

무선환경에서의 RTP를 이용한 실시간 원격
진료 시스템의 설계 및 평가

연세대학교 대학원
생체공학협동과정
전기전자공학전공
윤 하 영

무선환경에서의 RTP를 이용한 실시간 원격
진료 시스템의 설계 및 평가

지도 김 남 현 교수

이 논문을 석사 학위논문으로 제출함

2006년 1월 일

연세대학교 대학원

생체공학협동과정

전기전자공학전공

윤 하 영

<인준서 양식>

윤하영의 석사 학위논문을 인준함

심사위원 _____ 인

심사위원 _____ 인

심사위원 _____ 인

연세대학교 대학원

2006년 1월 일

감사의 글

어떤 때보다 빠르게 지나간 석사 생활을 이 논문으로 마무리 짓게 되었습니다. 힘들었던 만큼 많은 것을 배웠다는 만족감과 조금 더 열심히 하지 못했다는 아쉬움이 교차합니다. 언제나 저의 옆에서 많은 관심과 사랑을 주신 여러분께 감사의 말씀을 짧게 전하러 합니다.

먼저 한없이 부족한 저에게 끊임없는 관심을 가져주시고 많은 것을 가르쳐 주신 유선국 교수님께 감사의 말씀을 전합니다. 또한 언제나 저를 지켜봐 주시며 많은 도움을 주신 김남현 교수님과 신경외과 김선호 교수님께도 정말 감사드립니다.

언제나 힘들고 어려운 일을 하시면서도 저를 아껴 주시고 예뻐해 주신 동근이형, 너무 죄송하고 감사드립니다. 아무것도 모르던 연구실 생활 초창기에 많은 도움과 교감을 나눠 주셨던 병수형, 고마워요. 썰렁했던 연구실을 웃음으로 채워주신 우리 민규형, 허리 빨리 나으세요. 망사 패밀리 정진이와 국진이형, 기원이형!!! 올해는 좋은 일만 있자구요. 그리고 형으로써, 선배로써 많은 도움을 주고 싶었지만 잘 못해준 동헌아, 내무반에서 기다릴께. 구염동이 수정인 남자 친구 생겨도 날 버리지 말아주길 바란다. 서로 너무 달라서 매력적인 동갑내기 진호, 내 사랑 충기, 영재야 너무 고맙다. 멀리 있어서 자주 보지는 못했지만 너무 정겹고 좋은 배신자 재홍이 형과 덕찬이, 윤정이, 올해는 자주 봐요. 조금 늦게 알았지만 가슴 깊이 저를 챙겨주신 순만이형, 너무 감사드립니다. 그리고 마지막으로 호현이 형에게 남다른 감사의 인사를 드리고 싶습니다. 형을 만난 것은 정말 저에게 큰 정말 행운이자 기쁨이었습니다.

십수 년 간 저의 가장 큰 버팀목이자 경쟁상대가 되어준 친구들인 배르주아 석우, 제일 부러운 선형, 귀염둥이 성구, 내 애인 승준, 사랑스런 삼질맨 환빈, 언제나 한결 같은 현오(빠질까봐 가나다 순으로 한다.)야 너무 고맙다. 미국에 있는 태영아, 멀리 있지만 항상 내 마음에 있다. 무어라 표현할 수 없는 애정을 갖고 있는 진상 패밀리 원형, 형준아 너무너무 고맙다. 그리고 누구보다도 완벽한 synchronized frequency를 가지고 내가 가장 힘들고 괴로울 때 옆에 있어준 준구야!!!! 고맙고 사랑한다. 그리고 H씨도.....

마지막으로 언제나 내 삶의 기쁨과 영감이 되어준 누나, 올바르게 최선을 다하는 삶이 무엇인지 알려주신 어머니, 당신의 모든 것 하나하나가 제 삶의 기쁨이자 목표가 되어주신 아버지께 저의 감사와 사랑을 전합니다. 제가 가진 이 감사의 마음으로 더욱 노력하여 열심히, 그리고 올곧게 살아가겠습니다. 항상 건강하십시오.

윤하영 올림

차 례

| | |
|---|-----|
| 그림 차례 | iii |
| 표 차례 | iii |
| 국문 요약 | vi |
| 제1장 서론 | 1 |
| 1.1 연구 배경 | 1 |
| 1.2 논문 구성 | 3 |
| 제2장 인터넷을 통한 실시간 비디오 전송 | 4 |
| 2.1. 실시간 비디오 전송의 개요 | 4 |
| 2.2. 체증 제어(Congestion Control) | 6 |
| 2.2.1 전송률 제어 | 7 |
| 2.2.2 적응적 비디오 부호화 | 9 |
| 2.3. TCP 친화적 체증 제어(TCP-Friendly congestion control) | 10 |
| 2.3.1 TCP 친화적 체증 제어의 개요 | 10 |
| 2.3.2 TCP-Reno의 체증 제어 방법 | 10 |
| 2.3.3 TCP 친화적 전송률 추정 방법 | 12 |
| 제3장 RTP/RTCP | 14 |
| 3.1 실시간 수송 프로토콜(RTP) | 14 |
| 3.1.1 실시간 수송 프로토콜(RTP)의 특징 및 기능 | 14 |
| 3.1.2 실시간 전송 프로토콜(RTP) 패킷의 구조 | 15 |
| 3.2 실시간 수송제어 프로토콜 (RTCP) | 18 |
| 3.2.1 실시간 수송제어 프로토콜(RTCP)의 특징 및 기능 | 18 |
| 3.2.2 실시간 수송제어 프로토콜(RTCP) 패킷의 구조 | 19 |
| 제4장 CDMA 1X-EVDO 환경 하에서의 가변율 비디오 전송 원격의료 시스템 구현 | 25 |
| 4.1 CDMA 1X-EVDO | 25 |
| 4.2 시스템의 기본 개념 | 27 |
| 4.3 서버/클라이언트 메커니즘 | 29 |
| 4.4 RTP/RTCP 프로토콜의 적용 메커니즘 | 34 |
| 4.5 RTP 타임스탬프와 NTP 타임 스탬프와 지터(Jitter) 계산 | 35 |
| 4.5.1 RTP 타임 스탬프 | 35 |

| | |
|--|----|
| 4.5.2 NTP 타임 스탬프 | 36 |
| 4.5.3 지터 (Jitter) 계산 | 37 |
| 4.6 왕복지연 시간(RTT)과 패킷손실률 추정 메커니즘 | 38 |
| 4.6.1 왕복 지연 시간 측정법 | 38 |
| 4.6.2 패킷 손실률 측정법 | 49 |
| 4.6.3 패킷 손실률과 왕복지연 시간의 이동 평균(Moving Average) | 40 |
| 4.7 패킷 우선 전송 알고리즘 | 41 |
| 4.8 시스템의 구현내용 | 42 |
| 4.9 실험 환경 | 44 |
| 제5장 실험 결과 | 45 |
| 5.1 CDMA 1X-EVDO 상에서 원격 진료 시스템 평가 | 45 |
| 제6장 결론 및 토의 | 49 |
| 참고 문헌 | 51 |
| Abstract | 54 |

그림 차례

| | |
|--|----|
| 그림 2.1 실시간 비디오 전송 시스템의 계층적 구조 | 7 |
| 그림 2.2 TCP—RENO의 congestion window 변화 모습 | 11 |
| 그림 3.1 RTP 프로토콜 스택 구조 | 15 |
| 그림 3.2 RTP 패킷의 구조 | 16 |
| 그림 3.3 송신자 보고(SR:Sender Report) 용 RTCP 패킷의 구조 | 21 |
| 그림 3.4 SDES : 소스 설명 RTCP 패킷 | 22 |
| 그림 3.5 탈퇴(Bye) 패킷의 구조 | 23 |
| 그림 3.6 응용(APP) 패킷의 구조 | 24 |
| 그림 4.1 서버/ 클라이언트 모델 | 27 |
| 그림 4.2 가변율 비디오 전송 (VBR video transmission)시스템 | 28 |
| 그림 4.3 UDP를 사용하는 클라이언트 서버 Winsock 함수 흐름도 | 30 |
| 그림 4.4 TCP를 사용하는 클라이언트 서버 Winsock 함수 흐름도 | 31 |
| 그림 4.5 UDP 수송 방식에 의한 송수신 시나리오 | 33 |
| 그림 4.6 RTP/RTCP 프로토콜의 적용 메커니즘 | 34 |
| 그림 4.7 왕복 지연 시간(RTT) 획득 알고리즘 | 39 |
| 그림 4.8 CDMA 1X-EVDO 망을 사용한 원격 진료 시스템 개념도 | 42 |
| 그림 4.9 그림 4.9 (a)원격 진료 시스템 송신단 화면 (b) RTCP 서버 화면 | 43 |
| 그림 5.1 CDMA 1X-EVDO 망에서 송신 시 대역폭과 패킷 에러 분포도 | 45 |
| 그림 5.2 CDMA 1X-EVDO 망에서 송신 시 왕복 지연 시간 값 | 46 |
| 그림 5.3 손실률의 순간값과 이동 평균값 변화 | 47 |
| 그림 5.4 전송 실험 시 평균 대역폭 분포도 | 48 |

표 차례

| | |
|---|----|
| 표 5.1 각 프로토콜 전송 실험 시 PSNR(Peak Signal to Noise Ratio) 값 | 48 |
|---|----|

국문 요약

환경의 제약이 없는 인터넷 기반의 실시간 멀티미디어 서비스를 이용한 원격 의료는 시, 공간의 제약 없이 환자를 진단하고 적절한 조치를 취할 수 있게 하여 환자의 생존율, 회복률에 긍정적 효과를 끼친다. 이를 위해 응급 의료 서비스는 다양한 전문의의 종합적이고 동시적인 서비스가 제공되어야 한다. 인터넷을 통하여 실시간 비디오를 전송하기 위해서는 채널의 충분한 대역폭, 적은 지연과 적은 패킷 손실 등의 조건이 보장되어야 한다. 그러나 현재의 인터넷은 비디오 전송에 필요한 QoS (Quality of Service)를 만족시키기 위해 네트워크 계층에서 어떠한 기능도 제공하지 못한다. 따라서 사용자가 원하는 서비스의 품질 보장은 네트워크 계층 상위에서 수행되어야만 한다. 이를 위해 제안된 것이 수송계층(transport layer) 위에서 동작하는 실시간 수송 프로토콜(RTP : Real-time Transport Protocol)과 실시간 수송제어 프로토콜(RTCP : Real-time Transport Control Protocol)이다. 이들은 응용 계층과 수송계층 사이에서 패킷 생성 시 삽입되는 부분이라고 이해하는 것이 타당할 것이다. 실시간 수송 프로토콜(RTP)과 실시간 수송제어 프로토콜(RTCP)을 사용하면, 비디오 전송에서 시간제약에 따른 특성을 고려해 줄 수 있고, 네트워크 내에서 발생하는 손실에 대처할 수 있다.

본 논문에서는 RTP 프로토콜을 사용한 실시간 원격 의료 진료시스템을 설계, 구현하였다. 우선 의료 정보 데이터에 실시간성을 부여하기 위하여 UDP 기반에 RTP를 적용하였다. 무선망에서 TCP를 사용하였을 경우 실시간성이 완벽히 보장되지 않고 무선망에서의 링크 에러를 병목현상 등의 다른 에러로 인식하기 때문에 불필요한 윈도우사이즈 조절로 인한 대역폭 낭비현상이 있었다. 반면 UDP는 무조건적인 전송으로 인한 비 신뢰적인 전송과 QoS가 보장 되지 않는 단점이 있다. 때문에 본 시스템에서는 대역폭을 충분히 활용할 수 있는 UDP 기반에서 RTP 프로토콜을 적용하여 실시간성을 부여하고 중요한 환자 데이터는 우선권을 부여하여 전송하였다. 이와 같은 알고리즘을 사용함으로써 응용계층 아래 단에서의 실시간성 보장 효과를 확인할 수 있었다.

두 번째로는 네트워크의 상태에 유동적으로 대응하기 위한 전송률 제어 알고리즘을 제안하였다. 정확하고 주기적인 네트워크 성능 수집을 위하여 RTCP를 사용하였다. RTCP는 현재 네트워크의 대역폭, 지연, PER 등의 정보를 수집하여 최대 전송 보장 가능한 데이터 사이즈를 송신단에게 알려준다. 주어진 정보를 분석하여 적합한 전송률을 결정, 전송함으로써 안정적인 대역폭을 사용할 수 있고, 무조건적인 전송으로 인한 병목현상을 방지함으로써 환자 영상 품질의 보장에 우월한 성능을 보일 수 있음을 확인하였다.

핵심 되는 말 : 원격 진료, 실시간, CDMA 1X-EVDO, UDP, RTP, RTCP, 전송률 조절

1. 서론

1.1 연구 배경

세계 각국의 네트워크 광대역화로 인터넷 접속 속도가 비약적으로 증가함에 따라 멀티미디어 서비스에 대한 사용자들의 욕구가 증가하고 있다. 환경의 제약이 없는 인터넷 기반의 실시간 멀티미디어 서비스를 이용한 원격 의료는 시, 공간의 제약 없이 환자를 진단하고 적절한 조치를 취할 수 있게 하여 환자의 생존율과 회복률에 긍정적 효과를 끼치므로, 의료 분야에서 중요한 화두로 자리 잡아 가고 있다[12].

원격 진료 시스템과 같은 멀티미디어 서비스의 핵심적 미디어인 비디오는 실시간성과 많은 정보량으로 특징 지워진다. 실시간 비디오의 경우 부호화 시점과 재생 시점이 일정한 시간 관계를 유지해야만 만족할 만한 품질의 비디오를 재생할 수 있고 궁극적으로 응급환자 발생시 신속한 처치를 기대할 수 있다. 인터넷을 통하여 실시간 비디오를 전송하기 위해서는 채널의 충분한 대역폭, 적은 지연과 적은 패킷 손실 등의 조건이 보장되어야 한다. 그러나 현재의 인터넷은 비디오 전송에 필요한 QoS (Quality of Service)를 만족시키기 위해 네트워크 계층에서 어떠한 기능도 제공하지 못한다. 또한 병원 내부망, ADSL, 무선망 등과 같은 이질적 성능의 상용 네트워크에서는 균등한 품질의 서비스를 제공하는 것에 많은 무리가 있다. 특히 에러가 많은 무선망의 경우 TCP(Transmission Control Protocol)로 전송하면 손실 데이터의 재전송과 전송 윈도우 사이즈 축소로 인한 전송률이 저하 되고, UDP(User Datagram Protocol)로 전송할 경우에는 실시간성을 보장하는 대신 중요한 데이터의 누락을 야기할 가능성이 있는 trade off 현상이 발생한다. 따라서 사용자가 원하는 서비스의 품질 보장은 네트워크 계층 상위에서 수행되어야만 한다. 이를 위해 제안된 것이 수송계층(transport layer) 위에서 동작하는 실시간 수송 프로토콜(RTP : Real-time Transport Protocol)과 실시간 수송제어 프로토콜(RTCP : Real-time Transport Control Protocol)이다. 이들은 응용 계층

과 수송계층 사이에서 패킷 생성 시 삽입되는 부분이라고 이해하는 것이 타당할 것이다. 실시간 수송 프로토콜(RTP)과 실시간 수송제어 프로토콜(RTCP)을 사용하면, 비디오 전송에서 시간제약에 따른 특성을 고려해 줄 수 있고, 네트워크 내에서 발생하는 손실에 대처할 수 있다[6].

실시간 수송 프로토콜(RTP)과 실시간 수송제어 프로토콜(RTCP)을 사용하여 네트워크의 전송률을 제어 할 수 있는 방법 중의 하나가 TCP 친화적 체증제어(TCP-Friendly congestion control) 방식이다. 현재의 네트워크에서는 여전히 많은 데이터들이 신뢰성 있는 전송을 위하여 TCP를 사용하고 있다. TCP는 네트워크 혼잡이 발생했을 때 패킷 손실률을 줄이기 위해 전송률 제어를 하는데 이것이 혼잡제어(Congestion Control)이다. 반면 멀티미디어 정보의 전송에는 TCP대신 실시간성을 보장해 줄 수 있는 UDP를 사용한다. 그러나 UDP는 체증제어를 수행하지 않기 때문에 이러한 UDP 데이터를 같은 네트워크 망에서 아무런 제약 없이 TCP와 혼합해서 사용하면 TCP 데이터의 수율에 많은 악영향을 주거나 UDP자체의 비 신뢰적 전송으로 인하여 중요한 데이터가 누락될 수도 있다. 따라서 UDP를 통해 데이터를 전송할 때에도 혼잡제어가 필요하며 그 방식을 TCP와 유사한 방식으로 하는 혼잡제어 방식이 TCP 친화적 혼잡제어(TCP-Friendly congestion control) 방식이다. 이러한 TCP 친화적 혼잡제어(TCP-Friendly congestion control)를 수행하기 위하여 위에서 언급한 실시간 수송 프로토콜(RTP)/ 실시간 수송제어 프로토콜(RTCP)을 사용하는 것이다. 이와 같이하면 UDP를 사용하여 실시간성을 보장해 주는 동시에 네트워크의 혼잡제어도 수행하게 되어 전체 데이터의 수율에 큰 영향을 미치지 않게 되는 것이다.

본 논문에서는 앞에서 기술한 여러 가지 이슈들을 사용하여 실시간 비디오 전송 시스템을 구축하고 각종 테스트를 통하여 기존 알고리즘의 효용성을 검증하며, 실제 필드 테스트를 통하여 실제 적합성을 검증한다. 또한 체증제어를 위한 여러 방법들을 고찰해 보고, 차후과제로서 이번연구에서의 문제점과 해결방안 및 유선과 무선 네트워크가 결합된 환경에서 전송방법의 문제점을 다루어 보고자 한다.

1.2 논문 구성

논문의 구성은 다음과 같다. 2장에서는 인터넷을 통한 실시간 비디오 전송의 개념과 네트워크 혼잡제어의 정의 및 방법, 네트워크 혼잡제어의 한 가지 방법으로서 TCP 친화적 혼잡제어(TCP-Friendly congestion control) 방법에 대해서 살펴본다. 3장에서는 실시간 수송 프로토콜(RTP)과 실시간 수송 제어 프로토콜(RTCP)의 개념과 기본포맷, 이들 프로토콜의 활용방법에 대해서 살펴본다. 4장에서는 CDMA 1X-EVDO 환경 하에서 비디오 전송시스템의 전체적인 개념과 구성 및 기능에 대해서 자세히 살펴본다. 제 5장에서는 field test를 통하여 제안한 방법의 효율성을 측정 분석해보고, 6장에서는 결론과 함께 차후 과제를 살펴본다.

2. 인터넷을 통한 실시간 비디오 전송

2.1 실시간 비디오 전송의 개요

인터넷(Internet)을 통한 실시간 비디오의 전송은 많은 인터넷 멀티미디어 응용 분야에서 중요한 부분을 차지하고 있다. 실시간 비디오의 효율적인 전송을 위해서는 충분한 대역폭과 함께 전송 지연 및 패킷 손실이 작아야 한다. 하지만 현재의 인터넷은 비디오 전송을 위해 어떠한 QoS(Quality of Service)도 보장하지 않는다. 또한 많은 네트워크 환경 및 사용자 시스템의 이질성은 인터넷을 통한 비디오 전송을 매우 어렵게 하고 있다.

인터넷을 통한 실시간 비디오 전송에서 주요한 과제가 되는 QoS 이슈들은 다음과 같다. 첫째가 대역폭 문제이다. 사용자가 수용할 수 있는 재생 화질을 얻기 위해서 실시간 비디오는 최소한의 대역폭 제한을 가져야 한다. 둘째가 전송지연 문제이다. 실시간 비디오는 시간제약이 엄격하다. 첫 화면이 재생되면 그 다음 화면부터는 재생되는 시점이 정해지게 되어 일정 시간기준을 넘어서 들어오는 데이터는 손실과 똑같이 처리된다. 셋째가 패킷손실 문제이다. 패킷손실이 발생하면 전송된 영상의 일그러짐이 발생하고, 손실이 더욱 많아지면 지속적인 재생마저 어렵게 된다. 따라서 손실률이 일정한 범위 내에 있어야만 좋은 화질을 얻을 수 있다.

위에 설명한 QoS 이슈들과 더불어 인터넷에서 비디오를 전송할 때 이질성의 문제가 발생한다. 이것은 네트워크의 이질성과 수신단의 이질성을 분류된다. 첫째로 네트워크의 이질성이란 라우터의 처리능력, 네트워크의 가용 대역폭, 수용 능력, 체증 제어방식에서 서로 다른 네트워크들을 통해 비디오를 전송하면 수신단에서는 서로 다른 패킷 손실과 전송지연을 얻게 된다는 것이다. 둘째로 수신단의 이질성은 수신단 시스템이 서로 다른 성능을 가지고 있고, 또한 각 수신단마다 원하는 바가 서로 다르다는 것이다.

인터넷을 통하여 비디오를 전송할 때 채널에 과부하가 걸리면 라우터에서 패킷이 버려지거나 큐에서 서비스를 기다리는 패킷이 생기게 된다. 이를 체증

(Congestion)이라 한다. 패킷 손실이 발생하면 수신 단에서 재생된 비디오의 화질이 크게 저하되고, 큐에서의 지연이 증가하게 되면 전송지연의 제약조건을 넘어서게 되어 결국 쓸모없는 패킷이 될 수 있다 따라서 송신 단은 이러한 체증이 발생하지 않도록 비디오의 전송률을 조절해야 한다. 이러한 일련의 과정을 흐름 제어(flow control) 또는 체증 제어(congestion Control)라 한다.

이 장에서는 이러한 체증제어(congestion control) 방법에 대해서 살펴본다. 2.2절에서는 체증제어에 관하여 현재 연구되고 있는 내용을 설명하고 2.3절에서는 체증제어(congestion control)의 한 방법인면서 이번 시스템에서 구현한 TCP 친화적 체증제어 (TCP-Friendly congestion control) 방법의 개념과 내용에 대해서 설명한다.

2.2 체증 제어(Congestion Control)

체증제어는 전송률 제어(rate control)와 적응적 비디오 부호화(rate-adaptive video encoding)의 두 부분으로 나뉘어진다. 전송률 제어는 채널에서의 체증을 방지하기 위해 전송률을 조절하는 것이고, 적응적 비디오 부호화는 결정된 전송률에 맞는 비트량만큼 부호화 양을 조절하는 것이다.

그림 2.1은 실시간 비디오 전송 시스템의 구조를 나타낸다. 실시간 비디오 전송 시스템의 송신단과 수신단은 압축계층과 전송계층으로 나뉘어진다. 송신단에서 압축계층(compression layer)을 통해 압축된 비트 스트림은 RTP/ UDP/ IP 계층을 거쳐 인터넷을 통해 수신측으로 전달된다. 여기서 실시간 전송 프로토콜(RTP : real-time transport protocol)은 전송계층과 응용계층 사이에서 동작하며, 전송과정에서 아무런 제어를 수행할 수 없는 UDP의 단점을 보완하는 역할을 한다. 송신단에서 전송되는 패킷은 인터넷에서 손실되거나 시간지연 초과로 인해 수신측에서 버려질 수 있다. 그 외 제 시간에 도착한 패킷들은 IP/ UDP/RTP 계층을 거쳐 비디오 복호기 (video decoder)로 전달되어 복호된다. 수신측은 패킷을 전달 받은 후 실시간 전송 프로토콜(RTP) 헤더에 있는 각종 정보를 통하여 패킷 손실률과 지연 등 네트워크의 상태에 대한 장보를 계산하여 QoS(Quality of Service)를 모니터링 한다. 모니터링된 정보는 실시간 전송제어 프로토콜(RTCP)을 통해 송신단에 피드백된다. 송신단은 피드백된 정보를 이용하여 현재 채널의 가용 대역폭을 예측하고, 그 값을 적응적 비디오 부호화기로 이를 전달한다. 예측값을 전달받은 비디오 부호화부는 예측된 대역폭에 맞게 스트림의 출력 부호화율을 조절하게 된다.

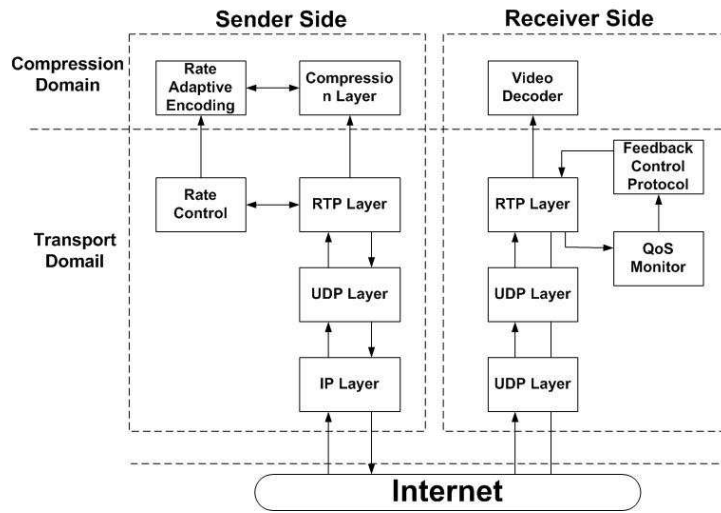


그림 2.1 실시간 비디오 전송 시스템의 계층적 구조

2.2.1 전송률 제어

전송률 제어는 방법에 따라 실험적 전송률 제어(probe-based rate control), 모델기반 전송률 제어(model-based rate control), 수신단 기반 전송률 제어(receiver-based rate control), 계층적 부호화에 의한 전송률 제어(rate control by layered coding), 복합 전송률 제어(hybrid rate control)등으로 분류된다[14]. 모델기반 전송률 제어는 "TCP 친화적 (TCP-Friendly) 전송률 제어"라고도 불린다.

가. 실험적 전송률 제어 (probe-based rate control)

송신측이 전송률을 조정하면서 네트워크의 가용 대역폭을 조사하는 방식이다.

단계 1 . 최소한의 화질을 얻을 때의 패킷 손실률을 임계치 P_{th} 로 정한다.

단계 2 : 초기에 R의 전송률로 데이터를 전송하여 패킷 손실률 P를 구한다.

단계 3 : 패킷 손실률 P가 P_{th} 보다 크면 전송률을 낮추고, P가 P_{th} 보다 작으면 높여서 점차적으로 네트워크의 가용 대역폭을 찾아간다.

전송률 R 값을 높이고 낮추는 방법은 AIMD (additive increase and multiplicative decrease)[2] 나 MIMD (multiplicative increase and multiplicative decrease)[3]를 사용한다. 이 방법은 체증상태에 따라 계속적으로 적응을 해가는 방법이므로 체증을 방지할 수 있는 장점이 있으나 전송률 R 이 $P = P_{th}$ 근처에서 증감을 반복할 수 있고, 초기에 R 값이 안정화되는데 시간이 걸리며, 채널의 가용 대역폭에 최적화된 값으로 안정화되지 않는다는 단점을 가진다.

나. 모델기반 전송률 제어(model-based rate control)

모델 기반 전송률 제어 방법은 TCP의 수율 모델에서 유도된 식을 사용하여 네트워크의 가용 대역폭을 예측한다. 이 방법에 대한 자세한 내용은 2.4 절에서 기술한다.

이 방식은 TCP의 데이터 흐름을 거스르지 않고 친화적으로 데이터를 보낸다고 하여 "TCP-친화적(TCP-Friendly)"이라고 부르며 데이터의 증감이 심한 TCP의 데이터 흐름을 따르기 때문에 전송률이 크게 증감할 위험이 있다. 따라서 네트워크의 상태가 심하게 변하지 않는다는 가정 하에 유용한 방식이다.

다. 수신단 기반 전송률 제어(receiver-based rate control)

앞에 설명한 두 방식은 송신단에서 전송률을 결정하는데 반해 이 방식은 수신단에서 전송률을 결정한다. 이 방식은 멀티캐스트 전송에서 이질성(heterogeneity) 문제의 해결에 초점을 두고 있다. 송신단은 비디오를 계층적으로 부호화하여 각 계층마다 다른 IP 멀티캐스트 그룹으로 전송하면, 각 수신단은 자신들의 전송률을 스스로 결정하고 그에 맞는 멀티캐스트 그룹에 참여함으로써 각 수신단마다 자신들의 상황에 맞는 전송률로 데이터를 수신하게 되는 것이다.

라. 계층적 부호화에 의한 전송률 제어(rate control by layered coding)

계층적 부호화는 비디오를 여러 개의 계층(layer)으로 나누어서 부호화한다. 재생을 위한 최하위 계층은 기본 계층(base layer)이며, 그 위의 계층은 향상 계층(enhancement layer)이다. 기본 계층은 최소한의 화질을 가지며 독립적으로 재생

가능하지만, 항상 계층은 기널 계층의 데이터가 없으면 재생이 불가능하며, 항상 계층이 늘어감에 따라 비디오의 화질은 향상된다. 따라서 가용 대역폭에 따라 기본 계층에 덧붙여 항상 계층의 수를 늘려가면서 데이터를 전송한다. 이 방식은 앞에서 설명한 수신단 기반 전송률 제어방식과 함께 사용되어 멀티캐스트 비디오의 전송 성능을 향상시킨다.

마. 복합 전송률 제어 (hybrid rate control)

수신단은 수신 채널 수(수신하는 IP 멀티캐스트 그룹의 수)를 통하여 수신률을 정할 수 있고, 송신단은 피드백 정보에 기반하여 각 채널(각 If 멀티캐스트 그룹)의 전송률을 조정할 수 있다. 송신단 기반 제어방식과의 차이점은 한 개의 채널로 데이터를 전송하는 송신단 기반 방식과 달리 복합 전송률 방식은 여러 개의 채널로 데이터를 전송한다는 점이다. 수신단 기반 제어 방식과의 차이점은 여러 채널을 통해 데이터를 전송할 때 각 채널의 전송률이 일정한 수신단 기반 방식과 달리 복합 전송률 제어 방식은 송신단이 네트워크의 체증상황에 따라 각 채널의 전송률을 조정할 수 있다는 점이다.

2.2.2 적응적 비디오 부호화

적응적 비디오 부호화란 전송률 제어에 의해 결정된 전송률에 맞게 부호화하기 위한 방식이다. 적응적부호화는 Q_p (quantization parameter)를 변화시키거나 프레임율을 변화시키거나 둘을 동시에 변화시킴으로써 가능하다. MPEG-4나 MPEG-4에서의 적응적 부호화 방식은 MB(Macro Block) 단위로 Q_p 를 변화시켜 부호화율을 맞추고 부호화된 데이터의 발생량이 많아서 Q_p 를 변화시키는 것만으로 부호화율을 만족시키지 못할 경우 프레임 스킵을 통하여 프레임율을 변경하는 방법을 사용한다. 이 방식의 기본적인 문제는 Q_p 를 어떻게 결정하는가 하는 것이다. 이 문제를 해결하기 위한 방법이 rate-distortion(R-D) 모델이며, 이 모델을 사용하여 주어진 데이터 양으로 왜곡(distortion)을 최소화 할 수 있는 Q_p (quantization parameter)를 결정할 수 있다.

2.3 TCP 친화적 체증 제어(TCP-Friendly congestion control)

2.3.1 TCP 친화적 체증 제어의 개요

인터넷에서 패킷 손실은 전송 오류와 체증에 의해서 발생한다. 패킷 손실이 발생할 경우 TCP는 자체적인 체증 제어 방법에 의해 전송률을 감소시킨다. 따라서 유사한 왕복지연시간(RTT)을 가진 TCP 연결들이 동일한 채널을 공유하고 있다면 그 TCP 연결들은 가용 대역폭을 균일하게 분배하여 갖게 된다. 이제까지는 대부분의 트래픽이 TCP기반 프로토콜을 사용하여 대역폭 분배가 문제되지 않았으나, 화상회의와 같은 실시간 응용 서비스와 음성/ 영상 스트리밍 서비스 같이 Non-TCP 트래픽의 양이 증가함에 따라 가용 대역폭의 분배문제가 중요하게 되었다. 하지만 Non-TCP 트래픽은 TCP와 양립할 수 있는 체증제어 방법이 없기 때문에 체증발생시 TCP는 전송률을 줄이지만 Non-TCP 트래픽은 원래의 전송률로 계속 전송하여 TCP 트래픽에 심각한 영향을 주게 된다. 따라서 Non-TCP 트래픽도 TCP와 양립할 수 있는 전송률 제어 메커니즘이 필요하며 Non-TCP 트래픽을 TCP 친화적으로 만들어주어 가용 대역폭을 공정하게 분배해주는 방법이 TCP 친화적 체증제어 방법이다. 이에 대한 설명을 하기 전에 우선 TCP에서의 체증제어 방법에 대해서 설명한 후, TCP 친화적 체증제어 방법에 대해 살펴본다.

2.3.2 TCP-Reno의 체증 제어 방법

TCP에서 ACK을 받지 않고 보낼 수 있는 패킷의 개수를 TCP 윈도우 크기(window size)라 하며 초기에 체증 윈도우 크기(congestion window size)는 1이다. 그림 2.2는 TCP-Reno의 체증 윈도우(congestion window)의 빈차를 나타낸 것이다.

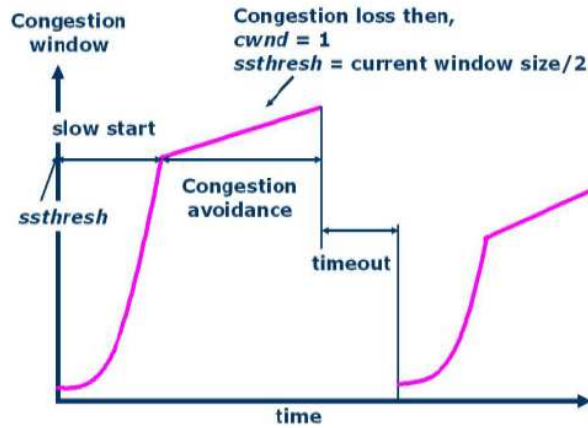


그림 2.2 TCP-RENO의 congestion window 변화 모습

단계 1 (slow start)

송신단에서 윈도우 크기(window size)만큼 패킷을 전송하여 타임아웃(timeout) 이내에 ACK을 전달받으면 윈도우 크기를 전달 받은 ACK의 개수만큼 늘인다. 이 단계는 윈도우 크기가 지수적으로 증가하여 윈도우 크기가 임계값을 넘으면 다음 단계로 진입한다.

단계 2 (congestion avoidance)

송신단에서 윈도우 크기(window size) 만큼 패킷을 전송하여 타임아웃(timeout) 이내에 ACK을 모두 전달 받으면 윈도우 크기를 1씩 증가 시킨다. 이 단계는 윈도우 크기가 선형적으로 증가한다.

단계 3 (fast recovery)

패킷 손실이 발생하여 3개의 중복된(duplicate) ACK을 수신하면 윈도우 크기(window size)를 현재 크기의 반으로 줄인다. 하지만 중복된 ACK 마저 손실이 발생하여 타임아웃(timeout)이 발생할 경우는 윈도우 크기를 1로 한다.

2.3.3 TCP 친화적 전송률 추정방법

TCP-친화적 전송률은 TCP 증가 수율 모델을 근거로 계산되는데 이는 TCP의 정상 상태에서의 동작을 고려하여 시간 평균 수율을 모델링한 것으로 고려된 TCP의 동작에 따라 여러 가지 형태로 나타낼 수 있으나 기본적으로 식 (2.1)과 같은 형태로 나타낼 수 있다.

$$R = f(P, RTT) \quad (2.1)$$

(R은 유효 전송률, P는 패킷 손실률, RTT는 왕복 지연 시간)

TCP Reno 모델에 대해 패킷 손실을 3개의 중복 ACK로만 확인할 경우 식 (2.2)와 같은 유효 전송률을 얻을 수 있다[5].

$$R = \frac{s}{RTT} \sqrt{\frac{3}{2bp}} + \sigma(1/\sqrt{p}) \quad (2.2)$$

여기서 s는 패킷의 크기, RTT(Round Trip Time)는 왕복시간, $\pi s m s$ \square 차fh 확인된 패킷의 개수를 나타낸다.

일반적인 방법으로는 TCP Reno 모델의 동작원리에 근거하여, TCP가 정상상태에서 체증 윈도우를 조절하고 있다고 가정하면

- 패킷이 송신될 때, 즉 윈도우의 크기가 W일 때

$$\text{평균 전송률 } R = \frac{W \times s}{RTT}$$

- 패킷 손실 후 윈도우의 크기가 W/2 이므로

평균 전송률 $R = 0.5 \times \frac{W \times s}{RTT}$ 이다.

따라서 전 구간에서의 평균 전송률은 $R = 0.75 \times \frac{W \times s}{RTT}$ 이 된다. 또한 손실률 p 에 대한 식은 $p = 1/(W/2+(W/2+1)+\dots+W) \approx 1/(3/8 \times W^2)$ 로 구해지므로, $W \approx \sqrt{\frac{8}{3p}}$ 이 되어 최종적으로 시간 t 에서의 유효 전송률 $R(t)$ 는 다음과 같은 근사식으로부터 얻어진다[5]. 구현한 시스템에서는 식 (2.3)을 응용하여 구현되었다.

$$R(t) = \frac{1.22 \times s}{RTT(t) \times \sqrt{p(t)}} \quad (2.3)$$

3. RTP/RTCP

본 장에서는 실시간 수송 프로토콜(RTP)과 실시간 수송제어 프로토콜(RTCP)에 대한 전반적인 내용을 설명한다. 3.1절은 실시간 수송 프로토콜(RTP)에 대해서 설명하고, 3.2절은 실시간 수송제어 프로토콜(RTCP)에 대해서 설명한다.

3.1 실시간 수송 프로토콜(RTP)

실시간 수송 프로토콜(RTP: real-time transport protocol)은 여러 명이 참여하는 영상회의의 필요성에 의해 고안된 프로토콜로서, 중단 간에 음성이나 영상 또는 모의실험 데이터 등 실시간 특성을 가지는 데이터의 전달이 필요한 응용에서 사용되는 프로토콜이다[6].

3.1.1 실시간 수송 프로토콜(RTP)의 특징 및 기능

멀티미디어 데이터의 전송 시 사용되는 UDP 프로토콜은 데이터를 빠르게 전송할 수 있지만 다음과 같은 단점이 있다.

- ◎ 신뢰성 보장 못함
- ◎ 네트워크의 체증제어를 수행하지 못함

UDP의 이러한 단점을 보완하기 위하여 등장한 것이 실시간 수송 프로토콜(RTP)로서, 이 프로토콜은 다음과 같은 특징 및 기능을 가진다.

- ◎ IETF RFC 1889로 다자간 영상회의를 위해 제안되었음,
- ◎ H.323등, 음성/ 영상 데이터를 송수신하기 위하여 사용
- ◎ 음성/ 영상 사이에 동기를 맞추고, 실시간성을 유지하는데 사용

- ◎ 패킷화하는 포맷이지 자체로서 상태를 갖는 프로토콜이 아님.
- ◎ 그 자신이 실시간 전송을 보장하지는 않음
- ◎ 패킷 손실에 대한 복원 메커니즘은 없음.
- ◎ 음성과 영상은 각각 다른 RTP 세션을 이용
- ◎ 하위 프로토콜(UDP 또는 TCP)에 독립적으로 동작.

그림 3.1은 실시간 응용이 RTP를 통해서 동작할 때 프로토콜 스택(Protocol stack) 구조를 나타낸다.

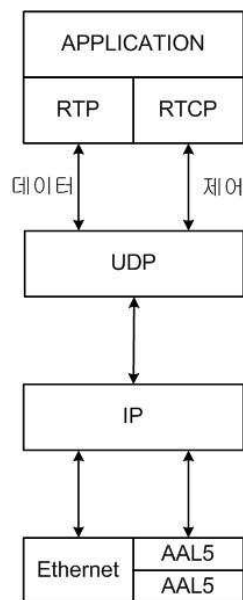


그림 3.1 RTP 프로토콜 스택 구조

3.1.2 실시간 전송 프로토콜(RTP) 패킷의 구조

그림 3.2는 RTP 패킷의 구조를 나타낸다.

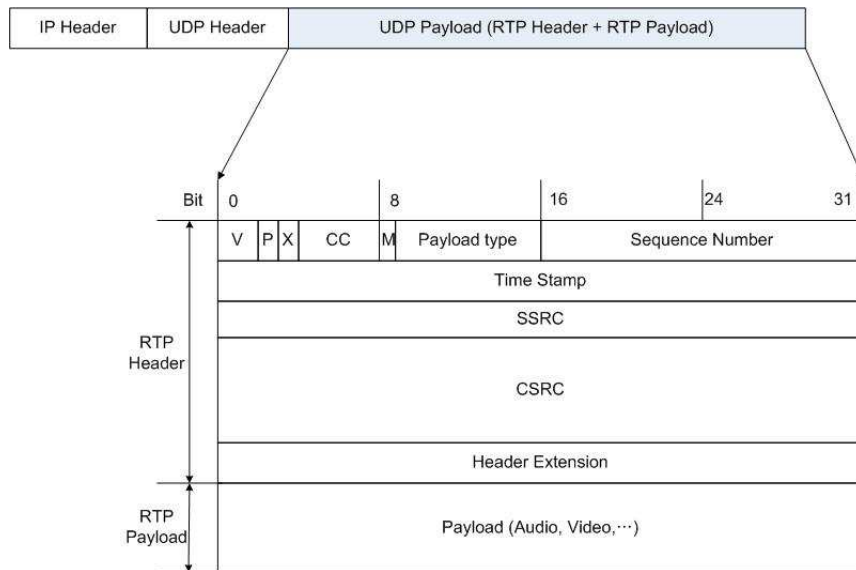


그림 3.2 RTP 패킷의 구조

◎ Version(V) : 2비트 필드로 현재는 항상 2 값을 가진다.

◎ Padding(P) : 1비트 필드로 1 값을 가지면 패킷에 하나 이상의 채워 넣기 바이트가 포함되어 있음을 나타낸다. 마지막 채워 넣기 바이트는 현 패킷에서 무시되어야 하는 채워 넣기 바이트의 수를 나타낸다.

◎ Extension(X) : 1비트 필드로 1 값을 가지면 고정 헤더 이후에 정확히 하나의 확장 헤더가 등장함을 의미한다.

◎ CSRC Count(CC) : 4비트 필드로 고정 헤더 이후에 나열되는 CSRC 식별자의 수를 나타낸다.

◎ Marker(M) : 1비트 필드로 이 필드의 해석은 프로파일에 의해서 결정된다. 이 필드는 패킷 스트림 내에서 프레임 경계와 같은 중요한 이벤트들을 표시하는데 이용된다. 프로파일은 추가 표시 비트들을 정의하거나 PT 필드를 확장하여 표시 비트를 없앨 수도 있다.

◎ Payload Type(PT) : 7비트 필드로 RTP 페이로드의 타입을 나타낸다. 프로파일에서 페이로드 타입의 값과 실제 페이로드 형식을 연결한다.

◎ Sequence Number(SN) : 16비트 필드로 송신되는 각 RTP 패킷에 대해 1씩 증가하는 값을 가진다. 수신 측에서는 패킷 분실을 검출하거나 패킷의 순서를 맞추는데 이용된다. 초기값은 보안을 위해서 무작위로 설정된다.

◎ Timestamp : 32비트 필드로 RTP 데이터 패킷의 첫번째 바이트의 샘플링 순간을 나타낸다. 시계의 주파수는 페이로드의 데이터 형식에 종속되고 형식을 정의하는 프로파일이나 페이로드 형식 문서에 정적으로 명시된다. 초기값은 순번과 마찬가지로 무작위 수로 설정된다.

◎ Synchronization Source(SSRC) Identifier : 32비트 필드로 동기화 소스를 나타낸다. 이 값은 같은 RTP 세션 내에서 같은 SSRC를 가진 동기화 소스가 두 개 이상 나타나지 않도록 무작위로 선택된다.

◎ Contributing Source(CSRC) Identifiers : 이 필드에는 0에서 15목록까지 포함될 수 있으며 각 목록은 32비트를 차지한다. CSRC는 현 패킷에 포함된 페이로드에 기여한 제공 소스들을 나타낸다. 제공 소스가 15개 이상일 경우에도 15개의 제공 소스만 기록된다. 이 필드는 혼합기에 의해 삽입되고 각 목록은 혼합되는 모든 소스들의 SSRC 식별자이다.

3.2 실시간 수송제어 프로토콜 (RTCP)

실시간 수송제어 프로토콜(RTCP: real-time transport control protocol)은 발신지 측의 응용과 목적지 측의 응용사이에서 RTP 세션을 제어하거나 관리하는 프로토콜이다[6].

3.2.1 실시간 수송제어 프로토콜(RTCP)의 특징 및 기능

RTCP는 RTP와 쌍(pair)으로 존재하며, 이 프로토콜의 특징 및 기능은 다음과 같다.

- ◎ RTP를 이용한 영상회의에서 QoS를 지원하기 위한 피드백 메커니즘 (채증 제어)
- ◎ 회의 참가자 전원에게 주기적으로 전송.
- ◎ 순서번호 (sequence number)를 통한 패킷 손실률 조사
- ◎ 타임스탬프(time stamp)를 이용한 지터(jitter) 검사
- ◎ 송신자 보고(SR), 수신자 보고(RR)를 이용하여 전송지연 검사
- ◎ 패킷은 고정길이 헤더 형식으로 시작하고, 언제나 32비트 경계로 끝남
- ◎ 패킷 유형에 따라 가변길이를 가질 수 있음
- ◎ 피드백 정보에 따라 직접 RTP의 유료부하의 데이터 발생량에 영향을 줄 수 있음.
- ◎ RTP 세션에 참가하고 있는 참가자를 파악하여 세션의 크기 추정 가능.

RTCP의 제약 사항은 다음과 같다.

- ◎ RTCP의 대역폭은 RTP 세션 대역폭의 5% 이내 여야 한다.
- ◎ compound RTCP 패킷에서 처음 패킷은 항상 SR 혹은 RR 이어야한다.

◎ SDES CNAME을 포함해야 한다.

3.2.2 실시간 수송제어 프로토콜(RTCP) 패킷의 구조

실시간 수송제어 프로토콜(RTCP) 패킷은 송신자 보고(SR), 수신자 보고(RR), 송신자 서술(SDES), 탈퇴(BYE), 응용(APP)의 5가지 패킷 유형을 가진다. 실제로 이 논문의 시스템에서 중요한 정보는 SR와 RR에 있기에 나머지 유형의 패킷들은 간단한 설명으로 맺는다.

가. 송신자 보고(SR) 패킷과 수신자 보고(RR) 패킷

송신자보고(SR) 패킷은 송수신을 하는 세션 참가자들이 수신한 통계 정보를 보내는 것이며, 수신자 보고(Ra) 패킷은 RTP 패킷을 송신하지 않고 수신만 하는 참가자들이 수신한 통계 정보를 보내는 것이다. 송신자보고(SR) 패킷은 그림3.3과 같이 크게 헤더, 송신자 정보, 수신보고블록의 세 부분으로 나뉘어 진다.

-헤더

- ◎ 버전 (v) 3 RTP 버전을 나타낸다. 현재는 버전 2.
- ◎ 패딩 (P) : 부가적인 패딩 옥텟. 주로 암호화를 위해 사용.
- ◎ 수신보고 카운트(HC) : 이 패킷에 포함된 수신보고블록의 개수.
- ◎ 패킷 유형 (PT) : SR='200', Rf='201', SDES='202', BYE='203',APP='204'

가 사용.

◎ 길이 (length) : 32비트 워드(word) 단위, RTCP 패킷의 길이에 '1'을 뺀 값이 기록

- ◎ 발신지 SSRC : SR패킷 보낸 발신지의 동기화 발신지식별자(SSRC)를 기록

- 송신자 정보

◎ RTP(network time protocol) 타임스탬프 : 패킷을 보내는 절대 시간(wallclock time)을 기록. 수신자들까지의 왕복지연들 계산하는데 사용하며 단위는

초 단위 값 32비트와 [us] 단위 값 32비트가 쓰인다. 여기서 [us] 단위 값은 초단위로 환산한 후 2^{-32} 으로 나누어준 값을 입력한다.

◎ RTP 타임스탬프 : NTP 타임스탬프와 동일한 시간에 대응한 RTP 타임스탬프이며, 매체들간, 또는 한 매체내의 동기화에 사용

◎ 송신 패킷 수 (sender's packet count) : 전송을 시작한 이래, 현재의 송신자보고 패킷을 보낼 때까지 전송한 총 RTP패킷 수

◎ 송신 옥텟수(sender's octet count) : 전송을 시작한 이래 현재의 송신자보고(SR) 패킷을 보낼 때까지 전송한 총 유료부하의 옥텟 수

- 수신 보고 블록

◎ SSRC_n(source identifier) · 수신 보고 블록에 들어 있는 정보와 관련된 발신지의 동기화 발신지 식별자 (SSRC)가 기록

◎ 손실비(fraction lost) : 이전 송신자보고(SR) 또는 수신자보고(RR) 패킷을 보낸 이래, 최근 SR 또는 RR 까지 발신지 SSRC_n이 보낸 RTP패킷이 손실된 비율

◎ 누적 손실 패킷수(cumulative packets lost) : 수신을 시작한 이래, 최근 SR 또는 RR까지 발신지 SSRC_n이 보낸 RTP 패킷 중 손실된 총 패킷 수

◎ 최대 수신 순서 번호(extended highest sequence number received) : 발신지 SSRC_n로부터 최근 SR또는 RR 까지 수신한 RTP패킷 중에서 가장 큰 순서 번호

◎ 도착지연변이 (inter-arrival jitter)는 RTP패킷들의 도착 시간간격과 송신시간 간격과의 차이를 통계적으로 추정 한 값으로서 RTP 타임스탬프 단위로 측정

◎ 마지막 SR (last SR) - 발신지SSRC_n으로부터 수신된 최근의 RTCP 송신자보고(SR) 패킷의 NTP 타임스탬프 64비트 중에서 가운데 32비트

◎ 마지막 SR이후 지연(delay since last SR) : 발신지 SSRC_n의 마지막 SR 패킷을 수신한 후, 수신 보고 블록을 보낼 때까지의 지연은 1/65,536 ch 단위로 나타낸 것이다.

| V = '2' | P | RC | PT = '200' | RTCP 패킷의 길이 (length) | 헤더 |
|---|---|----|------------|----------------------|---------------|
| 발신지 SSRC (SSRC of sender) | | | | | 송신자 정보 |
| NTP 타임스탬프(상위 32비트) (NTP time stamp) | | | | | |
| NTP 타임스탬프(하위 32비트) (NTP time stamp) | | | | | |
| RTP 타임스탬프 (RTP time stamp) | | | | | |
| 송신 패킷 수 (sender's packet count) | | | | | 수신 보고 블록 |
| 송신 옥텟 수 (sender's octet count) | | | | | |
| SSRC_1 (First source of SSRC) | | | | | |
| 손실비 (fraction lost) | 누적 손실 패킷 수 (cululative number of packets lost) | | | | |
| 최대 수신 번호 (extended highest sequence number received) | | | | | 수신 보고 블록 2 |
| 도착 지연 변이 (inter arrival jitter) | | | | | |
| 마지막 SR (last SR : LSR) | | | | | |
| 마지막 SR 이후의 지연 (DLSR : Delay since last SR) | | | | | |
| SSRC_2 (Second source of SSRC) | | | | | 수신 보고 블록 2 |
| | | | | | |
| 프로파일과 관련된 확장 정보 (profile-specific extensions) | | | | | |

그림 3.3 송신자 보고(SR:Sender Report) 용 RTCP 패킷의 구조

수신자 보고(RR) 패킷의 구조는 송신자 보고(SR) 패킷과 같은 구조를 보이나, 패킷 유형 구간에 수신자 보고 패킷임을 나타내는 값 '201'이 오고, 송신자 정보 부분이 생략된다. 나머지 구간은 송신자 보고 패킷과 동일한 의미로 사용된다.

나. 송신자 서술 (SDES : source description) 패킷

송신자를 식별하는 항목들이 포함되며, 패킷의 구조는 그림 3.4와 같다. Version(V), Payload(P), RTCP 패킷길이(Packet Length) 구간은 송신자보고 패킷과 동일하다.

| | | | | |
|------------------|---|----|------------|------------|
| V = '2' | P | SC | PT = '202' | RTCP 패킷 길이 |
| SSRC/CSRC_1 | | | | |
| SDES 항목 ***** | | | | |
| SSRC / CSRC_2 | | | | |
| SDES 항목 ***** | | | | |

그림 3.4 SDES : 소스 설명 RTCP 패킷

SDES는 헤더와 0개 이상의 Chunk로 구성된 3-계층 구조로 각 Chunk는 그 Chunk에 식별된 소스를 설명하는 Item들로 구성된다.

◎ Packet Type(PT) : 8비트 필드로, RTCP SDES 패킷의 경우에는 202 값을 갖는다.

◎ Source Count(SC) : 5비트 필드로, SDES 패킷에 포함된 SSRC/CSRC Chunk의 수를 나타낸다. 0 값도 유효하지만 아무런 의미도 없다.

각 Chunk는 SSRC/CSRC 식별자와 SSRC/CSRC에 관계되는 정보를 수송하는 0개 이상의 Item들의 리스트로 구성된다. 각 Chunk는 32비트 경계에서 시작한다. 각 Item은 8비트의 종류 필드와 텍스트의 길이를 나타내는 8비트의 바이트 계수, 그리고 텍스트 자체로 구성된다. 텍스트는 255 바이트까지 이용할 수 있지만 RTCP 대역폭 소비에 관계된다는 것을 인지해야 한다.

다. BYE : Goodbye RTCP 패킷

| | | | | |
|------------------|-------|----|------------|------------|
| V = '2' | P | SC | PT = '204' | RTCP 패킷 길이 |
| SSRC/CSRC | | | | |
| SDES 항목 | | | | |
| 길이 | 탈퇴 이유 | | | |
| 탈퇴 이유(계속) | | | | |

그림 3.5 탈퇴(Bye) 패킷의 구조

BYE 패킷은 소스가 더 이상 활성화 상태가 아님을 알린다.

◎ Packet Type(PT) : 8비트 필드로, RTCP BYE 패킷은 203 값을 갖는다.

◎ Source Count(SC) : 5비트 필드로 BYE 패킷에 포함된 SSRC/CSRC 식별자의 수를 나타낸다. 0 값도 유효하지만 의미는 없다.

BYE 패킷이 혼합기에 도착하면 혼합기는 SSRC/CSRC 식별자 변경 없이 그대로 전달한다. 혼합기가 수행이 중지될 때 자신의 SSRC 식별자와 자신이 처리하는 모든 제공 소스들을 나열한 BYE 패킷을 보내야 한다. 부가적으로 BYE 패킷은 8비트 바이트 계수와 계수 바이트 크기의 탈퇴 이유(예, “camera malfunction”, “RTP loop detected”)를 나타내는 텍스트를 포함할 수 있다.

라. APP : Application-defined RTCP 패킷

새로운 응용이나 새로운 기능이 개발되었을 때 패킷 종류 값을 등록할 필요 없이 실험을 목적으로 사용되기 위한 패킷이다. 인식할 수 없는 이름을 가진 APP 패킷은 무시되어야 한다. 실험 후에 광범위한 이용성이 판명되면 각 APP 패킷은 부 타입과 이름 필드 없이 재정의되어 RTCP 패킷 타입을 이용해서

IANA(Internet Assigned Number Authority)에 등록하도록 권고된다.

| | | | | |
|--|----------|-----------|-------------------|-------------------|
| V = '2' | P | SC | PT = '204' | RTCP 패킷 길이 |
| SSRC/CSRC | | | | |
| 이름 (ASCII) | | | | |
| 응용에 관련된 데이터 (application-specific data) | | | | |

그림 3.6 응용(APP) 패킷의 구조

◎ Subtype : 5비트 필드로 일련의 APP 패킷들이 하나의 유일한 이름으로 정의될 수 있도록 하는 부 타입으로 사용되거나 응용에 종속되는 데이터를 위해 사용될 수 있다.

◎ Packet Type(PT) : 8비트 필드로 RTCP APP 패킷은 204 값을 가진다.

◎ Name : 4바이트 필드로 이 응용이 수신할 다른 APP 패킷들에 대해 유일하도록 APP 패킷을 정의하는 이름으로 사용자에 의해 선택된다.

◎ Application-specific Data : 가변 길이의 필드로 APP 패킷에 나타날 수도, 나타나지 않을 수도 있다. 이 필드는 RTP에 의해 해석되지 않고 응용에 의해 해석된다. 이 필드의 길이는 32비트의 배수여야 한다.

4. CDMA 1X-EVDO 환경 하에서의 가변율 비디오 전송 원격의료 시스템 구현

이 장에서는 본 논문을 위해 구현한 가변율 비디오 전송 시스템에 대한 내용을 설명한다. 4.1절은 CDMA 1X-EVDO의 기본개념에 대해서 설명하고, 4.2절은 시스템의 기본 개념과 실시간 수송 프로토콜(RTP)과 실시간 수송제어 프로토콜(RTCP)의 적용 메커니즘에 대해서 설명한다. 이어서 4.3절은 서버-클라이언트 메커니즘에 대해서 설명하고, 마지막으로 4.4절은 시스템의 구현내용에 대해서 기술한다.

4.1 CDMA 1X-EVDO

CDMA 1X-EVDO(Evolution-Data Only)는 기존의 CDMA 1X 무선 프로토콜과는 완전히 다른 패킷 데이터 전송을 위한 전용 프로토콜로 최대 전송속도가 전방향의 경우 2.4576Mbps까지 가능하고 역방향에서 153.6kbps까지 가능한 비대칭 전송 속도 구조를 가진다. 기존의 CDMA 1X 역시 무선 구간 전송 속도를 153.6kbps까지 지원하지만 사업자 측면에서 무선 구간의 용량 문제로 VOD(Video On Demand)의 경우 64kbps까지 할당하고 무선 인터넷은 평균적으로 32kbps를 나타낸다. 또한 기본적으로 음성 활성화율(Voice activity rate)에 중점을 두어 지연 시간에 민감하게 반응한다. 하지만 1X-EVDO는 무선구간에서 가입자의 전송속도를 동적으로 할당하고, 패킷 데이터의 고유한 특성인 잠재 모드(dormant mode)를 최대한 활용하여 무선구간 및 시스템 사용 효율을 극대화하고 있다[20][21].

1X-EVDO 망에서 역시 기지국과 단말기 사이에 RLP를 사용하여 전송을 제어한다[22]. 1X-EVDO에서는 RLP 버전 3을 사용하는데 이것은 순열 번호에 해당하는 비트 수를 증가하여 많은 데이터 처리에 유리하도록 하고 가변 프레임 크기를 도입하여 전송 속도의 변화에 민감하게 작용하도록 한다. RLP는 단순히 best

effort delivery 역할로서 3번까지 오류를 처리하고 상위 계층으로 넘긴다. 그림 4.4는 CDMA 1X-EVDO 환경에서의 원격진료 시스템과 통신망 구조를 보인다. 이 당시 애플런스와 같은 곳에서 역방향 링크를 통해 최대 153.6kbps로 전송된 데이터는 기지국(BS)에서 사설 데이터망(PDSN)을 통해 IP 네트워크로 접속하여 3차 진료기관으로 전송된다.

CDMA 1X-EVDO의 역방향 채널에서 153kbps의 데이터 전송은 그림 4.4와 같이 EVDO 폰을 노트북에 연결하고 AT Command에서 HDR(High Data Rate) 서비스를 요청함으로써 이루어진다. 이것은 기지국으로부터 데이터 전송이 가능한 SCH(Supplemental Channel)를 할당받는 것으로 여러 개의 채널을 동시에 사용하여 넓은 대역폭을 확보하지만 일반적인 음성 통화 채널과 동시에 사용할 수 없고 높은 서비스 이용 요금이 요구된다. 이러한 EVDO에 대한 153.6kbps까지의 역방향 전송 속도는 CDMA 1X 보다 10배 이상의 대역폭을 보장하기 때문에 원격 진료를 위한 추가적인 데이터 전송이 가능하다. 본 논문에서는 이미지와 생체신호 이외에 화상 회의 데이터 중 비디오 신호의 경우 MPEG-4 코덱을 사용하고 음성 신호는 ACELP(Algebraic Code Excited Linear Prediction)을 이용하여 압축한 후 전송한다. 획득된 데이터는 UDP(User Datagram Protocol)/IP 프로토콜을 사용하여 PSDN(Packet Switch Data Network)을 통해 응급실의 유선 네트워크 망으로 연결된다. TCP (Transmission Control Protocol)가 연결 지향적인데 비해 UDP는 비연결지향 방식이므로 실시간 스트리밍 서비스에 유리하고 에러가 많은 환경에서 TCP의 재전송은 큰 전송 지연을 발생하기 때문에 무선환경에서는 UDP를 사용해야 한다[23][24][25].

4.2 시스템의 기본 개념

인터넷(Internet)을 통한 가변율(VBR : Variable Bit Rate) 비디오 전송 시스템은 서버/클라이언트 모델을 바탕으로 한 단대단 시스템이다(그림 4.1). 서버에서 보내는 영상데이터는 실시간 수송 프로토콜(RTP)을 이용하여 단 방향으로 클라이언트로 전송되며, 네트워크 부하상황에 따라 송신단에서 부호화되는 데이터의 양과 전송하려는 데이터의 양을 제어하기 위한 실시간 수송제어 프로토콜(RTCP) 패킷이 서버와 클라이언트 사이에 양방향 전송을 한다. 실시간 수송 프로토콜(RTP)의 하위계층에는 영상데이터의 실시간성을 보장하기 위해 UDP를 사용하지만 실시간 수송제어 프로토콜(RTCP)은 UDP를 사용할 수도 있고, 신뢰성 보장을 위해 TCP를 사용할 수도 있다.

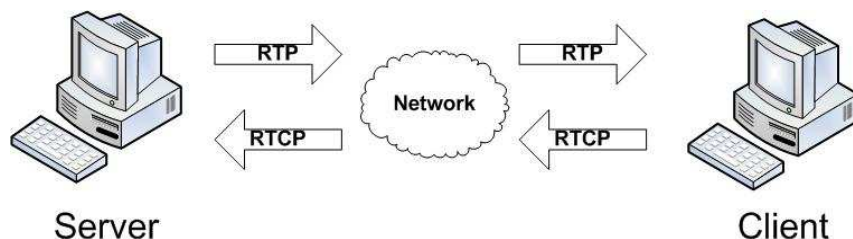


그림 4.1 서버/ 클라이언트 모델

그림 4.2는 인터넷(Internet) 실시간 비디오 전송 시스템의 전체적인 구조를 자세히 나타내고 있다. 시스템은 크게 MPEG-4 비디오 부호화부(MPEG-4 encoder)와 실시간 비디오 전송 서버(real-time video transmission server), 실시간 비디오 수신기(real-time video receiver) 의 3가지로 구분된다.

MPEG-4 비디오 부호화(encoder)부는 실시간으로 들어오는 영상데이터를 부호화하여 비트스트림으로 만든다. 부호화율은 사용자가 입력한 최소 전송률과 최대 전송률 중에서 최소 전송률로 초기화 하며, 수송 중에 RTCP 패킷을 수신하면 채

널 상태 정보를 통해서 부호화율을 다시 설정한다. 실시간 비디오 전송 서버(real-time video transmission server)는 부호화된 데이터를 네트워크를 통해 전송하며, 수신기로부터 제어정보를 수신한 후 전송률을 조정하는 역할을 수행한다. 이 때, 데이터 전송은 RTP/UDP를 사용하여 전송하고, 제어정보는 RTCP에 UDP 또는 신뢰성 있는 전송을 위해 TCP를 사용할 수 있다. 이 과정에서 고려해야 하는 두 가지 사항이 있다. 첫 번째가 서버나 클라이언트 측에 있는 버퍼에 고갈(under flow)이 발생하지 않게 해야 한다. 이를 위해 세션 초반에 일정량의 데이터를 수신한 후에 비디오 복호화부 프로세스를 동작시킨다. 두 번째가 네트워크에 과부하를 주어서는 안 된다는 것이고 이를 위하여 TCP 친화적 체증제어(TCP-Friendly congestion control)를 수행한다(2.3절 참조). 실시간 비디오 수신기는 서버로부터 받은 데이터를 또.263 비디오 복호기에 넘겨주고, 수신한 데이터의 전송 지연, 지터(jitter) 등의 정보를 분석하여 서버 측에 피드백을 해준다. 데이터 수신 시에는 일정 크기의 버퍼를 두어 비디오 보호기에서 고갈(underflow)이 발생하지 않도록 버퍼 제어를 수행한다.

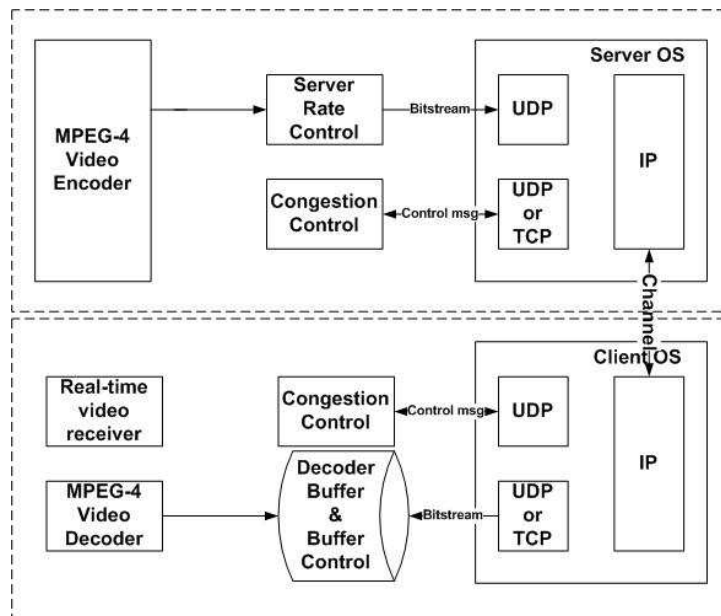


그림 4.2 가변율 비디오 전송 (VBR video transmission)시스템

4.3 서버/클라이언트 메커니즘

서버 와 클라이언트 메커니즘은 스트림의 전송과 관련되는 부분이다. 전송 프로토콜의 구조는 그림 4.3과 같다. 전송 계층에서 UDP를 사용하면 TCP보다 실시간 응용에 더욱 좋지만 전송의 신뢰성을 보장해 줄 수 없다. 따라서 RTCP와 같은 제어정보는 신뢰성을 보장해 주기 위해 TCP를 사용할 수 있다. 그림 4.3에서 볼 수 있듯이 실시간 비디오 전송 시스템 자체는 어플리케이션 계층의 응용 프로그램이다. 논리적으로는 서버와 클라이언트가 어플리케이션을 통해 바로 연결되지만, 실제 데이터의 송수신은 어플리케이션 계층의 하위계층을 거쳐서 이루어진다. 즉 서버측에서 보내는 데이터는 전송계층에서 일정한 크기마다 UDP 혹은 TCP 헤더를 추가하여 UDP 혹은 TCP 패킷을 만든다. 이 패킷은 인터넷 계층에서 IP 헤더를 삽입하여 IP 패킷이 되고, 다시 물리 계층으로 내려가서 실제 물리 계층인 LAN선을 타고 클라이언트로 전달된다. 클라이언트는 물리계층을 거쳐 데이터를 받고, IP 헤더를 통해 자신에게 오는 데이터가 맞는지 여부를 판단하게 된다. 자신에게 오는 데이터가 맞으면 IP 헤더를 제거한 데이터를 전송계층에 보낸다. 전송계층에서는 에러 여부를 판단하여 에러가 없으면 상위 어플리케이션 계층으로 패킷을 보내고, 에러가 있으면 그 패킷을 버리게 된다. 에러발생시 TCP의 경우는 재전송을 요구하지만, UDP의 경우는 재전송을 요구하지 않게 된다.

마이크로소프트사의 윈도우 OS에서 네트워크 프로그래밍은 윈속(winsock)을 통하여 이루어진다. 소켓이란 네트워크 프로그램을 개발하기 위한 인터페이스이며, 윈속은 윈도우가 제공하는 소켓이다. 윈속에서 UDP를 이용한 데이터의 송수신 과정은 다음 그림 4.4와 같으며, TCP를 이용한 데이터의 송수신 과정은 다음 (그림 4.5)과 같다[15].

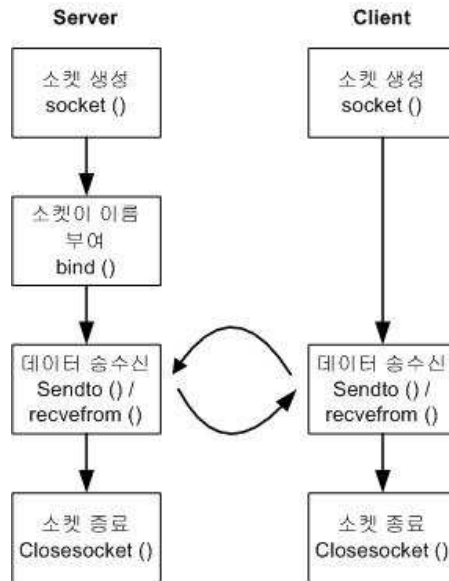


그림 4.3 UDP를 사용하는 클라이언트 서버 Winsock 함수 흐름도

UDP를 사용할 경우 2개의 소켓이 필요하다. 먼저 서버와 클라이언트가 socket() 함수를 호출하여 서버소켓과 클라이언트 소켓을 생성하고, 서버는 bind() 함수를 호출하여 소켓에 이름을 부여한다. bind() 함수를 통해 소켓에 IP 주소와 Port 번호가 입력된다. 소켓에 IP 주소와 Port번호를 입력하는 이유는 네트워크를 통해 들어오는 데이터는 결국 소켓 인터페이스를 통해 수신을 하게 되는데, 들어오는 데이터가 자신에게 오는 것이 맞는지의 여부를 소켓에 입력된 IP 주소와 Port번호를 통해 확인하기 때문에 소켓에 IP 주소와 Port 번호를 입력하는 것이다. 데이터의 송신은 sendto() 함수, 데이터의 수신은 recvfrom() 함수를 통해서 이루어진다. 소켓의 종료 시에는 closesocket() 함수를 호출하여 생성된 소켓을 소멸시킨다.

TCP를 사용할 경우 모두 3개의 소켓이 필요하다. 서버와 클라이언트가 각각 하나씩의 소켓을 가지며 서버는 클라이언트의 접속 요청을 받아들이기 위해 별도의 소켓을 가지고 있어야 한다.

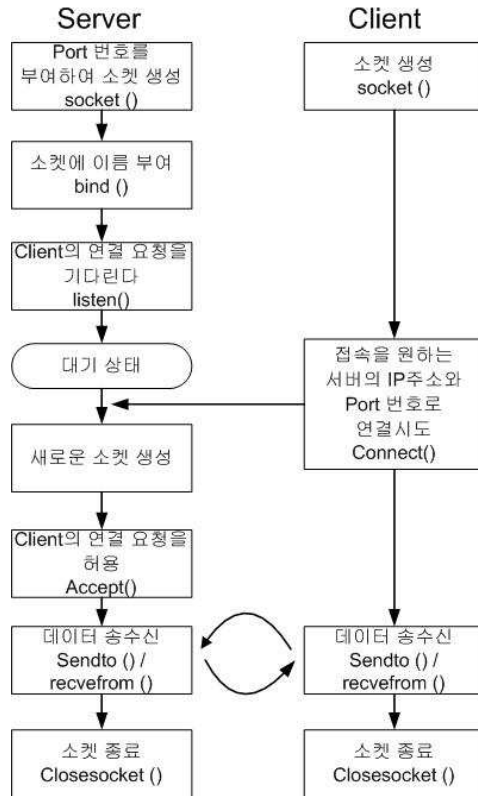


그림 4.4 TCP를 사용하는 클라이언트 서버 Winsock 함수 흐름도

서버는 socket() 함수를 호출하여 소켓을 생성하고, listen() 함수를 호출하고 클라이언트의 접속 요청을 감시하기 시작한다. 서버는 listen() 함수를 호출한 후부터 클라이언트와 접속 요청을 받아들일 수 있는 상태가 된다. 클라이언트가 서버에 접속하기 위해서는 역시 socket() 함수를 호출하여 소켓을 생성하고 서버의 IP 주소와 포트 번호를 지정하고 connect() 함수를 호출하여 서버에 접속을 요청한다. 클라이언트로부터 접속 요청을 받은 서버 소켓은 별도의 소켓을 생성하고, accept() 함수를 호출하여 새로 생성된 소켓과 클라이언트를 연결 시켜 준다. 데이터를 보내고자 할 때는 send() 함수, 받을 때는 recv() 함수를 호출한다. 따라서

RTP 패킷을 전송할 때는 RTP 헤더와 유로부하를 구성한 후, 그림 4.4의 절차에 의해서 UDP 소켓을 생성하고, UDP 소켓을 통해 RTP 패킷을 전송한다. RTCP 패킷을 전송할 때는 RTCP 패킷의 내용을 구성한 후, UDP 소켓을 통해 전송하려면 그림 4.4의 절차에 의해 전송하게 되고, TCP 소켓을 통해 전송하려면 그림 4.5의 절차에 의해서 UDP 또는 TCP 소켓을 생성하고, 생성된 소켓을 통해 RTCP 패킷을 전송한다.

위에 설명한 내용은 기본적인 송수신에 관련된 내용이고, 일반적으로 실제 시스템을 구현할 때 RTP와 RTCP는 별개의 소켓을 구성하여 사용하며, 또한 양방향성(full duplex)을 지원하기 위해서 송신과 수신용으로 RTP와 RTCP 소켓을 각각 2개씩을 두어 결국 RTP 송수신용 2개, RTCP 송수신용 2개로 총 4개의 소켓을 구성하여 구현한다. 따라서 서버와 클라이언트는 각각 4개씩의 소켓을 사용하지만, 현재의 시스템은 서버에서 클라이언트로 단 방향 전송만을 하므로 서버 측에서 RTP 패킷에는 송신용 소켓만으로 충분하며, 클라이언트 측에서 RTP 패킷 수신용 소켓만으로 충분하므로, 서버와 클라이언트 각각 3개씩의 소켓으로 시스템을 구성하였다.

서버는 채널 가용 대역폭의 변화에 따라 데이터 발생율을 바꿀 수 있어야 하며, 서버측의 패킷 전송속도 또한 조절할 수 있어야 한다. 이를 위한 실시간 비디오 전송 시스템의 송수신 시나리오는 다음과 같다. (그림 4.6)의 시나리오에 따라 우선 서버와 클라이언트는 초기화 정보인 이미지의 형태(QCIF 또는 CIF), 실시간 수송 제어 프로토콜(UDP 또는 TCP), 전송할 스트림의 종류 등을 설정한다. 그림 4.6은 UDP인 경우를 도시하였다. 서버는 일정크기로 패킷을 구성하여 클라이언트에 보내게 된다. 클라이언트는 서버로부터 받은 데이터를 파일에 저장하고, 저장된 데이터양이 임계 값을 넘으면 복호화와 디스플레이를 하게 된다. 이와 같이 하는 이유는 일종의 버퍼를 두어 고갈(underflow)을 방지하기 위함이다. 특히 복호화와 디스플레이가 독립적으로 동작하도록 멀티스레드로 구현함으로써 시스템 자원의 효율이 최대가 되도록 하였다. 데이터 전송 중에 서버는 실시간 수송 제어 프로토콜(RTCP)의 송신자 보고(SR) 패킷을 클라이언트에 수송하면, 클라이언트는 수신자 보고(RR) 패킷을 서버 측에 수송한다. 서버는 수신자 보고(RR) 패킷을 수신한

후 채널 가용 전송률을 수정한다. 그 후 수송이 모두 마무리되면 서버는 RTCP 탈퇴(BYE) 패킷을 클라이언트에 수송한 후 전송을 끝내고, 클라이언트는 탈퇴(BYE) 패킷을 수신한 후 수신을 마무리한다.

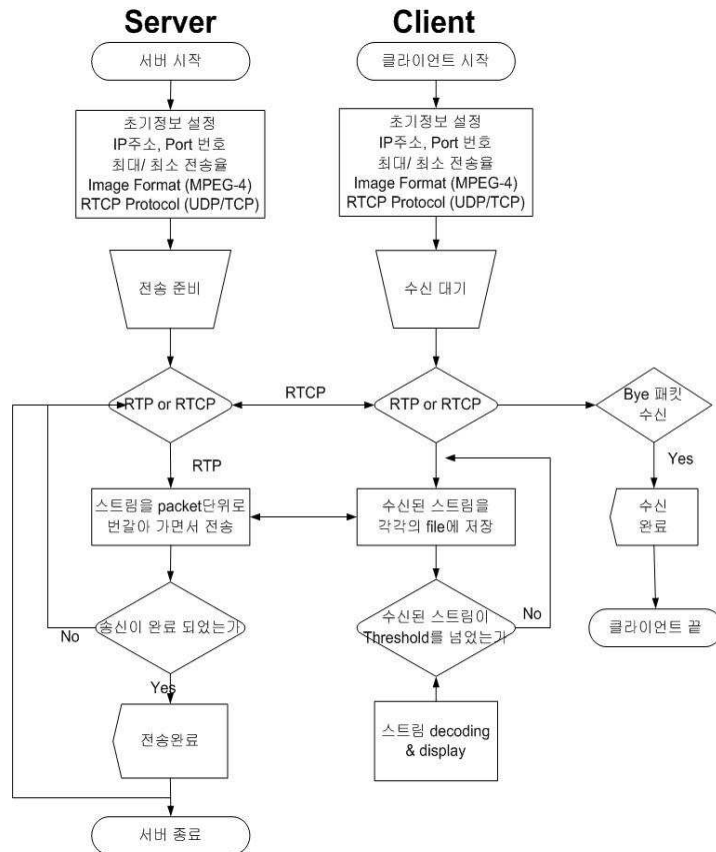


그림 4.5 UDP 수송 방식에 의한 송수신 시나리오

4.4 RTP/RTCP 프로토콜의 적용 메커니즘

그림 4.6은 실시간 전송 프로토콜(RTP)과 실시간 전송제어 프로토콜(RTCP)의 적용 메커니즘을 도시한 것이다. MPEG-4 부호화를 통해 나온 비트스트림은 RTP 모듈을 거치면서 RTP 헤더를 가진다. RTP 패킷은 전송 모듈에 의해 윈속(wsock)으로 전달되고, 소켓을 통해 네트워크로 전달된다. 전송된 RTP 패킷은 수신측에 전달되어 수신측 RTP 모듈에서 필요한 정보를 처리한 후 헤더를 제외한 유효부하 부분만 복호기에 입력되어 복호를 수행한다. 이 때 클라이언트에서 RTP 패킷을 받을 때마다 저장되는 정보는 RTCP 패킷을 구성할 때 사용된다. 즉 클라이언트에서 RTP 패킷에 있는 정보를 조합하여 RTCP 패킷을 서버 측에 전송한다. 서버는 수신한 데이터를 제어 모듈을 통해서 데이터를 분석하여 전송률 제어(rate control)를 하며, 전송부에는 흐름 제어(flow control)를 한다.

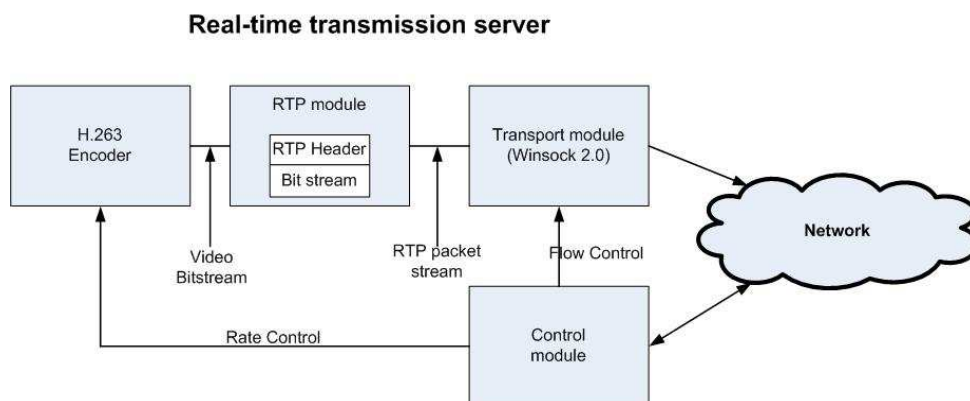


그림 4.6 RTP/RTCP 프로토콜의 적용 메커니즘

4.5 RTP 타임스탬프와 NTP 타임 스탬프와 지터(Jitter) 계산

이 절에서는 RTP에서의 타임스탬프와 RTCP에서 사용되는 NTP 타임스탬프에 대해서 기술한다.

4.5.1 RTP 타임 스탬프

미디어간 동기(synchronization)와 지터(jitter) 계산에 사용되며 RTP 데이터 패킷에서 첫 번째 바이트를 샘플링한 시간을 나타낸다. 타임스탬프의 초기값은 랜덤 넘버를 출력하는 rand(void) 함수에 의해 임의의 값을 사용한다. 타임스탬프의 쓰임은 다음과 같다. 예를 들자 초기에 RTP 타임스탬프 값이 임의의 값 1234000[ms]이 나왔다고 가정하자. 8kHz 샘플링을 하는 PCM coding된 음성의 경우 초당 8000바이트의 데이터가 발생하는데 사용자가 16000바이트마다 데이터를 RTP 패킷을 사용하여 보낸다고 하면 RTP 타임스탬프는 매번 2초씩 증가하게 된다. 따라서 RTP 타임스탬프는 1234000[ms] -> 1236000[ms] ->1238000[ms]로 변화하게 된다. 이와 같이 하지 않고 메모리에 있는 데이터를 일정시간마다 일정 데이터만큼만 보낸다고 가정하자. 초기에 RTP 타임스탬프가 1234000[ms]으로 설정되고, 매 2초마다 메모리에 있는 데이터를 4000바이트씩 보낸다고 가정하면 RTP 타임스탬프는 1234000[ms] -> 1236000[ms] -> 1238000[ms]로 변화하게 된다. 타임스탬프 값을 구하기 위해 사용하는 윈도우의 함수는 아래와 같다.

◎ int rand(void) - required header : <stdlib.h>

0에서 RAND_MAX 사이의 의사 난수(pseudorandom number)를 발생시킨다.

◎ DWORD GetTickCount(VOID)

PC가 켜진 시간부터 (ms) 단위로 증가하는 함수이다. 이 함수는 컴퓨터가 켜진 채 49.7일 지나면 다시 0으로 돌아가서 값을 증가시킨다.

© voidftime(struct_timeb*timeptr) - required header : <sys/types.h> and <sys/timeb.h> 1970년 1월 1일 자정을 기점으로 현재까지의 시간을 초로 표시한 값을 _timeb 구조체에 넣어준다. timeb 구조체에 [ms] 값도 있어서 현재가 몇 초 몇 ms 인지를 알 수 있다.

4.5.2 NTP 타임 스탬프

```
typedef struct (
    u_int32 secs; // 320bits
    u_int32 frac; // 320bits
) ntp64;
```

NTP 타임스탬프에 사용되는 값은 RTP 타임스탬프에서 쓰이는 값과 다른 개념이다. RTP 타임스탬프는 미디어간 동기화 지터를 구하기 위함이고 NTP 타임스탬프는 RTT를 구하기 위함이다. RTP 타임스탬프란 네트워크 타임스탬프의 약자로서 네트워크 내에 서버를 두고 그 서버에 속해있는 모든 컴퓨터들은 같은 NTP 타임스탬프 값을 사용하지만 실제적인 구현은 각 PC에서 _ftime(struct_timeb .timeptr) 값을 사용하여 나온 값을 NTP 타임스탬프 값에 맵핑하여 사용한다. 앞에서 설명했듯이 _ftime() 함수는 1970년 1월 1일 자정을 기점으로 한 시간이며 NTP 타임스탬프는 1900년 1월 1일을 기점으로 한 초 단위의 시간과 1초보다 작은 시간으로 구성된다. 1초보다 작은 시간은 $1 / (2^{-32})$ 초 단위의 시간을 가지고 있다. 따라서 위 구조체의 NTP 타임스탬프에서 초 단위의 값 secs는 _ftime()으로 구한 시간에서 초 단위의 값에 오프셋으로 2208988800U 칸을 더해주고, 1초보다 작은 시간 단위의 값 frac는 _ftime()으로 구한 [us] 단위의 시간에 1000을 곱하여 [us] 단위의 시간으로 바꾸고, 2^{-32} 으로 나누어 주면 된다. 아래 내용은 그 과정을 보여준다.

$$2^{32}/10^6 = 4096 + 256 - 1825/32;$$

```

unsigned_int_32_bits t = (usec * 1825) >> 5;
n,frat = ((usec << 12) + (usec << 8) - t);

```

이와 같이 하면 n_frac 값을 구할 수 있게 된다. NTP 타임스탬프를 구하기 위한 함수는 앞 절에서 설명한 _ftime 함수를 사용한다.

4.5.3 지터 (Jitter) 계산

지터는 수신 측에서 RTP 패킷을 받을 때마다 RTP 패킷의 타임스탬프 값을 이용하여 계산한다. 지터는 네트워크를 통해 패킷이 전달될 때 전달지연이 얼마나 변동하면서 전달되는지 알 수 있다. 계산 방법은 다음과 같다.

$$D(i, j) = (R_1 - S_1) - (R_j - S_i) = (R_i - R_j) - (S_j - S_i) \quad (4.1)$$

$$J(I) = J(i-1) * 15/ 16 + (| D(i-1, j) | - J(j-1)/ 16) \quad (4.2)$$

(R_i : i번째 패킷이 도착한 시간, S_i : i번째 패킷을 보낸 시간)

지터는 수신단에서 계산하여 수신자보고를 보낼 때 포함하는 값으로써 이 때 사용하는 값은 RTP 패킷의 RTP 타임스탬프와 수신단에서의 타임을 이용하여 구할 수 있다. 식 (4.1)에서 보면 현재 D(i, j)값은 결국 송신단에서 두 패킷을 송신한 시간 간격과 수신단에서 두 패킷을 수신한 시간간격과의 차이를 나타내는 값이다. 따라서 송신단에서 온 두 RTP 패킷의 RTP 타임스탬프의 차이를 얻고, 수신단에서는 두 패킷을 받은 시간 차이를 구하여, 두 값의 차이를 구하면 D(i, j)를 구할 수 있다. 그리고 이 값들을 식 (4.2)에서 지터 계산을 위해 사용하게 된다. 이 지터 또한 평균값들을 취해가고 있음을 볼 수 있다.

4.6 왕복지연 시간(RTT)과 패킷손실률 추정 메커니즘

이 절에서는 패킷 손실률과 왕복지연시간을 측정하는 메커니즘에 대해 기술한다.

4.6.1 왕복 지연 시간 측정법

그림 4.7은 왕복지연시간을 구하는 과정을 도시한 것이다. 그림에서 볼 수 있지만 왕복지연 시간은 송신단에서 구할 수 있는 값이다. 우선 송신단에서 송신자보고(SR)를 보내는 것으로 시작된다. 이 때 송신자보고(SR)를 보낼 패의 NTP 타임스탬프 값을 송신자보고(SR)에 넣어서 보낸다. 수신단은 송신자보고(SR)를 받은 시간을 기억하고 있다가 수신자보고(RR)를 보낼 때, 송신자보고(SR)를 받은 후 수신자보고(RR)를 보낼 때까지의 시간간격을 구하여 수신자보고(RR) 패킷의 DLSR (Delay since Last SR) 필드에 넣는다. 그리고 SR에서 받은 NTP 타임스탬프 64비트중에서 가운데 32비트를 떼어내어 수신자보고(RsR)의 LSR 필드에 넣고 송신단에 수신자보고 패킷을 보낸다. 송신단은 수신자보고 패킷을 수신할 때 받은 시간 A를 체크한다, 그러면 A라는 시간에서 송신자보고 패킷에 있는 DLSR과 LSR 값을 빼면 왕복지연시간이 나온다.

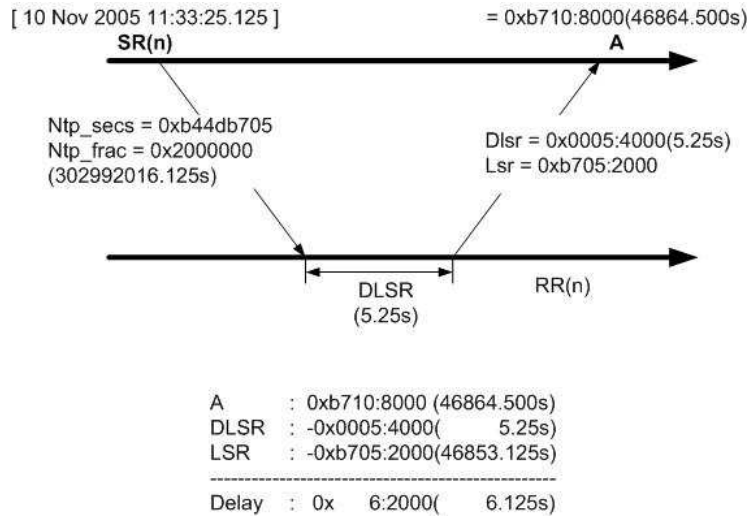


그림 4.7 왕복 지연 시간(RTT) 획득 알고리즘

4.6.2 패킷 손실률 측정법

송신단은 RTP 패킷을 보낼 때 순서번호를 붙여서 보낸다. 이 값은 처음에는 무작위 값을 사용하고 그 다음부터는 RTP 패킷을 보낼때마다 1씩 증가시켜서 보낸다. RTP 패킷을 수신한 수신단은 계속 순서번호를 저장하고 있게 된다. 이와 같이 하면 수신단은 제일 처음에 수신한 패킷의 순서번호를 알고 있고, 그 다음부터 계속 들어오는 패킷의 순서번호를 알고 있게 된다. 송신단이 송신자 보고를 보내면 수신단은 송신자 보고를 받은 후 그 때 까지 받았던 RTP 패킷에서 순서번호가 제일 큰 값에서 초기값을 빼서 전체 받은 패킷의 개수를 구한다. 그 후 지난번 SR 받았을 때까지 받았던 패킷의 개수와 그 이후로 받은 패킷의 개수를 빼서 누적 손실률을 구한다. 패킷 손실률의 경우는 가장 최근에 SR를 받았을 때의 가장 큰 순서번호에서 지금 현재 가장 큰 순서번호를 빼서 구간을 구하고, 가장 최근에 받은 SR 이후에 실제로 수신한 패킷의 개수를 빼면 가장 최근에 SR과 현재 SR 사이에 손실을 RTP패킷의 손실률을 구할 수 있게 된다. 이 관계를 pseudo code로

작성하면 다음과 같다.

```

expected = Highest_seqno - first_seqno + 1;
if (expected_ptr != NULL)
{
    *expected_ptr = expected;
}
cum_lost = (expected - num_pkts_rcvd_until_last_SR -
num_pkts_rcvd_since_last_SR);
expected_interval = expected - expected_prior;
lost_interval = expected_interval - num_pkts_rcvd_since_last_SR;
frac_lost = (lost_interval << 8) / expected_interval;

```

4.6.3 패킷 손실률과 왕복지연 시간의 이동 평균(Moving Average)

패킷 손실률과 왕복지연은 네트워크의 상태에 따라 큰 폭으로 변할 수 있다. 따라서 이 값들로부터 계산되는 유효 전송률 역시 큰 폭으로 변동할 수 있는데 가변을 비디오 전송 시스템에서 유효 전송률의 급격한 변동은 화질의 급격한 변동으로 이어지므로 시각적으로 좋지 않다. 비디오의 입장에서는 채널의 특성이 짧은 시간 간격으로 급격하게 변동하더라도 점진적인 화질 변동을 가져오는 완만한 전송률의 변화가 훨씬 바람직하다. 이와 같은 완만한 변화를 얻는 방법은 네트워크 상태 정보들의 평균값을 이용함으로써 가능하다. 본 연구에서는 패킷 손실률과 왕복지연시간에 대해 다음과 같은 이동 평균값(moving average)을 이용한다[7].

$$p(t) = (1-i) \times p(t-1) + i \times p_{\text{new}} \quad (4.3)$$

$$RTT(t) = (1-j) \times RTT(t-1) + j \times RTT_{\text{new}} \quad (4.4)$$

여기서 $p(t)$ 는 새로운 손실률 p_{new} 를 수신한 후 시간 t 에서 계산하는 패킷 손실률이고 $RTT(t)$ 는 새로운 RTT 값 RTT_{new} 를 계산한 후 시간 t 에서 계산하는 왕복 지연시간을 나타낸다. 손실률 p_{new} 는 RTCP 패킷의 손실비(fraction lost) 필드에 의해 주어지며, RTT_{new} 는 4.1.2절을 통해서 계산된다. i 와 j 는 가중치로서 네트워크 상태에 대한 적응속도를 결정한다. 이 값들이 클수록 새로운 상태에 빨리 적응할 수 있으나 유효 전송률의 급격한 변동을 가져오고, 반대로 작을수록 네트워크의 상태에 둔감하게 반응한다. 일반적인 TCP 체증제어방식에서 i 와 j 는 0.125가 사용되지만 이번 논문에서는 좀 더 빨리 네트워크의 상태에 적응하기 위해서 1와 j 를 0.2로 하여 시스템을 구현하였다[1]

4.7 패킷 우선 전송 알고리즘

제안한 시스템은 패킷 손실률 정보를 통하여 가변적으로 비디오 전송률을 조절함으로써 UDP의 무조건적으로 인한 전송으로 인한 체증을 제어 하여 PSNR값을 보장할 수 있다. 그러나 네트워크의 대역폭이 환자 영상을 전송할 수 없을 정도로 측정된다면, 환자의 진단에 우선시 되는 생체신호 데이터를 우선적으로 전송하여야 한다. 이를 위하여 생체신호의 RTP헤더에 Priority Marker를 추가하여 RTCP에서 얻어지는 네트워크 성능에 따라 우선적으로 전송하도록 한다. 우선 전송이 실시되는 최대 전송 대역폭 값은 네트워크 상태를 통하여 결정된다. 또한 PSNR값이 최고 임계치에 근사했을 경우는 누락되었던 생체신호를 재전송함으로써 중요 데이터의 복원을 가능케 설계하였다.

4.8 시스템의 구현내용

시스템에서 사용하는 영상 코덱은 MPEG-4 비디오 코덱을 사용하였고, 전송은 윈속(Winsock) 2.0을 사용하였으며, GUI 환경을 고려하여 Visual C++ 6.0 을 이용하여 구현하였다. 시스템을 구현한 PC 사양으로는 운영체제는 Windows XP를 사용하였고, CPU는Pentium IV 3.0Ghz, RAM 512Mbytes를 사용하였다. 실시간 비디오 전송 시스템은 크게 서버와 클라이언트로 나뉜다. 실시간 비디오 전송 시스템의 동작을 설명하면 다음과 같다. 서버 측에서는 들어오는 비디오 시퀀스를 부호화하여 비트스트림으로 만든 후, 그 비트스트림은 네트워크를 통해 클라이언트에게 전달한다. 클라이언트는 미리 수신대기 상태에 있다가 서버로부터 데이터를 수신하면 들어오는 데이터 복호화와 디스플레이화 한다. 이러한 일련의 송수신 과정에서 서버는 클라이언트로부터 수신하는 피드백 정보를 기초로 하여 사용자가 입력하는 최소 전송률과 최대 전송률 사이에서 체증제어를 하게 되며, 결과적으로 클라이언트는 주어진 환경에서 최적의 데이터를 전송 받게 된다.

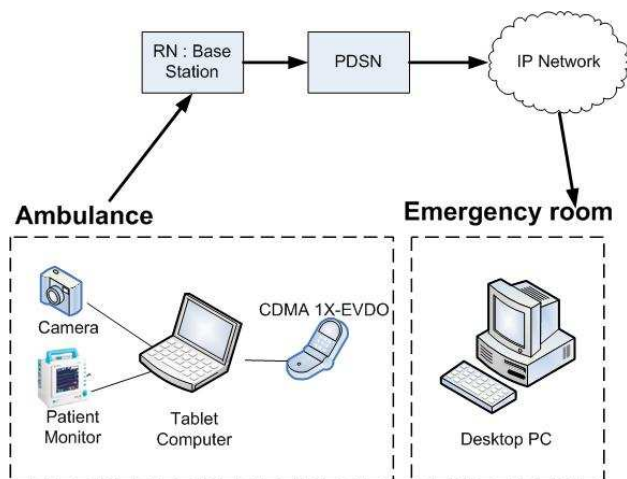
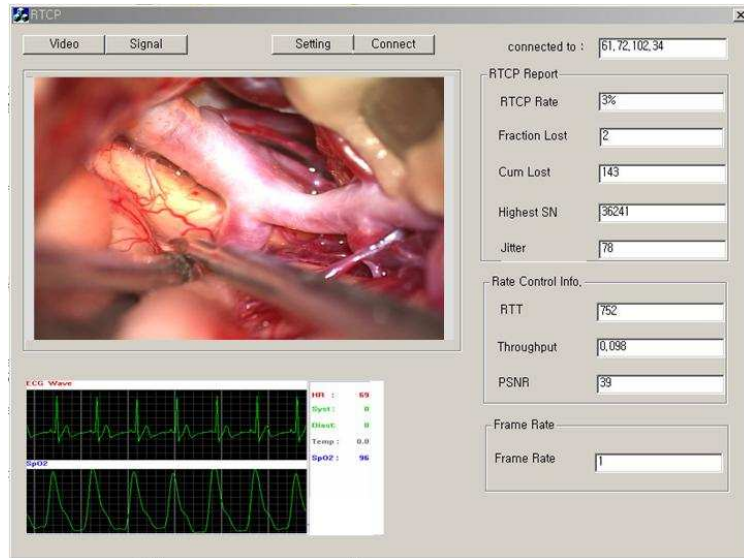
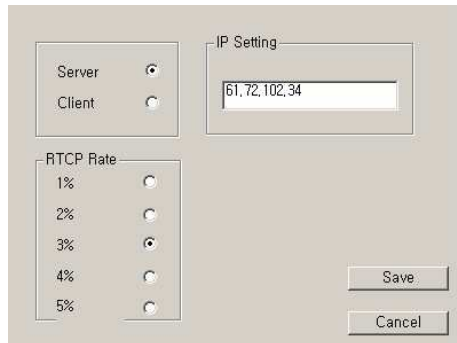


그림 4.8 CDMA 1X-EVDO 망을 사용한 원격 진료 시스템 개념도

그림 4.9 는 원격 진료 시스템의 송신단 화면과 RTCP서버 화면을 나타낸 것이다.



(a)원격 진료 시스템 송신단 화면



(b)RTCP 서버 세팅 화면

그림 4.9 (a)원격 진료 시스템 송신단 화면 (b) RTCP 서버 화면

4.9 실험 환경

CDMA 1X와 EV-DO 환경에 대한 실제 필드 테스트는 두가지 다른 위치에서 진행되었다. 첫 번째는 서울의 외곽지역 고속도로 및 산간이고 두번째는 서울시내 중심부 반경 20km 내외의 교통 정체 및 빌딩이 많은 지역이다. 실험은 2시간 간격으로 아침 10시부터 오후 10시까지 일주일동안 진행되었고 각각의 시도에서 이동 차량의 속도는 0에서부터 120km/hour까지 20km/hour씩 변화시켰다. 특정 시간대의 접속율의 변화와 특정 지역에서의 지역분포에 의한 영향을 보상하기 위해 측정된 값의 평균을 사용하였다. CDMA 1X 실험은 송신단과 수신단에 CDMA 1X 모뎀((Anydata, LG Telecom Co., Korea)을 사용하여 인터넷에 연결없이 직접 연결 되었다. 이때 RLP(Radio Link Protocol)에서 분할되어 전송되는 단위는 5ms 당 64비트 (8비트 패딩(padding)을 고려하면 72비트)이고 R-FCH(Reverse Fundamental CodeChannel)을 사용한다.

CDMA 1X EV-DO 실험은 UDP/IP 기반 프로토콜을 사용하여 이동 차량에는 EV-DO 핸드폰을 노트북에 연결하였고 수신단에서는 ADSL을 사용하여 연결하였다. 여기서의 RLP 전송 단위는 23.6ms당 6400비트이고 R-SCH(Reverse Supplemental Code Channel)을 사용한다. 실험을 통해 측정된 기본적인 값은 속도 변화에 따른 전송 속도, 에러 발생율, 시간에 따른 전송 속도 분포도이며 각각의 망 상태에 따른 지연 시간 및 재전송 시간을 측정하였다.

5. 실험 결과

5.1 CDMA 1X-EVDO 상에서 원격 진료 시스템 평가

CDMA 1X-EVDO를 이용한 의료정보데이터의 송신 시 전송 실험 결과로부터 그림 5.1과 같은 전송 속도의 변화와 패킷 에러율에 대한 분포도를 얻을 수 있다. 이상적 형태로는 이론적 상향 속도인 153.6kbps에 근접한 140kbps의 전송 속도를 평균으로 표준편차 20을 가진 가우시안 분포가 되어야 하지만 실제적인 실험 결과는 속도 변화에 대한 에러의 증가에 따른 RLP 계층에서의 재전송 지연, 핸드오프 지역과 같은 지리적 변화에 대한 전파세기의 약화, 인터넷 네트워크 환경 문제 등의 원인으로 인하여 100kbps 이하의 전송 속도가 간헐적으로 나타난다. 특히 빠른 속도에서의 핸드오프로 인한 지연과 전파 차단 지역에서는 수십kbps 이하의 낮은 전송속도가 유지될 수 있다. 따라서 수신단에서는 지속적으로 전송지연과 대역폭 등을 측정하여 송신단에 알려줌으로써 전송 데이터의 우선권 부여, 전송률 조절, 압축률조절 등과 같은 자동제어가 이루어지도록 설계하여야 함을 알 수 있다.

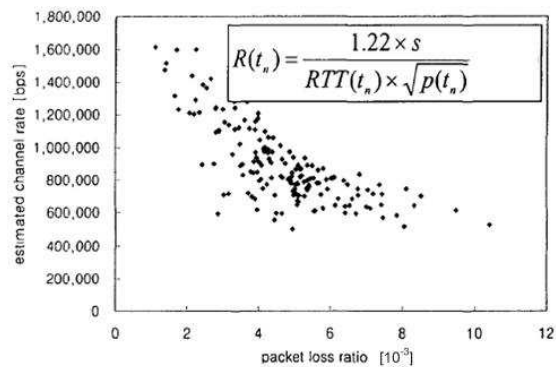


그림 5.1 CDMA 1X-EVDO 망에서 송신 시 대역폭과 패킷 에러 분포도

CDMA 1X-EVDO에서의 데이터 송신 시 평균 지연 값(Round Trip Time)은 그림 5.2와 같다. 무선망에서의 경우 117ms의 평균 지연 값과 평균 42ms의 평균 지연 편차(jitter)가 나타났다. 이는 유선망과 비교하여 평균 지연 값과 지연 값의 편차(jitter)가 큰, 불안한 네트워크 성능을 가지고 있음을 알 수 있다[26]. 이는 단말기와 기지국 간의 송, 수신 강도의 변화, 기지국 내 접속자수의 증감 등으로 인한 네트워크 성능 자체적 지연과 RLP단에서의 패킷 보정으로 인한 지연으로 크게 나누어진다, 이와 같은 불안정한 지연 시간을 가지는 네트워크의 경우, 원활한 실시간 멀티미디어 데이터 전송을 위해서는 응용 계층과 같은 네트워크 계층 위에서 지터에 대한 보정이 필요함을 알 수 있다.

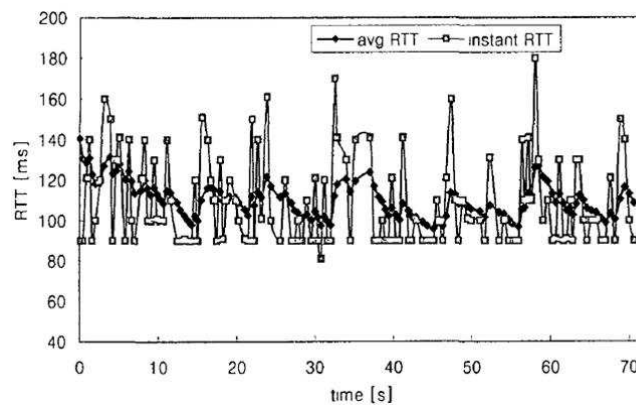


그림 5.2 CDMA 1X-EVDO 망에서 송신 시 왕복 지연 시간 값

그림 5.3은 네트워크 성능 평가를 위한 RTCP 전송을 그래프로 도시한 것이다. 평균손실률(average loss ratio)은 식 (4.3)에 의해 구한 이동 평균값(moving average)을 도시하였다. 그림의 결과와 같이 무선망에서는 네트워크의 순간적인 변화 값이 크기 때문에 이를 곧장 사용할 때는 급격한 전송률 변화로 인한 화질의 악화를 초래 할 수 있다. 따라서 순간 손실 값에 기반한 이동 평균값을 사용하면 순간 손실률(instant loss ratio)이 크게 변하는 경우에도 평균값은 중간부분에서 변하게 됨을 알 수 있다. 따라서 이동 평균 손실률을 사용하여 전송률을 추정하면 추정된 전송률이 크게 변화하지 않고 서서히 변하게 된다. 전송률이 서서히

변하면 부호화율도 서서히 변하게 되며 결과적으로 화질의 급격한 변화를 방지할 수 있다.

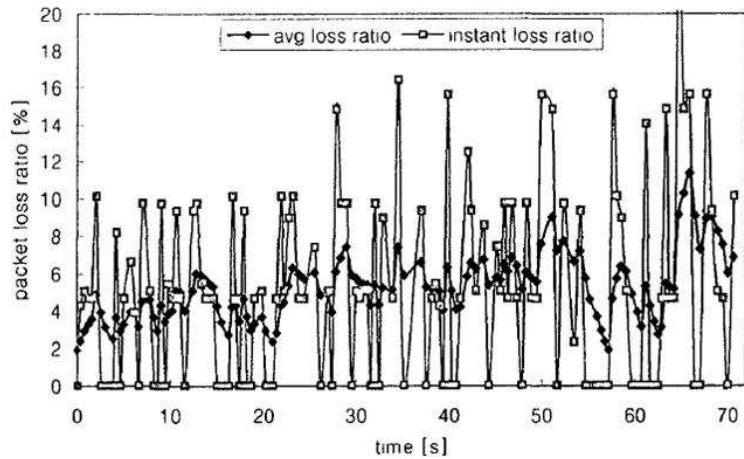


그림 5.3 손실률의 순간값과 이동 평균값 변화

그림 5.4 는 RTP만을 사용 하여 의료정보 데이터를 전송하였을 때와 RTCP 패킷을 이용한 네트워크 품질 데이터를 수집 후 전송률 제어 알고리즘 사용 시의 대역폭 분포도이다. 평균 대역폭 값을 1로 가정하였을 때 프레임 전송률 조절을 통한 데이터 전송 프로토콜이 안정적인 성능을 보장함을 알 수 있다. 이는 RTCP 를 통한 전송률 조절이 네트워크의 병목현상, 체증현상을 사전에 예방하기 때문이다.

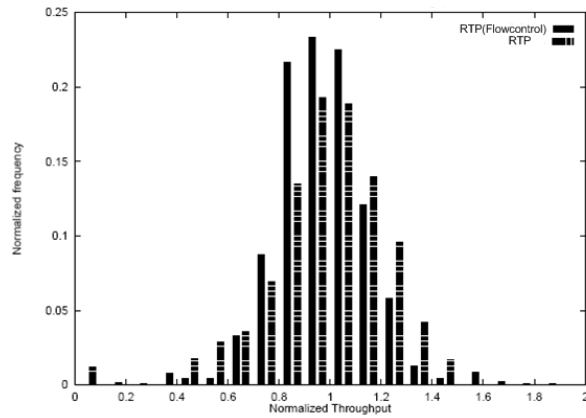


그림 5.4 전송 실험 시 평균 대역폭 분포도

표 5.1은 각 네트워크 프로토콜로 전송 시 평균 PSNR(Peak Signal to Noise Ratio)를 나타낸 것이다. RTP 프로토콜 적용 시 UDP보다 오히려 좋지 않은 성능을 보였다. 이는 RTP 프로토콜은 UDP프로토콜과 마찬가지로 데이터 전송 시 어떠한 신뢰적인 보장도 지원하지 않을 뿐만 아니라, 오히려 응용계층에서의 실시간성의 보장을 위한 헤더 추가와 동기화 과정에서의 부가적 지연 때문인 것으로 추정된다. 반면 프레임률 조절을 사용한 RTP의 경우에는 네트워크 상황에 적합한 대역폭을 선택하여 전송하기 때문에 병목 현상 등으로 인한 패킷 손실을 방지하여 보다 고품질의 환자 데이터를 전송할 수 있음을 알 수 있다.

| | UDP | RTP | RTP with flow control |
|----------|------|------|-----------------------|
| PSNR(dB) | 33.2 | 33.1 | 35.1 |

표 5.1 각 프로토콜 전송 실험 시 PSNR(Peak Signal to Noise Ratio) 값

6. 결론 및 토의

본 논문에서는 RTP 프로토콜을 사용한 실시간 원격 의료 진료시스템을 설계, 구현하였다. 시스템의 설계과정은 크게 두 가지 과정으로 진행되었다.

우선 의료 정보 데이터에 실시간성을 부여하기 위하여 UDP 기반에 RTP를 적용하였다. 무선망에서 TCP를 사용하였을 경우 실시간성이 완벽히 보장되지 않고 무선망에서의 링크 에러를 병목현상 등의 다른 에러로 인식하기 때문에 불필요한 윈도우사이즈 조절로 인한 대역폭 낭비현상이 있었다. 반면 UDP는 무조건적인 전송으로 인한 비 신뢰적인 전송과 QoS가 보장 되지 않는 단점이 있다. 때문에 본 시스템에서는 대역폭을 충분히 활용할 수 있는 UDP 기반에서 RTP 프로토콜을 적용하여 실시간성을 부여하고 중요한 환자 데이터는 우선권을 부여하여 전송하였다. 이와 같은 알고리즘을 사용함으로써 응용계층 아래 단에서의 실시간성 보장 효과를 확인할 수 있었다.

두 번째로는 네트워크의 상태에 유동적으로 대응하기 위한 전송률 제어 알고리즘을 제안하였다. 정확하고 주기적인 네트워크 성능 수집을 위하여 RTCP를 사용하였다. RTCP는 현재 네트워크의 대역폭, 지연, PER 등의 정보를 수집하여 최대 전송 보장 가능한 데이터 사이즈를 송신단에게 알려준다. 주어진 정보를 분석하여 적합한 전송률을 결정, 전송함으로써 안정적인 대역폭을 사용할 수 있고, 무조건적인 전송으로 인한 병목현상을 방지함으로써 환자 영상 품질의 보장에 우월한 성능을 보일 수 있음을 확인하였다.

위와 같이 제안된 알고리즘을 통하여 CDMA 1X-EVDO 망을 통해 환자 데이터를 전송한 결과 기존의 전송 방법 보다 효과적인 성능을 나타낼 수 있음을 확인하였다. 이것은 앰블런스 와 병원을 연결하는 원격 진료 시스템 뿐만 아니라 언제 어디서나 진료를 받을 수 있는 유비쿼터스 헬스 케어 서비스에 효과적으로 사용될 수 있을 것이다.

향후 과제는 크게 세 가지로 나눌 수 있다. 첫 번째는 가변적인 RTCP 전송을

통한 적절한 네트워크 성능 수집이다. IETF에 의하면 RTCP 패킷의 전송 간격은 전체 대역폭의 5% 미만으로 설정하여야 한다고 권고하고 있다. 이는 전체 대역폭의 전송에서 네트워크 상태 측정을 위한 RTCP 패킷이 큰 영향을 끼치지 않기 위함이다. 유선망과 같이 안정적인 네트워크 환경에서는 전송망의 품질 변화가 무선망과 같이 크지 않기 때문에 가변적인 RTCP의 필요성이 적다. 그러나 무선망의 경우 기지국내 접속자 수, 단말기의 이동 속도, 소프트 핸드오프, 하드 핸드오프 등 전송속도의 변화를 끼칠 수 있는 가능성이 동시 다발적으로 일어난다. 때문에 무선망의 안정성과 변화량에 적절히 대응할 수 있는 유동적인 RTCP 전송률 알고리즘이 필요하다.

두 번째로 네트워크 대역폭이 충분히 보장되지 않는 무선망에서 헤더의 비중을 줄일 수 있는 알고리즘의 적용도 고려해야한다. 현재 RTP의 헤더크기는 12byte로 UDP헤더와 IP헤더를 포함하면 40byte가 소요된다. 그러나 CRTP(Compressed Realtime Transport Protocol)를 적용할 경우 헤더사이즈를 최대 4byte 까지 줄일 수 있어 전송 가능한 멀티미디어 데이터 사이즈가 상대적으로 증가하는 효과를 기대할 수 있다.

마지막으로 무선망의 에러를 보정하여 QoS(Quality of Service)를 증가시키는 방법이다. 본 논문에서 구현한 시스템은 RTCP 피드백 메시지를 이용한 채널 적응적인 기법만을 고려하였다. 이는 네트워크의 전송률에 중점을 둔 보정 방법이었으나 HARQ(Hybrid Automatic Repeat reQuest), FEC와 같은 전송된 패킷의 에러를 보정하는 방안이 추가된다면 더욱 무선환경에 강인한 시스템을 구현할 수 있을 것이다.

참 고 문 헌

- [1] qian ZhanB, Weilwu Zilu, Ya-Qin Zhang, "Resource Allocation for Multimedia Streaming Over the Internet," IEEE trnsactions on Multimedia, Vol. 3, No. 3, pp.339-355, Sep.2001
- [2] D. Wu, Y. T. Hou, W. Zhu, H.-J. Lee, T. Chiang, Y.-O. Zhang, and H. J. Chao, "On end-to-end architecture for transporting MPEG-4 video over the Internet," IEEE Transactions Circuits Syst. Video Technology vol 10, pp.923-941, Sept. 2000
- [3] T. Turletti aild C. Huiteina, "Video confrerlclirig on the internet," IEEE/ACM Transactions Networking, vol. 4, pp. 340-351, July. 1996
- [4] Joerg Widnler, Robert Denda, and Ma.tin Mauve, Praktische Informatik IV, "A Survey on TCP-Friendly Congestion Control," IEEE Network, vol. 3, pp. 28-37, May/june 2001
- [5] Jitendra Padhye, Victor Firolu, Don Towsley, Jim Kurose, "Modeling TCP Throughput: A Simple Model and its Empirical Validation"
- [6] RTP : A Transport Protocol for Real-Time Applications, RFC 1889
- [7] 정송 외, IMT-2000망에 적합한 IP기반 멀티미디어 스트림 전송 및 흐름제어 기법, 1차년도 최종보고서, 한국과학기술원, 2002
- [8] James F Kurose, Keith W. Ross, "Computer Networking, A Top-Down Approach Featuring the Internet," Addison Wesley
- [9] 김재균저, "영상통신 시스템", 영지출판사,2000
- [10] J-C. Bolot, T. Turletti, "Experience with Control Mechanisms for Packet Video in the Internet," Computer Common Rev28(1), pp.4~15, Jan 1998
- [11] Janlshid Mahdavi & Sally Floyd, "TCP-Friendly Unicast Rate-Based Flow Control"
- [12] 권재철, 정요원, 강희덕, 정순홍, 서만근, "인터넷 망에서 방송데이터의 실시간

- 전송을 위한 전송률 추정기법 연구”, 한국과학기술원 ETRI 과제 최종 보고서, Nov.2002.
- [13] Text of fso/ rEC 14496-2 MPEG-4 Video FCS v. 18.0, N3908, Pisa, Jan.2001.
- [14] Dapeng Wu, Yiwei Thomas Hou, Y.0. Zhang, "Transporting Real-Time Video over the Internet Challenges and Approaches", Proceedings of the IEEE, vol.88, no. 12, Dec.2000
- [15] 원속 프로그래밍, SAMS publishing
- [16] D. Wu, Y. T. Hou, Y. 0. Zhang, "Scalable Video Coding and Transport over Broad-Band Wireless Networks", Proceedings of The IEEE, vol.89, no. 1, pp.6~20, Jan. 2001
- [17] Ming-Ting Suil, Anly R. Reibnlan, "Compressed Video over Networks", 2001
- [18] Text of ITU-T 015-A-59 Video Codec Test Model, Near-Term, Version 8, June 1997
- [19] "Introduction to the Special Issue on Wireless Conlunicationl," IEEE Trans. on Circuits and Systems fer Video Technology, vol. 12, no.6, June.2002-12-27
- [20] 김현욱 외, "IMT-2000 이동통신 원리", 진한도서, pp.1-200, pp.507-620, 2001.5.
- [21] 정보환, "CDMA 2000 1x EV-DO", 프로그램세계, pp.950-951, 2002.11.
- [22] F. Khafizov and M. Yaruz, "Analytical model of RLP in IS-2000 CDMA networks"
- [23] T. Sakamoto and D. Wei, "Implementation of Mobile ECG Telemonitor Using Real-time Transport Protocol(RTP)," in Proc. IEEE EMBS Asian-Pacific Conference, pp. 32-33, 2003
- [24] T. Sakamoto, T. Ohya, T. Kawahara and M. Etoh, "Rate and Robustness Control with RTP Monitoring Agent for Mobile Multimedia Streaming," in

- Proc. Int Conf, on IEEE Communications(ICC 2002), vol. 4, pp. 2513-2517, 2002
- [25] C.S. Pattichis, et al. Wireless Telemedicine Systems. An Overview IEEE Antenna's and Propagation Magazine, vol. 44, no.2, pp. 143-153, 2002
- [26] HY Yun, SK Yoo, DK Kim. "Performance Evaluation of Telemedicine System based on multicasting over Heterogeneous Network", International Conference of the IEEE Engineering in Medicine and Biology Society,2005.12 Sanghai in China

Abstract

For quick and correct treatment of disease, real-time telemedicine system that enables medical examination and treatments made by several specialists simultaneously was designed. The system must transmit good quality data without concerning network that has different bandwidth by real time. The important issue of real-time telemedicine system is end-to-end delay constraint. Therefore, UDP(User Datagram Protocol) is more suitable for transporting multimedia data than TCP(Transmission Control Protocol). But UDP does not control network congestion and guarantee QoS(Quality of Service). There are many research results to complement UDP. One of these results is RTP(Real-Time Transport Protocol) and RTCP(Real-Time Transport Protocol). RTP and RTCP is mainly designed for use with UDP(User Datagram Protocol) for multimedia transport over the Internet. It has the capability of media-synchronization and network's QoS feedback to compensate for the weakness of UDP. In this thesis, the RTP/RTCP UDP protocol is implemented over a windows pc system, and integrate with a MPEG-4 video codec system. The performance of channel rate control algorithms for VBR(Variable Bit Rate) video transmission is compared by the implemented test system. They are a TCP-Friendly rate control algorithm and a modified algorithm with a bounded minimum and maximum bitrate. It is shown that the modified algorithm provide wider range of controlled bit rates depending on the packet loss ratio and the throughput than the TCP or UDP algorithm.

In this paper, we designed and evaluated the real-time multimedia telemedicine system using RTP over CDMA 1X-EVDO. To evaluate this system, we designed the telemedicine system which is based on the RTP. RTP can guarantee realtime transmission at the transport layer, that can't guarantee at the network layer. and then we designed the RTCP protocol. The RTCP Packets can analyze network traffic and

packet loss rate. Using RTCP report, we can control the QoS(Quality of Service) such as transmission rate control or priority control.

The performances of the proposed algorithm and system in terms of throughput variation, RTT(Round Trip Time), jitter and PSNR(Peak Signal to Noise Ratio) were shown better performance more than UDP or native RTP.

Key words : real-time telemedicine, CDMA 1X-EVDO, UDP, RTP, RTCP, flow rate control, frame rate control