

## RTP 기반의 JPEG2000 영상 전송 시스템 구현

박 동 진, 정 영 기  
호남대학교 컴퓨터 공학과  
전화 : 940-5404 / 핸드폰 : 019-9164-1028

### Implementation of RTP based Image Transport System using JPEG2000

Dong-Jin Park, Young-Kee Jung  
Dept. of Computer Engineering, Honam University  
E-mail : mpeguydj@hotmail.com

#### Abstract

In this paper, we propose RTP(Real-Time Transport Protocol) based image transport system to transport still images in real-time after JPEG2000 compression, which is still image compression standard for next generation. To add RTP packet on UDP packet, the image transport system inserts packetizer and depacketizer process into transmitter and receiver of RTP data, respectively. We apply the proposed system to several image and compare the transport time to TCP-based method.

#### I. 서 론

영상은 정보를 표현하는 수단으로 사용되어 왔으며, 컴퓨터 기술의 발달과 고속 통신망의 보급으로 인터넷을 이용한 영상의 전송은 확대되고 있다. 이러한 영상은 압축의 과정을 거쳐 전송하게 되는데 정지 영상 압축에는 이산 역현 변환(DCT, Discrete Cosine Transform)을 기반으로 하는 JPEG 표준을 대표적으로 사용하여 왔다.

그러나, JPEG은 저조한 압축 성능, 통합되지 못한 유/무손실 압축 시스템, 대용량 영상에의 적용 불가능, 잡음 환경에서의 낮은 전송 안정성, 그래픽등과 같은 인위 영상의 압축성능 저하, 복합문서의 압축성능 열화등 여러가지 단점을 가지고 있다.

이에 JPEG 위원회에서는 기존 JPEG 보다 우수한 성능을 가지면서도 다양한 형태의 영상에 적용할 수 있는 새로운 정지영상 압축표준으로 JPEG2000을 제정하였다.

최근 압축된 정지영상 데이터를 네트워크상에서 전송하는 방법으로써 일반적으로 사용되고 있는 TCP기반의 HTTP, FTP들과는 달리 주로 미디어 스트림에 대한 전송을 목적으로 설계된 RTP를 UDP 패킷상에 추가시키는 방법으로 구현되고 있다.

UDP 전송이 사용된 배경에는 UDP 전송은 전송을 보장하거나 안정성을 보장하는 프로토콜이 아니므로 실제로 UDP 전송이 이상하게 여겨질 수 있으나, 데이터 전송의 손실에 대한 복구는 수신단의 부담으로 남겨두고, 또한 네트워크 오버헤드나 어플리케이션 구동에 대한 제약이 적다는 데 있다.

본 논문에서는 정지영상의 실시간 전송을 위해 차세대 정지영상 압축표준인 JPEG2000 부호화 / 복호화 알고리즘을 이용하고, RTP 기반의 UDP 패킷을 통해 전송하는 방식으로 영상 전송 시스템을 구현하였다.

그림 1은 영상 데이터를 입력 받아 실제 영상을 JPEG2000으로 압축한 후 실시간 프로토콜인 RTP를 사용하여 네트워크를 통하여 목적지 호스트로 전송하고 수신측에서 데이터를 복원하여 디스플레이하는 본 논문에서 구현한 시스템의 전체적인 흐름을 개략적으로 보여주고 있다.



### III. RTP기반 실시간 전송 시스템 구현

일반적인 엄격한 타이밍 요구조건을 갖는 TCP기반의 HTTP, FTP와는 달리, RTP는 주로 미디어 스트림에 대한 전송을 목적으로 설계되었으며, 모든 RTP 버퍼들은 타임스태프를 가지고 있어서, 타임스태프와 실제 전역적인 동기화된 클럭 사이의 매핑을 수행한다.

타임스태프는 동일한 번호를 지님으로서 특정한 동일시간에 함께 복호화 되어야함을 의미한다. RTP역시 UDP 기반의 프로토콜로서 수신측에서는 수신되어지는 데이터 패킷이 일정하게 알맞은 순서로 전송되었는지 알수 없다. 그러므로 수신측에서는 패킷의 전송순서번호를 이용하여 패킷을 재 정렬해야한다.

일반적인 다른 네트워크 응용들도 마찬가지로만 본 구현 시스템에서도 데이터 송수신 전에 수신측과 송신측 사이에 각각의 IP와 포트번호를 이용하여 세션을 확립한 다음 실제 압축 데이터를 전송하게 구현하였다. 또한 동시에 타임스태프와 전송순서번호를 초기화하고 같은 RTP 세션 내에서 같은 값을 갖는 동기화 소스(Synchronization Source)가 두개 이상 나타나지 않도록 식별하는데 사용하는 SSRC를 생성한다.

이때, RTP는 TCP/IP와 같이 데이터 전송을 제어하여 서비스 품질을 보장하는 기능이 존재하지 않는다.

RTP 데이터 전송 기능은 제어 프로토콜에 의해 확장되는데, RTCP(Real-Time Control Protocol)는 전송 제어 프로토콜로서 데이터의 전달 상황을 감시하며, 최소한의 제어 기능과 매체 식별 기능을 제공한다. 이 RTCP를 위한 포트번호가 기준 포트번호에 자동으로 '1'이 더해져서 설정되게 하였다. 그런 후 이렇게 만들어진 RTCP 소켓을 통해 실제 RTCP 패킷 전송을 위한 기반 자료들이 구해진다. 이러면 이제 송신측에서는 수신측의 정보만 있으면 데이터 전송을 개시할 수가 있다. 데이터를 전송할때도 여러 사항이 고려되어야 하는데 이때는 실제 목적 데이터를 패킷의 페이로드 크기만큼 페이로드 타입의 패킷 형태로 전송하는데 RTP 패킷의 마크필드와 전송할 때 사용되는 타임스태프 증가의 명시도 고려하여 구현하였다.

RTP 패킷을 전송한 이후 송수신 통계치의 정밀도를 최대화 시키기 위해 RTCP 패킷의 전송이 유효한지 체크한 후 유효하다면 RTCP 패킷이 전송된다.

수신측에서는 세션을 확립한 상태에서 수신대기 모드를 유지하다가 RTP나 또는 RTCP 소켓으로 데이터가 들어오면 RTP 패킷을 회수하거나 RTCP 패킷으로부터 정보를 수집한다. RTP 패킷을 회수하면 수신한 데이터를 그 데이터의 길이만큼 버퍼에 저장해 놓은 후 회수한 패킷을 송신측에 대한 주소와 포트번호를 알아낸다. 다음 사항으로는 버퍼에 저장된 데이터가

유효한지를 검사하는데, 즉 RTP 패킷의 헤더의 내용과 페이로드를 구분해 내는 과정이다. 이제 헤더와 페이로드가 구분지어 졌으므로 SSRC에 대한 정보를 추출하여 현재 수신측에 저장된 다른 소스의 SSRC와 충돌이 일어나는지를 검사한다. 그런 후에 실제 페이로드의 데이터를 추출하여 압축된 영상을 복원하게 된다.

RTCP에 대한 데이터 처리도 RTP의 것과 유사하다.

일단 RTCP 포트로 들어오는 RTCP 데이터를 획득하고 이를 저장한다. 다만 한가지 차이점은 RTCP 패킷들의 유형이 SR/RR + SDES + BYE로 구성되기 때문에 패킷을 처리할 때 이를 구분해서 처리하고 이에 대한 데이터도 나누어서 저장한다는 것이다.

그림 5는 이런 RTP를 이용한 데이터 전송시에 송신측 시스템의 흐름을 대략적으로 나타낸 것이다.

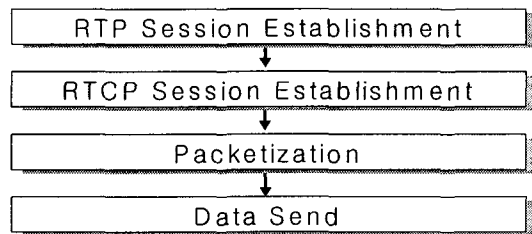


그림 5. RTP 이용 데이터 전송 시의 시스템 흐름.

그림 6은 본 구현 시스템이 RTP 송수신을 위해 데이터의 관리를 어떻게 처리하고 있는지 각 데이터가 처리되는 과정을 전반적으로 나타내고 있다. 버퍼를 제외한 각각의 과정은 쓰레드로 구현되었고, 각각의 해당 버퍼를 사용하기 위한 임계 구역을 형성하였다.

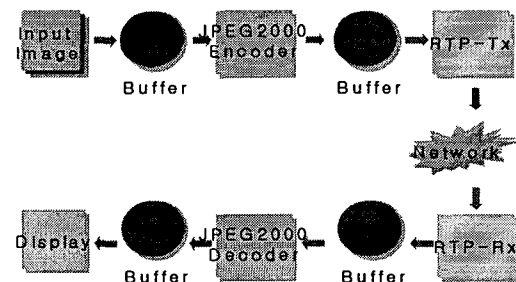


그림 6. RTP 송수신을 위한 시스템의 배치

입력 영상은 버퍼에 저장되며 이때 버퍼체크 플래그와 함께 버퍼에 대한 제어권이 임계구역에서 대기 중인 부호기에 넘어가게 되고 부호기에서는 버퍼체크 플래그를 사용하여 버퍼를 체크한 후 버퍼로부터 프레임 하나에 해당하는 데이터를 받아들여 부호화를 하

고, 다음 버퍼로 데이터를 저장시킨다. RTP\_Tx에서도 마찬가지로 버퍼에 대한 제어권과 해당 버퍼의 체크 플래그를 넘겨받아서 패킷타이제이션 과정을 수행한 후 실제 네트워크를 통하여 목적지로 데이터들을 전송한다.

#### IV. 실험 결과

그림 7은 SPIHT알고리즘의 BPP에 따른 정지영상 압축/복원에 대한 테스트 결과로써 샘플영상은 LENA.raw 파일로 크기는 256\*256 사이즈이고 용량은 65536 바이트이다. 처리되는 RTP 패킷의 페이로드 사이즈는 가변길이 설정을 갖는다. 따라서 압축한 LENA영상은 RTP 패킷을 수신측으로 전송할 때 단 하나의 패킷으로 전송해서 복원 할 수가 있다.



그림 7. BPP에 따른 전송

표 1은 160\*120, 256\*256, 320\*240 각각의 원영상과 압축영상들을 TCP와 RTP 전송했을 때의 전송시간을 측정 후 비교한 것이다. 이때 원영상들에 대한 TCP 전송은 데이터가 클수록 전송제어 절차로 인한 전송 지연이 발생하여 RTP전송이 TCP전송보다 약 36% 빨랐다. 또한 압축영상들의 전송 시간을 비교해 보았을 때 RTP전송이 약 34% 빨랐다. 그림 8은 표 1을 그래프로 나타낸 것이다.

단위 : ms

Bytes	TCP	RTP	Bytes	TCP	RTP
19200	5	3.2	14044	4	2
66536	15	11.6	28525	7	5
76800	27	13.4	40147	9	6.8

원영상
압축영상

표 1. 영상의 TCP와 RTP 전송 비교

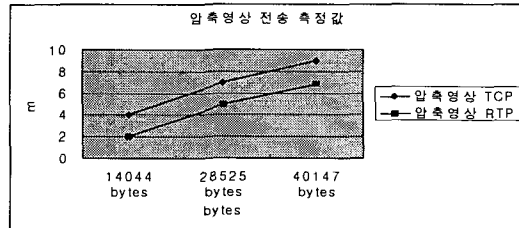
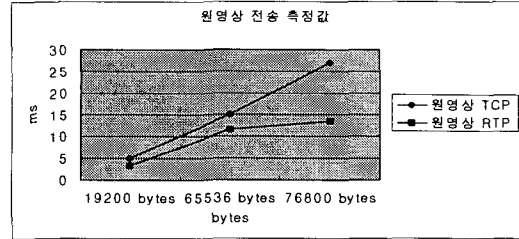


그림 8. 영상의 TCP와 RTP 전송 비교 그래프

#### V. 결론

본 논문에서는 SPIHT에 의해 이미지 하나만을 RTP 프로토콜을 사용하여 실시간 전송을 실행하는 시스템을 실제 구현하여 검증하였다. 제안 시스템은 그림 7에서 보여지고 있는 것처럼 적은 BPP를 가지는 저해상도의 이미지를 먼저 전송하고 시간의 경과에 따라 점차적으로 해상도가 높아지는 프로그레시브 전송을 RTP 기반으로 구현할 수도 있을 것이다. 또한 화상회의와 유사하게 화질상의 높은 성능을 요구하는게 아닌 전송속도에 비중을 둔 연속적인 영상 데이터의 전송같은 응용에도 적용 가능할 것이다.

#### 참조 문헌

- [1] David S. Taubman, Michael W. Marcellin, "JPEG 2000 Image Compression Fundamentals, Standard and Practice", 2002.
- [2] A. Said, W. A Pearlman, "A New Fast and Efficient Image Codec Based on Set Partitioning in Hierarchical Trees", IEEE Transactions on Circuits and Systems for Video Technology June. 1996.
- [3] Jori Liesenborgs, "Voice over IP in networked virtual environments", Phd. Thesis, 2000.
- [4] 정영기, "실시간 멀티미디어 전송 프로토콜 구현" 한국통신학회 춘계학술 발표회, 2000년.
- [5] RFC2435, "RTP Payload Format for JPEG - compressed Video", Oct. 1998.