

# RTP 패킷 손실 감소에 관한 연구

\*서원범, 서덕영

## A Study on Reduction of RTP Packet Loss

\*Wonbum Suh, Doug-Young Suh

Dept. of Electronic Eng. Kyung-Hee University

wbsuh@earth.kyunghee.ac.kr

### 요약

Network의 QoS를 결정하는 인자에는 손실과 지연, jitter 등이 있다.

그 중에서 손실의 양을 최소화하기 위하여 패킷의 크기를 될 수 있는 한 작게 보냄으로서 라우터에서 생기는 손실을 최소화하는 방법이 있다.

본 연구에서는 비디오 데이터의 전송에 있어서 패킷의 크기를 현재 망의 상황에 고려하여

여러 개의 소켓으로 나누어 보냄으로서 손실의 양을 최소화한다.

### 1. 서론

멀티미디어 데이터의 전송에 있어서 정해진 대역폭 안에서 최상의 효과를 이끌어 내기 위한 많은 연구가 진행되어지고 있다.

최근 들어 통신 속도의 급격한 성능 향상과 인터넷의 폭발적인 보급으로 인하여 멀티미디어에 관한 사용자의 요구가 증가됨에 따라, 많은 수의 인터넷 기반 방송용 어플리케이션이 나오고 있다. 이러한 방송용 어플리케이션의 최대 관심사는 실시간 전송이다. 실시간 전송을 하기 위하여 우리는 UDP(User Datagram Protocol)을 사용한다. 그러나 이러한 UDP는 실시간성은 보장되어지나, 데이터 손실의 문제와 전송되어지는 순서가 뒤바뀔 수 있는 문제가 있다.[3] 그래서 출현한 프로토콜이 RTP(Real-time Transport Protocol)이다.

이러한 RTP는 전송에 있어서 패킷의 크기 때문에 생기는 오류와 망의 상태에 따라 요구되어지는 QoS를 제어할 수 없다. 이러한 흐름 제어를 하기위하여 RTCP(RTP Control Protocol)라는 프로토콜이 제안되었다.

인터넷 환경에서의 비디오 데이터의 전송은 슬라이스 단위의 패킷 전송을 통해 패킷 손실률을 낮추도록 해야한다.[5]

실제적으로 이러한 프로토콜들을 응용프로그램에 적용시키기 위하여 소켓이라는 객체를 사용한다. 현재 API(Application Programming Interface)에서는 winsock version 1.1에 이어 version 2.0이 개발되었다. 2.0버전에서는 기존의 IP 기반의 프로토콜 이외에도 새로운 프로토콜들을 적용시킬 수 있으며, 그러한 연구도 활발히 진행되어지고 있다.

본 논문에서는 winsock 2.0을 사용하여 시뮬레이션 하였고, 소켓의 수에 따라 패킷의 크기를 조절함으로써 전송 효율이 나아짐을 보인다.

### 2. 본문

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

RTP는 멀티캐스트 또는 유니캐스트상에서 음성, 화상, 또는 모의 데이터와 같은 실시간 데이터를 전송하는 응용에 적합한 단대 단(end-to-end) 트랜스포트 기능을 제공한다. 그러나 RTP는 자원 예약에 대한 내용은 다루지는 않으며, 특히 적시 데이터 전송(timely

delivery), QoS 보장, 뒤바뀐 순서의 전송 방식과 같은 기능을 제공하지 않는다. 따라서 트랜스포트의 의미는 실시간 데이터의 특성에 중점을 두어 제정한 표준이라고 할 수 있다. RTP패킷은 UDP를 이용하여 전달된다. [1]

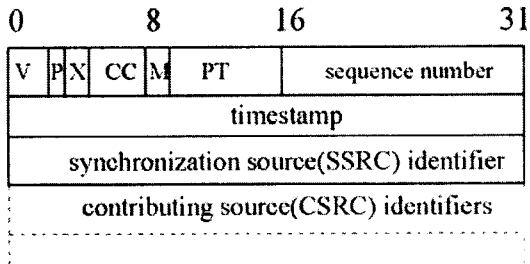


그림 1 RTP header format

V : version 필드이며 최근 버전은 2.0

P : 32비트 단위로 패킷을 구성하기 위해서 사용. P값이 세팅되면 payload 부분이 아닌 패딩 옥테트들이 패킷의 끝에 포함됨을 의미

X : 세팅되면 정확하게 한 개의 확장 헤더가 고정 헤더 다음에 온다

CC : 고정 헤더에서 CSRC identifier의 개수

M : 멀티미디어 정보에 대한 프레임 영역을 나타낸다. 즉 패킷 안에서 음성과 화상 정보를 구별하는데 사용

PT : RFC 1890에서 정의된 프로파일의 RTP payload 양식

sequence number : RTP 패킷이 송신될 때 마다 1씩 증가한다. 수신측은 이 필드를 이용하여 패킷분실을 감지 하고 패킷 순서를 재저장한다

timestamp : RTP 패킷의 첫 번째 옥테트가 샘플링된 시점을 나타낸다. 그 샘플링 시점은 일정하게 증가하는 클럭으로부터 생성된다. 이것은 실시간 데이터의 동기화와 지터 계산에 이용된다

SSRC(synchronization source) : 카메라 또는 마이크 등의 데이터 원천지의 식별자

CSRC(contributing source) : RTP 패킷이 중간 시스템에서 혼합된 경우에 그 소스들을 구별할 수 있는 식별자들

## 2.2 RTCP(RTP Control Protocol)

RTCP는 회의 참여간에 분실된 패킷 수, 지터 간격, 앞의 패킷과의 지연시간 등의 QoS 정보를 교환하여 응용이 적당한 QoS를 평가하여 adaptive encoding을 제공하도록 한다. 또한 RTCP는 많은 참여자들의 스케일을 위해서 패킷 송신율을 계산하고 사용자 인터페이스의 참여자 ID를 지칭하는 최소한의 세션 제어 정보를 나른다. 또한 데이터의 원천지 식별자가 충돌이 되거나 다시 만들 경우에 변경되어야 하므로 CNAME(Canonical NAME)이라 부르는 영구 트랜스포트 식별자를 나른다.

RTCP는 제어 패킷을 주기적으로 모든 참여자에게 전송한다. 이를 수행하기 위한 RTCP 메시지들은 다음과 같다.

SR (sender report) : active sender들은 자신의 송수신에 대한 통계 정보를 알리는 데 사용한다.

RR (receiver report) : active sender가 아닌 참여자들이 자신의 송수신에 대한 통계 정보를 알리는 데 사용한다.

SDES (Source Description) : CNAME을 포함하여 소스 이름을 기술하는 데 사용한다.

BYE (BYE) : RTP session을 빠져나올 때 사용한다.

APP (Application) : 새로운 응용 또는 새로운 기능을 시험할 때 그 응용에 한정된 기능을 지정하는데 사용한다.

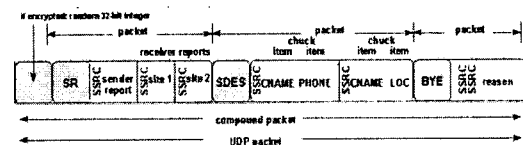


그림 2 RTCP 복합 패킷 양식

RTCP 패킷은 고정된 헤더 필드와 패킷 타입에 따라서 가변의 길이를 갖는 필드가 붙는다. 효과적인 정보 전송을 위해서 여러 RTCP 패킷들이 한 개의 UDP 패킷으로 전송될 수 있다. [RFC 1889]

## 2.3 RTP와 RTCP의 관계

RTP 데이터는 짝수 UDP 포트 번호에 실려

가고 대응되는 RTCP 패킷들은 그 바로 다음 포트 번호에 실려간다. [2]

## 2.4 RTP 소켓의 개수

패킷의 크기가 작을수록 전송 오류가 날 확률은 그만큼 작다. 그렇다고 소켓의 개수를 무한대로 열수는 없는 일이다.

망의 현재 상황과 패킷의 크기, 그리고 전송되어지는 데이터의 형태 등을 고려하여 소켓의 개수를 정하여야 할 것이다.

## 2.5 MTU(Maximum Transfer Unit)와의 관계

IP 데이터그램이 IP층(네트워크층)과 데이터 링크층 사이에서 교환될 때, 데이터링크 프레임의 크기에는 통상 물리층의 제약 등에 의한 상한이 있어, IP 데이터그램의 크기가 최대의 전송단위(MTU)를 넘을 때에는 분해 및 조립이 필요하다. MTU의 크기는 예를 들어 Ethernet과의 사이의 IP데이터그램에서는 1500바이트(Ethernet의 데이터부)이다. [통신프로토콜] 이러한 이유 때문에 패킷의 크기를 가급적 MTU 이내의 범위로 만드는 것이 시간적인 측면에서 볼 때 유리하다.

## 2.6 소켓의 수를 너무 많이 책정할 경우의 문제점

LAN(Local Area Network)의 특성상 한 스레드타임에 하나의 패킷만이 전송된다. 그래서 한 스레드 타임에 2개이상의 패킷을 전송할 수 없다.

멀티미디어 데이터의 전송에서는 이러한 시간의 간격을 무조건 길게 잡을 수는 없다. 왜냐하면 비디오나 오디오의 경우 샘플링 주기가 있고, 그 주기에 알맞은 전송 주기를 선택해야 하기 때문이다.

실제로 실시간 서비스가 가능 하도록 하기 위해서는 오디오의 경우 150ms 이내의 지연을 허용한다[6]

실제 문제되는 코딩법

```
while(전송할 데이터의 끝이 아니면){
    // 패킷을 만든다.
    packetize(전송할 데이터);
    send_cnt = sendto(소켓, ....);
}
```

보완된 코딩법

```
...
SetTimer(0, m_시간간격, NULL);
...
}
void OnTimer(UINT nIDEvent){
    // 패킷을 만든다.
    packetize(전송할 데이터);
    send_cnt = sendto(소켓, ....);
    if(전송할 데이터의 끝이면)
        KillTimer(nIDEvent);
}
```

## 3 실험 환경 및 결과

### 3.1 실험 환경

멀티미디어 데이터의 전송을 목적으로 하기 때문에 CIF(352\*288) 100프레임 짜리 MPEG-4로 인코드한 데이터를 사용하였으며, 전송단은 서울 여의도에서 수원 경희대로 하였다.

### 3.2 실험 결과

#### 3.2.1 실시간성을 고려한 전송 주기

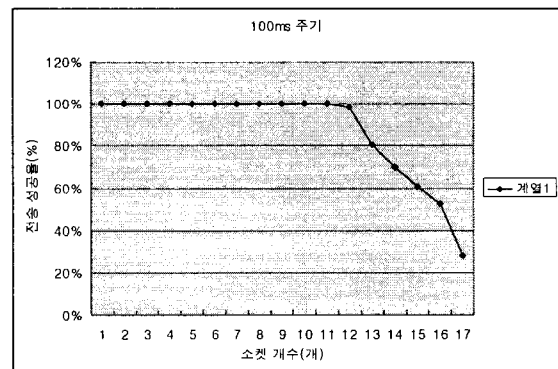


그림 5 100ms 주기 마다 전송을 시도한 경우 (시스템 사양 : Pentium-III 450MHz) 위의 결과와 같이 100ms마다 전송을 시도

할 경우 소켓의 개수가 11개가 넘어가면 전송이 되지 않는 경우가 생기기 시작한다.

위와 같은 결과는 단말기의 랜카드와 거쳐가는 경로의 망의 상황에 의존적이다.

### 3.2.2 소켓수에 따른 전송효율

#### 3.2.2.1 망의 상태가 좋은 경우

오후 7시 20분에 서울에서 수원까지 전송

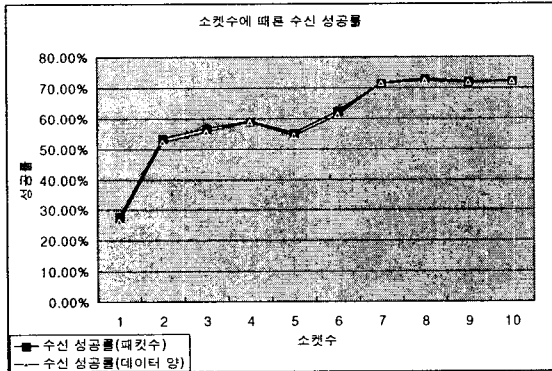


그림 6 소켓수에 따른 수신 성공률  
소켓의 개수가 7개에서 10개 사이가 수신율이 가장 높음을 알 수 있다.

#### 3.2.2.2 망의 상태가 좋지 않을 경우

오전 11시 45분에 서울에서 수원까지 전송

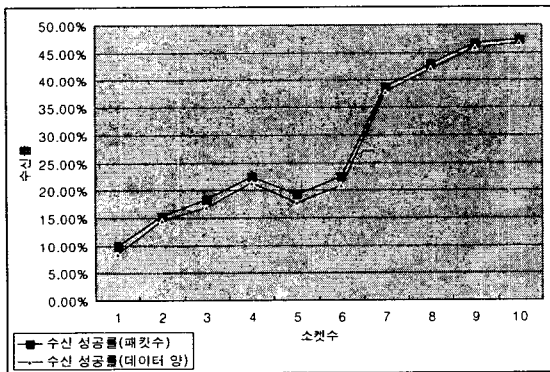


그림 7 소켓수에 따른 수신 성공률  
소켓의 개수가 7개부터 좋은 효과를 볼 수 있었다.

### 3. 결론 및 향후계획

비디오 데이터의 전송에 있어서 하나의 패킷 크기를 절정 하는 것은 현재 망의 상태와 랜의 MTU, 그리고 전송주기에 따라 적절한

크기를 결정해야 함을 알 수 있었다. 이러한 인자들을 이용하여 현재 상황에 맞는 소켓의 수를 결정하여 데이터의 손실을 최소화 할 수 있다.

비디오 데이터 전송의 최적화를 위하여 하나의 프레임을 여러개로 잘라서 보내는 경우에 있어 하나의 긴 데이터를 잃어버리지 않을 경우와 여러 개의 짧은 데이터 모두를 잃어버리지 않을 경우에 대한 분석도 필요하다. 향후 이러한 분석과 전송 시간에 관련된 분석이 수행될 것이다.

#### 참고 문헌

- [1] RFC 1889, RTP - A Transport Protocol for Real-time Applications. Audio-Video Transport Working Group, H. Schulzrinne, S. Casner, R. Frederick, V. Jacobson. January 1996.
- [2] RFC 1890, RTP Profile for Audio and Video Conferences with Minimal Control. Audio-Video Transport Working Group, H. Schulzrinne. January 1996.
- [3] RFC 768 UDP, User Datagram Protocol, IETF, Aug. 1980, <http://ds.internic.net/rfc/rfc768.txt>
- [4] 일본 멀티미디어 통신연구회, "통신 프로토콜", 정보통신 연구원, p.163-179
- [5] 석주명, "QoS를 고려한 인터넷 기반의 MPEG-4 비디오폰 및 VOD 구현", 1999년 2월 석사 학위 논문, 경희대학교
- [6] ITU-T Rec. G.711 - PULSE CODE MODULATION (PCM) OF VOICE FREQUENCIES