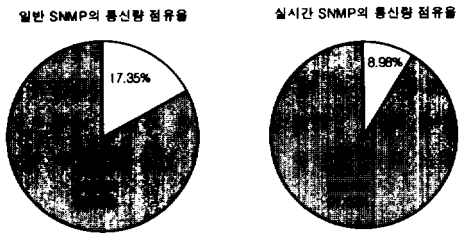
4.4 응답 시간과 응답 간격의 안정도

응답 시간과 응답 간격의 결과에서 한 가지 더 알 수 있는 사실은 실시간 SNMP의 경우 일반 SNMP에

비해 매 정보 수집에 따른 응답 시간의 변화가 급격하지 않음을 알 수 있다. 결과 그래프에서 로컬 네트워크 호스트 상에서의 비교 결과는 그 차이가 거의 없으므로 이는 고려하지 않는 것으로 한다. 그러나, 근거리 호스트와 원거리 호스트 상에서의 두 프로토콜 비교의 결과 그래프 상에 나타낸 바와 같이 급격한 성능 변화 점들이 일반 SNMP의 경우는 다수 존재한다. 반면, 실시간 SNMP에서는 일반 SNMP에 비해서 성능 변화점의 수가 적으며 또한, 그 변화폭도 일반 SNMP의 성능 변화에 비해서 크지 않음을 알 수 있다. 즉, 실시간 SNMP의 경우 일반 SNMP에 비해 여러 가지 요인 - 네트워크의 상황, 각 호스트에서의 프로세스 처리 시간 등 - 에 의한 응답 시간의 변화가 적다는 것이다. 이 결과의 이유는 두 프로토콜이 같은 조건하에서 일반 SNMP는 하나의 자료를 얻기 위해 두 번의 UDP packet의 전송이 이루어지나 일반 SNMP의 경우 한 번의 전송으로 가능하기 때문에 네트워크의 상황에 영향을 덜 받는 것이라 해석할 수 있다. 이는 특정 주기 동안 에이전트의 정보를 수집하는 데 있어서 일반적이고 정확한 정보 수집이 가능함을 나타낸다.

4.5 로컬 네트워크 통신량 점유율

다음으로는, 각 프로토콜이 로컬 네트워크의 자원을 얼마나 소모하는가를 알아 볼 수 있는 항목으로, 각 프로토콜을 이용하여 1초의 주기로 에이전트의 자료를 수집하는 동시에 네트워크 통신량 측정 프로그램을 이용하여 각 프로토콜마다 10분간 로컬 네트워크의 통신량을 5회씩 측정한 결과의 평균치이다. 결과 수치에 의하면, 다음의 그래프와 같이 일반 SNMP의 경우는 전체 IP 통신량에서 약 17.35%를 차지하는 반면, 실시간 SNMP의 경우는 8.98%를 차지함으로써 실시간 SNMP가 일반 SNMP에 비해 통신량을 약 1/2 가량 점유하는 것을 알 수 있다(그림 12).

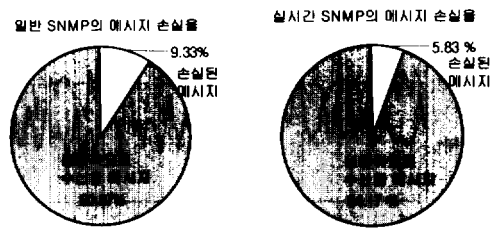


(그림 12) 각 프로토콜의 네트워크 통신량 점유율
(Fig. 12) Comparison of network traffic overheads for each protocol

이 것은 앞서 말한 바와 같이, 일반 SNMP의 경우 전통적인 요구-응답 방식을 사용하며, 실시간 SNMP의 경우 매니저의 요구 메시지의 응답 간격에 따라 주기적인 응답의 방식을 사용하는 단방향성 전송이라는 요인에서 기인한다.

4.6 메시지 손실율

마지막 비교 항목은 메시지 손실율이다. 앞서 말했듯이 SNMP의 메시지 손실은 UDP packet loss가 그 원인이며 네트워크의 혼잡도나 네트워크 자원의 부족 등으로 인한 것이다. 이는 각 프로토콜을 이용하여 에이전트에 600회의 폴링을 각각 5회씩 실시한 결과의 평균적인 수치이다. 네트워크의 통신량을 고려하여 timeout은 6000ms(6sec)로 설정하여 실험을 실시하였다. 실험의 결과에서 나타난 사항은 실시간 SNMP가 일반 SNMP에 비해 자료 손실이 약 1/2가량 적다는 것이다(그림 13).



(그림 13) 각 프로토콜의 메시지 손실율
(Fig. 13) Comparison of massage loss for each protocol

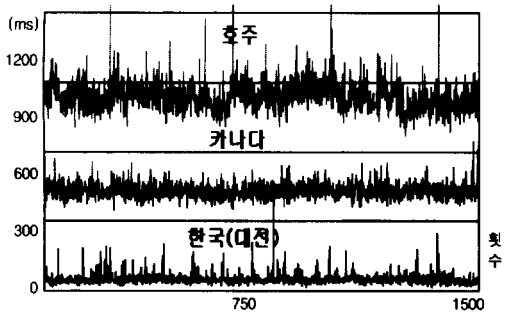
이 결과도 역시, 실시간 SNMP는 일반 SNMP와 달리 한 번의 UDP packet 전송이 이루어지므로 자료의 손실 확률이 적다는 것이 그 이유가 된다. 즉, 일반 SNMP의 경우에는 요구 메시지 손실율과 응답 메시지의 손실율의 합이 전체 메시지의 손실율이 되지만, 실시간 SNMP의 경우에는 일차 요구에 대한 단방향의 응답 메시지 손실율만이 전체 메시지의 손실율이 되는 것이다. 이는 실시간 SNMP의 커다란 장점이 된다. 네트워크 관리에서 관리 정보는 기본적인 요소이므로 값의 정확성이 필수적이며 관리국에 확실히 전달되어야만 하는 것이다. 그러나, 실제로 SNMP는 UDP를 기반으로 하는 프로토콜인 관계로 비 신뢰적 통신이 된다. 즉, 관리 정보가 분실될 가능성이 존재하며 정확한 시간에 관리국에 전달되어지지 않을 가능성이 높다. 결과에서 나타나듯이, 실시간 SNMP는

이러한 메시지 손실 가능성을 줄여줄 수 있다. 그러나, 만일 실시간 SNMP에서 요구 메시지가 손실된 경우에는 이후의 응답 메시지가 전혀 없게 되므로 이 것을 해결하기 위해 적절한 타임아웃 값을 설정해 주어야 할 것이다.

4.7 실시간 SNMP의 특징 분석

결과적으로 나타난 실시간 SNMP의 특징을 정리해 보면 첫째, 기존의 SNMP에 비해 적은 요구 메시지를 보내므로 네트워크의 통신량을 줄일 수 있다. 주기적 요구를 함으로써 매번 요구 메시지를 보내지 않고 한 번의 요구 메시지를 통해서 정확히 특정시간에 에이전트 정보를 받을 수 있는 기능을 제공한다. 결과적으로 네트워크의 통신량을 줄일 수 있는 효과를 갖는다.

둘째, 매번 폴링을 해서 정보를 주고받지 않아도 된다. 즉, 네트워크 상에서 GetRequest를 수행하는데 걸리는 시간을 절약 할 수 있다. 다음의 (그림 14)는 UNIX의 'ping'을 사용하여 호주, 카나다, 그리고 대전, 이렇게 3지역을 대상으로 각 지역마다 1500회의 응답 시간을 측정 비교해 보았다.



(그림 14) ping's rounding time 비교
(Fig. 14) Sample distribution for ping round trip time

(그림 3)에서 측정하는 컴퓨터와 거리가 멀수록 상대적으로 응답 시간도 오래 걸린다는 것을 알 수 있다. 만약 호주에 있는 에이전트의 MIB에 대한 주기적인 정보를 얻고자 할 경우 실시간 SNMP 주기적 요구를 사용한다면 기존의 SNMP보다 짧은 시간 안에 얻고자 하는 MIB의 정보를 얻을 수 있을 것이다.

셋째, 폴링의 성능 변화폭이 크지 않다. 즉, 평균적이며 안정적인 간격으로 관리 정보를 요구하고 응답을 수신할 수 있다.

넷째, 관리를 위한 메시지의 손실을 상당히 줄일 수 있다. 네트워크 관리를 위한 관리 정보의 정확성을 위해 이는 매우 중요한 장점이다.

다섯째, 기존의 SNMP 에이전트와 기능적으로 완전히 호환된다. 실시간 SNMP 에이전트는 기존의 SNMP 에이전트와 연동 함으로써 기존 SNMP 프로토콜을 수용하고 독자적인 실시간 모듈을 이용하여 실시간 SNMP GetRequest의 기능을 수행한다.

5. 결론 및 향후 계획

실험 결과와 같이 실시간 SNMP는 응답 시간의 단축, 수집 간격의 정확성 향상, 평준화된 응답 간격, 네트워크 자원의 절약, 수집 자료 분실 가능성의 축소 등의 장점으로 인해 안정적이며 정확하고 효율적인 네트워크 관리를 제공할 수 있다. 이러한 사실을 바탕으로 실시간 SNMP를 이용하여 현재의 복잡하고 그 범위가 확장되어 가는 네트워크 상황에서 자원과 비용의 절감 효과와 정확한 자료의 수집에 의한 적절하고 효율적인 네트워크 관리를 가능하게 해 줄 수 있다. 그리고, 실험 결과에서 나타나듯이 근거리보다는 원거리에서 관리 작업을 수행하는 데 있어서 그 성능과 효율성의 증가는 뚜렷하게 나타남을 알 수 있다. 이러한 특성을 이용하여 지역적으로 분산된 네트워크 관리에 있어서 각 지역적으로 독립된 관리 기능 수행이외에 하나의 지역에서 전 지역을 관리하는 중앙 집중적 관리가 필요한 경우에 이를 만족시킬 수 있다. 즉, 각 지역 단위에서의 개별적인 관리 기능과 함께 중앙에서 각 지역 단위에 대한 관리를 수행하는 계층적 관리기법에 커다란 효율 향상을 가져올 수 있는 것이다. 뿐만 아니라, 각 지역 단위가 공조 체제를 이루어 관리를 수행하는 수평적 분산 관리 기법에서도 그 지역 단위들 간의 자료 교환에 응용하여 역시 효율 향상을 가져올 수 있는 것이다.

추가적으로 연구가 필요한 부분은 자료 수집 간격에 대한 다양한 요구 - 수집 중단, 수집 간격의 변화 등 - 를 위한 요구 메시지 정의와 효율성 연구에 대한 것이 더 필요하겠다. 그리고, 실시간 SNMP의 요구 자료 응답 과정에서 IP source routing을 이용하여 시간대별로 혼잡한 네트워크 자원의 우회 등을 통한 응답 시간의 단축 등 IP option을 이용한 효율성 향상에 대한 연구가 필요하다.

참 고 문 헌

[1] J. Case, M. Fedor, M. Schoffstall, J. Davin, "A Simple Network Management Protocol (SNMP)," MIT Laboratory for Computer Science, RFC 1157, May 1990.

[2] M. Rose, K. McCloghrie, "Concise MIB Definitions," RFC 1212, March 1991.

[3] M. Rose, K. McCloghrie, "Structure and Identification of Management Information for TCP/IP -based Internets," RFC 1155, May 1990.

[4] D. Ferrari, D. Verma, "A Scheme for Real-Time Channel Establishment in Wide-Area Networks," IEEE Journal on Selected Areas in Communication(April 1990), pp.368-379.

[5] D. T. Perkins, "Understanding SNMP MIBs," September 1993.

[6] C. Picoto, P. Veiga, "Management of a WWW Server using SNMP," In 6th Joint European Networking Conference, May 15-18, 1995.

[7] M. R. Siegl, G. Trausmuth, "Hierarchical Networking Management : A Concept and its Prototype in SNMPv2," In 6th Joint European Networking Conference, May 15-18, 1995.

[8] W. Stallings, "SNMP, SNMPv2, and RMON : Practical Network Management Second Editions," Additon : on-Wesley Company, 1996.

[9] J. Stankovic and K. Ramamritham, "The Spring Kernel : A New Paradigm for Real-Time Systems," IEEE Software, Vol.8, No.3, pp.67-72, May 1991.

[10] K. H. Kim, L. Bacellar, Y. Kim and C. Subbaraman, "A Timeliness-Guaranteed Kernel Model -DREAM Kernel and Implementation Techniques," Workshop on Real-Time Computing Systems and Applications, October 1995.

[11] D. L. Stone, K. Jeffay, "An empirical study of delay jitter management policies," Proc. of Multimedia systems 2, 1995.

[12] C. M. Aras, J. F. Kurose, D. S. Reeves, and H. Schulzrinne, "Real-Time communication in packet -switched networks," Proc. of IEEE, Vol.82, pp. 122-139, January 1994.

[13] J. W. Dolter, P. Ramanathan, and K. G. Shin, "Performance analysis of virtual cut-through switching in HARTS : A hexagonal mesh multicomputer," IEEE Trans. on Computers, pp.669-680, June 1991.

김 동 수

e-mail : dskim@rtlab.skku.ac.kr

1998년 성균관대학교 정보공학과 졸업(학사)

1998년~현재 성균관대학교 전기전자 및 컴퓨터공학부 석사과정 재학중

관심분야 : 네트워크 관리, 보안 관리

정 태 명

e-mail : tmchung@ece.skku.ac.kr

1981년 연세대학교 전기공학과 졸업(학사)

1984년 University of Illinois at Chicago, 전자 계산학과 졸업(학사)

1987년 University of Illinois at Chicago, 컴퓨터 공학과 졸업(석사)

1995년 Purdue University, 컴퓨터 공학과 졸업(박사)

1985년~1987년 Waldner and Co., Systems Engineer

1987년~1990년 Bolt Bernek and Newman Labs., Staff Scientist

1995년~현재 성균관대학교 전기전자 및 컴퓨터공학부 교수

관심분야 : 실시간 시스템, 네트워크 관리, 보안 관리