

IP-ATM 망에서의 다중 ARP 서버 및 관리 대행자의 설계 및 구현

이 병 기*

A Design and Implementation of Multiple ARP Server and Management Agent on IP/ATM Internetwork

Byeong-Ki Lee*

요 약

IP over ATM은 ATM 망에서 IP 응용을 지원하기 위한 방법으로 개발되었다. 그러나 기존의 IP-ATM 표준에서는 하나의 LIS에 ARP 서버가 독립적으로 존재하는 것을 기본으로 설계되었으나, 본 논문에서는 하나의 ARP 서버가 다수개의 LIS를 수용할 수 있도록 하는 다중 ARP 서버를 설계 및 구현하였으며, 또한 이러한 다중 ARP 기능을 가진 ARP 서버 및 ATM 장치를 인터넷 표준 프로토콜인 SNMP를 통해 관리할 수 있도록 하기 위한 관리 대행자를 구현하였다.

Abstract

IP over ATM is developed to support IP application in ATM network. However, in typical IP-ATM standard, a ARP server exists independently at one LIS, this paper designs and implements a multiple ARP server allowing that a ARP server accommodates multiple LIS. Also, this paper implements a management agent to manage ATM equipment and a ARP server which has multiple ARP function through internet-based SNMP.

* 광운대학교 전자계산학과 박사과정
논문접수:98.9.3. 심사완료:98.10.20.

I. 서론

ATM(Asynchronous Transfer Mode)은 오디오, 비디오 및 고속의 데이터 전송 등 고속의 멀티미디어 통신에 대한 사용자의 요구를 충족시키기 위해 설계된 고속 통합 서비스망 기술이다. 그러나 현재 대부분의 응용 서비스들은 TCP/IP를 기반으로 하고 있으며 순수 ATM 응용 서비스는 ATM이 광범위하게 확산될 때까지는 일반화되기 어렵기 때문에 기존의 TCP/IP, IPX 등의 기존 망과 공존할 수 밖에 없다.

따라서 B-ISDN과 같은 고속의 통신망에서 LAN과 같은 비연결형 서비스를 지원하는 것이 무엇보다 필요하다. TCP는 연결형 서비스이지만 IP는 비연결형 서비스를 지원하므로, 기존의 TCP/IP 응용 서비스들을 연결형 망인 ATM 망에서 수용하기 위해서는 ATM 망의 연결형 기술 위에 비연결형 서비스를 제공하는 오버레이 방법이 필요하다. ATM 망을 이용하여 비연결형 서비스를 제공하는 방법으로 가장 많이 논의되고 있는 기술이 IETF(Internet Engineering Task Force)에서 제안하고 있는 Classical IP over ATM(IPOA)[2]과 ATM Forum에서 제안하고 있는 LAN Emulation[3]이다.

IPOA는 IP 패킷을 LLC/SNAP Encapsulation [10]을 이용하여 AAL5 패킷으로 캡슐화하고 캡슐화된 셀을 ATM 종단 시스템간에 ATM 연결을 통하여 통신을 한다. 즉 IPOA에서는 SVC(Switched Virtual Connection)를 지원하기 위해 ATM 신호와 IP 주소를 ATM 주소나 VC 식별자로 매핑시키기 위해 필요한 ATMARP(ATM Address Resolution Protocol)를 지원하는 것이다. 그러나 현재의 리눅스 기반의 IPOA에서는 각 LIS(Logical IP Subnet)에서 하나의 ARP 서버만을 고려하여 설계되었다. 또한 이러한 IPOA 기반의 망을 관리하기 위해 단순히 해당 시스템의 콘솔에서 관리 명령어를 통해 망의 구성이나 상태를 관리하기 때문에 IPOA를 지원하는 클라이언트와 서버에 대해 전체 망 차원에서 통합적인 관리가 어렵다.

따라서 본 논문에서는 구성이 간단하고 높은 성능을 가지면서 다중의 ARP 서버 기능을 수행하는 IPOA 설계에 초점을 맞추어 리눅스 기반 환경에서 다중 ARP 서버 기능을 갖는 IPOA 시스템을 설계 및 구현하였으며, 이와 더불어 인터넷 망 관리 표준 프로토콜인 SNMP(Simple Network Management Protocol)를 이용하여 망의 구성이나 상태 및 성능을 관리하고 제어할 수 있도록 하는 관리 대행자(Agent) 시스템을 구현하였다.

II. Classical IP over ATM

2.1 개요

CLIP는 ATM 망에서 기존의 IP 어플리케이션을 지원하기 위해 RFC1577[2]을 기반으로 하여 설계된 기본 모델로서 ATM 망은 하나 혹은 여러 개의 논리적인 IP 서브넷(Logical IP Subnet: LIS)으로 구성된다. LIS 내에서의 IPOA 시스템들 간에는 ATM 연결을 설정하여 통신을 하며, LIS 사이에는 라우터에 의해 연결된다. 또한 ATM 주소 해석은 각 LIS에 위치해 있는 ATMARP 서버를 통하여 이뤄지며 ATM 망에서 통신할 때 AAL 타입은 AAL5를 사용한다. 따라서 IPOA 시스템 간에 IP 데이터 패킷을 교환하기 위해서는 IP 데이터 패킷을 AAL5로 인코딩시켜 ATM 연결을 통하여 전달하여야 한다. 이때 사용되는 것이 LLC/SNAP Encapsulation이다. 즉, IP 데이터 패킷은 LLC/SNAP을 이용하여 캡슐화(Encapsulation)되어 AAL5 데이터의 페이로드에 저장되어 전달된다.

2.2 CLIP 요구사항

CLIP의 기본적인 요구사항은 다음과 같다.

- 하나의 LIS내의 모든 멤버는 동일한 IP 네트워크/서버넷 주소 및 네트워크 마스크를 가져야 한다.
- 하나의 LIS내의 모든 멤버는 ATM 망에 직접 연결되어야 하고 LIS내의 모든 멤버는 같은 LIS의 다른 멤버와 ATM을 통해 통신할 수 있어야 한다.
- 하나의 LIS내의 모든 멤버는 타 LIS의 멤버와

IP 라우트를 통하여 연결되어야 한다.

- LIS의 모든 멤버는 PVC를 사용할 때, In-ATMARP를 통해서 VC를 IP 주소로 매핑시키는 메커니즘을 제공해야 한다.
- LIS의 모든 멤버는 SVC를 사용할 때, ATMARP를 통해서 IP 주소를 ATM주소로 매핑시키는 메커니즘과 InATMARP를 통해서 ATM주소를 IP주소로 매핑시키는 메커니즘을 제공해야 한다.
- RFC1577에서 정의된 기본 IP 네트워크를 구현한다.
- ATM Forum에서 정의한 ATM 주소를 지원한다.
- IEEE 802.2 LLC/SNAP Encapsulation을 지원한다.
- 기본 MTU 크기는 9188 옥텟(9180 + 8 (LLC/SNAP 헤더))이다.
- 인코딩은 AAL5를 지원한다.

2.3 ATMARP 기능

Classical IP over ATM의 ATM ARP 기능은 통신할 상대 IP 주소를 이용하여, ATM ARP server로부터 그에 매칭되는 ATM 주소를 resolution 하는 것이다. IPOA client는 자신의 LIS내 ARP 서버의 ATM 주소를 미리 알고 있어야 하고 초기화 시에 SVC 연결 설정을 통하여, ARP server와 SVC를 가져야 하는데 이는 그림 1의 ARP 절차에 의해 이루어진다.

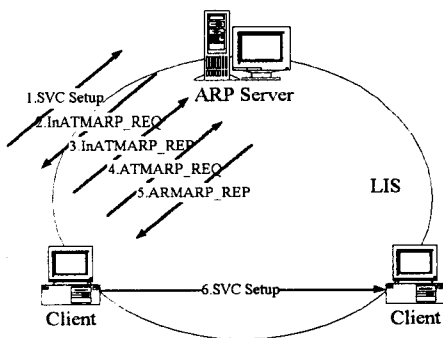


그림 1. SVC 기반 Classical IP 프로토콜
Fig. 1. SVC based Classical IP Protocol

- 1) 클라이언트는 ARP서버와 LLC/SNAP VC를 설정.
- 2) ARP서버는 InATMARP_REQ를 클라이언트로

전달

- 3) 클라이언트는InATMARP_REPLY에 ATM 주소와 IP 주소를 서버로 전달, 자신의 주소를 서버에 등록.
- 4) 송신할 IP 패킷이 있으면 패킷이 전달될 next hop 의 ATM 주소를 얻기 위하여 ATMARP-REQ를 서버로 전송.
- 5) 서버는 대응된 ATM 주소를ATMARP_REPLY에 실어 클라이언트로 전달.
- 6) 시그널링을 통하여 next hop과의 가상 연결을 설정한 후 IP 패킷을 전달.

III. 리눅스 기반의 ATM 구조

3.1 ATM on Linux 개요

ATM on Linux는 Werner Almesberger와 Heikki Vatiainen 의해서 개발된 ATM 패키지로 현재 공개 버전으로 사용되고 있으며, 소프트웨어는 개발자 커널을 지원하는 atm-0.38 버전과 안정 커널을 지원하는 atm-0.31버전으로 구분된다. ATM on Linux가 상업적인 목적이 없이 개발되었다는 점에서 안정성, 커널과의 인터페이스, 관리 프로그램 등에 있어 부족한 부분이 있긴 하지만, 이를 최적화하여 보완하면 상용 제품으로도 사용 가능한 것으로 판단된다.

3.2 ATM 디바이스 드라이버

리눅스 기반 ATM 개발 초기에는 SUN사의 Sbus 제품이 대부분이었고 PCI bus 방식의 제품은 거의 없는 상태여서 결국 ZeitNet(ZN1221) 및 Efficient Networks EN155p 어댑터를 선택하여 ATM 드라이버를 개발하였다. 현재 지원 커널 및 ATM 버전별로 리눅스 패키지에서 지원하는 ATM 어댑터 카드의 사양은 표 1과 같다.

표. 1. 리눅스 ATM 지원 커널 및 ATM 카드 사양
Table. 1. Linux ATM kernel and ATM NIC Spec.

ATM 버전	지원 커널	지원 ATM 카드
~atm-0.31	Linux-2.0.29	- Efficient ENI-155P - SMC ATM power 155 - TI TNETA1570 - Zeitnet ZN1221/1225 - IDT 77901/77903 - FORE PCA-200E
atm-0.32~ atm0.38	Linux-2.1.10 5	- Efficient ENI-155P - Zeitnet ZN1221/1225

3.3 리눅스 ATM 프로토콜 스택

리눅스에서는 Berkeley BSD 소켓 API를 이용하여 PVC 및 SVC를 지원하는 native ATM API를 정의하고 일반 소켓 계층과 디바이스 드라이버 간에 프로토콜 기능을 구현하여 ATM 소켓 API(5)가 개발되었다. 리눅스 네트워킹 프로토콜 스택은 그림 2와 같다.

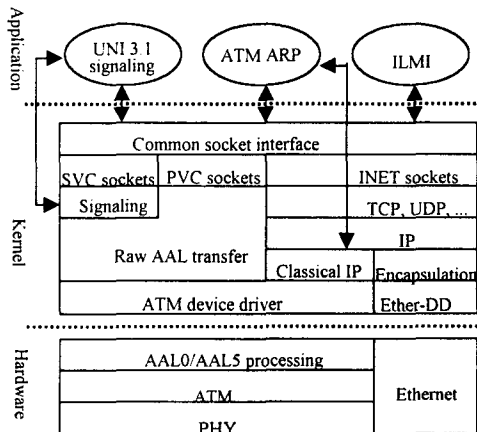


그림. 2. 리눅스 네트워킹 프로토콜 스택
Fig. 2. Linux Networking Protocol Stack

API는 C언어의 데이터 구조(Data Structure)와 함수 호출을 사용하여 기술되며 구조의 사용을 설명하기 위하여 관련 헤더 파일을 인용한다. 연결 설정 및 해제를 위한 연결제어는 다음의 4 단계로 이루어진다.

- connection preparation : socket descriptor 등의 파라미터 설정 및 로컬 자원의 할당이 이루어진다.

어진다.

- connection setup : 대역폭, 버퍼, 연결 식별자 등의 로컬 네트워킹 자원이 할당된다.
- data exchange : 설정된 경로를 통해 데이터의 송수신이 이루어진다.
- connection teardown : 데이터의 송수신이 종료되고 할당된 자원의 해제가 이루어진다.

3.4 시그널링

리눅스 커널은 ATM 시그널링(7,8,9)을 지원하기 위한 복잡한 과정을 사용자 모드의 데몬 프로세스에 의해 수행한다. 내부 신호 프로토콜은 네트워크에서 사용되는 Q.2931 혹은 다른 신호 프로토콜보다 간단한 형태이며 다음과 같은 몇 가지 가정을 기본으로 한다.

- Well-behaved communication : 체크섬, 프로토콜 타이머, 재전송 및 재순차 등의 필요성이 없는 것으로 가정
- Agreement on the protocol : 통신을 위한 프로토콜 버전과 revision 등 프로토콜의 상세 사항에 대해 일치된 동작을 하는 것으로 가정
- Cooperation : 통신을 위한 mapping operation을 단순화

일반적으로 커널과 시그널링 데몬간의 통신 메커니즘은 신뢰성이 높고 메시지가 순차적으로 전송되는 것으로 가정한다. 리눅스 상에서 ATM 시그널링 개념도는 그림 3과 같다.

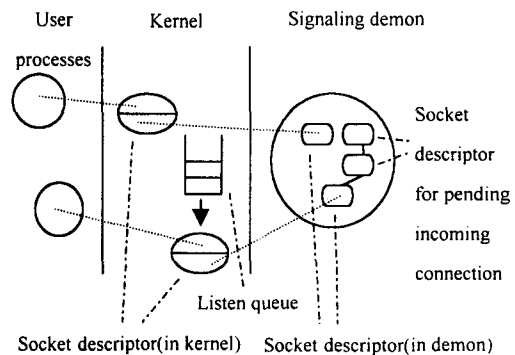


그림. 3. 리눅스 ATM 시그널링 개념도
Fig. 3. Linux ATM Signalling Concept Diagram

- connection preparation : socket descriptor 등의 파라미터 설정 및 로컬 자원의 할당이 이루어진다.

커널과 데몬 모두는 소켓과 연결에 대한 상태 정보(descriptor)를 유지한다. 소켓을 반납할 경우, 이전에 생성된 상태정보가 존재할 경우에는 커널은 데몬에게 반드시 이전에 생성된 상태정보를 데몬에게 알려야 한다. 그림 4는 커널과 시그널링 데몬에서 소켓 descriptor의 관계를 보여준다.

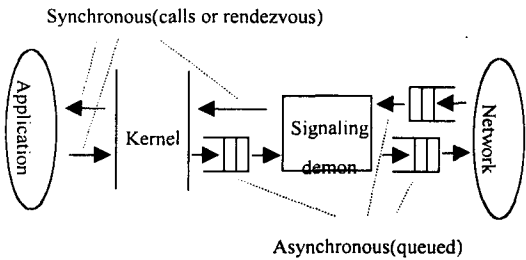


그림 4. 커널과 시그널링 데몬에서의 소켓 descriptor
 Fig. 4. Socket descriptor in kernel and signalling demon

양단간의 내부 시그널링 프로토콜을 위한 메시지 구조는 구조체(Structure)로 구성되어 있으며 각각의 메시지 필드의 파라미터 요구사항은 표 2와 같다.

표 2. 메시지 필드의 파라미터 요구사항
 Table. 2. Parameter Requirement of Message Field

메시지	방향	T1	T2	T3	T4	T5	T6	T7	T8
as_bind	K->D	•			•				•
as_okay	D->K	•							•
as_connect	K->D	•			•		•	•	•
as_okay	D->K	•					•	•	•
as_listen	K->D	•			•				
as_okay	D->K	•							
as_indicate	D->K		•			•		•	•
as_accept	K->D	•	•						
as_okay	D->K	•	•				•		
as_reject	K->D		•	•					
as_error	D->K	•		•					
as_close	K->D	•							
as_close	D->K	•	•						

K -> D : 메시지가 커널에서 시그널링 데몬으로 전송
 D -> K : 메시지가 시그널링 데몬에서 커널로 전송
 T1 : vcc T2 : listen_vcc T3 : reply T4 : aal
 T5 : pvc T6 : local T7 : qos T8 : blli

IV. 다중 ARP 서버

4.1 다중 ARP 서버 요구사항

ATM 기반 Classical IP over ATM 프로토콜의 다중 ARP 서버는 스위치와 라우터가 통합된 미래의 네트워크 형태에서 사용자에게 저렴한 가격과 통신 품질을 보장할 수 있는 Scalable한 대역폭을 제공할 뿐 아니라, 기존의 LAN과 WAN 기술을 통합할 수 있는 기술이다. 특히 다중 ARP 서버는 ATM 교환기의 UNI에 접속되어 LIS내의 CLIP(Classical IP over ATM) 클라이언트들이 IP 주소에 대응된 ATM 주소를 제공하는 역할을 수행하며 LIS간 트래픽을 전달해 주는 라우터의 역할을 선택적으로 수행한다.

다중 ARP 서버가 위치할 일반적인 LIS간의 망 구성은 그림 5와 같다.

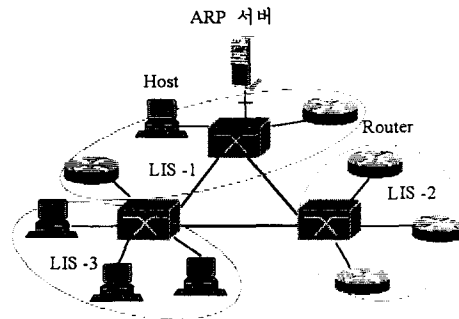


그림 5. LIS간 망 구성
 Fig. 5. Network Configuration between the LISs

다중 ARP 서버를 위한 요구사항은 다음과 같다.

- ATM Forum의 NSAP Address 및 Embedded E.164 AESA 주소 체계 지원 가능
- LIS당 128개 이상의 가상 연결(Vc) 설정이 가능하며, 하나의 ATM 인터페이스에는 1024개 이상의 Vc 설정 가능.
- 하나의 ATM 인터페이스에 16개 이상의 논리적 ARP 서버 구성 가능
- 각 논리적 ARP 서버는 주소체계에 의해 구분 가능
- 각각의 논리적 ARP 서버 생성시 selector value.

- IP 주소 및 서브넷 마스크 값을 입력하도록 구성
- 각각의 논리적 ARP 서버는 클라이언트의 기능을 동시에 수행 가능
- ARP 서버는 RFC1577 및 RFC2225 동시지원 중 선택적 사용 가능
- ARP 서버내 테이블이나 엔트리의 값을 수동으로 추가/삭제/변경 가능
- ARP 서버는 control connection과 data connection의 독립적 설정이 가능
- control/data connection 각각은 UBR 또는 VBR(PCR, SCR) 선택 가능

4.2 다중 ARP 서버 접속

ARP 서버는 클라이언트의 ATM 주소와 IP 주소 값을 등록 받아 저장하고 있다가 클라이언트에게 IP 주소에 대응되는 ATM 주소를 제공하는 역할을 수행한다. 그러나 하나의 ARP 서버는 LIS 범위내의 ATM 주소 정보만을 제공하므로 ARP 서버는 망내 다수의 LIS 구성이 가능하도록 복수의 논리적(logical) ARP 서버 기능을 제공할 수 있어야 한다.

ARP 서버의 구성과 LIS별 클라이언트 연결 형태는 그림 6와 같다.

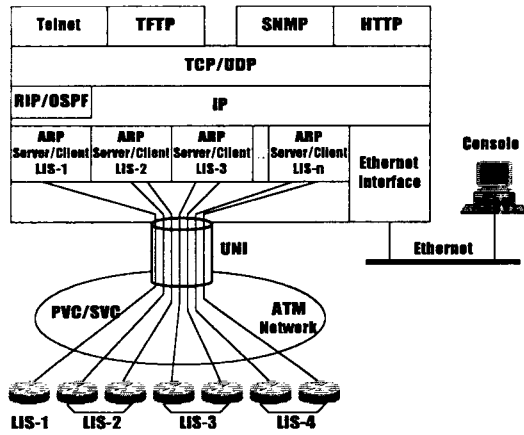


그림. 6. LIS별 클라이언트 연결 형태
Fig. 6. Client Connection for each LIS

이 ARP 서버는 16개 이상의 논리적 ARP 서버 기능을 제공하며 각각의 ARP 서버는 동시에 클라이언트 기능을 수행하여 IP 계층을 통하여 LIS간 라우팅이 가능하도록 한다. ARP 서버 및 클라이언트의 세부 동작은 RFC1577 및 RFC2225를 따른다. 또한 ARP 클라

이언트는 자신이 속해 있는 LIS의 ARP 서버를 지정하는 주소를 manual로 입력하여야 하며 이때 ARP 서버내의 ARP 서버에 할당된 SEL값을 포함한 20 옥텟의 전체 주소를 입력해야 한다.

V. IPOA 기반 망 관리 대행자

IPOA 기반의 ATM 기술을 사용하고 있는 장치나 응용들 대부분의 ATMARP 서버나 클라이언트에서는 지금까지 망관리 기능을 수행하지 않고 독자적인 운용 능력만을 가지고 관리하고 있었다. 그러나 본 논문에서는 이러한 IPOA 망의 구성 요소들을 효율적으로 관리하기 위해 인터넷 표준 망 관리 프로토콜인 SNMP를 통해 ATM 장치 및 ATMARP 서버와 클라이언트 그리고 가상 채널들에 대한 구성, 상태 및 통계 관리를 위한 에이전트를 구현하였다.

5.1 IPOA MIB

IPOA MIB[5]는 IPOA 기반 망에서의 LIS, 클라이언트 그리고 서버에 대한 각 관리 정보를 원격에서 통합 관리하고 제어할 수 있도록 하는 MIB를 정의한다. IPOA MIB 구조는 IPOA 클라이언트와 서버에서 공통으로 지원해야 하는 Basic Support MIB, 클라이언트에서 지원해야 하는 Client Support MIB 및 서버에서 지원해야 하는 ATMARP Server MIB 정의의 3부분으로 나누어져 있으며, 현재 본 논문에서 구현된 관리 정보들은 다음의 표 3과 같다.

표 3 IPOA MIB 관리 객체
Table 3 IPOA MIB Managed Object

관리정보	OID	설명
ipoaLisTrapEnable	46.1.1	LIS 테이블의 생성 유무를 표시
ipoaLisTable	46.1.2	모든 LIS에 대한 구성 정보를 유지
ipoaArpClientTable	46.1.4	ATMARP 클라이언트의 구성 및 입,출력 통계
ipoaArpSvrTable	46.1.5	ATMARP 서버의 구성 및 입,출력 통계
ipoaVcTable	46.1.7	ARP cache에 있는 각 VC에 대한 구성 및 통계 정보
ipoaConfigPvcTable	46.1.8	PVC 구성 정보

5.2 ATM MIB

ATM MIB[4]는 ATM 기반의 인터페이스, 장치, 네트워크 및 서비스를 관리하기 위해 사용되는 관리 객체들을 정의한다. ATM 관리 객체는 ATM 호스트, 스위치 및 네트워크에 의해 지원되는 ATM 인터페이스, ATM 가상 링크, ATM cross-connect, AAL5 엔티티 및 AAL5 연결들을 관리할 위해 사용되며, 여기에서 정의된 MIB는 기본적으로 ATM PVC를 관리하기 위한 것이며, 또한 ATM SVC 역시 이 MIB내 관리 정보에 의해 표현될 수 있지만, 완전한 관리는 추가적인 능력이 필요로 한다. 본 논문에서 구현된 관리 정보들은 다음 표 4와 같다.

표 4. ATM MIB 관리 객체
Table. 4. ATM MIB Managed Object

관리정보	OID	설명
atmInterface-ConfTable	37.1.2	해당 ATM 인터페이스의 ATM 셀 계층 구성 정보
atmVplTable	37.1.6	가상 경로 링크의 구성 및 상태 정보
atmVclTable	37.1.7	가상 채널 링크의 구성 및 상태 정보
aal5VccEntry	37.1.12	해당 ATM 인터페이스 상에서 있는 각 VCC에서의 AAL5 통계 정보

VI. 결론

기존의 IPOA 기반의 망 환경에서는 하나의 LIS에 ARP 서버가 독립적으로 존재하는 것을 기본으로 설계 되었으며 또한 독자적인 관리 능력만을 가지고 ATM 장치 및 응용들의 관리를 수행 함으로서 효율적으로 망의 상황에 대처하기가 어려웠다.

따라서 본 논문에서는 다수 개의 LIS로 구성된 ATM망에서 하나의 ARP 서버가 다수 개의 LIS를 수용하고 RFC1577 및 RFC2225를 지원하는 다중의 ARP 서버를 설계하고 구현하였으며, 또한 인터넷 표준 망 관리 프로토콜인 SNMP를 이용하여 IPOA 기반의 망 구성 요소들의 구성, 상태 및 통계 정보들을 효율적

으로 관리하고 제어할 수 있도록 하는 관리 대행인을 구현함으로써 망 관리자가 효율적으로 망의 상황에 대처할 수 있도록 하였다.

참고문헌

- [1] Almesberger, werner. "ATM on Linux". EP FL, Mar. 1997
- [2] M. Laubach. "Classical IP and ARP over ATM." RFC1577, IETF, Jan. 1994.
- [3] ATM Forum, LAN emulation version 1.0 specification, Jan. 1995.
- [4] M. Ahmed, K. Tesink , Definitions of Managed Objects for ATM Management Version 8.0 using SMiv2, RFC1695, IETF, August, 1994
- [5] M. Greene, J. Luciani, K. White, T. Kuo, Definitions of Managed Objects for Classical IP and ARP Over ATM using SMiv2, RFC2320, IETF, April, 1998.
- [6] Almesberger, werner. Linux ATM API, EPFL, Jul. 1996.
- [7] The ATM Forum, Technical Committee. "ATM User-Network Interface Signaling Specification version 4.0", The ATM Forum, Jul. 1996.
- [8] ITU-T Recommendation Q.2931. B-ISDN DSS2 User-network interface layer 3 specification for basic call/connection control, ITU, Feb. 1995.
- [9] ITU-T Recommendation Q.2971. B-ISDN DSS2 User-network interface layer 3 specification for point-to-multipoint call /connection control, ITU, Oct. 1995.
- [10] J. Heinanen, "Multiprotocol Encapsulation over AAL 5," RFC1483, IETF, Jul. 1993.

저자소개



이병기

- 1994년 : 광운대학교대학원 전자계산학과 (이학석사)
- 1994년 ~ 1996년 : 코오롱정보통신 기술연구소
- 1996년 ~ 현재 : 광운대학원 전자계산학과 박사과정
- 1997년 ~ 현재 : 가우리정보통신 선임연구원
- 1997년 ~ 현재 : 한국전자통신연구원 위촉연구원