

# ATM환경의 분산 MOD시스템을 위한 프레임 기반의 CM전송 기술과 서비스 관리 모델링

이근왕<sup>†</sup> · 박세준<sup>†</sup> · 오해석<sup>††</sup>

## 요 약

본 논문에서는 ATM-LAN상 MOD서버에서 클라이언트로의 CM데이터의 전송 기술과 서비스 관리 모델을 제안한다. 즉, Navigation서버를 이용한 브라우징 메카니즘, 서버와 클라이언트, 응용, AAL과 Qos간 협상 및 예약 절차의 세션 설정 기술, 프레임의 분열 및 재결합에 관한 알고리즘을 제안하고 구현한 후, 실험 평가한다. 그리고 사용자에게 효율적인 서비스를 제공하기 위한 서비스 관리 모델로서 서비스 매니저를 모델링한다.

## A Frame-based Continuous Media Transmission Technique and Service Management for ATM-specific Distributed Multimedia-On-Demand System

KeunWang Lee<sup>†</sup> · SeJoon Park<sup>†</sup> · HaeSeok Oh<sup>††</sup>

## ABSTRACT

In this paper, we suggest a technique for delivering frame based CM data streams from MOD servers to clients and the service management model on ATM-LAN environment. We suggest and show results of implementing the transparent browsing mechanism using a navigation server, session setting procedures using PVCs between the server and clients, application, AAL and QoS negotiation and reservation procedures, splitting and reassembling algorithm for frames over MTU size. And we suggest the service manager model for supporting efficient services to users.

### 1. 서 론

최근 들어 급속히 발전하고 있는 멀티미디어 처리용 컴퓨터 기술과 고속 멀티미디어 통신 기술은 LAN과 같은 컴퓨터 망을 통하여 분리되어 있는 컴퓨터 시스템 사이에 대용량의 디지털 멀티미디어 데이터를 실시간으

로 주고받을 수 있는 환경을 조성하였다. 실시간 ATM 멀티미디어 서비스 중에 가장 유용한 서비스로 주문형 멀티미디어 검색 시스템을 들 수가 있다. 주문형 멀티미디어 검색 시스템은 일반적으로 클라이언트-서버의 구조로 구성되어 있다. 일반적으로 이러한 주문형 멀티미디어 검색 시스템을 MOD 시스템이라 부른다[15]. MOD 시스템은 크게 서버, 네트워크, 클라이언트 단말의 세가지 요소들로 구성이 되며, 멀티미디어 타이틀이라는 내용 기반의 미디어 객체를 통하여 상호 통신을 실현한다[7][8]. MOD에 관한 국내 동향으로는, ATM-LAN 환경하에서 음성과 동화상 전송을 위주로

\* 이 논문은 1997년도 숭실대학교 교내 학술 연구비 지원에 의하여 연구되었음.

† 준 회원 : 숭실대학교 전자계산학과 박사과정

†† 정 회원 : 숭실대학교 부총장

논문접수 : 1998년 3월 7일, 심사완료 : 1998년 4월 20일

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[4]. 국외 동향으로는 Berkeley의 CMTToolkit [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 응용 개발을 위한 native ATM 서비스용 전송/수신 모듈을 고안하고, 이것을 탑재하여 실제 서비스에 활용할 MOD 서비스 모델을 설계하고 구현하는데 있다. 또한 효율적인 서비스 제공을 위해서 서비스를 총체적으로 관리해주는 서비스 관리자 모델을 제안한다. 이는 복잡해지는 컴퓨팅 환경에서 최적의 서비스를 제공하고 자 하는데 의의가 있다. 여기서 'native ATM 서비스'라는 용어는 ATM UNI의 사용자측면에 상주하는 장치들내의 S/W 프로그램과 H/W를 직접적으로 조정하기 위한 ATM-specific 서비스를 말한다. 또한 native ATM API를 통하여 멀티미디어 전송 효율을 떨어뜨리는 수송계층과 IP 계층의 개입없이 응용과 AAL을 연결하여 데이터 전송, SVC/PVC 설정 및 세션 관리 등의 작업들을 효율적으로 처리할 수 있다[10][11].

## 2. Multimedia On Demand

### 2.1 Multimedia On Demand의 정의

MOD는 사용자가 브라우징 시스템을 통하여 검색을 원하는 타이틀을 선택하고, 논리적인 재생 시간에 따라 의미의 왜곡없이 연속적(continuous)이고, 정시적(timely)으로 검색할 수 있는 서비스를 말한다[3].

### 2.2 MOD의 구성 요소

MOD의 구성요소는 클라이언트, 네트워크, 서버의 3 가지로 구분이 된다. 클라이언트 단말에서는 반드시 타이틀 주문에 사용할 브라우저가 설치되어 있어야 하고, 타이틀을 프리젠테이션할 응용과 연결되어 있어야 한다. 또한 비디오 MPEG/MJPEG/H.261 decoder과 오디오 audio PCM decoder, 그리고 network adapter의 하드웨어 구성을 가진다. MOD 서버는 서버 브라우저와 MOD 서버 프로세스의 소프트웨어 구성과

CD-ROM, 하드 디스크의 하드웨어 구성을 가진다.

### 2.3 MOD의 세션 설정

ATM-API는 AAL과 응용 사이에 SAP (Service Access Point)을 연결하여 직접적으로 데이터를 주고 받을 수 있는 파일 기술자를 바인딩하여 응용에 반환한다. 응용은 반환된 화일 디스크립터로 incoming VP/VC(Virtual Path/Virtual Channel)를 통하여 네트워크를 경유해 온 데이터를 ATM 적용 계층으로부터 받을 수 있으며, outgoing VP/VC를 통하여 네트워크로 전달할 데이터를 ATM 적용 계층으로 직접 내려 보낼 수 있다[14].

ATM API의 장점은 TCP/UDP/IP와 같이 기존의 OSI 계층 관점에서 볼 때 수송 계층과 IP 계층을 거치지 않고 직접 ATM 적용 계층으로 데이터를 내려보냄으로써 기존의 미들웨어에서 발생하던 짐을 줄여 상대적으로 높은 전송률을 이룰 수 있다는 점이다. 또한, 매번 실행을 시작할 때 마다 자동적으로 ATM 적용 계층과 QoS 협상 및 예약을 할 수가 있으며, PVC를 사용하는 경우 ATM 교환기에 등록된 자신의 VP/VC를 자동적으로 사용할 수가 있어 사용자로부터 투명한 세션 설정 및 유지 과정을 보장할 수 있게 하여 준다.

Fore의 경우와 같이 ATM API는 크게 SVC용과 PVC용의 함수들로 분리되어진다. 응용의 서비스적 특성에 따라 이 두가지 중 어느 것을 선택해야 할 것인지를 결정하는 것이 바람직하다. SVC를 채택한 NV-ATM[13]과 PVC를 채택한 DIAMOND의 비교분석은 다음과 같다.

NV-ATM은 다자간 화상회의에 사용된다. 이는 같은 내용의 비디오 프레임이 동시에 여러 참가자들에게 전송되며 이러한 경우 커다란 대역폭 손실을 초래하기 때문에 스위치로 하나의 패킷을 복사하여 전송하고 그것을 전송받은 스위치가 참가자들 수 만큼 복사하여 각 참가자들에게 전송하는 방법을 사용하며 방송 형태를 가진다. 이는 SVC를 이용한 멀티캐스트 트리(Fore's SPANS)로 구현되었다.

DIAMOND는 VOD, MOD에 사용된다. 이는 주문형 전송 서비스가 목적이므로, 같은 타이틀이라 하더라도 각 클라이언트 당 유일한 프레임 복사가 요구되며, 개인적인 가상 VCR(FF,RWD,PAUSE)기능을 위한 독립적인 채널이 요구된다. 주문형 전송은 보통 프레임 단위의 주문을 위해 각 프레임 당 Its(logical time

stamp)를 찍어 클라이언트에게 전달하며, point-to-point 형태를 가진다.

NV-ATM의 장점은 멀티캐스트이므로 스위치까지 메시지 하나의 복사만 전달되어 대역폭을 줄이고, 새로운 가입자의 자유로운 세션 작업이 용이하다. 그러나 실제 회의에 참가하지 않는 호스트들도 메시지를 받아 볼 수 있거나, 주문형 서비스에 부적합한 점, SVC의 시그널링이 세션설정시 응답시간을 연장시킨다는 단점을 가진다.

DiMOND는 클라이언트당 개인적인 PVC를 갖고 세션을 운영하기 때문에 주문형 서비스(가상 VCR, Its 등)가 가능하고 각 채널 당 보안이 보장된다는 장점을 가지는 한편, 각 채널 당 독립적인 프레임들이 전송되므로 대역폭이 많이 소모되고, SVC보다 세션 설정 과정이 좀 더 복잡하다는 단점을 가지고 있다.

### 3. 프레임 기반의 CM송수신 기술

#### 3.1 CM의 송신 모듈

ATM-API를 이용한 전송 모듈은 Fore의 ATM API 라이브러리를 이용하여 작성되었다.

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 X 480 X 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 기법이 필수적으로 마련되어야 한다[9][12].

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

```

For (i = 0; i < # of frames[user]; i++) {
    Loop = 1; sent = 0;
    while (loop) {
        if ((sent = atm sendto(fd, &tbuf[i]{sent},

```

```

        (total + MTU > MTU)
        : (total), vpvcc) < 0) {
            error(atm sendto error);
            total += sent;
            if (total == 0) loop = 0; /* already sent
                all */
        }
    }
}

```

(그림 1) ATM-API 전송 모듈의 전송 알고리즘  
(Fig. 1) Sending algorithm of ATM-API transmission module

구체적인 알고리즘은 C언어의 형식으로 (그림 1)에 나타내었다.

#### 3.2 CM의 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 프레임으로 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. 본 논문에서는 (그림 2)에 보여진 알고리즘을 통하여 이 문제를 해결하였다.

```

Loop = 1;
Rec = 0; i = 0;
while(loop){
    i++;
    If((rec += atm_recvfrom(fd, &rbuf[rec].MTU,
        vpvcc) < 0) {error(atm_receive error);
    }
    if(recv == MTU) continue;
    loop=0;
}

```

(그림 2) ATM-API를 이용한 수신 모듈에서의 재결합 알고리즘  
(Fig. 2) Reconsolidation in receiving module using ATM-API

## 4. 서비스 매니저의 설계

### 4.1 기존의 서비스 제어 방법

#### 4.1.1 비계층적 네트워크 전송망

비계층 구조는 비디오 서버가 논리적인 계층을 형성하지 않고 모두 동일한 계층에 존재하며, 그물망으로 서로 연결된 서버에 설치되는 형태이다. 사용자는 어느 비디오 서버에도 연결되어 원하는 타이틀을 전송 받을 수 있다. 따라서 이 구조는 사용자의 선호도가 높은 프로그램을 저장하고 있는 특정 비디오 서버에 트래픽이 집중될 수가 있기 때문에 부하 분산을 위해 각 중심국에는 인기 있는 프로그램을 적절하게 분산시켜 저장할 필요가 있다.

#### 4.1.2 계층적 네트워크 전송망

계층 구조는 비디오 서버가 계층적으로 존재하는 구조이다. 상위 계층에 있는 서버는 MOD 서비스의 대상이 되는 모든 타이틀을 저장하고 있다. 하위 계층에 있는 비디오는 성형적으로 상위 계층의 서버와 연결되어 필요한 프로그램을 전송받고, 사용자와 직접 연결되어 요구한 프로그램이 없을 경우에는 상위계층의 서버로부터 제공받으며, 같은 계층에 있는 비디오 서버사이에는 타이틀 전송이 불가능하다. 따라서 계층구조에서는 선호도가 높은 타이틀을 하위계층의 비디오에 두어 사용자의 요구를 가능한 한 하위계층에서 처리함으로써 프로그램 전송망 내의 트래픽과 서버의 부하를 줄인다.

이러한 계층구조와 비계층 구조의 시스템을 구성하는데 있어 프로그램을 분산하여 저장하는 방법에 관한 연구는 다음과 같다.

첫 번째는 비디오 서버를 가지는 중심국을 3계층으로 나누는 구조이다. 각 중심국의 비디오 서버는 저장 용량의 제약이 있다는 가정하에 각 계층에 할당하는 타이틀의 저장 비용과 계층간의 링크 비용의 합을 최소화하는 기준으로 모든 타이틀을 각각의 계층에 어떻게 나누어 할당할 것인가를 결정함으로써 부하 분산을 해결하고자 했다. 그러나, 각 계층에 있는 각각의 중심국에 대해 타이틀을 어떻게 할당할 것인가는 고려하지 않는다는 단점을 가지고 있다.

두 번째로는 타이틀을 저장, 제공해 주는 비디오 서버를 위치시킬 수 있는 중심국의 계층이 여러 개 존재하는 구조에서 같은 계층에 있는 중심국들은 링크 비용이 없이 서로 정보를 교환할 수 있다는 가정 하에 두

가지 형태의 문제를 다루었다. 우선 여러 계층에 타이틀을 분산하여 저장하는 경우이고 다음으로는 하나의 계층에 모든 타이틀을 저장한다는 정책 하에 타이틀을 저장하는 계층을 결정하는 문제를 다루었다.

세 번째로는 비디오 프로그램을 저장하는 서비스 센터를 중심 비디오 서버와 지역 비디오 서버의 두 계층으로 구성하는 계층적 구조에서, 프로그램 저장비용과 트래픽으로 인한 링크비용으로 이루어지는 총비용의 최소화를 기준으로 지역 비디오 서버에 저장하는 프로그램 수를 결정하는 것을 다루었으나, 지역 비디오 서버 간에는 어떠한 연결도 없다고 가정을 하는 단점을 가진다.

따라서 위의 문제점들을 해결해 줄 수 있는 제어 시스템으로서 서비스 매니저를 설계한다. 서비스 매니저는 모든 서버의 상태를 파악하여 실시간 서비스를 가능하게 하는 등 서비스의 효율을 증가시킨다. 또한 선호도가 높은 타이틀을 분산 배치시키고 낮은 타이틀을 일정 기간후 삭제하는 기능등 다양한 기능들을 제공한다.

## 4.2 서비스 매니저의 모델링 및 시뮬레이션

### 4.2.1 서비스 매니저의 모델링

서비스 매니저는 서버와 클라이언트간에 비디오, 오디오 스트림을 전송하는 과정을 제어한다. 그러므로 서비스 매니저는 등록된 클라이언트의 정보를 관리하며, 스트림이 저장되어 있는 서버의 상태를 파악하여 클라이언트로부터의 서비스 요청 및 전송사 효과적인 서비스를 제공할 수 있도록 한다.

서비스 매니저가 효율적인 서비스를 위해서 가져야 할 기능들은 다음과 같다.

- 1) 클라이언트와의 양방향 VCC의 연결 및 해제 기능
- 2) 클라이언트의 서비스 등록 및 해제 기능
- 3) 클라이언트의 서비스 등록 여부 조회 기능
- 4) 타이틀 메뉴 검색
- 5) 요구된 타이틀이 실제로 있는 미디어 서버 중에서 부하가 가장 적고, QoS 보장이 가능한 미디어 서버를 선택하는 기능
- 6) 사용빈도가 높은 타이틀을 분류하여 분산 배치하는 기능

위의 기능들 중에서 가장 중요시되는 부분이 5)에서 언급한 Load Balancing과 6)에서 언급한 분산 배치이다. 그래서 이 두 가지를 살펴보도록 한다.

### - Load Balancing

서비스의 효율을 높여주기 위해서 각각의 타이틀을 저장하고 있는 서버의 상태를 파악하여 부하가 적게 걸려 있는 서버를 통해 서비스가 이루어지도록 해주는 작업이다. 이는 가장 부하가 적게 걸린 서버를 선택하여 최적의 서비스를 제공하기 위함이다. 이를 위해서 Cache Priority, Stream Serving Capacity, Network Bandwidth Capacity의 3가지 요소를 사용하여 서비스 상태가 양호한 서버를 선택하게 된다.

Cache Priority (Lcache)

이는 각각의 타이틀에 우선 순위를 계산하여 서버의 저장 공간의 상태를 파악하는 것이다. 임의의 서버가 우선 순위가 높은 타이틀을 많이 가지고 있게 되면 저장 공간이 많지 않으며 더욱이 그 타이틀의 삭제는 불가능하다. 그러므로 Cache Priority의 값이 높게 되면 서버의 상태가 좋지 않음을 나타내게 된다. 그러므로 전체에서 Cpri의 값을 빼주게 된다.

다음은 Cache Priority의 값을 계산하는 식을 나타낸 것이다.

$$L_{cache} = 1 - C_{pri} \quad C_{pri} = \sum_{n=1}^N O^n_{pri} \frac{O^n_{size}}{C_{size}}$$

여기서 Cpri는 평균 Cache Priority를 나타내며 Onsize는 각각의 오브젝트의 Priority를, Onsize는 오브젝트의 크기를, Csize는 사용 가능한 전체 저장 공간의 크기를 나타낸다. Lcache 값은 0과 1사이의 값을 가지게 되며 0은 서버의 상태가 좋지 않음을 나타내고 1은 서버의 상태가 최적임을 나타내게 된다.

Stream Serving Capacity (Lstream)

이는 서버의 실제 유용한 저장 공간의 상태를 파악하는 것으로서 타이틀을 제공할 수 있는 서버의 능력을 나타낸다. 다음은 Stream Serving Capacity의 값을 계산하는 식을 나타낸 것이다.

$$L_{stream} = \frac{ST_{max} - ST_{curr}}{ST_{curr}} \quad ST_{curr} = \sum_{n=1}^N O^n_{size}$$

여기서 STmax는 서버의 최대 저장 공간을 나타낸 것이고, STcurr는 현재 사용되고 있는 저장 공간의 크기를 계산한 것이다. 여기서도 Onsize는 오브젝트의 크기를 나타낸다. 여기서는 STcurr의 값이 STmax의 값보다 크게 되면 서비스는 불가능해 지게 된다. 여기서도 마찬가지로 Lstream값은 0과 1사이의 값을 가지

게 되며 0은 서버의 상태가 좋지 않음을 나타내고 1은 서버의 저장 상태가 최적임을 나타내게 된다.

Network Bandwidth Capacity (Lbw)

이는 네트워크의 대역폭을 계산하는 것으로서 현재의 네트워크의 상태를 파악하기 위한 것이다. 다음은 Network Bandwidth Capacity의 값을 계산하는 식을 나타낸 것이다.

$$BW_{max} = \frac{BW_{max} - BW_{curr}}{BW_{curr}} \quad BW_{curr} = \sum_{n=1}^N \frac{BW_{max}}{O^n_{size}}$$

여기서 BWmax는 서버의 최대 저장 공간을 나타낸 것이고, BWcurr는 현재 사용되고 있는 네트워크의 대역폭을 계산한 것이다. 여기서도 Onsize는 오브젝트의 크기를 나타낸다. 마찬가지로 BWcurr의 값이 BWmax의 값보다 크게 되면 서비스는 불가능해 지게 된다. 역시 위와 같이 Lbw값은 0과 1사이의 값을 가지게 되며 0은 서버의 상태가 좋지 않음을 나타내고 1은 서버의 네트워크 상태가 최적임을 나타내게 된다.

$$L = L_{cache}L_{stream}L_{bw}$$

위의 3개의 식을 다음과 같이 합성할 수 있다.

위의 식에서 계산된 값(L)은 서버를 선택하는 기준이 되어 높은 값을 가지는 서버를 선택하게 된다. 여기서 L값 역시 0과 1사이의 값을 가지게 된다.

또한 타이틀의 분산 배치를 하기 위해서는 비율을 통한 기준을 설정한다. 여기에 사용되어지는 요소로는 조회 시간, 조회횟수, 타이틀 시간 등의 요소가 있다. 이러한 요소들은 타이틀의 우선 순위를 나타내는 역할을 한다. 이 기준을 사용하여 우선 순위가 높은 타이틀은 몇 개의 서버에 분산 배치되어 서비스의 효율을 향상시킬 수 있다. 이러한 기준에 대한 비율을 수식으로 표현하면 다음과 같다.

$$Rate(\%) = Time / (Num * T\_Time) * 100$$

where, Time = 조회시간

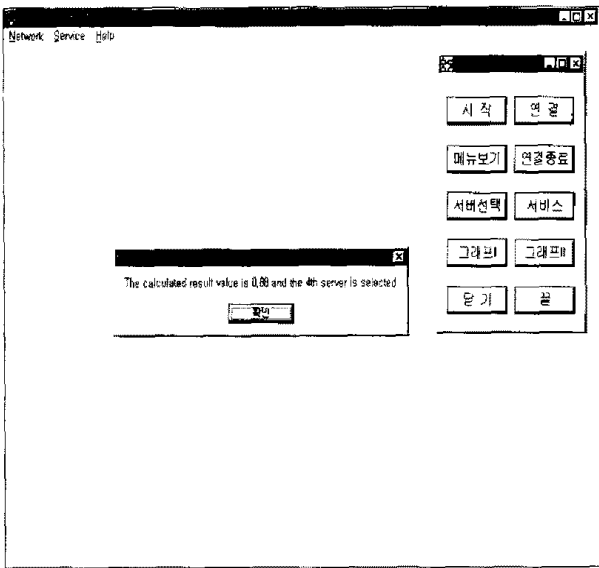
Num = 조회횟수

T\_Time = 타이틀 시간

위에서 서비스 매니저의 기능들을 바탕으로 서비스 매니저를 통한 서비스의 효율성을 분석하도록 하고자 한다.

(그림 3)과 (그림 4)는 위의 수식에 의해서 계산된 결과값과 계산되지 않은 랜덤한 결과값을 나타낸 것이

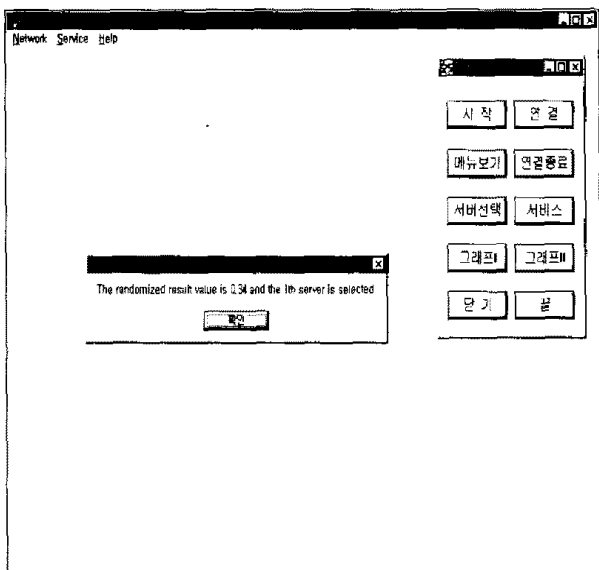
다. (그림 3)은 최적의 서버 상태값과 그 서버의 이름을 나타낸 것이며 (그림 4)는 랜덤하게 계산된 서버의 상태값과 그 이름을 나타낸 것이다. 여기서 서버의 상태값은 임의의 서버에 얼마나 많은 부하가 걸려 있는지를 판단해주는 값이다. 즉, 값이 0에 가까울수록 부하가 많이 걸려 있어 서버의 상태가 양호하지 않음을 나타내고 1에 가까울수록 부하가 없어 서버의 상태가 양호함을 나타낸다.



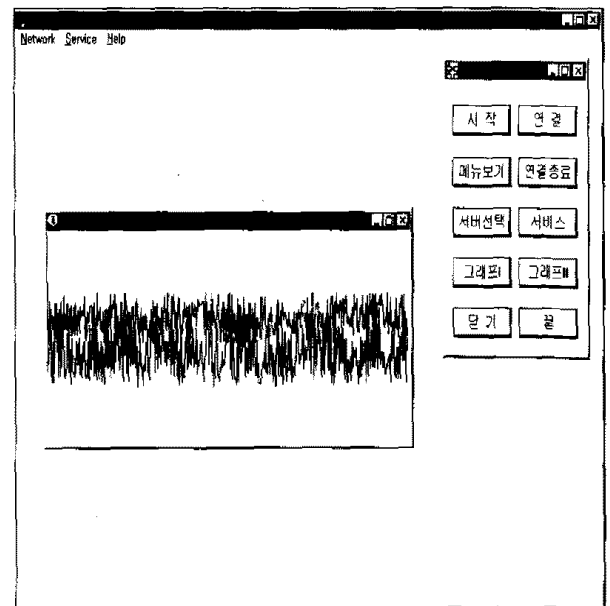
(그림 3) 계산된 결과값  
(Fig. 3) Result value by calculation

여기서 클라이언트/서버의 시뮬레이션은 스레드를 이용하는 방법을 사용하였다. 이는 스레드가 실행의 최소 단위이기 때문에 가능하였다. 임의의 10개의 스레드와 1개의 서비스 매니저 스레드를 생성하여 서버와 클라이언트의 역할을 하게 하였는데 이는 각각 10개의 스레드에 다른 일을 생성시켜 그 일을 처리하도록하고 또한 처리되는 시간을 달리하여 모든 클라이언트/서버의 상태가 동일하지 않게 하였다. 여기에 서비스 매니저 스레드라는 또 하나의 스레드를 생성하여 10개의 스레드의 상태를 파악하게 된다. 그래서 사용자가 원하는 시간에 어떠한 서비스가 요구되었을 때 서비스 매니저 스레드는 상태를 판단할 수 있는 값들을 각각의 스레드로부터 넘겨 받아 각각의 스레드의 상태정보를 파악하게 된다. 스레드의 실행은 `_beginthread`라는 함수의 호출로서 이루어지며 서비스 매니저 스레드는 `check_thread()`라는 함수를 사용하여 값을 검사한다. 서비스가 요구되었을 때의 스레드의 상황은 언제나 같지 않기 때문에 넘겨지는 값들은 항상 갱신되게 된다. 이러한 상태정보를 사용하여 어떠한 스레드가 가장 좋은 상태인지를 판단하여 서비스의 QoS를 증가시킨다.

(그림 5)는 두 개의 결과값을 시뮬레이션 한 것이다. 그래프에서 알 수 있듯이 알고리즘에 의해서 계산된 값이 랜덤하게 계산된 값보다 높은 평균값을 갖는 것을 알 수 있다.



(그림 4) 랜덤한 결과값  
(Fig. 4) Result value by no-calculation



(그림 5) 시뮬레이션 그래프  
(Fig. 5) Simulation graph

### 5. 구현

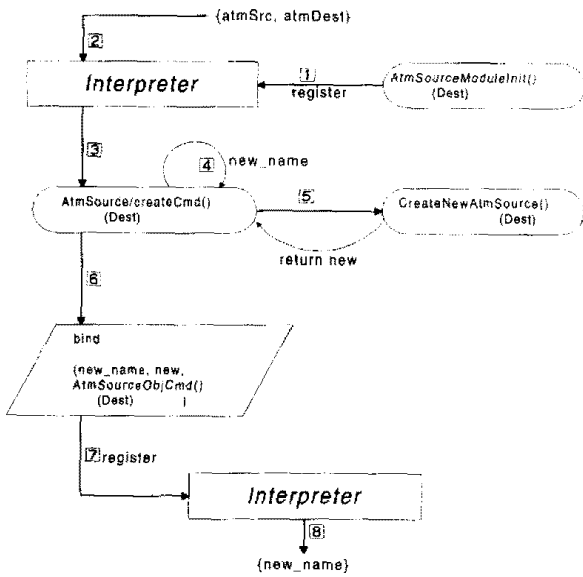
프로그램은 객체지향 프로그래밍 기법을 적용하여 C언어로 구현하였고, Fore의 ATM API 라이브러리, Tcl-Tk 라이브러리, Tcl-DP 라이브러리등을 사용하였다. 데이터 채널에서 제어 정보 채널은 TCP/IP & UDP/IP over ATM을 미들웨어로 하여 실험하였다. CM 스트림 채널은 Fore's ATM API를 미들웨어로 하여 실험하였다.

#### 5.1 오브젝트의 실행절차 구성도

오브젝트의 실행절차 구성도는 다음의 3 가지 경우로 구성된다.

##### 5.1.1 오브젝트의 생성/초기화

(그림 6)은 atmSrc 오브젝트와 atmDest 오브젝트의 생성과정을 도시한 것이다.



(그림 6) 오브젝트의 생성/초기화  
(Fig. 6) Creation/Initialization of objects

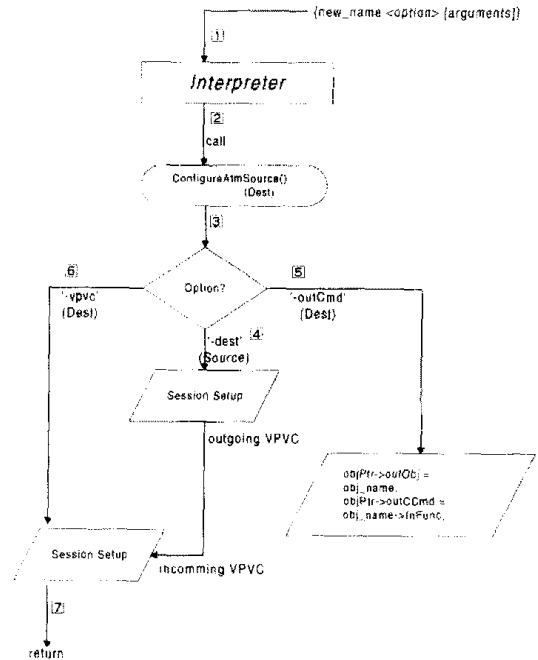
##### 5.1.2 오브젝트의 Configuring

(그림 7)은 이미 생성되어 초기화되어 있는 atmSrc 오브젝트와 atmDest 오브젝트의 각 인스턴스의 슬롯들을 특정 값으로 구성하는 과정을 도시한 것이다.

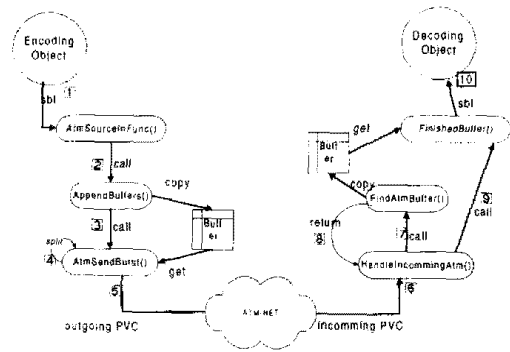
##### 5.2.3 전송 및 수신 과정

(그림 8)은 ATM망을 경유하여 전달되는 CM 스트

림의 전송 과정을 서버와 클라이언트의 양 지점에서 도식적으로 보여준다.



(그림 7) 오브젝트의 Configuring  
(Fig. 7) Configuring of objects



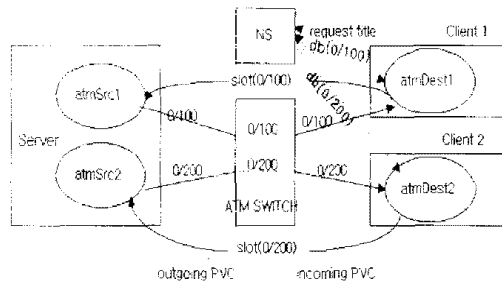
(그림 8) CM 송신/수신  
(Fig. 8) CM Sending/Receiving

#### 5.3 MOD서버/클라이언트간 1:N CM스트림 전송을 위한 시나리오

분산 MOD서비스 환경하에서 MOD서버와 클라이언트간에 1:N구조를 형성하는 것은 실세계에 적용하기 위한 필수적인 요구사항이라고 하겠다.

구체적으로 서버와 클라이언트간의 관계를 (그림 9)에 나타내었다. 서버와 클라이언트간의 데이터 채널은

크게 비디오 스트림 채널과 제어 메시지 채널로 나뉠 수가 있다. 비디오 스트림 채널은 PVC 세션 과정을 통하여 개설될 수 있으며, 본 논문에서 구현한 ATM-API 전송 모듈을 통하여 실제 비디오 타이틀 스트림을 전송할 수 있다. 컨트롤 채널은 서버에서 클라이언트로 데이터를 전송하기 위해 개설된 것이 아니고, 역으로 클라이언트에서 서버로 제어 정보가 넘어가는 통로를 말한다. 제어 정보들은 크기가 몇 바이트 이내인 극히 소수의 양이기 때문에 본 논문에서는 TCP/IP를 이용한 RPC를 이용해서 구현하였다.



(그림 9) 서버와 클라이언트간의 1:N 통신 구조  
(Fig. 9) Communication architecture between one server and n client

우선 한 서버가 동시에 여러 클라이언트에게 서로 다른 혹은 동일한 비디오 타이틀을 전송하기 위해서는 서버측의 송신 모듈들이 오브젝트 구조로 구성되고 실행되어야 한다. 각각의 오브젝트들은 송신 모듈이라는 하나의 클래스에 속하며, 개개의 독립적인 인스턴스로 구분된다. 각각의 오브젝트들은 자신만의 속성 슬롯들을 가진다. 이 슬롯들 중에서 세션에 관여하는 슬롯이 VP/VC이라는 슬롯이며, 클라이언트의 슬롯이라는 RPC 명령을 통해 채워지게 된다. 일단 이 슬롯이 채워지게 되면 전송 오브젝트는 이 슬롯을 참조하여 참조된 VPVC 통로로 데이터를 전송하게 된다. 각각의 오브젝트들마다 서로 다른 슬롯값을 가지기 때문에 동시에 서로 다른 클라이언트들에게 타이틀 전송이 가능하다.

## 6. 실험 평가

MOD 서버에서의 전송 실험은 본 논문에서 구현한 ATM API 수신 모듈과 송신 모듈의 기능 및 성능을 실험해 보기 위하여 실행되었다. 분산 MOD 환경하에서는 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 전송율 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 두 가지 비디오 타이틀을 마련하였다. 첫번째 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 앞부분에 fade in 효과를 감지할 수 있었다.

본 실험에 사용된 MOD 서버는 75 MHz CPU 두 대를 장착한 SUN SPARC 20 시스템과 Solaris 2.3 플랫폼으로 구성이 되어졌으며, 640 X 480 X 24 bits의 한 타이틀 스트림당 평균 디스크 액세스 타임은 0.4초의 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 X 480 X 24 bits였다.

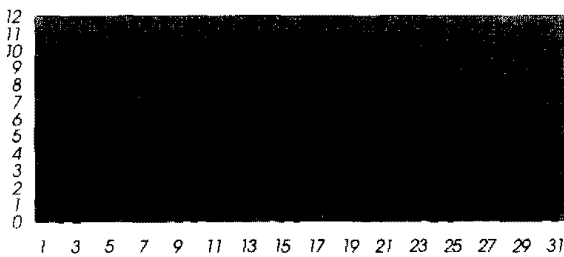
(그림 10)은 두 가지의 실험 중 첫 번째인 3D 애니메이션 타이틀에 대하여 기존의 UDP/IP와 socket 인터페이스로 구현이 된 Berkeley의 Simple UDP 전송 모듈과의 단위 시간당 전송율 비교 결과를 보여 준다. 참고로 Simple UDP 모듈은 CMTollkit 3.0에 포함된 것을 그대로 사용하였으며, 이 모듈 또한 32 KB 이상의 프레임에 대해 분할하고 재결합하는 루틴이 포함되어 있다.

본 논문에서 구현한 ATM-API 버전(그림 10의 (a))은 타이틀의 VBR특성을 그대로 살려 주면서 버클리의 Simple UDP 버전 (그림 10의 (b))보다 월등히 좋은 성능 결과를 나타내었다. 또한 64 KB의 MTU

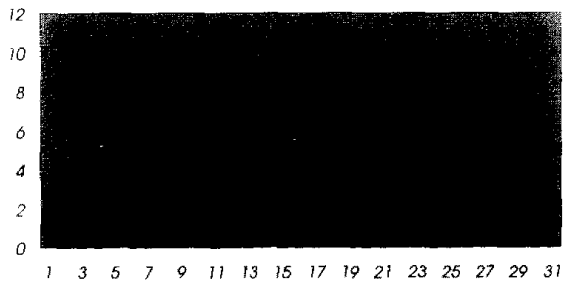


요기를 초과하는 프레임에 대해서도 일정한 전송 결과를 나타내었다. 그러나 UDP-IP를 통해 구현된 Simple UDP 모듈은 VBR의 특성을 살리지 못하고 오히려 CBR의 효과를 나타내었으며, 상대적으로 저조한 전송을 결과를 나타내었다. ATM-API 버전에서 평균 전송율은 5.162178 Mbps였으며, 최대 전송율은 10.530272 Mbps였다. 최대 프레임 사이즈는 69112 Bytes였다.

(a) ATM-API(1)



(b) Berkeley's Simple UDP

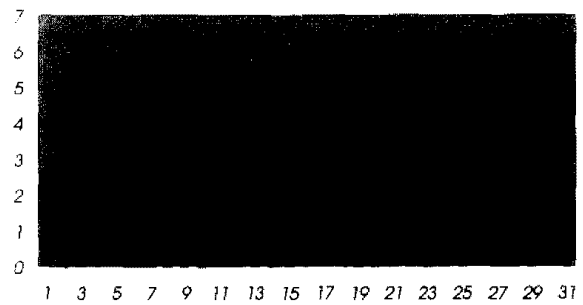


(그림 10) MOD 서버에서의 초당 전송률 실험  
(Fig. 10) Transmit rate per second experiment on MOD server

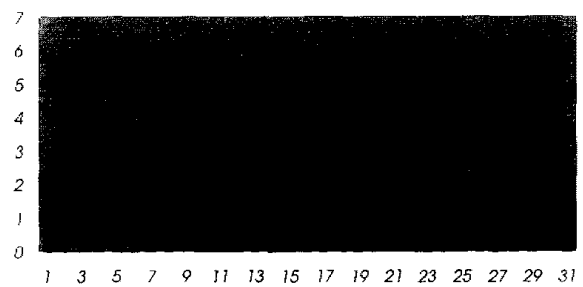
Simple UDP 버전은 평균 전송율이 2.368105 Mbps였으며, 최대 전송율은 2.961696 Mbps였다. 최대 프레임 사이즈는 38804 Bytes였다.

두번째 실험은 라이온 킹 MJPEG 타이틀인데, 본 실험에서 ATM-API버전(그림 11)의 (a)는 평균 전송율이 4.417141Mbps 이었으며, 최대 전송율은 6.069888 Mbps이었다. 최대 프레임 사이즈는 55056 Bytes 이었다. Simple UDP버전(그림 11)의 (b)는 평균 전송율이 2.473468Mbps 를 나타냈으며, 최대 전송율은 2.614048Mbps를 나타내었다. 최대 프레임 사이즈는 32108Bytes이었다.

(a) ATM-API(만화영화)



(b) Berkeley's Simple UDP



(그림 11) MOD 서버에서의 초당 전송률 실험  
(Fig. 11) Transmit rate per second experiment on MOD server

## 7. 결 론

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 응용들에게 툴킷으로 사용될 수 있도록 서버에서의 전송률 실험 등의 여러 가지 적합성 실험에 사용되었다.

적합성 실험 결과 기존의 TCP/UDP/IP over ATM과 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, 아직까지 표준화되어 있지 않은 ATM 기반의 MOD 서비스에 새로운 세션 설정 매커니즘을 제시하였다.

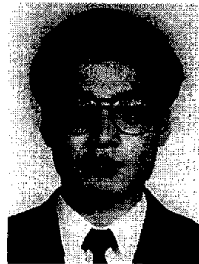
또한 복잡하고 분산된 환경에서 서비스를 효과적으로 제공할 수 있게 해주는 서비스 매니저를 설계하고 시뮬레이션 하였다.

앞으로 진행되어야 할 연구내용들은 전송 기술을 향상하여 전체적인 유곽에서 분산형 MOD를 조명하고, QoS 협상/예약의 최적화, 오브젝트 및 세션 설정 시간의 최소화를 통한 실시간 응답시간의 구현, 분산형 서버들을 일괄적으로 관리하고 클라이언트들의 서비스 요구를 허락해 줄 수 있는 서비스 메카니즘 등의 최적의 서비스 메카니즘을 제공할 수 있는 시스템 모델을 모델링하여 최적의 주문형 멀티미디어 환경을 이루는 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 reas 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] The ATM Forum Technical Committee, "Native ATM Services: Semantic Description", ATM Forum/95-0008R4 Straw Vote, Oct. 1995.
- [14] 허 홍, 이 근왕, 김 봉기, 오 해석, "Native ATM Service를 위한 MOD System의 구현", 한국정보처리학회 논문지 제 4 권 제 6 호, 1997.
- [15] Lawrence A. Rowe, David A. Berger, and J. Eric Baldeschwieler, "The Berkeley Distributed Video On Demand System", University of Berkeley, CA 94720-1776.

### 이 근 왕



1993년 대전 산업대학교 전자계산학과 졸업(공학사)  
 1996년 숭실대학교 전자계산학과 졸업(공학석사)  
 1996년~현재 숭실대학교 전자계산학과 박사과정  
 1998년 1월 1일~현재 초고속 선도 시험망 공동이용센터 근무

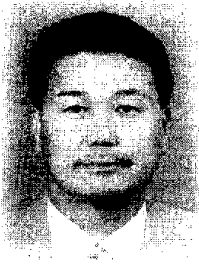
관심분야 : 멀티미디어 응용, 멀티미디어 통신

### 박 세 준



1996년 숭실대학교 수학과 졸업(이학사)  
 1998년 숭실대학교 전자계산학과 졸업(공학석사)  
 1998년~현재 숭실대학교 전자계산학과 박사과정

관심분야 : 멀티미디어, 멀티미디어 통신



## 오 해 석

- 1975년 서울대학교 응용수학과 (이학사)
- 1981년 서울대학교 계산통계학과 (이학석사)
- 1989년 서울대학교 계산통계학과 (이학박사)

1982년~현재 숭실대학교 정보과학대학 교수

1976년~1982년 태평양화학(주), (주)삼호 전산실

1990년~1991년 일본 동경대학교 객원교수

1997년~현재 숭실대학교 부총장

관심분야 : 멀티미디어, 데이터베이스, 영상처리