

# ATM 상에서 다양한 프로토콜을 지원하기 위한 MPOA의 구현

임 지 영<sup>†</sup>·김 미 희<sup>††</sup>·최 정 현<sup>†</sup>·이 미 정<sup>†††</sup>·채 기 준<sup>††††</sup>·최 길 영<sup>†††††</sup>·강 훈<sup>††††††</sup>

## 요 약

본 논문에서는 기존 LAN 응용뿐 아니라 다양한 3계층 프로토콜을 ATM 망상에서 서비스해주기 위하여 ATM 포럼에서 제안한 MPOA(MultiProtocol Over ATM)를 구현하고 이를 테스트하였다. MPOA 호스트와 엣지 디바이스의 기본이 되는 MPC(MPOA Client)와 라우터 상에 존재하는 MPS(MPOA Server)의 작동 프로시저에 중점을 두어 구현하였다. 구현한 MPC의 기능은 LEC(LAN Emulation Client)와 프리미티브 교환 작업, Ingress와 Egress 캐쉬의 관리와 유지, 그리고 LEC를 통한 디폴트 전송과 지름길 전송들이다. 또한 MPS는 LEC와 프리미티브 교환, MPOA와 NHRP(Next Hop Resolution Protocol)사이의 프레임 변환 및 교환 그리고 Ingress/Egress 캐쉬 관리와 유지 등을 구현하였다. 구현된 MPOA시스템이 올바르게 작동하는지 테스트하고자 가능한 모든 시나리오를 작성하여 시뮬레이션에 의해 테스트하였다.

## Implementation of MPOA for Supporting Various Protocols over ATM

Ji-Young Lim<sup>†</sup> · Mi-Hee Kim<sup>††</sup> · Jeong-Hyun Choi<sup>†</sup> · Mee-Jeong Lee<sup>†††</sup> ·  
Ki-Joon Chae<sup>††††</sup> · Kil-Young Choi<sup>†††††</sup> · Hun Kang<sup>††††††</sup>

## ABSTRACT

In this paper, we implemented and tested MPOA(MultiProtocol Over ATM) standardized in ATM Forum, which provides service for various layer 3 protocols as well as legacy LAN applications over ATM networks. The functions of MPCs(MPOA Clients) and MPSs(MPOA Servers) which are the components in MPOA systems are implemented. MPCs are located at the edge device and MPOA hosts and MPSs exist in routers. The implemented MPCs have the functions such as exchanges of primitives between an LEC(LAN Emulation Client) and an MPC, management and maintenance of Egress/Ingress cache, default transmission through LECs and shortcut transmission. Assuming that routing, convergence and NHRP(Next Hop Resolution Protocol) functions exist in routers, the implemented MPSs have the functions such as exchanges of primitives between an LEC and an MPC, conversion and exchanges of frames between MPOA and NHRP, and management and maintenance of Egress/Ingress cache. All of the possible scenarios are made up to test whether they run correctly. The implemented system is tested by simulation according to the scenarios.

\* 본 연구는 한국전자통신연구원 교환전송기술연구소 라우터기술 연구부 위탁연구과제에 의한 것임.

† 준 회원 : 이화여자대학교 대학원 컴퓨터학과

†† 정 회원 : 이화여자대학교 대학원 컴퓨터학과

††† 준 회원 : 이화여자대학교 컴퓨터학과 교수

†††† 종신회원 : 이화여자대학교 컴퓨터학과 교수

††††† 정 회원 : 한국전자통신연구원 선임연구원

†††††† 정 회원 : 한국아이티 벤처 투자(주) 상무이사

논문접수 : 1999년 3월 30일, 심사완료 : 1999년 12월 27일

## 1. 서 론

근거리 통신망은 Ethernet, Token Ring 및 FDDI가 주류를 이루고 있으며, 망을 이용한 대부분의 응용들은 IP, IPX, AppleTalk, SNA 등의 망계층 프로토콜을 사용하여 운용되고 있다. 이러한 기존 LAN에 비하여 실시간 전송 및 대역폭 면에서 월등한 장점을 가지고 있는 ATM 망이 등장하면서 이를 효과적으로 이용할 수 있는 응용들에 대한 연구가 활발히 이루어지고 있다. 또한 이러한 ATM 망과 기존 LAN 간의 연동을 위하여 ATM 포럼에서는 기존의 Ethernet, Token Ring, TCP/IP와 같은 다양한 상위 계층 프로토콜을 그대로 유지하며 망계층 라우팅을 제공하는 MPOA(Multi-Protocol over ATM)에 관한 연구가 진행되고 있으며, 스펙 버전 1.0이 완성되었다[1].

MPOA는 다양한 망계층을 수정 없이 사용하는 LANE(LAN Emulation)의 장점을 그대로 수용하면서 적용 범위가 하나의 ELAN(Emulated LAN)으로 제한되는 단점을 보완하기 위하여 3계층 주소 해석에 의한 직접적인 ATM 연결을 제공하는 NHRP(Next Hop Resolution Protocol)를 도입하였다[2]. 또한 주소 해석과 포워딩을 각각 라우터와 엡지 디바이스로 나누어 수행하는 가상 라우터의 개념을 도입하였다. 즉, MPOA는 다양한 프로토콜과 망 기술을 지원하면서 ATM의 효과적인 브릿징과 라우팅을 종합하는 하나의 프레임워크를 제공함으로써 LANE 환경에서 서브넷간의 효과적인 유니캐스트 데이터 전송을 지원한다.

Fore 시스템과 같은 외국 회사에서는 MPOA를 구현하여 제품으로 개발하였으나 아직 국내에서는 제품화된 상태는 아니다. 또한 MPOA에 대한 성능 평가에 대한 연구는 표준안에 입각하여 표준안에서 허용하는 범위 안에서 MPOA 매개변수의 값을 변경하면서 그 성능을 비교한 정도로서 아직 MPOA에 대한 성능 평가에 대한 연구가 많이 이루어진 상태가 아니다[3, 4]. 뿐만 아니라 MPOA 구현에 관련된 논문도 부재하다. 본 연구의 목적은 MPOA 기반 라우터 장비 개발을 위한 MPOA의 기능인 MPC와 MPS를 구현에 있다. 또한 이의 작동을 테스트하기 위해서 표준안에 따라 MPOA 시스템에서 가능한 시나리오를 작성하여 각 단계의 상태를 표준안과 비교하였다.

본 논문에서는 ATM 포럼의 MPOA 표준안에 입각하여 MPOA의 구성 요소인 MPS(MPOA Server)와 MPC

(MPOA Client)의 기능, LEC(LAN Emulation Client)와 MPOA 구성 요소와의 인터페이스, MPS와 NHS(NHRP Server)의 인터페이스를 프로세스로서 워크스테이션에서 구현하였다. 또한 4개의 MPC와 2개의 MPS를 갖는 MPOA 시스템을 모델로 하여 MPOA 기능이 올바르게 동작하는지 알아보기 위해 시뮬레이션하였으며, 인터넷워크 계층을 IP로 하여 구현한 MPOA 기능을 테스트하였다.

본 논문의 구성은 다음과 같다. 제2장에서는 MPOA의 개념에 대해서 알아보고, 제3장에서는 MPOA의 설계 및 구현에 대하여 설명한다. 제4장에서는 구현한 MPS와 MPC간의 작동 시나리오를 통한 실험에 대하여 설명하고 마지막으로 제5장에서는 본 논문의 결론 및 향후 연구 계획에 대하여 기술한다.

## 2. MPOA의 개요

### 2.1 MPOA의 정의

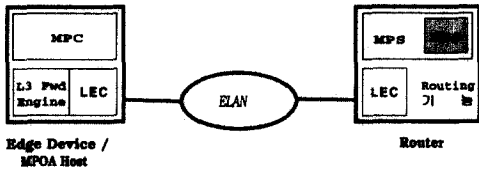
MPOA란 ATM 망상에서 3계층 프로토콜의 지름길 전송을 위해 ATM 포럼의 멀티프로토콜 서브워킹 그룹에서 개발한 방안이다. MPOA는 다양한 프로토콜과 다양한 망 기술, 가상랜 환경에서 ATM의 효과적인 브릿징과 라우팅을 종합하는 하나의 프레임워크를 제공함으로써 LANE 환경에서 서브넷간의 유니캐스트 데이터를 효율적으로 전송하는데 목적을 두고 있다. ATM 포럼의 LANE는 ATM 망상에 서브넷 내의 데이터를 전송하는 효율적인 수단을 제공하지만, 서브넷간의 데이터는 여전히 라우터를 통한 포워딩이 필요하다. MPOA도 2계층의 포워딩을 위해 LANE를 사용하지만, LANE가 기본적으로 하나의 LAN 서브넷을 ATM 망상에 연결하는 반면, MPOA는 다른 서브넷의 호스트들 사이에 직접적인 ATM 연결을 허용한다. MPOA는 서브넷간 지름길 설정에 있어서 라우터를 거치지 않고 서브넷간 ATM VCC를 설정함으로써 직접적인 연결을 설정해 주는 ION(Internetworking Over Non-Broadcast Multi-Access) 워킹 그룹의 NHRP를 사용한다. 이처럼 MPOA는 기존의 LANE와 NHRP의 통합으로 LANE의 이점과 라우터를 통하지 않고 지름길 전송을 가능하게 하는 NHRP의 이점을 통합하여 LANE 환경에서 효율적인 서브넷간의 유니캐스트 데이터 전송의 목적을 달성하고 있다.

MPOA는 지름길 전송에 필요한 ATM 주소 해석을

요청하는 MPC(MPOA Client)와 3계층 라우팅 계산을 통해 질의에 응답하는 MPS(MPOA Server)를 구성 요소로 하고, MPC들과 MPS들이 통신하는데 필요한 프로토콜을 정의하였다. 또한 MPC는 3계층 포워딩을 수행하게 하고, MPS는 3계층 라우팅 계산을 하게 하여 두 기능의 물리적 분리라는 가상 라우팅 기술을 도입하였다. 가상 라우팅 기술은 효율적인 서브넷간의 통신을 가능하게 하고, 3계층 라우팅 계산을 수행하는 디바이스의 수를 줄일 뿐 아니라, 옛지 디바이스는 라우팅이 아닌 포워딩 기능만을 수행함으로써 그 복잡성을 줄일 수 있다.

2.2 MPOA 구성 요소

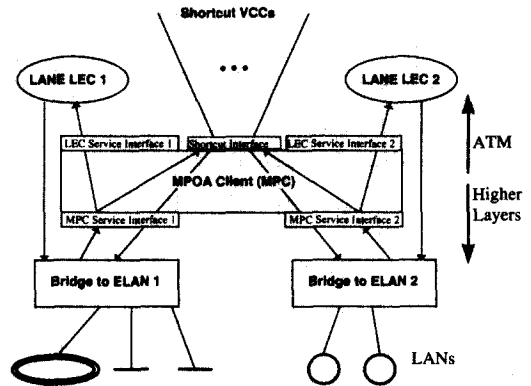
MPOA의 논리적 구성 요소는 (그림 1)과 같이 MPC와 MPS이다. MPC는 패킷을 상위 계층으로 전송하는 포워딩 엔진과 LEC와 함께 옛지 디바이스에 존재하며 내부에 NHS의 기능을 포함하는 MPS는 라우터의 기능인 라우팅 기능과 LEC와 함께 라우터에 존재한다[1].



(그림 1) MPOA 시스템 구성 요소

MPC의 주요한 기능은 3계층의 지름길 설정과 해제이며, 3계층의 포워딩을 수행한다. (그림 2)에서와 같이 MPC의 기본적인 작동 원리는 MPOA 시스템으로 들어가는 즉, ATM 망으로 나가는 측면의 Ingress 규칙과 MPOA 시스템에서 나오는 즉, 상위 계층으로 나가는 측면의 Egress 규칙으로 나누어 생각할 수 있다. Ingress 규칙에서는 MPC가 MPS가 있는 라우터로 포워딩되는 패킷을 발견하면, LANE에 의해 이미 설정된 경로로 전송(다플트 전송)할 때보다 지름길 전송을 했을 때가 더 유리한지를 체크한다. 만약에 지름길 전송이 유리하다면, NHRP에 기초한 질의-응답 프로토콜을 사용하여 목적지에 지름길 설정을 요청한다. 지름길 설정이 가능하다면, Ingress 캐쉬에 그와 관련된 정보를 저장하고 지름길 VCC를 통해 목적지에 패킷을 전송한다. 만약 지름길이 설정되지 않는다면, 해당 LEC 서비스 인터페이스를 통해 LEC로 보내진다. Egress 규칙

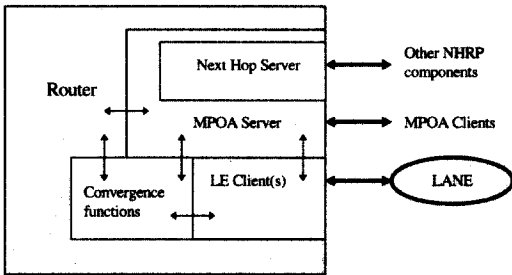
에서는 MPC가 다른 MPC들로부터 자신의 지역 인터페이스나 사용자에게로 전송되는 패킷을 받았을 때, 만약 그것이 지름길을 통한 전송이었다면 MPS로부터 DLL(Data Link Layer) 인캡슐레이션 정보를 받아서 데이터에 첨가, 상위 계층으로 그것을 포워딩하고 관련 정보를 MPC Egress 캐쉬에 저장한다. 그러나 그것이 지름길이 아닌 LEC 서비스 인터페이스를 통해 받은 것이라면 MPC의 관여 없이 바로 상위 계층으로 보내진다.



(그림 2) MPOA 옛지 디바이스 MPC의 예

MPS는 MPC에게 3계층 포워딩 정보를 제공해 주는 논리적 라우터 구성 요소이며, (그림 3)과 같이 위치한다. MPS는 포함된 NHS(Next Hop Server)를 통해 ELAN 간의 주소 해석 기능을 하고, Ingress MPC에게는 라우팅 기능을 통해 MPOA 질의에 대한 답을 제공해 주며 Egress MPC에게는 DLL 인캡슐레이션 정보를 제공한다. 또한 자신과 관련된 LEC에게 ATM 제어 주소를 알려주고, LEC들에게 이 주소를 포함하는 장치형 TLV(Type/Length/Value)를 LE\_REGISTER\_REQUEST, LE\_ARP\_REQUEST, LE\_ARP\_RESPONSE와 같은 LANE의 등록 요청 또는 주소 해석 관련 제어 프레임에 포함시켜 이러한 LANE 제어 메시지의 수신자로 하여금 ATM 제어 주소와 LEC가 MPS에 연결되어 있다는 것을 알려 준다. 또한 NHS를 포함하고 있기 때문에 다른 NHS(Next Hop Server)나 NHC(Next Hop Client)들로부터 NHRP 질의 요청을 받는다. 만약 질의 요청에 대한 주소 해석 결과인 다음 홉이 직접 연결된 MPC라면 그 MPC에게 MPOA 캐쉬 저장 요청을 보낸다. 그렇지 않으면 그 요청은 표준 NHRP 주소

해석 요청으로 처리되어 다음 NHS로 포워드된다. MPS는 Ingress/Egress 캐쉬 엔트리 상태 정보 뿐 아니라 요청에 대한 응답 정보도 저장해야 한다. 유효 시간이 지났거나 호스트 이동 또는 라우팅 정보의 변경으로 목적지 정보가 무효화되면, 주소 해석 요청의 근원지에게 통고해야 한다.



(그림 3) 라우터 MPS의 예

### 2.3 MPOA의 데이터와 제어 정보의 흐름

모든 데이터와 제어 정보의 흐름은 LLC/SNAP(Logical Link Control/SubNetwork Attachment Point) 인캡슐레이션을 사용하여 ATM VCC상에서 전송하는 것을 기본으로 한다. 구성 정보 흐름은 LANE의 포맷을 사용하며 MPS들과 MPC들은 자신의 구성 정보를 얻기 위해 LECS(LAN Emulation Configuration Server)와 통신한다.

제어 정보는 MPC-MPS와 MPS-MPS, MPC-MPC 간에 서로 교환하며 각각은 다음과 같다. MPC-MPS 간의 제어 정보 흐름은 MPC 캐쉬 관리를 위해 사용된다. Ingress MPC는 지름길 설정을 위해 MPOA 주소 해석 요청/응답을 이용하여 Ingress MPS와 통신한다. 이 때 Ingress MPS는 MPOA Trigger 메시지를 통해 Ingress MPC에게 지름길 설정 요청을 유발시킬 수도 있다. 이에 반해 Egress MPS는 Egress MPC에게 캐쉬 정보를 제공하기 위하여, MPOA 캐쉬 저장 요청/응답을 사용한다. Egress MPC나 Egress MPS가 유효하지 않은 캐쉬 저장 정보를 발견하면 엔트리 제거를 위해 메시지를 보내어 지름길 해제를 시도한다. MPS-MPS간의 제어 정보 흐름은 MPOA가 새로운 프로토콜을 정의하지 않고 기존의 NHRP와 표준 3계층 라우팅 프로토콜에 의해 처리한다. Egress MPC는 Ingress MPC로부터의 데이터가 자신이 서비스하는 호스트의 것이 아닐 때 데이터 플레인 제거 메시지를 보

내서, 캐쉬 정보를 무효화시킨다.

데이터 흐름은 주로 MPOA 지름길 VCCs를 통해 MPC들 사이의 데이터 전송을 위해 사용된다. 또한 MPC와 NHC간에 유니캐스트 데이터 전송을 위해 사용될 수도 있다.

## 3. MPOA의 구현

본 논문에서는 MPOA 기반 스위치드 라우터 개발에 일환으로 LANE를 지원하는 ATM LAN 환경에서 MPOA를 지원하기 위해 이 라우터에 탑재할 MPS와 MPC의 구현을 목적으로 하여 표준안에 입각하여 MPC와 MPS를 구현하였다. 개발할 라우터에 탑재된 언어가 C와 유사하므로 본 구현은 C로 하였다.

### 3.1 MPOA 프레임 포맷과 자료구조

MPOA 시스템은 기본적으로 RFC 1483에서 정의한 LLC 인캡슐레이션을 사용하므로 지름길을 통한 데이터 전송 시에 기본적으로 RFC 1483의 LLC 인캡슐레이션을 사용하고[5], 태그가 붙는 MPOA 인캡슐레이션 방법 또한 사용할 수 있다[1]. 이 때 태그 필드는 패킷의 3계층 프로토콜 타입을 결정하는데 사용되며 이 두 타입의 인캡슐레이션은 하나의 VCC상에서 혼용되어 사용할 수 있다. MPOA는 모든 제어 메시지에 대해 기본적으로 NHRP에서 정의한 LLC 인캡슐레이션을 사용한다[1].

LEC는 상위 계층으로부터 데이터 또는 제어 정보를 제공하고 받기 위한 방법으로 정해진 프리미티브를 사용하며, 이 프리미티브에 정해진 필드 외에 전달할 내용은 TLV 리스트로서 전달하게 된다[6]. 하나의 TLV 리스트는 Type, Value에 대한 Length와 Value 필드로 구성되며, 상위 계층으로부터 받은 TLV 리스트는 LEC가 요청 또는 응답 프레임의 헤더 뒤에 이 내용을 부쳐 전달한다. MPOA 구성 요소가 LEC의 상위 계층에 존재하므로 MPOA 구성 요소 또한 이 방법을 사용하여 LEC와 통신한다. MPOA 구성 요소는 MPC와 MPS의 구성 TLV와 장치형 TLV의 세 가지의 TLV를 갖고 있다[1].

MPOA 제어 메시지에 는 MPOA 주소 해석 요청, MPOA 주소 해석 응답, MPOA 캐쉬 저장 요청, MPOA 캐쉬 저장 응답, MPOA Egress 캐쉬 제거 요청, MPOA Egress 캐쉬 제거 응답, MPOA Keep-Alive, MPOA

Trigger, NHRP 제거(데이터 플레인에서 사용)가 있으며 이들은 모두 NHRP 제어 메시지 양식과 같게 정의된다[2]. MPOA에서 사용하는 확장양식으로는 MPOA DLL 헤더 확장, MPOA Egress 캐쉬 태그 확장, MPOA ATM 서비스 범주 확장, MPOA Keep-Alive Lifetime 확장, MPOA Hop Count 확장, MPOA Original Error Code 확장이 있다[2]. MPOA 제어 메시지들은 서로 다른 프레임 포맷을 가지며 다음은 본 구현에서 사용하는 각 필드의 길이에 맞게 만든 자료구조를 보여준다.

#### MPOA 주소 해석 요청

```
struct packet {
    int                input_type;
    struct fixed_header fix_header;
    struct common_header com_header;
    struct cie         cies;
    struct cache_tag   tag_ext;
    struct service     serv;
};
```

#### MPOA 주소 해석 응답, MPOA 캐쉬 저장 요청, MPOA 캐쉬 저장 응답

```
struct packet_dll {
    int                input_type;
    struct fixed_header fix_header;
    struct common_header com_header;
    struct cie         cies;
    struct cache_tag   tag_ext;
    struct service     serv;
    struct dll_h       dll_hd;
};
```

#### MPOA Egress 캐쉬 제거 요청, MPOA Egress 캐쉬 제거 응답

```
struct packet_dh {
    int                input_type;
    struct fixed_header fix_header;
    struct common_header com_header;
    struct cie         cies;
    struct dll_h       dll_hd;
};
```

#### MPOA Keep-Alive

```
struct packet_keep {
    int                input_type;
```

```
    struct fixed_header fix_header;
    struct common_header com_header;
    struct keep_alive   ka;
};
```

#### MPOA Trigger

```
struct trigger {
    int                input_type;
    struct fixed_header fix_header;
    struct common_header com_header;
};
```

#### NHRP 제거(데이터 플레인에서 사용)

```
struct data_plane {
    int                input_type;
    struct fixed_header fix_header;
    struct common_header com_header;
    struct cie         cies;
};
```

### 3.2 MPC

MPC의 기능은 Ingress MPC로서의 기능과 Egress MPC로서의 기능으로 나누어 볼 수 있으며 각각은 3계층 주소와 ATM VCC의 매핑에 관련된 정보를 유지하기 위하여 캐쉬를 관리한다. Ingress MPC는 먼저 장치 발견 절차를 통해 MPOA 주소 해석 요청을 보낼 Ingress MPS의 ATM 주소를 얻은 후, 지름길 설정을 위해 상대 ATM 주소를 얻기 위한 주소 해석을 수행한다. 주소 해석 요청에 대한 응답이 오면 Ingress 캐쉬 엔트리를 생성하고, NHRP 캐쉬 제거 요청을 받으면 캐쉬 엔트리를 삭제한다. Egress MPC는 Egress MPS로부터 캐쉬 저장 요청을 받아 Egress 캐쉬 엔트리를 생성, 삭제, 갱신한다. Ingress MPC와 Egress MPC는 모두 자신이 유지하고 있는 캐쉬 엔트리를 에이징(aging)한다.

#### 3.2.1 구성(Configuration)

구성 단계는 LANE의 서버중 하나인 LECS(LAN Emulation Configuration Server)로부터 구성 매개변수를 얻기 위해 수행되는 단계로서 MPOA MIB를 통하거나 또는 다른 매커니즘을 통해서도 가능하다. 그러나 현재 표준안에서는 LEC의 상위 계층으로서의 서비스 정의가 미흡하므로 "cache.h"라는 파일에서 구성 매

개변수 값을 얻는 방법으로 구현하였다. MPC는 LEC에 의해 얻어진 구성 TLV 리스트로부터 구성 매개변수와 상수를 각각 얻는다[1].

Ingress MPC가 지름길 설정을 위해 필요로 하는 구성 매개변수는 MPC\_p1과 MPC\_p2이다. MPC\_p1의 이름은 "Shortcut-Setup Frame Count"이며 지름길 설정 전에 LEC로 디폴트 전송되는 패킷 수를 나타내는데, MPC가 MPC\_p2(Shortcut-Setup Frame Time)시간 동안 같은 목적지로 MPC\_p1개의 프레임을 전송했다면 지름길을 설정하기 위한 절차를 시작한다. 또한 재시도 메커니즘을 위한 변수로서 MPC\_p5(Retry Time Maximum)와 MPC\_p6(Hold Down Time)을 두어 MPC\_p5 시간 동안 MPOA 주소 해석 응답이 오지 않으면 첫 번째 재시도 절차를 시작하며, 현재 재시도 시간에 MPC\_c1(Retry Time Multiplier)을 곱한 만큼의 시간이 되면 매번 재시도하게 된다. 마침내 MPC\_p6에 이를 때까지 MPOA 주소 해석 응답이 오지 않으면 재시도는 실패한 것으로 간주한다.

Egress MPC의 경우 Egress MPS로부터 Keep-Alive 메시지를 주기적으로 받아 자신에게 Egress 캐쉬 엔트리를 부여한 Egress MPS가 잘 작동 중인지 확인한다. 이 때 첫 번째 Keep-Alive 메시지를 받기 전에 사용하는 Keep Alive Lifetime으로 MPC\_c2가 쓰인다.

### 3.2.2 장치 발견(Device Discovery)

Ingress MPC는 모든 Inbound 패킷(상위 계층으로부터 LEC로 향하는 패킷)에 대해 캐쉬 엔트리가 존재하지 않고 캐쉬 엔트리가 임계치를 넘었을 때 MPOA 주소 해석 요청을 보내기 위해서 Ingress MPS의 ATM 주소를 알아야 한다. 따라서 Ingress MPC는 모든 Inbound 패킷의 MAC 주소를 조사하고, 장치 발견 절차를 통해 MPOA 주소 해석 요청이 보내질 Ingress MPS의 제어 ATM 주소를 얻는다.

MPC는 이런 장치 발견을 위해 LANE의 상위 계층으로서의 서비스를 이용하며, Ingress MPS의 ATM 주소를 얻기 위해서 LEC에게 LE\_ASSOCIATE\_request를 전송을 하면 LEC가 LE\_ASSOCIATE\_indication을 통해 필요한 MPS의 ATM 주소를 보내 주게 된다.

### 3.2.3 목적지 주소 해석(Target Resolution)

Ingress MPC는 ELAN에 있는 MPS들의 MAC 주소로 향하는 3계층 목적지 주소 기반의 패킷 흐름을

감지한다. 라우터를 통한 디폴트 전송을 하다가 같은 목적지로 MPC\_p1개의 패킷을 디폴트 전송하였다면 지름길 전송이 유용하다고 판단하여 목적지 ATM 주소를 얻기 위해 Ingress MPS에게 MPOA 주소 해석 요청을 보낸다. 이 요청에 대한 응답에는 Egress 장치의 ATM 주소와 인캡슐레이션, 태깅 정보가 포함된다. 한편 Ingress MPS가 Inbound 데이터 흐름을 감지하여 Ingress MPC에게 MPOA 주소 해석 요청을 개시하게 할 수도 있다. 이러한 주소 해석 요청으로 인하여 Ingress/Egress MPC와 MPS의 캐쉬가 생성되고 이들 캐쉬는 정보의 유지를 위하여 갱신 또는 삭제된다.

### 3.2.4 데이터 전송

MPOA 시스템은 효율적인 유니캐스트 데이터 전송을 위해, 한 목적지로의 전송할 데이터 양이 많지 않을 때는 LEC를 통한 이미 정해진 경로로 전송을 하고, 전송량이 많아 목적지로의 지름길 전송이 유리하다고 판단했을 때는 MPOA 목적지 주소 해석 메커니즘과 캐쉬 관리 메커니즘을 통해 지름길을 설정하여 전송한다. 이 때 MPOA 애플릿 디바이스는 디폴트 전송 시에는 2계층 브릿지로서 역할을 하고, 지름길 전송일 때는 3계층 포워드 기능을 한다.

### 3.2.5 캐쉬 관리

#### • Ingress MPC의 캐쉬 엔트리 생성

본 논문에서는 Ingress MPC의 캐쉬 엔트리 필드를 <표 1>과 같이 구성하였다. Ingress MPC의 캐쉬가 생성되는 경우는 두 가지로, 그 중 첫 번째의 경우는 일반적인 경우로 상위 계층에서 패킷을 받아 발생하는 경우이다. 패킷이 MPOA 시스템 내부로 들어오면 이 패킷이 다른 ELAN으로 가는 패킷인지 아닌지를 구분하여 다른 ELAN으로 가는 것이면 MPC가 처리하고 그렇지 않으면 LEC로 이 패킷을 보내어 전송하게 한다((그림 4)의 rec\_ip\_packet() 부분). MPC는 받은 패킷을 캐쉬에 있는지 조사하여((그림 4)의 search() 부분), 없으면 새로운 캐쉬 엔트리를 생성하고((그림 4)의 make\_c\_entry() 부분), 만약 받은 패킷의 내용이 캐쉬에 있으면 카운터를 증가시키며((그림 4)의 countup() 부분) 이러한 패킷은 LEC로 보내어 전송하게 된다. 그러나 이 패킷의 캐쉬 엔트리가 임계값을 초과하면 지름길 설정을 위해 Ingress MPS에게 MPOA 주소 해석 요청을 하고 응답으로 캐쉬 엔트리의 내용을 완전히

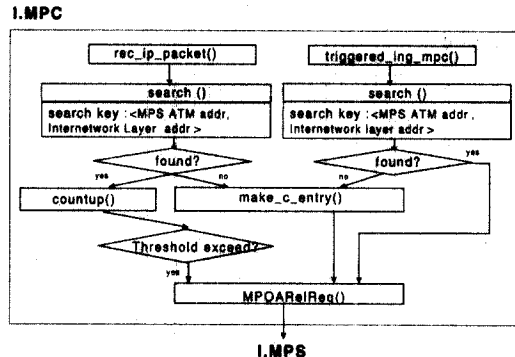
<표 1> Ingress 캐쉬 엔트리 필드 구성

이름	비고	설명
	Ingress 캐쉬 엔트리 Key	MPC로 들어오는 모든 Inbound 패킷들이 Ingress MPS의 MAC주소를 가졌는지 조사하고, 이 MAC 주소를 사용한 장치 발견 절차를 통해 연관된 제어 ATM 주소를 알아낸다. Inbound 패킷이 향하는 3계층 목적지 주소.
struct ing_state	entry_ste	Ingress 캐쉬 엔트리의 Resolving/Resolved/ Invalid 상태를 나타낸다.
	vcc_ste	지름길의 Open/ Closed 상태를 나타낸다.
	struct cies	MPOA 주소 해석 응답 시에 보내진 최근의 CIE코드이다. cies에 있는 hold_time은 MPOA 주소 해석 응답 시의 정보가 유효한 기간을 나타낸다. 추후에 Holding Time 을 사용하여 상태 정보가 Resolved인 엔트리를 Aging 한다.
ser_category		MPOA 주소 해석 응답 시에 ATM 서비스 범주 확장이 포함되었다면 지름길 VC 설정시 사용을 위해 저장한다.
struct ret	req_id	MPOA 주소 해석 응답이 도착하지 않은 MPOA 주소 해석 요청에 대한 요청 ID로서 계속적으로 재시도할 때 사용한다.
	retry_time	MPOA가 주소 해석 요청을 재시도 하기 전에 기다려야 하는 시간을 나타낸다.
f_count		특정 캐쉬 엔트리를 사용하여 포워드한 패킷 수를 나타낸다.
tag		지름길 VCC로 포워드하는 데이터를 인캡슐레이션하기 위해 사용한다.

채우게 된다(그림 4)의 MPOArelReq() 부분). 두 번째의 경우는 Ingress MPC가 Inbound 데이터 흐름을 감지하여 Ingress MPC에게 MPOA 주소 해석 요청을 개시하게 함에 의해 일어날 수 있다(그림 4)의 triggered\_ing\_mpc() 부분).

• Egress MPC의 캐쉬 엔트리 생성

본 논문에서는 Egress MPC의 캐쉬 엔트리 필드를 <표 2>와 같이 구성하였다. Egress MPC는 Egress MPS로부터 MPOA 캐쉬 저장 요청을 받아(그림 5)의 CacheImpRequest() 부분) 이에 대한 응답을 보내야 하는데 자신의 자원이 캐쉬 엔트리를 유지하고, 새로운

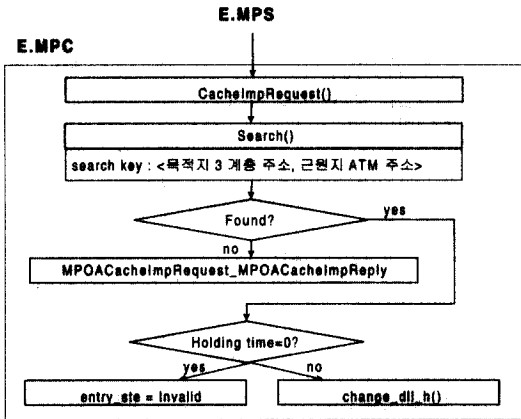


(그림 4) Ingress 캐쉬 생성과 주소 해석 요청

<표 2> Egress 캐쉬 엔트리 필드 구성

이름	비고	설명
	Egress 캐쉬 엔트리 Key	3계층 목적지 주소. 근원지 ATM 주소. 옵션으로서 각 캐쉬 엔트리에 유일하게 결정되어 캐쉬 엔트리 조사에 키의 전체 혹은 부분으로 이용된다.
struct eg_state	entry_ste	Resolved/Purge/Invalid 값을 갖고 엔트리가 Resolved 상태일 때에만 지름길 VCC로 전송된 패킷이 포워드된다.
	mpps_atm_addr[5]	Egress 캐쉬 엔트리를 부여한 Egress MPS를 나타낸다. Egress MPC에게 Egress 캐쉬 제거를 보내기 위한 통신을 유지하기 위해 사용된다.
h_time		Resolved 상태인 각 캐쉬 엔트리를 Aging하는데 사용된다.
f_count		패킷이 지름길 VCC로 전송되어 MPOA 시스템 바깥으로 나가는 MPC 서비스 인터페이스로 포워드하기 전에 카운트 값을 증가시킨다.
mpps_ctl_addr[5]		Egress MPC가 Egress MPS로부터 Keep Alive 메시지를 Lifetime내에 받지 못한다면 그 MPS로부터 생성된 모든 Ingress 캐쉬 엔트리들의 제거를 Ingress MPC에게 알린다.
struct dll_h	제어 메시지 패킷에서 기술	지름길 VCC로 전송된 패킷의 3계층 목적지주소에 대한 DLL 헤더.

VCC를 받아들일 만큼 충분한 지를 살펴야 한다(그림 5)의 Search() 부분). MPOA 캐쉬 저장 요청이 이미 존재하는 캐쉬 엔트리에 대한 갱신의 경우라면 자원 고갈 문제는 없을 것이다. Egress MPC가 캐쉬에 엔트리를 추가할 여유가 없거나 새로운 VCC를 받아들일 수 없다면 적절한 에러 상태를 표시해서 MPOA 캐쉬 저장 응답을 보낸다. 새로운 캐쉬 엔트리를 받아들일 만한 자원이 있는 경우에는 캐쉬 엔트리를 생성한 후, Egress MPC의 ATM 주소와 성공했다는 상태를 담은 MPOA 캐쉬 저장 응답을 Egress MPS에게 보낸다. 캐쉬 저장 요청에 MPOA 태그 확장이 포함되었다면 캐쉬 저장 응답에도 포함되며 MPOA 서비스 범주 확장이 요청에 포함되었다면, Egress MPC가 사용하는 서비스 범주를 캐쉬 저장 응답에 명시해 준다(그림 5)의 MPOACacheImpReuest\_MPOACacheImpReply() 부분).



(그림 5) Egress 캐쉬의 생성

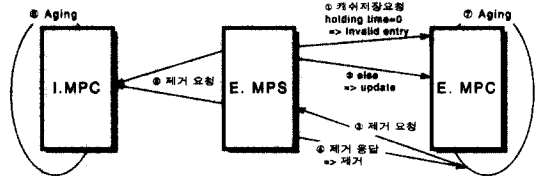
- Egress 캐쉬 엔트리의 갱신(그림 6)에서 ①, ②에 해당)

MPS로부터 받은 MPOA 캐쉬 저장 요청이 이미 존재하는 캐쉬 엔트리에 대한 것이고 0이 아닌 holding time을 담고 있으면, 그 엔트리에 대한 갱신을 요청한 경우이다. Egress MPC는 수정된 DLL 헤더로 해당 엔트리를 갱신한다(그림 5)의 change\_dll\_h() 부분).

- Egress 캐쉬 엔트리의 삭제(그림 6)에서 ③, ④에 해당)

Holding time이 0인 캐쉬 저장 요청에 대해서는 캐쉬 엔트리를 삭제하는데, 이 단계에서 물리적인 삭제

는 일어나지 않고 캐쉬 엔트리 상태를 Invalid로 변화시킨다. 물리적 삭제는 추후에 Egress 캐쉬 엔트리를 에이징할 때 일어나게 된다.



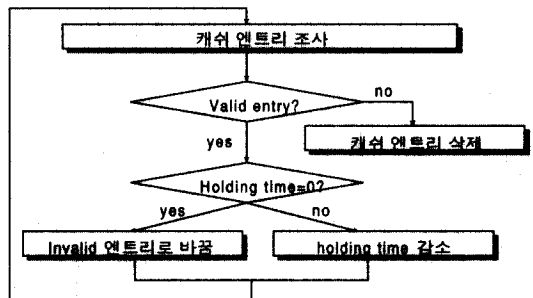
(그림 6) Ingress/Egress 캐쉬의 갱신, 삭제와 에이징

- Ingress 캐쉬 엔트리의 삭제(그림 6)에서 ⑤에 해당)

Ingress MPC가 NHRP 캐쉬 제거 요청을 받으면 Ingress 캐쉬 엔트리의 키값, 즉 Ingress MPS의 ATM 주소와 3계층 목적지 주소를 조사하여 대응하는 엔트리를 Invalid로 무효화한다. 물리적 삭제는 추후에 Ingress 캐쉬 엔트리를 에이징할 때 일어나게 된다.

- Ingress 캐쉬 엔트리의 Aging(그림 6)에서 ⑥에 해당)

Ingress MPC는 캐쉬 엔트리에 타이머를 두어 holding time을 줄여나가며 에이징하고 이 holding time이 0이 되면 엔트리의 상태를 Invalid로 바꾸게 된다. 또한 캐쉬 엔트리의 상태가 NHRP 캐쉬 제거 요청에 의해 Invalid 상태로 변한 엔트리를 물리적으로 삭제한다(그림 7) 참조).



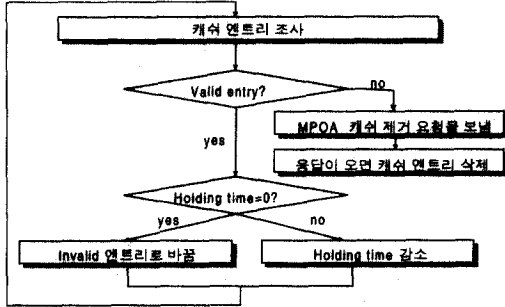
(그림 7) Ingress 캐쉬의 에이징

- Egress 캐쉬 엔트리의 Aging(그림 6)에서 ⑦에 해당)

Egress MPC는 0인 holding time을 받아 Invalid 상태로 된 캐쉬 엔트리를 삭제하기 위해 MPOA 캐쉬 제거 요청을 보내어 응답이 오면 해당 엔트리를 물리적으로 삭제하고, holding time을 시간의 흐름에 따라 줄



여 나가면서 Egress 캐쉬 엔트리를 Aging하게 된다 ((그림 8) 참조).



(그림 8) Egress Cache의 에이징

### 3.3 MPS

MPS의 기능은 크게 구성 단계를 위한 기능, 장치 발견 단계를 위한 기능, 목적지 주소 해석을 위한 기능, 연결 관리 기능, 데이터 전송 기능 등의 다섯 가지로 나누어 볼 수 있다. 본 절에서는 구현된 MPS의 기능들을 설명한다.

#### 3.3.1 구성

MPS의 구성 단계는 MPC의 구성 단계와 마찬가지로 "mps\_par.h"라는 파일에서 매개변수 값을 얻는 방법으로 구현하였다. 구성 매개변수의 첫 번째 변수인 MPS-p1(Keep-Alive Time)과 MPS-p2(Keep-Alive Lifetime)는 Egress MPS와 Egress MPC의 캐쉬 내용 일치도를 보장하기 위하여 MPOA Keep-Alive 메시지를 보낼 때 필요한 매개변수 값이다. 이 메시지는 Egress MPS로부터 Egress MPC들에게 일정 시간(MPS-p1)마다 보내져야 하고, 이 메시지 안에 명시된 메시지 유효 시간인 Keep-Alive Lifetime(MPS-p2)안에 다음 Keep-Alive 메시지가 오는지 Egress MPC들은 감시하여야 한다. 세 번째 구성 매개변수인 MPS-p3(Internet-Layer Protocols)은 MPOA 주소 해석이 지원해 주는 3계층 프로토콜의 집합이다. 다음 세 개의 매개변수 MPS-p4(MPS Give Up Time), MPS-p5(Initial Retry Time), MPS-p6(Retry Time Maximum)과 MPS 상수인 MPS-cl(Retry Time Multiplier)은 MPOA 재시도 매커니즘에서 사용되는 값으로서 MPOA 요청을 보낸 후 그에 대한 응답이 일정 시간 내에 도착하지 않을 때 이 값들을 이용해 MPOA 요청을 다시 보내게 된

다. 이러한 재시도 절차는 MPC의 재시도 매커니즘과 동일하다. 마지막 매개변수인 Default Holding Time (MPS-p7)은 NHRP 주소 해석 응답의 디폴트 유효 시간이고, Egress MPS는 더 적당한 유효 시간을 정하기 위해 지역 정보를 이용해도 된다.

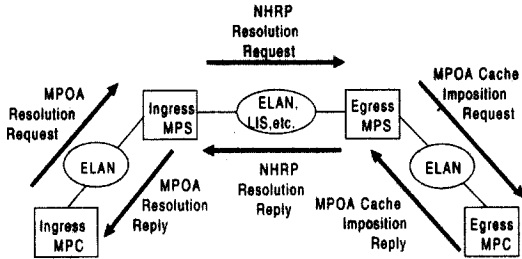
#### 3.3.2 장치 발견(Device Discovery)

Egress MPS는 NHRP 주소 해석 요청을 받았을 때, Egress MPC에게 캐쉬 저장 요청을 보내기 위해 MPC의 ATM 주소를 알아야 하고 MPS가 NHRP 메시지를 포워딩 할 수 있도록 다른 MPS의 주소를 알고 있어야 한다. 이런 장치 발견을 위해 MPS는 LANE의 서비스를 이용한다. MPS가 LEC에게 LE\_RESOLVE\_request를 보내면 LEC는 응답으로 LE\_ASSOCIATE\_indication을 보내며 여기에는 MPC의 제어 ATM 주소, MPS들의 MAC 주소, 제어 ATM 주소 등이 들어 있다.

#### 3.3.3 목적지 주소 해석(Target Resolution)

목적지 주소 해석을 위한 MPS의 역할은 크게 지름길 설정, 무효화된 지름길 제거, Egress MPCs에 부여한 캐쉬 엔트리의 유효성을 보장하기 위한 MPC와의 통신으로 나누어 볼 수 있다. MPOA 목적지 주소 해석은 지름길 전송 시에 필요한 종단간의 ATM 주소를 얻기 위해서 기존의 NHRP를 확장하여 사용한다. 목적지로의 디폴트 전송보다 지름길 전송이 이롭다고 판단을 내린 Ingress MPC로부터 MPOA 주소 해석 요청을 받으면 NHRP 주소 해석 요청으로 변환하여 MPS에 포함된 NHS를 통해 목적지를 향한 경로 상으로 전송한다. NHS들에 의해 포워드된 NHRP 주소 해석 요청이 자신이 관리하는 목적지에 관한 것이라면 이 때 Egress MPS로서 역할을 하는데, 받은 NHRP 주소 해석 요청을 MPOA 캐쉬 저장 요청으로 변환하여 적당한 MPC에게로 전송한다. 이렇게 하여 지름길 설정을 위해 필요한 종단 ATM 주소를 얻으면 근원지로 응답을 보내어 지름길 설정을 완성하게 된다. (그림 9)는 이 주소 해석 과정을 종합적으로 보여 주고 있다. 생성된 지름길은 유효 시간이 지나거나 호스트 이동 또는 라우팅 정보의 변경으로 목적지 정보가 무효화되면, 주소 해석 요청의 근원지에게 제거 요청을 해야 한다. 이를 위하여 MPS는 주소 해석 요청에 대한 응답 정보를 저장해야 한다. 그 때 사용되는 자료 구조는 <표 3>과 같으며 구현된 MPS의 목적지 주소 해석을 위한 함수

의 흐름도는 (그림 10)과 같다.



(그림 9) MPOA 주소 해석 프로세스

<표 3> MPS 캐쉬 엔트리 필드 구성

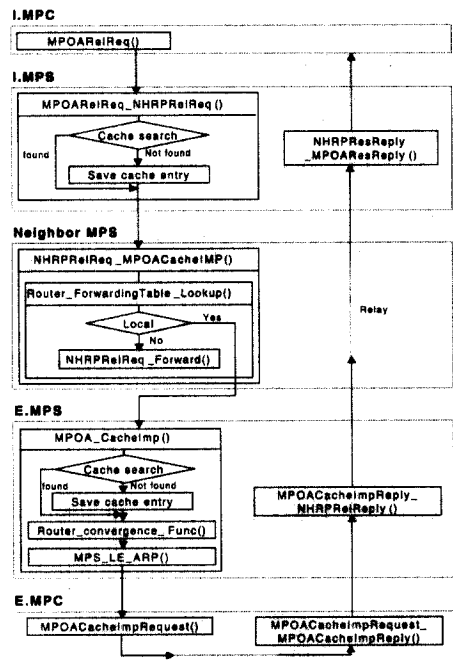
필드명	캐쉬	설명
req_id	Ingress, Egress	받은 패킷에 저장된 Request ID
imps_pro_adr[]	Ingress, Egress	Ingress MPS 프로토콜 주소
src_nbma_adr[]	Ingress, Egress	근원지 ATM 주소
des_pro_adr[]	Ingress, Egress	목적지 프로토콜 주소
req_new_id	Ingress, Egress	보낼 패킷에 부여되는 새로운 Request ID
hold_time	Ingress, Egress	캐쉬 엔트리 유효 시간
tag	Egress	Egress MPC에서 캐쉬 엔트리에 부여한 tag
state	Egress	캐쉬 엔트리의 상태('I': Invalid, 'V': Valid)
*prt	Ingress, Egress	다음 캐쉬 엔트리를 포인팅

- 지름길 설정 : MPC가 MPOA 주소 해석 요청을 보낸 경우

MPC가 지름길 설정이 유리하다고 판단한 경우 MPS에게 MPOA 주소 해석 요청을 보내면((그림 10)의 MPOARelReq() 부분), MPS는 자신의 캐쉬를 검색하여 없는 경우 새로운 캐쉬 엔트리를 만들어 저장하고 NHRP 주소 해석 요청 패킷을 만들어 이웃 MPS들에게 포워드한다((그림 10)의 MPOARelReq\_NHRPRelReq()부분). 이 기능을 위해 구현된 함수에서의 입력 패킷과 처리 후 출력 패킷의 내용은 <표 4>와 같다.

이웃 MPS로부터 NHRP 주소 해석 요청을 받은 MPS는 해당 목적지가 자신이 서브하는 목적지인지 체크하기 위해 라우터 포워딩 테이블 검색을 수행한다. 해당 목적지가 자신이 서브하는 것이 아닌 경우에는 다른 MPS로 포워드하고((그림 10)의 NHRPRelReq\_MPOACacheIMP() 부분), 자신이 서브하는 목적지인 경우에는 MPOA 캐쉬 저장 요청으로 변환한다. 이 때 Egress

MPS는 해당 요청에 대한 캐쉬 엔트리가 없는 경우 새로 만들어 저장한다. 또한 라우터 수렴 함수를 통해 목적지로 보낼 프레임의 DLL 헤더를 알아내고, LEC와 통신하여 MPOA 캐쉬 저장 요청을 보낼 Egress MPC의 주소를 알아낸다((그림 10)의 MPOACacheImp() 부분). 이 기능을 위해 구현된 함수의 입력 패킷과 처리 후 출력 패킷의 내용은 <표 5>와 같다.



(그림 10) Ingress/Egress MPS에서의 지름길 설정 과정

Egress MPS는 MPOA 캐쉬 저장 요청에 대한 응답을 통해 Egress MPC의 ATM 주소를 알아내어 NHRP 주소 해석 응답으로 변환하고 이를 Ingress MPS로 전송한다((그림 10)의 MPOACacheImpReply\_NHRPRelReply() 부분). 이 때 변환 패킷의 내용은 <표 6>과 같다. NHRP 주소 해석 응답을 받은 Ingress MPS는 MPOA 주소 해석 응답으로 변환하여((그림 10)의 NHRPResReply\_MPOAResReply() 부분) Ingress MPC에게 전달해 줌으로써 원하는 목적지로의 지름길이 설정될 수 있다. Ingress MPS에서 NHRP 주소 해석 응답을 MPOA 주소 해석 응답으로 변환하는 함수에서의 입력 패킷은 <표 7>과 같다.

<표 4> MPOA 주소 해석 요청을 NHRP 주소 해석 요청으로 변환

	MPOA 주소 해석 요청 패킷	NHRP 주소 해석 요청 패킷
Packet type		
Request ID		
Src protocol adr		
Des protocol adr		Des protocol adr
Src NBMA adr		I-MPC data ATM adr
Client protocol adr	Null	Null
Prefix length	Widest acceptable prefix length	Widest acceptable prefix length
Holding time		
Client NBMA adr	Null	Null
Extension	① Empty MPOA Egress cache tag ext ② MPOA ATM service category ext(Optional)	Received Ext

<표 5> NHRP 주소 해석 요청을 MPOA 캐시 저장 요청으로 변환

	NHRP 주소 해석 요청 패킷	MPOA 캐시 저장 요청 패킷
Packet type		
Request ID	Request ID2	Request ID3
Src protocol adr		
Des protocol adr	Des protocol adr	Des protocol adr
Src NBMA adr	I-MPC data ATM adr	I-MPC data ATM adr
Client protocol adr	Null	Null
Prefix length		
Holding time		>= 2*Holding time
Client NBMA adr	Null	Null
Extension	① Empty MPOA Egress cache tag ext	Received Ext

<표 6> MPOA 캐시 저장 응답을 NHRP 주소 해석 요청으로 변환

	MPOA 캐시 저장 응답 패킷	NHRP 주소 해석 응답 패킷
Packet type		
Request ID	Request ID3	Request ID2
Src protocol adr		
Des protocol adr	Des protocol adr	Des protocol adr
Src NBMA adr	I-MPC data ATM adr	I-MPC data ATM adr
Client protocol adr		
Prefix length	Actual prefix length	Actual prefix length
Holding time	NULL	Holding time
Client NBMA adr	E-MPC Data ATM addr	E-MPC Data ATM addr
Extension	① Empty MPOA Egress cache tag ext ② MPOA ATM service category ext(Optional) ③ MPOA DLL Header ext	Received Ext(①, ②, ③)

- 지름길 설정 : MPS가 지름길 설정에 유익한 데이터 흐름을 감지하여 MPC에게 MPOA 주소 해석 요청을 보낼 것을 trigger하는 경우  
MPC뿐 아니라 논리적 라우터의 구성 요소인 MPS도 데이터의 흐름을 보면서 지름길 설정이 디폴트 전

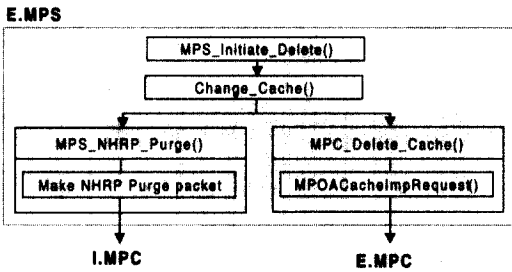
송보다 더 유리한지를 판단할 수 있다. 지름길 설정이 더 유리하다고 판단한 경우, 그 데이터의 근원지에 해당하는 Ingress MPC에게 MPOA 주소 해석 요청할 것을 트리거한다. 그 이후의 주소 해석 과정은 앞의 경우와 같다.

〈표 7〉 NHRP 주소 해석 응답을 MPOA 주소 해석 응답으로 변환

	NHRP 주소해석응답 패킷	MPOA 주소해석응답 패킷
Packet type		
Request ID	Request ID2	Request ID1
Src protocol adr		
Des protocol adr	Des protocol addr	Des protocol addr
Src NBMA adr	I-MPC data ATM addr	I-MPC data ATM addr
Client protocol adr	E-MPS protocol addr	E-MPS protocol addr
Prefix length	Actual prefix length	Actual prefix length
Holding time	Holding time	Holding time
Client NBMA adr	E-MPC Data ATM addr	E-MPC Data ATM addr
Extension	① Empty MPOA Egress cache tag ext ② MPOA ATM service category ext(Optional) ③ MPOA DLL Header ext	Received Ext(①, ②, ③)

● Egress MPS에 의한 설정된 지름길에 대한 정보 갱신 및 지름길 제거

Egress MPS가 자신의 유효한 캐쉬 엔트리들 중 일부에 영향을 주는 목적지 3계층 주소 변경을 감지했을 때, 자신이 부여한 Egress MPC 캐쉬도 그 변경된 정보로 갱신하거나 그 지름길을 제거해야 한다. Egress MPC에게 MPOA 캐쉬 저장 요청을 할 때 부여한 캐쉬 ID 및 변경된 DLL 정보를 넣어 보내면 된다(그림 11)의 Change\_Cache() 부분).



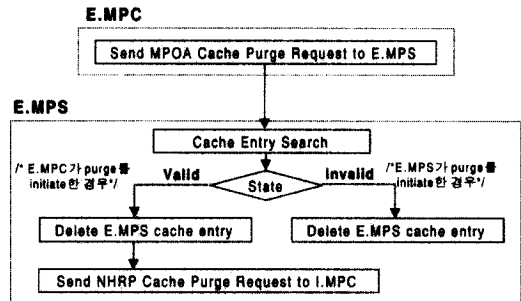
(그림 11) MPS가 초기화하는 Egress 캐쉬의 삭제

위와 같은 이유로 Egress MPS가 목적지 3계층 주소의 변경을 감지했을 때, 캐쉬 엔트리의 일관성을 위해 근원지와 목적지에 관련된 캐쉬 엔트리를 제거할 수 있다. 이 때 자신이 Egress MPC에게 부여한 캐쉬 엔트리를 삭제하기 위하여 0의 유효 시간을 갖는 MPOA 캐쉬 저장 요청을 보내어 캐쉬를 무효화시키고(그림 11)의 MPC\_Delete\_Cache() 부분, 처음 지름길에 대한 주소 해석 요청을 보낸 Ingress MPC에게는 NHRP 제거 요청을 보내어 Ingress MPC의 캐쉬 내용도 제거하

도록 한다(그림 11)의 MPS\_NHRP\_Purge() 부분). Ingress MPC에게 NHRP 제거 요청을 보내기 위해 Ingress MPS의 프로토콜 주소가 필요하며 이 정보는 Egress MPS의 캐쉬 엔트리를 통해 얻을 수 있다. 이 때 Egress MPS의 캐쉬 엔트리를 바로 제거하지 않고 캐쉬 엔트리 상태를 T(Invalid)로 바꾸고, Egress MPC의 캐쉬를 무효화시킨 후 일정 시간이 지나서 Egress MPC로부터 MPOA 캐쉬 제거 요청이 왔을 때 제거된다.

● Egress MPC에 의한 지름길 제거

Egress MPC에 의해 전송된 MPOA 캐쉬 제거 요청을 받은 Egress MPS는 제거하고자 하는 캐쉬의 정보를 지우고 그 정보가 유효한 것이면 Ingress MPC의 캐쉬 정보도 제거해야 하므로 Ingress MPC에게 캐쉬 제거 명령을 보내야 한다(그림 12 참조).



(그림 12) MPC가 초기화하는 Egress 캐쉬의 삭제

3.3.4 데이터 전송(Data Transfer)

디폴트 전송이 되어지는 데이터 흐름을 관찰하여 지

림길 전송이 유리하다고 판단되면 목적지 주소 해석 매커니즘을 사용하여 지름길 설정을 위한 목적지 Egress MPC ATM 주소를 근원지 Ingress MPC에게 알려줌으로써 이후의 데이터 패킷들이 지름길로 전송될 수 있도록 도와준다.

#### 4. 구현 시나리오

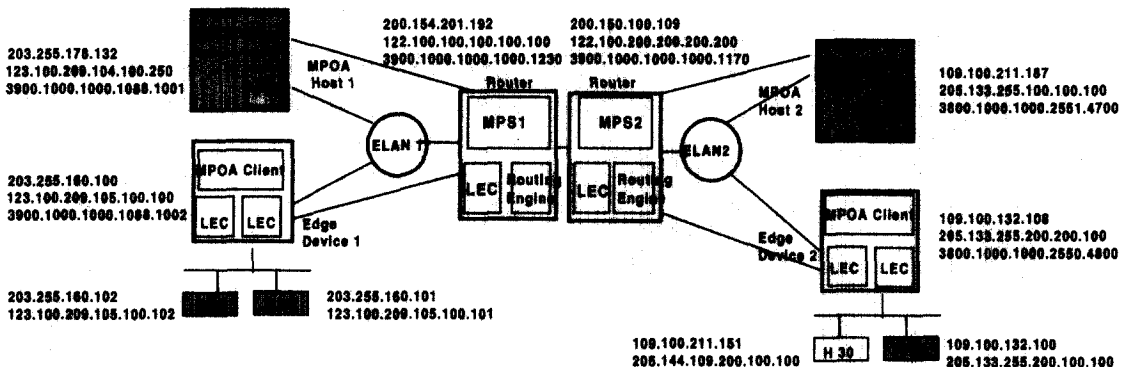
MPOA 기반 스위치드 라우터의 개발의 일환으로 구현한 MPC와 MPS의 기능을 테스트하기 위하여 MPOA 시스템에서 가능한 모든 시나리오를 구성하고 구성된 시나리오를 실행하기 위한 드라이버를 개발하였다. 본문에서 사용한 시나리오는 ATM 포럼 표준안의 부록에서 제공하는 MPOA 구성요소의 시스템 상태 기계와 MPOA의 데이터와 제어 흐름을 통합하여 네가지 시나리오를 구성하고 각 단계마다 이들의 상태를 표준안과 비교하여 올바른지에 대하여 검사하였다. 시나리오 1은 지름길 생성을 위한 두 가지 방법과 재시도 매커니즘을 테스트하며 시나리오 2는 Egress MPS에서 시작하는 캐쉬 엔트리 갱신과 제거, Egress MPC 타이머에 의한 캐쉬 엔트리 에이징의 경우를 테스트한다. 또한, 시나리오 3은 데이터 플레인 제거를, 시나리오 4는 Ingress 캐쉬 엔트리의 갱신, 삭제와 에이징을 각각 테스트한다.

구현한 MPOA 시스템 요소들은 프로세스 형태로 라우터나 엣지 디바이스에 탑재되므로 실제 ATM VCC에 대한 작업을 수행하지 않아도 된다. 따라서 구현 시나리오를 테스트하기 위한 가상 시스템은 (그림 13)과 같이 하나의 워크스테이션에 MPOA 구성 요소들을

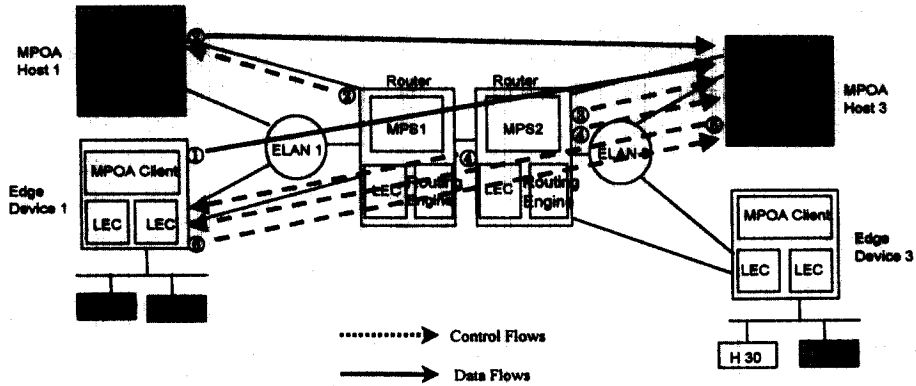
각각 프로세스로서 존재하도록 하였다. 두 개의 ELAN이 존재한다고 가정하고 각각의 ELAN에 하나의 MPS와 MPC 호스트, MPC 엣지 디바이스를 두어 테스트에 필요한 요소들로 구성하였다. (그림 13)에서 MPS1과 MPS2는 각각 ELAN1과 ELAN2를 담당하는 MPOA 서버로서 직접 연결되어 있는 것으로 하였으며 ELAN1에는 MPOA 클라이언트로서 MPOA 호스트 1과 기존의 호스트 H20과 H10으로의 연결을 담당하는 엣지 디바이스 1이 있다. ELAN2에는 MPOA 클라이언트로서 MPOA 호스트 2와 기존의 호스트 H30과 H50으로의 연결을 담당하는 엣지 디바이스 2가 있다. 따라서 본 구현 시나리오에서 사용되는 MPOA 시스템은 2개의 MPS 프로세스와 4개의 MPC 프로세스가 독립적으로 존재하게 되고 이 프로세스사이의 통신은 구현한 드라이버에 의해 전달되도록 하였다. MPC와 MPS가 갖고 있는 캐쉬 내용이 현재의 시스템이 올바르게 작동하는지를 반영하는 것이기 때문에 본 논문에서는 변화된 상태마다의 변화된 캐쉬 내용을 발췌하여 변화된 시스템의 상태를 보여주도록 하였다.

#### 4.1 시나리오 1: 지름길 생성

4.1.1 임계치를 넘어서 지름길 생성((그림 14)의 1) Host10의 상위 계층에서 MPOA Host3으로 보내는 IP 패킷이 들어오면 (그림 15)와 같이 Ingress 캐쉬 엔트리가 생성되어 같은 곳으로 가는 패킷을 계수하게 된다. 'Forwarded Packet'의 값이 임계치를 넘으면 엣지 디바이스 1은 주소 해석 요청 패킷을 MPS1에게 보내게 된다. 이 때 Ingress MPS가 된 MPS1은 다른



(그림 13) 구현 시나리오에서 사용된 MPOA 시스템 구성도



(그림 14) 구현 시나리오에서의 정보 흐름

```

Key ----->
* MPS ATM Address      : 3900 1000 1000 1000 1230
* Destination IP Address : 109 100 211 187

State Inf. ----->
* Last NHRP CIE Code
  Code          : 0
  Prefix Length : 0
  Unused        : 0
  Maximum Transmission Unit : 0
  Client Address Type & Length : 9
  Client Subaddress Type & Length : 9
  Client Protocol Length : 0
  Preference    : 0
  Client NBMA Address : 0 0 0 0 0
  Client NBMA Subaddress : 0 0 0 0 0
  Client Protocol Address : 0 0 0 0

Service Category ----->
* Service Category : 0

Aging Inf. ----->
* Holding Time : 0

Retry Inf. ----->
* Retry ID : 1
* Retry Time : 10
    
```

(그림 15) 엣지 디바이스 1의 생성된 Ingress 캐쉬 엔트리

```

Key ----->
* MPS ATM Address      : 3900 1000 1000 1000 1230
* Destination IP Address : 109 100 211 187

Service Category ----->
* Service Category : 45767

Aging Inf. ----->
* Holding Time : 10

Retry Inf. ----->
* Retry ID : 1
* Retry Time : 10

Usage Inf. ----->
* Forwarded Packet : 3

Purge Inf. ----->
* Tag : 1
    
```

(그림 17) 엣지 디바이스의 수정된 Ingress 캐쉬 엔트리

```

Key ----->
* Destination IP Address : 109 100 211 187
* Source ATM Address : 3900 1000 1000 1088 1002
* Tag : 1

State Inf. ----->
* Egress Cache Entry State : R
* Egress MPS ATM Address : 3900 1000 1000 1000 1171
* Cache ID : 1

Aging Inf. ----->
* Holding Time : 10

Usage Inf. ----->
* Forwarded Packet : 1

Purge Inf. ----->
* Egress MPS Control ATM Address : 3900 1000 1000 1000 1170
    
```

(그림 16) MPOA Host3에서 생성된 Egress 캐쉬 엔트리

MPS에게 NHRP 주소 해석 요청을 보내고, 이를 받은 MPS2는 자신이 서브하는 목적지로의 주소 해석 요청이므로 MPOA 캐쉬 저장 요청으로 변환하여 MPOA Host3에게 전송한다. 이로 인한 MPOA Host3의 Egress 캐쉬 엔트리가 (그림 16)과 같이 생성된다. 목적지 ATM 주소를 담은 MPOA 캐쉬 저장 응답이 MPS2에게 전송되면 주소 해석 요청이 왔던 경로대로 근원지에게 전송이 된다. 주소 해석 요청의 결과로 (그림 15)의 캐쉬 엔트리가 (그림 17)처럼 바뀌게 된다. 근원지로 응답이 오면 Edge Device 1에서부터 MPOA Host3까지

직접 VCC가 설정된다.

4.1.2 트리거되어 지름길 생성((그림 14)의 2)

MPS1이 디폴트 전송되는 데이터 흐름을 보고 MPOA Host1로부터 MPOA Host3으로의 지름길 설정이 유리하다고 판단하고 MPOA Host1을 트리거하면 MPOA Host1은 MPOA 주소 해석 요청을 보내게 되고 응답이 오면 MPOA Host1에서부터 MPOA Host3까지 direct VCC가 설정된다. MPS1이 MPOA Host1을 트리거 한 이후의 실험 과정은 위의(1)의 실험과 같으며 (그림 18)과 같은 Egress 캐쉬 엔트리가 생성된다.

```

<----- Key ----->
* Destination IP Address      : 109 100 211 187
* Source ATM Address         : 3900 1000 1000 1088 1001
* Tag                        : 2
<----- State Inf. ----->
* Egress MPS ATM Address     : 3900 1000 1000 1000 1171
* Cache ID                   : 2
<----- Aging Inf. ----->
* Holding Time               : 10
<----- Usage Inf. ----->
* Forwarded Packet           : 1
<----- Purge Inf. ----->
* Egress MPS Control ATM Address : 3900 1000 1000 1000 1170
    
```

(그림 18) 트리거에 의한 MPOA Host3에서 생성된 Egress 캐쉬 엔트리

4.2 시나리오 2 : Egress 캐쉬 엔트리의 갱신, 삭제와 Aging

이를 테스트하기 위하여 위의 시나리오 1의 결과로 MPOA Host 3에 생성된 Egress 캐쉬 엔트리를 이용한다.

4.2.1 Egress MPS가 초기화하는 캐쉬 엔트리 갱신 ((그림 14)의 3)

MPOA 호스트 이동 또는 라우팅 정보의 변경으로 기존 캐쉬 엔트리의 변경이 요구될 때, Egress MPS는 이전에 그 엔트리를 부여할 때 사용한 목적지 3계층 주소, 근원지 ATM 주소, 태그의 키값과 새로운 DLL 헤더, 0이 아닌 유효 시간을 담은 MPOA 캐쉬 저장 요청 Egress MPC에게 전송하여 캐쉬 엔트리를 갱신한다. (그림 19)는 (그림 18)의 캐쉬 내용이 수정된 것을 보여준다.

```

<----- Key ----->
* Destination IP Address      : 109 100 211 187
* Source ATM Address         : 3900 1000 1000 1088 1001
* Tag                        : 2
<----- State Inf. ----->
* Egress MPS ATM Address     : 3900 1000 1000 1000 1171
* Cache ID                   : 2
<----- Aging Inf. ----->
* Holding Time               : 10
<----- Usage Inf. ----->
* Forwarded Packet           : 1
<----- Purge Inf. ----->
* Egress MPS Control ATM Address : 3900 1000 1000 1000 1170
    
```

(그림 19) MPOA Host3에서의 Egress 캐쉬 수정

4.2.2 Egress MPS가 초기화하는 캐쉬 엔트리 삭제 ((그림 14)의 4)

위의 상황과 마찬가지로 호스트 이동 또는 라우팅 정보의 변경으로 기존 엔트리가 무효화되었을 때, Egress MPS는 그 캐쉬 엔트리를 삭제하기 위하여 Egress MPC에게 0인 유효 시간을 갖는 MPOA 캐쉬 저장 요청을 보내고 (그림 20)은 그 결과이다. 추후에 Egress MPC가 Egress 캐쉬 엔트리를 에이징할 때 무효화된 엔트리를 발견하면 Egress MPS에게 캐쉬 제거 요청을 보낸다. 이 때 Egress MPS는 Egress MPC에게는 캐쉬 제거 응답을, Ingress MPC에게는 NHRP 캐쉬 제거 요청을 보내고 (그림 21)은 그 결과이다.

```

<----- Key ----->
* Destination IP Address      : 109 100 211 187
* Source ATM Address         : 3900 1000 1000 1088 1002
* Tag                        : 1
<----- State Inf. ----->
* Egress MPS ATM Address     : 3900 1000 1000 1000 1171
* Cache ID                   : 1
<----- Aging Inf. ----->
* Holding Time               : 10
<----- Usage Inf. ----->
* Forwarded Packet           : 1
<----- Purge Inf. ----->
* Egress MPS Control ATM Address : 3900 1000 1000 1000 1170
<----- DLL Header ----->
* ELAN ID                    : 3
* DLL Header Length          : 102
* Preamble                   : 112112112112112112
* SFD                        : 115
* Destination Mac Address    : 205 133 255 100 100 100
* Source Mac Address         : 123 100 209 105 100 101
* Length                     : 100
* FCS                        : 1100
* DSAP                       : 100
* SSAP                       : 115
* Control                    : 100
    
```

(그림 20) 무효화된 그림 16의 Egress 캐쉬 엔트리

```

----- Key ----->
* MPS ATM Address      : 3900 1000 1000 1000 1230
* Destination IP Address : 109 100 211 187
----- State Inf. ----->
* Last NHRP CIE Code  :
  Code                : 0
  Prefix Length       : 0
  Unused              : 0 0
  Maximum Transmission Unit : 100
  Client Address Type & Length : 9
  Client Subaddress Type & Length: 9
  Client Protocol Length : 4
  Preference          : 10
  Client NBMA Address : 3800 1000 1000 2551 4700
  Client NBMA Subaddress : 3800 1000 1000 2551 5700
  Client Protocol Address : 200 150 100 109
----- Service Category ----->
* Service Category    : 45767
----- Aging Inf. ----->
* Holding Time        : 5
----- Retry Inf. ----->
* Retry ID            : 1
* Retry Time          : 10
----- Usage Inf. ----->
* Forwarded Packet    : 3
----- Purge Inf. ----->
* Tag                 : 1
    
```

(그림 21) 무효화된 그림 17의 Ingress 캐쉬 엔트리

#### 4.2.3 Egress MPC가 타이머 기반으로 캐쉬 엔트리를 Aging

주기적으로 캐쉬 엔트리를 에이징하는 함수를 호출한다. 이미 캐쉬 엔트리가 Invalid인 것은 캐쉬 엔트리를 삭제하고 0이 아닌 holding time을 가진 Valid 캐쉬 엔트리는 holding time을 줄여 주면서 에이징을 반복한다. (그림 22)는 MPOA Host 3에서 캐쉬 엔트리에 이징에 의하여 Invalid인 캐쉬 엔트리를 Purge로 바꾼 이후에, 대응되는 Ingress 캐쉬 엔트리를 제거하기 위해 MPOA 캐쉬 제거 요청을 보냈을 때 MPOA Host 1의 캐쉬 엔트리가 Invalid 상태로 변하는 실험 결과이다.

```

----- Key ----->
* MPS ATM Address      : 3900 1000 1000 1000 1230
* Destination IP Address : 109 100 211 187
----- State Inf. ----->
* Last NHRP CIE Code  :
  Code                : 0
  Prefix Length       : 0
  Unused              : 0 0
  Maximum Transmission Unit : 100
  Client Address Type & Length : 9
  Client Subaddress Type & Length: 9
  Client Protocol Length : 4
  Preference          : 10
  Client NBMA Address : 3800 1000 1000 2551 4700
  Client NBMA Subaddress : 3800 1000 1000 2551 5700
  Client Protocol Address : 200 150 100 109
----- Service Category ----->
* Service Category    : 45767
----- Aging Inf. ----->
* Holding Time        : 5
----- Retry Inf. ----->
* Retry ID            : 1
* Retry Time          : 10
----- Usage Inf. ----->
* Forwarded Packet    : 1
----- Purge Inf. ----->
* Tag                 : 2
    
```

(그림 22) MPOA Host1의 무효화된 Ingress 캐쉬 엔트리

4.3 시나리오 3 : 데이터 플레인 제거(그림 14)의 ⑤ MPOA Host 3은 이전의 Keep Alive 메시지에 명시된 Keep Alive Lifetime 값인 3초 이내에 다음 Keep Alive 메시지를 받지 못하여 다음과 같은 메시지로 MPS2의 오류를 명시한다.

```

----- Key ----->
* MPS ATM Address      : 3900 1000 1000 1000 1230
* Destination IP Address : 109 100 211 187
----- State Inf. ----->
* Last NHRP CIE Code  :
  Code                : 0
  Prefix Length       : 0
  Unused              : 0 0
  Maximum Transmission Unit : 100
  Client Address Type & Length : 9
  Client Subaddress Type & Length: 9
  Client Protocol Length : 4
  Preference          : 10
  Client NBMA Address : 3800 1000 1000 2551 4700
  Client NBMA Subaddress : 3800 1000 1000 2551 5700
  Client Protocol Address : 200 150 100 109
----- Service Category ----->
* Service Category    : 45767
----- Aging Inf. ----->
* Holding Time        : 5
----- Retry Inf. ----->
* Retry ID            : 1
* Retry Time          : 10
----- Usage Inf. ----->
* Forwarded Packet    : 3
----- Purge Inf. ----->
* Tag                 : 1
    
```

또한 MPS2로부터 받은 모든 Egress 캐쉬의 내용을 무효화하고(그림 20)과 같음), Ingress MPC들에게 데이터 플레인 제거를 보낸다. (그림 23)과 (그림 24)는 각각 데이터 플레인 제거 메시지를 받기 전과 후의 캐쉬 엔트리의 내용 중 해당 부분이다.

```

----- Key ----->
* MPS ATM Address      : 3900 1000 1000 1000 1230
* Destination IP Address : 109 100 211 187
----- State Inf. ----->
* Last NHRP CIE Code  :
  Code                : 0
  Prefix Length       : 0
  Unused              : 0 0
  Maximum Transmission Unit : 100
  Client Address Type & Length : 9
  Client Subaddress Type & Length: 9
  Client Protocol Length : 4
  Preference          : 10
  Client NBMA Address : 3800 1000 1000 2551 4700
  Client NBMA Subaddress : 3800 1000 1000 2551 5700
  Client Protocol Address : 200 150 100 109
----- Service Category ----->
* Service Category    : 45767
----- Aging Inf. ----->
* Holding Time        : 5
----- Retry Inf. ----->
* Retry ID            : 1
* Retry Time          : 10
----- Usage Inf. ----->
* Forwarded Packet    : 3
----- Purge Inf. ----->
* Tag                 : 1
    
```

(그림 23) 데이터 플레인 제거 명령을 받기 전 Edge Device 1의 유효한 캐쉬 엔트리

```

----- Key ----->
* MPS ATM Address      : 3900 1000 1000 1000 1230
* Destination IP Address : 109 100 211 187
----- State Inf. ----->
* Last NHRP CIE Code  :
  Code                : 0
  Prefix Length       : 0
  Unused              : 0 0
  Maximum Transmission Unit : 100
  Client Address Type & Length : 9
  Client Subaddress Type & Length: 9
  Client Protocol Length : 4
  Preference          : 10
  Client NBMA Address : 3800 1000 1000 2551 4700
  Client NBMA Subaddress : 3800 1000 1000 2551 5700
  Client Protocol Address : 200 150 100 109
----- Service Category ----->
* Service Category    : 45767
----- Aging Inf. ----->
* Holding Time        : 5
----- Retry Inf. ----->
* Retry ID            : 1
* Retry Time          : 10
----- Usage Inf. ----->
* Forwarded Packet    : 3
----- Purge Inf. ----->
* Tag                 : 1
    
```

(그림 24) 데이터 플레인 제거 명령을 받은 후 Edge Device 1의 무효화된 캐쉬 엔트리



4.4 시나리오 4 : Ingress 캐쉬 엔트리의 갱신, 삭제와 에이징(그림 14)-⑥

주기적으로 캐쉬 엔트리를 에이징하는 함수를 호출한다. (그림 25)는 에이징 전의 Ingress 캐쉬 엔트리를 보여주며 (그림 26)은 에이징 후에 Holding time이 줄고 이미 Invalid 상태의 캐쉬 엔트리가 삭제된 것을 보여준다. 에이징 과정에서 이미 Holding time이 0인 엔트리를 만나면 이 캐쉬 엔트리의 상태를 Invalid로 하고 다음 번 에이징 과정에서 이를 삭제한다.

```

<----- Key ----->
* MPS ATM Address      : 3900 1000 1000 1000 1230
* Destination IP Address : 109 100 211 187
<----- State Inf. ----->
* Last NHRP CIE Code
  Code                : 0
  Prefix Length       : 0
  Unused              : 0 0
  Maximum Transmission Unit : 100
  Client Address Type & Length : 9
  Client Subaddress Type & Length : 9
  Client Protocol Length : 4
  Preference          : 10
  Client NBMA Address : 3800 1000 1000 2551 4700
  Client NBMA Subaddress : 3800 1000 1000 2551 5700
  Client Protocol Address : 200 150 100 109
<----- Service Category ----->
* Service Category    : 45767
<----- Aging Inf. ----->
* Holding Time       : 10
<----- Retry Inf. ----->
* Retry ID           : 1
* Retry Time         : 10
<----- Usage Inf. ----->
* Forwarded Packet   : 3
<----- Purge Inf. ----->
* Tag                 : 1

<----- Key ----->
* MPS ATM Address      : 4700 1000 1000 1090 1230
* Destination IP Address : 108 100 211 151
<----- State Inf. ----->
* Last NHRP CIE Code
  Code                : 0
  Prefix Length       : 0
  Unused              : 0
  Maximum Transmission Unit : 0
  Client Address Type & Length : 0
  Client Subaddress Type & Length : 0
  Client Protocol Length : 0
  Preference          : 0
  Client NBMA Address : 0 0 0 0 0
  Client NBMA Subaddress : 0 0 0 0 0
  Client Protocol Address : 0 0 0 0
<----- Service Category ----->
* Service Category    : 0
<----- Aging Inf. ----->
* Holding Time       : 0
<----- Retry Inf. ----->
* Retry ID           : 1
* Retry Time         : 10
<----- Usage Inf. ----->
* Forwarded Packet   : 1
<----- Purge Inf. ----->
* Tag                 : 0
    
```

(그림 25) 에이징 이전의 Edge Device 1의 Ingress Cache 엔트리

```

<----- Key ----->
* MPS ATM Address      : 3900 1000 1000 1000 1230
* Destination IP Address : 109 100 211 187
<----- State Inf. ----->
* Shortcut Entry State : V
* Shortcut VCC State   : 0
* Last NHRP CIE Code
  Code                : 0
  Prefix Length       : 0
  Unused              : 0 0
  Maximum Transmission Unit : 100
  Client Address Type & Length : 9
  Client Subaddress Type & Length : 9
  Client Protocol Length : 4
  Preference          : 10
  Client NBMA Address : 3800 1000 1000 2551 4700
  Client NBMA Subaddress : 3800 1000 1090 2551 5700
  Client Protocol Address : 200 150 100 109
<----- Service Category ----->
* Service Category    : 45767
<----- Aging Inf. ----->
* Holding Time       : 10
<----- Retry Inf. ----->
* Retry ID           : 1
* Retry Time         : 10
<----- Usage Inf. ----->
* Forwarded Packet   : 3
<----- Purge Inf. ----->
* Tag                 : 1
    
```

(그림 26) 에이징 후의 Edge device 1의 Ingress 캐쉬 엔트리

5. 결론 및 향후 계획

기존의 방대한 LAN 응용 소프트웨어 뿐 아니라 다양한 3계층 프로토콜을 ATM 망상에서 지원하기 위해 MPOA는 ATM의 성공적인 정착을 위해 필수적이다. 현재 MPOA에 대한 구현은 Fore 시스템을 비롯한 외국의 몇몇 회사에서 LANE와 함께 구현되어 있는 상태이며 현재 국내에서는 한국전자통신연구원을 비롯한 여러 기업에서 장비 국산화를 위해 이를 개발 중이다.

본 논문에서는 MPOA 기반 라우터 장비 개발을 위해 MPOA의 핵심부분인 MPC와 MPS를 구현하였으며 구현한 MPOA 구성요소들간의 작동에 대하여 테스트하였다. 구현한 MPC의 기능은 LEC와 프리미티브 교환 작업, Ingress와 Egress 캐쉬의 관리와 유지, 그리고 LEC를 통한 디폴트 전송과 지름길 전송 등이다. 또한 MPS는 라우터와 수렴 함수, NHS는 존재한다는 가정 하에 순수한 MPS의 기능인 LEC와 프리미티브 교환, MPOA와 NHRP사이의 프레임 변환 및 교환, Ingress와 Egress 캐쉬 관리와 유지 등을 구현하였다.

구현한 MPOA 시스템은 데이터 전송을 위해 지름길과 디폴트 전송, Ingress/Egress 캐쉬 엔트리의 갱신 삭제와 에이징, 그리고 데이터 플레인 제거에 의한 캐쉬 관리를 드라이버를 통하여 테스트하였다. 향후 연구과제로서는 구현한 MPOA 시스템을 장착한 라우터를 설치하여 구축된 네트워크의 성능을 다양한 측면에서 평가하고자 한다.

### 참고 문헌

- [1] The ATM Forum, "Multiprotocol Over ATM Version 1.0(Letter Ballot)," May 1997.
- [2] James V. Luciani, Dave Katz, David Pscitello, Bruce Cole, "NBMA Next Hop Resolution Protocol (NHRP)," INTERNET-DRAFT <draft-ietf-rolc-nhrp-11.txt>, expires Sep. 1997.
- [3] Hao Che, San-qi Li, "MPOA Flow Classification Design and Analysis," *IEEE INFOCOM '99*, New York, USA, March 1999, pp.1497-1504.
- [4] Indra Widjaja, Haining Wang, Steve Wright, Amalendu Chatterjee, "Salability Evaluation of Multi-Protocol Over ATM," *IEEE INFOCOM '99*, New York, USA, March 1999, pp.1505-1512.
- [5] J. Heinanen, "RFC-1483 : Multiprotocol Encapsulation over ATM Adpatation Layer 5, July 1993.
- [6] The ATM Forum, "LAN Emulation Over ATM Version 2.0 - LUNI Specification - Straw Ballot," Apr. 1997.

### 임지영

e-mail : jylim@mm.ewha.ac.kr

1994년 이화여자대학교 전자계산학과 졸업(학사)  
 1996년 이화여자대학교 대학원 전자계산학과 졸업(이학석사)  
 1996년~현재 이화여자대학교 대학원 컴퓨터학과 박사과정

관심분야 : ATM 연동 프로토콜, 위성통신, 인터넷 VPN, QoS 지원 프로토콜

### 김미희

e-mail : kimmh@etri.re.kr

1997년 이화여자 대학교 전자계산학과 졸업(학사)  
 1999년 이화여자 대학교 대학원 컴퓨터학과 졸업(공학석사)  
 1999년~1999년 7월(주)인티 근무 : SNMP기반 NMS 개발

1999년~현재 한국전자통신연구원 교환전송 MPLS 시스템팀 근무 : MPLS 시스템 개발

관심분야 : Traffic Engineering, Diffserv, 라우터 개발 등

### 최정현

e-mail : anna@lgic.co.kr

1997년 이화여자대학교 전자계산학과 졸업(학사)  
 1999년 이화여자대학교 대학원 컴퓨터학과 졸업(공학석사)  
 1999년~현재 LG 정보통신 정보시스템 연구소 통신 S/W 실 연구원

관심분야 : 고속 통신 프로토콜 설계 및 성능 분석, ATM 연동 프로토콜, 멀티캐스트 전송과 QoS 지원 프로토콜, Next Generation Router 개발

### 이미정

e-mail : lmj@mm.ewha.ac.kr

1987년 이화여자대학교 전자계산학과 졸업(학사)  
 1989년 University of North Carolina at Chapel Hill 컴퓨터학과 졸업(이학석사)

1994년 North Carolina State University 컴퓨터학과 졸업(공학박사)

1994년~현재 이화여자대학교 공과대학 컴퓨터학과 조교수

관심분야 : 고속 통신 프로토콜 설계 및 성능 분석, ATM 연동 기술, 비디오 멀티캐스트를 위한 ATM 흐름 제어, QoS 지원을 위한 라우팅과 트래픽 제어

### 채기준

e-mail : kjchae@mm.ewha.ac.kr

1982년 연세대학교 수학과 졸업(학사)  
 1984년 Syracuse University 전자계산학과 졸업(이학석사)  
 1990년 North Carolina State University 컴퓨터공학과 졸업(공학박사)

1990년~1992년 미국 해군사관학교 전자계산학과 조교수  
 1992년~현재 이화여자대학교 공과대학 컴퓨터학과 부교수

관심분야 : 고속통신망, LAN, 무선통신망, 망관리, 성능평가, 정보보호시스템



**최길영**

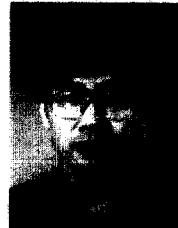
e-mail : kychoi@etri.re.kr

1985년 경북대학교 전자공학과  
졸업(학사)

1987년 경북대학교 대학원 전자공  
학과 졸업(공학석사)

1987년~현재 : 한국전자통신연구  
원 교환전송기술연구소

라우터기술연구부 ATM LAN팀 선임연구원



**강훈**

e-mail : hkang@etri.re.kr

1980년 연세대학교 전자공학과  
졸업(학사)

1989년 Iowa State University  
(공학석사)

1993년 Iowa State University  
(공학박사)

1993년~1999년 한국전자통신연구원 교환전송기술연구  
소 라우터기술연구부 ATM LAN팀장

2000년~현재 한국아이티 벤처 투자(주) 상무이사