

# Native ATM Service를 위한 MOD System의 구현

허 홍<sup>†</sup> · 이 근 왕<sup>††</sup> · 김 봉 기<sup>†††</sup> · 오 해 석<sup>††††</sup>

## 요 약

본 논문에서는 ATM-LAN 환경의 MOD 서버들로부터 클라이언트들에게 프레임 기반의 CM(Continuous Media) 데이터 스트림을 전송하기 위한 기술들을 제안한다. 여기서 프레임이란 CM의 디스플레이 단위를 의미한다. 세부적으로는 수송 계층과 IP 계층의 개입없이 애플리케이션과 AAL이 직접적으로 접속함으로써 ATM-Specific한 native ATM 서비스를 적용한다. 또한 네비게이션 서버를 통한 투명한 브라우징 메카니즘, 서버와 클라이언트들간의 PVC를 통한 세션 설정 과정, 애플리케이션과 AAL과 QoS 협상 및 예약 과정, MTU 사이즈를 초과하는 프레임들에 대한 조각/붙임 알고리즘 등을 제안하고 실험 결과를 제시한다.

## Implementation of an MOD System for Native ATM Service

Hong Heo<sup>†</sup> · Keun-Wang Lee<sup>††</sup> · Bong Ki Kim<sup>†††</sup> · Hae-Seok Oh<sup>††††</sup>

## ABSTRACT

In this paper, We suggest a technique for delivering frame (CM transferring unit) based CM data streams from MOD servers to clients on ATM-LAN environment. The term 'frame' indicates the CM transferring unit. In detail, each clients try to adapt native ATM-specific service which can be connected directly from application layer to AAL without convening transport and IP layer. Also, We suggest and show result of implementation about transparent browsing mechanism using navigation server, session setting procedures using PVC between server and clients, application, AAL and QoS negotiation and reservation procedure, splitting and reassembling algorithm for frames over MTU size.

## 1. 서 론

최근 들어 급속히 발전하고 있는 멀티미디어 처리용 컴퓨터 기술과 고속 멀티미디어 통신 기술은 LAN과 같은 컴퓨터 망을 통하여 지역적으로 분리되어 있는 컴퓨터 시스템 사이에 대용량의 디지털 멀티미디어 데이터(이미지, 동화상, 오디오, 그래픽, 애니메이션

등)를 실시간으로 주고받을 수 있는 환경을 조성하였다.

특히, 한 사용자 당 최고 155 Mbps까지 대역폭을 지원해줄 수 있는 ATM 망 기술의 발전으로 말미암아 ATM을 기반으로 하는 실시간 ATM 멀티미디어 서비스들이 우리의 실세계로 급속히 파고들고 있다. 실시간 ATM 멀티미디어 서비스는 여러 서비스 사용자들에게 멀티미디어 서비스를 지원해줄 수 있는 서버들을 가진 ATM LAN을 위해 고안된 시스템을 말한다.

이러한 실시간 ATM 멀티미디어 서비스 중에 가장 유용한 서비스로 주문형(On-Demand) 멀티미디어 검

† 정 회 원: 한국통신기술연구소 연구원

†† 준 회 원: 숭실대학교 전자계산학과 박사과정

††† 정 회 원: 한림전문대학 컴퓨터응용과 조교수

†††† 정 회 원: 숭실대학교 컴퓨터학부 교수

논문접수: 1996년 2월 28일, 심사완료:

색(retrieval) 시스템을 들 수가 있다.

주문형 멀티미디어 검색 시스템은 일반적으로 클라이언트-서버(client/server)의 구조로 구성되어 있다. 일반적으로 이러한 주문형 멀티미디어 검색 시스템을 MOD (Multimedia-On-Demand) 시스템이라 부른다. MOD 시스템은 크게 서버, 네트워크, 클라이언트 단말의 세가지 요소들로 구성이 되며, 멀티미디어 타이틀이라는 내용(contents) 기반의 미디어 오브젝트를 통하여 상호 통신을 실현한다.

MOD에 관한 국내 동향으로는, ATM-LAN 환경하에서 음성과 동화상 전송을 위주로 하는 ETRI의 DiAMOND(Distributed Apparatus for Multimedia-On-Demand)[13]와 분산 환경하에서 멀티미디어 자원을 네트워크에 투명하게 사용할 수 있도록 하기 위한 방안의 연구로 ETRI의 MuX 모델이 있다. 국외 동향으로는 Berkeley의 CMToolkit[1, 2], UT Austine의 On-Demand Multimedia Service에 관한 연구[6], ATM-LAN을 위한 GRAMS[5] 등이 있다.

본 논문의 연구 목적은 ATM-LAN 환경하에서 실시간으로 멀티미디어 타이틀을 전송할 수 있는 MOD 애플리케이션 개발을 위한 native ATM 서비스용 전송/수신 모듈을 고안하고, 이것을 탑재하여 실제 서비스에 활용할 MOD 서비스 모델을 설계하고 구현하는데 있다. 여기서 'native ATM 서비스'라는 용어는 ATM UNI(User-Network Interface)의 사용자측면에 상주하는 장치들내의 S/W 프로그램과 H/W를 직접적으로 조정하기 위한 ATM-specific 서비스를 말한다[14]. 또한 native ATM API(Application Programming Interface)를 통하여 멀티미디어 전송 효율을 떨어뜨리는 수송계층과 IP 계층의 개입없이 직접적으로 애플리케이션과 AAL(ATM Adaptation Layer)을 연결하여 데이터 전송, SVC/PVC 셋업 및 세션 관리 등의 작업들을 효율적으로 처리할 수 있다. 한편, 멀티미디어 타이틀이란 텍스트, 이미지, 동화상, 오디오 등의 두개 이상의 미디어 데이터들을 그 내용에 따라 저장하고 있는 미디어 오브젝트를 말한다.

## 2. MOD

### 2.1 MOD의 정의

MOD(Multimedia-On-Demand)는 사용자가 브라우

징 시스템을 통하여 검색을 원하는 타이틀을 선택하고, 논리적인 재생 시간에 따라 의미(meaning)의 왜곡 없이 연속적(continuous)이고, 정시적(timely)으로 검색할 수 있는 서비스를 말한다[3]. MOD 서비스에서 사용되는 멀티미디어 타이틀은 크게 CM(Continuous Media)과 DM(Discrete Media)의 두 가지 미디어로 구성이 되며, CM은 오디오, 동화상과 같은 세부 스트림들로 채널을 구성하고, DM은 그래픽 이미지나 텍스트 등의 스트림들로 채널을 구성한다. CM과 DM의 혼합적 검색으로 얻을 수 있는 가장 큰 장점은 한 내용에 따라 여러가지 언어로 자막을 받아볼 수 있다는 것이다.

### 2.2 MOD의 구성 요소

MOD의 구성요소는 클라이언트, 네트워크, 서버의 3가지로 구분되며, 각각의 항목들은 <표 1>과 같이 구성되어진다. 클라이언트 단말에서는 반드시 타이틀 주문에 사용할 브라우저(혹은 네비게이터)가 설치되어 있어야 하고, 타이틀을 프리젠테이션할 애플리케이션과 연결되어 있어야 한다.

<표 1> MOD의 구성 요소  
<Table 1> Components of MOD

클라이언트	
S/W	브라우저 애플리케이션
H/W	Video: MPEG/MJPEG/H. 261 decoder audio: audio PCM decoder network adapter
네트워크 (ATM-LAN)	
MOD 서버	
S/W	브라우징 서버(DBMS와 브라우징 데몬) MOD 서버 프로세스
H/W	CD-ROM, 하드 디스크

### 2.3 MOD 설계 포인트

#### 2.3.1 동기화(Synchronization)

동기화는 크게 inter-media synchronization과 intra-media synchronization으로 나뉘어 있다. 전자는 서로 다른 미디어들간의 동기화를 말하며, 후자는 한 미디어 내에서의 CM특성을 살리기 위한 동기화를 말한다

다. 여기서 CM 특성이란 연속적이고 정시적인 특성을 말한다.

우선, 미디어들간의 동기화는 오디오와 비디오의 일체형(interleaving)과 분리형이 있다. 일체형은 인코딩시 소스 화일 자체에 오디오와 비디오의 조각들을 각 트랙이나 색터별로 간지화시키는 기법을 말하며, MS사의 AVI 파일과 MPEG Systems 등이 대표적인 예이다. 그러나, 일체형에서는 필요에 따라 오디오나 비디오 트랙을 다른 것으로 갈아 끼우고자 할 때 수정이 불가능한 단점을 가지고 있다.

특히, 음성 다중 시청이나, 언어를 자신의 취향에 맞게 선택해서 듣고자 할 때 문제가 발생한다. 반면, 분리형은 오디오와 비디오가 분리되어 독자적인 스트림을 형성하므로, 일체형에서의 단점을 보완할 수 있지만 각 미디어 별로 동기화를 하기 위해서 클라이언트상에서의 오버헤드가 부가되어진다. 이러한 오버헤드가 어느 boundary를 넘어가면, 입술 동기화가 심하게 깨지게 되는 단점을 가지고 있다. 그러나, 개념적으로는 분리형이 주문형 멀티미디어 서비스에 적합하다고 판단되어진다. 그 이유는 자막과 같은 DM이 함께 동기화 될 수 있기 때문이다. 분리형은 (그림 1)과 같이 lts(logical time system)를 기반으로 하는 스토리보드를 통하여 개념화 될 수 있다.

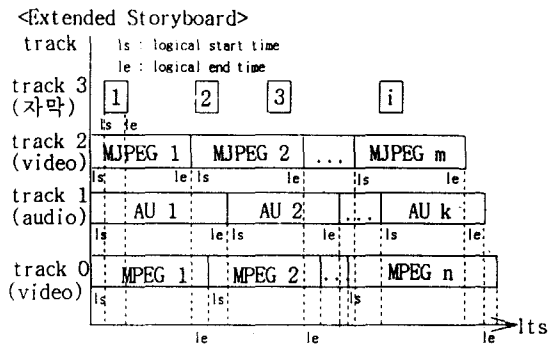
(그림 1)에서 제시한 스토리보드는 CMToolkit[1]의 스토리보드를 개선 확장한 것으로, 크게 CM 트랙과 DM트랙으로 구성이 되며, DM 트랙은 자막을 lts에 맞게 디스플레이할 수 있는 개념적인 모델을 제시한다[9].

현재까지의 연구를 통해 조명해 본다면, 스토리보드를 통하여 미디어간의 동기화 특히 DM과 CM의 동기화를 구현하기 위하여 어떠한 메커니즘을 사용할 것인가가 가장 큰 문제점이라고 할 수 있으며, 아울러 인코딩 포맷이 다른 두 개 이상의 상호 관련 비디오 화면을 어떻게 디스플레이할 것인가에 대한 해결책들이 제시되어야 한다.

두 번째로, intra-media synchronization을 구현하기 위한 방법으로는 우선적으로 이 동기화 메커니즘을 미디어 서버의 전송 모듈쪽에 설치할 것인지, 아니면 클라이언트쪽에 설치할 것인지에 대한 고려사항이 신중히 검토되어야 한다. 각각의 경우에 대하여 tradeoff가 있기 때문에 시스템 개발자는 자신의 환경에 맞는

쪽을 택하여야 한다. 전자의 경우에는, 전송하는 모듈 쪽에서 1 fps 만큼의 delay 간격을 두고 패킷을 전송해 줌으로써 구현이 가능하다. 그러나 이러한 메커니즘이 서버쪽에 설치될 경우, 서버 시스템의 부하를 가중시켜 다른 스트림 또는 다른 클라이언트에게 영향을 끼칠 수가 있으며, 무엇보다도 admission rate를 감소시키게 된다.

그러나, 클라이언트 시스템들이 이 메커니즘을 설치할 수 없는 경우, 서버가 이 메커니즘을 수용해도 무방할 정도의 성능을 가지고 있는 경우, 그리고 서비스 환경 자체가 소규모인 경우에는 이 방법이 신중히 검토될 필요가 있다.



(그림 1) 스토리보드 모델  
(Fig. 1) Storyboard model

후자의 경우에는, 디코딩 모듈에 동기화 메커니즘을 설치하는 것으로, 한 프레임이 자신의 디코딩 및 디스플레이 시간 동안(logical end time-logical start time) 다른 프레임의 진입을 차단하는 메커니즘이다. 그러나, 이러한 메커니즘을 구현하기 위해서는 비동기적(asynchronous)인 화일 디스크립터(file descriptor)를 생성하고 관리해야 하는데, 이 방법은 주기적인 폴링(polling)을 통하여 잦은 타임 인터럽트를 발생시킴으로 UNIX 시스템과 같은 time-sharing 운영체제 상에서는 구현이 까다로운 단점을 가지고 있다.

### 2.3.2 세션 설정시 SVC와 PVC의 선택

ATM-API는 AAL과 애플리케이션 사이에 SAP (Service Access Point)을 연결하여 직접적으로 데이터를 주고 받을 수 있는 화일 디스크립터를 바인딩하여

애플리케이션에 반환한다. 애플리케이션은 반환된 화일 디스크립터로 incoming VPVC(Virtual Path/Virtual Channel)를 통하여 네트워크를 경유해 온 데이터를 ATM 적응 계층으로부터 받을 수 있으며, outgoing VPVC를 통하여 네트워크로 전달할 데이터를 ATM 적응 계층으로 직접 내려 보낼 수 있다.

ATM API의 장점은 TCP/UDP/IP와 같이 기존의 OSI 계층 관점에서 볼 때 수송 계층(Transport Layer)과 IP 계층(IP Layer)을 거치지 않고 직접 ATM 적응 계층으로 데이터를 내려보냄으로써 기존의 미들웨어에서 발생하던 작업을 줄여 상대적으로 높은 전송률(throughput)을 이룰 수 있다는 점이다. 또한, 매년 실행을 시작할 때 마다 자동적으로 ATM 적응 계층과 QoS 협상 및 예약을 할 수가 있으며, PVC를 사용하는 경우 ATM 스위치에 등록된 자신의 VPVC를 자동

적으로 사용할 수가 있어 사용자로부터 투명한 세션 설정 및 유지 과정을 보장할 수 있게 하여 준다.

현재까지 발표된 ATM API로는 Fore Systems의 Fore ATM API가 있으며, ATM FORUM에서 native ATM API[14]의 표준화 작업이 진행 중이다.

Fore의 경우와 같이 ATM API는 크게 SVC용과 PVC용의 함수들로 분리되어진다. 애플리케이션의 서비스적 특성에 따라 이 두가지 중 어느 것을 선택해야 할 것인지를 결정하는 것이 바람직하다. <표 2>는 SVC를 채택한 NV-ATM[12]과 PVC를 채택한 DIAMOND[13]를 비교분석한다.

2.3.3 데이터베이스

다수의 클라이언트들에게 방대한 양의 멀티미디어 타이틀 정보 검색 서비스를 제공하기 위해서는 사용

<표 2> NV-ATM(SVC 채택)과 DIAMOND(PVC 채택)의 비교 분석  
 <Table 2> Comparison analysis of NV-ATM and DIAMOND

	NV-ATM	DIAMOND
application 용도	다자간 화상회의	VOD, MOD
특징	1)같은 내용(contents)의 video frame이 동시에 여러 참가자들에게 전송된다. 이러한 경우 같은 내용의 비디오 packet들을 각 참가자들의 수 만큼 복사해 전송한다면 커다란 대역폭 손실을 초래한다. 따라서, 스위치로 하나의 packet copy만 전송하고, 그것을 전송받은 스위치가 참가자들 수 만큼 복사하여 각 참가자들에게 전송하는 방법을 사용한다. 2)방송 형태(multicast)	1)주문형 전송 서비스가 목적이므로, 같은 타이틀이라 하더라도 각 클라이언트 당 유일한 frame copy가 요구되며, 개인적인 가상 VCR(FF,RWD,PAUSE) 기능을 위한 독립적인 채널이 요구된다. 주문형 전송은 보통 frame 단위의 주문을 위해 각 frame당 lts(logical time stamp)를 찍어 클라이언트에게 전달한다. 따라서, 어느 시청 시간상에서도 NV-ATM과 같이 같은 내용(contents)의 video packet이 동시에 존재할 수가 없기 때문에 multicast식 전송은 의미가 없어진다. 2)point-to-point 형태
구현방법	SVC를 이용한 multicast tree(Fore's SPANS)	Navigation server를 개입시킨 PVC
QOS Context	qos.peak_bandwidth qos.mean_bandwidth qos.mean_burst	qos.peak_bandwidth qos.mean_bandwidth qos.mean_burst
장점	1)multicast이므로 스위치까지는 메시지의 한 copy만 전달되어 대역폭을 줄인다. 2)새로운 가입자의 자유로운 세션 진입이 용이하다. (VP/VC값을 알 필요가 없다.)	1)클라이언트 당 개인적인 PVC를 갖고 세션을 운영하기 때문에 주문형 서비스(가상 VCR, lts 등)가 가능하다. 2)각 채널 당 security가 보장된다.
단점	1)multicast이므로 실제 회의에 참가하지 않는 호스트들도 메시지를 받아 볼 수 있다. (security 문제) 2)주문형 서비스에 부적합하다. 3)SVC의 시그널링이 세션설정시 응답시간을 연장시킨다.	1)각 채널 당 독립적인 frame들이 전송되므로 대역폭이 많이 소모된다. 2)SVC보다 세션 설정 과정이 좀 더 복잡하다. (Navigation Server로 해결가능)

자의 요구를 신속 정확하게 분석하고 해당 정보를 전달해 줄 수 있는 데이터베이스 시스템이 설치되어야 한다. 데이터베이스를 구축할 때에는 MOD 서비스를 위한 특별한 구조의 데이터베이스를 채택할 것인지, 기존의 상용 데이터베이스를 접목시켜 사용할 것인지를 상황에 맞게 결정해야 한다.

2.3.4 GUI

GUI는 클라이언트 시스템과 사용자가 상호대화적으로 통신하는 대화창구 역할을 담당한다. 현재까지 GUI를 쉽고 대화적으로 구현하기 위해 가장 많이 쓰이고 있는 언어는 Tcl-Tk 언어가 있다. CMToolkit이나 DIAMOND, NV-ATM과 같은 많은 대화형 멀티미디어 시스템들이 Tcl-Tk 언어로 GUI를 구현한 대표적인 예이다.

MOD를 위한 GUI 모듈 설계시 고려해야 할 사항들은 크게 다음과 같이 나눌 수가 있다.

- 가상 VCR 기능(FF, RWD, STOP, PAUSE, Slow Video)
- 세션 관리자

- QoS(Quality of Service) 모니터
- 비디오 재생 화면과 기타 조정자들과의 분리/통합 기능
- 오디오 조정자
- 비디오 조정자

2.3.5 파라미터

MOD 서비스를 위해 고려해야 할 파라미터들의 집합은 구축 환경에 따라서 변수적으로 작용하지만, 미디어 서버, 네트워크, 그리고 클라이언트에서 반드시 고려해야 될 파라미터들은 <표 3>과 같이 보여질 수 있다.

3. MOD의 설계

3.1 ATM-API를 이용한 송수신 모듈

3.1.1 전송 모듈

ATM-API를 이용한 전송 모듈은 Fore의 ATM-API 라이브러리를 이용하여 작성되었으며, 실제 MOD 서비스에 적용하기 위하여 CMToolkit 3.0[2]에 탑재하

<표 3> MOD 파라미터  
<Table 3> MOD parameter

fps : frames/sec

	미디어 서버	네트워크	클라이언트
최대 가용대역폭	MaxBandwidth(Mbps)	MaxBand(155bps)	MaxBandWidth(Mbps)
최대 허용 비디오 프레임수	MaxFPS(30fps 기본)		
비디오 해상도	WidthByHeight(pixel)		WidthByHeight(pixel)
비디오인코딩 프레임율	EncodeFPS(fps)		
전송/수신률	SendThroughput		RecvThroughput(Mbps)
마리 읽어들 시간 단위	SendAhead(sec)		SendAhead(sec)
디코딩 프레임율			DecodeFPS(fps)
디코딩 성능			DecodeFPS/(RecvFPS)
마들웨어	(UDP, TCP, CyclicUDP, RTP) +socket API, ATM-API		(UDP, TCP, Cyclic-UDP, RTP) + socket API, ATM-API
QoS Context(KB)	qos.peak_bandwidth qos.mean_bandwidth qos.mean_burst		qos.peak_bandwidth qos.mean_bandwidth qos.mean_burst
MTU size	MTU(KB)		MTU(KB)
주기적 검색단위	CycleTime(sec)		CycleTime(sec)

었다.

본 모듈의 핵심 기능은 ATM API에서 지원할 수 있는 최대 패킷 사이즈인 64 KBytes 이상의 프레임에 대해서도 안전한 전송을 보장해 줄 수 있는 스플릿팅 기능이다. 예를 들어 640×480×24 bits의 MJPEG 프레임들은 VBR 특성 때문에 64 KB를 넘는 경우가 간혹 발생한다. 그러나, ATM API로 한 번에 전송할 수 있는 최대 패킷 크기는 64 KB의 MTU 사이즈로 제한되어지기 때문에, 전송 모듈에서 한 프레임을 여러 개의 프레임으로 쪼개어 전송하는 메커니즘이 필수적으로 마련되어야 한다.

본 전송 모듈의 장점은 MOD용 전송 모듈의 가장 큰 요구 사항인 견고성과 신뢰성을 보장할 수 있는 스플릿팅 메커니즘을 통하여, 장시간 동안 비디오 타이틀을 시청을 해도 도중에 끊겨진 프레임을 보내거나, MTU 사이즈를 초과하여 전송을 시도할 때 발생하는 비정상적인 서비스 중단 사태를 방지할 수가 있다.

구체적인 알고리즘은 C 언어의 형식을 빌어 (그림 2)에 나타내었다.

```

For (i = 0; i < # of frames[user]; i++) {
loop = 1;
sent = 0;
while (loop) {
if ((sent = atm_sendto(fd, &buf[i][sent], (total > MTU ? MTU : total), vpvcc)) < 0) {
error(atm_sendto error);
}
total += sent;
if (total == 0) loop = 0; /* already sent all */
}
}

```

(그림 2) ATM-API 전송 모듈의 전송 알고리즘  
(Fig. 2) Transmission algorithm of ATM-API

3.1.2 ATM-API를 이용한 수신 모듈

ATM-API를 이용한 수신 모듈은 클라이언트의 수신 오브젝트의 가장 핵심적인 부분을 담당하며, 송신 모듈의 스플릿팅 기능에 대응할 수 있는 재결합 기능을 통하여, 제한된 MTU 사이즈 때문에 여러 개의 패킷들로 쪼개어져 오는 프레임을 다시 원래의 프레임

으로 조합해서 디스플레이 모듈로 보내는 역할을 담당한다.

수신 모듈의 재결합 기능에서 가장 중요한 부분은 어떤 방법으로 MTU 사이즈를 초과해서 오는 프레임을 인식하고 재결합할 것인가에 대한 문제이다. 본문에서는 (그림 3)에 보여진 알고리즘을 통하여 이 문제를 해결하였다.

```

Loop = 1;
rec = 0; I = 0;
while(loop){
I++;
if((rec+=atm_rcvfrom(fd, &buf[rec],MTU,vpvc)) < 0) {
error(atm_receive error);
}
if(recv == MTU) continue;
loop=0;
}

```

(그림 3) ATM-API를 이용한 수신 모듈에서의 재결합 알고리즘  
(Fig. 3) Reassembling algorithm of ATM-API

3.1.3 애플리케이션과 ATM망과의 QoS협상과정

애플리케이션과 ATM 망과의 QoS 협상 과정은 크게 애플리케이션이 사용할 ATM 적용 계층의 타입을 선택하는 과정과, 사용하고자 하는 대역폭의 양을 요청하고 예약하는 단계로 나뉜다. AAL 적용 계층의 선택은 3/4, 5 중의 하나를 선택할 수 있으며, 원하는 경우에는 NULL을 선택함으로써, 데이터 전송시 ATM 적용 계층을 거치지 않고 직접 ATM Layer로 전송할 수도 있다. 그러나 이러한 경우에는 애플리케이션 수준에서 ATM 적용 계층의 기능에 해당하는 루틴을 따로 작성해야 하는데, 애플리케이션의 특성에 따라 알맞은 방법을 선택하는 것이 바람직하다. AAL 타입 5는 타입 3/4에 있었던 MID(Multiplex Identifier)를 생략함으로써, 상이한 애플리케이션들로부터 내려오는 스트림들을 하나의 스트림으로 다중화하는 오버헤드를 간소화 시켰다.

사용하고자 하는 대역폭을 요청하고 예약하는 과정에서는 peak bandwidth와 mean bandwidth와 같은 QoS parameter들의 값을 결정하고 요청한다. peak band-

width는 한 애플리케이션의 데이터 스트림이 그 한계를 넘지 않도록 제한함으로써, 네트워크의 오버로드 현상을 방지하기 위함이며, mean bandwidth의 설정은 평균적으로 설정된 값만큼의 대역폭을 보장해 달라고 하는 네트워크 서비스 질에 대한 요구 사항이다. 마지막으로 mean burst size를 조정하는데, ATM 단말에서부터 네트워크로 데이터를 내려 줄 때 마다, 그 만큼의 버스트 크기를 가지고 보내라는 명령이다. (그림 4)는 C로 구현된 QoS Context를 보여준다.

```

Aal_type aal = aal_type_5;
Atm_conn_resource qos;

qos.peak_bandwidth = 60000;(Kbps)
qos.mean_bandwidth = 55000;(Kbps)
qos.mean_burst = 7000;(Kbps)
atm_connect_pvc(fd,vpvc,aal,&qos);
    
```

(그림 4) QoS 협상 로직  
(Fig. 4) Negotiation logic of QoS

### 3.2 브라우징 시스템

본 논문에서 설계한 브라우징 시스템은 사용자로부터 하여금 투명한 자료(서버 디렉토리 및 타이틀 항목에 관한 레코드)의 검색을 보장해주는 시스템으로 브라우징 서버와 MOD 브라우저로 구성된다.

#### 3.2.1 브라우징 서버

BS(Browsing Server)는 크게 DBD(DBMS Demon)과 데이터베이스 파일들로 구성 되어지며, 별도의 호스트상에서 운용될 수도 있고, 미디어 서버와 공존할 수도 있다.

##### 1)DBD

DBD는 BS의 핵심 모듈로 (그림 5)과 같이 DBMS와 데몬으로 구성되어 있다.

본 논문에서 설계한 DBD의 개념적 정의는 다음과 같다.

##### (정의 1) DBD

MOD브라우저의 GUI로 부터 네트워크를 경유해 들어오는 다양한 질의 사항들을 효율적으로 처리하기 위해 기존의 SQL 인터페이스 기반의 DBMS 개념

에 사건 지향적(event-driven) 기반의 데몬(demon) 개념을 혼합해 만든 GQE(GUI Query Event) 인터페이스용 DBMS이다.

##### (정의 2) GQE

GQE(GUI Query Event)는 사용자가 GUI의 특정 질의 위젯(Widget)을 선택하거나, 특정 질의 입력 위젯에 질의 사항을 입력했을 시, 그 내용을 DBMS 데몬에게 전달하기 위하여 구성되는 질의 데이터 전송 포맷이다.

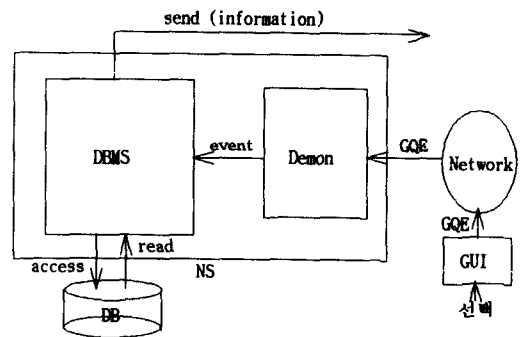
##### - 문법

<메시지>.<질의 내용>

##### ① 데몬

데몬의 역할은 다음과 같다.

- 지정된 포트상에서 특정 브라우저로부터의 연결 설정 요구를 기다린다.
- 만약 연결이 설정되면, GQE를 전송받을 새로운 수신 포트를 할당하고 자신은 계속 다른 브라우저로부터의 연결 설정 요구를 기다린다.
- 할당된 수신 포트상에서 어떤 GQE가 네트워크를 경유해 전달되면, 그 메시지의 의미를 해석하고 특정 행위를 발생시키는 이벤트로 바뀌어서 DBMS에게 전달한다.



(그림 5) DBD의 구조도  
(Fig. 5) Architecture of DBD

##### ② DBMS

DBMS의 역할은 다음과 같다.

- 데몬으로부터 이벤트를 받고, 그 이벤트에 해당하는 특정 행위를 실행한다.
- 행위의 종류
  - 데이터베이스에서 특정 레코드를 읽어 전송하라.

- 한 페이지 만큼의 레코드들을 읽어 전송하라.
- 각 레코드 당 특정 애트리뷰트 값만을 비교하여 일치하는 레코드를 전송하라.
- 새로운 레코드를 데이터베이스에 추가하라.
- 해당 레코드를 찾아 특정 애트리뷰트를 개선하라.
- 특정 레코드를 데이터베이스에서 삭제하라.

- 데이터베이스 형식

BS는 클라이언트로부터 서비스 요청을 감지하는 데몬과 그 요청에 따라 서비스를 요구한 사용자 시스템을 판별하고 그 사용자 시스템을 위해 등록되어 있는 데이터베이스의 데이터를 반환해 준다. 본 논문에서 설계한 데이터베이스의 각 레코드는 (그림 6)와 같은 양식으로 구성된다.

```

use.ent no.1 name.speed vpvc.100 pop.1
host.hanbit_atm port.3333
file.home/cmta2/cmt/cmx/solaris/speed.script
use.ent no.2 name.jangun3 vpvc.120 pop.2
host.vibaldy_atm port.3333
file.home/cmta2/cmt/cmx/solaris/jangun3.script
    
```

(그림 6) 데이터베이스 형식  
(Fig. 6) Database format

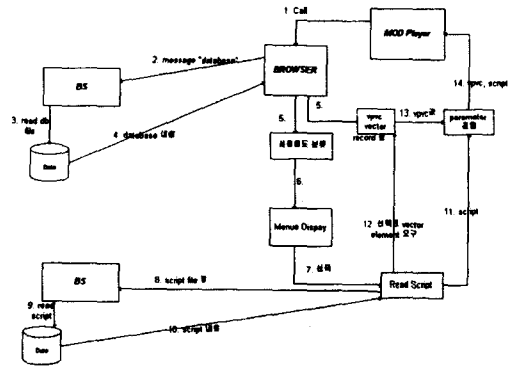
3.2.2 MOD Browser(Navigator)

MOD Browser는 BS로부터 데이터베이스 내용을 반환 받아 자신의 메뉴판에 각 타이틀 이름을 디스플레이 해주게 된다. 또한 그 타이틀을 수록하고 있는 서버와 연결할 수 있는 VPVC값을 연결해 VPVC 벡터를 구성해 놓고, 사용자의 메뉴 선택을 기다리게 된다.

사용자가 한 메뉴를 선택하게 되면, 그 메뉴에 해당하는 타이틀의 구체적인 정보가 담겨진 스크립트 화일이 BS로부터 넘어오게 되며, 이미 구성된 VPVC 벡터의 해당 엘리먼트 액세스를 통해 선택한 타이틀을 수록하고 있는 서버와 세션을 설정할 수 있는 VPVC 값을 애플리케이션에 돌려주게 된다(그림 7).

GUI의 설계는 (그림 8)과 같이 버튼 형식의 메뉴판이 주를 이룬다. 메뉴판에 나타나는 메뉴들은 크게 'INFORMATION'용과 'ENTERTAINMENT'용으로 나뉘어 진다. 그림에 보이는 메뉴들은 'ENTERTAINMENT'용의 비디오 타이틀들을 메뉴화하여 보

여 주고 있다. '<<' 버튼은 이전 페이지로, '>>'는 다음 페이지로 메뉴판이 이동하게 한다.



(그림 7) BS와 MOD Browser간의 상호작용  
(Fig 7) Interaction between BS and MOD browser

3.3 MOD Player

MOD Player는 MOD 서버로부터 전송 받은 멀티 미디어 스트림을 디스플레이 해주는 애플리케이션이다. 본 논문에서는 객체지향 방식으로 이 애플리케이션을 설계하고 구현하였다. MOD Player에 소속되는 오브젝트들은 <표 3>과 같다.

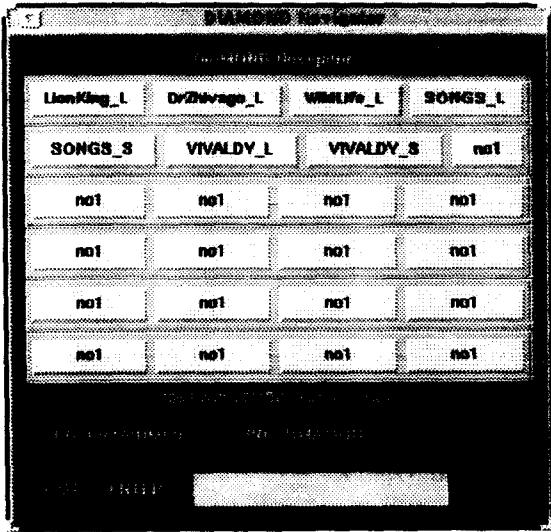
<표 3> MOD Player의 오브젝트들  
<Table 3> Objects of MOD player

사이트	오브젝트(오브젝트명)
MOD 서버	인코딩 (mjpegClipFile, auClipFile) 송신 (atmSrc)
클라이언트	수신 (atmDest) 디코딩 (mjpegDecode, auDecode) 디스플레이(mjpegDisplay, auDisplay)

MOD Player는 MOD Browser로 부터 스크립트 화일을 전송받아 그것을 분석하고 해당 미디어 서버와의 접속을 시도한다. 스크립트 파일의 양식은 (그림 9)과 같다.

스크립트 파일은 크게 '타이틀', '트랙', 그리고 '클립'들로 구성이 되어진다. 타이틀은 해당 비디오 타이틀의 제목을 나타낸다. 트랙은 크게 비디오 트랙과 오디오 트랙으로 나뉘어 진다. 각 트랙은 다시 하나 이상의 클립들로 구성이 되어지며, 각 클립은 다음과





(그림 8) MOD browser의 GUI  
(Fig. 8) GUI of MOD browser

```
Title {speed}
{video{cmtp://hanbit_atm:3333/speed.mjpgc video/mjpeg -
ss 0 -se END -ls 0 -le 100}
{cmtp://vivaldy_atm:3333/lion.mjpgc video/mjpeg -ss 0 -se
END -ls 101 -le 300}}
{audio{cmtp://hanbit_atm:3333/speed.au audio/au -ss 0 -se
END -ls 0 -le 100}
{cmtp://vivaldy_atm:3333/lion.au audio/au -ss 0 -se END -ls
101 -le 300}}
```

(그림 9) 스크립트 파일  
(Fig. 9) Scripts file

같은 문법을 따른다.

```
{cmtp://host_name:port_number/directory type -ss
start_time -se end_time -ls start_time -le end_time}
```

여기서, 'cmtp'는 오디오나 비디오 같은 CM을 클라이언트와 서버간에 주고받기 위한 일종의 통신 규약이다. 'type'은 'video/mjpeg', 'audio/au'중의 한 가지를 나타낸다. ss와 se는 시스템 타임을 가리키며, ls와 le는 프리젠테이션을 시작할 논리적 시간을 가리킨다.

스크립트 파일을 분석한 MOD Payer는 type에서 명시된 각 미디어 디바이스들을 초기화하고 실행 준비를 시킨다. 다음으로, 각 트랙별로 해당 호스트와의 접속을 실시한다. 접속이 성공하면, 그 호스트상에서 인코딩 오브젝트와 송신 오브젝트를 각각 생성시키고, 상호 바인딩을 시킨다. 다음으로 각 오브젝트들의 슬롯들을 초기화하고 적절한 값으로 설정하는 configuring을 실시한다. 리모트 호스트상에서의 작업이 모두 끝나면, 로컬 호스트상에서 수신 오브젝트와 디코딩 오브젝트를 생성하고, 이들을 바인딩 시킨다. 다음으로 그 오브젝트들을 configuring하며, 최종적으로 GUI로부터 사용자의 입력이 들어올 때까지 청취상태에 들어간다.

MOD Player의 GUI는 3D 효과를 지닌 버튼들과 타임 위젯(widget)들로 구성이 되며, 크게 디스플레이 판과 조정자(Control Panel)로 구분이 되어진다. (그림 10)는 기본적인 GUI를 보여 주는데, GUI는 다시 세부적으로 세 가지 단계로 나뉘어져 그 각각의 독특한 모양과 기능을 수행한다. 첫 번째 단계는 기본 단계로, (그림 10)과 같이 구성이 되어진다. 'Hit'라는 버튼을 누르면 사용자 견지의 QoS 파라미터들을 셋팅할 수 있는 위젯들이 나타난다. '↓' 단추를 누르면 가상 VCR 기능(FF., RWD., STOP, PAUSE, Slow Video, etc.)이 가능한 제 2 단계의 GUI로 돌입한다. 또한, 비디오가 디스플레이되고 있는 디스플레이판을 마우스로 더블 클릭하면 조정자가 없어지는 제 3 단계로 돌입하여 순수한 비디오 감상만을 즐길 수가 있다.

#### 4. 구현

프로그램은 객체지향 프로그래밍 기법을 적용하여

〈표 4〉 제어 정보 채널과 CM 스트림 채널  
(Table 4) Channel of control information and CM streams

채널	미들웨어	실험 환경
제어정보 채널	TCP/IP & UDP/IP over ATM	ASX-200 Switch : Fore's SPANS(SVC) ETRI's CANS : PVC
CM스트림	Fore's ATM API	ASX-200 Switch : PVC ETRI's CANS : PVC

C 언어로 구현하였다. 구현을 위하여 Fore의 ATM API 라이브러리, Tcl-Tk 라이브러리, Tcl-DP 라이브러리등을 특별히 사용하였다. 데이터 채널을 사용 목적에 따라 두 가지 종류로 구현하였다.

#### 4.1 오브젝트 atmSrc/atmDest의 정의

##### 4.1.1 atmSrc 오브젝트

다음은 각 atmSrc 오브젝트는 MOD 서버에서 미디어 스트림의 전송을 담당하는 오브젝트로 데이터 구조체의 타입은 다음과 같이 정의된다.

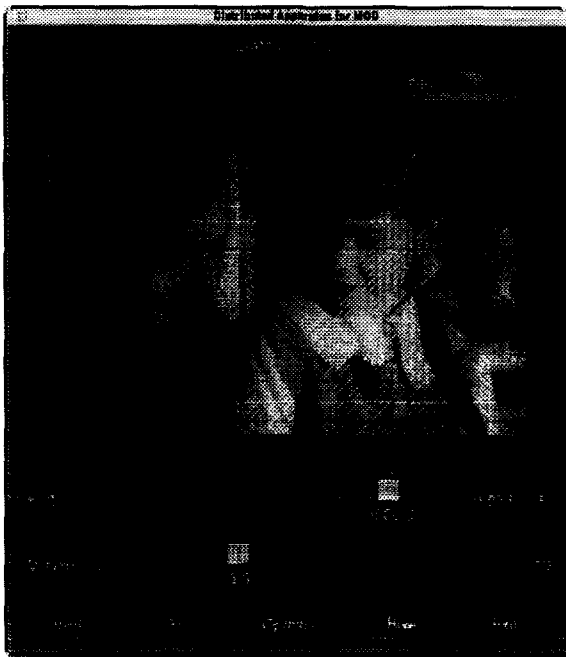
```
typedef struct Atmsrc {
    /* Section 1. General information for new session */
    Tcl_interp *interp; /* Tcl interpreter that created object. */
    char name; /* Tcl string name of this object */
    int atm; /* File descriptor for ATM device driver */
    Vpvc vpvc; /* Virtual path / Virtual channel for new session */
    int dest; /* Destination virtual channel */
    /* Section 2. Information about current buffers to send */
```

```
    ScatterBuffer ** buffer; /* Array of buffers */
    int numBuf; /* Number of active item in the array */
    int maxBuf; /* Number of allocated items in the array */
    /* Section 3. QoS context */
    u_int peak_bandwidth; /* Expected peak bandwidth */
    u_int mean_bandwidth; /* Expected mean bandwidth */
    int mean_burst; /* Expected mean burst size */
    /* Section 4. Intra-Synchronization */
    double lastSendTime; /* Time last SendBurst occurred */
    Tk_TimerToken sendTimer; /* Token for SendBurst callback */
} AtmSource;
```

##### 4.1.2 atmDest 오브젝트

atmDest 오브젝트는 클라이언트에서 전송되어 오 는 미디어 스트림을 수신하는 오브젝트로 데이터 구조체의 타입은 다음과 같이 정의된다.

```
typedef struct AtmDest {
    /* Section 1. Generic information for new session */
    Tcl_Interp *interp; /* Tcl interpreter that created object */
    char *name; /* Tcl string name of this object */
    Vpvc vpvc; /* VP/VC for new session */
    int fd; /* File descriptor for ATM device driver */
    /* Section 2. Data handling for decoding object */
    InFunc *outCCmd; /* C pointer to out command */
    void *outObj; /* The encoding object to send to */
    int shared; /* Should buffers be in shared memory */
    int xAttach; /* Should buffers be attached to the X server */
    /* Section 3. QoS Context */
    u_int peak_bandwidth; /* required peak bandwidth */
    u_int mean_bandwidth; /* required mean bandwidth */
    int mean_burst; /* required mean burst size */
} atmDest;
```



(그림 10) MOD player  
(Fig. 10) MOD player

#### 4.2 오브젝트의 실행절차 구성도

오브젝트의 실행절차 구성도는 다음의 3 가지 경우 로 구성 된다.

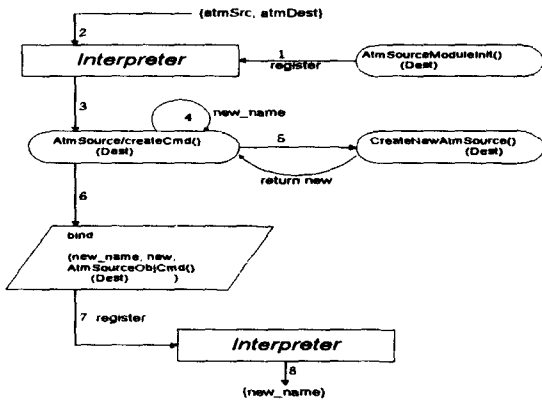
##### 4.2.1 오브젝트의 생성/초기화

(그림 11)은 atmSrc 오브젝트와 atmDest 오브젝트

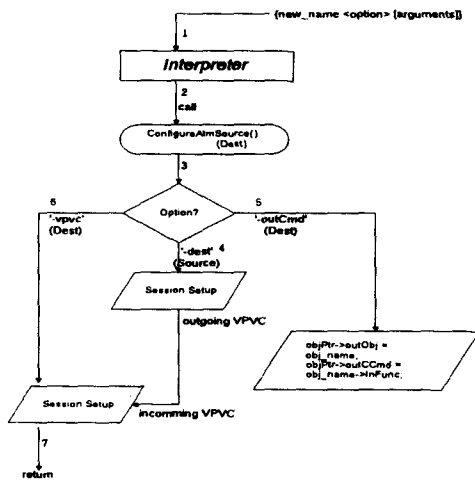
의 생성과정을 도시한 것이다.

### 4.2.2 오브젝트의 Configuring

(그림 12)은 이미 생성되어 초기화되어져 있는 atmSrc 오브젝트와 atmDest 오브젝트의 각 인스턴스의 슬롯들을 특정 값으로 configure하는 과정을 도시한 것이다.



(그림 11) 오브젝트의 생성/초기화  
(Fig. 11) Creation and initialization of object

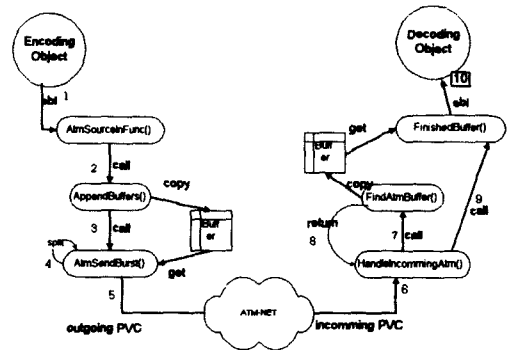


(그림 12) 오브젝트의 configuring  
(Fig. 12) Configuring of object

### 4.2.3 전송 및 수신 과정

(그림 13)는 ATM-NETWORK을 경유하여 전달되는 CM 스트림의 전송 과정을 서버와 클라이언트의

양 진영에서 도식적으로 보여준다.



(그림 13) CM sending/receiving  
(Fig. 13) CM sending/receiving

## 5. 실험 평가

MOD 서버에서의 전송 실험은 본 논문에서 구현한 ATM API 수신 모듈과 송신 모듈의 기능 및 성능을 테스트해 보기 위하여 실행되었다. 분산 MOD 환경 하에서는 예측할 수 없는 사태들이 많이 일어나기 때문에 이러한 실제 환경하에서 장시간 버틸 수 있는 견고성 실험과, 기존의 다른 전송 프로토콜을 이용해 개발되었던 전송 모듈과의 단위 시간당 전송을 비교 실험을 통하여 실제 MOD 환경하에서 이용할 수 있는가에 대한 적합성 여부가 판단되어 질 수 있다.

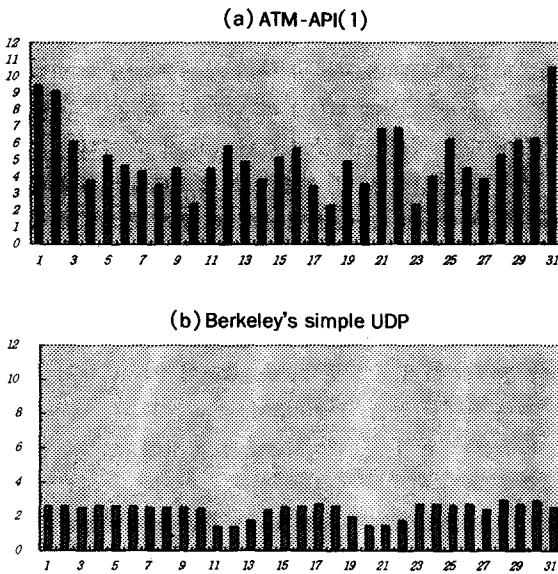
먼저 견고성에 대한 적합성을 테스트하기 위하여 다양한 해상도로 인코딩 되어진 20 개의 비디오 타이틀을 마련하였다. 한 타이틀씩 바꿔 가면서, 타이틀의 시작 시간 부터 끝 시간까지 전송 서비스를 시작하였고, 본 논문에서 구현한 송수신 알고리즘에 의해서 64K MTU 사이즈를 초과하는 프레임들에 대해서도 안전한 전송 결과를 나타내었다.

다음으로 단위 시간당 전송율을 확인해 보기 위한 실험을 하기 위하여 세 가지 비디오 타이틀을 마련하였다. 첫 번째 실험에 사용된 타이틀은 프레임들간의 명암값 및 화소값, 위치적 대조가 심한 3D 애니메이션 화일을 MJPEG 포맷으로 압축한 것이었다. 두 번째 실험에 사용된 타이틀은 '라이언 킹'이라는 만화 영화를 MJPEG 포맷으로 압축한 것으로 실험 그래프의 앞 부분에 fade in 효과를 감지할 수 있었다. 세 번

제 실험에 사용된 타이틀은 프레임 내의 오브젝트들 간에 움직임이 다이나믹한 영화를 MJPEG 포맷으로 압축한 것이었다.

본 실험에 사용된 MOD 서버는 75 MHz CPU 두 대를 장착한 SUN SPARC 20 시스템과 Solaris 2.3 플랫폼으로 구성이 되어졌으며, 640×480×24 bits의 한 타이틀 스트림당 평균 디스크 액세스 타임은 0.4 초의 성능을 가진다. 또한 ATM 어댑터로는 Fore SBUS-200을 사용하였으며, ATM SWITCH로는 ETRI의 CANS와 Fore의 ASX-200을 번갈아 가면서 사용하였다. 실험에 적용한 타이틀의 포맷은 MJPEG이었고, 프레임의 인코딩 해상도는 640×480×24 bits였다.

(그림 14)은 세 가지의 실험 중 첫 번째인 3D 애니메이션 타이틀에 대하여 기존의 UDP/IP와 socket 인터페이스로 구현이된 Berkeley의 Simple UDP 전송 모듈과의 단위 시간당 전송을 비교 결과를 보여 준다. 참고로 Simple UDP 모듈은 CMTToolkit 3.0에 포함된 것을 그대로 사용하였으며, 이 모듈 또한 32 KB 이상의 프레임에 대해 분할하고 재결합하는 루틴이 포함되어 있다.



(그림 14) MOD 서버에서의 초당 전송률 실험(가로: 초, 세로: Mbps)  
 (Fig. 14) Simulation of transmission rate for second in MOD server

본 논문에서 구현한 ATM-API 버전(그림 14의 (a))은 타이틀의 VBR(Variable Bit Rate)특성을 그대로 살려 주면서 버클리의 Simple UDP 버전 (그림 14의 (b))보다 월등히 좋은 성능 결과를 나타내었다. 또한 64 KB의 MTU 사이즈를 초과하는 프레임에 대해서도 안전한 전송 결과를 나타내었다. 그러나 UDP-IP를 통해 구현된 Simple UDP 모듈은 VBR의 특성을 살리지 못하고 오히려 CBR(Constant Bit Rate)의 효과를 나타내었으며, 상대적으로 저조한 전송을 결과를 나타내었다.

ATM-API 버전에서 평균 전송율은 5.162178 Mbps였으며, 최대 전송율은 10.530272 Mbps였다. 최대 프레임 사이즈는 69112 Bytes였다. Simple UDP 버전은 평균 전송율이 2.368105 Mbps였으며, 최대 전송율은 2.961696 Mbps였다. 최대 프레임 사이즈는 38804 Bytes였다.

## 6. 결 론

지금까지 본 논문에서는 ATM-LAN을 기반으로 하는 native ATM 서비스를 위한 MOD 시스템을 설계하고 구현하였다. 특히 본 논문에서 구현한 ATM-API 기반의 CM 전송기술은 ATM-LAN 환경하에서 MOD 서비스 구현을 위한 애플리케이션들에게 다음과 같은 특성들을 지원해줄 수 있는 툴킷으로 사용될 수 있도록 서버에서의 전송을 실험 등의 여러 가지 적합성 실험에 사용되었다.

- QoS Context를 사용하여 세션설정전에 QoS 협상 및 예약을 할 수가 있다.
- CM 전송 모듈은 기존의 전송 프로토콜들이 가졌던 불필요한 작업들을 최소화하기 위하여 트랜스포트 계층 및 IP 계층을 우회해 하위 AAL(ATM Adaptation Layer)과 애플리케이션을 직접 연결해 주는 ATM API(Fore's)를 사용한다.
- 브라우징 서버(BS)라는 제 3 자의 세션 관리자를 통해 서버와 클라이언트간에 투명한 세션(PVC)을 설정할 수 있다.
- 동시에 두 개 이상의 CM 스트림을 서로 다른 클라이언트들에게 전송할 수 있도록 객체지향 프로그래밍 기법이 사용되었으며, 아울러 동시에 여러 세션을 설정할 수 있다.

적합성 실험 결과 기존의 TCP/UDP/IP over ATM 과 같은 전형적인 미들웨어에서 발생하는 오버헤드의 제거로 인해 비디오 프레임들의 VBR의 특성을 그대로 살리면서 최적의 전송율을 올릴 수 있음이 수치적으로 제시되었다.

적합성 실험과 아울러 서버와 클라이언트간의 투명한 PVC 세션 설정에 관한 대안을 제시하였고, 실제 구현하여 사용해 봄으로써, 아직까지 표준화되어 있지 않은 ATM 기반의 MOD 서비스에 새로운 세션 설정 메카니즘을 제시하였다.

앞으로 진행되어야 할 연구내용들은 전송 기술에서 확장하여 전체적인 유폴에서 분산형 MOD를 조망하고, 최적의 서비스 메카니즘(QoS 협상/예약의 최적화, 오브젝트 및 세션 설정 시간의 최소화를 통한 실시간 응답시간의 구현, 분산형 서버들을 일괄적으로 관리하고 클라이언트들의 서비스 요구를 admission 해줄 수 있는 서비스 메카니즘 등)을 제공할 수 있는 시스템 모델을 모델링하는 것이다. 이 모델링을 통하여 ATM-API 전송 기술을 포함하는 최적의 MOD 환경을 이룰 수 있을 것이다.

### 참 고 문 헌

[1] Brian C. Smith, "Implementation Techniques for Continuous Media Systems and Applications", UC Berkeley, 1994.

[2] Rowerance A. Rowe, "Introduction to the CM Toolkit", UC Berkeley, 1994.

[3] P. Venkat Rangan, Harrick M. Vin, & Srinivas Ramanathan, "Designing an On-Demand Multimedia Service", IEEE Comm. Magazine, Vol. 30, No. 7, July 1992, pp. 56-64.

[4] Andre Zehl, P. Kusch, "IP over ATM: A Practical Performance Analysis", ICC 94.

[5] Joseph Y. Hui, Junbiao Zhang, and Jun Li, "Quality-of-Service Control in GRAMS for ATM Local Area Network", IEEE Journal on selected areas in communications. vol. 13, No. 4. May 1995.

[6] Harrick M. Vin, Pawan Goyal, Alok Goyal, and Anshuman Goyal, "A Statistical Admission Control Algorithm for Multimedia Servers", ACM

Multimedia 94, San Francisco, pp. 33-40, Oct. 1994.

[7] Tony King, "Pandora: An Experiment in Distributed Multimedia", Proceedings of Eurographics '92.

[8] Norihiko Sakurai, et al. "Multiple Access Server for Moving Picture Information", NTT Review Vol. 5, No. 5, SEP. 1993.

[9] Tatsuo Nakajima, et al., "A Continuous Media Application Supporting Dynamic QoS Control on Real-Time Mach", ACM Multimedia '94, San Francisco, pp. 289-297, Oct. 1994.

[10] FORE Systems, Inc., "ForeRunner SBA-100/-200 ATM SBus Adapter User's Manual", April, 1994.

[11] FORE Systems, Inc., "ForeRunner SBA-200 ATM SBus Adapter Reference Manual", April, 1994.

[12] David Aschkenasy, et al., "Video Multicast in an IP/ATM Environment", Oregon State University, April 12, 1995.

[13] Hong Heo, Kyoungmook Kook, Jang Rai Roh, et al., "Distributed Apparatus for Multimedia on Demand: Contents Placement and Dynamic Session Management", ETRI Journal, 1995.

[14] The ATM Forum Technical Committee, "Native ATM Services: Semantic Description", ATM Forum/95-0008R4 Straw Vote, Oct. 1995.



이 통신

### 이 근 왕

1993년 대전 산업대학교 전자계산학과 졸업(공학사)  
 1996년 숭실대학교 전자계산학과 졸업(공학석사)  
 1996년~현재 숭실대학교 전자계산학과 박사과정  
 관심분야: 멀티미디어, 멀티미디어

### 허 흥

1994년 숭실대학교 전자계산학과 졸업(공학사)  
 1996년 숭실대학교 전자계산학과 졸업(공학석사)  
 현재 한국통신기술연구소 연구원  
 관심분야: 멀티미디어, WWW



**김 봉 기**

- 1987년 숭실대학교 전자계산학과 졸업(공학사)
- 1989년 숭실대학교 전자계산학과 졸업(공학석사)
- 1995년 숭실대학교 전자계산학과 박사과정 수료
- 1994년~현재 한림전문대학 겸

퓨터응용과 조교수

관심분야: 멀티미디어, 멀티미디어 DB, 멀티미디어 통신



**오 해 석**

- 1975년 서울대학교 응용수학과 (학사)
- 1981년 서울대학교 대학원 계산통계학과(이학석사)
- 1989년 서울대학교 대학원 계산통계학과(이학박사)
- 1976년~1979년 태평양화학(주)

주임

1979년~1981년 (주) 삼호과장

1982년~현재 숭실대학교 정보과학대학 컴퓨터학부 교수

1995년~현재 정보통신부, 통상산업부, 내무부, 총무처, 경찰청, 특허청, 한국학술진흥재단 자문위원

관심분야: 멀티미디어, 데이터베이스, 영상처리