LAN 세그먼트 관리를 위한 PC 기반 RMON 에이전트의 설계 및 구현

박진호

송호대학 정보산업계열

요 약

현재 네트워크 관리를 위해서 SNMP(Simple Network Management Protocol), CMIP(Common Management Information Protocol), RMON(Remote network MONitoring) 등의 많은 표준안이 있다. 이것들 중 특히 RMON은 대규모 네트워크의 서브네트워크를 관리하기 위해서 개발되었지만, RMON 기능을 이용하기 위해서는 RMON 기능을 가진 허브나 라우터의 추가적인 장비도입이 필요고, 더욱이 이러한 장비의 가격이 고가이므로 네트워크 관리자에게는 관리비용의 부담이 크다는 단점이 있다. 본 논문에서는 PC 상에서 가상 장비 드라이버(VxD)를 이용하여 LAN 세그먼트의 교통량을 감시하는 PC 기반 RMON 에이전트 시스템을 제안한다. 이 시스템은 LAN 세그먼트 관리를 위한 새로운 장비의 도입이나 관리비용의 부담을 감소시킬 수 있다.

Design and Implementation of PC based RMON Agent System for LAN Segment Management

Park Jin Ho

ABSTRACT

Currently there are many standards of network management. They are: SNMP(Simple Network Management Protocol - for Internet management), CMIP(Common Management Information Protocol - standardized by ITU-T and ISO), RMON(Remote network MONitoring - for distributed management of the LAN segment), and so on. Especially RMON has created the many concerns in order to manage subnetworks of a large network, but it has negative aspects. For instance, routers or hubs with RMON capability are expensive to a network manager because of adding heavy management cost. Moreover it imposes a heavier burden on network manager, because it must use a network management tool which will be additionally needed with RMON device. This paper proposes a model of PC based RMON Agent system. The RMON Agent system monitors the traffic on LAN segment through the use of a Virtual Device Driver(VxD), based on PC. In term of cost this model will replace the expensive RMON device, and eventually enable a network manager to manage LAN segment more efficiently, due to reduced cost.

1. 서 론

오늘날 네트워크에 대한 의존도와 규모가 커지고, 통신선로의 고속화 및 대량화가 이뤄짐에 따라 네트워크 상에서 발생되는 트래픽의 양은 더욱 더 증가하게 되었고, 이로 인한 병목현상과 시스템 장애 등으로 인해 응답 속도의 저하뿐만 아니라 네트워크 전체의 다운 현상까지도 초래하고 있다. 이러한 상황은 네트워크 관리에 대한 요구를 증대시키고 있다.[1][2][3].

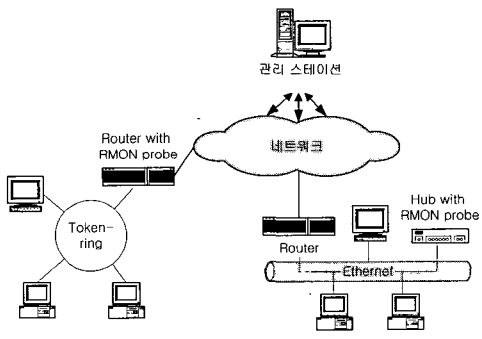
TCP/IP 기반의 인터넷 관리는 표준 프로토콜인 SNMP를 이용하며, 각 관리 대상 정보를 객체화(Object)하여 집합으로 표현한 관리 정보 데이터베이스(MIB)를 기본으로 관리를 수행한다[3][4]. SNMP의 중앙 집중 관리 구조에서 벗어나 분산 관리 형태의 개념을 채택한 RMON은 원격 세그먼트의 효과적이고 효율적인 관리를 위해 요즘 많은 관심을 불러일으키고 있으나, RMON probe를 탑재한 고가의 라우터나 허브 장비를 갖추고 있어야 하므로, 이 또한 네트워크 관리비용에 대한 큰 부담을 안게 된다.

본 논문에서는 PC기반의 Windows95/98 상에서 네트워크 VxD(Virtual Device Driver)를 이용하여 RMON probe 기능을 수행하는 RMON 에이전트 시스템의 설계 및 구현을 목적으로 한다. 본 연구를 통해 고가의 RMON probe 장비를 대체하는 효과를 기대함으로써 네트워크 관리자에게 네트워크 관리비용에 대한 부담 감소와 효율적인 LAN(Local Area Network) 관리를 수행할 수 있도록 한다.

2. RMON과 LAN 세그먼트의 관리의 중요성

현재 TCP/IP 환경의 네트워크 관리 구조는 표준 프로토콜인 SNMP를 이용하여 네트워크 상의 장비들을 폴링함으로써 관리 정보인 MIB을 이용하여 분석하는 형태로 이뤄지고 있다. 그러나 이 SNMP의 표준 MIB인 MIB-II는 SNMP 에이전트가 탑재된 시스템 자신만의 데이터를 보유하고 있기 때문에 LAN과 같은 많은 시스템들이 복잡하게 존재하는 네트워크 상에서의 전체 트래픽에 대한 통계 자료 또는 호스트들간의 트래픽 발생 현황 등을 얻어내기가 어렵다[5][6].

또한 대규모 네트워크에서 관리 시스템이 관리 대상 장비와의 정보 교환을 위해 주기적으로 폴링을 수행하게 되면, 이로 인해 발생되는 관리 트래픽은 오히려 네트워크 부하를 높이는 장애요인이 될 수 있다. 이러한 문제점의 대안으로서 RMON이 등장하게 되었으며, 각 세그먼트마다 트래픽 모니터링을 수행하는 RMON 장비를 두고, 관리 시스템은 RMON 장비로부터 가져온 세그먼트 전체 데이터를 토대로 네트워크 관리를 수행하게 된다[7]. (그림 1)은 RMON 장비를 이용한 네트워크 관리 구조도를 나타낸다.



(그림 1) RMON을 이용한 LAN 세그먼트 관리

TCP/IP 기반의 인터넷의 성장과 함께 LAN 상의 응용 프로그램들의 사용 증가는 트래픽의 병

목 현상을 보이는 WAN 뿐만 아니라, LAN상의 트래픽 양을 크게 증가시키게 되었고, 이로 인한 대부분의 장애 요소는 LAN 내부에서 원인이 발생된다. LAN에서 발생하는 문제들에는 사용자들의 네트워크 설정 오류, 사용하는 네트워크 소프트웨어 문제, LAN 인터페이스 카드와 같은 사용자의 네트워크 장비의 불량 등의 장애 요인과 과다한 이용율 같은 성능 요인이 있다. 이와 같은 장애/성능 요인에 의한 네트워크의 마비는 엄청난 비용 손실을 가져오고 작업 효율을 떨어뜨리게 된다. 뿐만 아니라, 어떤 경우에는 장애 원인의 발견 및 복구 없이 LAN장비에 과다한 투자를 수행하기도 한다. 따라서, LAN상의 장애/성능 관리는 장비의 투자 및 유지 보수 비용을 낮추고 성능을 최대로 유지하기 위한 필수 요소이다.

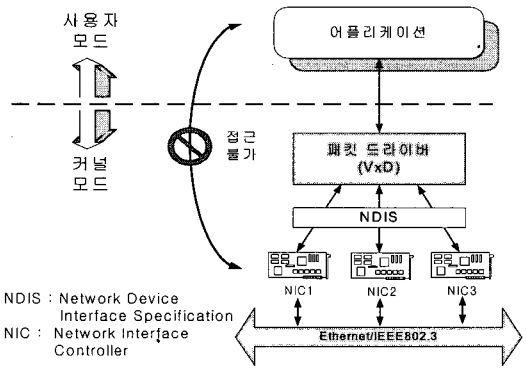
3. NIC 접근 기술과 패킷 모니터링

PC의 LAN 카드(NIC:Network Interface Card)는 단순히 PC 혹은 네트워크에서 전달되어 오는 정보를 상호 교환 할 수 있도록 만들어 준다. 즉 PC에서 전송 요구가 발생하면 NIC로 정보를 일정한 형태로 만들어 보내고 NIC에서는 이 정보를 일정한 버퍼에 저장한 다음 네트워크에 적당한 형태(Serial)로 보낸다.

여기서 PC와 NIC 사이에서 논리적으로 묶어주는 소프트웨어가 필요한데, 이 소프트웨어를 네트워크 드라이버라고 하며, 패킷 드라이버(Packet Driver)를 이용하여 NIC를 직접 제어하는 방법과 NOS (Network Operating System)를 통해서 NIC를 간접적으로 제어하는 방법이 있다. 논문에서는 NDIS (Network Driver Interface Specification) 표준을 따르는 패킷 드라이버를 사용했다.

Windows9x 기반의 네트워크 가상 장치 드라이버(VxD)는 WIN32 응용 프로그램이 PC에 설치된 네트워크 카드로의 직접적인 접근을 가능하게 한다. 이러한 네트워크 접근 방법은 네트워크를 모니터링하는 응용 프로그램과 특정 프로토콜 스택을 구현하고자 하는 프로그래머들에게는 매우 유용하다.

WIN32 기반의 프로그래밍은 로우-레벨 네트워크 접근(low-level network access)을 위해 직접 제공하지는 못하기 때문에, 이러한 접근을 필요로 하는 어플리케이션들은 하부의 네트워크 카드(NIC)와 그 위에 놓이는 WIN32 어플리케이션 사이에 인터페이스로서 제공되는 가상 장치 드라이버(VxD)를 사용해야 한다. 이러한 기본 구조는 (그림 2)와 같다.



(그림 2) NIC 접근 구조

(그림 2)에서 보여지듯이 인터페이스 추상층으로 간주되는 NDIS(Network Device Interface Specification) 층이 네트워크 하드웨어와 VxD 사이에 존재하게 된다. 이러한 중간 계층의 목적은 상위 프로그램이 하부의 다양한 하드웨어 어댑터들로부터 NIC 접근을 요구하는 투명성을 제공하기 위한 것이다. 소프트웨어를 보호하는 것이다. 그러므로, Packet Driver는 NDIS와 호환되는 모든 NIC와도 통신할 수 있다.

4.네트워크 VxD를 이용한 RMON MIB과 OID의 구성

어플리케이션에서 VxD 서비스(NIC 바인딩, 수신 모드, 송신 모드, 초기화등)를 불러내는 방법으로서는 Win32 API의 DeviceIoControl() 함수를 사용해서 VxD에 명령어를 보낸다. 어플리케이션 레벨에서 VxD 레벨로 하달된 서비스 요구 명령(바인딩, 패킷 송신, 패킷 수신, 모드 설정 등)은 실제로 NDIS 함수에 의해 실행된다. 즉, WIN32 함수인 DeviceIo Control()로부터 〈표 1〉의 서비스 제어 코드를 파라미터로 사용하여 서비스의 요구가 호출될 때, 이 서비스 요구는 VxD 안의 후크(Hook) 함수가 가로채고, 서비스 형태에 따라서 동작하게 된다. 이러한 서비스 중에서 실제로 NIC를 직접 제어하기 위한 명령어가 필요한데 NDIS는 4바이트 또는 2바이트 크기의 객체 식별자(Object Identifier, 이하 OID)로서 구분하고 있다. 이 OID들은 NdisRequest() 함수의 입력 파라미터로서 사용된다. 이 함수는 NIC의 상태(status)나 기능(capability)들을 질의하며, 특정한 상태로 NIC를 설정한다. General Objects Identifier는 LAN 타입에 상관없이 대부분의 NIC에 사용될 수 있으며, Ethernet Objects Identifier는 이더넷 NIC에 적용되는 식별자이다. 이들은 다시 NIC의 동작 모드를 설정하기 위한 것과 통계 정보를 얻기 위한 것들로 구분할 수 있다. 〈표 2〉에 주요 OID들을 나타내었다[8].

OID_GEN_CURRENT_PACKET_FILTER를 사용해서 NIC로부터 여러 패킷 필터를 설정할 수 있다. 사용 가능한 패킷 필터 유형은 다음 〈표 3〉과 같으며, 이들은 각각 조합해서 사용할 수 있다. 본 논문에서는 NDIS_PACKET_TYPE_ PROMISCUOUS를 사용해서 네트워크 상의 모든 패킷을 모니터링 할 수 있도록 NIC를 "Promiscuous 모드"로 설정했다.

<표 1> DeviceIoControl() 함수에 사용된 서비스 제어 코드

서비스 제어 코드	코드값	설 명
IOCTL_PROTOCOL_QUERY_OID	0x8000	Object ID 얻어내기
IOCTL_PROTOCOL_SET_OID	0x8001	Object ID 설정하기
IOCTL_PROTOCOL_STATISTICS	0x8002	Adapter statistic 알아내기
IOCTL_PROTOCOL_RESET	0x8003	어댑터 초기화
IOCTL_PROTOCOL_READ	0x8004	Packet 받기
IOCTL_PROTOCOL_WRITE	0x8005	Packet 보내기
IOCTL_PROTOCOL_MACNANE	0x8006	드라이버 이름 알아내기
IOCTL_PROTOCOL_BIND	0x8007	어댑터와 VxD와의 바인딩
IOCTL_PROTOCOL_UNBIND	0x8008	어댑터와 VxD와의 바인딩

<표 2> NIC의 동작 특성 및 통계 정보를 위한 OID

구분		OID명	설 명
General Objects	동작	OID_GEN_LINK_SPEED	NIC의 링크 속도(kbps)
		OID_GEN_CURRENT_PACKET_FILTER	NIC 드라이버로부터 특정 유형의 패킷을 수신하기 위한 패킷 필터 설정
	통계	OID_GEN_RCV_CRC_ERROR	체크섬(CRC 또는 FCS) 에러를 갖는 패킷 수
		OID_GEN_RCV_NO_BUFFER	NIC의 수신 버퍼 공간의 부족으로 인해 버려지는 패킷 수
Ethernet Objects	동작	OID_802_3_CURRENT_ADDRESS	NIC가 현재 사용하고 있는 주소
	통계	OID_802_3_RCV_ERROR_ALIGNMENT	Alignment 에러 프레임의 수
		OID_802_3_XMIT_MAX_COLLISIONS	과도한 Collisions로 인해 전송되지 못한 프레임의 수
		OID_802_3_RCV_OVERRUN	Overrun 에러로 인해 수신되지 못한 프레임의 수
		OID_802_3_XMIT_UNDERRUN	Underrun 애러로 인해 전송되지 못한 프레임의 수

이 OID들을 사용해서 RMON 에이전트를 구현하기 위한 기본적인 정보들을 얻어 올 수 있다. 이 기본 정보들은 RMON MIB 구성의 기본이 되는 RMON statistics 그룹의 MIB 변수들과 대응될 수 있으며, <표 4>에 나타내었다.

<표 3> NIC의 패킷 필터 설정을 위한 OID

OID명	설 명
NDIS_PACKET_TYPE_ALL_MULTICAST	모든 멀티캐스트 주소 패킷(멀티캐스트 주소 리스트에 속하지 않는 패킷 포함)
NDIS_PACKET_TYPE_DIRECTED	목적지 주소가 NIC의 주소와 동일한 패킷
NDIS_PACKET_TYPE_GROUP	현재의 그룹 주소로 전송된 패킷
NDIS_PACKET_TYPE_MULTICAST	멀티캐스트 주소 리스트에 포함된 패킷
NDIS_PACKET_TYPE_PROMISCUOUS	네트워크상의 모든 패킷

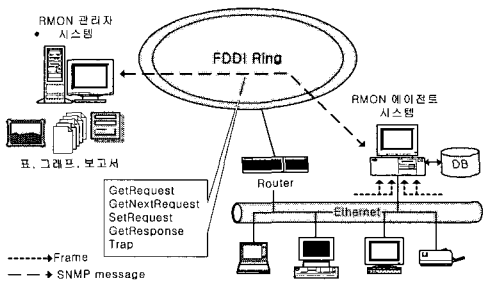
<표 4> OID와 RMON MIB

구분	OID명	관련 MIB 변수명
동작	OID_GEN_CURRENT_PACKET_FILTER (NDIS_PACKET_TYPE_PROMISCUOUS)	etherStatsOctets, etherStatsPkts, etherStatsBroadcastPkts, etherStatsMulticastPkts, etherStatsPkts64Octets, etherStatsPkts65to127Octets, etherStatsPkts128to255Octets, etherStatsPkts256to511Octets, etherStatsPkts512to1023Octets, etherStatsPkts1024to1518Octets
통계	OID_802_3_RCV_ERROR_ALIGNMENT, OID_GEN_RCV_CRC_ERROR	etherStatsCRCAlignErrors
	OID_GEN_RCV_NO_BUFFER	etherStatsDropEvents
	OID_802_3_XMIT_MAX_COLLISIONS	etherStatsCollisions
	OID_802_3_RCV_OVERRUN	etherStatsOversizePkts
	OID_802_3_RCV_UNDERRUN	etherStatsUndersizePkts

5. RMON 에이전트 시스템의 설계 및 구현

5.1 RMON 에이전트 시스템 구조

RMON 에이전트는 LAN에 연결되어 있는 PC에 탑재되어 네트워크 상에 흘러 다니는 모든 패킷을 수신하고 분석하여 체계적인 방법으로 데이터베이스화한다. 또한 RMON 관리자로부터 SNMP 요구에 대해 응답을 하며, 특정 이벤트 발생에 대한 트랩(trap) 메시지를 관리자 시스템에 전달한다. (그림 3)은 원격 LAN 세그먼트 관리를 위한 RMON 에이전트 시스템의 전체 구조도를 나타낸다.

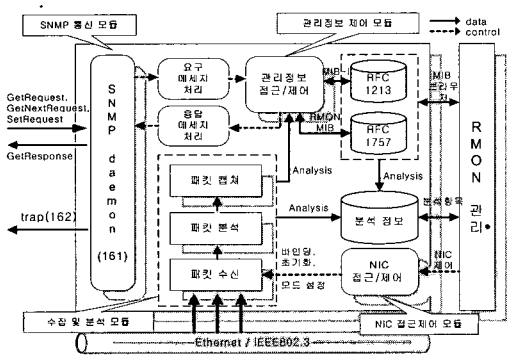


(그림 3) 원격 LAN 세그먼트 관리를 위한 RMON 에이전트 시스템의 전체 구조도

(그림 3)의 RMON 관리자 시스템은 PC나 유닉스, 또는 특정 밴더 기반에 네트워크 관리 소프트웨어를 장착한 NMS(Network Management System)로서, RMON 에이전트 시스템으로부터 SNMP 메시지들(GetRequest, GetNext Request, GetResponse, etc)을 통해 주기적으로 수집된 정보를 가공하여 실시간으로 분석 결과를 보여주거나 혹은 과거 누적된 정보를 바탕으로 분석기간 동안의 네트워크 상태를 보여준다. 네트워크 관리

자는 이 관리자 시스템으로부터 얻어진 표, 그래프, 보고서등을 활용하여 관리하고자 하는 LAN의 성능 및 장애 상태를 알 수 있다.

위와 같은 고려 사항들을 토대로 설계된 RMON 에이전트 시스템은 크게 4 부분의 기능 모듈들로 나뉜다. 4개의 기능 모듈들은 SNMP 통신 모듈, 패킷 수신 및 분석 모듈, 관리정보 제어 모듈, NIC 접근 제어 모듈로 구성된다. (그림 4)는 이러한 모듈간의 관계를 나타내는 RMON 에이전트 시스템의 전체 내부 구성 모듈을 나타낸다.



(그림 4) RMON 에이전트 시스템 전체 내부 구조

SNMP 통신 모듈은 외부 시스템의 SNMP 폴링 요구에 대한 SNMP 기본 동작을 위해 SNMP 데몬(daemon)으로서 동작한다. 이 모듈은 161번 포트로 수신되는 SNMP 요구 패킷을 분석하여, 인증(Authentication) 및 유효성(Validation)을 검사한 후 요구 메시지 처리기(Request Message Handler)로 제어를 넘긴다. 이 때, 인증과 유효성이 적합하지 않은 SNMP 메시지는 버려지게 된다. 또한 응답 메시지 처리기(Request Message Handler)로부터 받은 데이터에 에러 정보와 request id 등을 추가하여 송신 시스템에 응답을 보낸다. 요구 메시지 처리기는 SNMP GetRequest PDU(Protocol Data Unit)의 'variablebindings' 필드로부터 요구되어진 MIB 변수들의 식별자(identifier)들을 분리해 낸다.

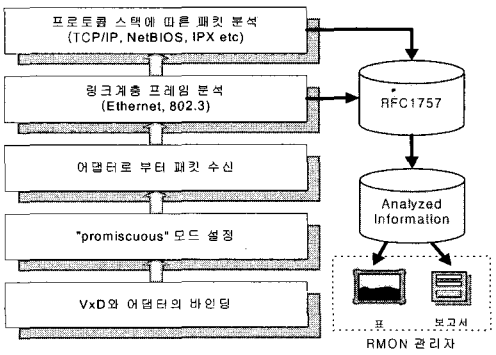
관리 정보 제어 모듈은 요구 메시지 처리기를 통해 얻어진 MIB 변수 식별자들을 파라미터로 사용하여 RMON MIB이 구현된 관리 정보 데이터베이스(RFC1757)에 접근하고, 검색과 질의(search and query)를 통해 요구된 MIB 객체의 저장된 값을 가져온다. 이 때, 최적의 검색 알고리즘을 설계하여 응답 시간을 줄이도록 한다. 이 데이터베이스로부터 얻어진 결과 값은 응답 메시지 처리기(Response Message Handler)에게 보내져 MIB 변수 식별자와 함께 SNMP GetResponse PDU(Protocol Data Unit)의 'variablebindings' 필드를 구성하도록 한다. RFC1213은 MIB-II에 대한 관리 정보 데이터베이스이다.

패킷 수신 및 분석 모듈은 패킷 수신 단계, 패킷 분석 단계, 패킷 캡처 단계로 이뤄지며, Promiscuous mode 상태에서 네트워크로부터 NIC 버퍼에 수신되는 모든 패킷들을 읽어들여 각 프로토콜 계층에 따라 패킷을 분석하고, 그 분석된 결과는 관리 정보 데이터베이스에 저장하는 RMON 에이전트의 핵심 모듈이다. 이 모듈은 PC상에서 순수한 소프트웨어로 구현되어야 하므로, 기존 RMON 하드웨어 장비가 갖는 고속성, 신뢰성, 지속성 등을 갖춰야 한다. (그림 5)는 패킷 수집 및 분석 모듈의 일련한 작업 처리 절차를 보여주고 있다.

NIC 접근 제어 모듈은 NIC 초기화 및 바인딩, promiscuous mode 설정, NIC 정보(선로 속도, MAC 주소 등) 요구와 같은 NIC와 관련한 기능들을 수행한다.

이외에 RMON 관리자 시스템에게 네트워크 분석 정보, 패킷 수집(Packet Capture), NIC 정보, 호스트 정보 등의 세그먼트 관리를 위한 분석

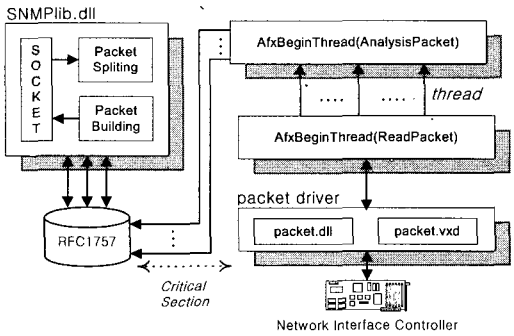
정보를 제공하기 위한 데이터베이스가 존재한다.



(그림 5) 패킷 수집 및 분석 절차

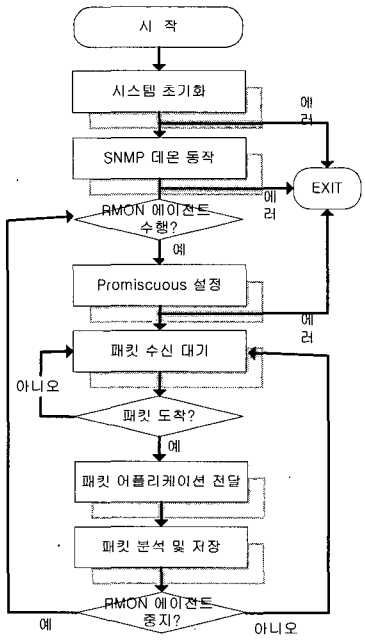
5.2 RMON 에이전트 시스템의 구현

(그림 6)은 RMON 에이전트의 주요 모듈을 표현한 내부 구현 모델이다. 설계된 SNMP 통신 모듈과 은 MFC 동적 링크 라이브러리로 구현하였다. SNMPlib.dll은 snmp 데몬의 역할을 수행하는 동적 링크 라이브러리의 구현 파일이다. 포트 161을 갖는 UDP(User Datagram Protocol) 소켓을 생성하여 SNMP 요구 패킷을 수신하며, 수신된 SNMP 요구 패킷 분석과 응답 패킷 생성의 역할을 수행한다. NIC로의 접근과 패킷 수신을 수행할 수 있도록 인터페이스 역할을 수행하는 packet.vxd(패킷 드라이버)와 이 드라이버가 제공하는 서비스를 어플리케이션이 쉽게 사용할 수 있도록 C 라이브러리로 구현한 packet.dll (wrapper function)이 NIC와의 인터페이스 역할을 한다. 수집 모듈(AfxBegin Thread(ReadPacket))과 분석 모듈(AfxBegin Thread (Analysis-Packet))은 패킷 수신의 신뢰성을 높이기 위해 쓰레드(thread)로 구현되었으며, 결과 저장 모듈은 크리티컬 섹션(Critical Section) 설정을 통해 데이터 무결성(Integration)을 보장하였다.



(그림 6) RMON 에이전트 시스템 구현 모델

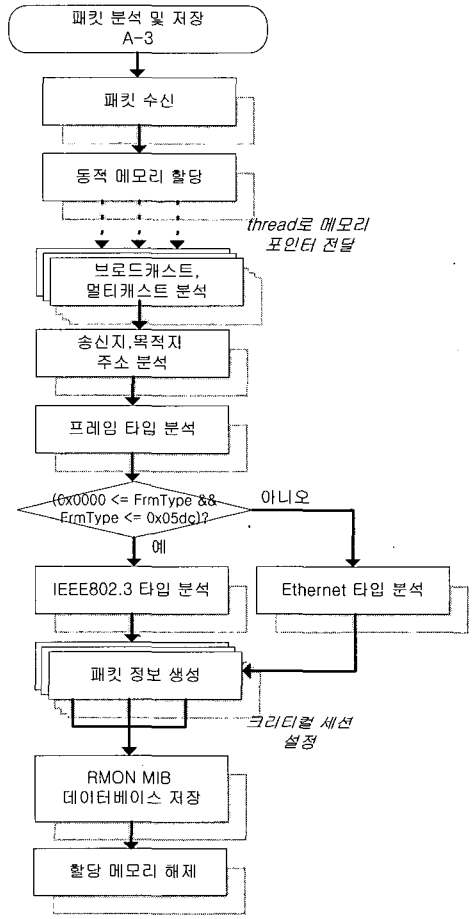
RMON 에이전트 시스템의 전체 수행 흐름도는 (그림 7)와 같다.



(그림 7) RMON 에이전트의 전체 동작 흐름도

RMON 에이전트가 실행되면 우선 변수 및 NIC 초기화, 시스템 환경 설정 등의 초기화 작업

이 이뤄지며, SNMP 질의에 대한 응답을 위해 SNMP 데몬이 동작한다. 구현된 시스템은 관리자가 RMON 에이전트 수행 버튼을 누르기 전까지는 패킷 수집을 하지 않으며, 네트워크 정보(NIC 정보, 선로 속도, MAC 주소, 시간)만을 출력한다. RMON 에이전트 수행을 실행시키면, NIC는 Promiscuous 모드로 설정되고, 이 때부터 네트워크 상의 모든 패킷을 수신하게 된다. 가상 장치 드라이버에게 패킷 도달 여부를 검사하고, 패킷이 수신되었으면, 상위 어플리케이션으로 패킷의 복사본을 전달하고, 수신된 패킷이 없으면 다시 패킷 수신 대기 상태로 돌아간다. 상위 어플리케이션으로 전달된 패킷은 프로토콜 스택에 따른 분석과 MIB 변수에 해당하는 값을 저장한다. VxD에 의해 어플리케이션 레벨로 전달된 패킷은 프로토콜 스택에 따라 분석을 하고, 분석된 정보는 RMON MIB 데이터베이스에 저장된다. 각 패킷에 대한 분석은 하나의 쓰레드(thread)로서 동작한다. 즉, 패킷이 수신하면, 패킷 크기만큼의 메모리를 힙(Heap)으로부터 할당하고, 이 메모리에 패킷 정보를 담고 이 메모리의 포인터를 쓰레드로 동작하는 분석 함수에 전달한다. 분석 함수에서는 각 패킷이 분석될 때마다 자신의 메모리에 분석 정보를 저장하며, 이후 하나의 공동 데이터베이스에 기록한다. 여러 쓰레드로부터 하나의 공동 저장소에 접근해서 기록하기 때문에 저장 과정 전후에 크리티컬 섹션을 두었다. 패킷 분석 순서는 목적지 주소(6바이트), 수신지 주소(6바이트), 패킷 타입 또는 패킷 길이(2바이트), 이후 이더넷 타입 또는 IEEE802.3 타입에 따라 각 계층별 프로토콜 분석 순으로 이뤄진다. (그림 8)는 패킷 분석 및 저장 과정을 보여준다.



(그림 8) 패킷 분석 및 저장 과정

현재 수신 프레임으로부터 분석되는 정보는 링크 계층을 중심으로 구현되었으며, 분석 내용은 수신 총 패킷 수, 수신 총 바이트 수, 브로드캐스트 패킷 수, 멀티캐스트 패킷 수, 프레임 유형별 패킷 수, 패킷 크기별 패킷 수, 네트워크 계층의 프로토콜 종류 분석 등이다. 이 정보들은 RMON Statistics 그룹의 MIB 변수들에 해당하는 정보들이다.

다음은 지금까지 구현된 RMON 에이전트 시스템을 외부 RMON 관리자시스템을 통해 RMON

에이전트의 동작 상태를 알아본다. RMON 에이전트 시스템은 연속적인 SNMP 요구 메시지에 응답을 수행하면서 네트워크 상의 패킷 수신 및 분석이 계속 이뤄져야 한다. 이를 위해 개발한 RMON 에이전트와 RMON probe 기능을 제공하는 상용 switching HUB를 같은 LAN 세그먼트에서 MIB 브라우저를 사용하여 SNMP 폴링을 통해 정상적으로 응답이 돌아오는 가를 살펴보았다. 실험한 세그먼트에서 MIB 브라우저가 개발한 RMON 에이전트 시스템과 switching HUB에게 실시간으로 1초 간격으로 30회 SNMP Request PDU를 발생해서 얻은 응답 메시지를 분석해서 얻은 각각의 실시간 결과가 일치하였다. 즉, SNAPS가 제공하는 여러 LAN 분석 항목들에 대해서 RMON probe 하드웨어 장비를 폴링하는 것과 같이 양쪽 모두 정상적인 동작 상태를 보였다.

6. 결 론

본 논문은 VxD를 이용한 LAN 세그먼트 관리를 위한 RMON 에이전트 시스템의 구현을 목표로 제안되었다. 이를 위해 기반 기술의 소개와 각 시스템의 전체 내부 구조 설계 및 각 구성 모듈의 기능과 역할에 대해서 정의하였고, LAN 세그먼트 관리를 위한 관리 항목들을 정의하였으며, 설계를 바탕으로 RMON 에이전트 시스템을 구현하였다.

기반 기술로는 UNIX와 Windows9x에서의 패킷 모니터링 기술에 대해서 소개했으며, 특히 네트워크 VxD(Virtual Device Driver)를 이용한 NIC(Network Interface Controller) 접근과 패킷 모니터링 기술에 대해 자세히 기술하였으며, RMON probe를 이용한 LAN 세그먼트 관리 구조에 대해 설명하였다. 기존의 RMON 하드웨어 장비가 수행하는 기능들을 포함하고, PC상에서 RMON 에이전트가 동작하기 위한 전체 내부 구조와 패킷 수신부터 분석까지의 일련 과정을 정의하였으며, RMON MIB의 데이터베이스 관리 구조를 정의하였다. 또한 구현된 이 시스템을 실제 송호대학 LAN 환경에 적용해 봄으로써 외부 RMON 관리자 툴과의 완벽한 호환성과 RMON 하드웨어 장비의 대체 가능성에 대해 검증해 보았다.

제안된 이 시스템은 이더넷에 연결된 어떠한 PC에서도 동작이 가능하므로 RMON probe를 갖추지 않은 LAN에서의 세그먼트 관리가 가능해짐으로 네트워크 관리자의 관리비용 부담을 줄이는 효과를 기대할 수 있다. 또한 하드웨어에 비해 소프트웨어가 갖는 유연성, 확장성, 설치용이 등의 장점들을 갖게 된다.

향후 연구 과제로는 아직 구현되지 않은 RMON MIB 그룹들의 구현과 이들을 이용한 분석 항목 개발이 남았으며, RMON-II의 지원에 대한 연구도 진행되어야 할 것이다.

참고문헌

[1] William Stallings, "SNMP, SNMPv2, and RMON : Practical Network Management", Addison-Wesley Publishing Company, 1996

[2] Heng Pan, "SNMP-Based ATM Network Management", Artech House, Inc., 1998

[3] J. Case, M. Fedor, M. Schoffstall, J. Davin, "Simple Network Management (SNMP)", RFC 1157, May 1990

[4] K. McCloghrie, M. Rose, "Management

Information Base for Network Management of TCP/IP-based Internets : MIB-II", RFC1213, March 1991

[5] Gilbert Held, "LAN Management with SNMP and RMON" John Wiley& Sons, New York, NY,1996. ISBN 0-471-14736-2

[6] Nathan J. Muller, "SNMP's Remote Monitroing MIB", International Journal of Network Management, WILEY, Vol 6, No 1 1996.

[7] Gilbert Held, "LAN Management with SNMP and RMON", John Wiley & Sons, Inc., 1996

http://msdn.miscrosoft.com/library/wcedoc/wceddk

[9] Snajay Dhawan, "Networking DEVICE DRIVERS", VNR Communications Library, 1995

박 진 호

1995년 대전대학교 전자계산학과(공학사)

1997년 대전대학교 컴퓨터공학과(공학석사)

1997년 ~ 현재 성균관대학교 전기전자 및 컴퓨터공학부(박사수료)

2000년 ~ 현재 송호대학 정보산업계열 전임강사