

# Motion JPEG와 RTP를 이용한 Web-based 화상 통화 구현

유재욱, 박인갑, 김중민  
건국대학교 전자정보통신공학과  
전화 : 02-450-3495 / 핸드폰 : 019-499-2597

## Implementation of Web-based Image Communication System using Motion JPEG and RTP

Jae-Wook Yu, In-Kap Park, Joong-Min Kim  
Dept. of Electronic Engineering, Konkuk University  
E-mail : ulty@konkuk.ac.kr

### Abstract

In this paper, we implemented a web-based communication system using Motion JPEG and RTP. MJPEG compression method is used for compression of the image in order to film and transmit the moving image information in the real time, and the RTP(Real-Time Transport Protocol) which differs from the existing TCP and UDP is used for the data transmission.

### I. 서론

인터넷 환경에서의 화상통신은 주로 화상채팅, 화상회의 등의 형태로 구현되어 있다. 이러한 서비스가 사용되는 이유는 문자만으로 이루어진 채팅, 회의 서비스는 상대가 누구인지 확인할 수 없다는 단점이 있어서 서로간의 신뢰를 주지 못하나 화상 통화를 이용하게 되면 서로의 모습을 영상물 통해 확인 할 수 있어 신뢰를 줄 수 있다. 또한 키보드를 주 입력기로 사용하므로 키보드 사용이 불편한 사람은 대화 참여가 원활하지 않다는 단점이 있으나 화상통화의 경우에 음성으로 대화가 가능하므로 키보드 조작이 미숙해도 대화에 참여가 가능하다는 장점이 있다.

본 논문에서는 인터넷 환경에서 쉽게 접할 수 있는 웹 브라우저에서 동작하는 화상통화 시스템을 구현하였다. 일반 인터넷 환경에서는 QoS가 보장되지 않아 영상과 음성을 빠르게 전송하기란 쉽지 않다. 이러한 인터넷 환경에서 최적의 영상과 음성을 빠르게 전송하기 위해서 전송방식은 기존의 TCP, UDP와 다른 개념의 RTP(Real-Time Transport Protocol)를 이용하였고 압축 방식은 기존의 MPEG2 대신 Motion JPEG 압축 방식으로 구현하였다. 또한 기존의 화상통화 시스템은 윈도우상에서 특정 프로그램을 이용하여 사용되었으나, 본 시스템은 웹 브라우저를 통해 이용할 수 있도록 하여 운영체제의 제약 없이 이용할 수 있도록 하였다.

### II. 배경이론 및 관련 연구

#### 2.1 RTP(Real-Time Transport Protocol)

화상통신의 경우와 같이 실시간으로 발생하는 멀티미디어 데이터의 경우에는 빠른 속도가 가장 중요하며 부가적으로 안정성을 포함하고 있어야만 한다. 이러한 특성으로 인해 기존 프로토콜과는 다른 프로토콜이 필요로 하게 되었고 멀티미디어 데이터에 맞도록 설계된 RTP를 화상통신에 이용하게 되었다.

RTP는 오디오, 비디오 및 시뮬레이션 데이터와 같은 실시간 데이터를 멀티캐스트 또는 유니캐스트 네트워크를 이용해서 전송하는 응용 서비스에 알맞은 단말-단말 네트워크 전송 기능을 제공한다. RTP는 자원 예약을 수행하지 않으며, 따라서 적시 전달, 순차 전달과 같은 서비스 품질도 보장하지 않는다. 따라서 RTP 데이터 전송 기능은 제어 프로토콜에 의해 확장되는데, RTCP라 불리는 제어 프로토콜은 데이터의 전달 상황을 감시하며, 최소한의 제어 기능과 매체 식별 기능을 제공한다. RTP와 RTCP는 하위의 전송 및 네트워크 계층에 무관하게 설계되었다.

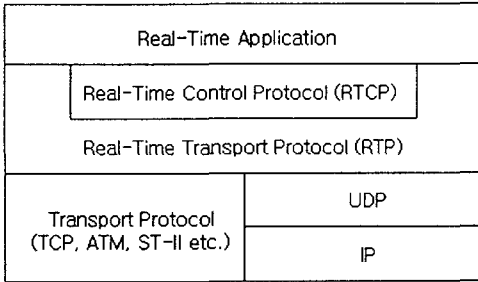


그림 1 RTP Architecture

그림 1과 같이 RTP와 RTCP는 TCP와 UDP의 상위에 존재한다. RTP는 실시간 멀티미디어 전송을 위해서 UDP를 이용하게 되고 RTCP는 정확한 컨트롤 정보를 전달하기 위해서 TCP를 이용한다. 이러한 RTP는 별개의 독립 계층으로 구현되기보다는 특정 응용에서 요구되는 정보를 제공하여 프로토콜의 처리가 응용의 처리 과정으로 통합될 수 있도록 설계되었다. 따라서 기존의 프로토콜들과는 달리 RTP는 응용의 필요에 따라 헤더를 변경하거나 추가하여 응용에 맞는 프로토콜이 될 수 있도록 하는 일종의 맞춤형 프로토콜이다.[1]

2.2 Motion JPEG

화상통화를 하기 위해서는 많은 양의 동영상 정보가 네트워크를 통해 전송되어야 한다. 카메라를 통해 입력 받은 동영상 정보는 압축과정을 통해 데이터의 크기를 줄여서 전송하게 된다. 동영상 압축과정에는 실제 데이터의 손실 없이 얼마나 깨끗한 영상을 전송할 수 있는가에 따라 여러 가지 압축 방식이 있다.

영상 압축 방식에는 대표적으로 MPEG와 JPEG로 구분 할 수 있다. MPEG는 시간에 따라 연속적으로 변화하는 동영상 압축과 코드 표현을 통해 정보의 전송이 이루어질 수 있는 방법으로 종류로는 MPEG1,

MPEG2, MPEG3, MPEG4가 있다.

JPEG는 정지된 화상을 압축하는 방법으로 풀 컬러(full-color)와 그레이 스케일(gray-scale)의 압축을 위하여 고안되었으며, 사진이나 예술분야의 작업에서 장점을 나타낸다. GIF와 함께 인터넷에서 가장 자주 사용되며 GIF에 비해 데이터의 압축 효율이 더 좋다. 또한 GIF는 256색을 표시할 수 있는데 반해 JPEG는 1,600만 색상을 표시할 수 있어 고해상도 표시장치에 적합하다. JPEG의 유용한 점은 이미지를 만드는 사람이 이미지의 질과 파일의 크기를 조절할 수 있다는 점이다.[2,3]

Motion JPEG 또는 Moving JPEG이라고 불리는 MJPEG 압축 방식은 정지화상 압축 방식인 JPEG를 동영상 압축에 응용한 것으로 MJPEG의 각각의 프레임은 JPEG으로 압축되어 있다. MJPEG에서 각 프레임은 앞과 뒤의 프레임과 관계없이 단일 프레임으로서 압축하고 동영상 재생정보를 함께 압축하여 MPEG보다 압축률이 더 높다. 또한 압축을 할 때 압축률을 조절할 수 있어 화질 조정이 가능하며 다른 압축 방식에 비해 계산량이 적어 빠르게 압축 할 수 있다.

2.3 Web-based program

웹 브라우저에서 동작하는 프로그램을 개발하기 위한 방법으로는 Microsoft의 ActiveX, SUN의 JAVA 등을 이용하는 것이 대표적이다. 이 중 ActiveX는 주로 윈도우 계열의 운영체제에서 동작하기 때문에 운영체제에 독립적인 시스템을 구현할 수 없다. 그러나 JAVA의 경우에는 Java VM(Virtual Machine)이 설치되어 있으면 운영체제에 상관없이 시스템을 동작시킬 수 있는 장점이 있다. 또한 대부분의 웹 브라우저에는 Java VM이 포함되어 있어 웹기반 프로그램 개발에 많이 이용된다. 본 논문에서도 JAVA를 이용하여 구현하였다.[4]

JAVA나 ActiveX에 관계없이 웹 브라우저에서 USB 카메라와 같은 H/W를 이용하거나 하드디스크에 있는 파일을 이용하기 위해서는 허용권한이 필요하다. 허용권한은 웹 브라우저를 통해서 실행되려는 코드가 안전하며, 사용자의 시스템에 악영향을 미치지 않는다는 것을 보장하는 것이다. Microsoft사의 IE 4.x에서는 HIGH, MEDIUM, LOW 세 단계의 허용 수준을 두고 있어 USB 카메라에 접근하기 위해서는 허용 수준을 LOW로 조절을 해야 한다. 웹 브라우저에서 허용 수준을 조정하지 않고 허용권한을 얻기 위해서는 프로그램 코드를 디지털 서명된 CAB 파일로 압축하여 배포해야 한다.[5]

### III. Web-Based 화상 통화 구현

#### 3.1 RTP 송수신 구현

RTP로 전송을 받는 부분은 기존의 네트워크 전송프로 그램하고 차이가 있다. 기존의 방식은 전송측과 수신 측이 먼저 네트워크 연결을 하고 연결이 성립되면 데이터를 주고받는 방식으로 동작했다. 그러나 RTP에서는 Session을 통해서 데이터를 주고받는다. Session을 이용하게 되면 전송하는 측은 Session을 지정된 IP와 Port번호로 만들고 데이터 전송을 시작한다. 수신측은 이와 반대로 Session을 지정된 IP와 Port번호로 만들고 데이터를 기다린다. 이렇게 Session을 이용하게 되면 수신측과 송신측이 연결이 성립되기 전에 데이터 전송부분과 수신부분이 동작하고 뒤에 양쪽의 Session이 모두 열리면 연결이 성립되면서 수신측은 데이터를 받아서 화면에 보여줄 수 있게 된다. 이러한 특성은 Real-Time 동영상 정보는 전달되는 현 시점이 아니면 이용할 수 없는 데이터이므로 먼저 전송을 해도 수신측에서 연결이 성립한 시점부터 받아들여 보여주므로 문제가 되지 않는다.

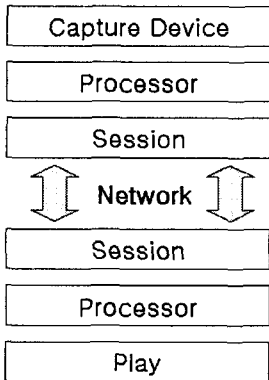


그림 2 RTP 송수신

그림 2에서 Capture Device는 음성의 경우 마이크, 화상의 경우 USB 카메라가 된다. Capture Device에서 데이터를 받는다. 데이터는 Processor를 거쳐서 전송이 가능하도록 압축 알고리즘을 이용하여 압축하게 된다. 압축 알고리즘은 음성의 경우 H.323을 이용하고 영상의 경우 MJPEG를 사용한다. 압축된 데이터를 전송할 때 Session을 이용한다. Session은 음성과 영상을 각각 분류하여 두개의 Session을 이용하는 방법과 음성과 영상을 합성하여 하나의 Session을 이용하여 전송하는 방법이 있다. 수신측에서는 송신측과 반대의 과정을 거쳐 Player를 통해 영상을 화면에 보여주고 음성은

스피커를 통해 들려준다.

#### 3.2 웹 브라우저 보안

웹 브라우저에서 USB 카메라에 접근을 하기위해서 본 화상 통화 시스템 코드를 디지털 서명된 CAB 파일 형태로 제작해야했다. 디지털 서명을 위해서 AnyCert 사로부터 발급받은 인증키(\*.pvk, \*.spc)를 이용하였다. 디지털 서명된 CAB 파일을 만들고 웹 브라우저에서 실행하게 되면 그림 3과 같은 인증기관과 코드의 정보를 담은 보안 경고를 보이면서 실행할 것인지를 묻는다. 사용자가 보안경고의 내용으로 코드가 믿을 수 있는 내용으로 판단하면 웹 브라우저를 통한 H/W 접근이 가능해진다.

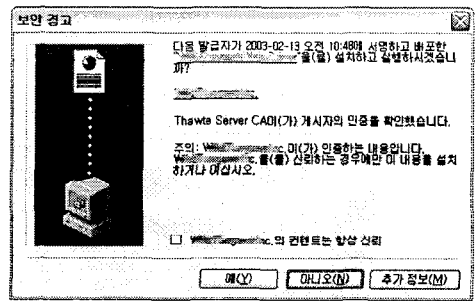


그림 3 웹 브라우저 보안 경고

#### 3.3 화면 구성

화면 구성은 그림 4와 같이 상대방과 본인을 모두 보여주는 화면과 상대방만을 보여주는 화면을 선택할 수 있도록 구현하였다. 음성은 Echo현상을 피하기 위해 상대방의 음성만을 들려주도록 구현하였다.

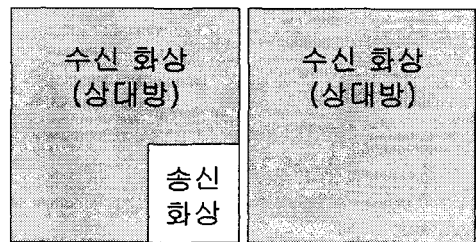


그림 4 화면 구성

본 논문에서는 320X240 pixel의 영상을 캡처하여 전송하도록 구현하였다. 이때 그림 4의 송신화상 부분은 320X240 pixel크기의 영상을 160X120 pixel 정도로 축소시켜 재생해야 하는데 압축을 거친 데이터를 이용하게 되면 영상의 화질이 떨어지게 되어 Capture Device로부터 받은 데이터를 실시간으로 복제하는 방법을 이

용한다. 즉, 데이터를 받는 순간 두개의 데이터를 만들어 하나의 데이터는 압축과정에 이용하고 다른 하나의 데이터는 송신화상 재생에 이용한다.

### 3.4 화상 통화 시스템의 처리 단계별 영상비교

테스트는 세 대의 PC를 이용하였다. 서버 PC는 IIS 웹 서버를 이용하였고 두 대의 클라이언트 PC는 USB 카메라가 설치된 윈도우 XP에서 IE6.0을 이용해 서버에 접속하였다.

아래 그림 5, 그림 6을 보면 Capture Device에서 받은 영상과 MJPEG 압축을 수행한 영상은 데이터 복제를 통해 생성된다. 따라서 Device에서 받은 영상과 압축을 수행한 영상을 동시에 캡처했을때 압축 알고리즘 수행 시간만큼 차이가 나타나야 하지만 두 영상에서 시간적인 차이점을 파악하기 힘들었고 네트워크를 통해 전송받은 영상인 그림 7의 경우에는 화질의 차이는 없었으나 전송시간에 따른 시간지연이 나타났다.



그림 5 Device에서 받은 영상



그림 6 MJPEG 처리 영상

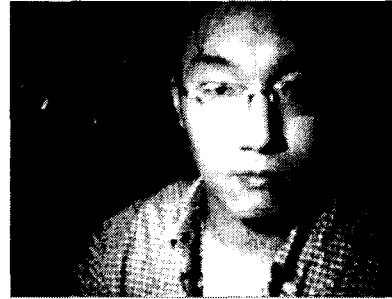


그림 7 전송받은 영상

## IV. 결론

본 연구에서는 프로그램을 따로 설치하지 않고 웹 브라우저를 이용하여 운영체제에 독립적으로 구동되는 화상통화 시스템을 구현하였다.

QoS가 보장되지 않는 일반 인터넷 망에서 동영상 정보를 빠르게 전송을 하기 위해서 영상압축 방법으로 MJPEG 압축 방법을 이용하였으며 데이터 전송 방법은 기존 TCP, UDP와는 다른 RTP를 이용하였다. 또한 시스템이 운영체제에 독립적으로 동작하도록 웹 브라우저 상에서 JAVA를 이용하여 구현하였다.

본 연구에서는 웹을 기반으로 화상통화 시스템을 구현하였으나 구현에 사용된 MJPEG과 RTP는 화상회의, 실시간 방송 등 여러 응용 시스템에 이용될 수 있을 것이다.

## 참고문헌

- [1] IETF RFC 1889, RTP: A Transport Protocol for Real Time Applications
- [2] 이문호, "Visual C++ 실용 영상 신호처리", 대영사, 2001년 6월
- [3] 최형일 외, "영상처리 이론과 실제" 홍릉과학출판사, 1997년 11월
- [4] 이상엽, "Internet Programming Bible", 영진출판사, 1996년 6월
- [5] MSDN platform SDK: Security, Microsoft, 2002년 7월