

유무선 통합 다중 응급 원격
진료시스템 설계

연세대학교 대학원
생체공학협동과정
전기전자공학전공
강 호 현

유무선 통합 다중 응급 원격
진료시스템 설계

지도 유 선 국 교수

이 논문을 석사 학위논문으로 제출함

2005년 1월 일

연세대학교 대학원
생체공학협동과정
전기전자공학전공
강 호 현

강호현의 석사 학위논문을 인준함

심사위원 _____ 인

심사위원 _____ 인

심사위원 _____ 인

연세대학교 대학원

2005년 1월 일

감사의 글

연구실에 들어 온지도 어느덧 2년이라는 시간이 흘렀습니다. 생각했던 것보다 시간이 빨리 지난 듯 합니다. 앞에서 끌어 주고 뒤에서 밀어주신 좋은 선후배 여러 분과 함께 할 수 있어서 힘든 고비에서도 꺾이지 않았다고 생각합니다. 특히 부족한 저를 깊은 관심을 가지시고 열정으로 가르쳐주신 유선국 교수님께 감사를 드립니다. 또한 항상 배려 속에서 밝은 미소로 인사를 받아주신 김남현 교수님께도 감사의 마음을 전합니다.

힘든 일을 묵묵히 하시면서 저희에게 싫은 소리 한번 안하신 착한 동근이형, 그리고 책상 앞에 수북이 논문을 쌓아가면서 열심히 연구한 기원이형(아마 몇 년 뒤엔 엄청난 지식의 소유자가 되실 듯)께 고마운 마음을 전합니다. 연구실에 유일한 동갑내기 친구 민규(‘인생 역전’ 하려면 장가 가기 전에 꼭 허리 나아라), 키스트로 출근한 후로 자주 보진 못했지만 착한 재복과 덕찬, 느끼하다고 항상 놀렸지만 알고 보면 순수함이 철철 넘치는 우리 국진이, 그리고 몸에 비해 상당히 민첩한 정진이, 옆집 아저씨와 같은 이미지를 가졌지만 풋풋한 충기, “항상 밝고 착하고 아름답고 특히나 나에게 친절했던 수정양.이라고 써주삼”이라고 써달라는 수정이, 오랫동안 생활을 같이 하진 못했지만 범상치 않은 영재와 정채, 그리고 충기가 인정한 똑똑한 윤정이, 내 옆에서 항상 농구멤버가 되어준 천사표 남자친구 동헌이, 이 모든 분들께 진심으로 감사한 마음을 전하고 싶습니다. 그리고 역시 군대에서도 동기가 남다른 것처럼 하봉이에게는 특별히 감사의 표현을 남기고 싶습니다. 같이 생활하면서 정도 많이 들었는데 졸업해서는 몸 건강해라.

마지막으로, 항상 저를 믿어 주시고 옆에서 격려해 주신 사랑하는 부모님, 옆에서 가끔 용돈을 주는 착한 누나들께 고개 숙여 감사의 마음을 전합니다. 그리고 변변찮은 삼촌을 뒤편 심기가 불편하신 혜원이, 승훈이, 민기, 성민이 조카분들께도 감사드립니다. 그럼 모두 항상 건강하시길 바랍니다.

2006년 1월

강 호 현 드림.

차 례

그림 차례	iv
표차례	ix
국문요약	x
제1장. 서론	1
제2장. 멀티미디어 데이터 처리기술	5
2.1. MPEG-4	5
2.1.1 MPEG-4의 개념	5
2.2. H.261	5
2.2.1 H.261의 개념	5
2.3 H.263	7
2.3.1 H.263의 개념 및 표준화	7
2.4 DICOM	8
2.4.1 DICOM의 개요	8
2.4.2 KDicom	8
2.5 DirectShow	12
2.5.1 DirectShow의 개요	12
2.5.2 DirectShow의 필터 연결	14
제3장 네트워크 통신 프로토콜	20
3.1 TCP/IP	20
3.1.1 TCP/IP 프로토콜 구조	33
3.1.2 IP주소, 포트 번호	22
3.1.3 클라이언트/서버 모델	24
3.2 TCP	25
3.2.1 TCP/IP 서버/클라이언트 구조	25
3.2.2 동작원리	26
3.3 UDP	28
3.3.1 UDP 서버/클라이언트 구조	28
3.3.2 동작 원리	29
제4장 유무선 통신 다중 기반 멀티미디어 처리 기술	31
4.1 Filter 구현	31
4.1.1 Tee Multi Render Filter	31
4.1.2 Tee Multi Source Filter	32

4.2 Setting	34
4.2.1 Setting	34
4.3 고품질 영상	39
4.3.1 구성	39
4.3.2 송신단	39
4.3.3 수신단	42
4.4 화상회의 음성 및 영상	43
4.4.1 구성	43
4.4.2 송신단	44
4.4.3 수신단	46
4.5 응급 환자의 생체 신호	47
4.5.1 생체 신호 데이터 처리 기술	47
4.5.2 효율적인 전송을 위한 생체신호 압축 기술	47
4.5.3 생체 신호 디스플레이	49
4.6 응급 환자 이미지 파일	51
4.6.1 구성	51
4.6.2 원격 이미지 파일 포인팅	52
제5장 유무선 통신 다중 기반 통신 프로토콜	55
5.1 TCP 전송	55
5.1.1 TCP 소켓 구현에서 고려해야할 문제	55
5.1.2 문제점을 해결한 TCP/IP 송수신 메커니즘	57
5.2 UDP 전송	60
5.2.1 UDP 소켓 구현에서 고려해야할 문제	60
5.2.2 문제점을 해결한 UDP 송수신 메커니즘	61
제6장 전체 시스템 구성	63
6.1 환자측 시스템 및 모니터링 시스템	64
6.1.1 병원 내부망의 환자 시스템 및 모니터링 시스템	65
6.1.2 병원 외부 유선망의 환자 시스템	66
6.2. 제어 서버	67
6.2.1 1단계 제어 서버	68
6.2.2 2단계 제어 서버	71
6.2.3 3단계 제어 서버	73
6.2.4 제어 서버 시스템 구성도	74
6.2.5 제어 서버의 NAT	78

제7장 실험 환경 및 결과	81
7.1 환자측 시스템의 어플리케이션 인터페이스	86
7.2 의사측 시스템 및 스트림서버의 어플리케이션 인터페이스	87
7.3 제어 서버	90
7.4 환자측 무선망(EVDO In Car), 의사측 유선망(병원망, ADSL), 스트림서버 유선망(병원망)의 환경에서의 실험	91
7.5 환자측 유선망(ADSL), 의사측 무선망(EVDO), 스트림 서버 유선망(병원망)의 환경에서의 실험	93
7.6 환자측 유선망(병원망), 의사측 유선망(병원망), 스트림 서버 유선망(병원망)의 환경에서의 실험	95
제8장 결론 및 토의	96
참고문헌	98
Abstract	100

그림 차례

그림 2-1 KDICOM에서의 데이터 모듈	9
그림 2-2 KDICOM에서의 이미지 모듈	9
그림 2-3 KDICOM에서의 오버레이 기능	10
그림 2-4 KDICOM에서의 히스토그램을 이용한 윈도우 조정	11
그림 2-5 KDICOM에서의 네트워크 모듈	11
그림 2-6 DirectShow의 필터 연결에 필요한 미디어형을 열거하는 첫 번째 과정	14
그림 2-7 DirectShow의 필터 연결에 필요한 미디어형을 열거하는 두 번째 과정	15
그림 2-8 DirectShow의 필터 연결을 시도하는 첫 번째 단계	16
그림 2-9 DirectShow의 버퍼 협상 과정중 출력핀에서 버퍼협상이 시작	17
그림 2-10 DirectShow의 버퍼 협상의 첫 번째 단계	18
그림 2-11 DirectShow의 버퍼 협상의 두 번째 단계	18
그림 3-1 인터넷의 구성도	20
그림 3-2 TCP/IP 프로토콜을 이용한 통신	21
그림 3-3 TCP/IP 프로토콜 구조	21
그림 3-4 TCP/IP 통신에서의 IP 주소와 포트 번호의 역할	23
그림 3-5 TCP 연결 지향형 소켓 시스템 호출	26
그림 3-6 TCP의 동작원리	27
그림 3-7 TCP의 서버/클라이언트 통신 형태	27
그림 3-8 UDP 비연결 지향형 소켓 시스템 호출	29
그림 3-9 UDP의 동작원리	30
그림 4-1 'Tee Multi Render Filter'필터의 개념도	31
그림 4-2 'ITransFilter'인터페이스의 순수가상함수	32
그림 4-3 'Tee Multi Render Filter'필터를 사용한 필터그래프 구성	32
그림 4-4 'Tee Multi Render Filter'필터의 데이터 종류에 대한 헤더구성	32
그림 4-5 'Tee Multi Source Filter'필터의 개념도	33

그림 4-6 ‘Tee Multi Source Filter’필터를 사용한 필터그래프 구성	33
그림 4-7 수신단에서 수신된 데이터 타입에 따른 함수호출의 예	33
그림 4-8 환자측 시스템의 Setting Dialog Interface	34
그림 4-9 환자측 시스템의 Center Information Dialog	36
그림 4-10 Setting Dialog로부터 레지스트리에 저장된 환자측 정보	38
그림 4-11 고화질 카메라 캐논 VC-C4의 사양	40
그림 4-12 환자측 시스템에서의 고화질 카메라와 Analog to Digital 컨버터 연결	40
그림 4-13 송신단에서의 고화질 비디오 필터 구성	41
그림 4-14 ‘FrameRate Control’필터의 인터페이스	42
그림 4-15 수신단에서의 고화질 비디오 필터 구성	42
그림 4-16 송신단으로부터 전송받은 MediaType과 Allocator 정보 셋팅	42
그림 4-17 고화질 데이터 전송 메커니즘의 시퀀스 다이어그램	43
그림 4-18 응급 진료용 화상회의 영상/음성 송수신 구조	44
그림 4-19 송신단 화상회의 데이터 흐름도	44
그림 4-20 송신단 화상회의 데이터 버퍼 컨트롤	45
그림 4-21 송신단의 화상회의 영상 데이터 필터그래프 구성	45
그림 4-22 송신단의 화상회의 음성 데이터 필터그래프 구성	46
그림 4-23 수신단의 화상회의 영상 데이터 필터그래프 구성	46
그림 4-24 수신단의 화상회의 음성 데이터 필터그래프 구성	46
그림 4-25 PC에 입력된 생체신호의 data 구조	47
그림 4-26 생체신호 데이터 압축/복원을 위한 전체 구조도	47
그림 4-27 생체 신호 데이터의 분류 및 표현	49
그림 4-28 생체 신호 표시하기 위한 MemDC를 이용한 화면 표현 구조	50
그림 4-29 생체신호 전송 메커니즘	50
그림 4-30 시스템내에 전송된 파일의 링크된 라이브러리	52
그림 4-31 환자 이미지 전송 시퀀스 다이어그램(DICOM)	52
그림 4-32 송신단에서 마우스 포인터 정보를 전송하기 위한 코드	53
그림 4-33 수신단에서 수신한 마우스 포인터정보를 매핑시키는 코드	53
그림 4-34 시스템내에서 전송되어지는 파일 메시지 정의	54

그림 5-1 TCP 패킷 전송과 수신 상태도의 예	55
그림 5-2 TCP 패킷 전송과 수신 상태 세부도	56
그림 5-3 헤더 정보를 추가한 TCP 패킷	57
그림 5-4 TCP에서 하나의 패킷을 전송하기 위한 코드 구현	58
그림 5-5 TCP에서 패킷을 수신하기 위한 코드 구현	59
그림 5-6 UDP 패킷을 리포매팅 하기 이전의 패킷	60
그림 5-7 UDP 패킷을 리포매팅한 후의 패킷	60
그림 5-8 리포매팅 된 후 데이터 프래그먼트 사이즈로 프래그먼트 하는 과정	61
그림 5-9 시퀀스 넘버가 첨가된 프래그먼트 데이터	61
그림 5-10 UDP프로토콜을 사용한 고용량데이터 전송을 위한 메커니즘	62
그림 6-1 전체 시스템 구성도	63
그림 6-2 다중 커넥션을 위한 환자측 접속 인터페이스	64
그림 6-3 병원망의 클라이언트 접속 및 데이터전송	65
그림 6-4 병원 외부 유선망의 환자 시스템 및 모니터링 시스템(의사, 스트림 서버)	66
그림 6-5 제어 서버를 사용한 다중 응급의료 시스템 기본 구조	68
그림 6-6 M-to-N 사용자의 독립적인 연결	68
그림 6-7 1단계 제어 서버 연결 흐름도	69
그림 6-8 1단계 제어 서버에서 사용자의 전환이 필요한 경우의 흐름도	70
그림 6-9 M-to-M 사용자의 다중 연결	71
그림 6-10 2단계 제어 서버의 다중 연결 처리 흐름도	71
그림 6-11 M-to-N 사용자의 독립연결과 다중 연결의 혼합	73
그림 6-12 3단계 제어 서버 흐름	74
그림 6-13 1단계 직접 연결 시퀀스 다이어그램	75
그림 6-14 2단계 다중 연결 시퀀스 다이어그램	76
그림 6-15 3단계 통합 시스템 제어 서버 시퀀스 다이어그램	77
그림 6-16 병원망의 NAT문제를 해결하기위한 터널링 방법	79
그림 6-17 제어 서버에 셋팅된 외부 IP 와 내부 IP	79
그림 6-18 외부 IP와 내부 IP를 셋팅하는 제어 서버 애플리케이션	80

그림 7-1 유선망(환자) to 유선망(모니터링) 시스템 구성도	82
그림 7-2 무선망(환자) to 유선망(모니터링) 시스템 구성도	83
그림 7-3 유선망(환자) to 무선망(의사) to 유선망(스트림 서버) 시스템 구성도	84
그림 7-4 무선망(환자) to 무선망(의사) to 유선망(스트림 서버) 시스템 구성도	85
그림 7-5 환자측 시스템의 애플리케이션	86
그림 7-6 모니터링 시스템의 환자 IP 리스트 선택 과정	87
그림 7-7 모니터링 시스템(의사, 스트림 서버)의 애플리케이션	88
그림 7-8 ECG 확장 디스플레이 컴포넌트	88
그림 7-9 스트림 서버, 환자측, 의사측 시스템 파일 다이얼로그의 DICOM 뷰어 컴포넌트	89
그림 7-10 제어 서버의 애플리케이션 인터페이스	90
그림 7-11 EVDO망을 통한 환자측 생체신호 및 고화질 영상 전송	91
그림 7-12 차안에서 EVDO 통신을 통한 환자측 생체 신호 및 환자이미지 전송	91
그림 7-13. 무선망의 환자측 시스템이 전송한 데이터를 확인하는 모니터링 시스템	
(a) 의사측 병원망 : 병원내에 있는 의사측 시스템 (LAN)	
(b) 의사측 병원외부망 : 가정등의 인터넷 상용망에 연결된 의사측 시스템(ADSL)	92
(c) 스트림 서버 병원망 : 병원 응급실에 위치하면서 여러 환자측 시스템을 모니터링(LAN)	
그림 7-14 인터넷 상용망(ADSL)을 사용한 환자측 시스템	93
그림 7-15 무선망(EVDO)의 의사측 시스템과 병원망내의 스트림 서버	
(a) 의사측 무선망 : 제어서버로부터 환자의 고화질 비디오 영상 및 생체신호수신	
(b) 의사측 무선망 : 제어서버로부터 환자의 이미지 데이터를 수신	94
(c) 스트림 서버 병원망 : 제어서버로부터 병원 LAN을 통해 환자의 데이터 수신	
그림 7-16 병원망 내의 환자측 시스템	95

그림 7-17 병원망 내의 의사측 시스템 과 스트림 서버

(a) 의사측 병원망 : 환자측 시스템으로부터 직접 데이터를 수신하는 병원망내의 의사측 시스템

95

(b) 스트림 서버 병원망 : 환자측 시스템으로부터 직접 데이터를 수신하는 병원망내의 스트림 서버

표 차 례

표 2-1 DirectShow에서의 멀티미디어 데이터 처리 특징	12
표 2-2 DirectShow에서의 멀티미디어 데이터 처리 방안	13
표 3-1 TCP와 UDP 프로토콜의 특징	22
표 3-2 네트워크 통신에 필요한 포트번호 분류	23
표 4-1 레지스트리 키의 다섯 가지 구성 요소	37
표 4-2 고화질 카메라의 프레임당 Raw 데이터크기	40
표 4-3 MPEG-4 계열의 코덱에 따른 압축률(720*480)	41
표 4-4 Data 종류에 따른 접속 방식, 데이터 양, 우선순위	44
표 6-1 각 전문의에 따른 우선순위 전송 테이블	73
표 7-1 유무선 통합 M-to-M 응급 원격 진료시스템에서의 멀티미디어 전송 네트워크 프로토콜	81
표 7-2 외부통신망에 따른 Residential network의 전송 속 도	83

국 문 요 약

유무선 통합 다중 응급 원격 진료시스템 설계

컴퓨팅 성능과 네트워크 대역폭의 비약적인 발전은 의료계에서도 원격진료와 같은 고품질의 의료서비스를 제공하게 되었다. 원격진료(Telemedicine)는 환자와 의사가 직접 대면하여 행해지는 기존의 의료서비스와는 달리, 유무선 네트워크 환경을 통해 원격자문(Tele-Consultation), 원격진단(Tele-Diagnosis)의 의료체계를 구축하여 원거리에 있는 환자나 병원으로 이송중인 환자에게 양질의 의료서비스를 제공한다[1]. 특히, 응급환자를 진단하는 인력 및 장비가 부족한 보건소, 교도소, 학교 등에서 응급환자가 발생했을 때, 원격진료 서비스를 통해 2차, 3차 의료기관의 응급의, 전문의로부터 원격 자문, 원격 진단을 받게 됨으로써 양질의 서비스를 받을 수 있다. 또한, 병원으로 이송되는 응급환자의 경우 실시간으로 환자 데이터를 빠르고 정확하게 전달하는 것이 환자의 생명에 큰 영향을 미치게 되기 때문에 원격 진료는 중요한 요소라고 할 수 있다.

응급원격진료서비스 연구는 이루어지는 장소에 따라 크게 의료 기관 간, 의료 기관과 가정 간으로 나누어 연구가 진행되어져 왔다[1]. 이러한 연구는 유선 네트워크 환경의 의존적인 응급원격진료서비스 시스템을 바탕으로, 전문의와 환자 간의 1 대 1 통신 방식을 채택함으로써 순차적인 서비스를 제공하고 있다. 그러나 IT기술의 발전과 무선 인터넷 및 통신 환경의 발달에 따라 의료 기관과 응급차간, 이동중인 전문의와 가정 간의 유무선 네트워크 환경 속에서의 실시간 원격 진료가 가능하게 되면서 응급원격진료서비스는 이질적인 네트워크 환경을 고려한 시스템 구성이 필요하게 되었다.

이에 따라 기존의 유선 네트워크 환경에 의존적인 응급원격진료서비스 시스템은 저 대역폭의 무선 네트워크 환경을 고려한 응급원격진료서비스 시스템을 재구성하여야 한다. 또한, 기존의 1 대 1 통신 방식은 응급원격진료서비스를 받고자 하는 환자의 연결 순서에 따라 순차적으로 원격진료가 수행되는 방식이므로, 응급

환자가 발생했을 경우 서비스를 제공하지 못하여, 환자의 상태를 악화시킬 수 있다. 그러므로 응급원격진료서비스 시스템은 다수의 환자와 다수의 전문의를 연결하여 환자를 선택적으로 서비스 하는 기능이 필요하다.

본 논문에서는 VDSL(Very High rate digital Subscriber Line), ADSL(Asymmetric Digital Subscriber Line), Cable Modem과 같은 유선 네트워크 환경을 가지는 수도권이나 각 지방의 보건소, 교도소등의 기관에서 응급 환자가 발생했을 때, 환자정보를 실시간으로 송신 하여, 2차, 3차 의료 기관의 응급 의가 환자정보를 바탕으로 원격자문을 서비스 할뿐만 아니라, 환자에게 적합한 전문의를 찾아 함께 응급원격진료서비스를 제공하는 시스템을 제안 한다 . 이때, 전문의의 네트워크 환경은 병원내의 LAN(Local Area Network)에 위치하거나, 혹은 가정에서 VDSL, ADSL등의 인터넷망을 사용하여 PC(Personal Computer)를 통해 진료할 수 있고, 이동 중에 CDMA 1x EV-DO(Code Division Multiple Access), Wireless Lan등의 무선 네트워크를 사용하여 원격진료 함으로써 기존의 응급원격진료서비스 시스템과 달리 이질적인 네트워크 환경에서도 고품질의 원격진료 서비스를 제공할 수 있게 된다. 본 연구에서는 혼재된 유무선 네트워크 환경에서 실시간으로 환자정보를 다수의 전문의에게 제공하는 시스템을 설계하여 응급원격진료서비스 시스템의 유용성을 실험하였다.

핵심 되는 말 : 원격 진료, 원격 진단, 원격 자문, 응급원격진료서비스 시스템, VDSL, ADSL, LAN, CDMA 1x EV-DO, Wireless Lan

제 1장 서론

컴퓨팅 성능과 네트워크 대역폭의 비약적인 발전은 의료계에서도 원격진료와 같은 고품질의 의료서비스를 제공하게 되었다. 원격진료(Telemedicine)는 환자와 의사가 직접 대면하여 행해지는 기존의 의료서비스와는 달리, 유무선망을 통해 원격자문(Tele-Consultation), 원격진단(Tele-Diagnosis)의 의료체계를 구축하여 원거리에 있는 환자나 병원으로 이송중인 환자에게 양질의 의료서비스를 제공한다. 특히, 병원으로 이송되는 응급환자의 경우 실시간으로 환자 데이터를 빠르고 정확하게 전달하는 것이 환자의 생명에 큰 영향을 미치게 되기 때문에 원격 진료는 중요한 이슈라고 할 수 있다. 이러한 원격진료 시스템은 여러 유무선 네트워크의 환경에 따라 환자의 멀티미디어 데이터(비디오, 오디오, 이미지, 생체신호)를 전송하기 위해 효과적인 프로토콜을 설계하고 동영상 압축 메커니즘에 따라 가공하여야 한다. 지금도 많은 연구기관에서는 화상, 동화상의 자료교환 등을 지원하는 멀티미디어와 고속정보망 구축에 대한 연구가 진행되고 있다[1-4]. 그러나 현재까지 제안된 원격 진료 시스템은 각각의 무선 통신망이나 유선 통신망만을 위한 통신 시스템을 구축하였으며, 또한 의사와 환자간의 1 대 1 통신 방식을 채택한 시스템 구조를 구성되었다[5-10]. 1 대 1 통신은 접속된 환자의 순차적인 서비스를 제공하기 때문에 응급환자가 발생했을 경우 서비스를 제공하지 못하여, 환자의 상태를 악화시킬 수 있다. 본 논문에서는 개발된 시스템은 유무선망을 통합한 다 대 다 통신 방식의 시스템을 구성하여 개발함으로써 다수의 환자에게 최상의 의료서비스를 제공한다.

개발된 유무선 통합 원격진료 시스템은 제어 서버, 스트림 서버, 환자측 시스템, 의사측 시스템의 4가지 호스트 및 서버로 구성하였다. 다 대 다 의료 서비스를 위해 환자와 의사의 세션을 관리하고 환자의 멀티미디어 데이터를 가공하거나 저장하는 제어 서버를 병원 내부에 위치함으로써 의사와의 통신을 최적화하였다. 또한 스트림 서버를 응급센터에 구축하여 제어 서버로부터 환자의 의료 정보 및 네

트위크 상태를 확인 및 모니터링 함으로써, 응급의로부터 환자의 상태를 컨설팅하는 서비스를 제공한다.

전체 시스템 구성 중 의사측 시스템과 스트림 서버, 제어 서버는 환자측 시스템으로부터 응급환자의 상태를 나타내는 고화질 영상, 생체신호, 환자 정지영상, 화상회의 데이터를 전송받는다. 이때, 환자측 네트워크 환경이 유선망인가(Wired), 무선망인가(Wireless), 유선망이라면 병원망의 노드인가, 아니면 인터넷망의 노드인가를 고려해야 한다. 병원망의 네트워크를 IP주소 및 보안 문제로 인해 NAT(Network Address Translation)을 사용하는 병원 및 의료원은 클라이언트/서버(Client/Server) 모델의 통신을 구현할 때 문제가 된다. 본 논문에서 개발된 시스템은 이러한 문제를 해결하기 위해 터널링 기법을 사용하였다.

본 논문에서는 무선망(CDMA 1x EVDO , Wireless LAN)과 유선망 (ADSL, VDSL, Cable Modem)의 혼재된 환경에서 환자의 멀티미디어 데이터를 전송하기 위해 다음과 같은 사항을 고려하여 개발하였다.

- ◇ Raw Data의 고화질 영상 데이터를 네트워크 환경에 맞게 인코딩 (MPEG-4)
- ◇ Raw Data의 화상회의 음성 및 영상 데이터를 네트워크 환경에 맞게 인코딩 (H.261, H.263 , G.711)
- ◇ DICOM, CT , MRI 등의 환자 정지 영상 데이터 처리 기술
- ◇ ECG, Resp, SpO2, NIBP의 환자 생체 신호 데이터 처리 기술
- ◇ 환자 데이터를 전송하기 위한 네트워크 프로토콜 설계
- ◇ 다수의 클라이언트와 통신하기 위한 소켓 통신 구현 (멀티플렉싱, 멀티프로세싱, 멀티스레딩)
- ◇ 유무선망과 NAT를 실행하는 병원 내부망과의 통신을 위한 터널링
- ◇ 전송된 환자 데이터의 저장 및 저장된 데이터의 동기적 모니터링
- ◇ 유무선 네트워크의 변화에 따른 인코딩 Factor의 협상
- ◇ 전송하는 환자 데이터에 대한 제어 서버의 분배 (ByPassing, Flowing)

특히, 통신망의 대역폭 변화는 전송되는 멀티미디어 데이터에 대한 변환 모듈

및 코덱을 필요로하게 되었다. 고화질의 영상데이터는 MPEG-2, MPEG-4 코덱을 통해 프레임율 및 압축율을 조절하고, ECG, SpO2, Resp, NIBP 등의 생체신호는 Wavelet 압축기법을 사용하여 통신망에 따라 효율적인 전송을 가능하도록 하였다. 환자에 대한 의료 정지 영상인 DICOM의 모듈을 탑재하여 PACS시스템과의 질의(Query), 검색(Retrieve)기능을 수행하고 JPEG2000을 통하여 압축된 영상을 획득할 수 있다.

전송되는 환자의 멀티미디어 데이터는 효율적인 네트워크 전송을 하기 위해 생체 신호 및 환자의 정지영상 데이터는 TCP(Transmission Control Protocol) 프로토콜을 사용하였으며, 고화질 영상 및 전문의들의 컨퍼런스 데이터는 실시간 전송에 적합한 UDP(User Datagram Protocol)프로토콜을 이용하였다. 이때 고화질 영상의 데이터는 UDP프로토콜을 사용하기 위해서 헤더에 대한 새로운 프로토콜을 적용하여야한다. 새로운 UDP프로토콜이 적용된 데이터그램은 전송되는 네트워크의 MTU 사이즈에 따라 프래그멘테이션되어 전송하고, 손실된 데이터그램에 대한 재전송 매커니즘을 통해 최적의 화질을 보장한다. 이 시스템의 구성은 환자측 시스템과 이동형 응급 센터에 상주하는 제어 서버 및 스트림 서버 그리고 의사측 시스템으로 구성되며, 모든 구성요소의 세션관리 즉 커넥션 관리 및 모니터링은 제어 서버가 자동적으로 관리하게 된다. 전송된 데이터는 스트림 서버를 통해 환자에 따라 저장되며, 저장된 데이터는 Play Back 애플리케이션을 통해 추후에 다시 확인할 수 있다.

환자측 시스템에서 스트림 서버나 의사측 시스템으로 데이터를 전송하는 방법은 제어 서버를 중심으로 크게 두 가지로 나눌 수 있다. 첫째, 환자 데이터를 직접 스트림 서버나 의사측 시스템으로 전송하는 'ByPassing' 기법이다. 또 다른 기법은 모든 환자의 데이터 스트림에 대한 Flow를 제어 서버가 관여하게 되는 'Flowing'기법이다.

NAT로 구성된 네트워크를 가진 병원망과 서로 상이한 유무선망이 세션을 맺기 위해서는 터널링 기법이 필요하다. NAT는 네트워크 계층에서 사설 IP 주소를 공인 IP 주소로 변환하는데 사용하는 통신망의 주소 변환기를 의미하는데 일반적 통신 프로그래밍에 있어서 IP 변환에 따른 문제점을 가지고 있다. 이에 응급의료

정보 시스템 개발 센터는 제어 서버에서 NIC를 통해 전송된 데이터에 대한 터널링 기법을 사용하였다. 즉, 병원 내부망과 상용망을 위해 IP를 설정하고, 이를 통해 전송된 데이터를 애플리케이션에서 컨트롤함으로써 NAT에 대한 문제점을 해결하였다.

제 2장 멀티미디어 데이터 처리기술

2.1 MPEG-4

2.1.1 MPEG-4의 개념

1) 개념

MPEG는 Moving Picture Expert Group 의 머리글자를 취한 말이며, 본래 멀티미디어 encode를 실시하고 있는 조직의 약칭이었지만, 최근에는 이 조직이 작성한 표준 규격의 통칭으로서도 사용되고 있다.

MPEG 규격은 원래, 축적 미디어, 방송, 통신 등을 (위해)때문에의 멀티미디어 encode의 규격이며, 주로 비디오 신호의 encode 방법에 관한 규정, 오디오 신호의 encode 방법에 관한 규정, 및 양자의 통합 방법 등의 시스템에 관한 규정, 의 3개 (살)로부터 완성되어 끊은 것이었다. 그러나 MPEG의 국면이 진행되는 것에 따라, 보다 광범위한 내용을 포함하는 것에 변모하고 있다. MPEG-4는 1993년에 검토가 시작되었다. MPEG-4는, 종래의 MPEG-1/2의 고화질을 추구하는 흐름과는 달리, 저비트율에서의 고화질의 달성을 목표로 표준화가 시작되었다. 현재는 고능률 encode, 강력한 에러 내성능력, 멀티미디어 대응의 특징을 가진 차세대의 멀티미디어 encode로서 국제 표준화 작업이 되가고 있다[11-12].

2.2 H.261

2.2.1 H.261의 개념

1) 개념

ITU-T(구 CCITT)에 의해 만들어진 국제표준인 H.261은 종합정보통신망(ISDN: Integrated Services Digital Network)을 이용한 영상전화 및 영상회의를 위한 동영상 압축방식이다. 이는 정지화 압축에 관한 국제표준인 JPEG과 더불어 오늘날 멀티미디어 혁명의 중심이 되고 있는 국제표준인 MPEG1과 2의 모태라고 할 수 있다.[13]

2) 특징

ISDN에서 규정하고 있는 채널에는 음성 및 팩스 등 기본 정보전송을 위한 B 채널 64Kbps 동영상 및 고속데이터 전송을 위한 H채널(H는 3백84Kbps, H납은 1천5백36Kbps), 그리고 여러 가지 제어신호용의 D채널(16Kbps 혹은 64Kbps)등 세 종류가 있다. 실제 사용 시에는 이 세 채널을 적절히 조합하여 기본접속 혹은 1차군 접속의 형태를 취한다. 가정에 연결되는 기본접속은 현재의 전화선과 같이 2선식 나선형 동선을 이용하고 있고, 두개의 B채널과 하나의 D채널(16Kbps)을 시분할다중화하여 (2B+D) 총 1백44Kbps의 데이터 전송속도를 갖는다. 맥내 배선은 4선식 버스방식으로 최대 8개까지의 단말을 연결할 수 있는데 전화-팩스-저속 컴퓨터 통신을 동시에 할 수 있고 1가구 2전화기 실현된다.

이 ISDN을 이용한서비스의 일환으로 나온 것 중 하나가 현재의 전화선을 이용한 G3팩스를 고속, 고해상도로 개선한 G4팩스이고 또 하나가 얼굴영상과 음성을 함께 전송하는 영상전화이다. 이때 음성은 G.711(64Kbps PCM) 혹은 G.728(16Kbps LD-CELP)에 의해 부호화하고 얼굴영상은 H.261에 의해 부호화한 후 이들을 H.221에 의해 다중화 하고 망 인터페이스를 부가하면 ISDN영상전화 단말기인 H.320이 된다. 이때 ISDN기본접속이 제공하는 비트속도가 1백44Kbps밖에 되지 않아 음성과 제어용으로 사용하는 비트를 빼고 남은, 즉 영상에 사용할 수 있는 비트는 매우 부족하다. 따라서 초당 보낼 수 있는 화면수와 화질에 제약이 많다. 이것은 영상전화의 성격이 음성을 통한 정보전달이주이고 영상은 보조기능이라는 점을 감안한 것이다. 사무실 업무용의 ISDN 접속은 가정의 기본 접속보다 전송속도가 훨씬 높은 1차군 접속일 때가 많다. ISDN 1차군 접속은 북미와 일본에서는 1천5백36Kbps, 유럽에서는 2천48Kbps로 B H D 등 세 종류의 채널을

여러 방법으로 조합하여 사용할 수 있다.

이 높은 전송률을 활용하는 방안 중 하나가 바로 영상회의이다. 영상회의는 멀리 떨어져 있는 다자간에 시간과 경비를 절약하면서 회의를 하고자 할 때 유력한 수단이다. 보통 여러 사람이 큰 화면을 통해 서로를 보면서 회의를 하기 때문에 두 사람이 작은 액정화면을 보며 대화하는 영상전화에 비해 화질과 음질이 훨씬 좋아야 한다. 따라서 음성코덱은 AM방송에 가까운 음질을 제공하는 G.722(64Kbps 이하, SB-ADPCM방식)를 사용하며, 영상코덱은 영상전화와 마찬가지로 H.261을 사용하는데 단지 사용가능한 비트가 훨씬 많다. 즉 H.261은 p*64Kbps의 영상 전화 및 영상회의를 위한 동영상 압축표준으로 ISDN 기본접속 및 1차군 접속을 감안한 것이므로 p는 1~30의 값을 취하는데 보통 영상전화는 p값이 1~2, 영상회의는 6 이상이다(H 채널은 p값이 6이됨).

2.3 H.263

2.3.1 H.263의 개념 및 표준화

H.263은 64kbit/s 이하의 저 전송률 영상통신을 위한 영상 부호화에 관련된 표준으로써 현재 폭넓은 응용에 사용되고 있다. H.261에 기반을 두고 있지만 H.261보다 다음과 같은 개선된 부호화방식을 갖는다.[14]

- ◇ 반 화소 단위의 정밀도를 갖는 움직임 보상
- ◇ 비제한적인 움직임 벡터(unrestricted motion vector)
- ◇ 문법 기반의 산술 부호화(syntax-based arithmetic coding)
- ◇ 고급 예측(advanced prediction)
- ◇ PB-프레임들을 사용

2.4 DICOM

2.4.1 DICOM의 개요

Digital Imaging and Communications in Medicine (DICOM)은 1993년 3.0 버전이 발표된 이후로 의료 영상 산업의 표준으로 자리잡고 있다[15]. 그러나 DICOM 표준은 매우 방대하여 의료 영상 업체 마다 개별적으로 이를 구현하는 일은 많은 시간과 노력을 필요로 하였다. 그러므로 대부분의 의료 영상 관련 업체에서는 DICOM 표준을 구현한 소프트웨어들을 사용하고 있다.

2.4.2 KDICOM

K-DICOM Toolkit (KDTK)는 DICOM 표준 구현과 영상 라이브러리를 모두 제공한다. 또한 KDTK에 사용되는 영상 라이브러리는 DICOM 데이터 셋만을 출력하도록 제작되어 의료 영상에 필요한 기능들이 대부분 구현되어 있어 개발자가 따로 개발해야 할 수고를 덜어준다.

KDTK의 주요 특징들을 살펴보면 다음과 같다.

가) Data Module

DICOM 데이터 셋을 처리하는 모듈은 그림 2-1과 같이 세 개의 모듈로 구성되어 있다. Data Element Module은 DICOM 3.6에 명시된 Data Element의 속성들을 저장할 수 있는 정보 객체이다.

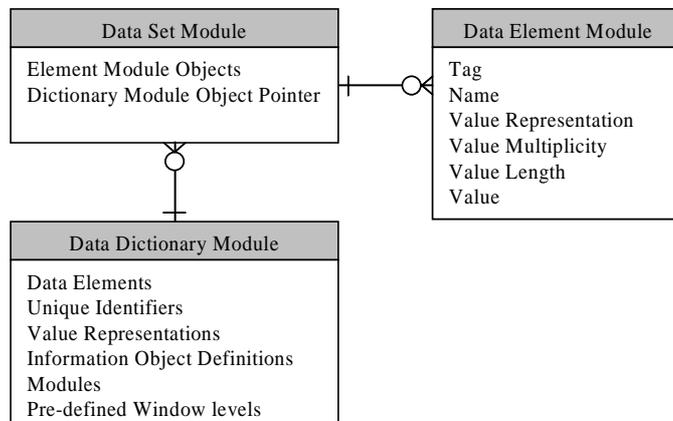


그림 2-1. KDICOM에서의 데이터 모듈

나) Image Module

Image Module은 Dataset Module의 자료를 토대로 최적 영상을 생성하여 화면에 출력하며 그림 2-2과 같이 네 개의 모듈로 되어있다. Image Module은 주석의 입출력을 위하여 Draw List Module들을 가진다. Draw List Module은 주석 객체인 Draw Object Module들의 생성, 삭제, 변경을 관리한다. Thumbnail Image Module은 Image Module에서 파생된 모듈로서 Image Module에 비하여 출력 속도가 빠르므로 미리 보기, 실시간 영상 처리 등에 사용한다.

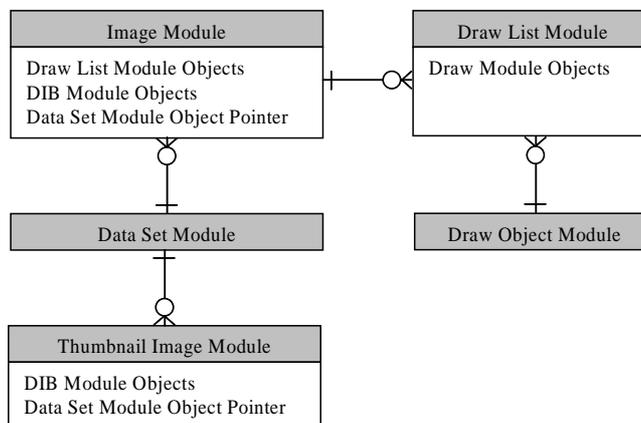


그림 2-2. KDICOM에서의 이미지 모듈

Image Module은 Run Length Encoding (RLE) 및 JPEG Baseline 압축의 경우 자체 코덱을 내장하여 바로 출력이 가능하다. 그 외의 DICOM 표준에 명시된 압

축 Transfer Syntax는 출력할 수 없다. 화면 Overlay 요소에는 그림 2-3과 같이 영상의 정보를 출력하는 텍스트, 영상의 실제 크기를 보여 주는 Ruler, 사용자가 추가할 수 있는 주석, 모니터를 조정하는데 참조할 수 있는 그레이스케일, 영상의 위치를 알려 주는 영상 위치, 심전도 등을 보여주는 파형 등이 있다.

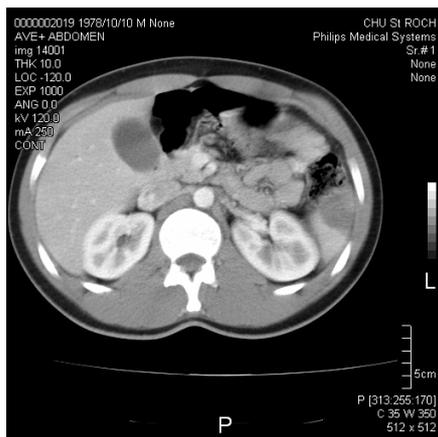


그림 2-3. KDICOM에서의 오버레이 기능

Image Module에서는 윈도우 프린터에 출력할 수 있는 기능을 제공하며 DICOM 프린트를 위하여 아래에 설명할 별도의 모듈을 사용한다. 데이터 셋이 동 영상일 경우에는 Play, Stop 등과 같은 다양한 동영상 모드를 제공하며 시간 간격을 조정할 수 있으며 심전도와 같은 1차원 신호도 같이 출력한다. 영상 조작 기능으로 Zooming, Panning, Histogram Window, Magnify, Inverse를 제공하며 영상 처리 기능으로 Blur, Sharpen, Emboss, Edge Detect, Rotate, Flip 등을 제공한다. 영상이 확대되었을 경우 보간 모드와 일반 모드를 선택할 수 있다. 주석 객체에는 선, 네모, 원, 텍스트, 자유 그리기, 거리, 각도를 제공한다. 네모, 원, 자유 그리기 객체에서는 ROI 기능을 통하여 넓이, 높이, 면적, 기술 통계량을 구할 수 있다. 추가된 주석은 별도의 파일로 저장할 수도 있고 영상에 추가할 수도 있다. 윈도우를 조절하기 위해서는 마우스 오른쪽 버튼을 사용하며 그림 2-4와 같이 히스토그램을 보며 윈도우를 조정할 수 있는 기능도 제공한다.

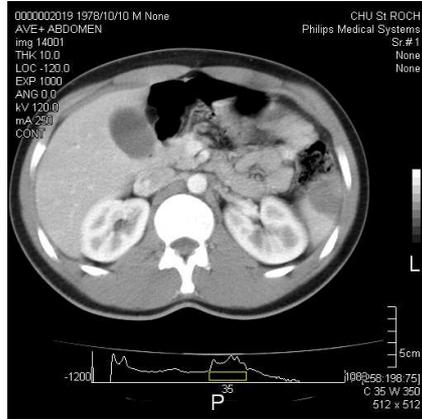


그림 2-4. KDICOM에서의 히스토그램을 이용한 윈도우 조정

다) Network Modules

KDTK는 네트워크 상에 있는 DICOM 어플리케이션과 통신하기 위하여 TCP/IP 기반의 윈도우 소켓을 사용하며 모듈들의 구성은 그림 2-5와 같다. DICOM 어플리케이션과의 Association이 성공한 후 Release 될 때까지의 작업을 하나의 쓰레드로 처리한다. Thread 기반으로 DICOM Message Service Element (DIMSE) Service를 처리하기 때문에 영상을 다중으로 전송할 수 있고 영상을 전송하며 다른 작업을 수행할 수도 있는 등 멀티 쓰레드 작업이 가능하다. Socket Manager Module은 상기한 쓰레드의 생성 및 파괴를 관리하는 모듈이다. Socket Thread Module는 하나의 Socket Module을 가지며 Socket Module은 두개의 Associate Module을 갖는다.

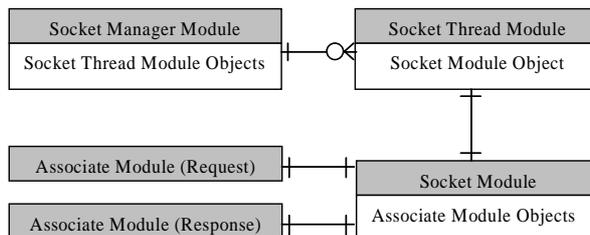


그림 2-5. KDICOM에서의 네트워크 모듈

2.5 DirectShow

2.5.1 DirectShow의 개요

다이렉트쇼의 모체는 비디오 캡처의 주요기술인 'VfW(video for Windows)'라고 할 수 있다. 그 이유는 VfW의 다양한 기술이 모두 다이렉트쇼로 흡수되었기 때문이다. 비디어 캡처 기술은 처음에 단순히 비디오 편집을 위해 입력된 영상 파일을 저장하는 것을 주된 목적으로 하였지만, PC에서 멀티미디어에 대한 요구가 점점 늘어나면서 단순 저장보다는 화상회의 같은 다양한 애플리케이션 형태로 발전하게 되었다. VfW기술은 이러한 다양한 요구를 받아들이기에는 역부족이었으며, 벤더들은 자사의 독점 기술을 사용하여 VfW기술을 확장하고 이를 자사의 애플리케이션에 사용하게 되었다. 이 때문에 애플리케이션은 멀티미디어에 대한 요구에 부합하여 확장되었지만, 시장의 표준은 무너지고 서로 호환되지 않는 기술들이 난립하여 결과적으로 사용자의 불편만 가중시켰다. 이러한 시점에서 마이크로소프트는 새로운 애플리케이션을 분석하기에 이르렀다. 그 결과 그 동안 벤더들이 독자적으로 확장한 VfW 기술을 흡수하고 윈도우에 흩어져 있는 각종 멀티미디어 기술을 통합하는 다이렉트쇼를 발표하였다[16].

특징	세부내용
데이터 처리	멀티미디어 데이터는 대용량일 뿐만 아니라 신속히 처리해야하는 데이터이다.
동기문제	비디오와 오디오, 그리고 부가 데이터 간의 동기 문제를 처리해야 한다.
다양한 입력	로컬, 네트워크, 지상/위성/케이블 TV, 비디오 카메라, 사운드 카드등 다양한 입력을 처리해야한다.
다양한 포맷	AVI, ASF/WMV, MPEG, DVD, MJPEG, MP3, WAV 등 다양한 포맷을 처리할 수 있어야 한다.
다양한 출력	비디오 카드와 사운드 카드, 로컬, 네트워크 등 다양한 출력을 처리해야 한다.
하드웨어 장치	하드웨어 존재 유무를 감지하여 하드웨어가 존재한다면 최대한 하드웨어를 활용할 수 있어야 한다.

표 2-6. DirectShow에서의 멀티미디어 데이터 처리 특징

특징	해결방안
데이터 처리	다이렉트드로와 다이렉트사운드를 사용하여 데이터 처리의 효율성을 극대화한다.
동기문제	멀티미디어 데이터를 타임스탬프를 사용한 멀티미디어 샘플로 포장하여 동기문제를 처리한다.
다양한 입·출력 및 포맷	'필터(filter)'라고 불리는 컴포넌트 구조로 다양한 입력, 다양한 포맷, 그리고 다양한 출력문제를 해결할 수 있다. 다양한 상황에 대해서도 적절히 컴포넌트를 조합함으로써 문제를 해결할 수 있다.
다양한 하드웨어 장치	역시 컴포넌트 구조로 커널 모드(kernel-mode)를 대표하는 필터를 사용하거나 다이렉트드로와 다이렉트사운드의 하드웨어 추상화 방식을 활용한다.

표 2-2. DirectShow에서의 멀티미디어 데이터의 처리 방안

다양한 입력, 다양한 포맷, 그리고 다양한 출력에 대한 문제를 해결하기 위해 마이크로소프트에서는 컴포넌트 구조를 도입하였다. 이러한 다양성은 고정된 것이 아니라 계속 변화할 수 있으며 필요한 상황 또한 다양한 형태로 발생할 수 있으므로, 고정적인 구조로는 이러한 성질을 수용할 수 없기 때문이다. 따라서 컴포넌트 구조가 갖는 유연성을 활용하여 다양한 환경에 대해 마치 레고 블록을 맞추듯 컴포넌트를 적절히 조합함으로써 이러한 문제점을 해결할 수 있다. 다양한 하드웨어 장치들이 커널 모드에서 동작하기 때문에, 커널 모드와 통신할 수 있는 특별한 사용자 모드(user-mode)필터를 사용해야 한다. 또한 다이렉트드로와 다이렉트사운드가 포함하고 있는 하드웨어 추상화 계층을 활용하여 비디오와 오디오 하드웨어 장치를 해결할 수도 있다.

2.5.2 DirectShow의 필터 연결

다이렉트쇼 필터의 연결 과정은 크게 두가지 협상(negotiation)과정을 포함하고 있다. 하나는 '미디어형 협상'이고 다른 하나는 '버퍼협상'이다. 순서는 미디어협상이 먼저 이루어진 다음에 버퍼 협상에 들어가게 된다[17].

1) 미디어형 협상

가) 미디어형 열거

IPin::Connect() 함수가 호출되었을 때 미디어형이 입력되지 않았다면 미디어형 열거 과정을 거쳐야 한다. 그렇지 않고 미디어형이 명시되었다면 바로 연결을 시도하는 단계로 넘어간다.

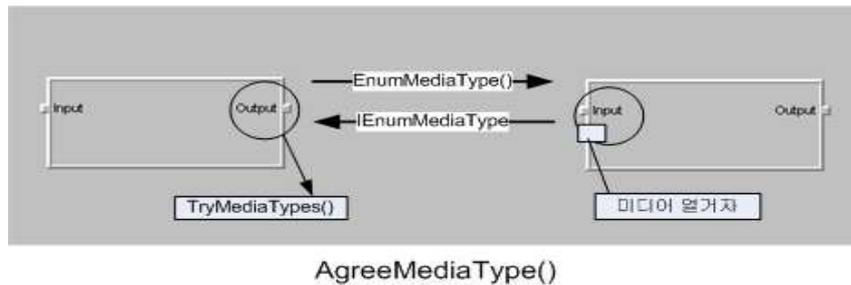
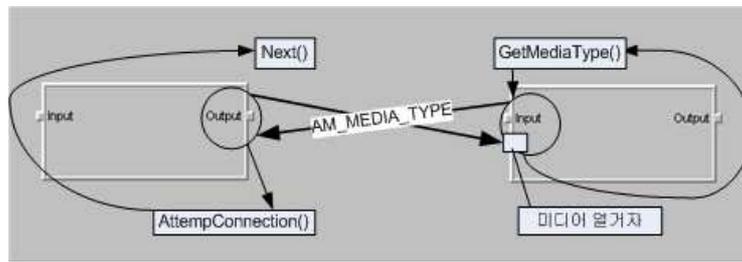


그림 2-6. DirectShow의 필터 연결에 필요한 미디어형을 열거하는 첫 번째 과정

CBasePin::AgreeMediaType() 멤버함수는 명시된 미디어형이 없을 경우, 그림 2-6과 같이 먼저 상대 핀의 IPin::EnumMediaType() 함수를 호출한다. 연결 작업이 출력 핀부터 시작되었으므로, 대부분의 경우 아래쪽 필터의 입력 핀에 먼저 IPin::EnumMediaType() 함수가 호출된다. 이 함수는 CBasePin::EnumMediaTypes() 멤버 함수가 구현하고 있는데, 함수가 호출되면 IEnumMediaType 인터페이스를 구현하고 있는 미디어형 열거자를 생성한 후, IEnumMediaType 인터페이스를 위쪽 필터의 출력핀에 전달한다. 이 방식은 CBaseFilter::EnumPins() 멤버함수가 IEnumPins 인터페이스를 리턴하는 방식과 매우 흡사하다. DirectShow에서는 이러한 열거자 방식을 많이 사용한다. IEnumMediaType 인터페이스를 전달 받은 출력핀은 CBasePin::TryMediaTypes() 멤버 함수를 호출한다. 이 함수는 말 그대로 미디어형 열거자가 제시하는 미디어형에 대해 가능성을 시도해 보는 함수이다.



TryMediaTypes()

그림 2-7. DirectShow의 필터 연결에 필요한 미디어형을 열거하는 두 번째 과정

그림 2-7은 CBasePin::TryMediaTypes() 멤버 함수의 절차를 표현한 것이다. IEnumMediaType 인터페이스를 전달받은 CBasePin::TryMediaTypes() 멤버 함수는 IEnumMediaType::Next() 함수를 호출하여 AM_MEDIA_TYPE 변수를 요청한다. 이 과정을 수행하기 위해 미디어형 열거자는 입력 핀의 CBasePin::GetMediaType() 멤버 함수를 호출한다. IBaseFilter 인터페이스에 핀을 가져올 수 있는 함수가 부족했던 것처럼 IPin 인터페이스에도 IPin::ConnectionMediaType() 함수 외에는 미디어형을 가져올 수 있는 함수가 없다. 더구나 IPin::ConnectopmMediaType() 함수는 연결에 사용된 미디어형을 전달하는 함수로, 색인을 지정하는 CBasePin::GetMediaType() 멤버 함수와는 성격이 다르다. 이는 DirectShow 하부구조가 열거자를 기본으로 사용하도록 설계되었기 때문이다. 출력 핀은 미디어 형 열거자로부터 전달받은 미디어형, 즉 AM_MEDIA_TYPE 변수를 이용해 연결을 시도하게 된다. 연결을 시도하는 함수는 CBasePin::AttemptConnection() 멤버 함수이다. 이 함수는 전달된 미디어 형으로 연결을 시도하고, 실패하면 오류값을 전달한다. 연결을 시도하다 오류가 발생하면 CBasePin::TryMediaTypes() 멤버 함수는 다시 입력 핀의 미디어형 열거자에 IEnumMediaType::Next() 함수를 호출하고, 전달받은 미디어형으로 다시 연결을 시도한다.

나) 연결 시도

CBasePin::AgreeMediaType() 멤버 함수는 IPin::Connect() 함수로부터 제대로 명시된 미디어형을 전달받았을 경우, 바로 CBasePin::AttemptConnection() 멤버 함수를 호출하여 연결을 시도한다. 그렇지 않은 경우라면 미디어형을 열거한 후 여기서 얻은 미디어형으로 연결을 시도한다. 미디어형을 열거할 때는 아래쪽 필터의 입력핀에서 위쪽 필터의 출력핀으로 그 순서를 진행하지만, 연결을 시도할 때는 항상 필터그래프와 같은 방향으로 시도된다.

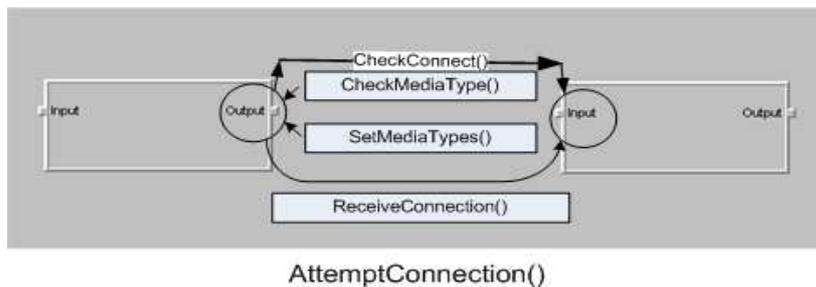


그림 2-8. DirectShow의 필터 연결을 시도하는 첫 번째 단계

그림 2-8 에서도 볼 수 있듯이 CBasePin::AttemptConnection() 멤버함수는 출력 핀에 CBasePin::CheckConnect() 멤버함수를 호출하여 핀의 방향을 확인하고, CBasePin::CheckMediaType() 순수 가상 멤버 함수를 호출하여 출력 핀이 연결할 수 있는 미디어형인지 확인한다. 이 멤버 함수는 순수 가상 함수(pure virtual function)이므로 CBasePin을 상속받은 클래스에서 반드시 구현해야 하며, 이로 인해 상속받은 클래스에서 미디어형을 명확히 확인해야 한다. CBasePin::CheckMediaType() 순수 가상 멤버 함수에서 성공하면 CBasePin::SetMediaType() 멤버 함수를 이용해 일단 `m_mt` 멤버 변수에 현재의 미디어형을 저장한다. 그리고 나서 아래쪽 필터의 입력 핀에 CBasePin::ReceiveConnection() 멤버 함수를 호출하면서 출력 핀의 IPin 인터페이스를 매개변수로 넘겨준다. 이 함수는 출력 핀의 연결 시도를 수락해 줄 것을 요청한다.

CBasePin::ReceiveConnection 멤버 함수를 처리 하는 순서는 CBasePin::AttemptConnection() 멤버 함수의 절차와 유사하다.(그림 2-12 참조) 입

력 핀에 CBasePin::CheckConnect() 멤버 함수를 호출하여 핀의 방향을 확인하고, CBasePin::CheckMediaType() 순수 가상 멤버 함수를 호출하여 입력 핀이 연결할 수 있는 미디어형인지 확인한다. CBasePin::CheckMediaType() 순수 가상 멤버 함수가 성공하면 CBasePin::SetMediaType() 멤버 함수를 이용해 m_mt 멤버 변수에 현재 미디어형을 저장한다. 그리고 나서 CBasePin::CommpleteConnect() 멤버 함수를 호출한다. 이 함수 CBaseFilter를 상속받는 클래스에서 사용할 수 있다. 그리고 나서 CBasePin::ReceiveConnection() 멤버 함수 호출에 성공하면 CBasePin::AttemptConnection() 멤버 함수는 성공 메시지를 받아들여 출력 핀에도 CBasePin::CompleteConnect() 멤버 함수를 호출하고, 이로써 미디어 협상이 완료된다.

2) 버퍼협상

버퍼 협상은 미디어형 협상 과정을 마무리 하는 CBasePin::CompleteConnect() 멤버 함수가 호출되면 바로 시작된다. 하지만 CBasePin::CompleteConnect() 멤버 함수는 아무 역할도 하지 않는다. 출력 핀의 경우 CBaseOutputPin 클래스에서 CompleteConnect() 멤버 함수를 오버라이딩하여 버퍼 협상을 진행하도록 수정하였다. 그림 2-9와 같이 아래쪽 필터의 입력 핀의 CBasePin::CompleteConnect() 멤버 함수가 아무런 역할도 하지 않는 반면, 위쪽 필터 출력 핀의 CBaseOutputPin::CompleteConnect() 함수는 CBaseOutputPin::DecideAllocator() 멤버 함수를 호출됨으로써 버퍼 협상을 시작한다.

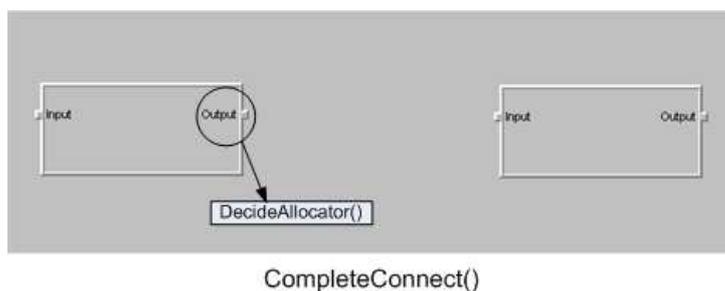


그림 2-9. DirectShow의 버퍼협상 과정 중 출력 핀에서 버퍼 협상이 시작

아래 시나리오의 실행 주체는 위쪽 필터의 출력 핀이고 요청 대상은 아래쪽 필터의 입력 핀이다.

◆ 버퍼 협상 과정 시나리오

- ① 출력 핀의 할당자 요구 사항의 요구 (IMemInputPin::GetAllocatorRequirments() 함수)
- ② 할당자를 받는다. (IMemInputPin::GetAllocator() 함수)
- ③ 할당자 설정 (CBaseOutputPin::DecideBufferSize() 순수 가상 멤버 함수)
- ④ 할당자 사용 (IMemInputPin::NotifyAllocator() 함수)
- ⑤ 할당자 거부시 출력핀이 할당자를 생성 (CBaseOutputPin::InitAllocator() 멤버 함수)
- ⑥ 할당자 설정 (CBaseOutputPin::DecideBufferSize() 순수 가상 멤버 함수)
- ⑦ 입력핀에게 할당자 확인 요구 (IMemInputPin::NotifyAllocator() 함수)

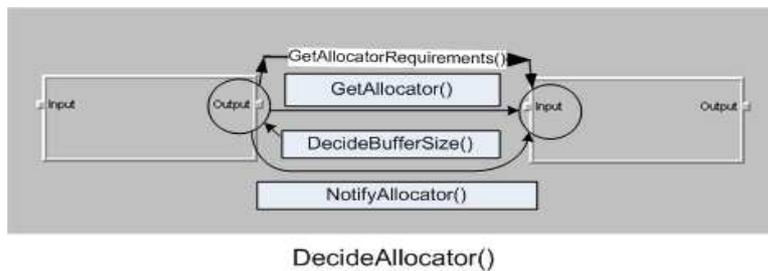


그림 2-10. DirectShow의 버퍼협상의 첫 번째 단계

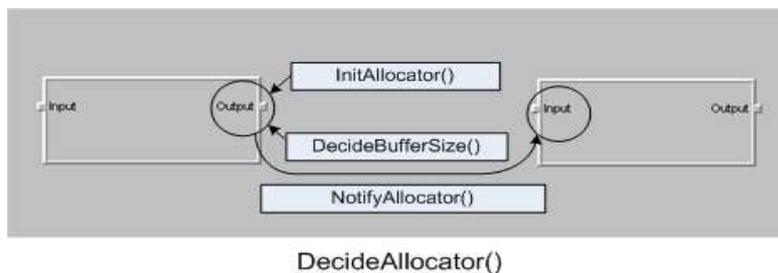


그림 2-11. DirectShow의 버퍼협상의 두 번째 단계

그림 2-10과 그림 2-11는 버퍼 협상 시나리오에 따른 버퍼 협상과정을 표현한 것이다. 그림 2-10의 첫 번째 단계가 성공하면 그림 2-11의 두 번째 단계는 진행할 필요가 없다. 그러나 첫 번째 단계가 실패하면 두 번째 단계로 진행하여 다시 버퍼 협상을 시도한다. 두 번째 버퍼 협상도 실패한다면 다시 미디어 협상의 처음 단계로 돌아가야 한다.

제 3장 네트워크 통신 프로토콜

3.1 TCP/IP

3.1.1 TCP/IP 프로토콜 구조

인터넷을 통해 통신을 수행하는 개체는 크게 종단 시스템과 라우터로 나눌 수 있다. 종단 시스템(end-system)은 최종 사용자(end-user)를 위한 애플리케이션을 수행하는 주체가 되며, 인터넷에 연결된 PC, 워크스테이션, PDA, 휴대폰 등이 여기에 속한다. 라우터(router)는 종단 시스템이 속한 네트워크와 다른 네트워크를 연결함으로써 서로 다른 네트워크에 속한 종단 시스템끼리 상호 데이터를 교환할 수 있도록 하는 장비다. 종단 시스템과 라우터 간, 라우터와 라우터 간, 그리고 종단 시스템과 종단 시스템 간 통신을 수행하기 위해서는 정해진 절차와 방법을 따라야 하는데 이를 프로토콜이라 부른다. 그림 3-1 에서 1~4는 모두 상호 약속한 프로토콜을 이용하여 통신을 수행한다.



그림 3-1. 인터넷의 구성도

인터넷에서 사용하는 핵심적인 프로토콜은 TCP와 IP로, 이를 포함한 각종 프로토콜을 이용하여 TCP/IP 프로토콜이라 부른다[18-19]. TCP/IP 프로토콜은 일반적으로 운영체제에서 그 구현을 제공하며, 일반 애플리케이션은 운영체제가 제공하는 TCP/IP 프로토콜의 서비스를 사용하여 통신을 수행하게 된다(그림 3-2). 그림에서 통신을 수행하는 주체며 최종적인 목적지가 되는 것은 종단 시스템 자체가 아

닌 중단 시스템에서 수행되는 애플리케이션이다.

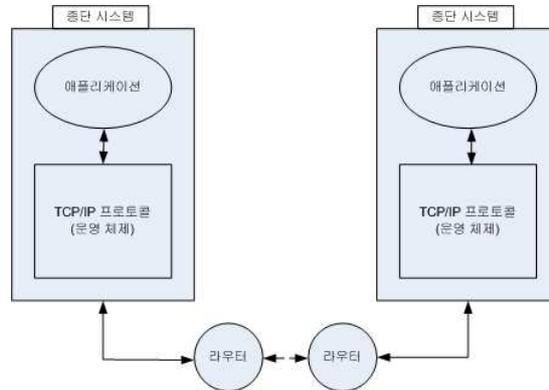


그림 3-2. TCP/IP 프로토콜을 이용한 통신

애플리케이션간 통신을 수행하기 위해서는 다양한 요구 조건을 만족시켜야 한다. 이러한 요구 조건의 예로는 통신할 대상(중단 시스템 자체, 중단 시스템에서 수행되는 애플리케이션)을 지정하는 상호 약속된 방법, 전송 오류 확인 기능, 오류 발생 시 재전송 기능, 데이터의 순서 관계 유지 등을 들 수 있다. 일반적으로 이와 같은 기능은 프로토콜 수준에서 제공한다.

일반적으로 프로토콜은 기능별로 나누어 계층적으로 구현하며, TCP/IP 프로토콜도 이와 같은 구조를 따르고 있다. 그림 3-3은 TCP/IP 프로토콜 구조를 도시한 것이다.



그림 3-3. TCP/IP 프로토콜 구조

한편, 인터넷 계층의 IP가 제공하는 전송 서비스는 최선을 다하지만(best effort), 신뢰성은 없다는(unreliable)특징이 있다. 즉, 데이터에 문제가 생기면 목적지에 도

달하지 못하는 상황이 발생할 수 있으며, 목적지에 도달하였다더라도 실제 내용이 손상되었을 가능성도 있게 된다. 전송 계층에서는 이러한 데이터 손실 또는 손상을 검출하여 잘못된 데이터가 목적지 프로세스에 전달되는 것을 방지한다.

TCP/IP 프로토콜에서 전송 계층에 해당되는 프로토콜로는 TCP(Transmission Control Protocol)과 UDP(User Datagram Protocol)가 있다. 두 프로토콜의 특징은 다음과 같다.

TCP	UDP
연결형(Connection-oriented)프로토콜 - 연결이 성공해야 통신 가능	비연결형(connectionless)프로토콜 - 연결없이 통신 가능
데이터 경계를 구분하지 않음 - 바이트 스트림(byte-stream)서비스	데이터 경계를 구분함 - 데이터그램(datagram)서비스
신뢰성 있는 데이터 전송 - 데이터를 재전송함	비신뢰적인 데이터 전송 - 데이터를 재전송하지 않음
1대1 통신(Unicast)	1 대 1 통신(Unicast) 1 대 다 통신(Broadcast) 다 대 다 통신(Multicast)

표 3-1. TCP 와 UDP 프로토콜의 특징

3.1.2 IP 주소, 포트 번호

TCP/IP 프로토콜을 이용하여 통신할 때는 IP주소와 포트번호를 사용한다. IP 주소는 32비트 값으로, 일반적으로 8비트 단위로 구분하여 각각 10진수로 표기한다.

예) 147.46.114.70

폐쇄된 네트워크거나 IP를 공유하는 경우가 아니라면 IP주소는 전세계적으로 유일한 값을 가진다. 라우터가 라우팅을 할때는 IP헤더에 포함된 수신자 IP주소를 이용하여, 데이터를 받은 쪽에서 답장을 보내고자 할때도 역시 IP헤더에 포함된 송신자 IP 주소를 이용한다.

IP주소는 인터넷에 존재하는 호스트(종단 시스템, 라우터)를 유일하게 구별할 수 있지만 통신의 최종 주체인 프로세스를 식별하지는 못한다. 포트번호는 각 프로세스를 구별하는 식별자다. 그러나 실제로는 한 프로세스가 두개 이상의 포트

번호를 사용하는 경우가 종종있으며, 한 포트 번호를 두개 이상의 프로세스가 사용하는 것도 가능하다(그림 3-4). 따라서 포트번호는 프로세스를 구별하는 식별자라기 보다는 통신의 종착지(하나 혹은 여러개의 프로세스)를 나타내는 식별자로 보는것이 바람직하다.

TCP와 UDP는 포트번호로 부호 없는 16비트 정수를 사용하므로 0~65535범위가 가능하지만, 모든 범위를 자유롭게 쓸 수 있는 것은 아니다. 표 3-2는 포트번호를 영역별로 나눈 것이다. 여기서 포트번호 0~1023은 용도가 정해져 있으므로 함부로 사용해서는 안된다는 것이다. 일반적으로 서버를 작성할 때는 1024~49151 범위에서 하나 또는 일부를 선택하여 사용하는 것이 바람직하다.

포트번호	분류
0~1023	Well-known ports
1024~49151	Registered ports
49152~65535	Dynamic and/or private ports

표 3-2. 네트워크 통신에 필요한 포트 번호 분류

IP주소와 포트 번호의 관계를 도식화 하면 그림 3-4와 같다. 이 그림은 두개의 IP주소를 가진 중단 시스템을 나타낸다. 한 프로세스가 두개 이상의 TCP혹은 UDP 포트를 사용하거나, 한 포트 번호를 두개 이상의 프로세스가 사용하는 것을 보여주고 있다.

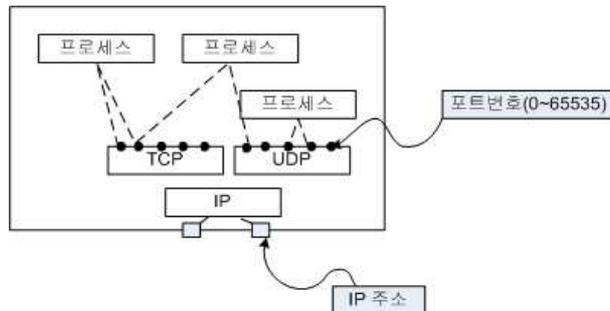


그림 3-4. TCP/IP 통신에서의 IP주소와 포트번호의 역할

3.1.3 클라이언트/서버모델

네트워크 프로그램은 일반적으로 클라이언트/서버 모델(client/server model)로 작성한다. 여기서 클라이언트/서버는 두개의 애플리케이션이 상호 작용하는 방식을 나타내는 용어로, 클라이언트가 서버에 요청을 하면 서버는 이 요청을 받아 처리하게 도니다. 하나의 컴퓨터에서 실행되는 두 프로그램간에 클라이언트/서버 모델이 적용될 경우, 두 프로그램은 프로세스간 통신(IPC, Inter-Process Communication)기법을 사용하여 상호 정보를 교환한다. 반면, 네트워크로 연결된 두 컴퓨터에서 실행되는 두 프로그램 간에 클라이언트/서버 모델이 적용될 경우, 두 프로그램은 통신 프로토콜(TCP/IP)을 사용하여 상호 정보를 교환한다.

클라이언트/서버 모델에서는 한 프로세스가 먼저 실행하여 대기하고, 다른 프로세스가 나중에 실행하여 접속한다. 이때 먼저 실행하는 쪽이 서버(server)이고, 나중에 실행하는 쪽이 클라이언트(client)이다. 클라이언트가 서버에 접속하기 위해서는 서버의 IP 주소(또는 도메인 이름)와 포트 번호를 미리 알고 있어야 한다. 반면, 서버는 클라이언트의 주소를 미리 알 필요가 없다. 클라이언트가 보낸 패킷에는 클라이언트 주소 정보가 모두 들어 있기 때문에, 서버는 이 정보를 이용하여 언제든지 해당 클라이언트에 데이터를 보낼 수 있다.

3.2 TCP

3.2.1 TCP 서버/클라이언트 구조

TCP는 연결 지향성을 가지는 신뢰성 있는 데이터 전송 프로토콜이다. 즉, 전송도중에 발생하는 에러나 데이터의 손실을 재전송이라는 메커니즘을 통하여 극복한다. TCP의 특성은 아래와 같다.

1) 에러나 데이터의 손실 없이 무사히 전달된다.

2) 전송하는 순서대로 데이터가 전달된다.

전송라인이 하나이기 때문에 뒤에서 보낸 데이터가 먼저 도착할 수는 없다.

3) 전송되는 데이터의 경계(Boundary)가 존재하지 않는다.

즉, 호스트 대 호스트가 연결 지향 소켓을 생성해서 통신을 하는 경우, 한 번의 데이터를 전송했다고 해서 수신측에서 한번의 read함수를 통해 데이터를 모두 수신한다는 보장이 없고 반대로 송신측에서 여러 번 데이터를 보내도, 수신측에서 한번의 read함수를 통해서 모든 데이터를 수신할 수도 있다.

위와 같은 특성을 가진 TCP의 서버/클라이언트의 동작 방식은 다음과 같다(그림 3-5).

- ① 서버는 먼저 실행하여 클라이언트가 접속하기를 기다린다.(listen)
- ② 클라이언트가 서버에 접속(connect)하여 데이터를 보낸다(send, write).
- ③ 서버는 클라이언트 접속을 수용하고(accept), 클라이언트가 보낸 데이터를 받아서(recv, read)처리한다.
- ④ 서버는 처리한 데이터를 클라이언트에 보낸다(send, write).
- ⑤ 클라이언트는 서버가 보낸 데이터를 받아서(recv, read)자신의 목적에 맞게 사용한다.

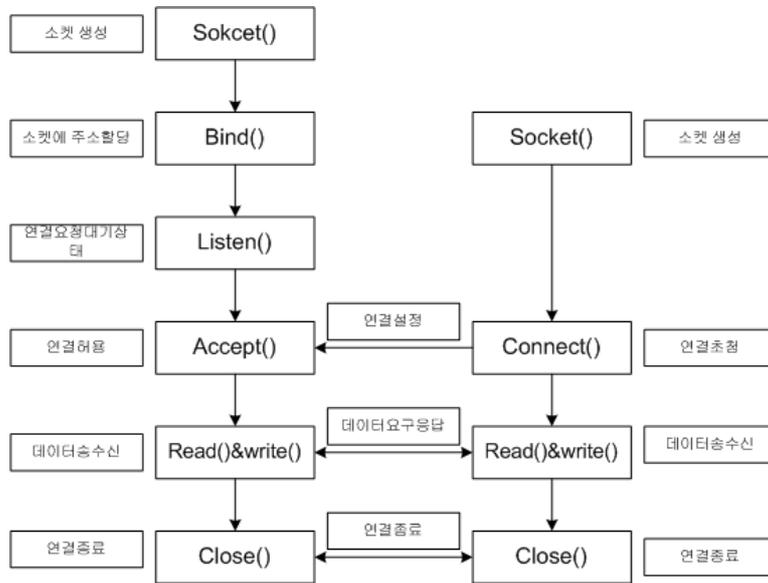


그림 3-5.TCP 연결 지향형 소켓 시스템 호출

3.2.2 동작원리

그림 3-6은 TCP 서버에 TCP 클라이언트가 접속하여 통신을 수행하는 과정을 보여준다.

각 단계별 동작을 요약하면 다음과 같다.

- 그림 3-6의 (a) : 서버는 소켓을 생성한 후 클라이언트가 접속하기를 기다린다. 이때 서버가 사용하는 소켓은 특정 포트 번호(예를 들면 9000)와 결합되어(bind) 있어서 이 포트번호로 접속하는 클라이언트만 수용할 수 있다.
- 그림 3-6의 (b) : 클라이언트가 접속한다. 이때 TCP 프로토콜 수준에서 연결 설정을 위한 패킷 교환이 이루어진다.
- 그림 3-6의 (c) : TCP 프로토콜 수준의 연결 절차가 끝나면, 서버는 접속한 클라이언트와 통신할 수 있는 새로운 소켓을 생성한다. 서버가 클라이언트와 데이터를 주고받을 때는 이 소켓을 사용한다. 기존의 소켓은 새로운 클라이언트 접속을

수용하는 용도로 계속 사용한다.

- 그림 3-6의 (d) : 두 클라이언트가 접속한 후의 상태를 나타낸 것이다. 서버측에는 총 세 개의 소켓이 존재하며, 이 중 두 소켓이 실제 클라이언트와 통신하는 용도로 사용된다.

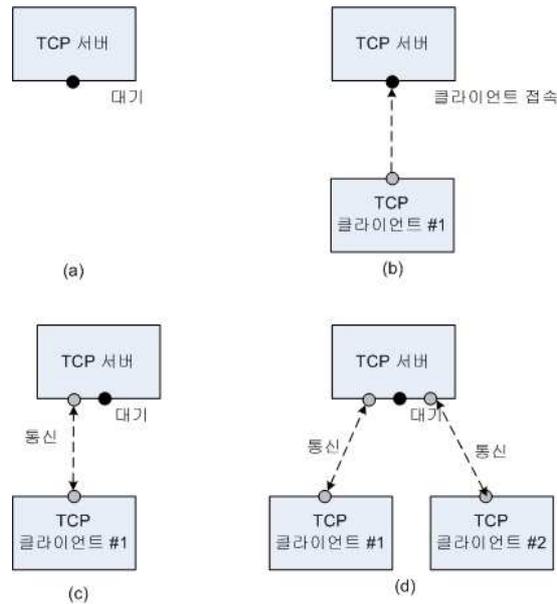


그림 3-6. TCP의 동작 원리

TCP 서버/클라이언트가 통신 하는 상황을 일반적인 형태로 나타내면 그림 3-7과 같다. 서버측 소켓과 클라이언트 측 소켓이 1대 1로 대응하는 것을 알 수 있다. 'TCP 클라이언트#n'의 경우처럼 한 클라이언트가 두개 이상의 소켓을 사용하여 서버에 접속하는 것도 가능하다.

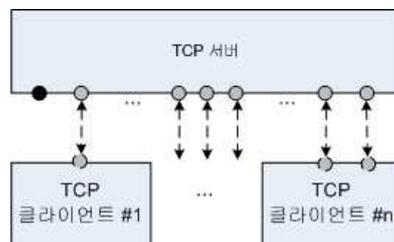


그림 3-7. TCP 서버/클라이언트 통신 형태

3.3 UDP

3.3.1 UDP 서버/클라이언트 구조

UDP는 비연결 지향성을 가지는 비신뢰성 데이터 전송 프로토콜이다. UDP가 비신뢰성 소켓임에도 실시간 데이터 전송에 쓰이는 이유는 TCP가 해주던 흐름 제어(flow control)를 UDP는 하지 않기 때문에 TCP에 비해 빠르게 전송할 수 있어 데이터 지연이 없기 때문이다. 멀티미디어 데이터는 특성상 일부가 손실이 되어도 크게 문제가 되지 않는다. 실시간 서비스에서 보다 중요한 것은 전송 시간이다. 이러한 이유 때문에 UDP가 실시간 데이터 전송에 쓰인다. UDP의 특성은 아래와 같다.

1) 전송되는 순서에 상관없이 가장 빠른 전송을 지향한다.

전송하는 순서가 크게 차이가 없는 한 각 데이터그램은 가장 빠른 길을 택하여 전송하므로 어떤 데이터그램이 먼저 도착할지 예측 불가능하다.

2) 전송되는 데이터는 손실될 수도 있고 에러가 발생할 수도 있다.

3) 전송되는 데이터의 경계(Boundary)가 존재 한다.

데이터를 전송하는 호스트가 세 번의 함수 호출을 통해서 데이터 전송했다면, 수신하는 호스트도 반드시 세 번의 함수 호출을 거쳐야 데이터를 완전히 수신할 수 있게 된다.

4) 한번에 전송되는 데이터의 크기는 제한된다.

전송되는 데이터의 클 경우 데이터를 나누어서 보내야한다.

UDP의 서버/클라이언트의 동작 방식은 그림 3-8과 같다.

① 서버는 소켓을 생성하여 주소를 바인드 하고 클라이언트의 요구사항이 있을 때까지 기다린다(recvfrom).

② 클라이언트는 소켓을 생성하고 서버로 자신의 요구 사항을 전송한다(sendto).

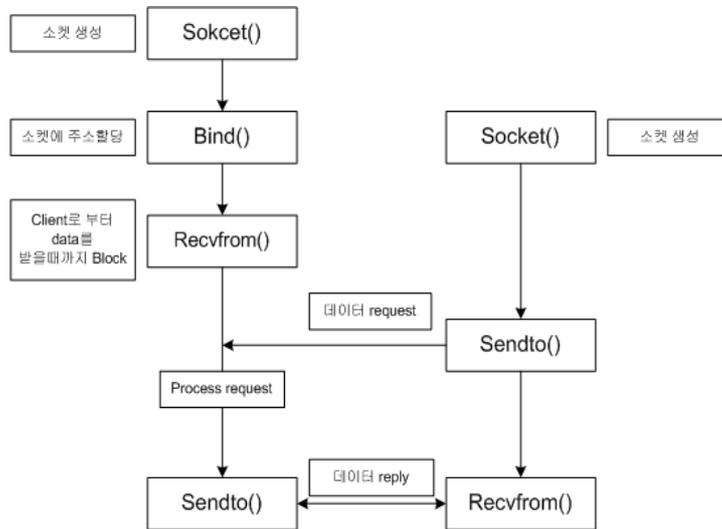


그림 3-8. UDP 비연결 지향형 소켓 시스템 호출

3.3.2 동작 원리

UDP 서버는 TCP와 달리 하나의 소켓으로 다수의 클라이언트에 대한 서비스를 할 수 있다. 그림 3-13은 UDP 서버와 UDP 클라이언트가 통신을 수행하는 원리를 보여준다.

- 그림 3-9의 (a) : 서버는 소켓을 생성한 후 클라이언트가 데이터를 보내기를 기다린다. 이때 서버가 사용하는 소켓은 특정 포트 번호(예를 들면 9000)와 결합(bind) 있어서 이 포트번호로 보내는 데이터만 수신할 수 있다.
- 그림 3-9의 (b) : 클라이언트는 접속 과정 없이 곧바로 서버와 데이터를 주고 받는다.(sendto, recvfrom)
- 그림 3-9의 (c) : 또 다른 클라이언트 역시 접속 과정 없이 곧바로 서버와 데이터를 주고 받는다(sendto, recvfrom)
- 그림 3-9의 (d) : UDP 서버/클라이언트가 통신하는 상황을 일반적인 형태로 나타낸 것이다. 'UDP 클라이언트 #n'의 경우처럼 한 클라이언트가 두개 이상의 소켓을 사용하여 서버와 통신하는 것도 가능하다.

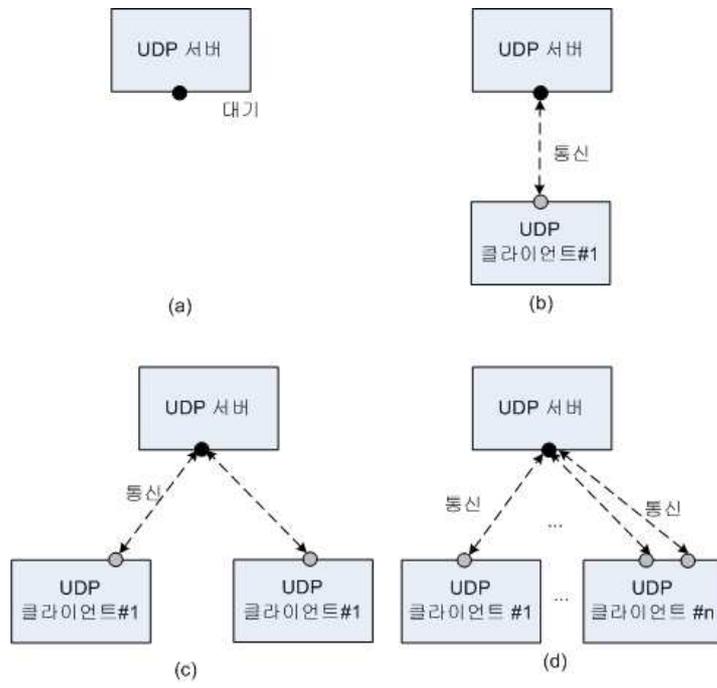


그림 3-9. UDP의 동작원리

제 4장 유무선 통신 다중 기반 멀티미디어 처리 기술

4.1 Filter 구현

4.1.1 Tee Multi Render Filter

시스템에서 고화질 비디오 데이터 및 화상회의 음성, 영상 데이터를 네트워크로 전송하려면 애플리케이션이 고화질 카메라와 화상회의 카메라, 마이크로부터 데이터를 획득할 수 있어야 한다. 기존의 제공 되어지는 다이렉트쇼의 필터그래프를 통해 멀티미디어 디바이스로부터 영상이나 오디오 데이터를 렌더링 할 수 있지만 애플리케이션에 데이터를 올려주는 역할의 필터는 직접 구현하여야 한다.

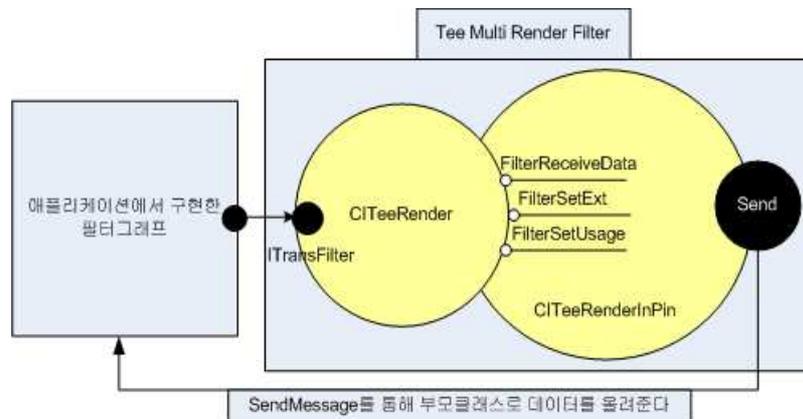


그림 4-1. ‘Tee Multi Render Filter’필터의 개념도

그림 4-1과같이 애플리케이션에서는 그림 4-2의 ITransFilter의 인터페이스를 통해(CITeeRender::FilterReceiveData | CITeeRender::FilterSetExt | CITeeRender::FilterReceiveData) 함수를 접근할 수 있다. 필터그래프를 구성할

때 "Tee Multi Render Filter"필터를 추가하고 부모클래스의 핸들을 FilterSetExt 함수를 통해 셋팅하면 멀티미디어 디바이스의 소스필터로부터 ::SendMessage()를 통해 스트림을 부모클래스가 획득할 수 있다.

```

DECLARE_INTERFACE_(ITransFilter, IUnknown)
{
    STDMETHOD(FilterReceiveData) (BYTE * pBuffer, int nLength) PURE;
    STDMETHOD(FilterSetExt) (HWND hWnd) PURE;
    STDMETHOD(FilterSetUsage) (int nUsage) PURE;
};
    
```

그림 4-2. 'ITransFilter' 인터페이스의 순수가상함수



그림 4-3. 'Tee Multi Render Filter'필터를 사용한 필터 그래프 구성

그림 4-3과같이 필터그래프가 구성되면 부모클래스는 데이터에 대한 MediaType 과 Allocator에 대한 정보를 수신단에게 전송하여야한다. 이 정보는 수신단에서 필터그래프를 구성하고 렌더링 하기 위한 정보이며, 렌더링 하기 전에 반드시 셋팅 되어있어야 한다.

"Tee Multi Render Filter"필터는 그림 4-4와 같이 부모클래스로 데이터를 올려 주기 전에 데이터 종류에 대한 정보를 버퍼 헤더에 삽입한다.



그림 4-4. 'Tee Multi Render Filter' 필터의 데이터 종류에 대한 헤더 구성

4.1.2 Tee Multi Source Filter

송신단으로부터 전송된 비디오, 오디오 데이터는 수신단에서 "Tee Muti Source Filter"를 추가한 필터그래프를 구성하고 "Tee Muti Source Filter"에서 스트리밍 되어 화면에 렌더링 되거나 스피커로 재생된다. "Tee Multi Render Filter"와 반대

로 수신단의 부모 애플리케이션이 네트워크를 통해 송신단의 멀티미디어 데이터를 수신하고 구성된 필터그래프로 데이터를 내려주어야 한다. 이 때 사용되는 필터가 “Tee Multi Source Filter”이다.

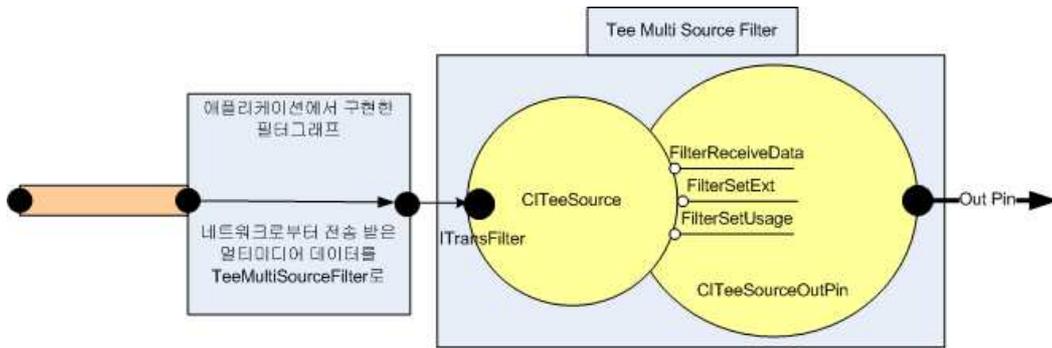


그림 4-5. ‘Tee Multi Source Filter’필터의 개념도



그림 4-6. ‘Tee Multi Source Filter’필터로 구성된 필터그래프

부모 애플리케이션은 수신된 데이터의 헤더의 4byte를 조사해서 데이터의 종류를 확인한 후 수신단이 구성한 각각의 필터그래프로 데이터를 내려 준다. 그림 4-7은 수신된 데이터에 따른 함수를 호출하는 예제 코드이다.

```

데이터 타입 = 버퍼의 헤더 4byte
switch(데이터 타입)
{
    HV_VIDEO: AO:
        break;
    CONF_VIDEO: BO:
        break;
    CONF_AUDIO: CO:
        break;
}
    
```

그림 4-7. 수신된 데이터의 타입에 따른 함수 호출의 예

4.2 Setting

4.2.1 Setting

환자측 시스템에서 멀티미디어 데이터를 전송하기 위해 디바이스 선택 및 코덱 선택, 제어 서버의 IP저장, Patient Monitor와의 COM 시리얼 포트 설정등의 셋팅 프로시저를 수행해야한다. 그림 4-8은 환자측 시스템의 애플리케이션에서 셋팅을 위한 다이얼로그를 나타낸 것이다.

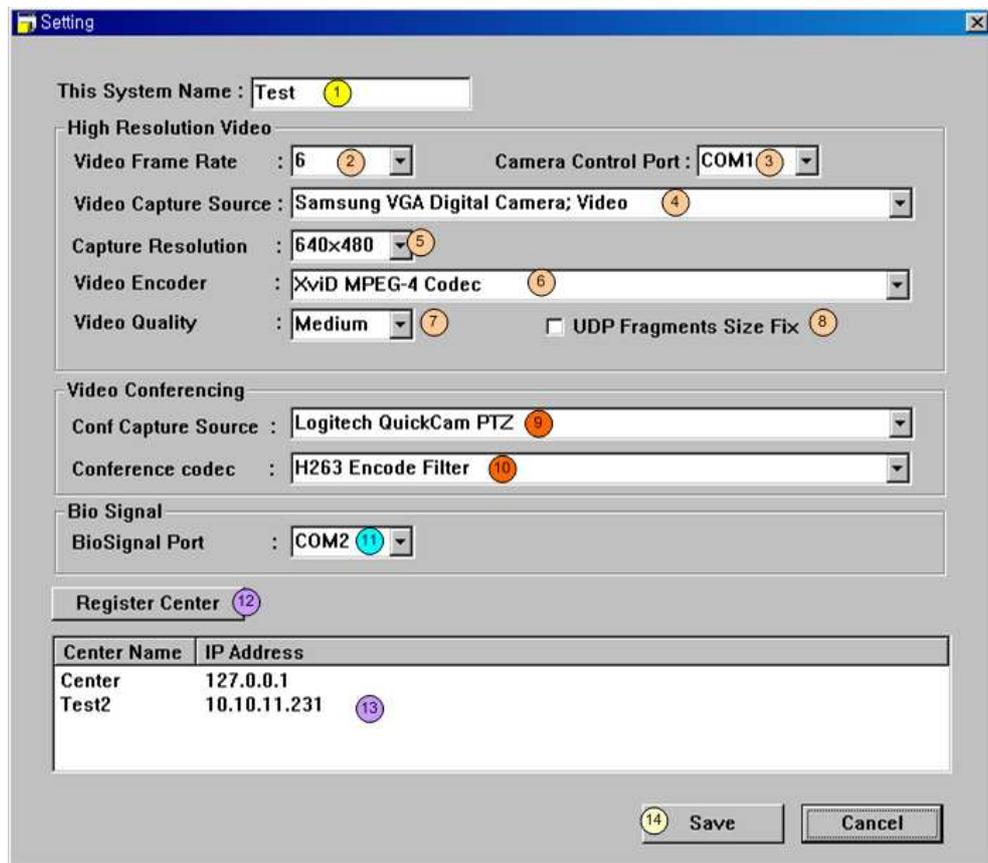


그림 4-8. 환자측 시스템의 Setting Dialog Interface

다음은 각 컴포넌트의 세부사항이다.

- ① 시스템 이름 : 환자측 시스템의 애플리케이션 이름을 나타낸다.

- ②~⑧은 고화질 비디오에 관한 컴포넌트 셋팅을 나타낸다
- ② 프레임율: 초기의 고화질 비디오의 프레임 레이트 값을 설정한다.
- ③ 고화질 카메라 포트 : 고화질 카메라를 컨트롤 하기 위한 시리얼 포트 설정
- ④ 비디오 캡처 소스 : 고화질 비디오 필터그래프를 구성하기 위한 고화질 비디오 캡처 소스 선택 리스트
- ⑤ Resolution : 고화질 비디오 캡처 소스가 제공하는 Resolution 선택 리스트
- ⑥ 코덱 : 고화질 비디오에 대한 인코딩 코덱 선택 리스트
- ⑦ 비디오 Quality : MPEG-2 비디오 코덱 선택시 제공되는 인터페이스를 통해 비디오 영상 Quality를 선택할 수 있다. (“Very low”, “Low”, “Medium”, “High”, “Very High”)
- ⑧ UDP 프래그먼트 사이즈를 고정할 것인가 : 고화질 비디오의 영상을 네트워크로 전송하기 위한 UDP 프래그먼트 사이즈를 고정할 것인가 아니면 프레임율에 따라 Flexible하게 자동적으로 변화 시킬 것인가에 대한 선택적 인터페이스.

- ⑨,⑩은 화상회의를 위한 컴포넌트 셋팅을 나타낸다.
- ⑨ 비디오 캡처 소스 : 화상회의 비디오 필터그래프를 구성하기 위한 화상회의용 카메라 비디오 캡처 소스 선택 리스트
- ⑩ 코덱 : 화상회의 비디오에 대한 인코딩 코덱 선택 리스트(H.261, H.263)

- ⑪ 생체신호를 얻기 위한 포트 : Patient Monitor와 RS232C를 통한 통신 시리얼 포트 설정

- ⑫,⑬ 제어 서버의 IP를 셋팅하기 위한 컴포넌트

⑫ 그림 4-9와 같은 다이얼로그를 생성함으로써 제어 서버의 IP를 셋팅한다.

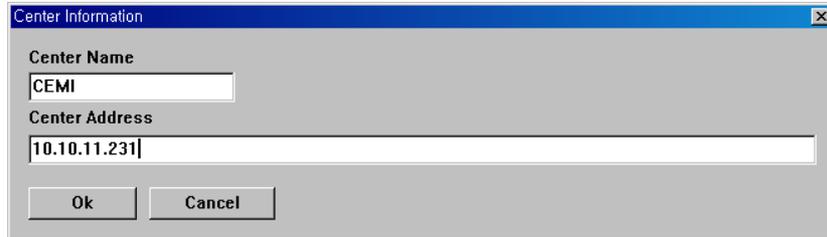


그림 4-9. 환자측 시스템의 Center Information Dialog

제어 서버의 이름을 저장하는 "Center Name"과 IP주소값을 저장하는 "Center Address" 필드가 있다.

⑬ 기존에 저장해 놓은 제어 서버의 IP를 나타낸다.

⑭ ①~⑬의 선택된 사항에 대한 정보를 시스템 레지스트리에 저장한다.

보통 프로그램을 실행할 때 사용자가 입력한 옵션 설정이나 프로그램 스스로 만들어낸 정보들을 다음 실행을 위해 저장해 두어야 하며 이런 정보를 INI파일을 이용하여 프로그램의 설정 상태를 저장한다. 그러나 INI파일은 다음과 같은 단점을 갖기 때문에 문제가 될 수 있다.

(a) INI 파일은 텍스트 파일 포맷으로 저장되므로 사용자가 임의로 조작할 수 있다.

(b) 프로그램당 하나 이상의 INI 파일을 생성시키므로 하드 디스크에는 무수한 INI 파일들로 꽉 차게 되고 결과적으로 하드 디스크 공간을 낭비하게 된다.

(c) INI 파일이 손상되었을 경우에 복구할 수 있는 방법이 없다.

(d) INI 파일은 실제 디스크상에 존재하는 텍스트 파일이므로 정보를 읽고 쓸 때마다 하드디스크를 액세스해야 한다.

그러므로 이러한 단점을 극복한 대안으로 레지스트리(Registry)를 사용한다.

레지스트리의 디렉토리 구조와 유사한 계층적 구조로 구성되어 있다. 루트로부터 복수 개의 가지가 갈라져 나오는 트리 구조이며 트리의 각 노드를 키(Key)라고 한다. 키 안에 또 다른 서브키가 포함될 수 있으며 서브키 안에 또다른 서브키가 포함될 수 있다. 레지스트리의 키는 도스의 디렉토리에 비유될 수 있다. 키는 다음의 다섯 가지 구성요소로 이루어져 있다.

요소	설명
이름	키의 이름을 지정하는 문자열이다. 숫자, 영문자 등을 사용할 수 있으며 백슬래시 문자는 사용할 수 없다. 최대 길이는 512문자(유니코드인 경우 256문자)이다. 디렉토리나 마찬가지로 같은 레벨에서는 유일한 이름을 가져야 한다. 즉 한 부모키에 속한 서브키가 같은 이름을 가질 수 없다.
클래스	오브젝트 클래스의 이름이다.
보안 속성	NT의 보안 속성과 동일한 표준 보안 속성이다.
액세스 시간	키가 최후로 수정된 시간이다
값	키안에 포함되는 정보이며 여러 개가 존재할 수도 있고 없을 수도 있다.

표 4-1. 레지스트리 키의 다섯 가지 구성 요소

다음은 레지스트리에 저장하기 위해 구현된 Function이다.

// 레지스트리로부터 정수값을 읽는다.

```
UINT SHRegReadInt(HKEY hKey, LPCTSTR lpKey, LPCTSTR lpValue, INT nDefault)
```

// 레지스트리로부터 문자열을 읽는다.

```
BOOL SHRegReadString(HKEY hKey, LPCTSTR lpKey, LPCTSTR lpValue, LPCTSTR lpDefault, LPTSTR lpRet, DWORD nSize)
```

// 레지스트리로부터 정수값을 쓴다.

```

BOOL SHRegWriteInt(HKEY hKey, LPCTSTR lpKey, LPCTSTR lpValue,
UINT nData)

```

// 레지스트리로부터 문자열을 쓴다.

```

BOOL SHRegWriteString(HKEY hKey, LPCTSTR lpKey, LPCTSTR lpValue,
LPCTSTR lpData)

```

그림 4-10은 환자측 시스템에서 레지스트리에 저장된 셋팅된 값들이다. 셋팅된 필드의 값으로 저장된 것을 알 수 있다. 레지스트리에 저장된 데이터는 고화질 비디오나 화상회의 비디오, 오디오의 필터그래프를 구성하거나 환자 모니터로부터 생체신호를 전송받는 시리얼 통신을 위해 애플리케이션이 구동될 때 다시 읽어 드리게 된다. 또한 입력된 데이터는 추후 셋팅 다이얼로그를 띄울 때 다시 로딩되어 사용자가 다시 각 컴포넌트를 셋팅 할 필요가 없다.

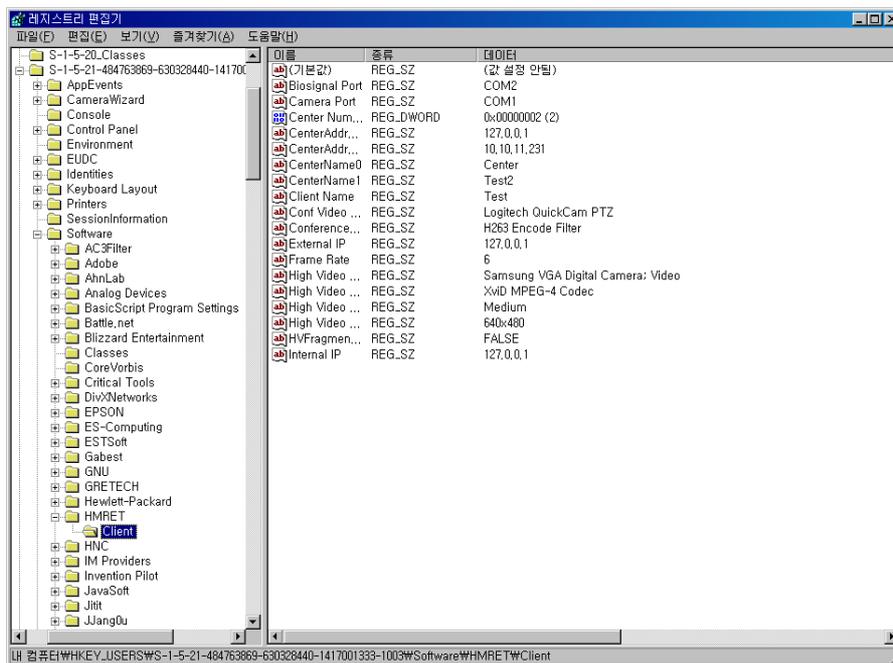


그림 4-10. Setting Dialog로부터 레지스트리에 저장된 환자측 정보

4.3 고품질 영상 (High Quality Video)

4.3.1 구성

원격의 의사가 환자의 상태를 진단하기 위해 환자측 시스템은 고품질 카메라를 통해 High Quality 영상을 획득한 후 네트워크를 통해 스트림 서버나 의사측 시스템으로 전송한다. 고품질 카메라를 통해 얻은 Analog 신호는 Digital Converter를 사용하여 Digital 신호로 변환을 하고 IEEE 1394 인터페이스를 통해 환자측 시스템의 애플리케이션으로 전송된다. 환자측 시스템의 애플리케이션은 다이렉트쇼의 "Video Capture Source" 카테고리에 인식된 고품질 카메라필터와 네트워크의 변화에 따른 인코딩 파라미터를 갖는 MPEG-4계열 인코딩 필터, 환자측 시스템에서 모니터링 하는 "Video Render" 필터, 데이터를 애플리케이션에 올려주는 "Tee Multi Render Filter"필터로 구성된 필터그래프를 작성하여 네트워크를 통해 스트림서버와 의사측 시스템으로 전송하여 준다. 스트림 서버와 의사측 시스템은 "Tee Multi Source Filter"를 통해 환자측 시스템으로부터 전송된 데이터를 애플리케이션으로 올려 받게 되고 MPEG-4 계열의 디코더 필터와 "Video Render" 필터로 필터그래프를 구성하여 영상을 디코딩한 후 모니터링 하게 된다. 이 때, 환자측 시스템으로부터 전송된 고품질 영상을 다시 렌더링하기 위해서는 환자측 시스템의 필터그래프에서 고품질 영상의 미디어 협상 및 버퍼 협상을 통해 얻은 MediaType과 Allocator를 미리 전송받아 렌더링 하기 전에 "Tee Multi Source Filter"의 ITransFilter 인터페이스를 통해 Setting되어야 한다.

4.3.2 송신단

환자측 시스템은 고품질 카메라를 통해 환자의 상태를 의사측 시스템이나 스트림서버 혹은 제어 서버로 전송한다. 본 시스템에서 사용한 환자측 고품질 카메라는 캐논의 VC-C4 고품질 카메라로써, 프레임율에 따른 raw 데이터 사이즈는 표

4-2와 같다. 고화질 카메라와 A/D컨버터를 연결하고 A/D컨버터의 IEEE 1394와 환자측 시스템을 연결하여 고화질 영상을 획득할 수 있다. 고화질 카메라 연결이 완료 되면 다이렉트 쇼 필터의 “Video Capture Source” 디렉토리에 고화질 카메라를 확인할 수 있다. 그림 4-12는 실제 환자 시스템에서 고화질 카메라와 A/D 컨버터를 연결한 모습이다.



그림 4-11. 고화질 카메라 캐논의 VC-C4 사양



그림 4-12. 환자측 시스템에서 고화질 카메라와 Analog to Digital 컨버터 연결

	1frame	5frames	10frames	15frames	30frames
720*480	5.27Mbps	26.3Mbps	52.73Mbps	71.32Mbps	158.19Mbps
640*480	4.68Mbps	22.6Mbps	43.24Mbps	67.12Mbps	142.35Mbps

표 4-2. 고화질 카메라의 프레임당 Raw 데이터 크기

표 4-3은 표 4-2의 720 * 480의 고화질 데이터를 MPEG4 계열의 인코더 코덱으

로 압축하였을 때의 데이터 사이즈이다. 즉 30frame으로 설정되었을 때 1초 데이터는 MPEG4로 압축했을시에 약 1.67Mbps의 데이터가 발생함으로 이를 네트워크로 실시간으로 전송되어진다.

	1 frame	5 frame	10frame	15frame	30frame
XviD MPEG-4 Codec	0.055 Mbps	0.25 Mbps	0.53 Mbps	0.883 Mbps	1.671 Mbps
DivX MPEG-4 Fast-Motion	0.66 Mbps	0.312 Mbps	0.612 Mbps	0.92Mbps	1.722 Mbps
DivX MPEG-4 Low-Motion	0.65 Mbps	0.35 Mbps	0.653 Mbps	0.901 Mbps	1.73 Mbps

표 4-3. MPEG4 계열의 코덱에 따른 압축률(720*480)

그림 4-13는 송신단의 고화질 비디오 영상을 얻기 위한 필터 그래프의 구성을 나타낸다. 그림 4-8의 4항목에서 선택한 고화질 카메라의 'Video Capture Source' Out pin과 2항목의 프레임율을 셋팅하기 위한 필터를 연결하고 환자측 시스템에서 캡처한 영상을 보기 위해 'Video Render' 필터, 그리고 애플리케이션에 영상 데이터 및 Media Type, Allocator 정보를 올리기 위한 'Tee Multi Render Filter' 필터의 두 필터를 연결하기 위해 'SmartTee' 필터를 필터그래프에 추가한다. 그림 4-8의 6항목에서 선택한 MPEG-4계열의 비디오 압축 필터를 필터 그래프에 추가하여 비디오 영상을 'Tee Multi Render Filter'로 올리기 전에 압축을 하게 된다. 'FrameRate Control' 필터 그래프의 인터페이스는 그림 4-14와 같다.

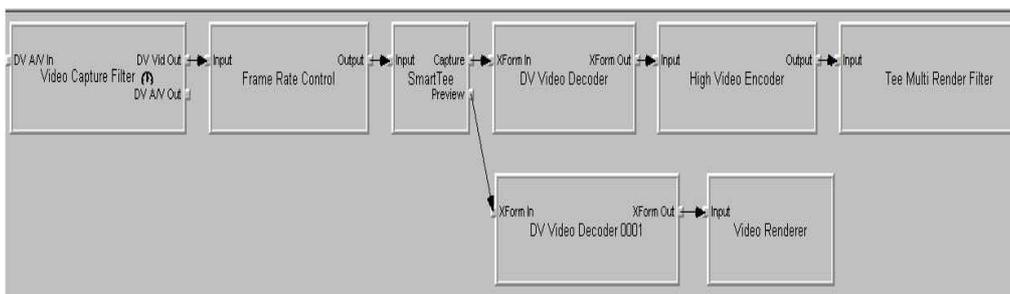


그림 4-13. 송신단에서의 고화질 비디오 필터 구성

```

DECLARE_INTERFACE_(IFrameRate, IUnknown)
{
    STDMETHOD(SetFrame)(long IFrame)PURE;
};

```

그림 4-14. 'FrameRate Control' 필터의 인터페이스

4.3.3 수신단

환자측 시스템의 송신단으로부터 전송 받은 고화질 영상 데이터를 렌더링 하기 위해서 스트림 서버 및 의사 측 시스템의 수신단은 송신단의 비디오 영상에 대한 'Media Type'과 'Allocator' 정보를 가지고 있어야한다. 그림 4-15는 수신단에서 구성한 필터 그래프의 모습이다.

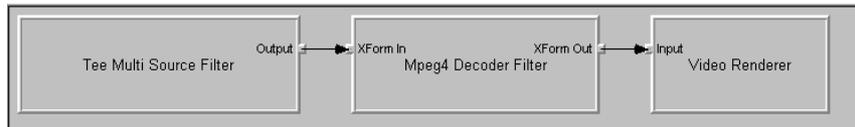


그림 4-15. 수신단에서의 고화질 비디오 필터 구성

'Tee Multi Source Filter'의 ITransFilter 인터페이스를 통해 'FilterReceiveData'의 파라미터 값으로 송신단의 'MediaType'과 'Allocator' 정보를 그림 4-16처럼 셋팅한다.

```

piTf->FilterReceiveData((BYTE *)pParentDlg->m_pMediaTypeVideo, 4 + sizeof(MY_MEDIA_TYPE) +
((MY_MEDIA_TYPE*)(pParentDlg->m_pMediaTypeVideo + 4))->cbFormat);
piTf->FilterReceiveData((BYTE *)pParentDlg->m_pAllocatorVideo, 4 + sizeof(ALLOCATOR_PROPERTIES));

```

그림 4-16. 송신단으로부터 전송받은 MediaType과 Allocator 정보 셋팅

이렇게 전송 받은 미디어 타입과 할당자 정보를 설정하고서 실제 송신단으로부터 전송받는 영상데이터를 'Tee Multi Source Filter' 필터로 올려주게 되고, 데이터는 MPEG-4 계열의 디코더 필터를 통해 복원되어져 수신단 화면에 렌더링 되게 된다. 그림 4-17은 송신단과 수신단의 고화질 비디오 영상을 전송하는 메커니즘을 시퀀스 다이어그램으로 나타낸 그림이다.

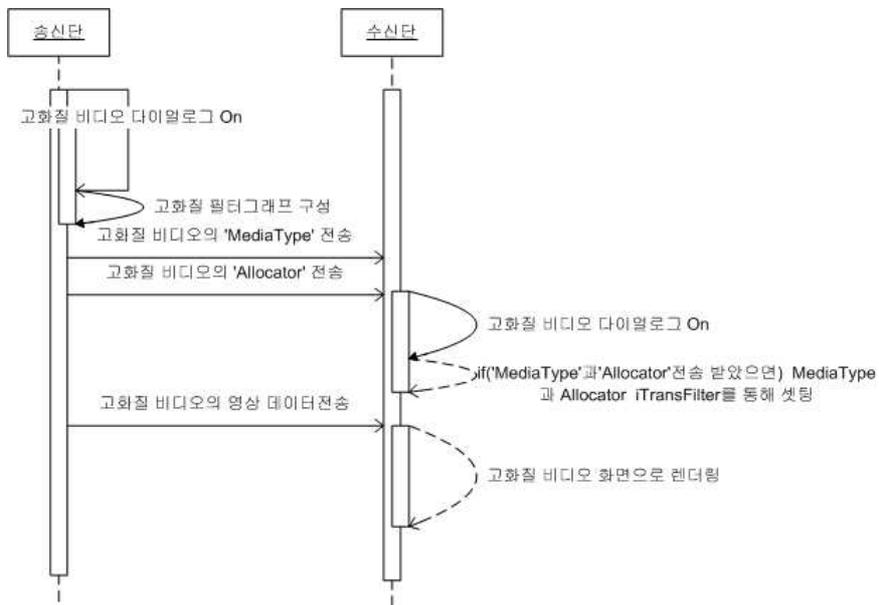


그림 4-17. 고화질 데이터 전송 메커니즘의 시퀀스 다이어그램

4.4 화상회의 음성 및 영상

4.4.1 구성

원격진료에 있어서 환자측 의사와 응급 센터에 상주하는 응급의 그리고 의사측 시스템의 전문의 간에 의사 소통을 원활히 하기 위해 화상회의 데이터를 송수신 한다. 화상회의의 비디오 신호는 카메라에서 획득한 영상신호를 지속적으로 전송해 주어야 하기 때문에 많은 데이터 양을 요구하게 된다. 또한 원활한 의사소통을 위해 실시간에 가까운 영상 및 음성신호가 전송되어야 하며, 두 신호의 동기화 이루어 져야 한다. ITU에서 다른 기종간의 호환과 원활한 기능보장을 위해 H.320을 화상회의의 표준안으로 삼고 있다. H.320은 영상의 경우 H.261을, 음성의 경우 G.711, G.722, G.728의 세 가지를 표준으로 한다. 화상회의 카메라를 USB 2.0 인터페이스를 통해 연결되고, 헤드셋을 연결하여 음성 데이터를 획득한다. 연결된 디

바이스는 필터그래프 매니저에서 확인할 수 있다. 고화질 영상과 같이 화상회의 비디오, 오디오 영상에 대한 미디어 타입과 할당자 정보는 수신단에서 미리 설정되어 있어야 한다.

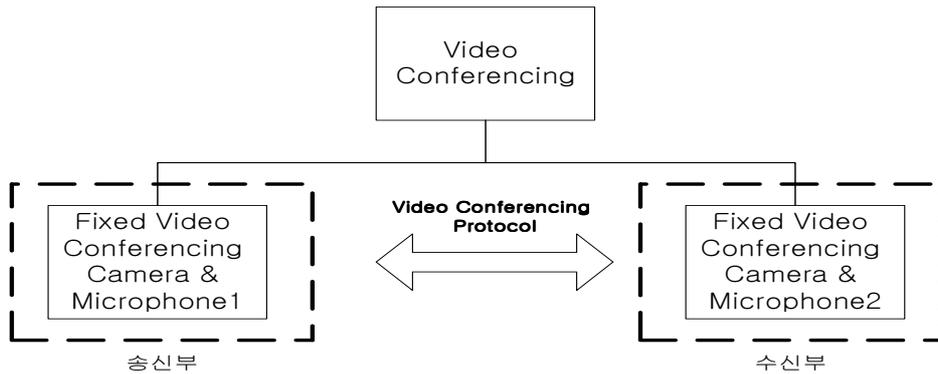


그림 4-18. 응급진료용 화상회의 영상/음성 송·수신 구조

데이터 종류	접속방식	데이터 양	우선순위	실시간	기 타
영상 데이터	USB	98Kbps	낮 음	실시간	H.261 Protocol, 160×120이상 10frame/sec, color
음성 데이터	PCI	32Kbps	높 음	실시간	G.711 Protocol

표 4-4. Data 종류에 따른 접속방식, 데이터 양, 우선순위

4.4.2 송신단

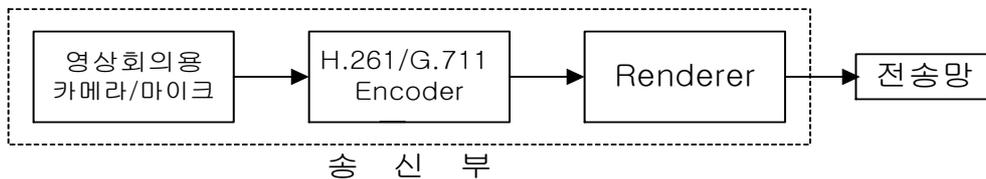


그림 4-19. 송신단의 화상회의 데이터 흐름도

Encoder에서 변환된 stream data는 buffer를 통하여 다음 단계인 Renderer로 보내어진다. 이때 buffer는 feedback되어 Rate control을 통하여 다시 Encoder로 유입되고 buffer size에 맞게 stream의 quantization step size를 결정해 보내진다.

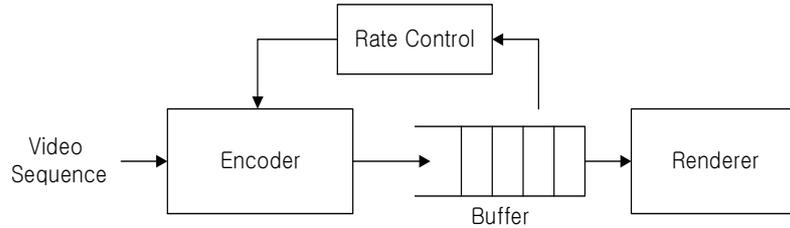


그림 4-20. 송신단의 화상회의 데이터 버퍼 컨트롤

그림 4-21은 송신단에서 구성한 화상회의 영상데이터를 위한 필터 그래프 구성이며, 그림 4-22는 음성데이터를 위한 필터 그래프의 구성이다. 영상데이터는 H.261, H.263의 인코더를 통해 압축되어지고, 압축된 데이터는 'iTee Multi Render Filter'를 통해 애플리케이션에 올려진다. 음성데이터 역시 G.711의 인코더 필터를 통해 압축되어진다. 화상회의 영상 및 음성 데이터를 받은 애플리케이션은 UDP 네트워크 세션을 통해 수신단으로 데이터를 전송한다. 물론 화상회의 데이터에 대한 미디어 타입 정보와 할당자 정보는 미리 수신단에서 셋팅되어야 한다.

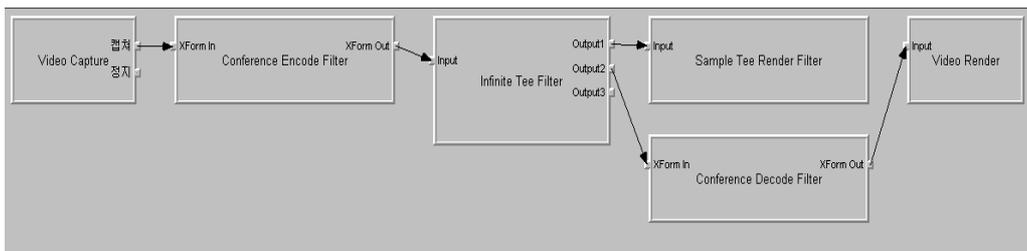


그림 4-21. 송신단의 화상회의 영상 데이터 필터그래프 구성

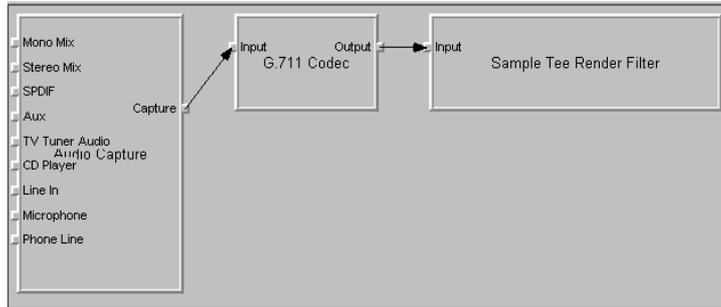


그림 4-22. 송신단의 화상회의 음성 데이터 필터그래프 구성

4.4.3 수신단

시스템 구성원 즉, 환자 시스템, 스트림 서버, 의사측 시스템은 자신의 화상회의 영상 및 음성 데이터를 렌더링 하며, 다른 구성원으로부터 전송되어진 화상회의 데이터를 렌더링하기 위한 준비를 해야 한다. 수신단은 송신단으로부터 미리 전송받은 미디어 타입 정보와 할당자 정보를 셋팅하여 초기화함으로써 렌더링 준비를 한다.

그림 4-23, 4-24는 수신단에서의 화상회의 영상 및 음성 데이터를 위한 필터 그래프 구성이다.

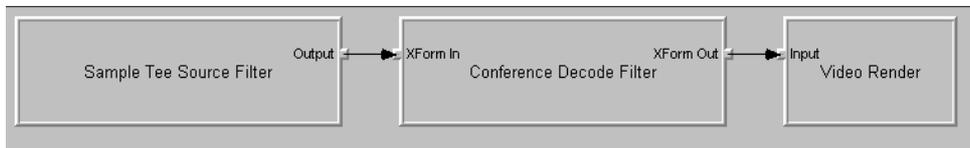


그림 4-23. 수신단의 화상회의 영상 데이터 필터그래프 구성

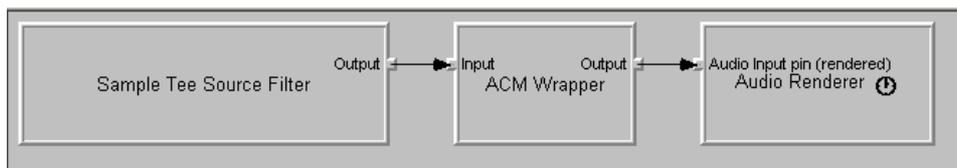


그림 4-24. 수신단의 화상회의 음성 데이터 필터그래프 구성

송신단으로부터 전송된 화상회의 데이터는 Decoder Filter를 복원된 후 영상데이터는 화면에 렌더링 되고, 음성 데이터는 시스템의 스피커를 통해 들리게 된다.

4.5 응급 환자의 생체신호

4.5.1 생체신호 데이터 처리 기술

응급환자의 생체신호는 환자를 진단하기 위한 중요한 데이터이다. 환자측 시스템은 환자의 ECG, SpO2, Resp, IBP등의 생체신호 데이터를 의사측 시스템과 통신함으로써 환자에 대한 진단을 효율적으로 수행한다. 원격 진료 시스템에서 환자측 시스템과 Patient monitor간에 interface 하기 위해 환자측 시스템의 RS-232C를 이용한 시리얼 통신을 한다. 애플리케이션에서 비동기적으로 생체 신호 데이터를 처리하기 위해 스레드를 사용한다. 즉, Thread monitoring으로 Serial port에 입력되는 각각의 생체신호 획득하고, 시스템의 애플리케이션에서 네트워크 세션을 통해 데이터를 전송한다. 그림 4-25는 Patient Monitor에서 시리얼로 입력되는 데이터의 포맷을 나타낸다.

1초당 DATA(WAVE+PARAMETER) (1140 bytes : [0]~[1139])				
WAVE DATA (1050 bytes : [0]~[1049])				PARAMETERS (90 bytes : [1050]~[1139])
ECG	SpO2	IBP	Resp	
600 bytes	150 bytes	150 bytes	150 bytes	90 bytes

그림 4-25. PC에 입력된 생체신호의 data 구조

4.5.2 효율적인 전송을 위한 생체신호 압축기술

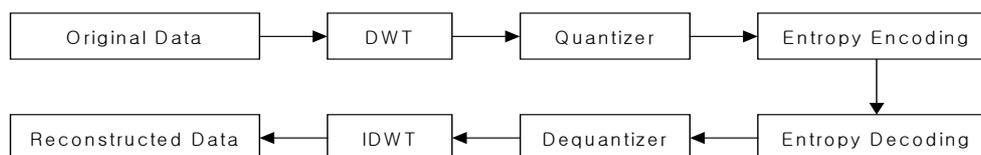


그림 4-26. 생체신호 데이터 압축/복원을 위한 전체 구조도

Wavelet 변환을 통해 입력 신호는 저주파 성분(approximation)과 고주파 성분(detail)으로 나누어지는 과정을 반복하여 서로 다른 주파수 성분을 가지는 몇 개

의 부 대역으로 나누어진다. 이때 나누어진 각 부 대역들은 대역 별로 서로 상관도가 낮으면서 데이터 크기의 분포 수준이 서로 다르게 되고, 이 각 부 대역들에 대하여 양자화를 거쳐 엔트로피 부호화를 하고 중요하지 않은 정보를 담은 대역의 데이터는 무시함으로써 데이터에 대한 압축 효과를 얻을 수 있다. 이때 양자화 결과에 따라 압축의 효과가 달라지게 되는데, 압축율과 복원시 충실도를 고려하여 각 부 대역들에 대한 양자화 수준을 결정하는 것이 중요하다. 생체신호의 경우 낮은 주파수 대역에 대부분의 에너지가 포함되어 있으므로 낮은 주파수 대역의 신호를 부호화할 때는 충분한 수의 양자화 bit를 할당하고 높은 주파수 대역의 신호에는 상대적으로 적은 수의 bit를 할당한다.

① EZW(Embedded Zerotree Wavelet) 압축 기법 구현

EZW 부호화는 원하는 bit rate로 부호화가 가능하고, 부호화된 bit stream을 복호화기에서 처리할 때 어느 순간에서 끊어지더라도 복호화가 가능한 부호화 방식이다. 이러한 특성은 신호나 영상의 점진적 전송(progressive transmission)을 가능케 한다. 이것은 먼저 부모 계수의 크기를 통해 자식 계수의 중요도까지 판별하는 제로 트리 부호화를 수행하여, 다음 중요 계수로 판정된 계수들에 대하여 임베디드 부호화를 수행한다. 이것은 2차원 영상의 효율적인 압축 기술로써 1차원 생체 신호에 대해 적용하여 data의 실시간 처리를 가능케 한다. 주부호화 과정은 제로 트리 부호화에 의한 계수의 중요도 판정을 하는 것이고, 종속 부호화 과정은 주부호화 과정에서 중요 계수로 판명이 난 계수들의 값을 한 비트씩 보내 주는 것이다. 주부호화 과정에서는 각 계수들의 위치에 대한 정보가 포함되고, 종속 부호화 과정에서는 계수들의 크기 정보가 포함되게 된다. 이 부호화 방식은 원하는 비트율에서 부호화가 중지될 수 있고, 복호화도 원하는 비트율까지만 할 수 있는 특징이 있다. 따라서, 데이터 압축에 있어서 비트율을 조정할 수 있는 특징이 있다.

- EZW Encoding 알고리즘

- a. ECG 신호의 저주파 영역에 대한 연속적 wavelet 변환
- b. EZW Tree 구성 및 Scan 순서 결정

c. 입력에 대해 symbol을 구분하는 "Dominant pass(주부호화 과정)"와 code stream을 생성하는 "Subordinate pass(종속 부호화 과정)"의 2단계로 구성

d. 입력의 절대값과 Threshold 비교를 통해 Symbol의 중요도에 대한 정보를 2비트로 표현

e. "Dominant pass"에서는 symbol을 추출하여 전송, "Subordinate pass"에서 "POS"와 "NEG" symbol에 해당하는 노드의 코드를 부여하고 전송

② EZW Decoding 알고리즘

a. 전송된 "Threshold"와 bit stream 추출하면서 symbol tree 검사

b. 추출된 sign을 이용하여 code를 추출하고 원래값의 근사값을 생성

4.5.3 생체신호 디스플레이

RS-232 통신을 통해 들어온 데이터를 wave 데이터와 문자 데이터로 분류하여 화면에 display한다. 입력된 데이터를 임의의 버퍼에 저장하고 환자감시장치의 데이터 포맷에 의해 필요한 데이터들을 분류한 뒤 각각 표현 모듈에 연관된 변수에 저장한다. 그림 4-27은 생체신호 데이터의 데이터에서 문자 데이터(온도, 혈압등의 수치)와 wave 데이터를 분류하여 비트 메모리를 이용한 디스플레이 과정을 나타내고 있다.

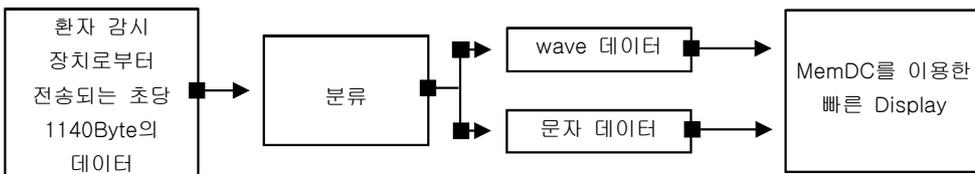


그림 4-27. 생체 신호 데이터의 분류 및 표현

그림 4-28에서 보는 것과 같이 두개의 메모리 맵을 사용하여 데이터를 디스플레이 한다. 즉 DrawDC에는 이전에 입력되어진 데이터를 그려놓은 상태로 왼쪽으로 이동하면서 나타내어지고, BackDC에는 새로 입력되는 데이터를 DrawDC의 마지막 포인터와 연결하여 그려진다. 이 후 다시 새로운 데이터가 입력되면 DrawDC와 BackDC의 데이터를 DrawDC에 Copy하여 DrawDC를 갱신하게 된다. 이와 같

은 방법을 통해 생체신호가 실시간으로 디스플레이된다.

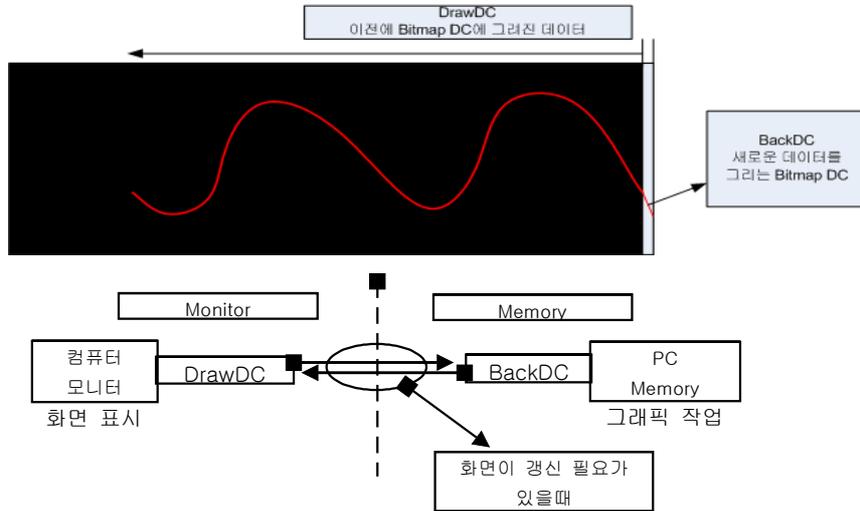


그림 4-28. 생체 신호 표시하기 위한 MemDC를 이용한 화면 표현 구조

그림 4-29는 생체신호 데이터를 송수신하고 디스플레이 하는 과정을 나타내는 시퀀스 다이어그램이다.

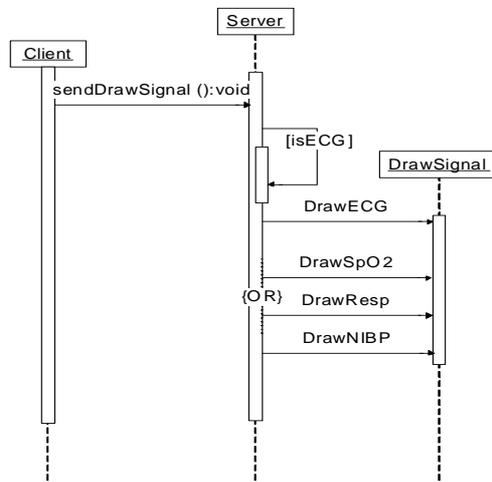


그림 4-29. 생체신호 전송 메커니즘

4.6 응급 환자 이미지 파일

4.6.1 구성

개발된 시스템의 환자측 시스템에서는 응급환자의 상태를 나타내는 X-ray 파일등의 방사선 이미지 파일, JPEG , BMP 파일을 여러 전문의에게 전송한다. 특히 X-ray 이미지는 병원 내의 정보 시스템인 PACS에서 DICOM 표준안에 맞추어 획득하게 되는데, 개발된 시스템은 DICOM 표준에 맞춘 인터페이스를 통해 방사선 영상을 PACS시스템에게 Query , Retrieve를 요구함으로써 획득하게 된다. 이 이미지 파일은 TCP 프로토콜의 세션을 통해 의사측 시스템의 전문의에게 보내지게 된다. 원격의 전문의와 컨설팅 하기 위해 이미지 영상 파일에서의 원격 포인터 Function을 구현하여 환자측 시스템과 상대측 시스템의 마우스 포인터를 매핑하여 전문의간의 서로의 의견을 교환할 수 있는 기능을 구현하였다. PACS 시스템으로부터 전송된 DICOM 영상 파일은 KDicom 라이브러리가 지원하는 여러 함수를 통해 기존의 DICOM 뷰어와 같은 기능을 구현하였다. 전송된 DICOM 파일은 환자의 정보와 시간정보에 따라 각 시스템에 저장되어 추후에 재진이나 확인을 위해 사용되어질 수 있다. 환자측 시스템과의 통신을 시작한 의사측 시스템과 스트림 서버 시스템은 각 시스템에서 전송되어진 DICOM파일을 원격으로 동일하게 모니터링하기 위해 마우스 포인팅뿐만 아니라, 전송된 이미지 파일에 대한 존재 유무를 확인하여야 한다. 만약 환자측 시스템에서 전송한 환자 이미지 파일이 세션에 참여한 스트림 서버는 전송되어져 존재하는데, 세션에 늦게 참여한 의사측 시스템에는 존재하지 않는다면 그에 대한 메시지를 나타내어야 한다.

각 시스템에서 송수신 하는 파일의 정보는 파일 전송에 관여하는 다이얼로그에 나타내어 파일의 크기 및 생성 날짜, 파일명을 확인할 수 있고, 송신한 파일 과 수신한 파일을 구분하여 나타내어진다. 그림 4-30은 네트워크 세션을 통해 각 시스템에 전송된 이미지 파일의 정보를 사용하여 라이브러리를 통해 모니터링 되는 파일 모니터링 클래스를 나타낸다.

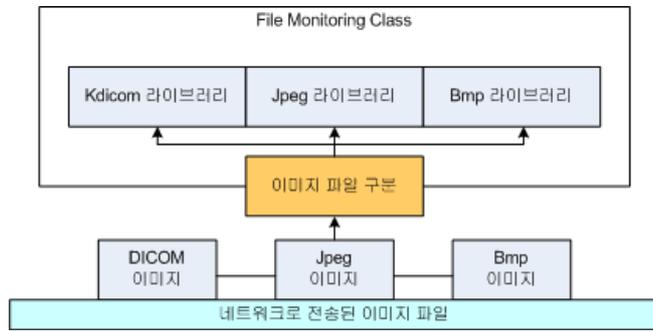


그림 4-30. 시스템내에 전송된 파일의 링크된 라이브러리

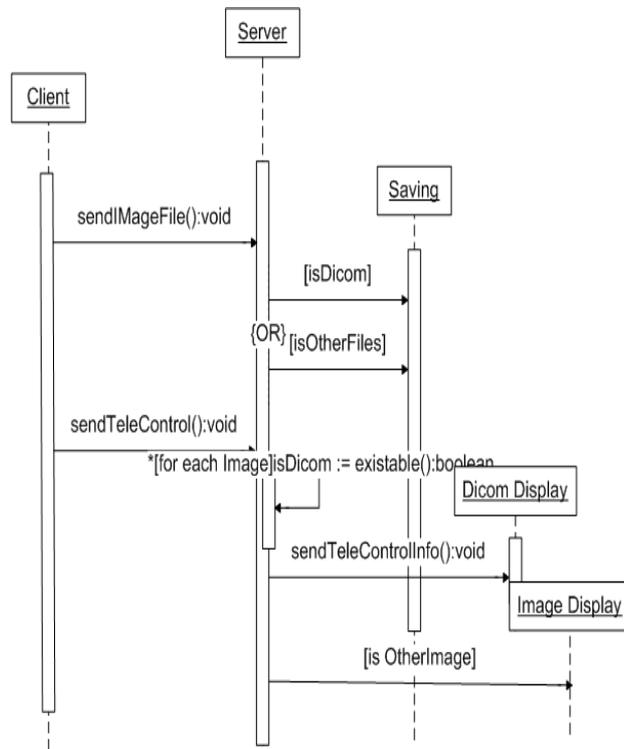


그림 4-31. 환자 이미지 전송 시퀀스 다이어그램(DICOM)

4.6.2 원격 이미지 파일 포인팅

송신단은 환자 이미지 다이얼로그의 마우스 포인터의 위치를 획득한 후 수신단에게 전송하여 디스플레이 되어 있는 환자의 이미지위에 같은 포인터에 위치시

킨다. 그림 4-32는 이미지 다이얼로그에서 마우스가 Move할 때 호출되는 함수이다. 이때, 마우스의 왼쪽 버튼을 누른 상태에서만 포인터 정보를 전송한다.

```

void CFileView::OnMouseMove(UINT nFlags, CPoint point)
{
    // TODO: Add your message handler code here and/or call default
    UINT l,w,m_nLength;
    CString strWinText;
    CDlgImage * pParentDlg = (CDlgImage *)GetParent();
    CDlgFile * pParentDlg = (CDlgFile *)pParentDlg->GetParent();

    if(pGrandParentDlg->bReceive != TRUE){
        if(bLButtonDown == TRUE){ // 왼쪽 마우스가 눌러있는지를 확인
            this->GetParent()->GetWindowText(strWinText);
            m_nLength = strWinText.GetLength();
            BYTE* pBuffer = new BYTE[16 + m_nLength];
            w = FILE_MESSAGE_SHAREDMOUSE; // 전송되는 메시지 헤더
            l = nFlags;
            CopyMemory(pBuffer,&w,4);
            CopyMemory(pBuffer+4,&l,4);
            CopyMemory(pBuffer+8,&point.x,4);
            CopyMemory(pBuffer+12,&point.y,4); // 마우스의 x, y 좌표값
            CopyMemory(pBuffer+16,
                strWinText.GetBuffer(m_nLength),m_nLength);
            ((CDlgImage *)GetParent()->SendBuffer(pBuffer,16+ m_nLength);

        }

        CScrollView::OnMouseMove(nFlags, point);
    }
    else
        pParentDlg->bReceive = FALSE;
}

```

그림 4-32. 송신단에서 마우스 포인터 정보를 전송하기 위한 코드

```

case FILE_MESSAGE_SHAREDMOUSE: // Shared Mouse Control(Still Image Window)
    m_pDlgFile->bReceive = TRUE;
    CopyMemory(&nFlags,&pBuffer[4],4);
    CopyMemory(&ptx,&pBuffer[8],4);
    CopyMemory(&pty,&pBuffer[12],4);
    if(NULL != m_pDlgFile->m_pDlg->pScrollView){
        m_pDlgFile->m_pDlg->pScrollView->pointM.x = ptx;
        m_pDlgFile->m_pDlg->pScrollView->pointM.y = pty;
        ::SendMessage(m_pDlgFile->m_pDlg->pScrollView->m_hWnd,WM_MOUSEPOINT,nFlags,0);
    }
}

```

그림 4-33. 수신단에서 수신한 마우스 포인터정보를 매핑시키는 코드

그림 4-32의 코드로부터 전송된 마우스 포인터에 대한 정보는 그림 4-33의 수신단의 코드에서 매핑 되어진다. 이미지 파일에 대한 정보는 이외에도 그림 4-34의 정보들이 있다. DICOM 파일에 대한 메시지와 파일의 이벤트에 관한 메시지들이 포함된다.

#define FILE_FIRST		73000
#define FILE_MESSAGE_FILENAME		73100
#define FILE_MESSAGE_FILE		73110
#define FILE_MESSAGE_OPENEDFILENAME		73120
#define FILE_MESSAGE_WINDOWSTATE		73130
#define FILE_MESSAGE_SCROLL		73140
#define FILE_MESSAGE_SHAREDMOUSE		73150
#define FILE_MESSAGE_WHEEL		73160
#define FILE_MESSAGE_CURSOR		73170
#define FILE_LAST		73999
#define DICOMVIEWER_FIRST		75000
#define DICOM_MESSAGE_DICOMVIEW_ON	75001	
#define DICOM_MESSAGE_DICOMVIEW_OFF		75002
#define DICOM_MESSAGE_DICOMFILELIST	75003	
#define DICOM_MESSAGE_DICOMVIEWER_OFF	75004	
#define DICOMVIEWER_MY_CONTROL_ON	76001	
#define DICOMVIEWER_MY_CONTROL_OFF	76002	
#define DICOMVIEWER_1BY1		76003
#define DICOMVIEWER_2BY2		76004
#define DICOMVIEWER_2BY3		76005
#define DICOMVIEWER_3BY3		76006
#define DICOMVIEWER_4BY4		76007
#define DICOMVIEWER_OVERAYREARRANGE		76008
#define DICOMVIEWER_OVERAYSHOW		76009
#define DICOMVIEWER_OVERAYHIDE		76010
#define DICOMVIEWER_OVERAYTEXTSHOW	76011	
#define DICOMVIEWER_OVERAYTEXTHIDE	76012	
#define DICOMVIEWER_MMODEANNOTATION		76013
#define DICOMVIEWER_MMODEHISTOGRAM		76014
#define DICOMVIEWER_MMODEMAGNIFY	76015	
#define DICOMVIEWER_MMODEPAN		76016
#define DICOMVIEWER_MMODEPOINTER	76017	
#define DICOMVIEWER_MMODEZOOM		76018
#define DICOMVIEWER_IPFILPV		76019
#define DICOMVIEWER_IPFILPH		76020
#define DICOMVIEWER_IPINVERSE		76021
#define DICOMVIEWER_IPREALSIZE		76022
#define DICOMVIEWER_IPRESET		76023
#define DICOMVIEWER_IPROTATECCW		76024
#define DICOMVIEWER_IPROTATECW		76025
#define DICOMVIEWER_ANNOTATIONDISTANCE	76026	
#define DICOMVIEWER_ANNOTATIONDEGREE	76027	
#define DICOMVIEWER_MOUSE_MOVE		76028
#define DICOMVIEWER_MOUSE_LBUTTONDOWN	76029	
#define DICOMVIEWER_MOUSE_LBUTTONUP		76030
#define DICOMVIEWER_LAST		76031

그림 4-34. 시스템내에서 전송되어지는 파일 메시지 정의

제 5장 유무선 통신 다중 통신 프로토콜 설계

5.1 TCP 전송

5.1.1 TCP 소켓 구현에서 고려해야할 문제

TCP 프로토콜은 가상으로 연결된 상태이기 때문에, 서버와 클라이언트 프로그램에서 주의해야 한다[20]. 또한 1회 전송한 패킷이 한번에 전송되리라는 보장이 없다. 반대로 여러 번에 걸쳐 전송한 패킷이 한 번에 전송될 수도 있다. TCP가 동작하는 방식은 겉으로 나타나지 않는다. 중요한 것은 연결이 된 이후부터 연결을 닫을 때까지 전송한 데이터가 모두 하나의 데이터라는 점이다. 가령 100MB 정도 되는 큰 버퍼를 할당하고 앞에서부터 차례대로 채워나간다고 생각되면 된다. 변수를 읽어 올 때 하나의 변수가 어디서부터 어디까지가 하나의 변수인지 확인이 필요하다. 모든 변수가 동일한 크기를 갖는 것은 아니기 때문에, 잘못 읽어 올 경우 의미가 달라진다. 따라서, TCP에서는 어디서부터 어디까지가 의미있는 하나의 패킷인지 확인하는 작업이 필수이다.

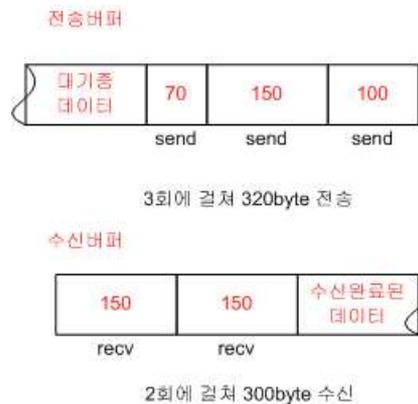


그림 5-1. TCP 패킷 전송과 수신 상태도의 예

그림 5-1에서 윗부분은 전송 소켓 버퍼이고, 아랫부분은 수신 버퍼이다. 둘 다 양

쪽의 버퍼 모양을 보면, 한 쪽 끝이 종료되지 않은 상태임을 알 수 있다. 문제는 전송버퍼와 수신버퍼의 상태가 다르다는 것이다. 전송 소켓이 처음에 100바이트를 전송하였으나 수신 소켓이 데이터를 수신하지 않은 상태이다. 왜냐하면 150바이트를 수신하기 때문에 50바이트가 부족하기 때문이다. 전송 소켓이 두 번째로 150바이트를 전송하면 수신 소켓은 전체 250바이트 중에서 150바이트만 가져온다. 그런데 가져온 150바이트는 하나의 패킷이 아니므로 첫 번째 패킷의 100바이트만 사용하고 50바이트는 버퍼에 뒤야한다. 그 후 전송 소켓에서 세 번째 패킷으로 70바이트를 전송한다. 전체 수신 소켓에는 이전에 있었던 100바이트와 지금 수신한 70바이트가 있다. 수신 소켓으로부터 다시 150바이트를 가져오면 수신 소켓에는 20바이트만 남아 있다. 이제 이전에 남아 있던 50바이트와 수신한 150바이트를 합해서 수신 버퍼에는 200바이트가 있게 된다. 두 번째 패킷을 읽어야 함으로 이중 150바이트를 사용한다. 수신 버퍼에 50바이트가 남게 되고 세 번째 패킷을 처리하기 위해 소켓 버퍼에 남아 있는 20바이트를 읽어 와서 70바이트를 세 번째 패킷을 처리한다. 여기서 중요한 것은 전송 측이나 수신 측 모두 두개의 버퍼를 사용한다는 것이다. 소켓 라이브러리는 전송 또는 수신을 위해 자체적으로 별도의 버퍼를 사용한다. send 함수를 호출했다고 해서 패킷이 실제로 상대 소켓으로 전달되었다고 가정해서는 안 된다. send 함수는 소켓 라이브러리의 버퍼로 데이터를 옮겨놓는 순간 반환한다. 이 소켓 라이브러리 버퍼 때문에, 실제로 한 번에 전송되는 양을 알 수 없다는 것이다. 그림 5-2는 전송 소켓으로부터 두 번째 패킷이 전송된 이후의 소켓 상태를 표현한 그림이다.

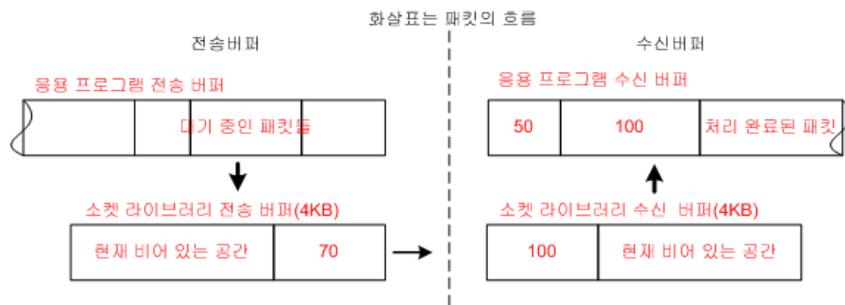


그림 5-2. TCP 패킷 전송과 수신 상태 세부도

5.1.2 문제점을 해결한 TCP 송수신 메커니즘

TCP는 본 논문에서 제안한 시스템에서 환자의 데이터 중 네트워크 환경에서 손실이 없어야 하는 환자의 생체신호, 방사선 이미지등의 데이터 송수신을 위해 소켓 라이브러리 버퍼에 의한 TCP의 특징인 “메시지의 경계가 없다”라는 특성을 해결하여야 하며 이를 위해 다음과 같은 메커니즘을 사용하였다.

- ① 전송하는 데이터의 경계를 나타내기 위해 데이터 사이즈를 나타내는 헤더정보를 추가한다.
- ② 송신단에서 한번의 send를 통해 하나의 패킷을 전송하였는지를 확인한다.
- ③ 수신단은 수신한 데이터가 다음의 두 가지 경우인지를 확인한다.
 - (a) 수신 버퍼로부터 recv한 데이터가 여러 패킷으로 구성되어 있는가?
 - (b) 수신 버퍼로부터 recv한 데이터의 헤더를 분석했을 때 패킷사이즈 정보와 recv한 수신 버퍼의 사이즈가 동일한가?

위와 같은 사항을 고려하여 Socket 클래스를 구성하여야 한다. 우선 송신단의 패킷을 TCP로 전송하기 위해 패킷헤더에 패킷 헤더임을 알리는 Flag값고 실제로 전송하고자 하는 패킷의 사이즈 정보를 추가한다. 그림 5-3은 패킷헤더를 생성하여 추가한 구성도와 코드를 나타낸다.

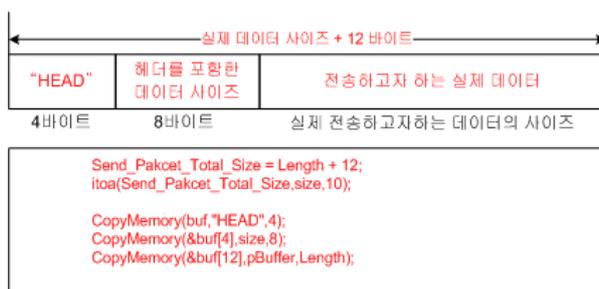


그림 5-3. 헤더 정보를 추가한 TCP 패킷

그림 5-3에서와 같이 실제 전송할 데이터에 헤더정보를 추가한 패킷을 새로 생성한 후 패킷을 모두 전송하기 위해 그림 5-4와 같은 과정을 통해 패킷을 모두 송신한다. send 함수 호출 후 리턴 된 값을 5-3에 의해 생성된 패킷의 사이즈에서

빼고, 패킷의 offset을 리턴된 값만큼 이동한다. 이러한 과정을 패킷의 사이즈만큼 전송할 때 까지 반복적으로 수행하여 모든 패킷의 데이터를 전송한다.

```

while(nLeft > 0) // 남은 패킷의 버퍼가 존재하는 동안
{
    if((Send_Packet_Size = send(clientsocket,offset,nLeft,0)) == SOCKET_ERROR)
    {
        if ( WSAGetLastError() == WSAEWOULDBLOCK )
        {
            QueueSave(offset, nLeft);
            break;
        }
        else
        {
            return false;
        }
    }
    else
    {
        nLeft -= Send_Packet_Size; // send한 패킷의 사이즈를 실제
        // 데이터에서 뺀다
        offset += Send_Packet_Size; // send한 패킷만큼
        // 송신 버퍼의 offset을 이동
    }
}

```

그림 5-4 TCP에서 하나의 패킷을 전송하기 위한 코드 구현

그림 5-4와 같은 과정을 통해 TCP전송을 수행하면 전송하고자 하는 하나의 패킷을 모두 전송할 수 있다. 반면 TCP 데이터 전송을 수신하는 메커니즘은 패킷을 수신하여 패킷의 헤더를 조사함으로써 수행된다. 즉, 패킷의 헤더를 조사하여 수신한 패킷이 헤더를 나타내는지 조사하고 하나의 패킷을 모두 수신하기 위한 패킷 사이즈 정보를 얻어와 패킷의 수신이 모두 이루어 질때 까지 recv함수를 호출하게 된다. 그러나 이때 한번의 recv 함수를 통해 전송된 데이터가 수신 버퍼에 의해 여러 패킷이 하나의 버퍼로 수신되는 경우가 있기 때문에 수신된 데이터를 조사하여 패킷 내에 헤더가 존재하는지를 체크하고, 만약 여러 개의 헤더가 존재하게 되면 그 패킷을 세그먼트하여 각각의 패킷으로 구분하여야 한다. 패킷을 구분한 후 마지막 패킷에서 헤더를 조사하여, 마지막 패킷의 사이즈를 알아낸다. 헤더내에 패킷의 사이즈와 마지막 패킷의 사이즈가 같으면, recv를 수행하지 않지만, 만약 다르다면 헤더 내의 패킷 사이즈만큼 recv함수를 수행해야 한다. 그림 5-5는 이러한 과정을 나타내고 있다.

```

While(1)
{
    size = recv( buf, size);

    SegmentationRecvData(buf,size);    // 버퍼내에 여러 패킷의 있는지 조사한 후 각
                                        // 패킷을 분류한다.

    Packet_Size = PacketHeader(buf);    // 패킷 헤더에서 패킷 사이즈 정보를 알아낸다.

    while( 패킷 사이즈만큼 수신할 때 까지 )
    {
        Recv_Size = recv(buf,size);

        Recv_Packet_Size += Recv_Size;

        CopyMemory(Buffer+Recv_Packet_Size,buf,Recv_Size);
    }
}

```

그림 5-5 TCP에서 패킷을 수신하기 위한 코드 구현

5.2 UDP 전송

5.2.1 UDP 소켓 구현에서 고려해야할 문제

UDP 프로토콜은 비연결 지향형 프로토콜로써 각 패킷이 sendto 함수에 의해 전송이 되면 그 때마다 네트워크에서 가장 빠른 루트로 데이터가 전송된다. 그렇기 때문에 전송되는 패킷의 순서가 달라지는 경우가 발생한다. 또한 UDP 특성상 전송되는 데이터는 손실될 수도 있고 에러가 발생할 수 있기 때문에 데이터를 전송하고자할 때 UDP 프로토콜을 새로운 메커니즘을 사용하여 이러한 문제점을 보완해야 한다. 데이터의 빠른 전송 특성에 실시간 응급환자의 고화질 비디오 영상이나 영상회의 데이터를 전송하기에 적합하나 진단에 사용하는 고화질 영상 정보는 데이터 손실을 최소화해야 한다. 또한 고화질 비디오 영상 데이터의 사이즈는 UDP가 한번에 전송할 수 있는 데이터의 크기보다 크기 때문에 패킷을 프래그먼트하여 전송해야하는 과정을 거쳐야 하며, 수신단에서도 이러한 패킷을 분석하여 송신단에게 재 전송을 요구하는 기능을 구현해야한다. 이때 송신단에서 큰 데이터 사이즈의 패킷을 여러 패킷으로 프래그먼트해서 보낼 때 수신단에서 몇 번

recvfrom을 해야 하는지에 대해서 헤더에 정보를 포함시켜 리포매팅 하여야 한다. UDP 특성중 TCP와 달리 '전송되는 데이터의 경계(Boundary)가 존재 한다'는 특성이 있다. 즉 송신단에서 sendto 콜함수를 호출하는 수 만큼 수신단에서 recvfrom을 해야 한다는 말이다. 이와 같은 특성 때문에 송신단에서 패킷을 몇 개의 패킷으로 나눠서 보낼 것인가에 대해서 헤더에 정보를 추가하면 수신단은 그 헤더를 분석한 후 그 수만큼 recvfrom 콜함수를 수행하면 된다. 이렇게 수신한 패킷을 하나의 패킷으로 다시 합치게 되면서, 고용량 데이터의 UDP 전송이 이루어지게 된다.

5.2.2 문제점을 해결한 UDP 송수신 메커니즘

응급환자의 고화질 영상 데이터를 전송하고자 했을 때 예를 들어 'XviD MPEG-4' 코덱 필터를 이용하여 압축된 15frame의 영상 데이터는 한 프레임 당 약 11311byte를 갖는다. UDP의 특성상 데이터의 사이즈가 큰 데이터는 한번에 전송되기 힘들기 때문에 송신단에서 데이터를 프래그멘테이션 하여야 한다. 프래그먼트 사이즈는 전송측 호스트의 MTU 사이즈에 따라 정하여진다.



그림 5-6. UDP 패킷을 리포매팅 하기 이전의 패킷

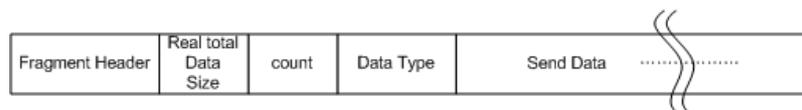


그림 5-7. UDP 패킷을 리포매팅한 후의 패킷

(1) Fragment Header(10 byte) : 전송되는 데이터의 프래그먼트 패킷 중 첫 패킷을 나타 낸다. 첫 패킷은 전송되는 총 패킷의 대한 정보를 담고 있다.

- (2) Real total Data size(4 byte) : 리포매팅시 추가된 필드를 제외한 application에 게 넘겨줄 실제 데이터의 사이즈를 나타낸다. 전송된 패킷을 합쳤을 때 데이터가 모두 전송되었는지를 체크하기 위해 사용한다.
- (3) Count(4 byte) : 전송되는 데이터의 경계가 존재한다는 UDP의 특성을 이용한 송신측 패킷 전송 카운트(send count)를 나타낸다.
- (4) DataType(4 byte) : 전송되는 데이터의 종류를 나타낸다.
- (5) SendData(실제 데이터 크기): 전송되는 실제 데이터

실제 전송되는 데이터(그림 5-6)를 그림 5-7과 같이 리포매팅한 후 사전에 정한 프래그먼트 사이즈만큼 프래그먼트하고 그림 5-9와 같이 전송되는 패킷에 시퀀스 넘버를 붙여서 전송한다. 이렇게 전송한 패킷은 수신단에서 count 필드를 조사하여 몇 번의 수신을 할 것인지를 정하게 되고, 시퀀스 넘버를 조사하여 손실된 패킷의 재 전송을 요구하게 된다.

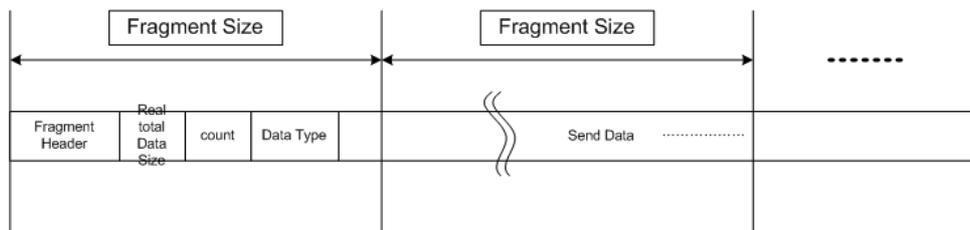


그림 5-8 . 리포매팅된 후 데이터 프래그먼트 사이즈로 프래그먼트하는 과정



그림 5-9 . 시퀀스 넘버가 첨가된 프래그먼트 데이터

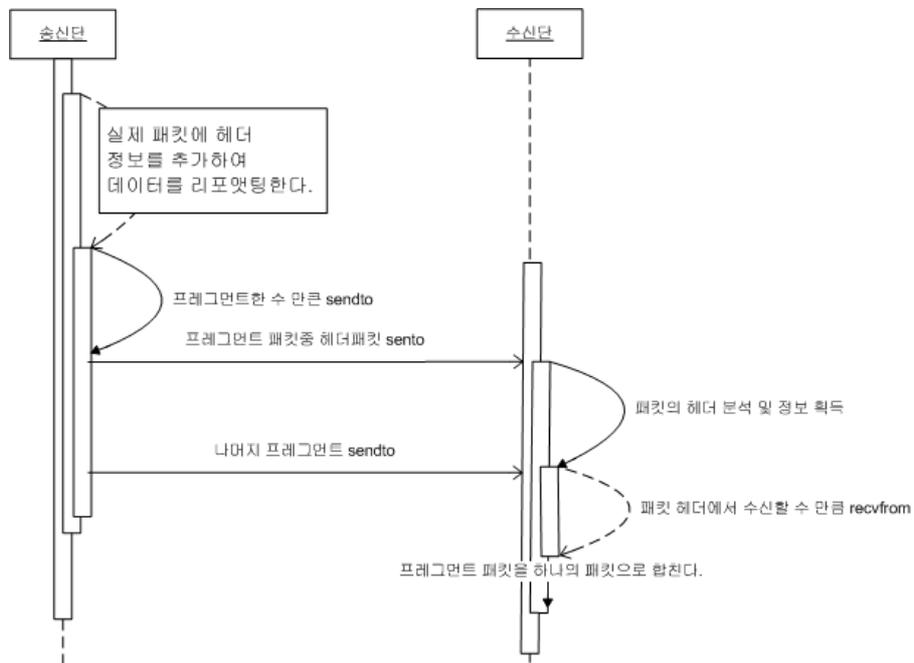


그림 5-10. UDP프로토콜을 사용한 고용량데이터 전송을 위한 메커니즘

제 6장 전체 시스템 구성

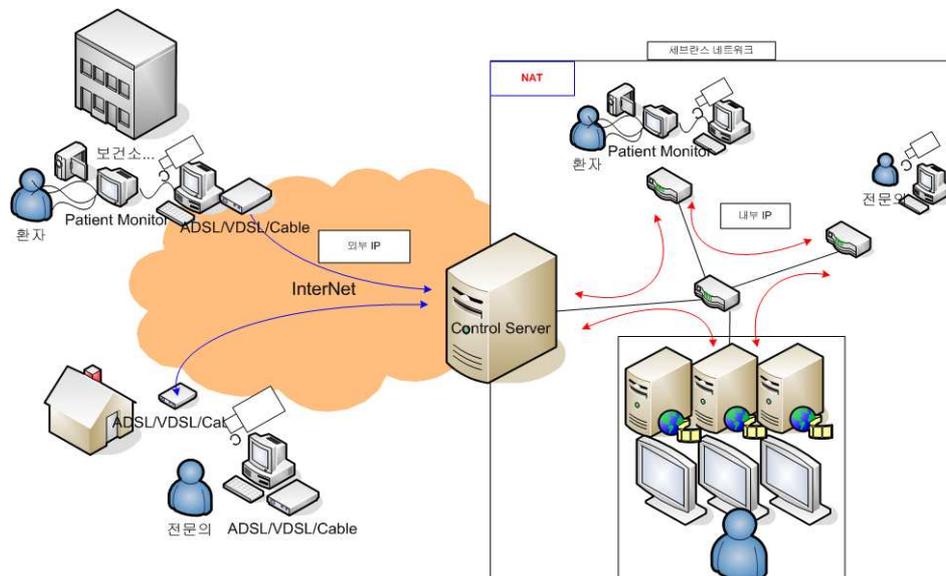


그림 6-1. 전체 시스템 구성도

응급환자 모니터링 시스템은 LAN, VDSL, ADSL, Cable Modem 등과 같은 유선망과 WLAN, CDMA 등의 무선망과 같은 다양한 접속 환경에서 선택적으로 접속할 수 있는 접속 모듈 개발하여 Seamless하게 접속하도록 개발되었다. 또한 이동통신환경의 유동 인터넷 주소를 기반으로 하는 무선망에서 다중 연결 관리 세션 모듈의 개발하여 다양한 접속환경에서 효과적인 연결 세션을 관리하도록 하였다.

유무선 망의 환자와 의사의 애플리케이션 세션을 맺기 위해 시스템 구성의 컨트롤 서버는 환자의 데이터를 전송하는 기법을 ByPassing 기법과 Flowing 기법을 사용한다. Bypassing 기법은 환자와 의사가 모두 병원망에 위치하게 되는 경우와 같이 직접 세션을 맺음으로써 데이터를 전송하는 기법으로 제어 서버는 이 세션에 대한 정보와 프레임 레이트 컨트롤 등 협상과정만을 관여하고 모니터링하게 된다. Flowing 기법은 네트워크의 변화에 따라 데이터를 control Server가 직접 송수신에 관여하는 기법이다. 이와 같은 경우는 wired(ADSL, VDSL, Cable Modem)-to-Wired(병원망), wireless(CDMA1x, EVDO, Wireless Lan)-to-Wireless, wired-to-wireless 등의 토폴로지를 가지는 경우에 해당한다.

6.1 환자측 시스템 및 모니터링 시스템(의사, 스트림 서버)

전체 시스템 구성 중 응급 환자의 멀티미디어 데이터 및 생체 신호 데이터, 방사선 이미지 파일등을 획득하여 스트림 서버와 의사측 시스템으로 전송하는 능동적인 시스템으로써 환자의 환부 및 상태를 나타내는 고화질 영상 데이터, Patient Monitor를 통해 환자의 생체 신호를 획득하는 모듈 및 디스플레이, PACS 시스템과 유기적으로 연동되어 환자의 방사선 영상의 획득, 다수의 의사와 컨설팅을 위한 컨퍼런스 기능을 구현하고 있다. 환자측 시스템은 제어 서버에 접속하는 네트워크 환경에 따라 3가지로 분류 될 수 있다. 첫째, 병원 내부 유선망을 사용하는 환자측 시스템이 존재했을 때의 경우, 둘째, 병원 외부의 ADSL, VDSL, Cable Modem을 사용하여 제어 서버에 접속했을 때, 셋째, EVDO, WLAN을 사용하는 환자 시스템으로써 제어 서버에 접속했을 때의 경우이다. 그림 6-1은 환자측 시스템에서 제어 서버로 접속하기 위한 다이얼로그로써 위 세 가지 경우의 인터페이스를 갖게 된다.

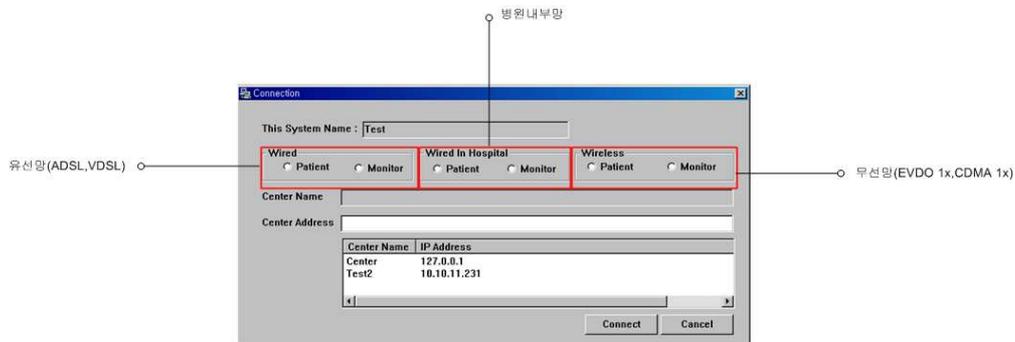


그림 6-2. 다중 커넥션을 위한 환자측 접속 인터페이스

6.1.1 병원 내부망의 환자 시스템 및 모니터링 시스템

병원 내부망의 환자는 병원 내부망의 응급 센터에 상주하면서 다중 세션을 관리 하고 모니터링하는 제어 서버에 접속하여 환자 시스템의 고화질 비디오 영상 압축 코덱, 화상회의 영상 압축 코덱, 고화질 비디오 영상의 프레임율등의 정보를 전송한다. 이후 응급 센터에서 응급의가 환자측 시스템과 컨설팅하기 위해 스트림 서버를 접속하였을 때, 또는 병원 내부의 의사가 환자를 모니터링하기 위해 접속하였을 때, 제어 서버의 ByPassing Policy에 의해 환자측 시스템과 직접 데이터 송수신이 이루어진다. 즉, 내부망에 존재하는 M개의 클라이언트(환자측 시스템)와 M개의 클라이언트(의사, 스트림서버)가 직접 세션을 맺으며 데이터를 주고 받게 된다. 환자측 시스템은 제어 서버로부터 'ByPassing'의 Policy를 결정하게 된다면, 의사측 서버와 스트림서버와의 연결을 위해 M개의 소켓을 할당하여, 통신을 하게 된다. 또한 다수의 의사와 스트림 서버간의 화상회의를 하기위해 환자측 시스템은 직접 화상회의 데이터의 Flow를 컨트롤 해주게 된다. 그림 6-2은 환자측 시스템에서 데이터를 전송을 주관하는 모습을 나타낸다.

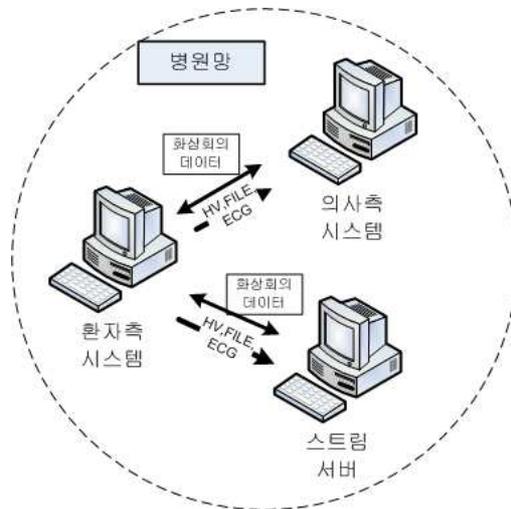


그림 6-3. 병원망의 클라이언트 접속 및 데이터전송

6.1.2 병원 외부 유선망의 환자 시스템

환자측 시스템이 병원외부의 보건소 및 가정의 외부 네트워크(ADSL, VDSL, Cable Modem)망에 존재하게 되면 환자측 시스템이 전송하는 멀티미디어 데이터 및 생체신호 데이터는 제어 서버가 주관하여 데이터 Flow을 컨트롤 하게 된다. 즉, 환자측 시스템이 의사측 시스템이나 스트림 서버가 세션을 맺어 데이터를 송수신하고자 할 때 제어 서버는 환자측 시스템으로부터 데이터를 전송 받아 직접 의사측 시스템과 스트림 서버로 전송하여 준다. 상이한 네트워크 환경을 갖는 시스템들은 네트워크 대역폭에 따라 데이터를 가공한 후 전송되어야 하는데 이러한 기능을 클라이언트(환자, 의사측 시스템)가 아닌 서버(제어 서버)가 구현하게 되어 클라이언트들의 프로세싱 부하를 덜어주게 되고, 서버가 직접 인코딩 파라미터를 조작하기 때문에 인코딩 파라미터 메시지 전송에 의한 지연이 줄어들게 된다. 또한 환자측 데이터를 직접 제어 서버가 수신하기 때문에 스트림 서버를 경유한 환자의 데이터 저장을 하지 않아도 된다. 즉, 제어 서버가 직접 데이터를 수신하기 때문에 직접 데이터베이스 서버에 데이터를 저장할 수 있다. 이러한 방식의 데이터 전송 방식을 'Flowing'이라 하고 제어 서버가 클라이언트들의 환경에 따라 'ByPassing'을 할지 'Flowing'을 할지를 자동적으로 결정하게 된다.

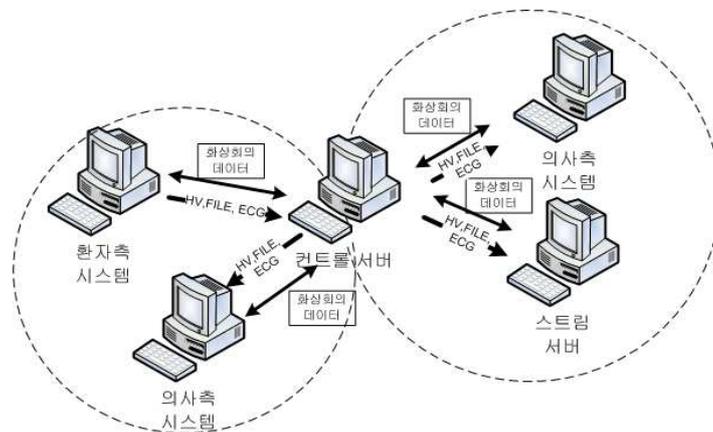


그림 6-4. 병원 외부 유선망의 환자 시스템 및 모니터링 시스템(의사, 스트림 서버)

6.2 제어 서버(Control Server)

데이터 제어 서버의 기능은 4가지 측면으로 구분할 수 있는데 모니터링 측면에서 사용자들간의 연결상태를 수시로 체크하여 다수의 1-to-M 연결을 관리한다. 데이터 스트리밍 측면에서는 의료 멀티미디어 데이터를 의사의 전송상태와 요구조건에 따라 유기적으로 실시간 전송한다. 디스플레이 측면에서는 다수의 모니터를 사용하여 여러 데이터 정보를 동시에 디스플레이 하며 세이브 측면에서는 실시간으로 환자의 정보와 멀티미디어 데이터를 데이터베이스 서버에 저장한다. 이러한 데이터 제어 서버는 다중 사용자의 상황과 특성에 따라 3단계로 개발하였다. 첫 번째 단계로는 M개의 환자와 N개의 의사가 연결되어 있지만 한 명의 환자가 오직 한 명의 의사와 통신할 경우 제어 서버는 두 사용자간의 협상을 통해 대역폭에 따라 연결 작업을 원활하게 하고 직접 통신 할 수 있도록 중계하도록 개발하였다. 두 번째 단계로 한 명의 환자에 대해 여러 명의 의사가 필요할 때 각 의사와 협상을 통해 전송 대역폭과 전문의에 특성화된 요구조건에 따라 전송 테이블을 작성한다. 전송 테이블에서는 각 전문의마다 서로 다른 우선순위 데이터(modality)를 결정하여 지속의 통신환경일 때 우선순위에 따라 데이터를 차별적으로 전송한다. 세 번째 단계는 첫 번째와 두 번째 시스템을 통합한 것으로 한명의 환자가 몇 명의 의사에 연결되느냐에 따라 중계할지 제어 서버를 통해 통신할 지 결정할 수 있도록 개발하였다. 첫 번째 시스템의 경우 제어 서버는 연결 상태 및 전송 대역폭 확인과 중계만을 담당하기 때문에 낮은 전송 지연과 처리 지연을 가지는 장점을 가진다. 하지만 다수 사용자의 통합된 제어가 불가능하다. 두 번째 시스템의 경우 다중 환경의 접속에서 다양한 대역폭에 최적화된 통신을 할 수 있고 데이터 베이스의 저장이나 다수 사용자 관리가 용이하다는 장점을 가진다. 세 번째 통합 시스템은 첫번째 개발된 제어 서버와 두번째 개발된 제어 서버의 장점만을 결합하여 응급의료 시스템에 최적화되도록 설계되었다.

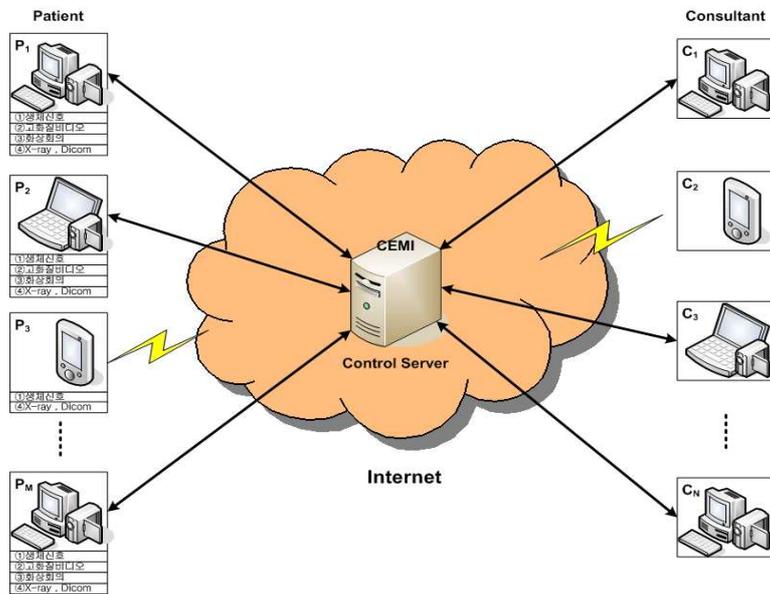


그림 6-5. 제어 서버를 사용한 다중 응급의료 시스템 기본 구조

그림 6-5는 제어 서버를 사용한 다중 기반 응급의료 시스템의 기본 구조이다. M개의 환자는 기본적으로 4가지 멀티미디어 데이터(생체신호, 고화질 비디오, 화상회의, X-ray 또는 DICOM 이미지)를 전송한다. 이것은 환자의 위치 또는 사용 환경에 따라 전송 데이터의 수가 달라질 수 있다. 이러한 데이터는 제어 서버를 통해 N개의 의사로 연결되어 서로간의 통신이 가능하다.

6.2.1 1단계 제어서버(By-Passing)

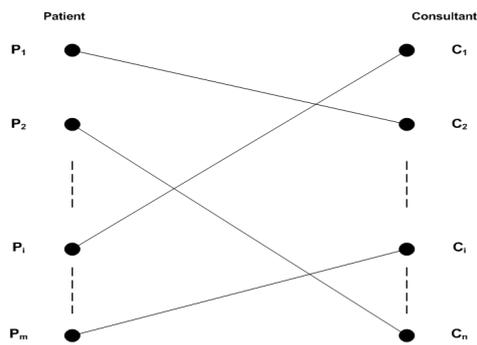


그림 6-6. M-to-N 사용자의 독립적인 연결

1단계 제어 서버는 그림 6-6과 같이 M개의 환자와 N개의 의사가 전송 서버에 연결되어 있지만 상호 독립적인 통신이 필요한 경우를 위해 개발되었다. P1 환자가 오직 C1 의사와 통신하고자 할 때의 제어 서버 흐름도는 그림 6-7과 같다.

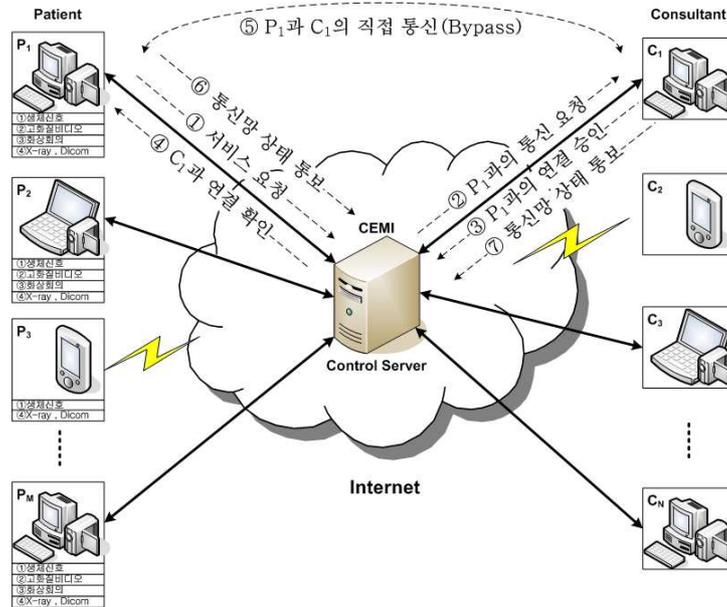


그림 6-7. 1단계 제어 서버 연결 흐름도

환자 P1의 응급 상태가 전문의 C1과의 컨설팅이 요구되는 경우 최초 P1은 제어 서버에 응급 상황에 따른 서비스를 요청한다. 제어 서버는 환자의 상태와 전송 대역폭을 받아들여 C1에게 P1의 네트워크 주소, 전송 대역폭 정보와 멀티미디어 데이터 종류 정보를 전달한다.

C1은 P1과의 최적의 통신이 가능하도록 압축률 등을 조절한다. 이것은 다시 역순으로 P1에게도 적용되어 C1과의 연결 확인을 받으면 C1의 전송 환경과 전문의 상황에 따라 전송 데이터의 압축률과 종류를 결정한다. 이상의 협상 과정을 제어 서버는 주관한 후 상호간의 정보 교환이 완료되면 P1과 C1은 제어 서버를 거치지 않고 직접적으로 통신(Bypass)하면서 1분 간격으로 통신망 상태를 제어 서버에 전송하여 한쪽의 통신망 상태가 급격히 변화하더라도 제어 서버를 통해 데이터의 손실 없이 신속한 처리를 가능토록 한다.

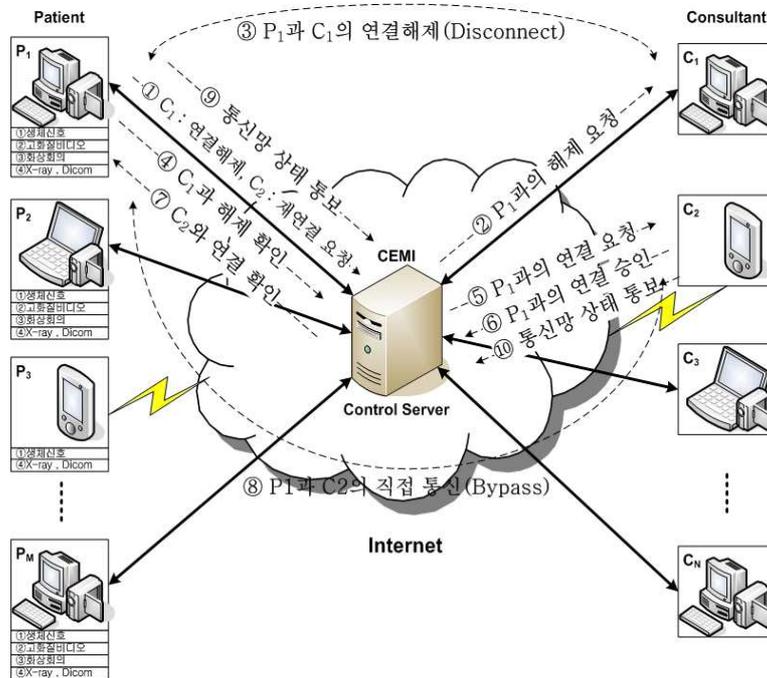


그림 6-8. 1단계 제어 서버에서 사용자의 전환이 필요한 경우의 흐름도

1단계 제어 서버는 여러 명의 사용자가 1대 1로 연결이 되기 때문에 환자나 의사 쪽에서 사용자 전환이 필요한 경우가 발생할 수 있다. 예를 들어 한 명의 환자가 다른 전문의와의 통신이 필요한 경우 또는 반대로 다른 환자가 통신 중인 전문의와 다른 연결이 필요한 경우이다. 첫 번째 경우는 그림 6-8과 같이 P1이 C1과의 연결을 해제하고 C2와의 연결을 요청하면 제어 서버는 C1에 해제 요청을 통보하여 안정적으로 P1과 C1의 연결이 해제되도록 한다. 이후 P1과 C2의 협상 및 연결 후 직접 통신 할 수 있도록 하는 과정은 그림 3과 동일하다. 두 번째 경우처럼 한 명의 전문의에 대해 다른 환자가 접속을 요청할 때는 기존의 P1과의 통신은 유지한 채 다른 포트와 다중 모니터를 사용하여 동시에 통신이 가능하도록 한다.

이러한 1단계 제어 서버는 사용자들간의 직접 통신이 가능하도록 제어하기 때문에 서버를 통과하는 통신에 비해 불필요한 전송 시간과 처리 시간을 줄일 수 있다는 장점이 있다. 반면에 여러 명의 의사와 동시에 접속하는 경우나 데이터 베이스 저장

불가능하다는 단점을 가진다.

6.2.2 2단계 제어 서버(Flow-Control)

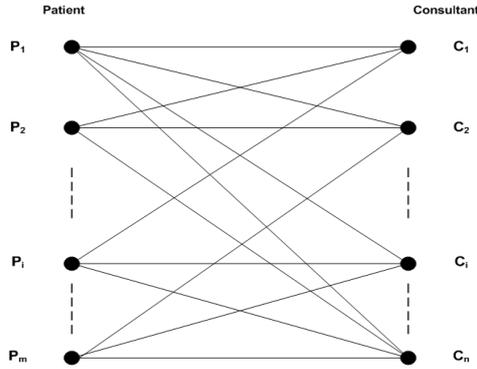


그림 6-9. M-to-M 사용자의 다중 연결

2단계 제어 서버는 데이터 베이스 서버의 추가로 저장 기능을 강화하고 그림 6-9와 같이 환자가 여러 명의 의사와 동시에 접속하여 통신하고자 하는 경우를 위해 통합적인 제어가 가능하도록 개발되었다.

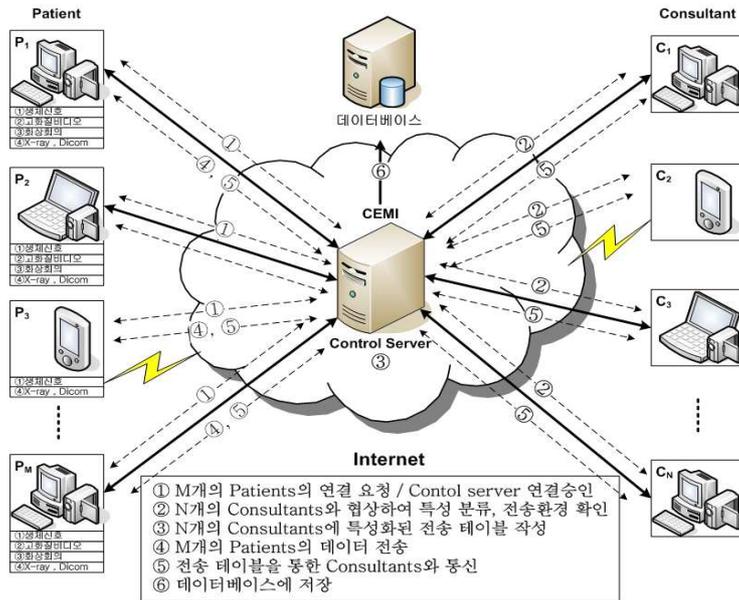


그림 6-10. 2단계 제어 서버의 다중 연결 처리 흐름도

그림 6-10은 M명의 환자(P1, ,PM)가 N명의 의사(C1, , CN)와 실시간 동시 연결에 대한 제어 서버의 처리 흐름도를 보인다. 다중 접속에서 가장 중요한 고려 사항은 LAN, x-DSL, WLAN과 같은 다양한 네트워크 환경에서 발생하는 대역폭의 변화 문제를 처리하는 것이다. 정보의 송신부라 할 수 있는 환자 부분은 각자의 대역폭을 최대한 활용하여 제어 서버로 전송하기 때문에 큰 문제가 되지 않지만 수신부가 되는 의사 쪽에서 낮은 대역폭을 가진 상태라면 심각한 전송 지연 또는 데이터 손실이 발생할 수 있다. 기본적으로 N명의 의사에게 환자의 정보를 전송하는 것은 다중 복사(multiple copy) 방식을 사용한다. 그리고 다양한 전송 대역폭 문제는 의료진료 시스템의 특성에 따라 N명의 전문의가 환자의 상태를 컨설팅 하기 위해 필요한 데이터는 서로 상이하다는 관찰로부터 해결책을 제시할 수 있다. 예를 들어 심장 전문의인 경우 4가지 멀티미디어 데이터(생체신호, 고화질 비디오, 화상회의, X-ray 이미지) 중 생체신호를 우선적으로 요구하고 방사선과 전문의인 경우 X-ray 이미지를 우선적으로 요구한다. 그러므로 제어 서버에 해당 전문의에 특성화된 우선순위 전송 테이블을 응급 전문의와 각 분야의 전문의와 상의하여 작성한 후 입력하였다. (표6-1)

제어 서버는 그림 6-10의 ①번 과정에서 각 환자의 데이터 종류와 압축 정보 및 크기, 전송 대역폭을 확인하고 ②번 과정에서 다중 복사를 위해 각 의사의 전송 상태와 전문의의 종류를 확인한다. 다수 환자가 전송한 데이터의 대역폭 보다 각 의사의 가지는 대역폭이 크면 전송테이블에 관계없이 모든 데이터를 안정적으로 보낼 수 있지만 반대로 의사의 대역폭이 크게 떨어진다면 전문의에 따라 부여된 우선순위 데이터만을 대역폭에 맞게 전송한다. 또한 환자 정보 데이터와 전송에 사용되는 멀티미디어 관련 데이터 및 제어 정보를 실시간으로 데이터베이스 서버에 저장한다.

전문의	우선순위 1	우선순위 2	우선순위 3	우선순위 4
응급의학과	생체신호	고화질 비디오	X-ray 이미지	화상회의
심장내과	생체신호	화상회의	고화질 비디오	X-ray 이미지
방사선과	X-ray 이미지	생체신호	화상회의	고화질 비디오
내과	생체신호	X-ray 이미지	화상회의	고화질 비디오
외과	고화질 비디오	화상회의	X-ray 이미지	생체신호
신경외과	생체신호	화상회의	X-ray 이미지	고화질 비디오

표 6-1. 각 전문의에 따른 우선순위 전송 테이블

6.2.3 3단계 제어서버(통합 제어)

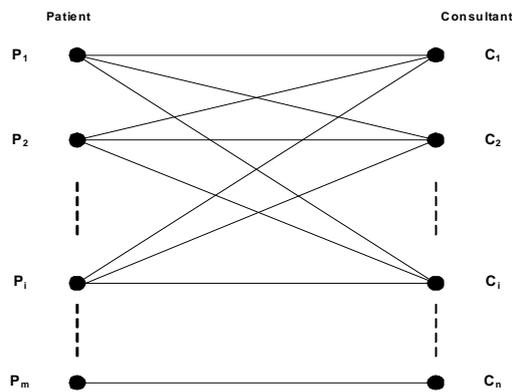


그림 6-11. M-to-N 사용자의 독립연결과 다중 연결의 혼합

3단계 제어 서버는 1단계와 2단계에서 개발한 제어 서버를 통합시킨 것으로 그림 6-11과 같이 환자와 의사간에 독립적인 연결(Pm과 Cn)과 다중 연결이 혼합되어 있을 때 유기적으로 통신 설정을 조절한다.

3단계 제어 서버는 서비스 받고자 하는 환자가 몇 명의 의사와의 컨설팅이 필요한지를 우선적으로 확인한다.

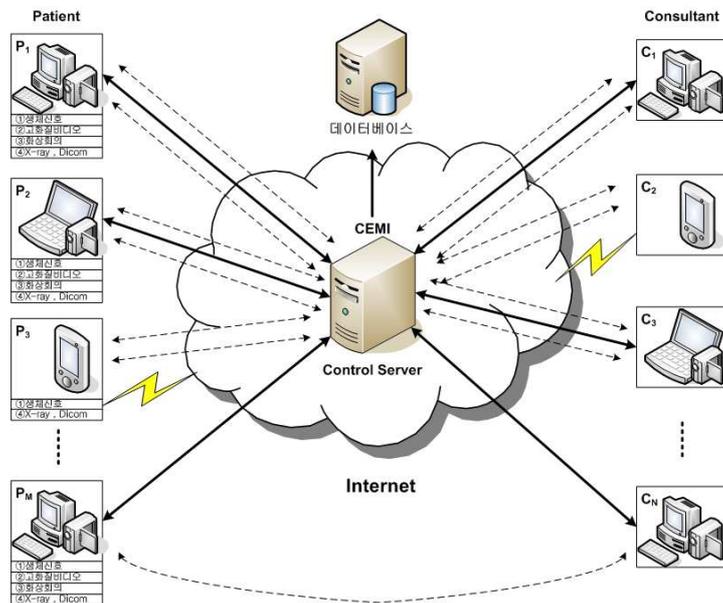


그림 6-12. 3단계 제어 서버 흐름

한 명의 전문의 만이 필요한 경우 1단계 제어 서버의 연결 흐름도에 따라 상호간의 연결 정보를 교환하여 직접적인 통신을 하도록 하고 여러 명의 전문의가 필요할 경우에는 2단계 제어 서버와 같이 각 전문의의 대역폭을 측정하여 전송 테이블에 따라 다중 복사(Multi-copy) 방식을 사용한다. 이러한 통합된 시스템은 불필요한 전송 지연 및 처리 지연을 최소화 하면서 사용자의 요구사항에 최적화된 통신이 가능할 뿐만 아니라 데이터의 저장 문제도 해결 할 수 있는 장점을 가진다

6.2.4 제어 서버 시스템 구성도

① 1단계 직접 연결 제어 서버

환자는 먼저 Control Server에 접속을 요청하고 Server에서 connection을 승인하면 환자 측은 전송할 Data의 종류와 정보를 전송한다. 이때 Control Server에는 환자의 IP정보와 Port정보가 저장되고 그 후 전문의의 Consultant를 기다리게 된다.

전문의를 역시 Control Server에 접속한 후 환자의 정보(환자 IP, Port등)를 확인한 후에 환자측 IP와 Port를 가지고 와 직접 환자에게 접속 요청을 하게 되고 이로써 환자와 직접

세션을 맺게 된다. 즉, 환자 측은 Control Server에 접속한 후 전문의의 접속을 허용하기 위한 Server Socket을 생성하여 Listen 하고 있는 상태가 되어있다.

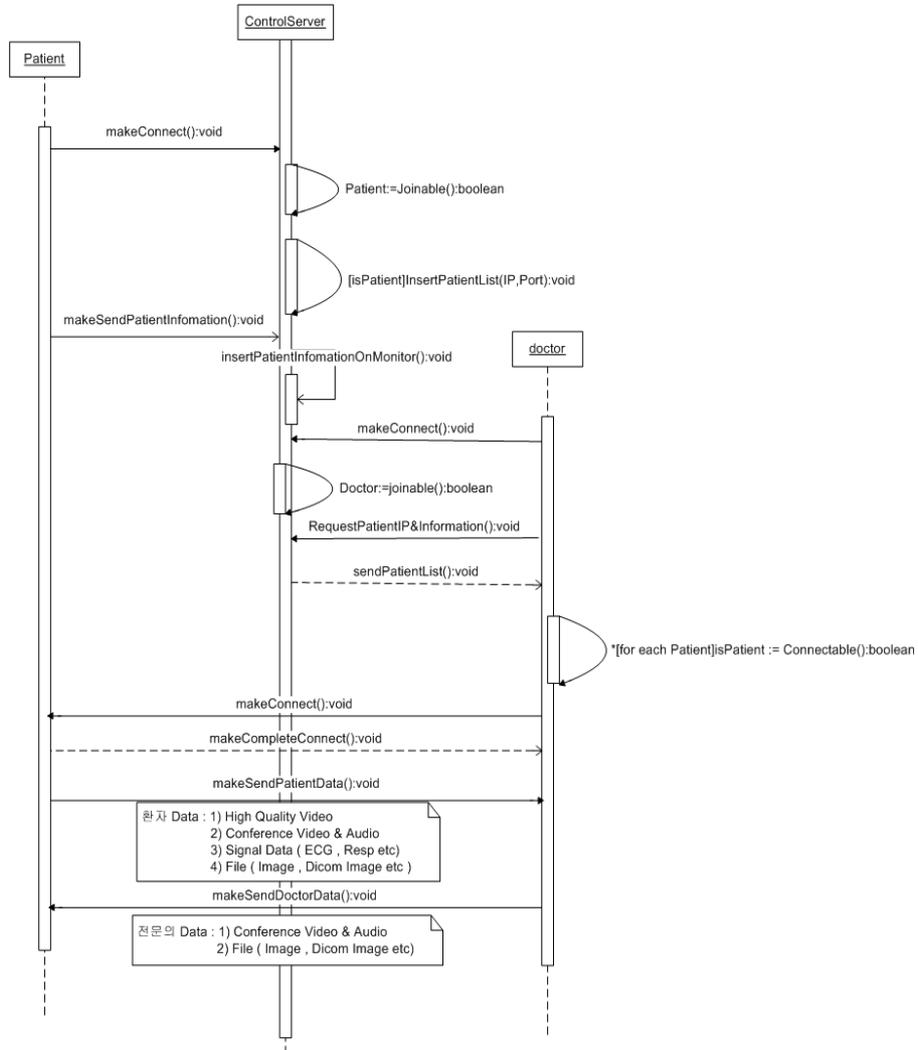


그림 6-13. 1단계 직접 연결 시퀀스 다이어그램

② 2단계 다중 연결 제어 서버

1 단계 직접 세션을 연결 하는 것 과는 다르게 Control Server가 환자의 데이터를 직접 받게 되는데 이 받은 데이터는 선택된 전문의의 네트워크 상태에 따라 환자 정보를 우선순위를 두어 전송함으로써 전문의에 따른 효율적인 Monitoring을 가능하게 된다.

1단계와 같이 환자 측이 control Server에 접속이 되면 IP,Port 등의 정보가 저장되며 control server가 전문의와의 세션을 정할 때 이 IP와 Port를 사용하여 환자 측과 전문의의 데이터를 중재하게 된다.

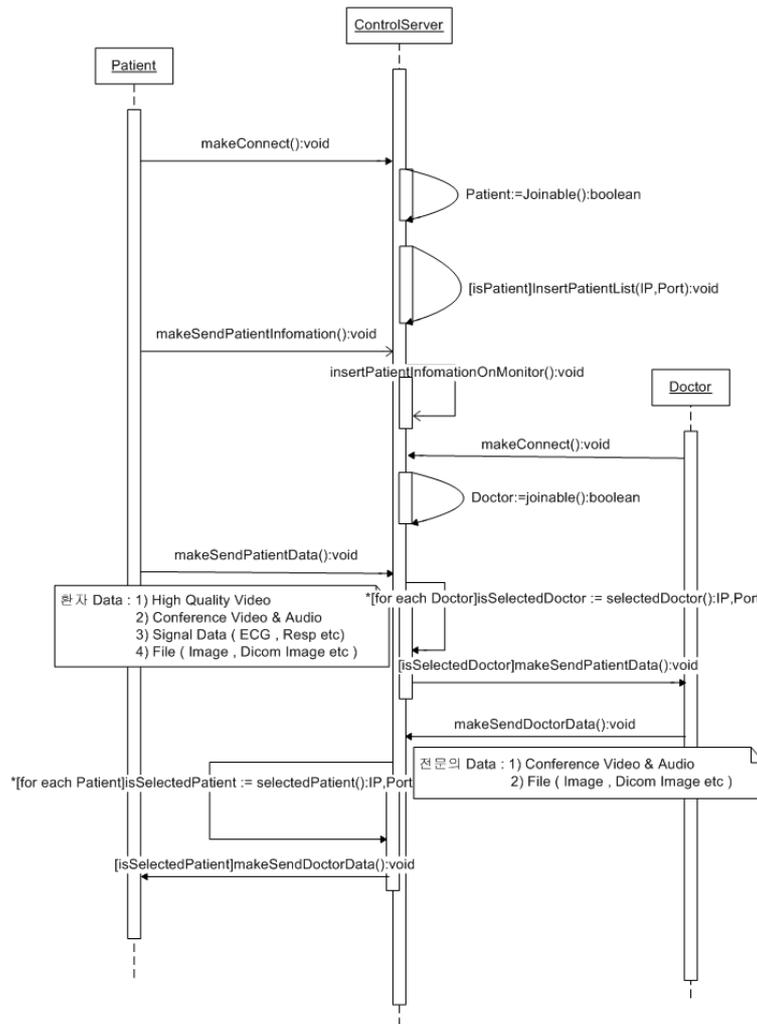


그림 6-14. 2단계 다중 연결 시퀀스 다이어그램

③ 3단계 통합 시스템 제어 서버

1단계와 2단계의 기술을 혼합한 3단계 통합 서버는 Control Server가 환자 측과 전문의 측을 계속적으로 Monitoring 하면서 1,2 단계 서버기능 중 보다 효율적인 데이터 전송을 할 수 있는 서버기능을 선택한 후 환자 측 데이터를 전문의로 전송하게 된다.

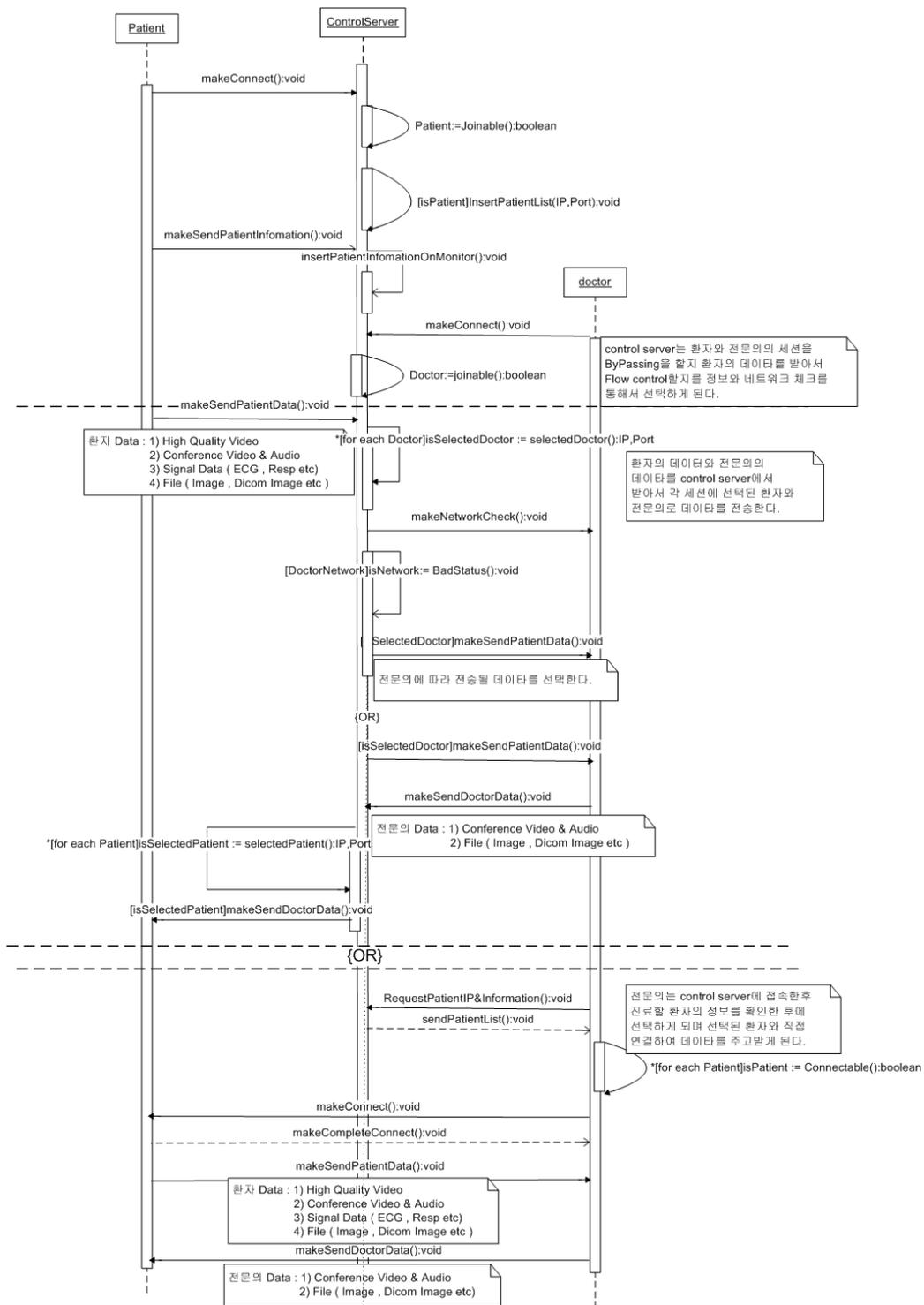


그림 6-15. 3단계 통합 시스템 제어 서버 시퀀스 다이어그램

6.2.5 제어 서버의 NAT(Network Address Translator)에 의한 문제점 해결방안

Network address translator의 약어로 OSI 모델의 3계층인 네트워크 계층에서 사설 IP 주소를 공인 IP 주소로 변환하는데 사용하는 통신망의 주소 변환기를 의미한다. 인터넷의 공인 IP 주소는 한정되어 있기 때문에 가급적 이를 공유할 수 있도록 하는 것이 필요한데 NAT를 이용하면 사설 IP 주소를 사용하면서 이를 공인 IP 주소와 상호변환할 수 있도록 하여 공인 IP 주소를 다수가 함께 사용할 수 있도록 함으로써 이를 절약할 수 있다.

NAT를 사용하는 목적에는 2가지가 있는데, 첫째는 위에서 설명한 경우처럼 인터넷의 공인 IP 주소를 절약할 수 있다는 점이고 둘째는 인터넷이란 공공망과 연결되는 사용자들의 고유한 사설망을 침입자들로부터 보호할 수 있다는 점이다. 공개된 인터넷과 사설망 사이에 방화벽(Firewall)을 설치하여 외부 공격으로부터 사용자의 통신망을 보호하는 기본적인 수단으로 활용할 수 있다. 이 때 외부 통신망 즉 인터넷망과 연결하는 장비인 라우터에 NAT를 설정할 경우 라우터는 자신에게 할당된 공인 IP 주소만 외부로 알려지게 하고, 내부에서는 사설 IP 주소만 사용하도록 하여 필요시에 이를 서로 변환시켜 주어 외부 침입자가 공격하기 위해서는 사설망의 내부 사설 IP 주소를 알아야 하기 때문에 공격이 불가능해지므로 내부 네트워크를 보호할 수 있다. 외부에 의해 공격된 네트워크로 인해 데이터가 변형되는 것은 환자의 생명에 직접적인 영향을 줄 수 있기 때문에 병원망에서의 NAT를 사용하는 것이 일반적이다.

그러나 이런 기본적인 NAT 기능으로 인하여 여러 대의 PC가 하나의 공인 IP를 가지고 인터넷을 동시에 사용할 수 있다는 장점이 있는 반면 외부에서 인식이 가능한 공인IP로는 공유기까지는 접근이 가능하나 그 밑단에 있는 PC들은 가상IP를 가지고 있기 때문에 외부에서 필요에 의하여 공유기 하단에 연결된 PC로 접근을 하려면 제한이 된다는 단점이 있다.

이러한 단점을 해결하기 위해 그림 6-16과 같은 터널링 방법을 사용하여 병원망을 포함한 전체 시스템을 구성하였다.

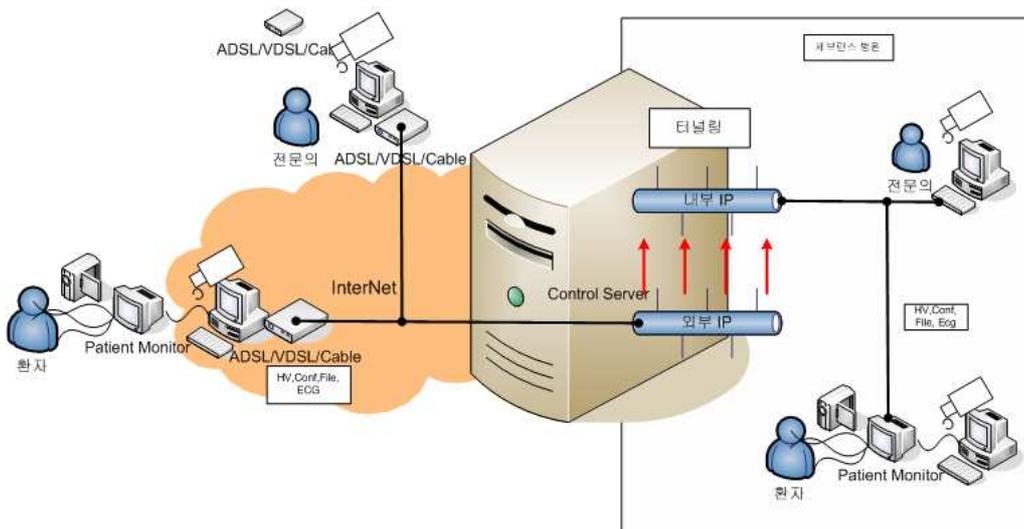


그림 6-16. 병원망의 NAT문제를 해결하기위한 터널링 방법

병원망에 속해 있는 응급 센터의 제어 서버는 외부망의 클라이언트와 세션을 맺기 위해 외부 IP를 갖는 NIC 와 병원망 내부의 클라이언트와 세션을 맺기 위한 NIC를 가지게 된다. 그림 6-16에서의 외부망의 환자 및 전문의는 제어 서버의 외부IP와 통신을 하게 되고, 이를 통해 전송 받은 데이터는 병원망의 내부 클라이언트와 통신하기 위해 내부IP의 세션으로 데이터를 전송하게 된다. 병원망 내부 클라이언트도 이와 같은 방법을 통해 병원외부망의 클라이언트와 통신하게 된다.

```

C:\Documents and Settings\Administrator>ipconfig

Windows 2000 IP Configuration

Ethernet adapter 로컬 영역 연결 2:

    Connection-specific DNS Suffix  . : 
    Autoconfiguration IP Address. . . : 169.254.77.156
    Subnet Mask . . . . . : 255.255.0.0
    Default Gateway . . . . . : 

Ethernet adapter 로컬 영역 연결:

    Connection-specific DNS Suffix  . : 
    IP Address. . . . . : 10.19.11.120
    Subnet Mask . . . . . : 255.255.255.0
    Default Gateway . . . . . : 10.19.10.1

C:\Documents and Settings\Administrator>_
  
```

그림 6-17. 제어 서버에 셋팅된 외부 IP 와 내부 IP

그림 6-17은 제어 서버에 NIC 두개를 설치한 후 병원내부 IP와 외부 IP를 설정한 그림이고, 이렇게 설정된 IP는 제어 서버 애플리케이션에서 그림 6-18과 같이 IP를 설정하게 된다.

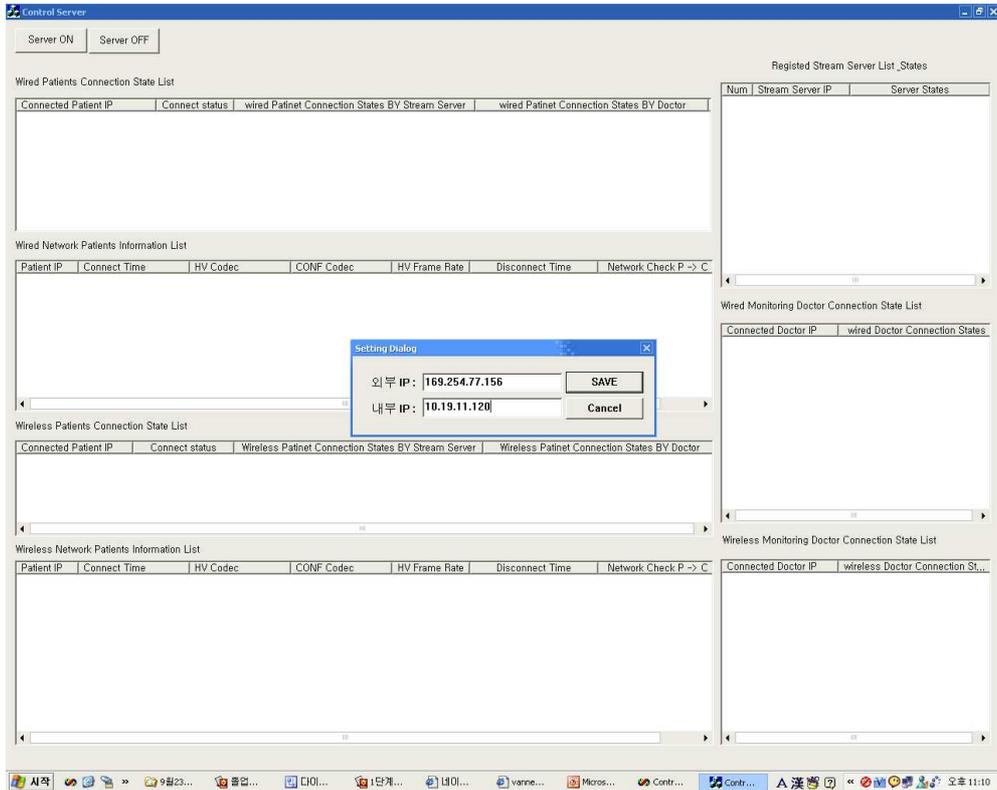


그림 6-18. 외부 IP와 내부 IP를 셋팅하는 제어 서버 애플리케이션

제 7장 실험 환경 및 결과

개발된 유무선 통합 다중 응급 원격 진료시스템은 전송되는 응급환자의 멀티미디어 데이터를 특성상 표 7-1과 같은 네트워크 프로토콜을 사용하여 전송한다. 환자의 상태를 진단하기 위한 실시간 고화질 영상 데이터는 고용량의 데이터를 전송하기 위해 본 논문에서 제안한 UDP 메커니즘에 의해 전송하고 전문의들간의 화상회의 데이터는 실시간 데이터 전송에 적합한 기존의 UDP 프로토콜을 사용하였다. 또한 환자의 생명에 직접적인 영향을 주는 환자의 이미지 파일이나 생체 신호 데이터는 신뢰적인 데이터 송수신을 보장하는 TCP 프로토콜을 사용하였다.

데이터	기반 네트워크 프로토콜
HV (High Quality Video)	UDP
Conference Video & Audio	UDP
File(Dicom ,jpeg,bmp etc)	TCP/IP
Signal (ECG, Resp, NIBP, SpO2)	TCP/IP

표 7-1. 유무선 통합 다중 응급 원격 진료시스템에서의 멀티미디어 전송 네트워크 프로토콜

응급환자 모니터링 시스템이 LAN, VDSL, ADSL, Cable Modem 등과 같은 유선망과 WLAN, CDMA 등의 무선망과 같은 다양한 접속 환경에서 데이터 송수신 환경을 실험하기 위해 다음과 같은 4가지 환경을 기반으로 시스템을 구성하였다.

본 논문에서 개발된 시스템은 **연세대학교 세브란스 병원**의 병원망을 사용하여 임상 실험을 테스트 하였으며 스트림 서버는 병원망에 속해 있다.

① 유선망 - to - 유선망

환자측 시스템과 모니터링 시스템(의사측 시스템 , 스트림 서버)의 네트워크 환경이 유선망 환경일 때를 말한다. 실험 환경은 다음과 같다.

(a) In Hospital Patient (LAN)

(b) 고화질 (30 frames) 720 * 480

- raw data : 158Mbps (1 frame 약 5.27Mbps)

- XviD Mpeg-4 Codec : 1.67Mbps (1 frame 약 0.055Mbps)

- (c) 화상회의(Video, Audio)
 - Video (H261 : 약 60Kbps)
 - Audio (G711 : 약 70~110kbps)
- (d) 환자 이미지 파일 (Dicom, Jpeg , Bmp)
- (e) 생체신호(Ecg, SpO2, Resp, NIBP) : 약 8kbps

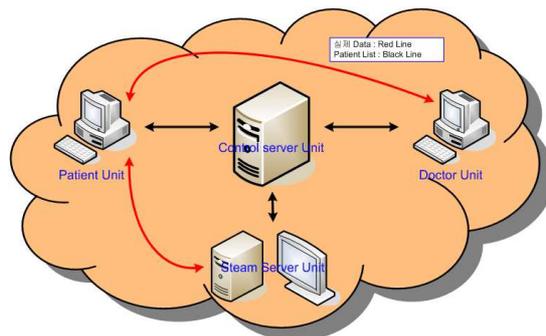


그림 7-1. 유선망(환자) to 유선망(모니터링) 시스템 구성도

② 무선망 - to - 유선망

환자측 시스템은 CDMA 1x , EVDO, WLAN등의 무선망의 토폴로지로 구성되어 있으며 모니터링 시스템(의사측 시스템 , 스트림 서버)의 네트워크 환경이 유선망 환경일 때를 말한다. 실험 환경은 다음과 같다.

- (a) 무선망의 환자 : In Car (EVDO)
- (b) 고화질 (1 frames) 720 * 480
 - XviD Mpec-4 Codec : 약 55kbps
- (c) 환자 이미지 파일 (Dicom, Jpeg , Bmp)
- (d) 생체신호(Ecg, SpO2, Resp, NIBP) : 약 8kbps

무선망의 환경에서는 네트워크의 대역폭의 변화에 따라 전송할 응급환자의 데이터가 제한적이다. 예를 들어 위와 같은 실험 환경에서 EVDO망을 사용한 환자의 데이터는 약 153.6kbps의 업로드와 약 2.5Mbps의 다운로드를 갖기 때문에 고화질 영상 데이터는 2프레임 이상을 전송할 수 없고, 화상회의 데이터는 제어 서버의 전송 우선순위 정책에 의해 생략될 수 있다.

	ADSL		VDSL		Wireless LAN		Ethernet		Bluetooth		IEEE1394		EVDO	
	Down	Up	Down	Up	Down	Up	Down	Up	Down	Up	Down	Up	Down	Up
Min	2.07	0.29	4.22	1.13	2.01	1.29	4.78	4.23					1.3	0.06
Max	6.56	0.66	12.3	6.27	11.3	6.84	38.3	21.65					2.0	0.13
Mean	4.42	0.62	7.29	3.19	6.01	3.57	28.76	16.3	0.72		150.2 4		1.61	0.11

(Down : Download, Up : Upload, ADSL 및 VDSL : KT에서 제공한 Megapass, Wireless

LAN : KT에서 제공한 Netspot, Bluetooth : bluespot 단위 : Mbps)

표 7-2. 외부통신망에 따른 Residential network의 전송 속도

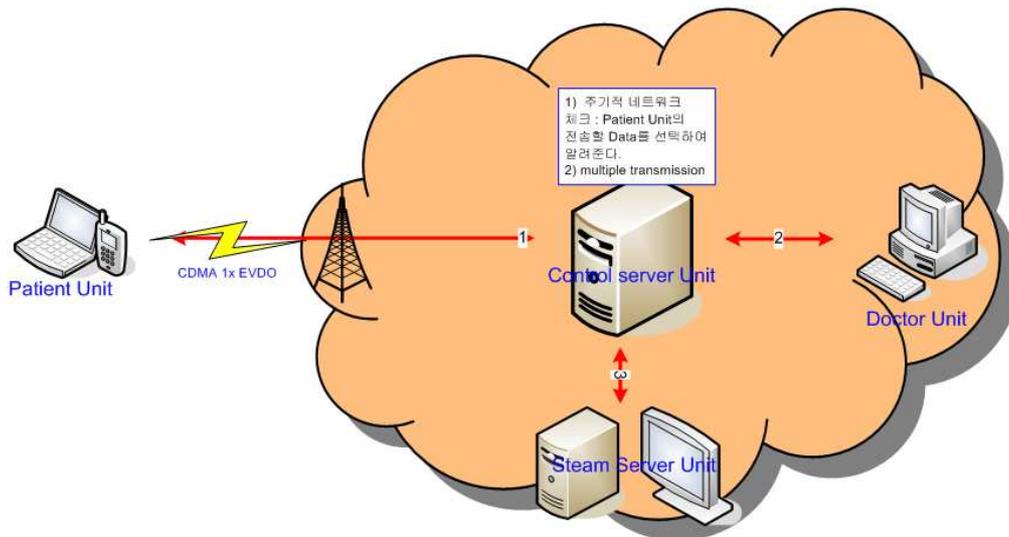


그림 7-2. 무선망(환자) to 유선망(모니터링) 시스템 구성도

③ 유선망(환자) - to - 무선망(의사) - to - 병원망(스트림 서버)

환자측 시스템은 ADSL, VDSL, Cable Modem등의 외부 유선망이나 병원내 유선망의 토폴로지로 구성되어 있으며 스트림 서버는 병원내의 LAN에 존재하고, 의사측 시스템은 EVDO, WLAN 등의 네트워크 환경일 때를 말한다. 실험 환경은 다음과 같다.

- (a) 외부 유선망의 환자 : ADSL, LAN
- (b) 무선망의 의사 : EVDO
- (c) 고화질 (9 frames) 720 * 480
 - XviD Mpec-4 Codec : 0.545Mbps
- (d) 화상회의(Video, Audio)
 - Video (H261 : 약 60Kbps)
 - Audio (G711 : 약 70~110kbps)
- (e) 환자 이미지 파일 (Dicom, Jpeg , Bmp)
- (f) 생체신호(Ecg, SpO2, Resp, NIBP) : 약 8kbps

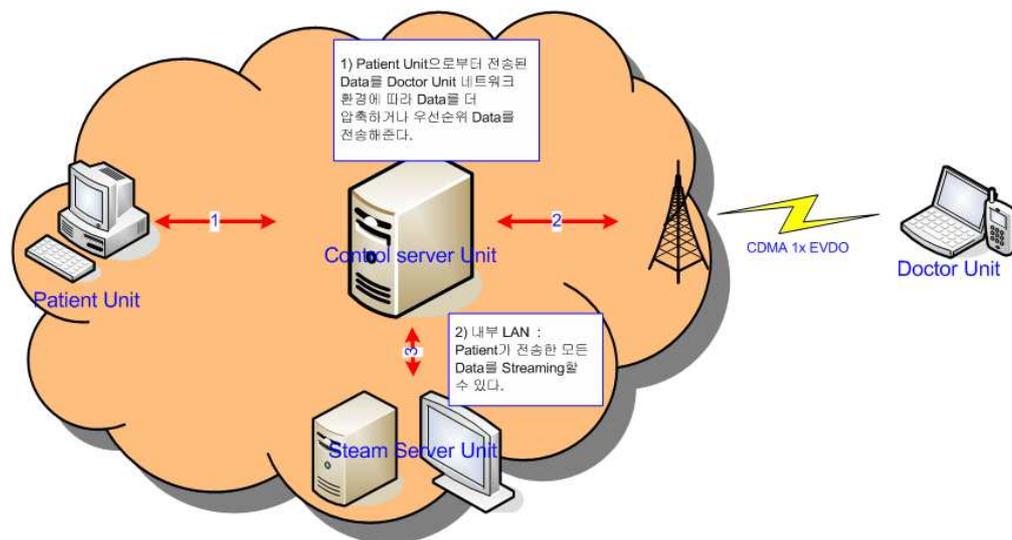


그림 7-3. 유선망(환자) to 무선망(의사) to 유선망(스트림 서버) 시스템 구성도

④ 무선망(환자) - to - 무선망(의사) - to - 유선망(스트림 서버)

환자측 시스템과 의사측 시스템은 EVDO, WLAN 등의 무선망의 네트워크를 구성하고 스트림 서버는 병원내의 LAN에 존재하는 환경일 때를 말한다. 실험 환경은 다음과 같다.

- (a) 무선망의 환자 : In Car (EVDO)
- (b) 무선망의 의사 : In Car (EVDO, WLAN)

- (c) HV (1 frame) 720 * 480
 - XviD : 약 55kbps
- (d) Signal (Ecg, Resp, SpO2, NIBP) : 약 8kbps
- (e) File (Dicom , Jpeg, Bmp)

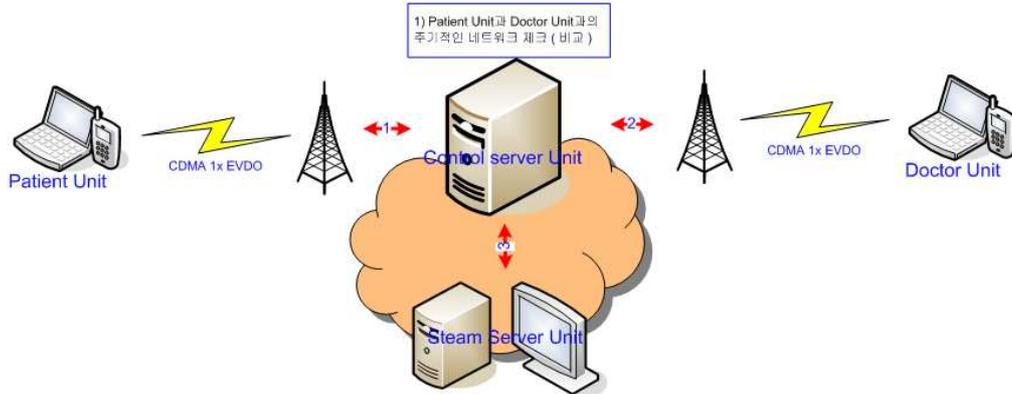


그림 7-4. 무선망(환자) to 무선망(의사) to 유선망(스트림 서버) 시스템 구성도

7.1 환자측 시스템의 애플리케이션 인터페이스



그림 7-5. 환자측 시스템의 애플리케이션

그림 7-5는 환자측 시스템 애플리케이션을 나타내고 있으며, 그중 1번의 인터페이스는 환자의 상태를 나타내는 고화질 영상 데이터를 나타낸다. 2번은 화상회의 인터페이스로써 각각의 시스템의 시스템명과 화상데이터의 영상 데이터를 나타낸다. 화상회의 오디오 데이터는 시스템의 오디오 장치 필터로 출력되어 스피커 및 헤드셋을 통해 확인할 수 있다. 3번 인터페이스는 환자의 생체 신호 데이터에 대한 인터페이스로써 상단의 고정적인 ECG 신호 비트맵과 아래쪽(NIBP, Resp, SpO2를 오른쪽 라디오 버튼으로 선택)비트맵을 통해 다른 신호 데이터를 선택적으로 확인할 수 있다. 4번 인터페이스를 통해 환자 및 의사, 스트림 서버간의 파일을 전송할 수 있고, 파일 다이얼로그의 컴포넌트를 통해 파일이름, 파일 사이즈, 파일이 생성된 날짜, 파일의 있는 경로를 확인할 수 있다.

7.2 의사측 시스템 및 스트림 서버의 애플리케이션 인터페이스

의사측 시스템 및 스트림 서버의 모니터링 시스템은 제어 서버로부터 환자에 대한 정보를 제공받아 컨설팅할 환자를 선택하게 된다. 이때 환자측 시스템의 IP 정보도 함께 제공받게 되며, 제공받은 환자측 시스템의 IP 리스트중 하나의 환자 시스템의 IP를 선택하면, 제어서버는 'Bypassing'기법을 사용할 것인지, 'Flowing'기법을 사용할 것인지를 결정하여 세션에 관여된 환자, 의사, 스트림 서버에게 알려주게 된다. 그림 7-6은 모니터링 시스템에서 제어서버로부터 전송된 환자의 IP리스트를 확인하고 선택하는 과정을 나타낸다. 그림 7-7은 모니터링 시스템의 애플리케이션 인터페이스를 나타내는 것으로써 환자측 시스템과 동일한 인터페이스를 가지게 된다.

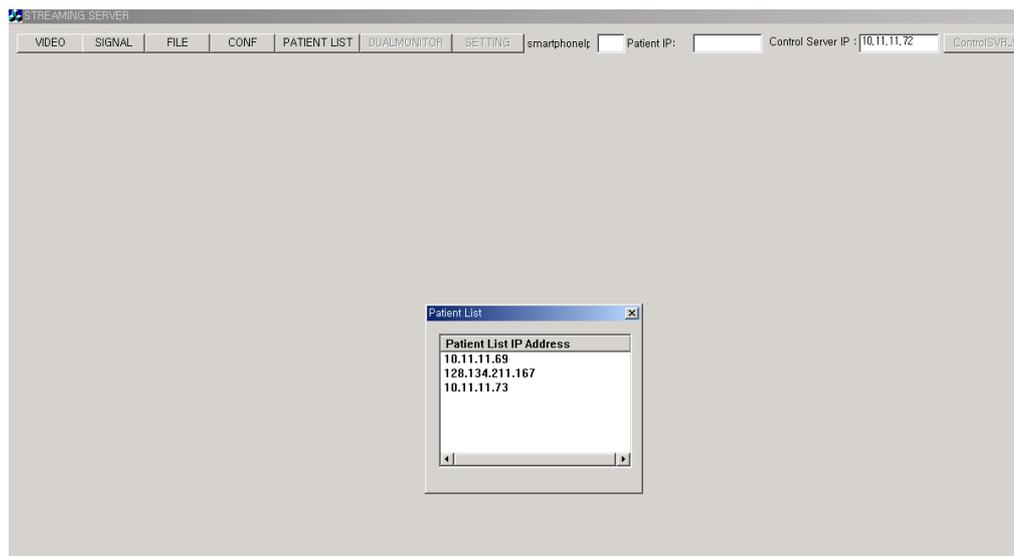


그림 7-6. 모니터링 시스템의 환자 IP 리스트 선택 과정



그림 7-7. 모니터링 시스템(의사, 스트림 서버)의 애플리케이션



그림 7-8. ECG 확장 디스플레이 컴포넌트

그림 7-8은 환자측 시스템 및 모니터링 시스템의 생체신호 다이얼로그 컴포넌트 중 'ECG_SPECIFY'의 버튼을 클릭 했을 때 나타낸 모습이다. 시스템의 생체 신호 다이얼로그의 ECG 비트맵은 디스플레이 되는 구간이 짧기 때문에 ECG를 확인하는 전문의가 좀더 긴 구간 동안 ECG를 확인하고자 할 때 이 컴포넌트가 필요하게 된다.

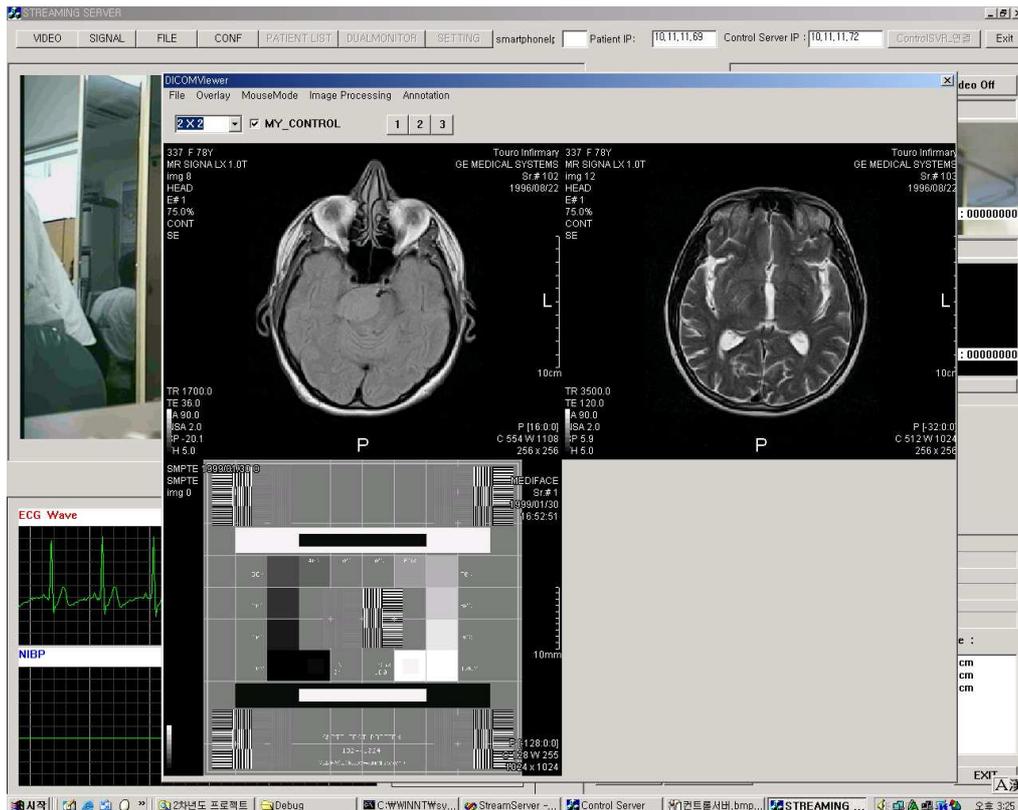


그림 7-9. 스트림 서버, 환자측 , 의사측 시스템 파일 다이얼로그의 DICOM 뷰어 컴포넌트

클라이언트들(환자, 의사, 스트림서버)의 파일 송수신이 이루어졌을 때 전송된 파일 중 DICOM 파일은 환자의 정보와 시간 정보에 의해 따로 저장된다. 그림 7-9는 애플리케이션의 파일 다이얼로그 컴포넌트중 'DICOM VIEWER'을 이용하여 저장된 환자의 DICOM파일을 검색한 후 KDICOM 라이브러리의 함수를 사용하여 상용중인 DICOM 뷰어와 같은 기능을 구현한 모습이다.

7.3 제어 서버

개발된 시스템의 클라이언트 시스템 세션 관리, 스트리밍 데이터 흐름 제어, 환자측 시스템 자원 관리 및 모니터링 기능을 구현하는 제어 서버는 그림 7-10과 같은 애플리케이션 인터페이스를 갖는다.

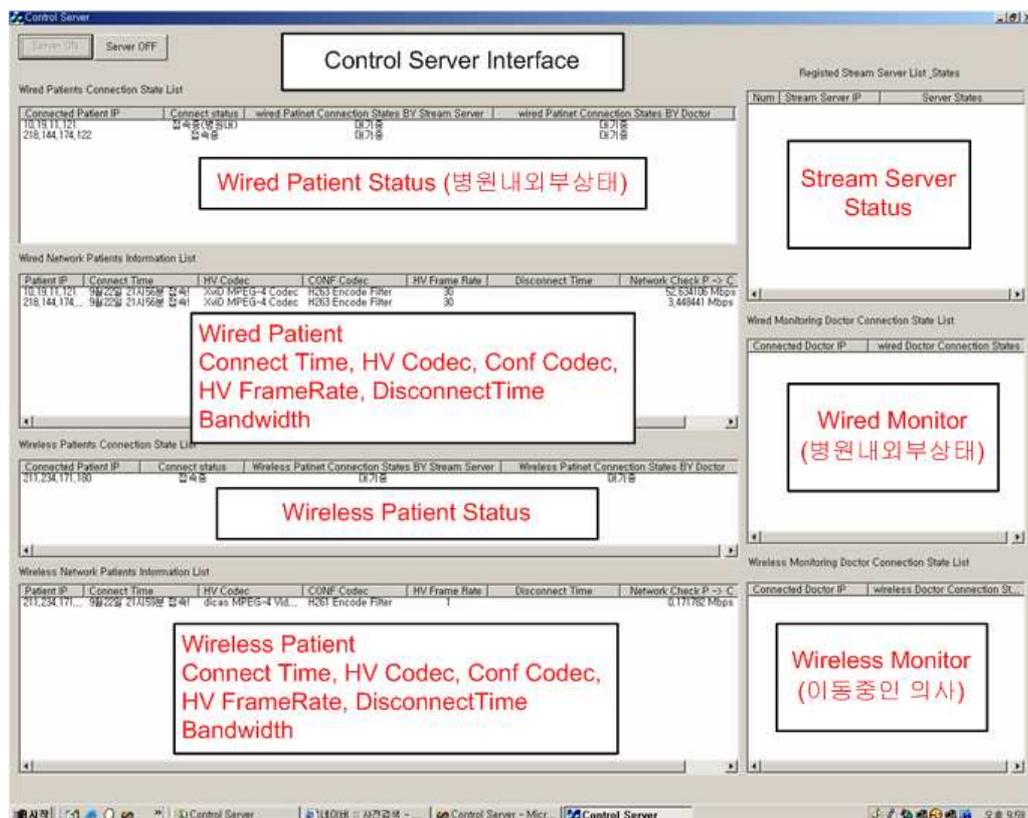


그림 7-10. 제어 서버의 애플리케이션 인터페이스

그림 7-10에서와 같이 환자가 지금 병원 내부에 존재하는지, 병원 외부에 존재하는지 또는 무선망을 사용하는지에 대한 상태 정보와 환자측 시스템에서 사용하는 고화질 비디오 영상 코덱, 프레임 레이트, 화상회의 코덱정보를 확인할 수 있다. 또한 환자측 시스템으로부터 제어서버로 데이터를 전송할 수 있는 대역폭에 관한 정보도 15초마다 업데이트 됨으로써 실시간으로 변하는 네트워크의 변화를 확인할 수 있다. 어떤 환자가 어떤 의사, 어떤 스트림 서버간의 세션을 맺어 진료를 받는지에 관한 정보도 모니터링 할 수 있다.

7.4 환자측 무선망(EVDO In Car) , 의사측 유선망(병원망, ADSL) , 스트림 서버 유선망 (병원망)의 환경에서의 실험



그림 7-11. EVDO망을 통한 환자측 생체신호 및 고화질 영상 전송

그림 7-11은 EVDO망을 연결한 환자측 시스템으로써 EVDO망의 특성상 생체신호 및 실시간 고화질 비디오 영상을 제한적으로 전송하는 모습이다. 그림 7-11과 같은 시스템을 차안에 구성하여 그림 7-12에서와 같이 실제적인 필드 테스트를 실행한 모습이다.



그림 7-12. 차안에서 EVDO 통신을 통한 환자측 생체신호 및 환자이미지 전송

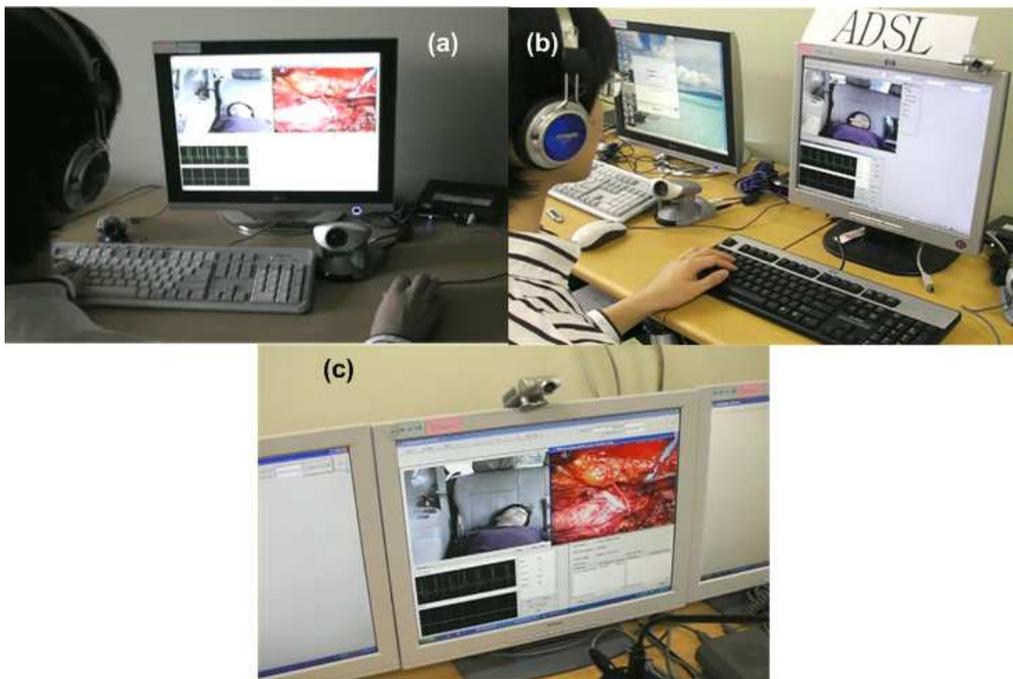


그림 7-13. 무선망의 환자측 시스템이 전송한 데이터를 확인하는 모니터링 시스템

(a) 의사측 병원망 : 병원내에 있는 의사측 시스템 (LAN)

(b) 의사측 병원외부망 : 가정등의 인터넷 상용망에 연결된 의사측 시스템(ADSL)

(c) 스트림 서버 병원망 : 병원 응급실에 위치하면서 여러 환자측 시스템을 모니터링(LAN)

그림 7-13은 그림 7-12에서의 무선망의 환자가 전송한 데이터를 유선망의 의사측 시스템과 스트림 서버가 모니터링하는 모습이다. 환자측 시스템의 EVDO망을 사용한 데이터는 실시간 변화하는 대역폭에 따라 1~2프레임의 고화질 데이터와 생체신호를 전송한다. 파일을 전송할 때는 고화질 영상의 대역폭을 일정한 대역폭으로 줄이고, 나머지 대역폭을 파일 전송에 사용하여 멀티미디어 데이터의 전송에 대한 최대한의 효율을 가진다.

7.5 환자측 유선망(ADSL), 의사측 무선망(EVDO), 스트림 서버 유선망 (병원망)의 환경에서의 실험

환자가 보건소 및 가정에서 인터넷 상용망을 사용하여 시스템에 접속한 경우 또한 의사가 이동중에 EVDO망을 사용하여 환자를 진단하는 경우에 해당하며, 그림 7-14에서와 같이 환자 모니터와 연결된 환자측 시스템에서 COM 포트를 이용하여 생체신호를 획득하고, 이를 ADSL망을 이용하여 TCP프로토콜을 사용하여 전송한다. 고화질 비디오 데이터는 인터넷의 MTU사이즈에 맞게 1460바이트씩 프래그멘테이션하여 전송하게 된다.



그림 7-14. 인터넷 상용망(ADSL)을 사용한 환자측 시스템

그림 7-15는 그림 7-14를 통해 인터넷 망을 통해 전송된 데이터가 제어 서버를 통해 스트림 서버와 EVDO의 의사측 시스템으로 전송되어 디스플레이 되는 모습이다. 특히 (b)에 나타난 이미지는 스트림 서버 및 의사측, 환자측 시스템과 포인터를 공유하여 전문의 간의 의견을 교환할 수 있다.



그림 7-15. 무선망(EVDO)의 의사측 시스템과 병원망내의 스트림 서버
 (a) 의사측 무선망 : 제어서버로부터 환자의 고화질 비디오 영상 및 생체신호수신
 (b) 의사측 무선망 : 제어서버로부터 환자의 이미지 데이터를 수신
 (c) 스트림 서버 병원망 : 제어서버로부터 병원 LAN을 통해 환자의 데이터 수신

7.6 환자측 유선망(병원망), 의사측 유선망(병원망), 스트림 서버 유선망 (병원망)의 환경에서의 실험

환자측 시스템, 의사측 시스템, 스트림 서버 모두 병원 내부에 있는 토폴로지 구성으로써 제어 서버의 정책에 따라 'ByPassing' 기법을 사용하여 환자측 시스템과 직접 데이터를 송수신한다. 병원망의 충분한 대역폭을 사용하여 30프레임의 고화질 영상을 전송하고, 화상회의 비디오, 음성 데이터, 환자 이미지, 생체 신호 데이터를 전송할 수 있다. 그림 7-16은 병원망 내의 환자 시스템을 나타낸다.



그림 7-16. 병원망 내의 환자측 시스템

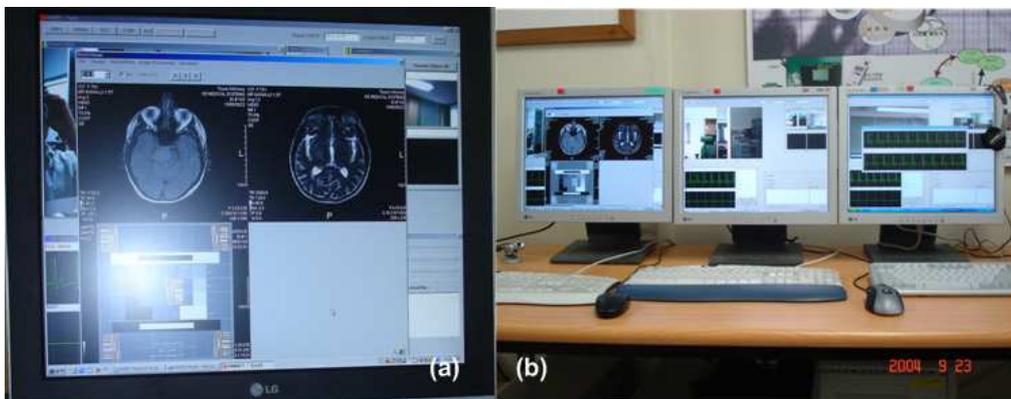


그림 7-17. 병원망 내의 의사측 시스템 과 스트림 서버

(a) 의사측 병원망 : 환자측 시스템으로부터 직접 데이터를 수신하는 병원망내의 의사측 시스템

(b) 스트림 서버 병원망 : 환자측 시스템으로부터 직접 데이터를 수신하는 병원망내의 스트림 서버

제 8장. 결론 및 토의

응급의료 시스템은 초기처치에 따라 환자의 생명과 직결된 의료 시스템이다. 가정이나 산간, 벽지, 농어촌 등의 의료 사각지대에서 응급환자가 발생하였을 경우, 보건소 및 중소병원에서 기초적인 처치만이 가능하기 때문에 3차 의료기관과의 통신 연결을 통해 응급전문의 및 전문의사에 의한 신속한 처치를 요구한다. 여러 명의 환자에 대해 3차 의료기관에서 여러 전문의가 진단하기 위해서는 M(Multi Patients) to M(Multi Consultants) 기반의 통신이 필요하다. 본 논문에서 개발된 유무선 통합 다중 응급원격진료시스템은 다수의 환자와 다수의 전문의 간의 독립적인 세션을 구성하여 여러 명의 환자를 동시에 진단을 받을 수 있게 함으로써, 기존의 1-to-1 방식의 순차적인 의료서비스의 단점을 극복하였다.

전체 시스템은 환자측 시스템, 의사측 시스템, 스트림 서버 및 제어 서버로 구성된다. 다수의 환자와 의사의 세션을 연결해 주기 위해선 제어 서버가 필수적으로 필요하다. 의사측 시스템과 응급실에 위치한 스트림 서버는 다수의 환자와의 데이터통신을 위해 제어 서버로부터 환자에 대한 네트워크 정보 및 환자 정보등을 요구한다. 제어서버로부터 확인된 환자 정보를 바탕으로 스트림 서버 및 의사측 시스템은 진단하고자 하는 환자를 선택하여 제어 서버로 선택된 환자의 정보를 전송하게 된다. 제어 서버는 모니터링 시스템(의사측 시스템, 스트림 서버)으로부터 선택된 환자정보를 바탕으로 'By-Passing' 및 'Flowing' 정책을 결정하게 된다. 'By-Passing' 기법은 세션에 참여한 환자와 모니터링 시스템간의 통신 상태가 환자의 정보를 가공없이 전송할 수 있는 통신 상태일 때 채택되는 방법으로써 병원망내의 환자 및 모니터링 시스템간의 통신이라고 할 수 있겠다. 반면 'Flowing' 기법은 세션에 참여한 환자와 모니터링 시스템간의 통신 상태가 환자가 전송하는 환자 정보 데이터 및 멀티미디어 데이터를 모니터링 시스템이 직접 수신할 수 없을 때 채택되는 방식이다. 예를 들어 환자측 시스템이 통신하는 네트워크망이 병원망의 내부 LAN에 속해있고, 환자 데이터를 수신하는 의사측 시스템이 EVDO 무선망의 환경에 있을 때 제어 서버는 'Flowing' 정책을 채택하고, 환자측 데이터를 수신 가공하여 의사측 시스템으로 전송하거나 환자측 시스템으로 인코딩 파라미터를 전송하는 역할을 하게 된다.

본 논문에서 개발된 시스템은 환자측 시스템에서 데이터를 전송하는데 있어서 다음과

같은 두 가지 제약사항이 있었다. 첫 번째로, 환자측 시스템에서 전송되는 실시간 고화질 영상 데이터는 프로토콜 특성상 UDP를 사용하게 된다. 그러나 일반적인 UDP 프로토콜을 사용하게 되면, UDP 특성상 전송할 수 있는 데이터의 크기는 제한적이기 때문에 고용량의 고화질 영상 데이터는 손실 되고 만다. 환자의 진단에 있어 고화질 영상 데이터의 손실은 오진을 유발할 수 있으므로, 본 시스템에서는 UDP 특성 중 '송수신 되는 데이터는 경계가 있다'라는 특성을 사용하여 헤더를 수정한 새로운 UDP 프로토콜을 개발하여 데이터를 전송하였다.

두 번째는 병원망 내부의 시스템 구성원과 병원망 외부 및 공용 IP를 갖는 무선망 구성원과의 통신을 할 때 병원망의 NAT는 IP의 Encapsulation에 의해 통신장해를 주기 때문에 본 논문에서 개발된 시스템은 터널링을 기반으로 개발되었다. 제어 서버는 공용 IP와 병원망 내부 IP를 가지게 되고, 외부망과 내부망의 통신을 제어 할 때 두 세션 사이의 데이터를 터널링 해줌으로써 NAT문제점을 해결하였다.

유무선 통합 다중 응급 원격진료시스템은 다음과 같은 연구가 추가적으로 보완되어야 할 것이다. 제어 서버의 'Flowing'기법을 통해 세션을 맺은 환자측 시스템과 모니터링 시스템이 제어 서버의 시스템 다운등의 상황이 발생했을 때, 모두 세션이 끊기는 경우가 발생할 수 있기 때문에 이에 대한 연구가 필요하다.

참 고 문 헌

- [1] Y. C. Lu, Y. Xiao, A. Sears, J. A. Jacko, "A review and a framework of handled computer adoption in healthcare," *International Journal of Medical Informatics*, 2005, 74, pp. 409-422.
- [2] E. S. Berner, B. Adams, "Added value of video compared to audio lectures for distance learning," *International Journal of Medical Informatics*, 2004, 73, pp. 189-193.
- [3] E. A. Mendoca, E. S. Chen, P. D. Stetson, L.K. Mcknight, J. Lei, J. J. Cimino, "Approach to mobile information and communication for healthcare," *International Journal of Medical Informatics*, 2004, 73, pp. 631-638.
- [4] J. R. Barrett, S. M. Strayer, J. R. Schubart, "Assessing medical residents' usage and perceived needs for personal digital assistants," *International Journal of Medical Informatics*, 2004, 73, pp. 25-34.
- [5] J. S. Lee, C. T Tsai, C. H. Pen, H. C. Lu , "A real time collaboration system for teleradiology consultation," *International Journal of Medical Informatics* ,2003, 72, pp. 73-79.
- [6] G. Demiris, S. Vijaykumar, "A comparison of communication models of traditional and video-mediated health care delivery," *International Journal of Medical Informatics*, 2005, xxx, pp. 1-6.
- [7] H. Muller, N. Michoux, D. Bandon, A. Geissbuhler, "A review of content-based image retrieval systems in medical applications-clinical benefits and future diretions," *International Journal of Medical Informatics*, 2004, 73, pp. 1-23.
- [8] H. Takeda, Y. Matsumura, S. Kuwata, H. Nakano, J. Shanmai, Z. Qiyan, C. Yufen, H. Kusuoka, M. Matsuoka, "An assessment of PKI and networked electronic patient record system: lessons learned from real patient data exchange at the platform of OCHIS(Osaka Community Healthcare Information System)," *International Journal of Medical informatics*, 2004, 73, pp. 311-316.

- [9] P. Cao, M. Hashiba, K. Akazawa, T. Yamakawa, T. Matsuto, "An integrated medical image database and retrieval system using a web application server," *International Journal of Medical Informatics*, 2003, 71, pp. 51-55.
- [10] Sun K. Yoo, Sun H. Kim, Nam H. Kim, Y.T. Kang, K.M. Kim, S.H. Bae, Michael W. Vannier, "Design of a PC-based mulitmedial telemedicine system for brain function teleconsultation," *International of Journal of Medical Informatics*. 2001, 61, pp. 217-227.
- [11] 미키 스케이치, 고성제, 김종욱, "MPEG-4의 세계," *영풍문고*, 1999, pp. 25-173.
- [12] Y. W. Lee, F. Kossentini, R. Ward, M. Smith, "Towards MPEG4: An improved H.263-based video coder," *Signal Processing: Image Communication* , 1997, 10, pp. 143-158.
- [13] A.C. Downton, "Speed-up trend analysis for H.261 and model-based image coding algorithms using a parallel-pipeline model," *Signal Processing: Image Communication* ,1995, 7, pp. 489-502
- [14] A. Drigasa, S. Kouremenosa, Y. Bakopoulou, V. Loumosb, "A study of H.263 traffic modeling in multipoint videoconference sessions over IP networks," *Computer Communications* , 2005, 5 , pp. 1-20
- [15] J. Chabriaux , B. Gibaud, "DICOM, the standard for medical imaging," *EMC-Radiologie* ,2004, 1, pp. 577-603.
- [16] 신화선, "DirectShow 멀티미디어 프로그래밍," *한빛미디어*, 2002, pp. 174-191.
- [17] 신화선, "DirectShow 멀티미디어 프로그래밍," *한빛미디어*, 2002, pp. 508-518.
- [18] 윤성우, "TCP/IP 소켓 프로그래밍," *프리렉*, 2003. pp. 53-300.
- [19] 김선우, "윈도우 네트워크 프로그래밍," *한빛미디어*, 2004, pp. 24-139.
- [20] 김정훈, "TCP/IP 소켓 프로그래밍," *교학사*, 2003, pp. 72-83.

ABSTRACT

Design of Multiple Emergency Telemedicine System in Wired and Wireless Integrated Network

Kang, Ho Hyun
Graduate program in
biomedical engineering
The Graduate School
Yonsei University

The rapid progress of the computing capabilities and the network bandwidth could provide high quality medical service like telemedicine even in the medical profession. Unlike the existing medical service which the doctor provided for the patients directly, the telemedicine provides the high quality medical service to the patients at the long distance or the patients being transferred to the hospital with constructing the medical system consists of the tele-consultation and the tele-diagnosis through the wired-wireless network environment.

When an emergency patient occurs in the area where the proper first aid is impossible, the telemedicine service connects the doctors of second, third medical centers to the patient. Then the doctors can provide tele-consultation and tele-diagnosis. Even for the patient in the ambulance, this telemedicine service play an important role since delivering the patient's data precisely at realtime affects the life of the patient.

The research of emergency telemedicine service can be divided into 2

category according to the place where the service is made; between one medical center and another and between the medical center and home. This research has supplied the service one after the other with the one to one communication configuration. For this, the emergency telemedicine service system in the wired network environment was required. However the progress of the IT technology and the growth of the wireless-internet and communication system has made the real time telemedicine be possible in wired-wireless network environment. This telemedicine needed a different system construction considering the different network environments.

Therefore the emergency telemedicine service system subordinate to the existing wired network environment should reconstruct the telemedicine system which considers the wireless network environment with the low-bandwidth. Also, since the existing one to one communication configuration performs the telemedicine only according to the connection orders of the patients, it could not provide proper service the patients in the state of emergency. So the function which connects multi-patients to multi-doctors is required for the emergency telemedicine service system.

The purpose of this thesis is to suggest a better emergency telemedicine service system. This system will transmit the patient's data at real time to the doctors in the second, third medical centers so that they could provide tele-consultation to the patients. It doesn't matter where that emergency patient occurs; in wired network environment, in the district public health center, or even in the prison. At this point, the networks environment of the doctors should locate in the LAN of the hospital. If doctors are at their home, they can use the VDSL or ADSL for the telemedicine. Also if the doctors are in move, they can use the wireless network like CDMA 1x EV-DO(Code Division Multiple Access), Wireless

Lan for it. Then unlike the existing emergency telemedicine service system, this system can provide high quality telemedicine service even in the different network environments.

In this research, I construct a system which provides the patient's data to the multi-doctors at real time where the wired and wireless network environment is mixed and confirms the availability of this emergency telemedicine service system.

key words : telemedicine, tele-diagnosis , tele-consultation, emergency telemedicine service system , VDSL, ADSL, LAN, CDMA 1x EV-DO, Wireless Lan