

# ATM/B-ISDN 기반의 원격 의료정보 시스템을 위한 멀티미디어 데이터베이스 원격 접속기능 설계 및 구현

김호철<sup>†</sup> · 김영탁<sup>\*\*</sup>

## 요 약

멀티미디어 원격 의료 정보 시스템에서는 멀티미디어 형태로 저장 관리되는 환자의 의료정보를 신속하게 원격 검색할 수 있어야 한다. 또한, 대용량의 멀티미디어 의료정보를 효율적으로 관리하기 위한 멀티미디어 DBMS가 사용되어야 하며, 분산처리 환경에서의 원격 검색 기능이 구현되어야 한다. 멀티미디어 원격 의료정보 시스템과 같이 실시간 정보 전송 및 각 정보 형태별 연결 관리가 필요한 경우의 DB 원격 검색을 위해서는 Native ATM Service와 같이 개별 연결설정 및 QoS(Quality of Service)를 보장하는 초고속 정보 통신망이 필요하다. 멀티미디어 DB의 원격 검색을 위해 상용 DBMS가 제공하는 API를 이용할 경우 해당 DBMS만을 지원하는 DBMS 의존적인 멀티미디어 원격 의료 정보 시스템이 되어 병원 규모 및 특성에 맞는 DBMS의 선정 및 DB 구축이 어렵게 된다. 또한, 상용 DBMS가 제공하는 TCP/IP Socket 기반의 전송 방식으로는 전송 특성이 각기 다른 멀티미디어 데이터의 개별적인 연결관리 및 QoS 보장이 힘들다. 그러므로 본 논문에서는 멀티미디어 원격 의료 정보 시스템을 위한 멀티미디어 DB 원격 접속 기능 구현에서 현재 상용 DBMS가 제공하지 않는 Native ATM API를 사용한 DBMS 원격 접속 기능 구조를 제안하고, 이를 기반으로 한 원격 검색 기능을 구현 그 성능을 분석한다.

## A Design & Implementation of Remote Access Function for A Multimedia Database of The Tele-medical System Based on ATM/B-ISDN

Ho-Cheal Kim<sup>†</sup>, Young-Tak Kim<sup>\*\*</sup>

## ABSTRACT

In the multimedia tele-medical system the medical informations are stored and managed in multimedia database. Also, multimedia DBMS is essential in order to manage large scale medical informations, and the remote access function is necessary for the distributed processing at all around the hospital. For the multimedia tele-medical information that is composed of image/picture, data, video, and audio, a high-speed telecommunication network is necessary that can provide separated connections for each medical information type with different QoS. The commercial DBMSs are based on the TCP/IP socket API(Application Programming Interface) that does not provide multiple QoS. Also, each commercial DBMS has its own API that is incompatible with other DBMS.

In this paper, we propose a multimedia DBMS agent for the remote access of the multimedia database in the tele-medical system. The proposed multimedia DBMS agent is based on the ATM API that can provide high-speed data transfer capability and multiple QoS connections. Also, the proposed multimedia DBMS agent is independent of the commercial DBMSs. We explain the functional architecture of the multimedia DBMS agent, implementation technology on the ATM network environment, and the result of performance analysis.

<sup>†</sup> 영남대학교 대학원 멀티미디어 통신공학과

<sup>\*\*</sup> 영남대학교 공과대학 정보통신공학과

## 1. 서론

의료기관에서 환자의 진료와 관련하여 사용하는 진료 정보는 텍스트, 이미지, 비디오, 오디오와 같은 멀티미디어의 형태로 구성된다. 멀티미디어 데이터의 경우 각 미디어의 전송 특성이 각기 다르고 데이터의 크기가 크고 다양하다. 이러한 멀티미디어 의료 데이터를 원격으로 검색하고 활용하기 위해서는 생명과 관련된 의료정보의 특성상 실시간 검색을 요구하며 각 미디어를 개별적으로 연결관리할 수 있는 특성을 갖는 고속의 통신망이 필요하다. 현재 광범위하게 사용중이며 상용 DBMS API가 원격 접속을 위해 지원하는 TCP/IP 통신망에서는 각 미디어의 개별적인 전송관리가 힘들며 전송 QoS를 보장할 수 없다. 각 미디어별 QoS를 지정할 수 있는 통신망으로 현재 사용 가능한 것으로는 ATM 통신망이 있으나 상용 DBMS API가 아직까지는 Native ATM API를 지원하지 않고 있지 않다.

ATM 통신망 서비스의 다른 응용인 LAN Emulation이나 IP over ATM을 사용할 경우 별도의 기능 설계 없이 원격 검색 기능을 구현할 수 있으나 이 경우, 대역폭의 증가 이외는 TCP/IP의 특성을 그대로 따르기 때문에 각 미디어의 개별적인 연결관리는 힘들게 된다[1-3]. 따라서, 멀티미디어 데이터들의 전송 특성을 만족하는 Native ATM Service를 사용한 데이터베이스 원격 접속기능을 구현하기 위해서는 데이터 전송 구간을 데이터베이스 API와 독립된 구조로 원격 접속기능을 구현하는 것이 필요하다.

본 논문에서는 이와 같은 멀티미디어 의료정보를 데이터베이스로 구축하고, 멀티미디어 원격 의료회의 기능을 통한 첨단 의료 서비스 제공을 하기 위한 멀티미디어 원격 의료 시스템 개발에서의 원격 데이터베이스 접속 기능을 설계, 구현하고, 그 성능을 분석한다[1,2].

## 2. 원격 의료 정보 시스템에서의 멀티미디어 DBMS와 구현 환경

그림 1은 ATM 통신망을 기반으로 멀티미디어 원격 의료 시스템을 구성한 것으로, 각 병원별로 분산된 데이터베이스와 멀티미디어 원격 의료 회의를 진행

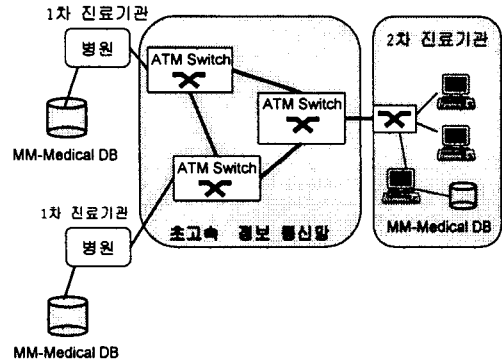


그림 1. 원격 의료 정보시스템 구성도

하기 위한 개별 화상회의 시스템으로 구성되어 있다.

멀티미디어 데이터를 데이터베이스 시스템으로 구축하기 위해서는 고속의 검색이 가능하고, 데이터의 증가에 따른 속도 저하 현상이 없으며, 텍스트뿐 아니라 다양한 형태의 데이터를 지원하는 DBMS가 필수적이다. 현재 많이 사용되고 있는 DBMS로는 관계형 DBMS, 객체지향형 DBMS, 객체관계형 DBMS등이 있다. 관계형 DBMS는 SQL을 이용한 강력한 검색기능을 제공하지만 텍스트를 기반으로 한 DBMS로 설계되어 멀티미디어 데이터에 대한 고려가 미흡하고, 객체지향형 DBMS의 경우 멀티미디어 데이터의 취급이 용이한 반면 데이터의 검색이 함수 형태로 이루어짐으로 인해 검색 기능이 관계형 DBMS에 비하여 미흡하다. 객체관계형 DBMS는 관계형 DBMS의 특성을 그대로 유지하면서 객체개념을 도입한 형태의 DBMS로 멀티미디어 데이터베이스를 목적으로 설계되었다. 본 연구에 사용된 DBMS인 Illustra는 객체관계형 DBMS로 Blade 모듈이라는 확장 개념을 도입하여 멀티미디어 데이터를 지원한다[4].

원격 의료 정보 시스템의 구현 환경으로는 그림 2에서처럼 Solaris 2.5 OS의 SUN 워크스테이션 3대와 Pentium PC 2대를 사용하였다. 네트워크 장비로는 Fore ASX-200 ATM 스위치와[5] ForeRunner SBA-200 Adaptor를[6] 장착한 워크스테이션 3대를 광케이블로 연결하고, ForeRunnerLE PCI Bus PC Adaptor를[7] 사용하는 PC 2대를 UTP로 연결하였다.

소프트웨어 구현 환경은 Illustra DBMS API[8], Fore XTI ATM API[9] 및 winsock2를 사용하였다.

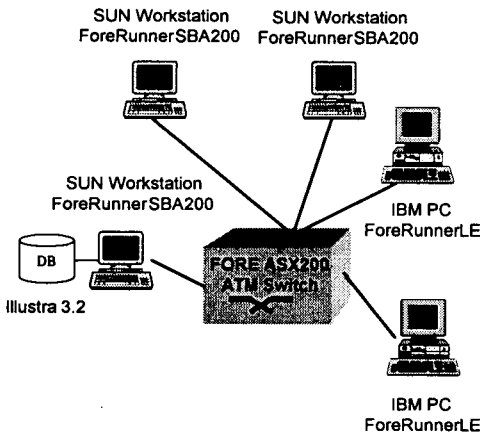


그림 2. 멀티미디어 DB 원격 접속기능 구현 하드웨어 환경

### 3. ATM API를 이용한 원격 DB 접속기능 설계

상용 DBMS 제공 API를 이용한 원격 검색기능 구현은 그림 3과 같은 구조를 가진다. 이러한 구조는 원격 검색 프로그램 내에 상용 DBMS가 제공하는 DBMS 접속 API를 포함하고 있어 DBMS에 의존적이게 됨으로 다음과 같은 문제점을 안고 있다. 첫째로, 원격 의료 정보 시스템을 설치하는 모든 병원이 동일한 DBMS를 사용하여야 한다. 이는 병원 규모에 적합한 DBMS의 선택 및 데이터베이스 구축에 걸림돌이 된다. 둘째로, DBMS API가 ATM 통신망(Native ATM)을 지원하지 않으므로 대용량 멀티미디어 데이터 전송시 각 미디어별 전송 특성에 맞는 QoS를 설정할 수 없다. 셋째로, 향후 DBMS를 교체하거나 갱신할 경우에 원격 의료 정보 시스템(DB

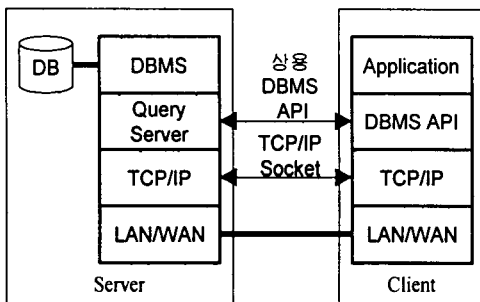


그림 3. DBMS 제공 API를 이용한 검색기능 구조

서버 및 모든 Client)을 수정하여야 한다. 이와 같은 문제점을 해결하기 위해서는 DBMS에 독립적이면서 Native ATM Service를 지원하는 구조의 원격 검색기능 구현이 필요하다.

본 논문에서는 그림 4와 같이 상용 DBMS API를 이용한 원격 접속기능을 계층화된 에이전트의 기능으로 구현하고, DBMS와 동일 시스템 상에 존재시킴으로써 데이터베이스 검색을 대행하도록 하였으며, 이를 ATM 통신망에서 사용하기 위해 에이전트와 원격 검색 프로그램간의 연결설정, 데이터전송에 관련된 DBMS 에이전트 API를 설계 및 구현하였다 [10,11].

그림 4에서 제안한 에이전트를 이용할 경우 Client에 설치되는 원격 검색 프로그램 내에서 상용 DBMS 제공 API를 사용하지 않게 되며, DBMS에 독립적인 분산처리 환경을 구현할 수 있다. 에이전트와 원격 검색 프로그램 사이의 ATM 연결 설정은 FORE Native ATM API의 SVC(Switched Virtual Connection)를 이용하여 구현하고, B-ISDN 서비스 유형 중 가변 비트율의 연결형, 비연결형 서비스가 가능한 AAL5를 사용하였다[3].

DBMS 에이전트는 제어 에이전트와 서비스 에이전트의 두 가지 기능으로 구성된다. 그림 4의 검색기능구조는 데이터베이스 질의를 수행하고 검색요청 및 결과를 전송하는 서비스 에이전트를 도식화한 것이다. 제어 에이전트는 초기 연결설정 및 관리 기능을 수행한다. 제어 에이전트에 의하여 서비스 에이전트가 다중 활성화될 수 있도록 하여 다중연결 서비스가 지원되도록 하였다. 즉, 원격 검색 프로그램의 연결

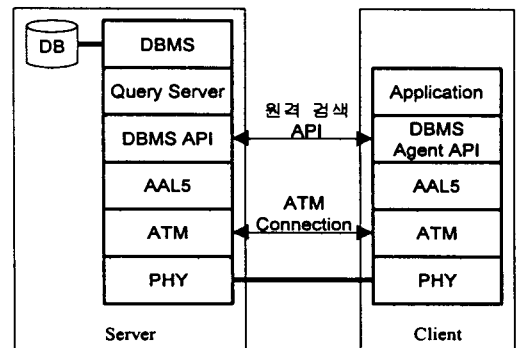


그림 4. 에이전트에 의한 검색기능 구조

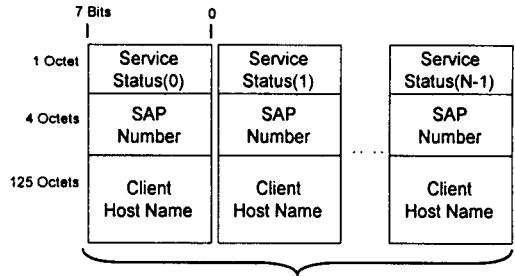
요청을 제어 에이전트가 수신하게 되면 서비스 가능 여부를 확인한 후 서비스 에이전트 하나를 활성화하고 연결 요청 프로그램에 할당함으로써 다중 연결설정이 수행된다.

#### 4. 원격 접속 에이전트 기능의 설계 및 구현

##### 4.1 제어 에이전트의 기능 구현

제어 에이전트는 데이터베이스에 직접적인 접근을 수행하지 않는다. 그림 5에서 보여주는 제어 에이전트는 AAL5를 이용하여 연결에 필요한 정보를 DBMS Client에 설치된 원격 검색 프로그램과 송수신한다. DBMS 서버에 설치되는 제어 에이전트의 내부에는 두 종류의 Thread가 존재하는데[12], 원격 검색 프로그램들로부터 연결 요청을 수신하는 Listen Thread와 수신된 메시지로부터 서비스 여부를 결정하고 서비스 에이전트를 활성화하는 Accept Thread가 그것이다. 이들 두 Thread를 통칭 Listen/Accept Coordinator라 부른다[12].

Accept Thread의 서비스 가능 여부 결정은 CCD (Connection Control Data) 자료구조의 빈 슬롯을 검색함으로써 수행된다. 빈 슬롯이 확인되면 제어 에이전트는 서비스가 가능한 것으로 인식하고 연결 설정 절차를 수행하며, 만약 빈 슬롯이 없을 경우에는 서비스가 불가능함을 Client의 원격검색 프로그램에 통보한다. 제어 에이전트 구조에서 CCD 자료구조는 연결관리에 필요한 정보를 저장하는 구조체이며, 그림 6과 같이 슬롯의 사용 여부를 결정하는 서비스



N개의 원격 단말기 연결 가능

그림 6. Connection Control Data (CCD) 자료구조

상태 필드, 서비스 에이전트의 SAP(Service Access Point) 번호를 보관하는 SAP 번호 필드, 원격 검색 프로그램이 동작하고 있는 컴퓨터의 ATM 주소를 저장하는 서비스 주소 필드로 구성된다.

Listen Thread는 제어 에이전트가 실행 될 때 Native ATM API를 초기화하고 원격 검색 프로그램들로부터의 연결 요청을 기다린다. 연결 요청은 ATM/B-ISDN Signaling에 의하여 수행되는데 원격 검색 프로그램이 자신이 필요한 QoS를 이용하여 연결을 요청하면 Listen Thread는 CAC(Call Admission Control)를 통하여 연결 수락을 결정하게된다. 일단, 연결이 설정되면 Listen Thread는 서비스 연결에 필요한 정보를 원격 검색 프로그램으로부터 수신한 후 Accept Thread를 하나 활성화시키고 다음 연결 요청을 기다린다. Listen Thread는 제어 에이전트가 종료할 때까지 수행되는 반면 Accept Thread는 서비스의 가능 여부가 결정이 되고 해당 동작이 취해지면 종료 하게된다.

그림 7은 제어 에이전트의 초기 연결설정 절차와 전송 패킷 구조이다. Packet Type 필드는 전송되는 패킷이 연결요청, 연결수락 또는 연결거부인지를 표시하고, Payload Size는 Payload 필드에 기록된 데이터의 크기를 나타낸다. Payload 필드에는 서비스 프로그램과, 에이전트가 주고받는 데이터가 기록되는데, 연결요청일 경우 서비스 프로그램 호스트명이, 연결수락의 경우는 서비스 에이전트의 SAP(Service Access Point) 번호가 기록된다. 연결거부의 경우는 NULL 데이터를 기록한다. CRC(Cycle Redundancy Check) 필드는 어플리케이션에서의 에러검사를 위해 예약된 필드로 초기 연결설정에서는 사용하지 않았다.

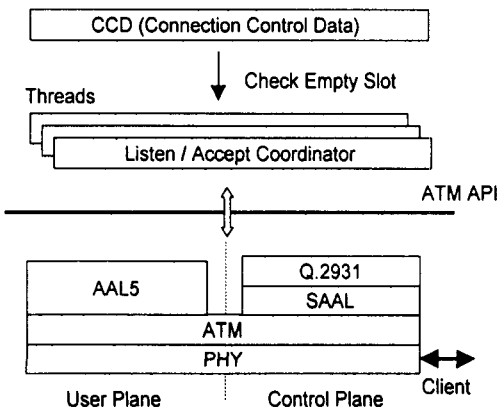
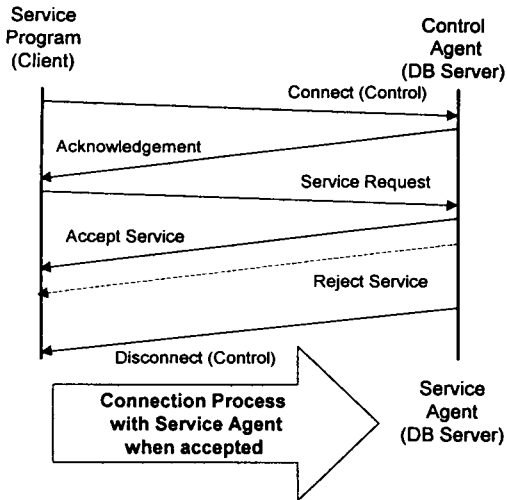
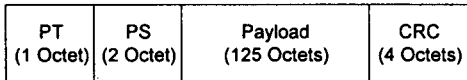


그림 5. 제어 에이전트 구조



(a) 초기 연결설정 절차



PT: Packet Type (Request, Accept, Reject, Disconnect)  
PS: Payload Size (Real loaded size)

• 제어 패킷

- PT : Payload에 저장된 정보 형태
- PS : Payload에 저장된 유효 데이터 크기
- Payload : 연결설정, 해제에 필요한 정보
- CRC : Application에서의 에러검출 코드

(b) 제어 에이전트 패킷 구조

그림 7. 초기연결설정 절차 및 패킷 구조

연결 설정 시에 교환되는 데이터의 크기는 아주 작기 때문에 교환되는 Packet의 수는 한번에 하나씩 교환하게 된다. 그림 8은 제어 패킷을 이용하여 ATM Cell로 전송하는 과정을 보이고 있다. 수신 하고자 하는 제어 패킷은 AAL5의 CPCS-PDU (Common Part Convergence Sub Layer - Protocol Data Unit)의 Payload 필드에 적재되어 SAR (Segmentation and Reassembly) 부 계층으로 전송된다[3].

SAR 부 계층에서 CPCS-PDU는 ATM Cell의 Payload 사이즈인 48 Octets의 SAR-PDU로 분할되어 ATM 계층으로 전송되고 각각의 SAR-PDU에

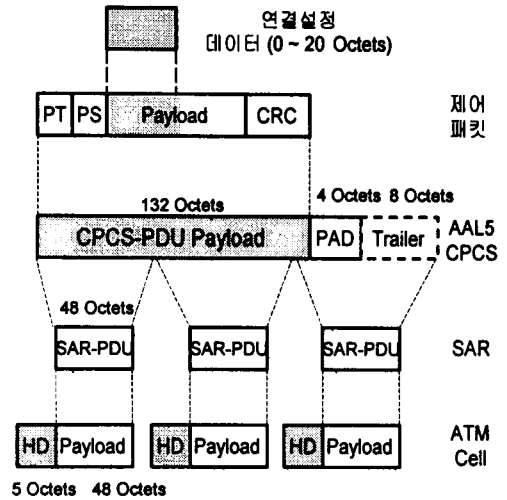


그림 8. ATM 통신망에서의 제어 패킷 전송

ATM Cell Header를 추가하여 물리 계층의 전송 Frame에 실어 전송하게 된다[3].

#### 4.2 서비스 에이전트의 기능 구현

서비스 에이전트는 원격접속 서비스 프로그램과 일대일 대응을 한다. 제어 에이전트에 의해 활성화된 서비스 에이전트는 초기에 Basic Query Connection만을 구성하고, 대역폭의 낭비를 최소화하기 위해 멀티미디어 채널은 구성하지 않도록 하였다. 그림 9는 서비스 에이전트의 구조를 보여준다.

Basic Query Connection 계층은 원격 의료 정보 데이터베이스를 검색하고 제어하는 역할을 수행하며, 주로 텍스트 형태의 데이터를 처리한다. 질의 (Query) Proxy는 원격검색 서비스 프로그램이 전송한 패킷에서 데이터베이스 검색에 필요한 SQL구문을 추출하여 질의 서버에 전송하는 역할을 수행한다. 결과(Result) Proxy는 질의 수행에 따른 결과를 처리하는데 여기에는 질의 결과의 조합뿐 아니라, SQL구문에러, 서버에러 등에 대한 처리도 담당한다.

멀티미디어 데이터 전송을 위한 연결 설정은 각 미디어의 전송 특성에 적합한 QoS를 설정하도록 미디어별 Thread를 각각 사용하도록 설계, 구현하였다 [12].

그림 10의 서비스 에이전트 패킷 구조에서 PT는 전송 데이터의 길이가 Payload의 크기보다 클 때 연

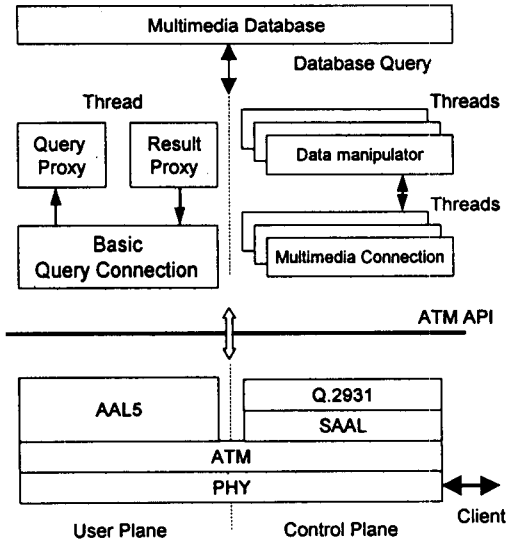


그림 9. 서비스 에이전트 구조

PT (1 Octet)	RT (1 Octet)	CC (2 Octets)	RC (2 Octets)	PS (2 Octets)	Payload (256 Octets)	CRC (4 Octets)
-----------------	-----------------	------------------	------------------	------------------	-------------------------	-------------------

PT : Packet Type  
 RT : Result Type  
 CC : Column Count  
 RC : Row Count  
 PS : Payload Size

- 서비스 패킷
  - PT : Payload에 저장된 정보 형태 (Start, Normal, End)
  - RT : 데이터베이스 쿼리 결과 형태
  - CC : 데이터베이스 검색 항목 수
  - RC : 데이터베이스 검색 결과 수
  - PS : Payload에 저장된 유호 데이터 크기
  - Payload : 쿼리 정보 및 결과
  - CRC : Application에서의 에러검출 코드

그림 10. 데이터 전송 패킷 구조

속 데이터를 표시하고, RT 필드는 전송 데이터가 결과 값인지, 에러에 관한 정보인지를 표시한다. CC와 RC는 각각 검색된 데이터 레코드의 행과 열의 숫자를 기록한다. PS 필드는 Payload 필드에 기록된 데이터의 크기를 기록한다. CRC는 전송 데이터의 신뢰도를 위한 어플리케이션에서의 32Bit 에러검출 기능이다.

서비스 에이전트와 원격 검색 프로그램간의 데이터 교환을 위한 전송 구조는 그림 11과 같다. AAL5를 이용한 전송 구조는 제어 에이전트와 동일하나

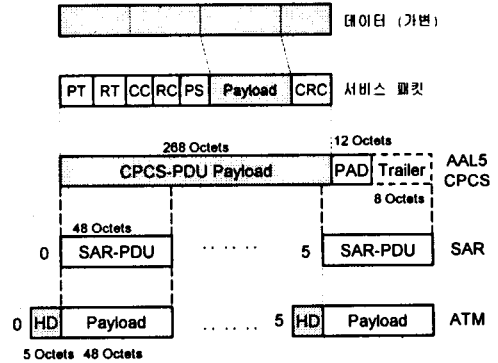


그림 11. ATM 통신망에서의 서비스 패킷 전송

데이터의 크기가 서비스 패킷의 크기를 넘을 수 있다는 것이 차이점이다.

DBMS 서버에 설치된 서비스 에이전트의 연결설정 단계는 그림 12와 같이 Basic Query Connection과 멀티미디어 데이터 채널 구성으로 구분된다. Client의 원격검색 프로그램은 작업이 종료할 시점에서 제어 에이전트와의 연결해제 절차를 통해 할당

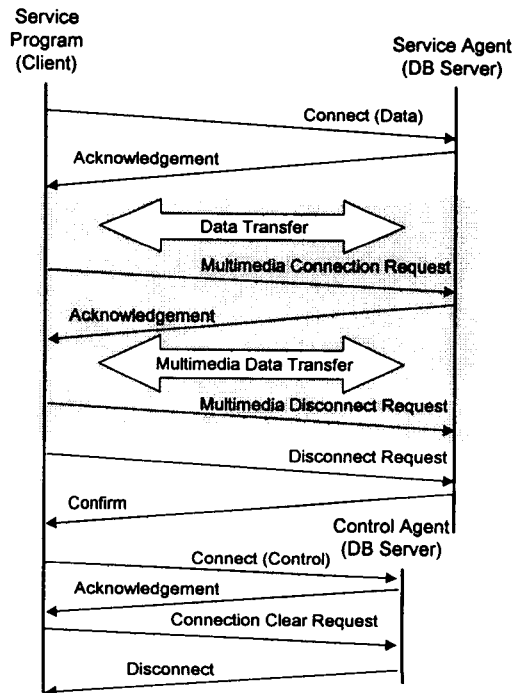


그림 12. 서비스 에이전트 프로토콜 구조

되었던 슬롯을 재사용을 위해 해제한다.

서비스 에이전트는 전송기능 외에 데이터베이스 검색 기능을 제공해야 한다. 원격 검색 프로그램으로부터 서비스 에이전트는 SQL형태로 데이터베이스 검색 요구를 받는다. 서비스 에이전트의 질의 Proxy는 전송 받은 SQL문장을 이용하여 데이터베이스에 질의를 수행하고 결과를 원격 검색 프로그램으로 전송해 준다. 질의 Proxy는 다음과 같이 3개의 영역으로 구성되어 있다 :

- 질의 수행부 : 전송 받은 SQL문장을 문장 형태 (select, delete, update, insert등)에 맞는 질의 수행
- 결과 전송부 : 질의 수행부의 수행 결과를 전송 형태에 맞게 재구성하고, 전송
- 에러 처리부 : 잘못된 SQL문장에 의한 에러 메시지를 원격 검색 프로그램으로 전송

그림 13은 질의 Proxy가 SQL문장을 받아서 처리하는 과정을 보여준다. 질의 Proxy의 각 영역은 Illustra DBMS에서 제공해주는 API를 이용하여 구성하였다[8]. 만약 다른 DBMS를 이용하게 될 경우 질의 Proxy를 해당 DBMS의 API를 이용하여 재구성하여야 한다. 질의 Proxy를 재구성하더라도 원격 검색 프로그램은 변경 없이 사용이 가능하므로 원격

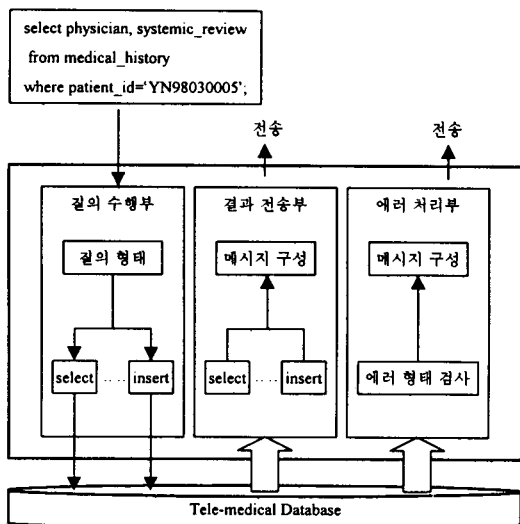


그림 13. 질의 Proxy

검색 프로그램의 DBMS 독립성을 보장한다. DBMS 독립성은 하나의 원격 검색 프로그램이 다종의 DBMS에 접근할 수 있음을 의미한다.

### 4.3 원격 검색 프로그램의 에이전트 API

원격 검색 프로그램이 에이전트를 통하여 검색을 수행하기 위해서는 DBMS Agent API를 이용하는 데 본 논문에서 구현한 API는 표 1과 같다.

표 1. DBMS 에이전트 API

DBMS Agent API	기능
ATM_Setup()	제어 에이전트와 초기연결 설정 및 Basic Query Connection 설정
ATM_Close()	Basic Query Connection 종료
ATM_SetupT()	멀티미디어 데이터 전송을 위한 미디어별 연결설정
ATM_CloseT()	미디어별 연결설정 종료
DoCommunication()	데이터베이스 질의 전송 및 질의 결과 수신
DoTransfer()	멀티미디어 데이터 송수신

ATM\_Setup()은 NULL Parameter를 가지며 ATM Descriptor를 리턴 한다. 리턴된 Descriptor는 Basic Query Connection Descriptor로 연결설정이 ATM\_Close()에 의하여 종료 될 때까지 사용된다. ATM\_Close()는 Basic Query Connection Descriptor를 Parameter로 갖는다. ATM\_SetupT()는 연결하고자 하는 미디어의 형태 지시자를 Parameter로 가지며 미디어별 QoS를 만족하는 Descriptor를 리턴 한다. ATM\_CloseT()에 미디어별 Descriptor를 Parameter로 넘겨줌으로써 미디어 연결 설정이 종료하게 된다.

DoCommunication()은 Basic Query Connection Descriptor와 SQL 문장을 Parameter로 사용하고 질의 결과를 Character Pointer로 리턴 한다. 원격 검색 프로그램은 이 Pointer를 Parsing함으로써 원하는 결과를 얻을 수 있다. DoTransfer()는 미디어별 Descriptor와 미디어의 전송방향 지시자를 기본 Parameter로 하여 전송일 경우에는 데이터의 시작 포인터를 수신일 경우는 NULL 포인터를 Parameter

로 넘겨준다. 각 API들은 C와 Fore사의 ATM XTI API를 이용하여 구현하였다.

### 5. 성능 분석

DBMS 에이전트를 통한 원격 DB 접속 기능의 구현은 DBMS 독립성과 ATM 통신망의 서비스를 수용할 수 있다는 장점이 있지만, 연결 설정에 따른 시간적인 오버헤드와 전송 패킷에 따른 데이터 오버헤드가 발생된다. 본 절에서는 Native ATM API를 이용하는데 따르는 오버헤드를 측정하여 성능 평가를 한다.

에이전트의 전송 패킷으로 인한 데이터 오버헤드는 256 Bytes에 대하여 12 Bytes가 발생한다. 1 Mbytes의 데이터를 기준으로 하여 데이터 오버헤드를 계산하면 다음과 같다 :

- 전체 전송 데이터 =  $(1024000 / 256) \times 268 = 1072000$  Bytes
- 전송 오버헤드 =  $(1024000 / 256) \times 12 = 48000$  Bytes
- 오버헤드율 =  $(48000 / 1072000) \times 100 \approx 4.5 \%$

이 분석에서 살펴본 바와 같이 전송 패킷의 오버헤드율이 5% 아래로 유지되고 있으므로 에이전트 전송 패킷으로 인한 데이터의 증가는 요구 QoS가 만족된다면 수용 가능하다고 본다. 하지만 데이터의 사이즈가 수십 메가바이트 단위로 증가하게 되면 전체적으로는 비율과는 상관없이 상대적으로 큰 오버헤드 데이터가 전송되게 되어 대역폭을 낭비하게 된다.

에이전트를 사용함으로써 인한 시간적 오버헤드를 보면, 그림 14에서 보는바와 같이 초기 연결 설정시 QoS 설정 지연시간, 서비스 에이전트의 활성화 시간, Basic Query Connection 구성 지연시간, 멀티미디어 채널구성 지연시간 등이 있다.

그림 14는 상용 DBMS API를 사용했을 때와 본 논문에서 제안한 DBMS 에이전트 API를 사용했을 때의 지연시간을 비교하여 도식화한 것이다. 상용 DBMS 제공 API를 사용하였을 경우 API가 UDP를 지원한다면 연결설정에 소요되는 오버헤드가 없다고 볼 수 있으며, 만약 TCP를 사용하는 API의 경우 초기연결 지연시간이 존재한다. 에이전트를 사용한 접속 구현에서는 연결 설정시 QoS 설정으로 인해

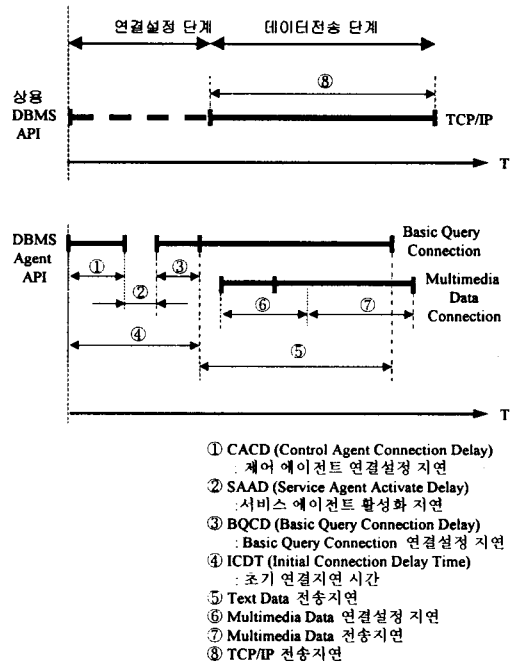


그림 14. 연결설정 지연시간

지연이 발생하나, 그림 14에서와 같이 초기 연결지연은 원격접속 서비스 프로그램이 활성화될 시점의 지연이므로 실제 데이터 전송에는 영향을 주지 않는다.

#### 5.1 초기 연결설정 지연시간

그림 15는 초기 연결설정 지연시간을 측정할 결과이다. 가로축이 반복 회수를 의미하고 세로축이 연결설정에 소요된 시간을 ms 단위로 측정할 값이다. 초기 연결설정 지연시간은 그림 14의 ICDT(Initial Connection Delay Time)에서 BQCD(Basic Query Connection Delay)를 뺀 시간으로 이의 수식 표현은 다음과 같다 :

$$t_{fdt} = t_{ICDT} - t_{BQCD} = t_{CACD} + t_{SAAD}$$

$t_{fdt}$  : 초기 연결설정 지연시간

$t_{ICDT}$  : 초기 연결 지연시간

$t_{BQCD}$  : Basic Query Connection 지연시간

$t_{CACD}$  : 제어 에이전트 연결 지연시간

$t_{SAAD}$  : 서비스 에이전트 활성화 시간

초기 연결 지연시간을 210회 반복 측정할 결과 125 Kbps의 QoS로 연결 설정을 수행한 경우 평균 70 ms (표준편차 : 21.4 ms) 가 소요되었다.



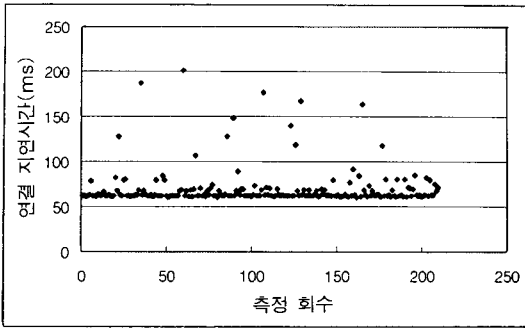


그림 15. 초기 연결설정 지연시간

### 5.2 Basic Query Connection 연결설정 지연시간

문자정보의 전송을 위한 Basic Query Connection 은 멀티미디어 Connection에 비하여 상대적으로 낮은 전송 대역폭을 필요로 한다. 그림 16은 서비스 에이전트에서의 Basic Query Connection 구성 시간을 측정할 것이다. 가로축은 측정 반복 회수를 나타내고 세로축은 연결 설정에 소요된 시간을 ms 단위로 나타낸 것이다. Basic Query Connection 연결설정 지연시간은 그림 14의 BQCD에 해당하는 시간으로 서비스 에이전트가 활성화된 후 실질적인 질의와 질의 결과를 전송하기 위한 연결 설정시간에 해당한다.

Basic Query Connection에 소요된 시간을 210회 반복 측정한 결과 512 Kbps의 QoS로 연결 설정한 경우 평균 37.8 ms (표준편차 : 10.6 ms)가 소요되었다.

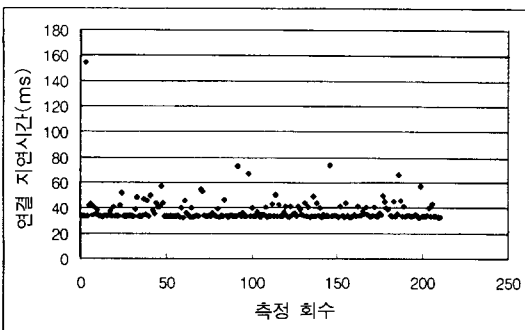


그림 16. Basic Query Connection 지연시간

### 5.3 멀티미디어 데이터 전송시간

크기가 큰 멀티미디어 데이터를 Native ATM Service를 이용하여 전송할 경우 얻을 수 있는 장점

은 미디어별 QoS를 만족하면서 고속의 전송이 가능한 것이다. 에이전트 이용으로 인해 발생하는 연결설정 지연이나 전송 패킷 오버헤드를 전송 속도로써 극복할 수 있을 것이다.

멀티미디어 데이터의 전송 속도를 측정하기 위하여 65,601,530 Bytes의 내시경 클리동영상을 사용하였다. 20 Mbps의 QoS를 비디오 데이터에 설정하고 200회의 전송시간을 반복 측정하여 그림 17에 그래프로 표현하였다. 평균 전송시간은 28.6 s (표준편차 : 0.53 s)가 소요되었다.

데이터 전송시 전송 패킷의 오버헤드가 약 4.5%인 것을 감안하여 전송률을 다음과 같이 구할 수 있다. 계산 결과에 따라 평균 전송률이 약 18.8 Mbps로 설정된 QoS를 만족하고 있음을 알 수 있다.

$$\begin{aligned} \text{오버헤드} &= (65,601,530/256) \times 12 \approx 3,075,072 \text{ Bytes} \\ \text{총 전송량} &= 65,601,530 + 3,075,072 = 68,676,602 \text{ Bytes} \\ \text{평균 전송률} &= (68,676,602/28.6) \times 8 \approx 19,210,238 \text{ bps} \end{aligned}$$

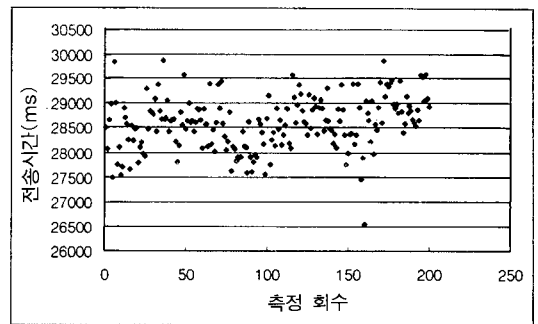


그림 17. 멀티미디어 데이터 전송시간

### 5.4 에러검출 방식에 대한 고찰

상기의 측정결과는 전송 Error Control 기능을 제거한 상태에서의 결과이다. CRC32 Error Control 방식으로[13,14] 사용하였을 경우에는 전체 전송률이 1/3 수준으로 떨어졌다. 이는 CRC32 연산에 따른 CPU의 부하 증가 및 Error Control을 위한 Acknowledge 신호의 전송에 의한 지연에 따른 것으로 판단된다. 이와 같은 결과로 Error Control 기능이 포함된 원격 검색 에이전트를 이용할 경우 CRC32의 고속 처리능력을 보유한 컴퓨터가 필요함을 알 수 있다.

## 6. 결 론

본 연구에서는 멀티미디어 원격 의료 정보 시스템을 위한 의료 정보 데이터베이스 원격 접속을 위하여 Native ATM API를 이용한 원격 DB 접속 에이전트를 구현하였다. 본 논문에서 제안한 멀티미디어 DBMS 원격 접속 기능은 Native ATM Service를 사용함으로써 각 미디어별 전송 특성에 맞는 QoS를 설정할 수 있게 되었으며, 원격 접속 에이전트를 이용한 원격 접속기능은 DBMS에 독립적인 구조로 설계되어 SQL 기반의 여러 DBMS에 능동적으로 대응할 수 있다는 장점을 가진다. DBMS의 독립성은 멀티미디어 원격 의료 정보 시스템이 병원에 설치될 경우 서로 다른 DBMS를 사용하는 병원마다의 멀티미디어 의료 정보 데이터베이스를 하나의 원격 검색 프로그램으로 접속할 수 있음을 의미한다.

성능 분석 결과 원격 DB 검색 에이전트를 이용함으로써 발생하는 오버헤드는 ATM 통신망 이용에 의한 전송속도의 증가와 QoS에 의해 극복이 됨을 알 수 있다. 이는 Native ATM Service의 QoS설정을 위한 연결설정 지연이 TCP/IP 통신망을 사용하는 것 보다 많이 소요되고 있으나 이는 멀티미디어 원격 의료 정보 시스템이 시작할 시점의 지연이며 실질적인 성능 기준이 되는 것은 설정된 QoS의 보장이라 할 수 있기 때문이다.

본 연구에서 제안하고 구현한 DBMS 에이전트 API를 이용한 멀티미디어 의료 정보 데이터베이스 원격 검색 기능은 멀티미디어 데이터별 QoS를 보장해 주면서 특정 DBMS에 의존하지 않고 하나의 원격 DB 접속 기능으로 이 기종 DBMS에 접근할 수 있도록 하는 장점을 지니며 다수의 멀티미디어 의료 정보 데이터베이스 시스템 구축에 효과적이라 할 수 있다. 또한 향후 상용 DBMS가 Native ATM API를 지원하더라도 제안된 DBMS 에이전트 기능은 특정 DBMS에 의존하지 않는다는 장점을 가진다.

## 참 고 문 헌

[1] 영남대학교 정보통신연구소, "멀티미디어 원격

의료 정보 시스템 개발 (1차 년도 보고서)", 정보통신부, 1996.

[2] 영남대학교 정보통신연구소, "멀티미디어 원격 의료 정보 시스템 개발 (2차 년도 보고서)", 정보통신부, 1997.

[3] 김영탁, "광대역 종합 정보 통신망(B-ISDN)", 한국정보문화센터 정보기술교육원 교재, 1997.

[4] Informix, "Illustra User's Guide Release 3.2", Illustra Information Technologies, 1995.

[5] FORE Systems, "ForeRunner ASX-200 ATM Switch User's Manual", FORE Systems, Inc. 1995.

[6] FORE Systems, "ForeRunner SBA-200 ATM SBus Adaptor User's Manual", FORE Systems, Inc., 1996.

[7] FORE Systems, "ForeRunnerLE ATM PCI Bus PC Adaptors Installation Guide", FORE Systems, Inc., 1996.

[8] Informix, "Application Programming Interface Guide. Release 3.2", Illustra Information Technologies, 1995.

[9] ATM Forum, "Native ATM Services : Semantic Description Ver 1.0", ATM Forum, 1996.

[10] W. Richard Stevens, "UNIX Network Programming", Prentice Hall, 1994.

[11] David E. McDysan, Darren L. Spohn, "ATM Theory and Application", Mc Graw-Hill, 1995.

[12] Charles J. Northrup, "Programming With UNIX Threads", John Wiley & Sons, Inc. 1996.

[13] Carl Symborski, "AAL5 CRC32 Sample Code", <http://cell-relay.indiana.edu/cell-relay/FAQ/ATM-FAQ/f/f3.htm>

[14] Richard Black, "Fast CRC32 in software", <http://www.cl.cam.ac.uk/Research/SRG/bluebook/21/crc/crc.html>



김 호 철

1989년 2월 경북대학교 공과대학  
전자공학과(전산전공) 졸  
업(공학사)

1989년 3월~1995년 8월 삼성전  
관 종합연구소 CAD실 근  
무(대리)

1995년 9월~1996년 4월 삼성상  
용차 기술지원팀 CAD 파트 근무

(대리).

1997년 3월~현재 영남대학교 공과대학 대학원 멀티미  
디어통신공학과 재학.중

관심분야는 ATM/B-ISDN 기반의 초고속 정보통신망,  
TINA 기반의 통신망 운용관리.



김 영 탁

1984년 2월 영남대학교 공과대학  
전자공학과 졸업(공학사)

1986년 2월 한국과학 기술원 (KA  
IST) 전기 및 전자공학과  
졸업(공학석사)

1990년 3월~1994년 8월 한국통  
신 통신망연구소 전송망  
구조연구실장

1994년 9월~현재 영남대학교 공과대학 정보통신공학과  
조교수. 관심분야는 ATM/B-ISDN 기반의 초고  
속 정보통신망, GI(Global Information Infra-  
structure), 차세대 인터넷(NGI), TMN/TINA  
체계의 통신망 운용관리.