
경향분석 정보의 효율적인 처리를 통한 네트워크 트래픽 감소 방안에 대한 연구

윤천균[†]

A study for the reduction of network traffic through an efficient processing of the trend analysis information

Chunkyun youn[†]

요 약 스마트 시대를 맞아 다양한 장비와 이를 이용한 많은 서비스 제공으로 동영상 서비스가 폭발적으로 늘어나면서 유무선망에 트래픽 과부하가 걸리고 있다. 이처럼 데이터 트래픽이 폭증함에 따라 네트워크의 안정적 관리 및 품질유지를 위한 네트워크관리시스템의 중요성은 증가되고 있다. 네트워크관리시스템은 수많은 네트워크 장비와 시스템들에 대한 다양한 관리를 위해 SNMP로 자원들과 소통하는 과정에서 상당히 많은 량의 트래픽이 발생하여 종종 네트워크의 트래픽 정체를 발생시킨다. 본 논문에서는 관리용 트래픽을 최소화하기 위해 특정기간 동안의 추세를 모니터링 하거나 누적 관리하는 경향분석 정보에 대해 SNMP의 관리자와 에이전트 간 불필요한 트래픽 발생을 최소화할 수 있는 방안을 제안하고 구현하였다. 시험 분석결과 기존 SNMP와 호환성을 유지하면서 네트워크 트래픽 부하감소 및 응답시간 단축에서 양호한 효율성을 보여주었다.

주제어 : 네트워크, SNMP, 트래픽 부하, 네트워크관리시스템, 응답시간

Abstract Network traffic demand is increasing explosively because of various smart equipment and services on smart era. It causes of traffic overload for wireless and wired network. Network management system is very important to control the explosion of data traffic. It uses SNMP to communicate with various network resources for management functions and creates lots of management traffic. Those are can be serious traffic congestion on a network. I propose an improving function of SNMP to minimize unnecessary traffics between manager and agent for collecting the Trend Analysis Information which is mainly used to monitor and accumulate for a specific time period in this paper. The results of test show it has compatibility with the existing SNMP and greatly decreases the amount of network traffic and response time.

Key Words : Network, SNMP, Traffic load, Network management system, response time

1. 연구 배경

스마트 시대를 맞아 네트워크 관리의 중요성이 어느 때 보다 커지고 있다. 아이폰이 출시 후 1년도 안되어 국내 스마트폰 가입자가 1000만 명을 돌파하고 올해 안에 2000만을 돌파할 전망이다. 이와 더불어 금융, 교통, 제조, 생산 등 산업전반에 걸친 스마트 장비를 이용한 다양한 서비스 제공으로 모바일 트래픽은 기하급수적으로 늘

어날 것으로 예상된다. 빠른 속도와 고품질로 전 세계인의 부러움 대상이었던 유선 네트워크 역시 P2P 사이트들의 범람과 IPTV, Smart TV 등의 출현으로 동영상 서비스가 폭발적으로 늘어나면서 무선망뿐만 아니라 유선망까지 트래픽 과부하가 걸리고 있다. 이러한 트래픽 문제는 스마트 장비 이용자가 늘어나고 이를 기반으로 한 서비스가 확산될수록 더 심각한 양상으로 나타날 것이다[1]. 이제 트래픽 관리는 유무선 가릴 것 없이 통신사업자

[†] 호남대학교 인터넷콘텐츠학과 교수(교신저자)

논문접수 : 2012년 1월 18일, 1차 수정을 거쳐, 심사완료 : 2012년 2월 17일

본 논문은 2010년도 호남대학교 학술연구비 지원을 받아 연구되었음

들의 가장 민감한 문제로 떠올랐다. 향후, 4년간 무선 데이터 트래픽은 지금보다 20배가, 유선 트래픽 역시 7배가 증가할 것으로 예상되고 있다. 시스코(Cisco)사의 2010년 보고서에 의하면 2009년 월평균 0.09엑사바이트(1Eb=109Gbytes)에 불과한 모바일 트래픽이 연평균 108%씩 증가하여 2014년에는 3.6Eb/월에 도달할 것으로 전망하고 있다[2].

이처럼 데이터 트래픽이 폭증함에 따라 네트워크의 안정적인 관리 및 품질유지를 위해 설치된 유무선 네트워크 장비들의 효율적인 운용과 관리를 위한 네트워크관리시스템의 중요성은 날로 증가되고 있다[3]. 네트워크관리시스템은 네트워크에 대한 성능관리, 구성 및 장애관리, 계정 및 보안 관리 등을 수행하기 위하여 SNMP(Simple Network Management Protocol)를 기반으로 네트워크상에 존재하는 다양한 자원들과 소통한다. 이는 이미 과부하 상태의 네트워크에 네트워크관리시스템에 의한 트래픽들을 추가시키는 것이 때문에 전체 네트워크에 심각한 트래픽 정체를 가중시킬 수 있다. 따라서 사용자들의 트래픽 수요를 최대한 수용하고 관리 트래픽의 추가를 최소화하면서 네트워크 장비들의 관리와 운용을 효율적으로 할 수 있는 연구는 네트워크관리시스템들에 있어서 매우 중요한 사항이다.

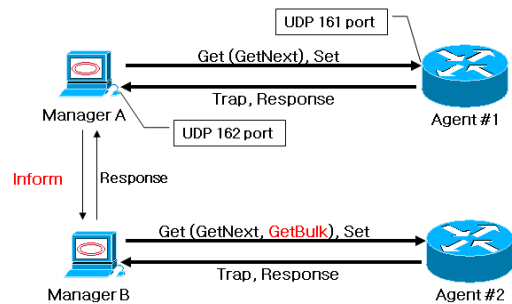
본 논문에서는 관리 트래픽을 최소화하기 위한 여러 가지 방안 중 네트워크관리시스템들이 네트워크 자원들과 통신을 위해 사용하는 SNMP에 대한 개선 연구 중의 하나인 ‘SNMP PDU(Protocol Data Unit)의 시간변수 추가를 통한 네트워크 성능향상에 관한 연구’[3]를 좀 더 개선함으로써 관리용 트래픽 발생을 최소화시켜 전체 네트워크의 부하를 감소할 수 있는 방안에 대하여 연구하고자 한다. 2장에서는 SNMP 구성과 동작 방식 및 관련 연구내용과 문제점들에 대해 알아보고 이의 개선안을 제안한다. 3장에서는 제안 내용의 프로토타입 시스템을 구현하여 그 결과를 기존 방식과 관련 연구와의 비교를 하고, 마지막으로 결론 및 향후 연구 방향에 대하여 기술하고자 한다.

2. 연구 내용

2.1 SNMP 구성 및 동작방식

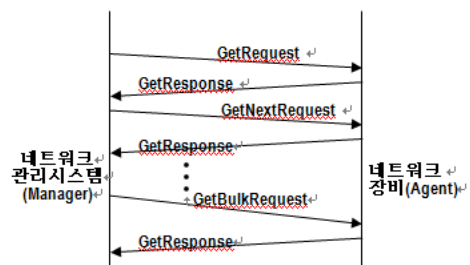
일반적으로 네트워크관리시스템들은 다양한 분석 항

목들에 대하여 TCP/IP 기반의 SNMP을 사용하여 각 네트워크 장비들의 MIB(Management Information Base)를 활용하여 수집한다. SNMP는 [그림 1]과 같이 Manager에서 운영하는 네트워크관리시스템의 관리화면을 통해서 측정하고자 하는 네트워크 장비(Agent)의 MIB object에 대한 정보 전송을 요청하고 해당 에이전트가 이에 응답하는 방식으로, 단일 object에 대해 Manager의 정보 요청과 Agent가 응답하는 형태의 정보 송수신이 발생한다[4][5][6][7].



[그림 1] SNMPv2 메시지 동작 과정

SNMPv2의 명령은 [그림 1]과 같이 GetRequest, GetNextRequest, GetResponse, GetBulkRequest, InformRequest, SetRequest, Trap으로 구성되어 있으며, 이 명령어들을 이용하여 매니저와 에이전트 간에 정보를 송수신 한다. 기존 SNMP에서는 분석항목들에 대한 관리정보 수집 시 [그림 2]와 같이 매니저가 정보 수집을 하고자 하는 에이전트의 MIB object에 대하여 상기 명령들을 보내고 해당 명령에 대하여 에이전트가 응답하는 방식의 데이터 송수신이 이루어진다[4][5][6][7].

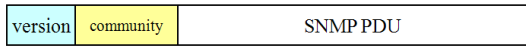


[그림 2] SNMP의 동작과정

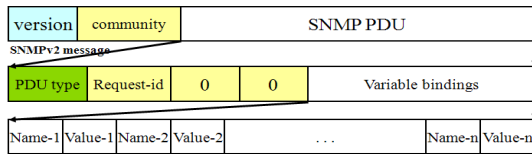
SNMPv2 PDU는 [그림 3]과 같은 포맷으로 구성되어 있으며, Version은 매니저와 에이전트 간에 사용 중인

SNMP 버전이다. v1, v2, v3가 있으며 메시지 포맷은 서로 약간 다르다. 만일 매니저와 에이전트간의 버전이 일치하지 않으면 에러가 발생한다[4][5][6][7].

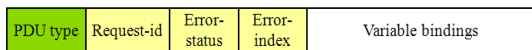
Community는 매니저와 에이전트 간에 간단한 인증을 위한 것으로 미리 정의된 값이 서로 일치해야 한다. 기본적으로 Get 에는 'public', Set에는 'private'를 사용한다.



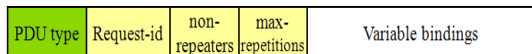
(a) SNMPv2 message



(b) GetRequest-PDU, GetNextRequest-PDU, Set Request-PDU, SNMPv2-Trap-PDU, Inform Request-PDU



(c) GetResponse-PDU



(d) GetBulkRequest-PDU

[그림 3] SNMPv2 PDU Format

SNMP-PDU는 매니저와 에이전트 사이의 요청과 응답에 대한 실제 정보가 저장되는 필드이다.

PDU-type은 매니저가 에이전트에게 요구하거나 에이전트로부터 매니저로 결과가 올 때 어떤 요청인지를 구분하기 위한 PDU 종류로 <표 1>과 같다.

<표 1> SNMPv2 메시지를 위한 PDU 형식

PDU type	value
0	GetRequest-PDU
1	Get-next-request-PDU
2	GetResponse-PDU
3	Set-request-PDU
4	Obsolete
5	Get-bulk-request-PDU
6	Inform-request-PDU
7	SNMPv2-Trap-PDU

RequestID는 매니저가 에이전트에 요구를 보낼 때 부여하는 ID로 해당 에이전트의 결과값 구분을 부여한 ID와 동일한 ID를 보고 판단 할 수 있다. 이를 이용하여 수신된 응답들을 ID로 정렬하여 구분할 수 있으므로 여러 에이전트로부터 다중 수신이 가능하다[4][6].

Error-status는 SNMP 에이전트가 SNMP 메시지를 해석하고 해당 객체를 처리할 때 해당객체에 또는 다른 문제가 발생한 경우 표시한다.

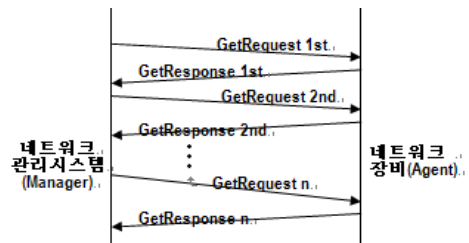
Error-index는 Variable bindings에서 특정한 위치의 값이 오류를 발생했을 때 몇 번째 변수인지를 표시하는 정수형 off set이다. 오류표기를 보고 다시 정보를 요청할 수 있다.

variable bindings는 매니저가 한번에 1개의 MIB object에 명령을 내리는 것 이외에 원하는 MIB object를 1~N 개까지 (OID + VALUE).....(OID + VALUE) 형태로 묶어서 동시에 내릴 수 있다. 에이전트는 각 객체의 OID와 그 객체의 값을 한 묶음으로 한 (OID+Value)(OID+Value) 형태로 메시지를 만들어 전달한다. 이것은 MTU(Maximum Transfer Unit)가 허용하는 범위까지 가능하다[4][6].

2.2 기존 SNMP와 관련연구의 문제점

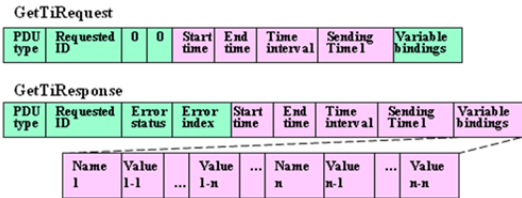
네트워크관리시스템에서 이용하는 다양한 정보들 중 '경향분석' 항목들은 어떤 object에 대하여 일정기간 동안 특정 주기로 측정된 정보를 이용하여 해당 기간 동안의 trend를 모니터링 하는데 주로 이용되고 있다[3].

이와 같은 특성을 가진 분석항목들의 경우 기존 SNMP 방식을 적용하면 [그림 4]와 같이 매니저와 에이전트 간 일정주기로 동일한 요청(GetRequest)과 응답(GetResponse)을 반복하는 트래픽이 발생하게 된다. 만일 네트워크관리 시스템에서 여러 장비들로부터 이 분석항목에 대해 짧은 주기로 장시간 수집을 할 경우 대량의 트래픽 발생 및 관리시스템의 과부하가 발생한다[3][8][9][10].



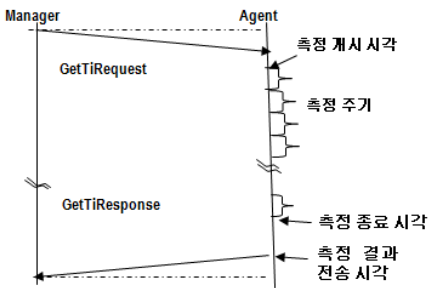
[그림 4] 경향분석항목에 대한 기존SNMP 처리방식

관련 연구에서는 네트워크 관리를 위한 인터넷 분석 항목에 대해 실시간분석, 기본분석, 심화분석으로 대별하여 심화분석 중 반복수집이 필요한 경향분석 정보(Trend Information)에 대한 성능개선에 대하여 연구를 하였다[3]. 이를 위해 경향분석 정보용 SNMP PDU의 개선 시 기존 SNMP PDU외에 [그림 5]와 같이 2개의 GetTiRequest와 GetTiResponse PDU를 추가하였다. 추가 PDU는 기존 GetRequest, GetResponse PDU 구조에 “start time”, “end time”, “Time interval”, “Sending time” 필드 추가와 variable binding 필드를 확장한 형태이다[3].



[그림 5] 경향분석 정보용 추가 SNMP PDU

동작방식은 [그림 6]과 같이 매니저가 [그림 5]에서 정의한 GetTiRequest를 에이전트에 전송하고 에이전트는 이 PDU 정보를 분석하여 요구한 ‘start time’에 맞추어 해당 MIB들에 대한 정보를 정의된 ‘interval’ 주기로 ‘end time’까지 수집하여 ‘sending time’에 ‘GetTiResponse PDU’를 생성하여 수집된 결과를 매니저에게 일괄 전송한다[3].



[그림 6] 관련연구의 경향분석 정보 용 처리 수순

관련 연구에서는 [그림 5]와 같이 기존 SNMP의 구조 변형이 상대적으로 크기 때문에 기존 SNMP와의 호환성 문제와 경향분석 정보 수집 시 매니저와 에이전트 간 송수신 프레임 크기 증가에 따른 문제가 제기되었다.

본 논문에서는 관련 연구 결과를 더 개선하여 SNMP PDU 구조 변경을 최소화하고 매니저와 에이전트 간 송수신 프레임 크기도 최소화시켜 네트워크의 부하를 더 경감할 수 있는 방안에 대하여 연구하고자 한다.

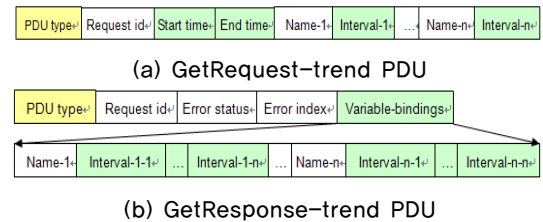
2.3 경향분석 정보에 대한 처리방식 개선 제안

‘경향분석’ 정보들은 주로 특정 기간 동안의 추세를 그래프 형태로 모니터링 하거나 누적된 정보를 통계처리 목적으로 이용하기 때문에 에이전트는 매니저의 요청에 대해 실시간 응답보다는 특정 기간의 데이터를 수집이 완료된 시점에 일괄 전송해 주면 해당 목적을 달성하는데 지장이 없다[3].

따라서 ‘경향분석’에 대한 정보수집 시 SNMP의 동작 방식을 기존의 [그림 4]와 [그림 6]과 같은 방식에서 [그림 8] 방식과 같이 개선한다. 경향분석 정보 관리를 위해 기존 SNMP의 PDU type외에 운용자가 네트워크관리시스템의 화면을 통해 분석항목을 중 ‘경향분석’ 항목들을 구분하여 설정할 수 있도록 <표 2>와 같이 ‘PDU type 10’을 추가하였고, 에이전트가 수집 결과를 전송할 수 있도록 ‘PDU type 11’를 추가하였다. [그림 7]과 같이 ‘Get Request PDU’의 형태를 그대로 유지한 ‘GetRequest-trend’와 ‘variable bindings’ 필드를 일부 확장한 ‘GetResponse-trend PDU’를 추가하였다.

<표 2> 추가 PDU 형식

PDU type	value
10	GetRequest-trend
11	GetResponse-trend



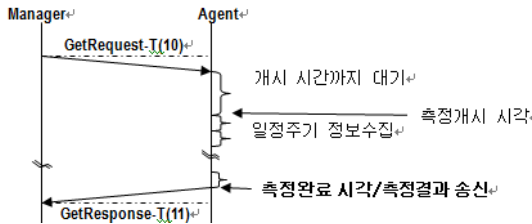
[그림 7] 제안한 SNMP PDU Format

본 논문에서 제안한 GetRequest-trend (GetRequest-T) PDU의 구성은 기존 ‘GetRequest PDU’의 구조를 변형시키지 않고 “0”으로 고정된 3,4번째 필드와 ‘null’ 값으로 송신되는 ‘variable binding’ 필드의 ‘value-n’ 필드를

활용한다. [그림 7]과 같이 해당 필드들을 ‘start time(측정 시작시각)’, ‘end time(측정 종료시각)’과 측정하고자 하는 MIB의 ‘interval(측정 주기)’로 정의한다.

GetResponse-trend(GetResponse-T) PDU는 [그림 3]의 기존 SNMP 형태와 유사하나 [그림 7]과 같이 ‘variable binding’ 필드에 매니저가 요구한 MIB 변수들의 값을 측정 주기 별로 저장할 수 있도록 ‘variable binding’ 필드의 사이즈를 확장하였다.

동작방식은 운영자가 네트워크관리시스템의 관리 화면을 통해서 ‘경향분석’ 유형의 정보 수집을 요청할 경우에만 매니저에서는 GetRequest PDU의 PDU type 필드에 GetRequest-trend를 의미하는 ‘10’을 설정하고 “0”으로 고정된 3,4번째 필드에 ‘start time’과 ‘end time’을 설정하고, null 값으로 송신되는 ‘variable binding’ 필드의 ‘value-n’ 필드에 MIB별 ‘interval’을 설정하여 에이전트에 송신한다. 이 때 에이전트로부터의 결과값 수신을 위해 해당 요청의 ‘Request id’, ‘start time’, ‘end time’, ‘interval’ 값들을 기억해 둔다. 물론 경향분석 이외의 경우에는 기존 SNMP의 방식을 준수한다.



[그림 8] 제안 SNMP의 처리 수순

에이전트에서는 수신한 PDU의 ‘PDU type’ 필드의 값을 보고 ‘경향분석’임을 인식하고 요구한 ‘start time’에 맞추어 [그림 8]과 같은 방식으로 해당 MIB들에 대한 정보를 ‘interval’ 주기로 ‘end time’까지 수집하여 [그림 7]의 ‘GetResponse-trend PDU’를 생성하여 일괄 전송한다. 이 때 ‘Request id’는 매니저로부터 수신한 ‘Request id’를 사용한다.

에이전트로부터 ‘GetResponse-trend PDU’를 수신한 매니저는 ‘Request id’를 확인하여 매니저에서 송신했던 해당 ‘GetRequest-trend’ PDU의 ‘start time’, ‘end time’, ‘interval’ 값과 ‘GetResponse-trend PDU’의 ‘variable binding’에 저장된 측정값들을 이용하여 정확한 그래프를 표시한다.

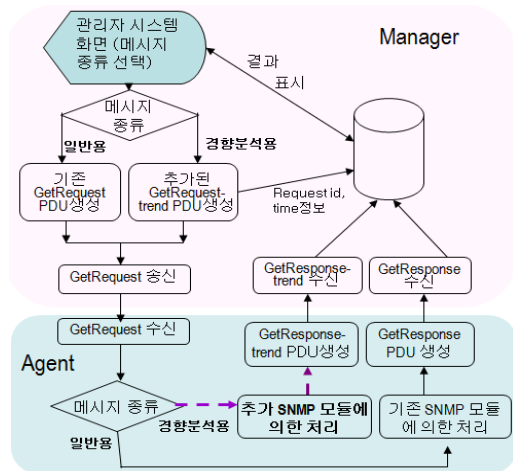
3. 시험 및 결과 분석

제안한 SNMP의 성능비교 시험을 위해 프로토타입(prototype) 시스템을 구현하고 이를 통해 기존 및 관련 연구와 비교분석을 실시한다.

3.1 프로토타입 주요 모듈의 기능구성

제안한 SNMP 방식을 구현한 프로토타입 시스템의 주요 모듈 구성은 [그림 9]와같이 매니저와 에이전트로 구분되며, 매니저는 운영자가 사용하는 네트워크관리시스템의 화면을 통해서 입력된 관리정보의 종류가 경향분석용 또는 일반용인지에 따라 ‘GetRequest/GetRequest-trend’ PDU를 생성하여 에이전트로 전송하는 ‘GetRequest 송신’ 모듈과 에이전트로부터 수신한 GetResponse/GetResponse-trend를 처리하는 모듈들로 구성된다.

에이전트는 매니저가 보내온 GetRequest를 수신하여 일반용과 경향분석용으로 구분하고 이 구분에 따라 기존 SNMP 방식 또는 추가 SNMP 모듈에 의해 정보를 수집하고 그 결과를 GetResponse/ GetResponse-trend PDU로 생성하여 전송하는 모듈로 구성된다.



[그림 9] 제안 SNMP의 주요 모듈 구성도

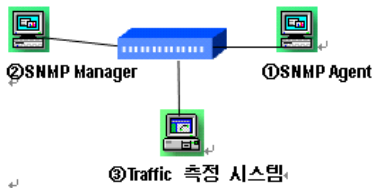
3.2 시험 환경

제안을 구현한 프로토타입 시스템의 정상동작 여부와 기존 SNMP 및 관련 연구와의 비교를 위하여 관련 연구의 시험환경과 유사한 조건을 유지하였다.

이를 위해 [그림 10]과 같이 각 시스템들은 Shared media인 단순 허브를 공유하는 Ethernet LAN 형태의

test bed를 구축하고, 매니저(IP:211.227.254.116) 및 측정 시스템(IP:211.227.254. 217)으로 Window XP 시스템을 사용하였고, 에이전트(IP:211.227.254.214)는 Window XP 에 Fedora9를 설치하였다. 매니저와 에이전트용 SNMP 프로그램은 관련 연구에서 사용한 UCD SNMPv3.4를 C 언어를 이용하여 모듈을 수정하고 추가하였으며, 외부에서 인터넷으로 접속하여 매니저 시스템을 이용하여 실험을 할 수 있도록 매니저 시스템에 웹 서버를 구축하고 매니저 기동용 관리화면을 Web에서 동작할 수 있도록 shell 기반 CGI를 이용하여 작성하였다[11][12].

측정 시스템에는 프로토타입 시스템의 정상동작 여부를 확인하기 용이한 네트워크 관리 툴인AppDancer FA (AppDancer Networks, inc.)를 설치하여 매니저와 에이전트 간 전송 과정과 트래픽 유형과 프레임 크기, 응답시간 등을 측정하였다.



[그림 10] Test bed 구성도

3.3 프로토타입 시스템의 작동 시험

[그림 10]의 측정 시스템에서 Web을 이용하여 원격으로 매니저 시스템을 기동시켜 SNMP 매니저와 에이전트 간의 SNMP 송수신을 발생시키고, 네트워크 관리 Tool들로 네트워크상의 트래픽을 측정한다. 실험조건은 관련연구에서 사용한 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 5 초 주기로 30분간 전송한 경우와 단일 항목을 1회 측정하는 시험을 하였다.

	Src. Addr	Dst. Addr	Protocol	Errors	Avg. Resp. Time	Total Frames	Update Time
1	211.227.254.217/138	211.227.254.255/138	NETBIOS	0	0.000000	5	14/12/2011 20:47:38.693.506
2	211.227.254.217/1461	211.227.254.116/80	HTTP	0	0.000756	2263	14/12/2011 20:47:19.023.316
3	211.227.254.116/38340	211.227.254.214/161	SNMP	0	0.001973	2	14/12/2011 20:46:54.024.356
4	211.227.254.116/38347	211.227.254.214/161	SNMP	0	0.001905	2	14/12/2011 20:46:48.905.338
5	211.227.254.217/137	211.227.254.255/137	NETBIOS	0	0.000000	9	14/12/2011 20:46:48.041.888
6	211.227.254.116/38346	211.227.254.214/161	SNMP	0	0.001977	2	14/12/2011 20:46:43.785.424
7	211.227.254.116/38345	211.227.254.214/161	SNMP	0	0.001947	2	14/12/2011 20:46:38.687.772

Frame #	Delta Time	Rel. Time	Len
4488	0.000000	0.000000	87
4489	0.001973	0.001973	80

[그림 11] 기존 SNMP traffic 처리 수순 갈무리 화면

[그림 11]은 기존 SNMP에서 매니저가 에이전트로 GetRequest를 보내고 에이전트에서는 이에 대한 응답으로 GetResponse를 매니저로 보내는 과정이 [그림 14]와 같이 반복적으로 발생함을 알 수 있고, 이 때 1회의 메시지 송수신을 위하여 소요된 응답시간은 약 2msec이었으며, 각 프레임의 크기는 87Bytes와 90Bytes임을 알 수 있다.

Frame #	Delta Time	Rel. Time	Len
1	0.000000	0.000000	87
2	0.001947	0.001947	160

[그림 12] 제안 SNMP traffic 처리 수순 갈무리 화면

[그림 12]는 본 논문에서 제안한 경우로 관련 연구에서와 같은 조건에서 5초 주기로 30분간 측정된 것을 SNMP 프로토콜만 필터링하여 본 것이다. "Error: Couldn't parse variable bindings header: Wrong type for SNMP" 메시지가 나타난 이유는 <표 2>와 같이 추가된 송수신 PDU type들이 표준 SNMP에 등록되어 있지 않기 때문에 네트워크 관리 툴인AppDancer가 인식하지 못하여 발생한 것이며, 메시지 송수신이 [그림 8]과 같이 이루어지고 있다는 것을 알 수 있다. 매니저에서 송신한 PDU 크기는 기존 SNMP와 동일한 87B이며, 에이전트로부터 수신한 PDU 크기는 1,493B로 관련 연구에 비해 각각 21B씩 작음을 알 수 있다. 평균 응답시간은 30분간(1800=30*60)의 실적수집 시간과 추가로 약11.7초가 소요되었고 관련 연구와 거의 동일하나 본 논문에서 제안한 경우가 약67msec 빠르다.

위의 [그림 11], [그림 12]에서 알 수 있듯이 본 논문의 제안을 구현한 프로토타입 시스템이 정상적으로 동작되고 기존 SNMP와의 호환성에도 문제가 없음을 알 수 있다.

Frame #	Delta Time	Rel. Time	Len
1	0.000000	0.000000	80
2	0.001973	0.001973	161

[그림 13] 관련연구 traffic 처리 수순 갈무리 화면

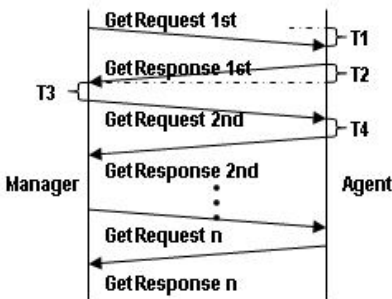
[그림 13]은 관련연구의 경우를 동일조건으로 측정한 것으로 앞의 경우와 같이 "Error: Wrong type" 메시지가 나타난 이유 역시 송수신되는 PDU 형태가 표준 SNMP PDU와 달라 분석 툴에서 인지하지 못하기 때문이다. 메시지의 송수신 수준은 정상적으로 이루어진 것을 알 수 있다.

매니저에서 송신한 PDU 크기는 [그림 5]와 같이 기존 SNMP에 비해 "Start time", "End time", "Time interval", "Sending time" 필드 추가 때문에 21B가 큰 108B이고, 에이전트로부터 수신한 PDU 크기도 제안 SNMP의 경우에 비해 21B가 큰 1,514B임을 알 수 있다. 평균 응답시간은 약1812.4초로 제안 경우에 비해 약 67msec 더 걸렸음을 알 수 있다.

3.4 시험 결과 비교 분석

3.4.1 네트워크 트래픽

기존 SNMP에서 매니저와 에이전트간의 경향분석 정보에 대한 네트워크 트래픽 발생은 [그림14]와 같다.



[그림 14] 경향분석 정보 수집 시 기존 SNMP의 동작 수준

매니저가 발생시킨 GetRequest PDU에 의한 네트워크 트래픽 양을 Trq 라 하고, 에이전트의 응답인 GetResponse PDU에 의한 트래픽 양을 Trs 라 하면, 1회의 정보수집 시 발생하는 네트워크 트래픽 량(To)은 (식 1)과 같으며, 정보 수집기간이 Dt 이며 수집 간격이 It 일 경우의 발생하는 총 트래픽 양(Tot)은 (식 2)와 같다.

$$To = Trq + Trs \quad (식 1)$$

$$Tot = Dt/It * To \quad (식 2)$$

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

관련 연구에서와 본 논문에서 제안한 경우의 트래픽 발생 수준은 [그림 6] [그림 8]과 같으며, 매니저에 의해 송신된 PDU의 트래픽 량을 $Trqt$ 라 하고 에이전트로부터 수신한 PDU에 의한 트래픽 량을 $Trst$ 라 하면, 1회의 정보수집 시 발생하는 네트워크 트래픽 량(Top)은 (식 3)과 같다. 정보 수집기간이 Dt 이며 수집 간격이 It 일 경우의 발생하는 총 네트워크 트래픽 량($Totp$)은 (식 4)와 같다. 여기서 (식 3)과 (식 4)가 동일한 형식이나 그 결과는 서로 다르다.

$$Top = Trqt + Trst \quad (식 3)$$

$$Totp = Trqt + Trst \quad (식 4)$$

1) 기존 SNNMP와 제안 SNMP 비교

단일 전송의 경우 [그림 3]과 [그림 7]의 PDU구성에서 알 수 있듯이 GetRequest-trend($Trqt$)와 GetRequest(Trq)의 크기가 동일하고 GetResponse-trend($Trst$)와 GetResponse(Trs)도 동일하기 때문에 $Trqt=Trq$, $Trst=Trs$ 의 관계가 성립되어 (식1)과 (식3)에서 $Top=To$ 관계가 된다. 이들 간의 차이는 요청하는 변수와 측정횟수에 따라 차이가 발생하지만 단일 변수를 1회 측정하는 경우 [그림 11]과 [그림 15]에서 알 수 있듯이 동일한 량의 트래픽이 발생한다.

Src Addr	Dst Addr	Pro.	Errors	Avg. Resp. Time	Total Frames	Update Time
1	211.227.254.116 f32816	211.227.254.214 f181	SNMP	0	1.981625	2/14/12/2011 17:53:53.971.625

Frame #	Delta Time	Rel. Time	Len
1	0.000000	0.000001	97
2	1.981625	1.981625	98

[그림 15] 단일 항목 1회 측정 시 갈무리 화면 (본 논문 제안 경우)

본 논문에서 중점을 둔 경향분석 정보의 경우 순수 SNMP에 의하여 발생된 트래픽만을 비교해 보면 기존 SNMP는 [그림 11]과 [그림 14]를 바탕으로 (식1)과 (식 2)에 따라 계산해보면 63,720 bytes ((87B + 90B) * 60/5 초 * 30분)가 발생되었고, 제안한 SNMP의 경우에는 [그림 8]과 [그림 12]를 바탕으로 (식3)과 (식4)에 의해 계산해보면 1,580 bytes(87B+1,493B)가 발생하여 제안한 SNMP의 네트워크 트래픽이 기존 SNMP에 비해 약 40.3 배 감소한다.

2) 관련 연구와 제안 SNMP 비교

관련 연구에 의한 경향분석 정보 수집 시의 매니저와 에이전트 간의 네트워크 트래픽 발생 수준은 [그림 6]과 같으며 본 논문에서 제안한 과정은 [그림 8]과 같다. 두 경우의 차이점은 측정완료 후 측정결과를 즉시 전송하느냐 아니면 특정시각을 지정하여 대기 후 전송하느냐 일뿐 동일하다고 할 수 있다.

그러나 발생하는 트래픽 량의 경우 전송하는 PDU의 format이 [그림 5]와 [그림 7]에서와 같이 다르기 때문에 본 논문에서 제안한 경우가 관련 연구보다 'start time'(6B), 'end time'(6B), 'time interval'(3B), 'sending time'(6B) 필드 크기 만큼인 21bytes가 감소한다. 따라서 (식3)과 (식4)를 적용하면 제안한 경우의 트래픽 발생은 관련 연구에 의해 발생된 트래픽 량 대비 (식 5)와 (식 6)과 같이 표현할 수 있다.

$$Top = (Trqt - 21B) + (Trst - 21B) \quad (\text{식 } 5)$$

$$Totp = (Trqt - 21B) + (Trst - 21B) \quad (\text{식 } 6)$$

즉, 단일 전송 경우와 일정기간 측정 후 전송하는 두 경우 모두 제안 시스템의 네트워크 트래픽이 관련 연구에 비해 42bytes씩 감소한다.

Src Addr	Dst Addr	Protocol	Errors	Avg. Resp. Time	Total Frames	Update Time
211.227.254.116	211.227.254.214	SNMP	0	1.974971	2	14/12/2011 17:14:47.974.971

Frame #	Statistics	Delta Time	Rel. Time	Len
1	ERROR: Couldn't parse variable bindings header. Wrong	0.000000	0.000000	195
2	ERROR: Couldn't parse variable bindings header. Wrong	1.974971	1.974971	195

[그림 16] 단일 항목 1회 측정 시 갈무리 화면(관련 연구의 경우)

각각의 구현된 시스템에서 시험 측정한 결과 5초 주기로 30분간 측정할 경우에는 [그림 12]와 [그림 13]에서 알 수 있듯이 본 논문의 제안 시스템에서 각 21B씩 총 42B의 트래픽이 감소되었다. 그러나 단일 항목을 1회 측정할 경우 관련 연구에서는 [그림 16]에서와 같이 108B와 152B의 트래픽 발생되었고, 제안 시스템에서는 [그림 15]에서와 같이 87B와 90B가 발생되어 제안 시스템이 21B와 62B씩 감소하였다. (식5)에 의하면 21B 감소가 발생해야하나 시험에서는 41B가 더 발생하였다. 원인분석 결과 관련 연구의 프로그램 구현 시 에이전트에서 매니저로 전송하는 PDU 내의 Variable binding 필드의 측정 항목과 측정값을 저장하는 방법에 차이가 있었다.

3.4.2 응답시간

1) 기존 SNNMP와 제안 SNMP 비교

기존과 제안한 SNMP에서 1회의 측정 시 소요되는 응답시간(Rt)은 [그림 11]과 [그림 15]에서 알 수 있듯이 PDU크기가 동일하므로 논리적으로는 (식 7)과 같이 표기할 수 있다. 그러나 시험을 통해 측정된 평균응답시간(Avg.Resp.Time)은 1.973msec와 1961.625 msec로 매우 차이가 크다. 그 이유는 본 논문 제안 시스템의 경우 도입된 시각 필드들의 설정값 최소단위가 초(sec)이고, 가장 짧은 시간에 단일항목을 1회 측정하기 위해서는 [그림 8]에서 알 수 있듯이 에이전트의 수신 후 대기시간이 거의 "0"이 될 수 있도록 설정하고 측정개시 시각과 측정완료 시각차를 1초로 설정할 수밖에 없기 때문이다.

$$Rt = T1+T2+T4 \approx T1+T2 \quad (\text{식 } 7)$$

여기서([그림 14] 참조) 각 시간은 다음과 같은 의미를 갖는다.

- T1 : 매니저에서 에이전트에 전송 시 소요되는 전송 지연 시간
- T2 : 에이전트에서 매니저에 전송 시 소요되는 전송 지연 시간
- T3 : 매니저에서 에이전트의 응답 수신 후 소요되는 처리시간 및 대기 시간
- T4 : 에이전트에서 매니저의 요청 수신 후 CPU에서 처리하는데 걸리는 시간으로 다른 시간에 비하여 매우 짧기 때문에 거의 0ms 라고 해도 무리가 없다.

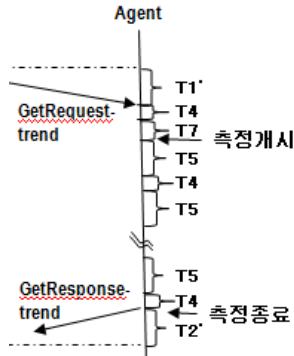
기존 SNMP의 경우 정보 수집기간이 Dt이며 수집주기가 It일 경우인 경향분석 정보를 수집 시 소요되는 총 응답시간(Rtt)은 (식8)과 같다.

$$Rtt = Dt/It * Rt \approx Dt/It * (T1+T2) \quad (\text{식 } 8)$$

제안 SNMP의 경우 [그림 8]과 같이 동작하므로 매니저의 GetRequest-trend에 대하여 에이전트에서는 [그림 17]과 같은 처리가 이루어진다.

- T7 : 수신한 GetRequest-trend를 측정개시시각까지 대기하는 시간, 응답시간에는 무관하므로 제외시킨다.
- T5 : 요청한 객체의 값을 측정 후 다음 주기까지 대기하는 시간, 기존 SNMP에서는 [그림 14]와 같이

결과를 매니저에 전송하고 매니저로부터 다시 GetRequest를 수신할 때까지 대기하는 T3에 해당한다.



[그림 17] 제안한 SNMP의 에이전트 처리 과정

제안 SNMP와 기존 SNMP간의 측정방법이 상이하어 엄격한 정의에 따른 응답시간 비교는 곤란하지만 성능비교를 위하여 관련 연구와 같은 개념을 적용하면 제안 SNMP에서의 총 응답시간(Rttp)은

$$R_{ttp} \doteq D_t / I_t * T_4 + T_1' + T_2' \quad (\text{식 9})$$

과 같이 표기할 수 있다. T1, T2와 T1', T2'의 차이는 [그림 3]과 [그림 7]에서 알 수 있듯이 PDU들의 크기 차에 의한 전송 시간차이로 T1'=T1, T2' ≥ T2의 관계가 성립되고 (식 7)에서와 같이 T4을 0msec로 적용하면 (식 10)과 같이 표기할 수 있다.

$$R_{ttp} \doteq D_t / I_t * T_4 + T_1 + T_2' \doteq T_1 + T_2' \quad (\text{식 10})$$

(식 8)과 (식 10)에서 알 수 있듯이 제안 시스템의 응답시간은 수집시간이 길고 측정주기가 짧을수록 기존 SNMP에 비해 빨라지기 때문에 제안 SNMP가 경향분석 정보 관리에 매우 효과적임을 알 수 있다.

2) 관련 연구와 제안 SNMP 비교

관련 연구와 본 논문에서 제안한 SNMP의 차이점은 [그림 6]과 [그림 8]에서 알 수 있듯이 측정완료 후 측정결과를 즉시 전송하느냐 아니면 특정시각을 지정하여 대기 후 전송하느냐 뿐이다. 따라서 관련연구에서 'sending time'을 0으로 설정하면 서로 같다. 다만 [그림 5]와 [그림

7]에서 알 수 있듯이 시각 필드들에 의한 PDU의 크기가 다르기 때문에 본 논문에서 제안한 경우가 관련 연구보다 21bytes를 전송하는데 소요되는 시간만큼 빠르다. 따라서 (식 7)과 (식 10)을 적용하면 제안 SNMP(Rtp, Rttp)와 관련연구(Rtr, Rtrr)에서의 응답시간과 총 응답시간은 아래와 같이 표현할 수 있다.

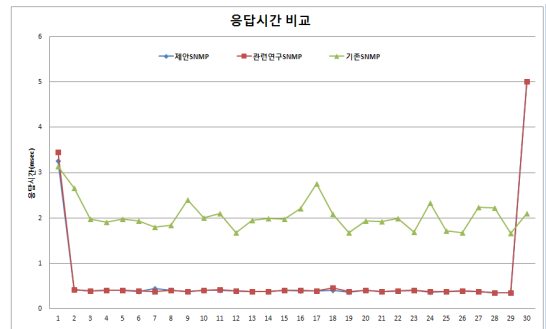
$$R_{tr} = (T_1' + 21B) + (T_2' + 21B) = R_{tp} + 42B = R_t + 42B \quad (\text{식 11})$$

$$R_{trr} = (T_1' + 21B) + (T_2' + 21B) = R_{tpp} + 42B \quad (\text{식 12})$$

(식 12)의 총 응답시간의 경우 단일전송 경우인 (식 11)에 비해 T2'가 매우 크므로 42B차가 응답시간에 미치는 영향력이 상대적으로 작으므로 (식 12)는 (식 13)과 같이 표현할 수 있다.

$$R_{trr} = R_{tpp} + 42B \doteq R_{tpp} \quad (\text{식 13})$$

(식 11)과 (식 13)에서 알 수 있듯이 단일전송의 경우 제안 시스템의 응답시간이 관련연구에 비해 빠르고, 총 응답시간의 경우에는 거의 동일하다고 볼 수 있다.



[그림 18] LAN 환경에서의 응답시간 시험 결과 비교

[그림 18]은 관련 연구에서와 동일한 환경을 구성한 [그림 10]과 같은 test bed에 제안 SNMP와 관련연구 SNMP의 구현 프로그램을 각각 설치하여 MIB 객체 "ifInOctets (1.3.6.1.2.1.2.2.1.10)"을 매니저에서 에이전트로 5초 주기로 30분간 송신하여 1분 주기의 평균값으로 표기한 시험 결과이다. 시험의 편의를 위해 관련연구에서의 'sending time'값을 0초로 설정하였다.

시험 초기가 중간보다 상대적으로 응답시간이 긴 것은 측정 시스템에서 원격으로 매니저를 기동 시 발생하는 HTTP 트래픽과 매니저가 에이전트로 GetRequest-

trend를 전송하기 위한 네트워크 연결에 필요한 트래픽 등이 추가로 발생하기 때문이다.

초기 응답시간이 제안 SNMP 3.26msec, 관련연구 SNMP 3.46msec, 기존 SNMP가 3.14msec로 비슷하지만 다소 차이를 보이는 것은 제안 SNMP가 기존 SNMP에 비해 프로그램 모듈이 추가되어 0.12msec 느리고, 관련 연구 SNMP모듈은 프로그램 모듈 추가와 PDU가 21B 더 크기 때문에 0.32msec 느린 것을 알 수 있다.

중간부분에서 기존 SNMP가 제안 SNMP와 관련연구 SNMP에 비해 상대적으로 큰 차이를 나타내는 이유는 기존 SNMP 경우 상대적으로 시간소모가 많은 [그림 14]와 같은 반복적인 PDU 송수신과 에이전트에서의 MIB값 수집에 많은 시간이 소요된 반면 관련연구와 제안 SNMP는 [그림 6]과 [그림 8]과 같이 동작하므로 초기 이후에는 상대적으로 빠른 MIB값 수집을 위한 시간만 소요되기 때문이다.

마지막 수집결과 전송 시 제안 SNMP와 관련연구 SNMP가 기존 SNMP에 비해 PDU 크기가 매우 크므로 더 많은 시간이 소요됨을 알 수 있고, 제안 SNMP와 관련연구 SNMP는 크기가 유사하기 때문에 응답시간이 5.00msec, 5.01msec로 거의 동일하다.

평균 응답시간은 제안 SNMP 0.459msec, 관련연구 SNMP 0.469msec, 기존 SNMP 2.054msec로 제안 SNMP가 기존 SNMP에 비해 약 4.5배 빠르고, 누적 응답시간은 제안 SNMP 13.76msec, 관련연구 SNMP 14.06msec, 기존 SNMP가 61.62msec로 제안 SNMP가 기존 SNMP 대비 약 4.8배가 개선되었음을 알 수 있다.

본 논문에서 시험한 결과를 정리하면 <표 3>과 같다.

<표 3> 시험 결과 비교표

항목	상세구분	기존 SNMP	관련연구	본 논문
트래픽	1회	177B	260B	177B
	5초주기 30분	63,720B	1,622B	1,580B
응답 시간	처음 1회	3.14msec	3.46msec	3.26msec
	5초주기 30' 평균	2.054msec	0.469msec	0.459msec
	“ 누적	61.62msec	14.06msec	13.76msec

관련 연구에서와 동일한 환경 하에서 시험한 제안 SNMP의 시험과 관련 연구에서의 시험 결과 값들이 차이가 있는 것은 [그림 10]의 test bed 구성 시스템의

CPU, memory, 네트워크 장비 등의 성능이 향상되었기 때문이다.

4. 결론 및 향후 연구방향

본 논문에서 제안한 SNMP을 바탕으로 개선된 특징을 정리해보면 첫째, 관련연구에서와 같이 기존 SNMP에 비해 매니저와 에이전트 간 송수신 메시지 횟수를 격감시킴으로써 네트워크 트래픽 부하 대폭 감소 및 응답시간 단축 등으로 네트워크 관리의 효율성을 크게 향상시켰다.

둘째, 관련 연구에 비하여 매니저와 에이전트 간 송수신 PDU들의 크기를 각각 21bytes씩 감소시켜 네트워크 부하 감소 및 응답시간 단축에서 상대적으로 양호한 효율성을 보여주었다. 이는 최근 네트워크관리시스템이 관리장비의 다양화 및 관리범위의 global화 추세여서 제안 방식 적용 시 매우 효율적인 네트워크 관리방안이 될 것이다.

셋째, 본 논문의 제안을 구현한 프로토타입 시스템에서 볼 수 있듯이 기존 SNMP의 PDU 프레임 변형을 최소화하여 기존 SNMP 기능과 완전한 호환성을 유지하였다.

향후 연구방향은 본 논문에서 다루지 않은 일정시간 동안 주기적인 측정이 필요한 분석정보를 실시간으로 수신하여 관리하고자 할 때 매니저와 에이전트 간의 메시지 전송을 최소화할 수 있는 방법에 대하여 연구하고자 한다. 더불어 위 연구들을 통합한 효율적인 네트워크 관리시스템에 대하여 연구하고자 한다.

참 고 문 헌

- [1] 오정숙 (2011). 국내외 네트워크 장비 시장 현황 및 시사점. KISDI 정책자료 제23권 9호.
- [2] CISCO (2011). 2011 Cisco Visual Networking Index, Cisco.
- [3] 윤천균, 정일용 (2003). SNMP PDU의 시간변수 추가를 통한 네트워크 모니터링 성능향상에 관한 연구. 한국멀티미디어학회 논문지, 제6권 제7호. 101-111.
- [4] 크레이그헨트/김대영, 김태경 공역 (2003). TCP/IP 네트워크 관리(개정 3판), 한빛미디어.
- [5] 이종일(2005), 실무자를 위한 네트워크 관리의 이해,

진한엠앤비.

- [6] William Stallings (1999) SNMP, SNMPv2, SNMPv3, and RMON 1 and 2, 3rd edition, Addison-Weseley Professional.
- [7] Douglas R. Mauro, Kevin J. Schmidt (2005). Essential SNMP Paperback, 2nd Edition, O'Reilly.
- [8] 김동수, 정태명 (1999). 실시간 네트워크 관리를 위한 SNMP의 확장에 관한 연구, 한국정보처리학회 논문지, 제6권 제3호. 449-458.
- [9] Sang-chul Shin, Seong-jin Ahn, jin-Wook Chung (1997). Design and Implementation of SNMP-based performance parameter extraction system. 1997 Asia-pacific network operations and Management Symposium, 137-146.
- [10] Sang-chul Shin, Seong-jin Ahn, jin-Wook Chung (1997). A new approach to gather network management data periodically. ITC-CSCC '97, 529-532.
- [11] 유승근, 안성진, 정진욱 (1999). SNMP MIB-II를 이용한 인터넷 관리 시스템의 웹 인터페이스 설계 및 구현, 한국정보처리학회 논문지, 제6권 제3호, 699-709.
- [12] 최재원, 고영탁, 김한정, 이광희(2009). 웹 인터페이스를 사용하는 SNMP 기반의 무선센서 네트워크 통합관리, 전자공학회논문지, 제46권 제10호, 1011-1019.

윤 천 균



- 1982년 : 인하대학교 전자공학과(학사)
- 1997년 : 포항공과대학 정보통신학과(석사)
- 2003년 : 조선대학교 전자계산학과(박사)
- 1982년 ~ 1998년 : (주)POSCO(과장)
- 1998년 ~ 현재 : 호남대학교 인터넷

콘텐츠학과 부교수

- 관심분야 : Network, 정보가전, 공장자동화
- E-Mail : chqyoun@honam.ac.kr