논문 2011-1-15

자동 망 구성 기능을 갖는 네크워크 모니터링 시스템

A Network Monitoring System with Automatic Network Configuration

정인환*

In-Hwan Jung

요 약 본 논문에서는 효율적인 망 관리를 위해 네트워크 감시 대상 노드들과 망 구성에 대한 정보를 자동적으로 구축하는 네트워크 모니터링 시스템을 설계하고 구현하였다. 구현된 모니터링 시스템은 ARP 동보 패킷을 이용하여 노드들을 자동 인식하며 인식된 노드들에 대하여 NETBIOS 이름 확인 절차와 ICMP 기능을 통해 수집된 정보를 바탕으로 해당 서브넷의 망 구성 정보를 자동적으로 생성하고 이를 보여준다. 또한 각 노드들에 대하여 포트 스캐닝 감시를 통하여 침입을 탐지할 수 있는 실시간 감시 기능을 갖는다. 구현된 모니터링 시스템을 이용하면 자동 노드 인식이 가능함으로써 망 관리를 위해 망에 연결된 노드들의 주소를 관리해야하는 부담을 해소시켜주며 이를 통해 효율적인 망 관리를 수행 할 수 있도록 도와준다.

Abstract In this paper we describe an efficient and easy to use network monitoring system which can identify network configuration automatically by means of capturing and analyzing the ARP broadcasting packets. After identifying network nodes, it gathers detail information of each node such as NETBIOS name and number of hop counts using ICMP and then shows subnet configuration with graphical method. This monitoring system also has a subset of intrusion detection system that can monitor any port scanning trial. With this automatic network configuration functions, it helps to lessen address keeping track overhead which is crucial for network monitoring so that it provides efficient network management.

Key Words : ARP, Automatic Network Configuration, NETBIOS, ICMP, Network Monitoring.

Ⅰ. 서 론

네트워크 모니터링 시스템은 감시할 대상인 네트워크 노드들의 IP 주소를 목록으로 가지고 있으면서 지속적으로 노드들에 대한 네트워크 사용을 모니터링한다. 네트워크 모니터링 시스템에서 감시대상 노드들에 대한 정보는 관리자가 수동으로 입력하는 것이 일반적이다. 이 경우 관리자는 네트워크의 전체 노드들에 대한 정보를 가지고 있어야 하므로 네트워크의 상황 변화에 실시간으로 대처할 수 없는 문제점이 있다. 본 논문에서는 이러한 문제점을 해결하는 방법으로 자동 망 구성 기능을 갖는 네트워크 모니터링 시스템을 설계하고 구현하였다. 구현된 모니터링 시스템에서는 실시간으로 수집된 ARP 동보 패킷을 통하여 감시 대상 망에 연결된 노드의 IP 주소를 인식한다. ARP[2] 동보 패킷은 이더넷[1]에서 네트워크 노드들이 네트워크에 패킷을 전송하기 위해서는 필수적으로 선행되어야하는 패킷이며 이 패킷에는 해당 노드의 이더넷 주소와 IP 주소가 담겨있다. 또한 ARP 동보 패킷은 공유허브(shared hub) 뿐만 아니라 스위칭 허브(switching hub)를 통해서도 모든 노드들에게 전달되므로 모니터링 시스템이 동작하는 PC에서도 해당 ARP 패

*정회원, 한성대학교 컴퓨터공학과 교수
접수일자: 2011.1.25 수정일자: 2011.2.10
게재확정일자: 2011.2.11

킷을 수신할 수 있다. 따라서 ARP 동보 패킷을 이용하면 네트워크에 연결되어 있으면서 네트워크 활동이 있는 모든 노드들을 인식하고 이를 이용하여 네트워크 구성 정보를 생성할 수 있다.

본 모니터링 시스템은 ARP 패킷을 이용하여 노드들을 인식한 후 NETBIOS 프로토콜을 이용하여 해당 컴퓨터의 이름, 즉NetBIOS 이름을 확인하고 ICMP 프로토콜을 이용하여 노드를 실시간 감시한다.

구현된 네트워크 모니터링 시스템은 노드 정보를 실시간으로 표시하는 기능, 네트워크 서브넷 정보를 자동적으로 도식화하는 기능, 침입탐지 기능 중의 하나인 포트 스캐닝 탐지 기능 그리고 SMS 경고 메시지 전달 기능 등의 주요 기능을 가지고 있다.

구현된 네트워크 모니터링 시스템은 자동으로 망에 연결된 노드들을 인식하고 망 구성을 함으로써 망 관리의 부담을 줄여주고 망 구성의 변화에 효과적으로 대치할 수 있는 도구로 활용될 수 있다.

본 논문의 구성은 다음과 같다. 2장에서는 관련연구로 본 논문에서 사용한 ARP 프로토콜과 패킷 드라이버를 소개한다. 3장에서는 제안된 네트워크 모니터링 시스템의 설계 및 구현에 대해 기술하고 4장에서는 네트워크 모니터링 시스템의 실행 결과 화면을 설명한다. 5장에서는 자동 노드 인식 실험 및 결과에 대해 기술한다. 마지막으로 6장에서 결론 및 향후 연구에 대하여 기술한다.

II. 관련연구

1. ARP 프로토콜

이더넷(Ethernet)[1] 환경에서 어떤 컴퓨터가 다른 호스트로 네트워크 연결을 하기 위해서는 상대방 호스트의 이더넷 주소를 알아야 한다. 즉, 사용자는 IP 주소를 기반으로 연결을 하지만 이더넷 상에서는 이더넷 주소를 이용하게 된다. 이때 사용하는 프로토콜이 ARP(Address Resolution Protocol)[2]이다.

네트워크상에 모든 호스트에 ARP request라고 불리는 이더넷 프레임을 보낸다. 연결하고자 하는 호스트의 IP 주소를 포함한 ARP request는 이더넷상의 모든 호스트들에게 전송한다. ARP request를 받은 각 호스트는 자신의 IP 주소와 비교하여 해당 IP 주소를 사용하는 호스트는 자신의 하드웨어 주소(이더넷 주소)를 ARP Sender 에게 보내주게 되는데 이를 ARP reply라고 한다. 이렇게 얻어진 상대방의 MAC(Mediaum Access Control) 주소를 사용하여 통신이 가능하게 된다. 그림 1은 ARP 패킷의 구조를 보여주며 그림 2는 ARP Request/Reply 메시지 처리 예를 보여준다.

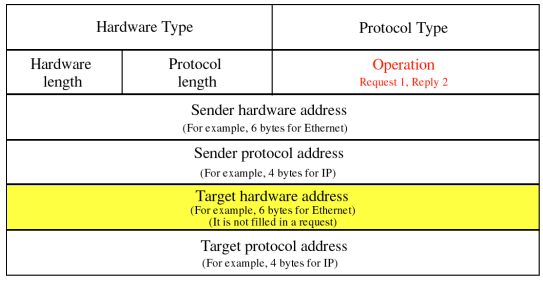


그림 1. ARP 패킷 구조
Fig. 1. ARP packet format

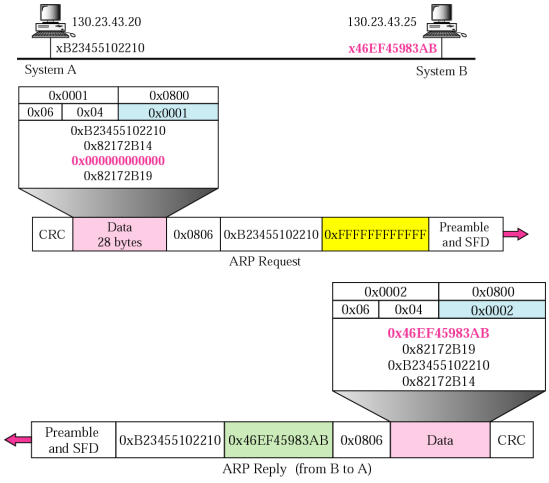


그림 2. ARP Requst 와 Reply 예
Fig. 2. ARP Request and Reply example

어떤 IP 서브넷에 속한 컴퓨터는 인터넷과 같은 외부와 통신을 위해서는 라우터를 통해야 한다. 이 때 그 컴퓨터는 IP 패킷에 이더넷 헤더를 붙이면서 목적지 이더넷 주소를 라우터의 MAC 주로로 채우게 된다. 그렇게 되면 그 패킷은 라우터가 읽어가게 되어 결국 라우팅에 의한 최종 목적지까지 전달이 이루어지는 것이다. 이것이 IP 라우팅의 원리이다. 그러므로 어떤 컴퓨터든지 라우터에 대한 ARP Request/Reply 절차가 선행된다. 각각의 노드들이 라우터의 MAC 주소를 알기 위해서는 ARP Request가 발송되며 이 ARP Request에는 해당 노드의

IP 주소와 MAC 주소가 들어있다. 또 ARP Reply에는 송, 수신 노드들의 IP 주소와 MAC 주소가 들어 있으므로 이를 이용하면 망에 연결된 노드들의 정보를 파악할 수 있다.

본 논문에서는 네트워크에 연결된 노드들이 네트워크를 사용하려면 선행적으로 발생시키는 ARP 동보 패킷을 수집하고 분석함으로써 활동중인 노드들을 인식하였다.

2. 패킷 캡쳐 드라이버

본 논문에서는 네트워크에 전달되는 모든 패킷을 수집하기 위하여 Winpcap 패킷 드라이버[3]를 이용하였다. Winpcap은 libpcap[4]와 호환성을 갖는 Windows 기반 패킷 캡쳐 라이브러리다. 일반적으로 이더넷 환경에서 네트워크 카드는 네트워크에 오고가는 모든 패킷을 수신하지만 이더넷 주소가 자신의 것이 아니면 폐기하게 된다. 그런데 네트워크 모니터링을 위해서는 자신의 이더넷 주소가 아닌 패킷도 모두 수신하여 처리할 수 있어야 한다. Winpcap과 같은 패킷 캡쳐 드라이버를 이용하면 자신의 이더넷 주소가 아닌 패킷을 포함한 네트워크 카드로 들어오는 모든 패킷을 수집하여 응용프로그램으로 전달해 준다. 본 논문에서 구현한 네트워크 모니터링 시스템은 그림 3과 같으며 패킷 드라이버는 랜 카드와 LAN 카드 장치 드라이버 사이에 있으면서 랜카드로 들어오고 나가는 모든 패킷을 복사해서 링 버퍼를 통해 모니터링 프로그램으로 전달해준다.

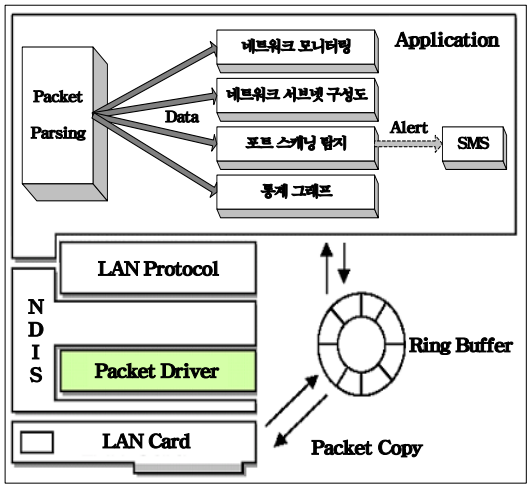


그림 3. 네트워크 모니터링 시스템 구성도
Fig. 3. Network Monitoring System configuratrion

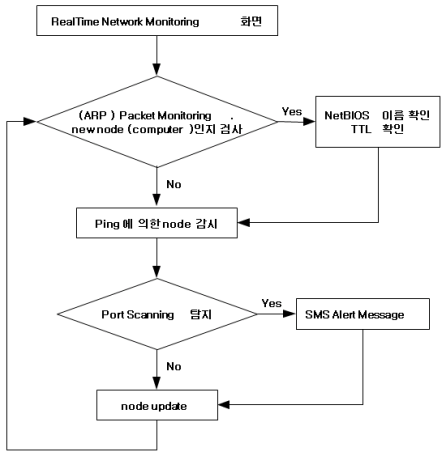


그림 4. 프로그램 흐름도
Fig. 4. Program flow chart

III. 설계 및 구현

1. 프로그램 구성과 기능 설계

본 논문에서 구현된 네트워크 모니터링 프로그램은 그림 3과 같이 네트워크 드라이버 계층과 응용프로그램 계층으로 구성으로 되어있다. 네트워크 계층의 패킷 드라이버는 랜카드를 통해 오고 가는 모든 패킷을 복사하여 원형 버퍼를 통해 모니터링 프로그램으로 패킷을 넘겨준다. 응용계층에 구현된 모니터링 시스템은 패킷 드라이버로부터 실시간으로 수집되는 패킷을 분석하여 그림 4와 같은 흐름으로 다음과 같은 기능을 수행하도록 설계하였다.

- 노드의 IP를 자동 인식하여 표시
- NetBIOS를 이용하여 컴퓨터 이름 표시
- 주기적인 PING을 통한 노드 상태 검사
- 포트스캔 실시간 감시 및 SMS 전송
- 서브넷 정보 자동 구성

가. 자동 노드 인식

Winpcap 을 통해 수신된 패킷이 ARP Reuqest 패킷이라면 ARP 헤더의 <Sender Hardware Address, Sender Protocol Address> 를 <MAC 주소, IP 주소>로 해석한다. 만약 새로운 노드라면 NetBIOS 이름 확인 절차를 거친후 PING 감시 및 Port Scanning 감시 대상으로 리스트에 추가한다. 수신된 패킷이 ARP Reply라면

<Sender/Target> 모든 주소에 대해 새로운 노드인지 검사 절차를 거쳐 새로운 노드 리스트에 추가한다. 마약 수신된 Packet이 ARP 패킷이 아닌 일반적인 IP 패킷이라면 IP 패킷의 <Src IP, Dst IP>를 대상으로 <Src Mac Address, Dst Mac Address>에 대하여 새로운 노드인지 판단한다. 네트워크 모니터링 시스템이 이미 활발하게 통신 중인 노드들에 대하여 감시를 시작한다면 ARP 메시지 대신 IP 패킷에 대하여 노드 인식 기능을 수행할 필요가 있다.

나. NetBIOS 이름 확인

일반적으로 PC들을 네트워크에 연결하면 컴퓨터 등록정보에 컴퓨터 이름을 설정하고 이 컴퓨터 이름은 네트워크 상에서 NetBIOS[5] name service로 확인이 되며 이는 망 관리에서 중요한 정보의 하나다. 본 논문에서 구현된 모니터링 시스템에서는 자동 인식된 노드들에 대하여 NetBIOS name query 기능을 실행하여 노드들의 이름을 자동적으로 확인하는 기능을 갖도록 하였다. NetBIOS 이름 확인 절차는 NCB(Network Control Block) 헤더를 정의한 구조체 변수에 명령어를 설정하고 Netbios() API 함수를 호출한다.

다. PING을 통한 노드 상태 검사 및 TTL 표시

만약 ARP로 발견한 IP 주소가 이미 존재하는 IP주소라면 PING을 이용한 노드 감시를 실행하여 그 노드가 살아있는지 주기적으로 검사하게 된다. PING 명령어는 ICMP(Internet Control Message Protocol)[7]의 Echo Request/Reply 를 통해 해당 노드 또는 컴퓨터가 살아있는지 검사한다. 본 논문에서는 자동 인식된 노드들에 대하여 주기적으로 ICMP 메시지를 보내고 확인하여 그 노드가 정상적으로 응답하는지 검사한다. 이를 위해 Windows의 icmp.dll API를 사용한다. Ping 검사의 결과로 수신한 패킷을 분석하여 모니터링 컴퓨터와 해당 Node 사이의 Hop Count 또는 TTL 값을 구해서 표시하고 이를 기반으로 네트워크 구성 정보를 생성한다.

라. 포트스캔 실시간 감시 및 SMS 전송

자동으로 인식된 노드들에 대하여 Ping 검사를 실시한 다음. 해당 노드에 대하여 Port Scanning과 같은 악의적 시도가 있을 경우 이를 감지하여 필요하다면 SMS 를 통해 관리자에게 통보하는 기능을 수행한다. 포트 스캔을 감지하기 위한 알고리즘은 다음과 같다.

> *수신한 Packet의 <Src IP, Dst IP, Dst Port>를 확인해서 같은 Src IP 에서 같은 Dst IP 로 Port 만 변경해서 2분 안에 10개 이상 반복되는 것이 발견된다면 이를 Port Scanning으로 판단한다.*

만약 위와 같은 검사를 통해 포트 스캔이 감지된다면 .rc IP와 Dst IP 를 갖는 컴퓨터에게는 Windows의 "*Net send*" 명령어를 이용하여 경고 메시지를 전송하고 미리 설정된 SMS 기능을 이용해 관리자에게 SMS 문자를 보내게 된다.

마. 서브넷 정보 자동 구성

각 서브넷에 해당하는 라우터 주소를 얻는 방법은 ICMP Echo Request와 Time Exceed 메시지를 이용하는 TraceRouter 모듈을 이용하여 특정 노드의 IP 주소에 해당되는 라우터 주소를 수집하고 서브넷에 해당하는 라우터 주소와 대응되는 노드들의 IP 주소를 SUBNET_INFO 구조체를 사용하여 저장하였다. 그림 5는 네트워크 서브넷 구성도를 그리기 위하여 사용되어지는 구조체이다.

```
struct SUBNET_INFO
{
  static int SubnetArrayCount;
  int Subnet TTL;
  int SubnetHostNumber;
  int SubnetRouterIP[ADDR_SIZE];
  int SubnetHostIP[MAX_HOST_NUM][ADDR_SIZE];
}
```

그림 5. 서브넷 정보 구조체
Fig. 5. Subnet Information Structructure

SUBNET_INFO 구조체는 하나의 서브넷이 가지고 있어야 할 정보들을 나타내는 구조로서, 구조체 배열로 선언된다. 각 서브넷은 자신의 노드를 기준으로 서브넷 계층을 증가시켜 나가고, 각 서브넷 계층에는 해당하는 라우터 주소를 가지게 된다. 또한 서브넷의 TTL 정보와 각 서브넷이 현재 가지고 있는 노드들의 수와 노드 IP 주소를 저장한다.

2. 화면 구현 및 부가 기능

가. 사용자 화면 구현

그림 6는 구현된 모니터링 프로그램의 화면이다. 모니터링->시작 또는 시작 아이콘을 누르면 모니터링이 시작되고 모니터링->중지 또는 중지 아이콘을 누르면 모니터링이 중지된다.

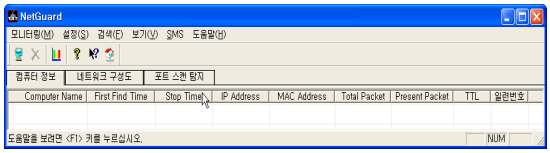


그림 6. 구현된 화면
Fig. 6. User Interface

컴퓨터정보 화면에서는 인식된 노드들의 목록이 표시되고 각각의 노드들에대한 정보가 표시된다. 네트워크구성도 화면에서는 서브넷 정보가 표시된다. 포트 스캔 탐지 화면에서는 포트스캔 감시 결과가 표시된다.

나. SMS 문자 전송 기능

본 논문에서는 SMS를 이용하여 네트워크 관리자가 부재중일 경우라도 네트워크의 부하 또는 침입 시도가 있을 경우에 관리자에게 SMS 메시지를 전송하여 신속히 대응할 수 있도록 하였다. 본 논문에서는 SMS 전송을 위한 프로토콜을 지원하는 SMS 전송 라이브러리인 SMS Sender[9]를 이용하였다. SMS Sender는 WinSender.h, WinSender.dll 그리고 WinSender.lib로 구성되어 있고 표 1은 SMS Sender의 중요 함수들이다.

표 1. SMS Sender API 함수
Table 1. SMS Sender API

SMSSend	문자 전송
SMSJoin	회원가입
SMSExit	회원탈퇴
SMSLogin	로그인
SMSPhoneChange	폰 번호 변경
SMSPassChange	비밀번호 변경

다. 부가 기능

그 밖에 각각의 노드들에 대해 다음과 같은 기능을 구현하였으며 자세한 내용은 다음 장에서 기술한다.

- 노드당 수신된 패킷의 프로토콜 분류
- 노드당 수신된 패킷의 크기별 분류
- 노드에 대한 포트 스캔 검사
- 노드에 대한 Well-Known 포트 검사
- 노드에 메시지 보내기

Ⅳ. 프로그램 실행 시험

1. 자동 노드 인식 기능

그림 7은 모니터링 시작후 10초간 자동 노드 인식을 통해 노드들이 인식되는 화면이며 그림 8은 1분 후의 화면이다. 그림에서 보듯이 시간이 흐름에 따라 노드들이 자동적으로 추가되는 것을 확인할 수 있다.

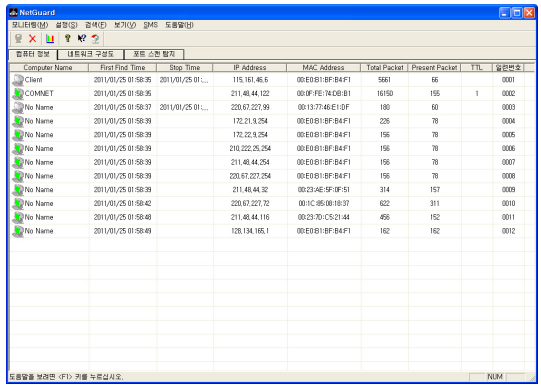


그림 7. 실시간 모니터링 화면 (10초 후)
Fig. 7. Real Time Monitoring Screen (after 10 secs)

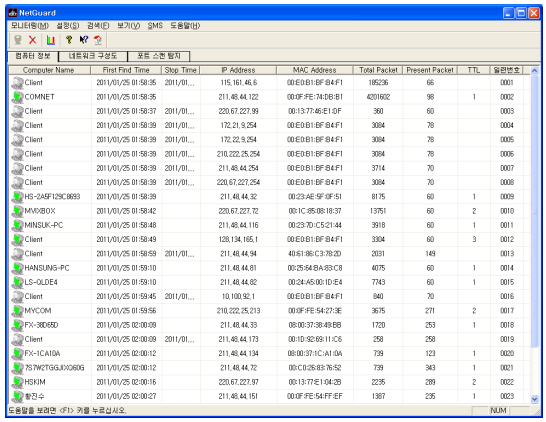


그림 8. 실시간 모니터링 화면 (1분 후)
Fig. 8. Real Time Monitoring Screen (after 1 min)

그림에서 보듯이 실시간 모니터링은 노드 이름, 노드 감지 시간(Node Generated Time), 노드 감지 종료 시간(Node Terminated Time), IP 주소, MAC 주소, 총 데이터 량, 최근 데이터 량 및 TTL 값 등의 정보를 가지고 있다. 노드 이름은 NetBIOS[5] 프로토콜을 통하여 각 노드들의 실제 사용되는 이름을 추출하고, 노드의 네트워크 접속 상태에 따라 아이콘을 다르게 표시하였다. 즉, 녹색은 ICMP Echo Request에 대하여 Reply 응답을 받은 노드들로 네트워크에 접속된 상태를 의미하고 ICMP Echo Reply를 하지 않는 노드들은 비록 ARP 또는 IP 패킷에의해 자동 노드 감지는 되었으나 네트워크에 접속되어 있지 않은 상태인 회색으로 표시하였다.

어떤 노드의 IP는 인식되었으나 TTL 값이 표시되지 못하는 노드들은 대부분 라우터에서 ICMP 메시지 차단을 하는 이유로 TTL 값을 확인할 수 없는 노드들이다. 이러한 노드들의 MAC Address 는 대부분 라우터의 MAC Address로 표시된다.

2. 네트워크 구성도 확인

서브넷 구성도는 실시간 네트워크 모니터링을 통하여 얻어진 정보를 이용하여 사용자가 쉽게 서브넷[6] 별로 망을 인식할 수 있도록 하기 위하여 자동적으로 도식화 한 것이다.

본 논문에서는 모니터링 되고 있는 노드들을 서브넷 별로 그룹을 형성하여 그림을 실시간으로 도식화하는 기능을 구현하였다. 각 서브넷의 계층은 자신의 노드를 기준으로한 TTL값을 적용하여 구분하였고, 각 서브넷은 해당하는 게이트웨이 IP주소와 함께 서브넷에 포함되는 노드들의 IP주소를 묶어서 도식화하였다. 그림 9는 자동적으로 만들어진 네트워크 구성도 화면이다. 그림 9에서 모니터링 PC와 같은 서브넷으로 판단되는 노드들의 IP 주소가 첫줄에 표시되었으며 그 다음 줄에 첫번째 라우터의 주소가 표시되고 있다. 그리고 두번째 라우터 그룹에 속한 IP 주소들이 표시되고 있다. 이 서브넷 정보는 ICMP Echo Request와 Time Exceed 를 통한 TTL 정보를 이용하여 도식화 한 것이다. 이 그림에서 보여지는 노드들은 Ping 응답을 하는 노드들 뿐이므로 관리자 PC에서 직접 접속이 가능한, 즉, 관리 대상인 노드들로 판단될 수 있는 것이다.

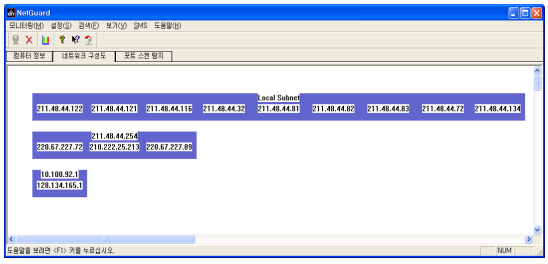


그림 9. 네트워크 구성도 표시
Fig. 9. Network Configuratin Display

3. 포트 스캐닝 탐지 기능

침입을 목적으로 하는 포트 스캐닝인 경우에는 한 소스 IP 패킷을 정의된 간격으로 동일 목적지 IP 주소에 있는 10개의 포트로 전송하여 사용자의 노드에 침입을 시도하는 것을 말한다. 이러한 공격 방법의 목적은 하나의 포트가 응답할 것이라는 기대로 가용 서비스를 검색함으로써 대상으로 선택될 서비스를 파악하는 것이다.

본 논문에서는 하나의 노드에서 검색된 많은 다른 포트를 내부적으로 기록하여 특정 노드가 2분 내에 10개의 포트를 스캐닝 하는 경우, 이를 포트 스캐닝 공격으로서 표시한다. 특정 노드로부터 1회 공격일 경우에는 NetMessageBufferSend API를 이용하여 원격 노드에는 경고 메시지를, 목적지 노드에는 침입이 탐지되었다는 메시지를 전송하도록 하였다. 3회 이상일 경우에는 SMS 기능을 이용하여 관리자의 휴대전화에 침입탐지 메시지를 전송하도록 하였다. 그림 10은 포트 스캐닝이 탐지된 예를 보여 주고 있다.

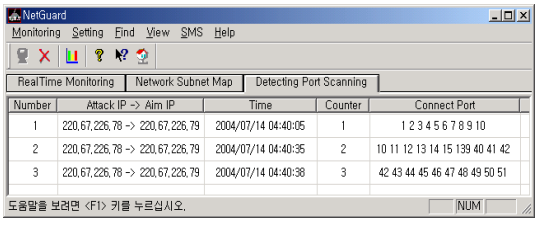


그림 10. 포트 스캐닝 감지 화면
Fig. 10. Port Scanning Detection Display

4. 부가기능

본 논문에서 구현된 모니터링 화면에서 각각의 노드들에 대해 우측 버튼을 누르면 그림 11과 같이 프로토콜 정보, 패킷량정보, 포트검색, Well-Know 포트검색 및 메시지 보내기 및 그래프 보기 부가 기능을 추가하였다.

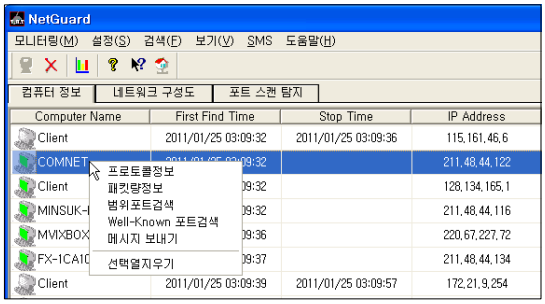


그림 11. 모니터링 부가기능
Fig. 11. Addtional Monitoring Features

가. 프로토콜 및 패킷 크기 분석

그림 12는 프로토콜 유형별 분석은 네트워크를 통하여 전송되는 패킷들을 프로토콜 계층별로 분석한 화면이고 그림 13은 패킷 크기별 분석 화면이다. 크기별 분석은 이더넷 패킷 프레임의 크기를 200byte로 분할하여 노드에 전송된 총 패킷을 크기별 패킷 수와 비율로 확인한다. 이는 특정 노드가 주로 사용하는 서비스와 대역폭을 독점하는 노드를 확인 및 분석하여 웜 바이러스[8] 및 업무 중 네트워크 오용 등을 판단할 수 있다.

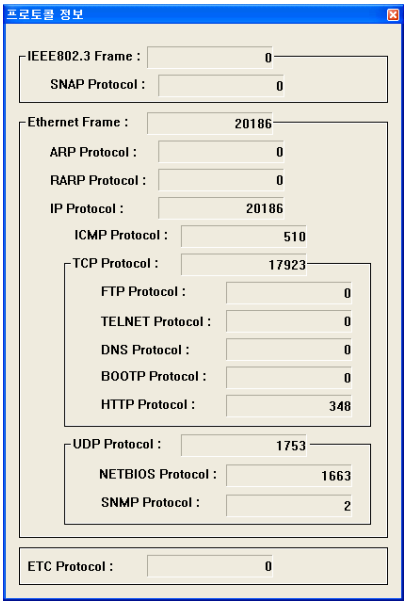


그림 12. 프로토콜 유형별 분석
Fig. 12. Protocol Type Analysis

그림 13. 패킷 크기 분석
Fig. 13. Packet Size Analysis

나. 포트 스캔 검사

포트 스캐닝은 특정 노드로부터 어떤 서비스가 운영되고 있는지 판단하는 것이다. 이를 통하여 얻어진 포트 정보는 어떤 응용프로그램이 실행중인지를 알게 해주기 때문에 상당히 치명적이다. 그러므로 해당 노드의 보안 상태를 사전에 점검 및 해킹에 대비할 수 있도록 포트 스캐닝 기능을 구현하였다. 구현된 포트 스캐닝은 특정 범위를 지정하여 포트를 검색하는 기능과 일반적으로 많이 사용되는 포트들의 사용여부를 확인하는 기능으로 이루어져 있다. 그림 14는 포트 스캐닝을 실행한 화면이다.

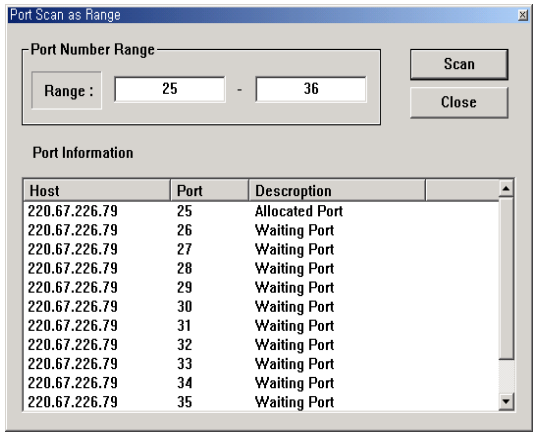
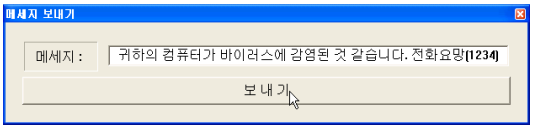


그림 14. 포트 스캔 검사 화면
Fig. 14. Port Scanning Test

그림 6에서 보듯이 포트 범위 스캐닝 기능은 검색할 포트의 범위를 사용자로부터 입력을 받아서 해당 포트가 Binding 되어있을 때는 Allocated Port로 표시되며 그렇지 않은 경우에는 Waiting Port로 표시된다.

(a)



(b)

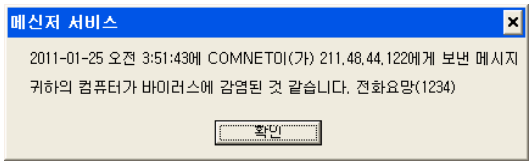


그림 15. 메시지 전송 및 확인
Fig. 15. Message Sending

다. 메시지 보내기

만약 관리자가 특정 컴퓨터에 이상을 발견하여도 그 컴퓨터가 어느 장소에 있고 또 누가 사용하는지 확인하는 것이 쉽지 않다. 이런 경우 그 컴퓨터 화면에 경고 메시지를 보낸다면 시간이 절약 될 수 있을 것이다. 그림 15는 관리자가 이상이 발견된 컴퓨터에 메시지 전송을 하고(a) 전송된 메시지가 화면에 표시되는 것을 보여준다(b).

라. 그래프 기능

통계 그래프는 실시간 네트워크 모니터링을 이용하여 수집되는 정보를 표시하는 방법으로 그래프를 사용함으로써 네트워크의 상태를 쉽게 판단할 수 있도록 한다. 일반적으로 그래프를 표현하기 위하여 사용되어지는 GDI(Graphic Device Interface)는 프로그램 재사용성이 떨어지고 강력한 그래프 기능을 구현하는데 있어서 제한적인 요소가 존재한다.

따라서 본 논문에서는 ActiveX 컨트롤을 사용하여 동적인 그래프 표현 뿐만 아니라 사용자로 하여금 네트워크의 상태를 판단함에 있어 용이성을 제공하였다. 통계 그래프는 패킷 크기별 그래프, 프로토콜 유형별 그래프, 실시간 패킷 그래프 및 패킷 드라이버 상태 그래프로 이루어져 있다. 그림 16은 패킷 크기별 그래프 화면이다.

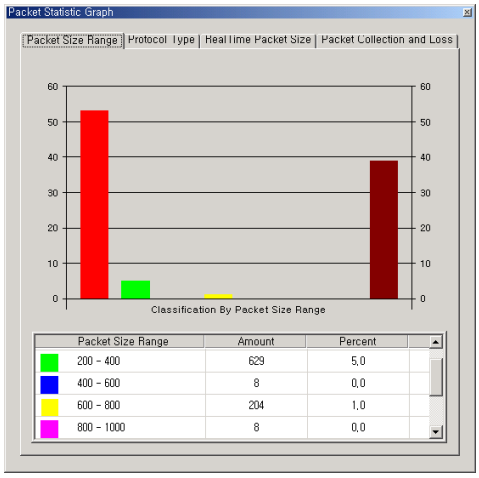


그림 16. 패킷 크기 그래프
Fig. 16. Paket Size Graph

그림 16은 패킷 크기를 0~200, 200~400, …, 1400~1514로 200(bytes)씩 8개의 구간으로 구분하여 패킷 크기에 해당하는 패킷 량(패킷 수)을 표시하였다. 그림 17은 프로토콜 유형별 그래프 화면이다.

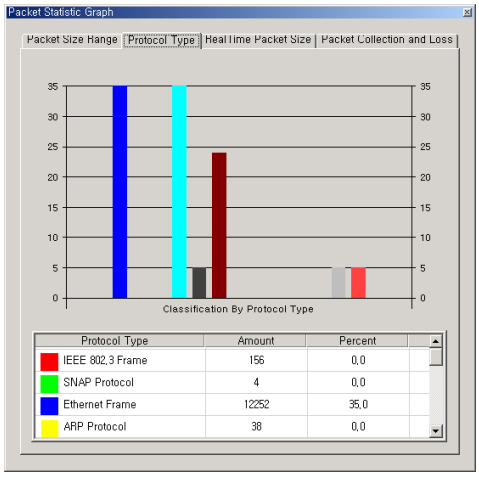


그림 17. 프로토콜 유형별 그래프
Fig. 17. Protocol Type Graph

그림 17에서 보듯이 미리 정의된 17개의 프로토콜을 프로토콜 량(패킷 수)과 전체 프로토콜중 해당 프로토콜이 차지하는 비율을 표시하였다.

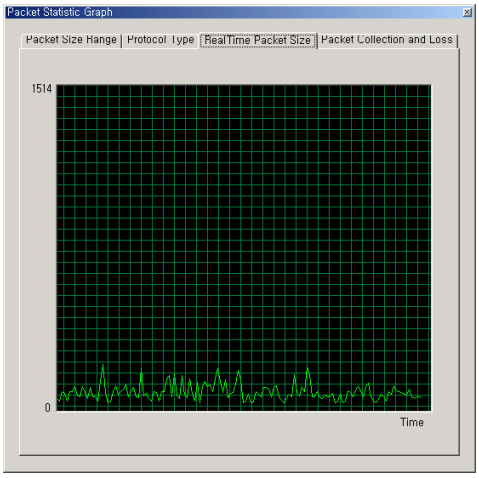


그림 18. 실시간 패킷 그래프
Fig. 18. Real Time Packet Graph

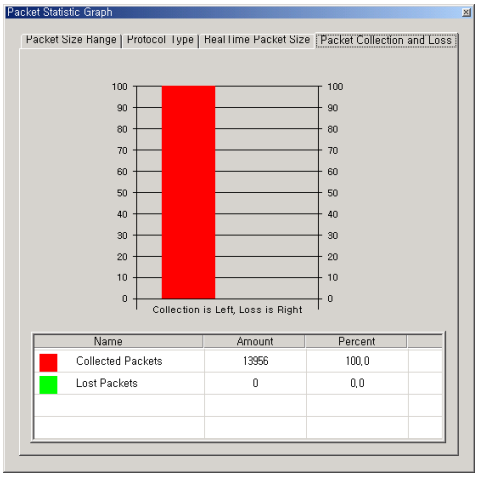


그림 19. 패킷 드라이버 상태 그래프
Fig. 19. Status of Packet Driver

그림 18은 실시간 패킷 그래프 화면으로 실시간으로 패킷의 크기를 나타낸 것으로 전체적인 트래픽 상태를 확인할 수 있다.

그림 19는 패킷 드라이버 상태 그래프 화면으로 패킷 드라이버의 상태 정보를 드라이버를 통하여 전달된 획득량과 드라이버에서 손실된 량을 보여준다.

V. 노드 인식 실험

1. 실험 방법

네트워크에 연결된 전체 노드들에 대해서 네트워크 모니터링 시스템이 자동적으로 정보를 구축하는 실험을 하여 성능을 평가하였다. 실험 환경은 네트워크에 연결된 전체 노드들의 대수는 230대이며 모두 idle 상태이고 4개의 서브넷으로 된 local 네트워크이다. 본 논문에서 구현된 네트워크 모니터링 시스템을 하나의 서브넷에 있는 노드에 설치하였다.

정보를 구축 방법은 다음과 같다. idle 상태에 있는 노드들은 local 네트워크에 속해있는 다른 노드들에 상태 정보를 얻기 위하여 arp 동보(broadcasting) 패킷을 전송한다. 수집된 arp 패킷을 이용하여 해당 노드에 대한 정보를 역으로 수집한다.

2. 실험 결과 및 분석

대상 네트워크에 대해 자동 정보 구축 기능을 테스트한 결과는 그림 20과 같다. local 네트워크에 속해있는 노드들은 idle 상태이기 때문에 패킷이 발생되지 않는다. 그러므로 특정 노드가 다른 노드에 대한 정보를 요청할 때만 패킷이 발생되므로 230대의 노드들의 정보를 구축하기 위해서는 모든 노드들이 한번이라도 통신을 해야 하므로 본 실험에서는 약 4700초가 소요되었다. 그러나 만약 노드들이 사용자에 의해 네트워크 사용이 활발해 진다면 local 네트워크에 대한 정보를 수집하는데 소요되는 시간은 기존의 실험 결과보다 현격하게 줄어들 것이라고 판단된다.

이 실험을 통해 어떤 서브넷에 대하여 평균 Idle Time과 각각의 노드 중에서 Active한 노드와 Idle한 노드를 구별할 수 있다.

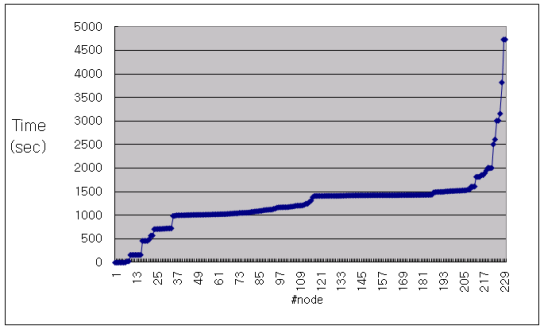


그림 20. 노드별 발견 시간 분석 결과
Fig. 20. Node Identificaiton Time Result

Ⅵ. 결론 및 향후 연구

본 논문에서는 효율적인 망 관리를 위해 ARP 동보 패킷을 이용하여 네트워크 감시 대상 노드들과 망 구성에 대한 정보를 자동적으로 구축하는 네트워크 모니터링 시스템을 설계하고 구현하였다. 본 논문에서 구현한 네트워크 모니터링 시스템은 이더넷에 연결된 노드들이 통신을 위해 필수적이고 선행적으로 발생시키는 ARP 동보 패킷을 통해 노드들을 인식한 뒤 서브넷의 구성 정보를 생성한다. 게다가 구현된 시스템은 자동 인식된 노드들에 대하여 NETBIOS 이름 확인 절차를 통해당 서브넷의 망 구성 정보를 보여줌과 동시에 PING 기능을 이용하여 각각의 노드를 감시한다. 또한 각 노드들에 대하여 포트 스캐닝 탐지를 통하여 침입을 탐지할 수 있는 실시간 감시 기능을 갖는다. 구현된 모니터링 시스템을 이용하면 자동 노드 인식이 가능함으로써 망 관리를 위해 망에 연결된 노드들의 주소를 관리해야하는 부담을 해소시켜주며 이를 통해 효율적인 망 관리를 수행 할 수 있도록 도와준다.

본 연구에 이은 향후 연구는 네트워크 모니터링 기능의 보완 및 통계 기능의 추가 및 사용자 인터페이스 개선이다. 예를 들면 서브넷 구성도를 트리 형태로 표시하는 기능 등이다. 아울러 무선랜 환경에 적용하여 실효성을 검증하는 것이다.

참 고 문 헌

[1] Metcalfe, Robert M. and Boggs, David R. (July 1976). "Ethernet: Distributed Packet Switching for Local Computer Networks". Communications of the ACM 19 (5): 395?405.

[2] David C. Plummer (1982-11). "RFC 826, An Ethernet Address Resolution Protocol ". Internet Engineering Task Force, Network Working Group. http://tools.ietf.org/html/rfc826.

[3] F. Risso and L. Degioanni, "An Architecture for High Performance Network Analysis", Proceedings of the Sixth IEEE Symposium on Computers and Communications, pp. 686 - 693, 2001. http://www.winpcap.org/

[4] V. Jacobson, C. Leres and S. McCanne, libpcap, Lawrence Berkeley Laboratory, Berkeley, CA. Initial public release June, 1994. Available now at http://www.tcpdump.org/

[5] IETF, Protocol standard for a NetBIOS service on a TCP/UDP transport: Concept and methods, RFC 1001

[6] IETF, Internet Standard Subnetting Procedure, RFC 950

[7] IETF, Internet Control Message Protocol, RFC 792

[8] Eugene H. Spafford, "The Internet Worm Program: An Analysis", Technical Report of Purdue Univ.

[9] Arreo Communications, SMS Sender, http://developer.arreo.com

저자 소개

정 인 환(정회원)

- 2000년 KAIST 정보및통신공학과 박사
- 1985 ~ 1998년 삼성전자 시스템사업부 선임연구원
- 2001년 ~ 현재 한성대학교 컴퓨터공학과 부교수

<주관심분야 : 망관리, 멀티미디어통신, VoIP>

※ 본 연구는 한성대학교 교내연구비 지원 과제임.