

# 단순 네트워크 관리 시스템의 설계 및 구현

최 소 영\* · 임 재 홍\*\*

## Design and Implementation of a Simple Network Management System

So-Young Choi, Jae-Hong Yim

### Abstract

Recently, network management application based on the SNMP(Simple Network Management Protocol), CMIP(Common Management Information Protocol) or CMOT(CMIP over TCP/IP) is used for network monitoring. However, network equipment is required to be installed with these protocols, so it costs very expensive and has the defects of user-unfriendly interface.

In this paper, a model of a simple dual network to reduce the system failure time is proposed, and an implementation of a simple network management system for monitoring the system resources on the network with low cost and ease is described. For this, monitoring protocol and algorithm is implemented and tested on the dual network. The result shows that the network status is easily detected on several network fault events.

## 1. 서론

최근 들어 사회적 요구의 다양화 및 기술의 발전에 따라 네트워크의 구조 및 구성요소 등이 구조적인 측면이나 서비스적인 측면에서 양적, 질적으로 팽창하고 있다. 이에 따라 통신사업자는 다양한 종류의 시스템을 수용해야 할 뿐 아니라 다양한 서비스를 요구하는 사용자의 요구를 충족시키기 위해 기존의 음성위주의 서비스를 탈피하여 멀티미디어, 초고속 및 이동체 통신 등의 여러 가지 서비스를 제공하여야 한다. 이러한 네트워크 환경의 변화는 단순한 네트워크 관리 개념에서 벗어나 단위 네트워크 구성요소의 관리 및 서비스 관리 차원까지 포함된 네트워크 관리 개념으로 확대되고 있으며, 또한 네트워크 관리를 위한 기술적 배경도 객체 지향형 및 분산 관리 형태로 진화되고 있는 실정이다[1]. 급변하는 정보처리 환경으로 인한 네트워크 상의 자원들의 구성은 수시로 변경될 수 있고 따라서 분산 컴퓨팅 환경을 원활히 운영하기 위해서는 지속적인 네트워크의 관

\* 한국해양대학교 대학원 전자통신공학과

\*\* 한국해양대학교 전자통신공학과 조교수

리를 통한 시스템 자원의 최적 배치로 네트워크 트래픽을 분산하여 병목현상을 방지하여야 한다. 또한 시스템 자원의 운영 상태를 수시로 모니터링하여 시스템 장애시간(failure time)을 최소화 하여야 한다.

본 논문에서는 네트워크 장애시간을 최소화 하기 위하여 간단한 이중 네트워크(dual network) 모델을 제시하고 적은 비용으로 쉽게 네트워크 상의 시스템 자원들을 모니터링하기 위한 간단한 네트워크 관리 시스템의 설계 및 구현에 관하여 논한다. 이를 위하여 관리 프로토콜 및 알고리즘의 설계 및 구현, 그리고 실제 구성된 이중 네트워크 상에서 다양한 네트워크 장애의 경우, 즉 한 시스템이 다운되는 경우, 두 개의 네트워크 어댑터 카드를 장착한 한 시스템에서 하나의 네트워크 어댑터 카드에 장애가 발생하는 경우, 허브나 라우터에 장애가 발생하는 경우 등에 대하여 시험한 결과 관리 시스템이 정상적으로 자원의 상태를 표시하여 한 눈에 장애가 발생한 위치를 파악할 수 있었다

본 논문의 2장에서는 일반적인 네트워크 관리의 개요에 대하여 논하고, 3장에서는 네트워크 관리를 위한 화면 및 시스템 설계에 대하여, 4장에서는 이중 네트워크의 구성 및 관리 시스템 구현에 관하여 논한다. 마지막으로 5장에서는 결론을 서술한다.

## 2. 네트워크 관리의 개요

네트워크 관리란 기본적인 네트워크 기능의 작동을 초기화하고 감시하고 갱신하도록 하는 것을 말한다. 여기에서 기본적인 네트워크 기능이란 직접적인 방법으로 사용자의 요구사항을 지원하는 것을 말한다. 예를 들어 사용자가 네트워크에 접속하도록 하거나 사용자간 데이터를 교환하도록 하는 등의 행위로서 설계시에 설계자에 의해서 수행되고 구현되어진다.

네트워크 관리의 5가지 주요영역은 다음과 같다.

- 1) 구성 관리(configuration management) : 네트워크 구성요소를 계획하고 확장하고 구성정보와 문서를 유지한다.
- 2) 성능 관리(performance management) : 시스템의 수행능력을 유지하고 강화하기 위해 네트워크의 활동을 추적하고 감시한다.
- 3) 장애 관리(fault management) : 네트워크 상의 문제를 예방하고 검출하고 해결하는 것이다.
- 4) 계정 관리(accounting management) : 사용자 계정을 신설하고 유지하며 자원에 접근할 수 있는 적당한 권한을 부여한다.
- 5) 보안 관리(security management) : 네트워크에 연결된 장비에서 발견되는 보안이 필요한 정보에 대한 접근을 제어함으로써 그 정보를 보호하는 기능이다[2].

### 2.1 SNMP와 CMIP

인터넷과 OSI(Open System Interconnection) 네트워크 위원회가 네트워크 관리구조를 개발하고 발전시켰으며 이 기관에서 네트워크 요소인 관리대상 자원의 대리자(agent)에 의해 제공되는

관리 정보와 이를 가져오고 조작하는 프로토콜 구조를 정의하였다. 네트워크 위원회는 네트워크 관리를 위한 2개의 표준을 발전시켰는데 인터넷 위원회는 SNMP(Simple Network Management Protocol)를, ISO(International Standard Organization)는 CMIP(Common Management Information Protocol)를 정의했다.

SNMP의 구조모델은 네트워크 관리 스테이션과 네트워크 요소(호스트나 게이트웨이 같은 장치들)사이에 관리 스테이션이 관리 요소들을 제어하고 모니터하는 방식이다. 대리자는 관리스테이션이 요구하는 네트워크 관리 기능을 수행한다. 즉 컴퓨터간, 라우터나 게이트웨이간 또는 터미널 서버나 허브 상의 상태정보들, 다시 말해 이들 시스템 간이나 호스트상의 어떤 특별한 이벤트가 일어났을 때나 현재 시스템에 장착되어 있는 하드웨어, 소프트웨어 또는 사용자나 관리자 정보 등을 관리 스테이션에게 알리는 일을 수행한다. SNMP는 이때 대리자와 관리 스테이션 사이에서 사용하는 프로토콜이다. SNMP 관리 스테이션에서 수행하는 동작에는 get, get-next, set, trap 등이 있다. SNMP 관리 스테이션이 대리자에 요구하는 관리 정보는 관리정보 베이스(MIB: Management Information Base)에 담겨져 있다. MIB는 네트워크상의 시스템에 대한 다양한 형태의 정보, 즉 호스트의 운영체제 버전이라든가 액티브 세션(active session)의 수와 같은 것들을 표시하고 있는 관리 가능한 객체의 집합이다. 다시 말해 MIB는 네트워크 관리에 사용되는 정보의 논리적 저장장소이며 ASN.1(Abstract Syntax Notation One)의 규칙에 따라서 관리정보를 기술해 놓은 데이터 베이스 스키마를 지칭한다. 초기의 MIB I은 114개의 객체로 구성되었고 그 후 실험과 작업을 거쳐서 MIB II에서는 170개의 객체들을 포함하고 모두 아홉 개의 그룹(system, interface, ip, icmp, tcp, udp, egp, snmp, transmission)으로 구성되었다[3].

OSI 관리모델은 시스템 관리 엔티티(entity) 사이의 관계설정에 의해 분산 분배방식으로 관리 엔티티를 수행하는 관리 애플리케이션의 측면에서 정의되었는데 관리 스테이션과 대리자 사이의 관리 정보교환에 CMIP 프로토콜을 사용한다. CMIP는 모든 네트워크 환경에 공통으로 적용될 상당히 포괄적인 것을 목표로 하였으므로 설계하는데 긴 시간이 소요되었다. CMIP는 잘 정리되고 훌륭한 개념임에도 불구하고 지나치게 방대하게 규정됨으로써 구현하기가 쉽지 않고 시스템도 SNMP에 비해서 방대하다.

## 3. 시스템 설계

### 3.1. 프로토콜

일반적으로 네트워크의 구성에는 인터넷에서 사용되는 TCP/IP 프로토콜을 주로 사용하고 있다. TCP/IP 프로토콜에는 연결지향의 TCP 프로토콜과 비연결지향의 UDP(User Datagram Protocol) 프로토콜이 제공되는데 본 논문에서는 UDP 프로토콜을 사용하도록 한다.

TCP의 경우는 연결지향이므로 반드시 통신 시스템들 사이에 상호연결이 이루어진 후에 데이터를 전송할 수 있으나, UDP는 비연결지향 프로토콜로써 통신대상 시스템에 대한 가상회선 연결 없이도 데이터를 송수신 할 수 있다[7,8]. 따라서 네트워크 관리 및 모니터링을 위한 프로토콜로써는

UDP가 더 적합하다 할 수 있다. 그림 1은 네트워크의 모니터링을 위한 프로토콜 계층도를 나타낸다.

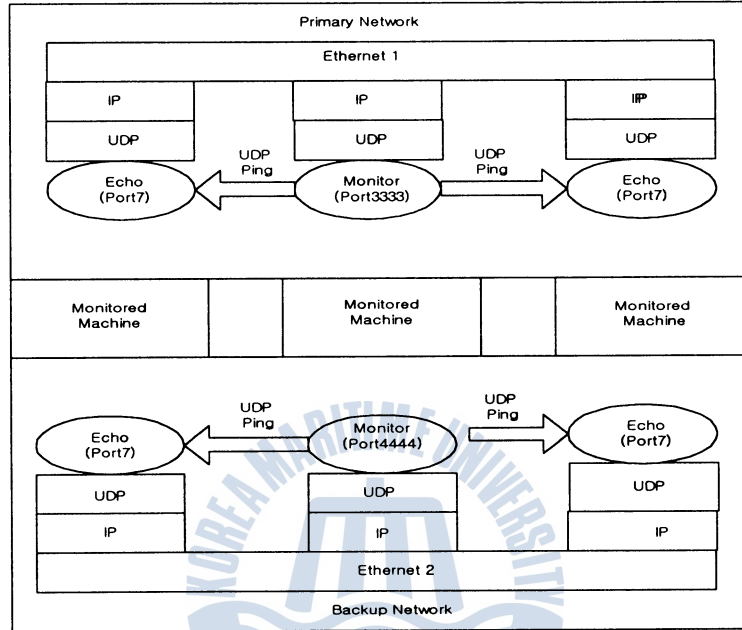


그림 1. 모니터링을 위한 프로토콜 계층도  
 Fig. 1. Protocol hierarchy for monitoring

TCP/IP 서비스에는 UDP 또는 TCP의 포트 7번을 사용하는 에코(echo) 기능이 제공되고 있다. 에코 기능은 단순히 수신한 데이터그램을 송신한 시스템측으로 재전송하는 서비스인데 이를 사용함으로써 시스템의 정상동작 여부를 간단히 점검할 수 있으므로 본 논문에서는 에코 기능을 이용하여 망에 대한 모니터링을 하도록 한다. 에코 기능은 TCP/IP를 탑재한 시스템에선 일반적으로 제공되고 있으며, TCP/IP를 탑재한 라우터에서도 지원되어 운영되므로 네트워크 상의 시스템들과 더불어 라우터의 상태도 모니터링할 수 있다는 장점이 있다. 단 마이크로소프트의 윈도우 NT 운영체제를 사용하는 시스템의 경우에는 단순 TCP/IP 서비스를 설치하여야 에코 기능을 지원하게 된다[9].

### 3.1.1 패킷의 구성

네트워크의 모니터링을 위하여 사용되는 패킷의 형식은 그림 2와 같다.

"HELLO"(6bytes)	시스템 주소	TimeStamp(12byte)
-----------------	--------	-------------------

그림 2. 모니터링 패킷 형식  
 Fig. 2. Monitoring packet format

시스템 주소는 모니터링 대상이 되는 시스템의 IP 주소 혹은 호스트명으로 길이는 가변적이다. 'Time Stamp'는 모니터링 시스템에서 모니터링 패킷을 전송할 시점의 시스템 시간이다. 이를 이용하여 같은 모니터링 대상 시스템에 반복적으로 모니터링 패킷을 전송하더라도 패킷을 유일하게 구분할 수 있다.

### 3.2 화면 설계 및 구성

#### 3.2.1 화면 설계

네트워크 모니터링 시스템은 사용자의 이해와 운영을 쉽게 하기 위하여 마이크로소프트의 윈도우 NT 혹은 윈도우 98을 기반으로 한 GUI 환경에서 구현하도록 한다. 이를 위하여 네트워크 구성도 및 시스템, 네트워크의 상태를 그래픽(메타파일, 비트맵)으로 처리하여 한 눈에 네트워크의 상황을 파악할 수 있도록 한다.

본 논문에서 모니터링하기 위한 네트워크 상의 시스템 자원은 서버, 클라이언트, 라우터가 있으며 네트워크의 구성을 한 눈에 파악할 수 있도록 허브, 프린터, 네트워크를 그림 3과 같이 나타내었다.

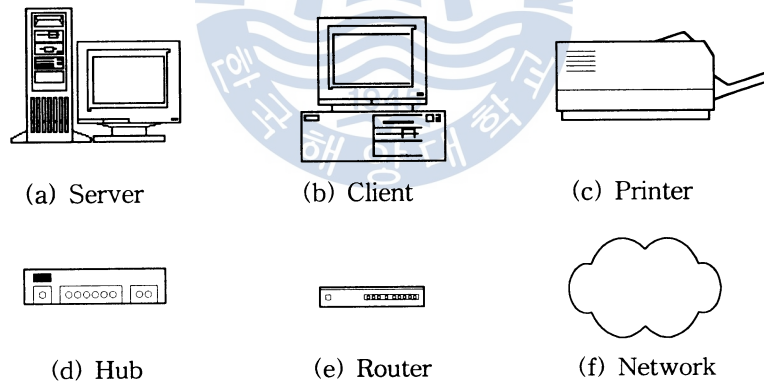


그림 3. 네트워크 자원  
Fig. 3. Network resources

또한 시스템 모니터링의 상태는 '정상', '비정상', '모니터링 중' 세 가지로 구분하고 상태를 나타내는 모양을 그림 4와 같이 나타내었다.

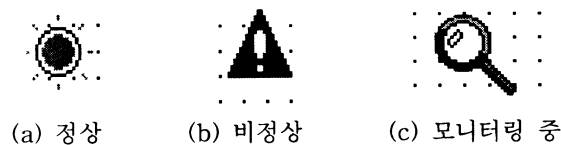


그림 4. 모니터링 상태  
Fig. 4. Monitoring status

### 3.2.2 시스템 자원의 구성

본 논문에서는 모니터링하기 위한 네트워크 상의 각 시스템 자원들을 모두 다음과 같은 속성을 갖는 하나의 객체로 모델링하였다.

- TYPE : 시스템 자원의 형태.
- CAPTION : 시스템 자원의 이름.
- DETAILS : 사용자의 이해를 돕기 위한 설명.
- IP1 : 이중 네트워크에서 주 네트워크에 연결되어 있는 시스템 자원의 IP 주소 혹은 호스트 명.
- IP2 : 백업 연결되어 있는 시스템 자원의 IP 주소 혹은 호스트명.
- NetSTATUS1 : 주 네트워크에 연결된 시스템의 상태.
- NetSTATUS2 : 백업 네트워크에 연결된 시스템의 상태.

표 1은 네트워크 자원의 종류와 속성지원여부를 나타낸다.

표 1. 시스템 속성  
Table 1. System properties

속성 TYPE	CAPTION	DETAILS	IP	상태모니터링
SERVER	○	○	○	○
CLIENT	○	○	○	○
PRINTER	○	○	×	×
HUB	○	○	×	×
ROUTER	○	○	○	○
NETWORK	○	○	×	×

## 4. 시스템 구현

### 4.1 네트워크의 구성

본 논문에서 구성한 실제 네트워크는 그림 5와 같이 이중 네트워크로 구성하였다. 네트워크 장애는 네트워크 어댑터, 네트워크 케이블, 네트워크 중계기(라우터, 허브 등) 등에서 발생의 소지가 많으며, 네트워크 장애시 발생부위를 찾아내기 위해서는 많은 시간이 소요된다[10]. 그러므로 네트워크의 장애시간을 최소화하기 위하여 케이블을 복선화하고 각 시스템에 네트워크 어댑터를 2개씩 장착하여 이중 네트워크를 구성하였다. 이때 주의해야 할 사항은 한 시스템에 장착되는 2개의 네트워크 어댑터에 각각 IP 주소가 할당되어야 한다. 각각의 네트워크는 허브를 중심으로 성형 토폴로지로 구성되며 이더넷을 기반으로 한 TCP/IP 프로토콜을 사용한다. 또한 데이터베이스 서비스,

파일 서비스, 프린트 서비스 등을 제공하는 서버는 항상 동작 상태에 있어야 하므로 주 서버와 백업 서버로 서버 시스템 역시 이중화하여 주 서버에 장애가 발생할 경우 백업 서버가 주 서버에서 운영 중이던 프로세스를 인계받아 서버의 역할을 대신 수행하도록 하였다. 시스템의 백업기법에는 장애허용(fault tolerance), 클러스터(cluster), hot stand-by 등 여러 가지가 있다[11]. 그림 5에서 서버는 펜티엄 PC에 윈도우 NT 4.0 서버 운영체제를 사용하고 클라이언트는 펜티엄 PC에 윈도우 NT 4.0 워크스테이션 운영체제를 사용한다.

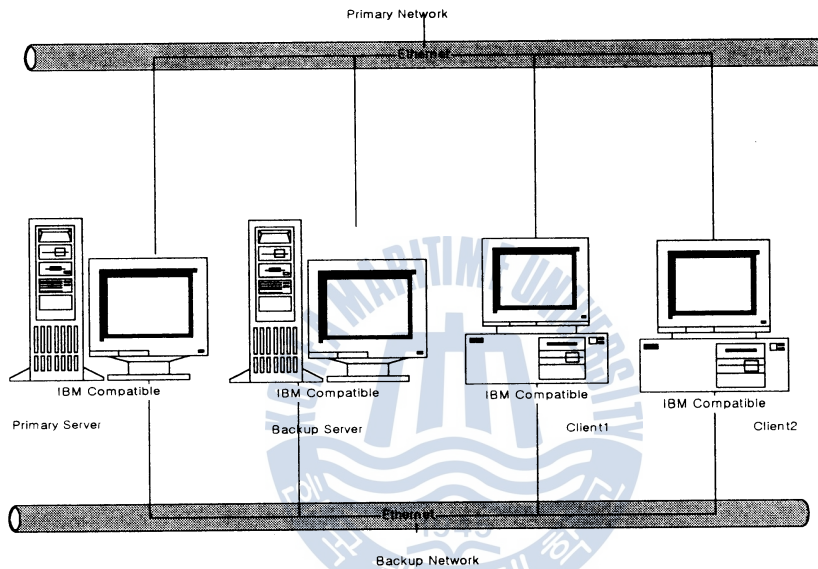


그림 5. 네트워크 구성

Fig. 5. Network configuration

## 4.2 구현

본 논문에서는 GUI 및 프로그래밍 개발 툴로써 비주얼 베이직(Visual Basic) 5.0[12]을 사용하고 비주얼 베이직과 TCP/IP 프로토콜과의 연계를 위해서 Net Manage사의 NEWT 6.02[13]를 사용하여 모니터링 시스템을 구현하였다. NEWT 6.02는 ActiveX 기술을 기반으로 하는 TCP/IP 시스템 개발 키트(SDK: System Development Kit)이다.

일반적으로 네트워크 관리 항목은 구성관리, 장애관리, 성능관리, 계정관리, 보안관리 등 5가지로 구분할 수 있는데, 본 논문에서는 이들 관리 항목 중 구성관리와 장애관리에 대해서만 구현하였다.

### 4.2.1 네트워크 구성관리

네트워크 구성관리는 네트워크 상의 신규, 변경된 자원의 주소관리 및 자원들의 이력을 관리하여 네트워크 구성을 유지, 보수하는데 이용되며, 네트워크 구성도를 나타내는 사용자 인터페이스 화면상의 시스템 자원 위에 마우스 포인터를 올려놓고 오른쪽 버튼을 클릭하면 그림 6과 같은 구

성관리 화면이 나타나 자원의 주소 및 속성을 입력하여 관리할 수 있도록 하였다. 신규 입력 또는 변경된 자원의 주소 및 속성은 'netobj.cfg' 파일에 그림 7과 같은 형식으로 저장되도록 하였다

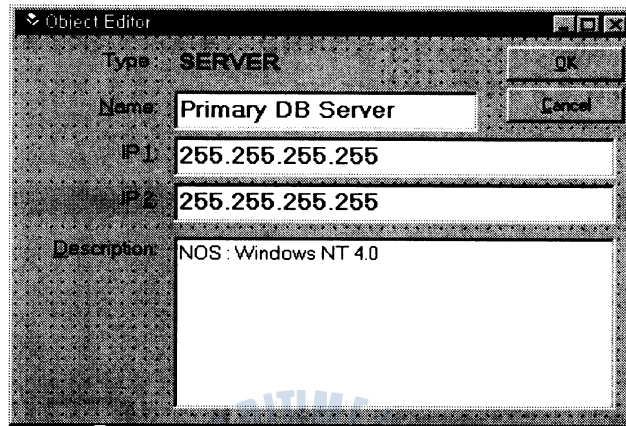


그림 6. 구성관리 화면

Fig. 6. Configuration management window

```

TYPE=SERVER
CAPT=Hanara
DESC=H/W: PentiumPro 200
DESC=NOS: Unix System 5
DESC=DBMS: Sybase 6.0
DESC=Address:
DESC= (Located in Radio Room)
IP_1=hanara.kmaritime.ac.kr
IP_2=203.255.212.4
TYPE=SERVER
CAPT=Hanbada
DESC=H/W: PentiumPro 200
DESC=NOS: Unix system 5 relese 4.0
DESC=DBMS: Sybase 6.0
DESC=Address:
DESC= (Located in Deck Office)
IP_1=hanbada.kmaritime.ac.kr
IP_2=203.255.212.10
TYPE=CLIENT
CAPT=DCN3
DESC=H/W: PentiumPro 200
DESC=NOS: Windows NT 4.0
DESC=Application: Noon Report
DESC=Address:
IP_1=dcn3.kmaritime.ac.kr
IP_2=203.230.253.97
TYPE=ROUTER
CAPT=Cisco OK
DESC=H/W: PentiumPro 200
DESC=NOS:
DESC=Application: Engine Report
DESC=Address:cnt2.kmaritime.co.kr
IP_1=203.255.215.1
IP_2=203.255.215.1
    
```

그림 7. Netobj.cfg 파일

Fig. 7. Netobj.cfg file



#### 4.2.2 네트워크 장애 관리

네트워크 장애관리는 네트워크의 장애를 미연에 방지하기 위하여 정기적인 자원의 모니터링을 수행하여 자원의 장애가 감지되는 경우 즉각 장애복구를 위한 조치를 취할 수 있도록 한다. 모니터링 기능 설정을 위한 화면은 그림 8과 같다. 그림 8에서 'Local IP'는 네트워크 모니터링 시스템이 설치되어 있는 시스템의 IP 주소를 나타내며, 'Interval'은 모니터링을 반복적으로 수행할 시간 간격을 초로 설정한다. 'Network Monitoring' 체크 박스는 모니터링 기능의 시작 또는 정지를 수행한다.

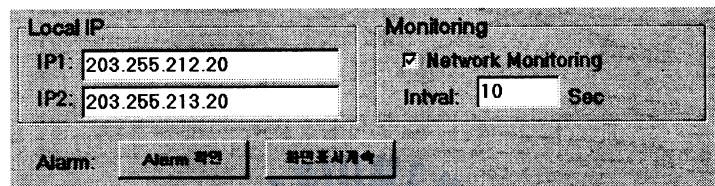


그림 8. 모니터링 기능 설정 화면

Fig. 8. Monitoring function setting window

##### 4.2.2.1 시스템 자원의 알람 기능

본 논문에서는 화면상의 시스템 자원이 비정상인 상태가 되었을 때 시스템 자체 즉 스테이션과 시스템 모니터링의 상태를 알려주는 상태 아이콘을 깜박이게 함으로써 관리자가 한눈에 시스템 자원의 장애를 감지할 수 있도록 하였다. 이 시스템 자원을 깜박이게 하기 위하여 타이머 객체를 이용하였다.

그림 9와 그림 10은 시스템 자원의 알람 프로그래밍의 알고리즘과 네트워크 상태 아이콘의 알람 알고리즘을 보여준다.

그림 9에서의 알고리즘은 네트워크 상의 어느 한 스테이션에서 송신한 패킷이 에코되어 돌아오지 않았을 때 그 시스템의 이미지가 깜빡이도록 한 흐름도이다. 각 스테이션은 정상상태를 나타내는 이미지와 비정상 상태를 나타내는 두 가지 이미지를 가지고 있다. 시스템이 모니터링에 실패하였을 경우 두 가지 이미지가 몇 초의 간격을 두고 화면상에 적재되면서 스테이션이 깜빡이는 효과를 내도록 하였다. 이 때 스테이션이 모니터링에 실패하였다는 것을 알리는 경고음을 내도록 하여 관리자가 이를 즉각 감지할 수 있도록 하였다.

마찬가지로 그림 10의 알고리즘에서 보면 'NetStatus1'은 주 네트워크의 상태를 나타내는 변수이고 'NetStatus2'는 백업 네트워크의 상태를 나타내는 변수이다. 어떤 스테이션에서 이들 변수가 비정상 상태를 나타낼 때 네트워크 상태 아이콘 중 비정상 상태를 나타내는 아이콘과 무 이미지의 아이콘이 화면상에 교체되어 적재되도록 하였다. 이 때에도 역시 경고음을 내도록 하여 관리자가 네트워크 상에 문제가 발생했음을 감지할 수 있도록 하였다.

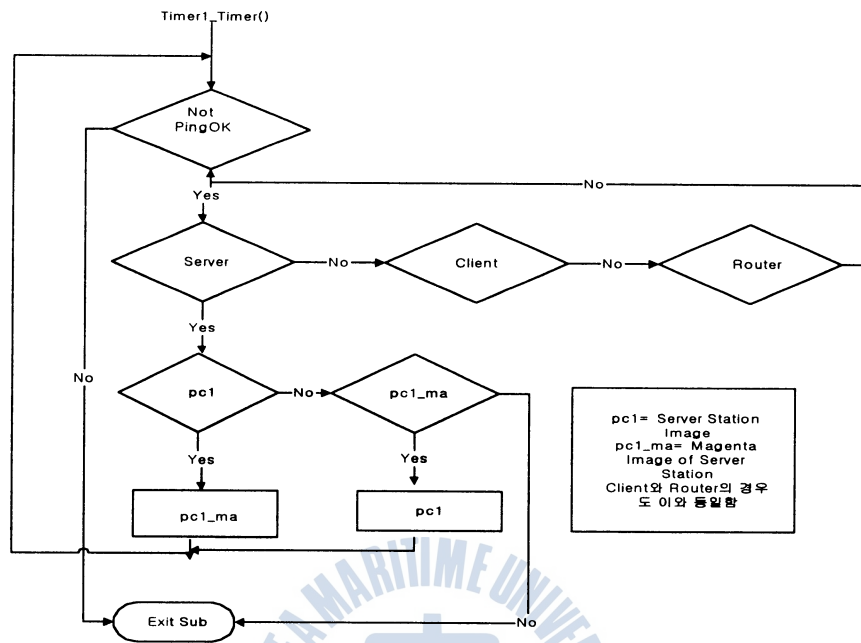


그림 9. 시스템 알람 알고리즘  
Fig. 9. System alarm algorithm

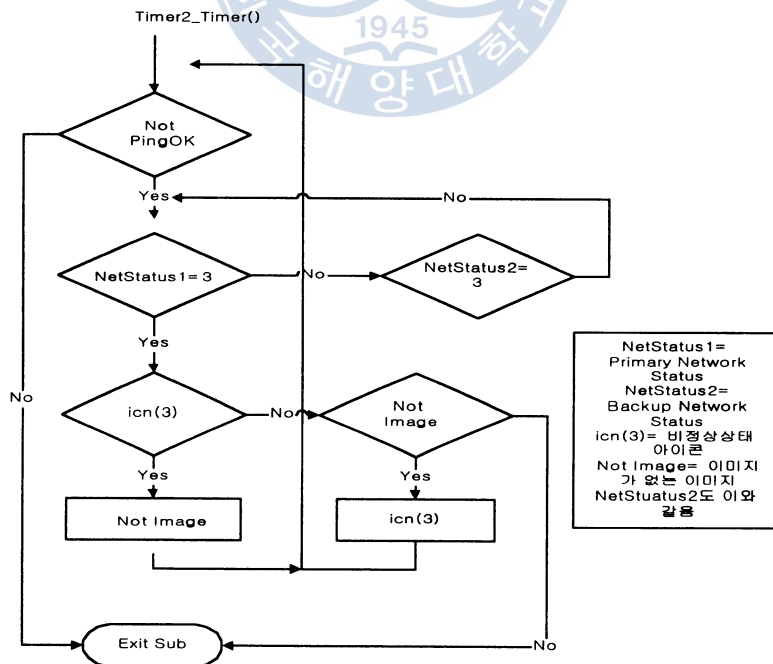


그림 10. 아이콘 알람 알고리즘  
Fig. 10. Icon alarm algorithm

### 4.2.3 모니터링 알고리즘

네트워크 모니터링을 위한 알고리즘을 그림 11에 나타내었다. 타이머에 의해서 모니터링 시간간격마다 Trigger\_Timer 서브루틴이 실행되며 초기의 상태는 '모니터링 중', 즉 NetStatus = 0 이다. 그리고 모니터링 시스템은 모니터링 하고자 하는 시스템에 'TOKEN\$'이라는 모니터링용 데이터를 송신하여 그 시스템으로부터 데이터가 에코되어 수신되기를 'ExpiredTime'까지 기다린다. 이때 UDP 포트로 어떠한 데이터가 수신되면 UDPI\_DataArrival() 서브루틴이 이벤트 처리기로서 실행된다. 만일 수신된 데이터가 'TOKEN\$'이면 시스템이 정상적임을 나타내므로 PINGOK=True 가 되고, NetStatus = 2, 즉 정상상태가 된다. 반면 수신된 데이터가 없을 경우에는 'ExpiredTime' 까지 다시 반복하여 기다리고 'ExpiredTime' 을 초과하면 NetStatus = 3, 즉 비정상 상태가 된다.

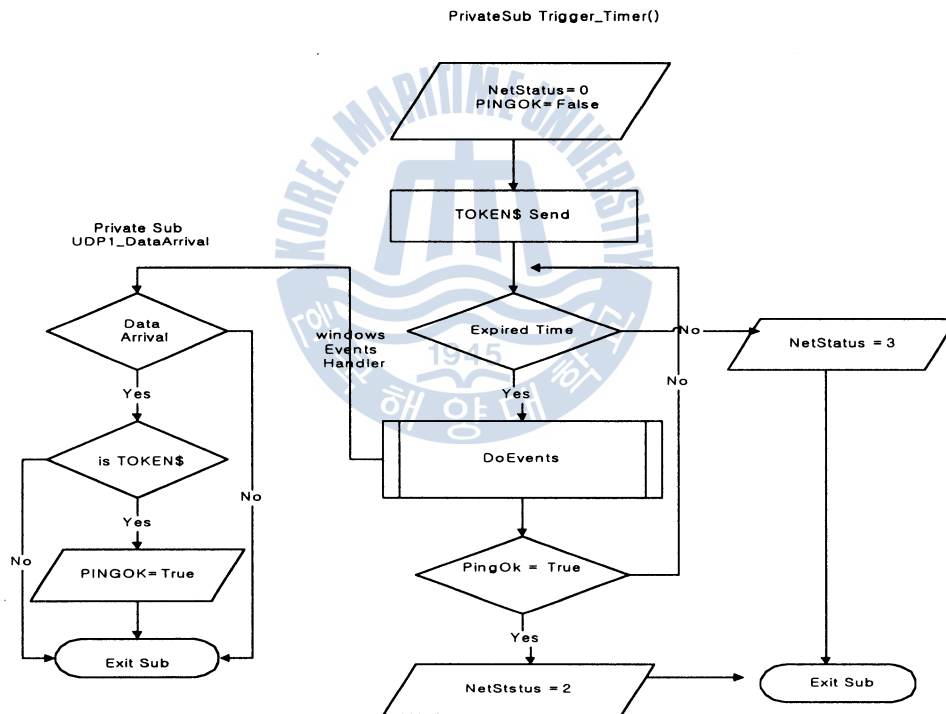


그림 11. 모니터링 알고리즘  
Fig. 11. Monitoring algorithm

### 4.3 시험 및 고찰

본 논문에서 구현한 네트워크 모니터링 시스템의 사용자 인터페이스 화면은 그림 12와 같다. 그림 12에서와 같이 모니터링 시스템의 동작 시험을 위하여 간단한 이중 네트워크를 구성하고 다양한 네트워크 장애의 경우에 대하여 모니터링 시험을 하였다. 한 시스템이 다운되는 경우, 두 개의

네트워크 어댑터 카드를 장착한 한 시스템에서 하나의 네트워크 어댑터 카드에 장애가 발생하는 경우, 허브나 라우터에 장애가 발생하는 경우 등에 대하여 시험한 결과 관리 시스템이 정상적으로 자원의 상태를 표시하여 한 눈에 장애가 발생한 위치를 파악할 수 있었다.

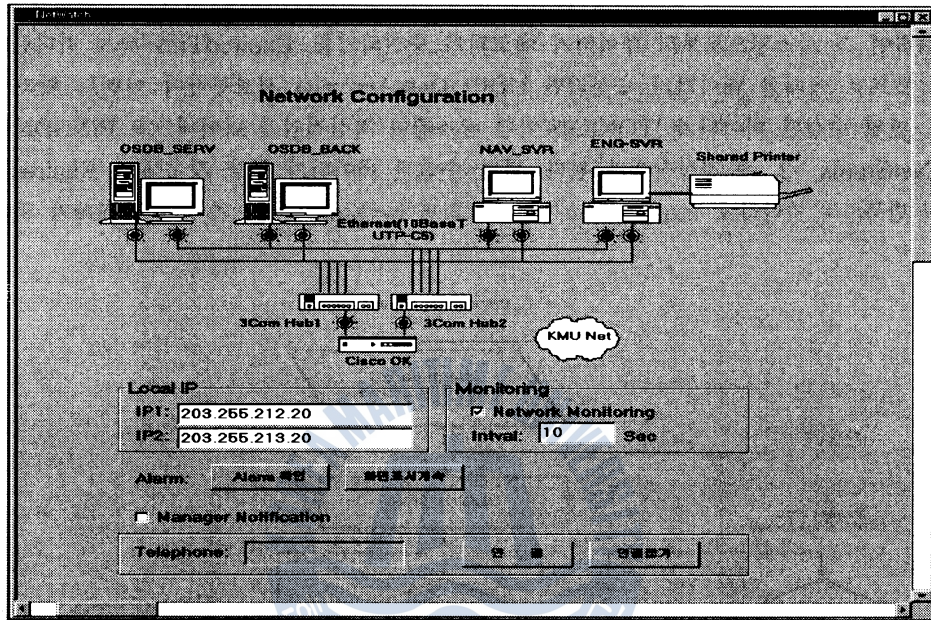


그림 12. 네트워크 모니터링 시스템  
Fig. 12. Network monitoring System

SNMP, CMIP, CMOT 등의 네트워크 관리용 응용을 사용하는 경우는 별도의 프로토콜을 시스템마다 인스톨해야하고 사용하기 어려운 반면, 본 논문에서 구현한 네트워크 모니터링 시스템은 적은 비용으로 사용자가 쉽게 사용할 수 있다는 장점이 있다. 하지만 본 논문의 모니터링 시스템은 UDP/IP의 에코 기능을 이용하므로 이러한 기능을 지원하지 않는 시스템은 모니터링이 불가능하다는 문제점이 있다. 즉 허브와 같은 네트워크 장비는 TCP/IP를 지원하지 않으므로 허브 자체의 상태를 모니터링하는 것은 불가능하며, 다른 자원의 상태로 허브의 상태를 유추하는 수밖에 없다.

## 5. 결 론

현재 네트워크 모니터링을 위해서는 SNMP, CMIP, CMOT를 기반으로 하는 네트워크 관리용 응용이 이용되고 있으나 이를 위해서는 네트워크 장비에 별도의 프로토콜을 장착해야 하므로 많은 비용을 부담하여야 하며, 사용하기 어려운 단점이 있다. 본 논문에서는 네트워크 장애시간을 최소화하기 위한 간단한 이중망 모델을 제시하고 적은 비용으로 사용자가 쉽게 사용할 수 있는 GUI 환경으로 네트워크상의 시스템 자원들을 모니터링하기 위한 단순 네트워크 모니터링 시스템에 관

하여 기술하였다. 이를 위하여 모니터링 프로토콜 및 알고리즘을 설계, 구현하고 실제 구성된 이종 망에서 다양한 네트워크 장애의 경우, 즉 한 시스템이 다운되거나 두 개의 네트워크 어댑터 카드를 장착한 한 시스템에서 하나의 네트워크 어댑터 카드에 장애가 발생하는 경우, 허브나 라우터에 장애가 발생하는 경우 등에 대하여 시험한 결과 관리 시스템이 정상적으로 자원의 상태를 표시하여 한 눈에 장애가 발생한 위치를 파악할 수 있었다. 하지만 본 논문의 모니터링 시스템은 UDP/IP 에코 기능을 이용하므로 이러한 기능을 지원하지 않는 시스템은 모니터링이 불가능하다는 문제점이 있다. 즉 허브와 같은 네트워크 장비는 TCP/IP를 지원하지 않으므로 허브 자체의 상태를 모니터링하는 것은 불가능하여, 다른 자원의 상태로 허브의 상태를 유추하는 수밖에 없다.

향후 연구로는 허브와 같이 TCP/IP를 지원하지 않는 네트워크 장비에 대한 모니터링 방법과 본 논문의 모니터링 시스템 기능을 확장하여 SNMP, CMIP, CMOT 등을 활용할 수 있는 방안이 추진되어야 할 것이다.

## 참고문헌

- [1] 박남영, 김성범, 통신관리망(TMN) 표준화 현황 및 추진방향, 한국통신 통신망 연구소 TMN 연구실, 1998
- [2] Microsoft, Networking Essentials, Microsoft Press, pp. 435 - 436, 1996
- [3] Microsoft, Internetworking with Microsoft TCP/IP on Microsoft Windows NT 4.0, Microsoft press, pp. 317 - 331, 1996
- [4] [http://my.netian.com/~vs909/nms/cmip\\_cmi/cmip\\_cmi.htm](http://my.netian.com/~vs909/nms/cmip_cmi/cmip_cmi.htm), CMIS/CMIP
- [5] Juergen Scoenwalder, Scotty-Tcl Extension for Network Management Application, July, 1995
- [6] Advent Network Management Inc. Advent Web NMS Architecture, Advent Network Management Inc, 1996
- [7] W.Richard Stevens, Unix Network Programming, Prentice Hall, 1990
- [8] 김성호(편역), 이기종 접속과 TCP/IP, 도서 출판 동서, 1995
- [9] Microsoft, Windows NT Server Installation Guide, Microsoft Press, 1997
- [10] EVI Nemeth, Garth Snyder, Scott Seebass, Trent R. Heing, Unix System Administration HandBook 2nd Edition, Prentice Hall, 1995
- [11] Scottn H Davis, Clusters for Windows NT, Microsoft Developer's Network Library, 1996
- [12] 주경민, 박성완, 정동길 공저, Visual Basic Programming Bible ver5, 영진 출판사, 1998
- [13] NET-SDK TCP/IP Application for Windows Developer's Guide, Netmanage, 1996
- [14] 하이테크 정보, '95 LAN, 하이테크 정보 출판부, 1995

