

IP 망을 통한 MPEG-2 TS 기반의 프로그레시브 스트리밍을 위한 de-jitter 버퍼링 시간 추정 기법

서광덕^{a)‡}, 김현정^{a)}, 김진수^{b)}, 정순홍^{c)}, 유정주^{c)}, 정영호^{c)}

Estimation of De-jitter Buffering Time for MPEG-2 TS Based Progressive Streaming over IP Networks

Kwang-deok Seo^{a)‡}, Hyunjung Kim^{a)}, Jin-soo Kim^{b)}, Soon-heung Jung^{c)}, JeongJu Yoo^{c)}, and Youngho Jeong^{c)}

요 약

본 논문에서는 MPEG-2 TS (transport stream)를 포함하는 TCP 패킷들을 유무선 인터넷 환경에서 프로그레시브 스트리밍 서비스를 통해 클라이언트로 전달할 때 발생하게 되는 네트워크 jitter 를 추정하는 기법을 제안한다. 추정된 네트워크 jitter의 크기를 바탕으로 네트워크 jitter를 흡수할 수 있는 수신측에서 필요한 de-jitter 버퍼링 시간을 계산할 수 있다. 이를 위해서 TS 패킷에 기록되어 있는 PCR (program clock reference) 정보를 활용하여 TCP 패킷 헤더의 optional 필드에 jitter 크기 추정에 필요한 새로운 타임스탬프 정보를 생성한다. 제안된 de-jitter 버퍼링 기법을 IP 망을 통한 프로그레시브 스트리밍 서비스에 활용할 경우 기존의 T-STD 버퍼 모델의 동작 원리를 그대로 따를 수 있게 되어 기존의 T-STD 버퍼 모델을 수정 없이 활용할 수 있다. 제안된 기법은 최근에 국제표준으로 개발된 MPEG DASH (dynamic adaptive streaming over HTTP) 기술에 응용될 수 있다.

Abstract

In this paper, we propose an estimation of network jitter that occurs when transmitting TCP packets containing MPEG-2 TS in progressive streaming service over wired or wireless Internet networks. Based on the estimated network jitter size, we can calculate required de-jitter buffering time to absorb the network jitter at the receiver side. For this purpose, by exploiting the PCR timestamp existing in the TS packet header, we create a new timestamp information that is marked in the optional field of TCP packet header to estimate the network jitter. By using the proposed de-jitter buffering scheme, it is possible to employ the conventional T-STD buffer model without any modification in the progressive streaming service over IP networks. The proposed method can be applicable to the recently developed international standard, MPEG DASH (dynamic adaptive streaming over HTTP) technology.

Keyword: de-jitter buffering, MPEG DASH, progressive streaming, timestamp, MPEG-2 TS

- a) 연세대학교
Yonsei University
- b) 한밭대학교
Hanbat National University
- c) 한국전자통신연구원
ETRI

‡ 교신저자 : 서광덕 (kdseo@yonsei.ac.kr)

※ 본 연구는 방송통신위원회의 ETRI 연구개발지원사업의 일환으로 수행되었습니다. [11921-03001, "Beyond 스마트TV 기술개발"]
· 접수일(2011년6월28일), 수정일(2011년8월25일), 게재확정일(2011년8월25일)

1. 서론

프로그레시브 스트리밍 서비스는 전송 프로토콜로써 TCP (transmission control protocol)^[1]를 이용한다는 측면에서 다운로드와 유사하지만, 다운로드 되는 도중에 수신된 미디어 데이터를 실시간으로 재생한다는 측면에서는 실시간 스트리밍과 유사한 개념의 서비스 기술이다^[2]. 결국 다운로드와 스트리밍 두 가지 기술의 장점만을 활용하는 기술로 요약이 될 수 있다^[3]. 기존의 다운로드 방식의 문제점인 서비스 지연 시간을 줄이기 위해 수신 측에 도착한 TCP 패킷이 일정량 만큼 버퍼에 찼을 때부터 실시간으로 도착한 데이터를 재생하여 사용자에게 디스플레이 한다. 이렇게 함으로써 서비스 지연시간을 단축하고 QoS를 보장해 줄 수 있는 스트리밍 서비스가 가능하다.

기존의 프로그레시브 스트리밍 서비스에서 전송되는 콘텐츠 파일은 압축된 비디오 및 오디오를 하나의 파일로 저장하기 위해서 QuickTime (.mov), MPEG-4 (.mp4), Windows Media Video (.wmv), Flash Video (.flv), 3GPP (.3gp) 등의 파일포맷 규격에 의해 제작 되었다. 이러한 파일 포맷 규격들은 ISO/IEC MPEG에서 개발한 ISO base media file format^[4]을 기반으로 확장되었다. 이러한 다양한 파일포맷 규격들을 바탕으로 제작된 멀티미디어 콘텐츠 파일은 프로그레시브 스트리밍 서비스에 의해 유무선 IP (Internet Protocol) 망을 통해 광범위하게 제작, 배포 및 유통되어 왔다^[5].

한편, 최근에 방송과 통신의 융합 (convergence) 에 의해 방송 서비스를 IP망을 통해 제공하는 인터넷 TV, IPTV 서

비스 등이 확대되고 있다. 이로 인해 방송용 콘텐츠를 프로그레시브 스트리밍 서비스에 의해 제공하기 위해서 MPEG-2 TS (transport stream)^[6] 규격으로 제작된 방송용 콘텐츠를 기존의 mov, mp4, wmv, flv, 3gp 등의 파일포맷 대신에 활용하여 프로그레시브 스트리밍을 제공하는 새로운 유형의 서비스가 등장할 것으로 예상된다. 그림 1은 MPEG-2 TS 규격에 의해 제작된 멀티미디어 콘텐츠를 HTTP 스트리밍에 의해 제공하는 서비스의 개요를 나타낸다. MPEG-2 TS 방송용 콘텐츠는 다양한 멀티미디어 인코더로 압축된 비디오/오디오 압축 데이터를 MPEG-2 TS 규격으로 제작하여 프로그레시브 스트리밍을 통해 서비스할 수도 있고, 또한 기존의 파일포맷 규격으로 제작된 콘텐츠 파일을 MPEG-2 TS 로 변환하여 얻어진 MPEG-2 TS 콘텐츠를 활용하여 프로그레시브 스트리밍 서비스에 활용할 수 있다. 클라이언트는 서버에게 단순히 HTTP GET Request 를 통해 원하는 MPEG-2 TS 콘텐츠를 요청한다^{[7][8]}.

본 논문에서는 그림 1과 같은 MPEG-2 TS 콘텐츠를 활용하는 프로그레시브 스트리밍 서비스에서 MPEG-2 TS 콘텐츠를 IP 망을 통해 전송할 때 MPEG-2 TS 패킷의 헤더에 포함되어 있는 PCR (Program Clock Reference) 클럭 (clock) 정보를 활용하여 클라이언트에 도착하는 패킷의 도착 시간 지연 변이 (delay variation)인 네트워크 지터 (jitter) 를 효과적으로 추정하는 방법을 제안한다. 이를 바탕으로 패킷들 간의 도착 간격 지연 변이 (delay variation in arrival time)인 inter-arrival jitter 를 추정할 수 있다. de-jitter 버퍼링은 첫번째 패킷이 목적지에 도달하면 바로 재생과정을 시작하지 않고 뒤에 도달하게 되는 패킷들 과의 도착 시간

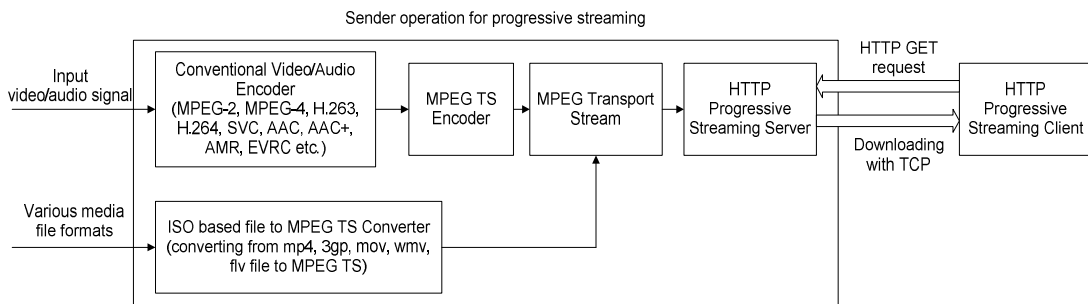


그림 1. MPEG-2 TS 기반의 프로그레시브 스트리밍 서비스 개요
 Fig. 1. Service concept of MPEG-2 TS based progressive streaming

에 있어서의 차이를 흡수하여 원래의 패킷 간의 시간 간격을 유지할 수 있도록 어느 정도 패킷들을 버퍼에 모아둔 후에 재생을 시작할 수 있게 한다. 이 de-jitter 버퍼링 시간의 크기를 너무 작게 설정하면 네트워크 지터들을 흡수하기 위해 필요한 충분한 시간적 여유가 부족해져서 재생 후이른 시간 내에 버퍼 언더플로우 (underflow)의 문제가 발생할 가능성이 크며, 이는 스트리밍 도중에 수신 측에서 잦은 리버퍼링 (rebuffering)을 유발하게 되어 재생의 중단을 초래하여 서비스 품질을 심각하게 저하시킨다. 반대로, de-jitter 버퍼링 시간의 크기를 너무 크게 설정하면 de-jitter 버퍼링 직후에 시작되는 초기 재생 (initial playout) 과정이 리버퍼링의 크기에 비례하여 늦어지기 때문에, 사용자의 서비스에 대한 latency를 증가시키게 된다. 본 논문에서는 MPEG-2 TS 기반의 프로그레시브 스트리밍 서비스에서 TCP 패킷의 도착 간격 지터 추정을 통해 de-jitter 버퍼링 시간의 적합한 크기 및 de-jitter 버퍼링의 크기에 해당하는 초기 재생 지연 시간 (initial playout delay)을 결정하는 방법을 제안한다. 제안된 기법은 최근에 ISO MPEG에서 표준화를 추진 중인 DASH (dynamic adaptive streaming over HTTP) 서비스에 응용될 수 있다^[9].

II. 관련된 기존 기술 개요

MPEG-2 TS 패킷은 188바이트 크기의 고정된 길이의 패

킷이며 MPEG-2 TS 패킷의 연속적인 스트림을 MPEG Transport Stream (TS) 이라고 부른다. MPEG-2 TS는 통상적으로 디지털방송 서비스를 목적으로 개발된 규격이기 때문에 전송되는 TS 패킷들은 채널 품질이 비교적 안정적인 서킷 스위치 망 (Circuit switched network)인 방송망을 통해 수신기에 전달된다. 따라서, MPEG-2 TS 패킷들은 전송 채널에서 경험하게 되는 패킷 지연 시간이 비교적 짧고 일정하며 수신기에 도착된 TS 패킷들을 순차적으로 처리하기 위한 타이밍 버퍼 모델 (timing buffer model) 도 효과적으로 적용될 수 있다. MPEG-2 TS를 처리하기 위한 수신측에서의 타이밍 버퍼 모델을 MPEG 시스템 표준에서는 T-STD (Transport Stream System Target Decoder) 버퍼 모델이라고 부른다^[5]. 그림 2는 MPEG-2 TS로 다중화되어 수신되는 비디오, 오디오 및 시스템정보 관련 TS 패킷들을 역다중화하여 해당되는 스트림 별로 순차적으로 거치게 되는 버퍼들을 나타내는 표준 T-STD 버퍼 모델의 구조를 보이고 있다.

III. 제안된 de-jitter 버퍼링 시간 추정 기법

1. IP 망에서의 네트워크 jitter 현상

MPEG-2 TS는 방송망 또는 ATM (Asynchronous Transfer Mode) 망과 같은 비교적 대역폭 품질이 높고 전달 지연

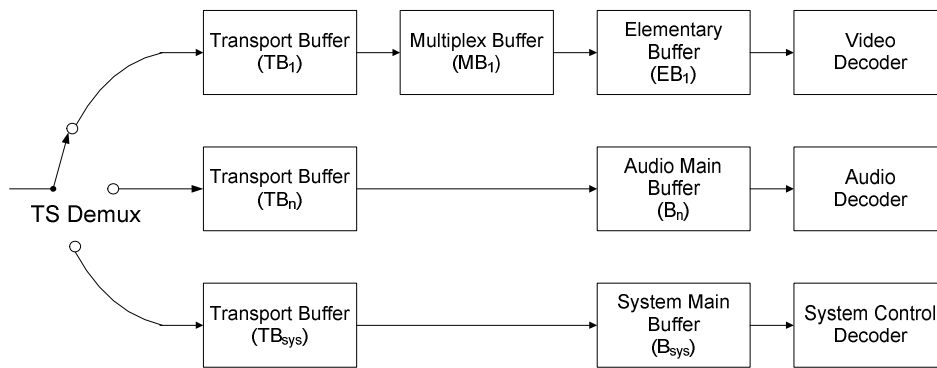


그림 2. 표준 T-STD 버퍼 모델
Fig. 2. Standard T-STD buffer model

이 낮은 서킷 스위치 망 (circuit switched network)을 통해 멀티미디어 데이터를 전달하는 것을 목표로 규격이 개발되었다. 따라서, IP 망과 같은 패킷 스위치 망 (packet switched network)인 인터넷을 통해 MPEG-2 TS를 전송하기 위해서는 인터넷의 고질적인 문제점인 패킷들 간의 도착 지연 시간 변이 (delay variation in arrival time) 를 정확하게 추정할 필요가 있고, 방송망을 통한 MPEG-2 TS 전송에 적합하게 설계된 그림 3에 나타난 표준 T-STD 모델도 이를 바탕으로 수정될 필요가 있다. 인터넷 망의 경우 전달되는 패킷들이 거치게 되는 라우팅 경로가 서로 다를 수 있고 각 라우터마다 처리해야 할 트래픽의 량과 처리 시간이 다르기 때문에 패킷들의 목적지 도착 시간이 서로 차이가 날 수 있다^[10]. 이러한 시간적 갭 (playout gap)을 제거해야만 T-STD 버퍼 모델 기반의 디코더가 버퍼 오버플로우 (overflow) 및 버퍼 언더플로우 (underflow)를 겪지 않고 원활하게 동작하게 된다. 따라서, 도착한 패킷들을 디코더로 전달하기 이전에 이러한 시간적 갭을 흡수하기 위해 de-jitter 버퍼링과 같은 별도의 버퍼링 과정을 통한 재생 시간의 지연 (playout delay)이 필요하다. 이때의 버퍼링 과정에 필요한 적합한 크기를 결정하기 위해서는 패킷들 간의 도착 시간 지연 변이를 정확하게 추정할 수 있는 방법이 필요하다. de-jitter 버퍼링의 크기가 너무 작으면 버퍼 언더플로우의 위험이 증가하여 잦은 리버퍼링이 발생하게 되고, 반대로 너무 크면 도착한 데이터에 대한 버퍼링 시간이 길어지는 관계로 서비스의 지연 (latency)이 발생하게 되어 서비스 품질이 저하된다. 이러한 문제를 해결하기 위해서 본 논문

에서는 TCP 패킷에 실리게 되는 MPEG-2 TS 패킷 헤더에 존재하는 PCR (program clock reference) 값을 활용하여 수신 측에서 패킷들 간의 도착 지연 시간의 변이인 inter-arrival jitter 를 효과적으로 추정할 수 있는 방법을 제안한다.

2. jitter 계산을 위한 타임스탬프 생성 기법

MPEG-2 TS 패킷은 188바이트 크기의 고정된 크기이므로 여러 개의 TS 패킷을 하나의 TCP 패킷의 페이로드에 실을 수 있다. 인터넷의 MTU (maximum transfer unit) 크기가 1500 바이트임을 감안할 때 이론적으로 최대 7개의 TS 패킷들을 TCP의 페이로드에 실을 수 있다. 그림 3은 n 개의 TS 패킷들을 하나의 TCP 패킷의 페이로드에 사상 (mapping)하는 TCP 페이로드 포맷을 나타낸다.

전송단과 수신단에서 사용되는 시스템 타임 클럭 (system time clock : STC) 간의 동기화를 서로 맞추기 위한 시간 정보의 전달을 위해서 TS 패킷의 헤더에는 PCR 클럭 정보가 존재한다. PCR 값은 시스템 인코더에서의 시간을 27 MHz (SCIF: system_clock_frequency) 의 시스템 클럭 주파수로 샘플링한 값이다. PCR 클럭은 총 42 비트로 구성되는데, 90 KHz (SCF/300) 단위로 표현되는 33 비트 길이의 PCR_base와 27 MHz 단위의 SCF로 표현되는 9비트의 PCR_ext로 구성된다. PCR은 전송단과 수신단 간의 STC동기화를 지속적으로 유지하기 위해 통상적으로 100 ms 이내에 주기적으로 전송이 되도록 규정하고 있다. 따라서 그림 3과

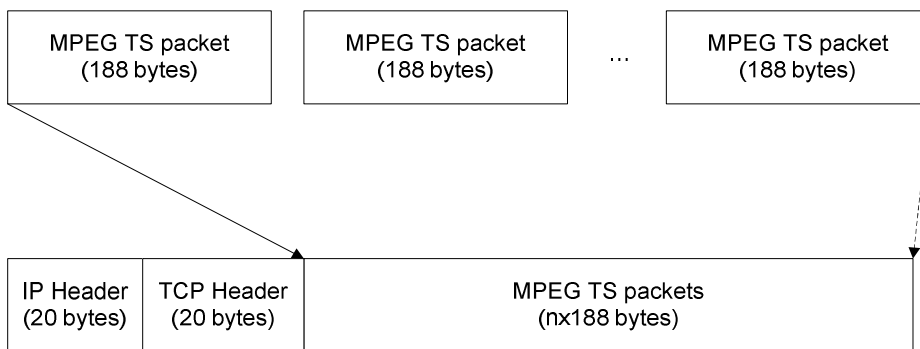


그림 3. n 개의 MPEG-2 TS 패킷들로 구성되는 TCP 페이로드 포맷
Fig. 3. TCP payload format consisting of n MPEG-2 TS packets

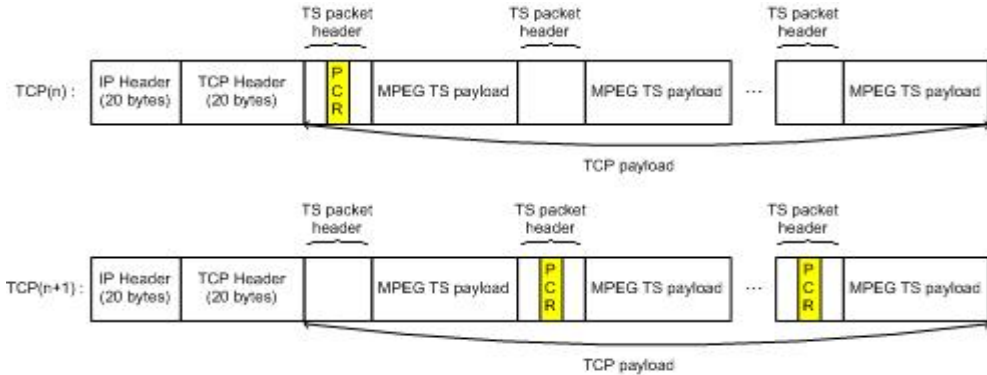
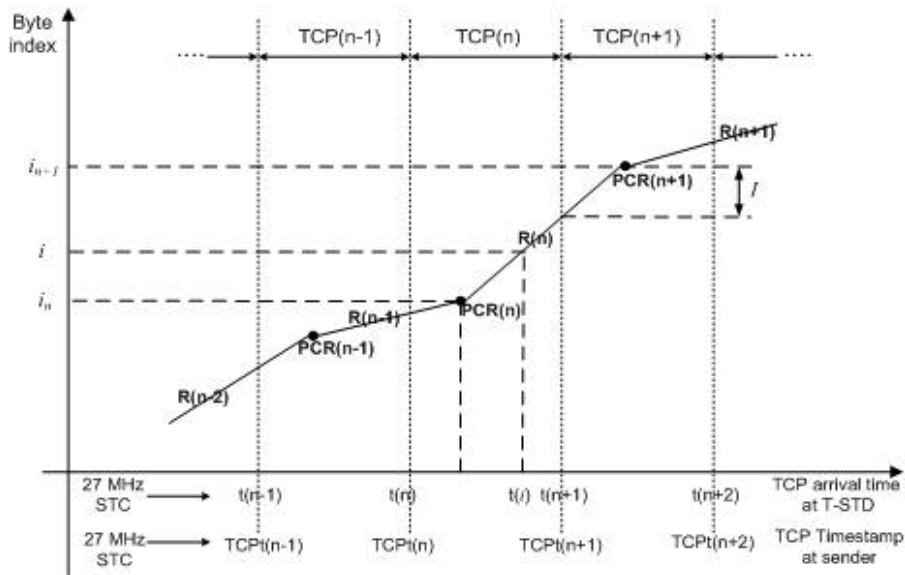


그림 4. PCR 정보를 포함하는 MPEG-2 TS 패킷들을 포함하는 TCP 패킷
 Fig. 4. TCP packet containing MPEG-2 TS packets including PCR information

같이 TCP 패킷의 페이로드 포맷을 구성하는 TS 패킷들 가운데 PCR 정보를 헤더에 포함하는 TS 패킷이 존재할 수 있다. 그림 4는 PCR 정보를 헤더에 포함하고 있는 TS 패킷들을 TCP 페이로드에 담고 있는 n 번째 TCP 패킷인 TCP(n)과 (n+1)번째 패킷인 TCP(n+1)의 페이로드 포맷을 각각 나타낸다. TCP(n)과 TCP(n+1)은 헤더에 PCR 정보를

포함하는 TS 패킷을 하나 이상 TCP 페이로드에 포함하면서 전송 순서가 순차적으로 이웃하는 두 개의 TCP 패킷들을 나타낸다.

본 논문에서는 수신 측에 도착한 TCP 패킷의 도착 시간 지연 변이를 추정하기 위하여 TCP 페이로드에 포함되어 있는 PCR 정보를 활용하는 방법을 제안한다. 그림 5는



- n : the index of the TCP i.e. the value n in TCP(n)
- PCR(n) : the index of the first PCR in TCP(n)
- I : the number of bytes preceding the byte that contains the last bit of PCR(n+1)
- R(n) : transport rate of the transport stream between PCR(n) and PCR(n+1)

그림 5. TCP 패킷의 도착 시간을 송신 측에서 예측하기 위한 타이밍 모델
 Fig. 5. Timing model to predict arrival time of TCP packets at the sender side

TCP 패킷이 수신 측에 도착하는 시간을 송신 측에서 예측하기 위한 타이밍 모델을 나타낸다.

그림 5에서 n번째 TCP 패킷인 TCP(n) (n=0, 1, 2, 3, ...)은 그림 4과 같은 포맷으로 여러 개의 연속적인 TS 패킷들을 페이로드에 포함하고 있고, PCR 정보를 포함하는 첫 번째 TS 패킷의 PCR 값을 TCP(n)을 대표하는 PCR 값인 PCR(n)으로 활용하게 된다. 그림 5의 가로축 (horizontal axis)은 수신 측에 도착하게 되는 이론적인 시간과 이에 해당하는 TCP 타임스탬프 값을 나타내고, 세로축 (vertical axis)은 전송되는 연속적인 데이터 바이트의 인덱스 값을 나타낸다. 만약 i번째 데이터 바이트가 수신측에 도착하게 되는 시간을 추정한다면 다음과 같이 계산할 수 있다.

$$t(i) = \frac{PCR(n)}{27MHz} + \frac{i - i_n}{R(n)} \quad (i_n < i < i_{n+1}) \quad (1)$$

식 (1)에서 R(n)은 PCR(n)과 PCR(n+1) 클럭 사이의 데이터 전송률 (transport rate)을 의미하며 다음과 같이 계산된다.

$$R(n) = \frac{(i_{n+1} - i_n)}{(PCR(n+1) - PCR(n)) / 27MHz} \quad (2)$$

한편, TCP(n+1)의 페이로드의 첫 번째 바이트가 수신 측에 도착하게 되는 시간을 추정한다면 다음과 같이 계산할 수 있다.

$$t(n+1) = \frac{PCR(n+1)}{27MHz} - \frac{I}{R(n)} \quad (3)$$

식 (3)에서 I는 TCP(n+1)에 포함되는 모든 데이터 중에서 PCR(n+1)의 정보를 나타내는 마지막 비트를 포함하는

바이트보다 앞서 위치하여 전송되는 모든 데이터의 크기를 바이트 단위로 나타낸 것이다.

그림 5에서 t(n+1)에 해당하는 시간을 27MHz SCF를 활용한 STC의 클럭 카운트 값으로 표현한 것이 TCP 타임스탬프인 TCPT(n+1)이다. TCP 타임스탬프는 TCP 패킷 스트림에서 스트리밍 데이터에 대한 시간 기준값을 나타내며, 이 기준값은 PCR이 유도되는 동일한 STC로부터 계산된다. 따라서, PCR과 마찬가지로 90KHz (SCF/300) 단위로 표현되는 기본 (base) 클럭과 27MHz 단위의 SCF에 의해 표현되는 확장 (extension) 클럭으로 나뉘어 진다. 식 (4)는 TCPT(n+1)에 대한 기본클럭, 확장클럭 그리고 이 두 가지 클럭에 의해 계산되는 27MHz 정밀도의 TCP 타임스탬프 값을 나타낸다.

식 (4)로부터 TCPT_base(n+1)은 총 33비트의 길이로 표현되고 TCPT_ext(n+1)은 총 9비트의 길이로 표현되며, 이 둘을 합한 TCPT(n+1)은 PCR과 마찬가지로 총 42비트의 길이를 갖는다. TCPT(n+1)이 최대값에 도달한 경우에는 다시 0으로 리셋 (reset) 되는 modulo의 연산이 적용된다.

90 KHz 단위의 클럭으로 시간을 표현했을 때 발생하는 최대의 오차가 $\frac{1}{90KHz} \cong 11\mu s$ 이기 때문에 응용에서 요구되는 도착 시간 지연 변이의 정밀도가 11μs를 초과하여도 무방할 경우에는 TCPT_base(n+1)만을 사용하게 되며, 이로 인해 발생 가능한 최대 11μs의 오차는 도착 시간 지연 변이 자체에 포함되게 된다. 반대로 도착 시간 지연 변이의 정밀도가 11μs의 오차를 허용할 수 없는 응용에서는 42비트의 완전한 길이 (full length)의 TCPT(n+1)을 사용한다.

식 (4)를 통해 얻어진 TCP(n+1) 패킷의 이론적인 수신 측의 도착 시간에 해당하는 TCP 타임스탬프 값인 TCPT(n+1)은 TCP 패킷의 확장 필드에 저장된다. 그림 6은 TCP 패킷 헤더의 기본적인 구조와 선택적으로 활용할 수 있는 확장 필드의 구조를 나타낸다^[1]. TCP 패킷의 헤더는 기본적으로 20 바이트의 크기로 이루어져 있고 추가적인

$$\begin{aligned} TCPT_base(n+1) &= ((system_clock_freq \times t(n+1)) DIV 300) \% 2^{33} \\ TCPT_ext(n+1) &= ((system_clock_freq \times t(n+1)) DIV 1) \% 300 \\ TCPT(n+1) &= TCPT_base(n+1) \times 300 + TCPT_ext(n+1) \end{aligned} \quad (4)$$

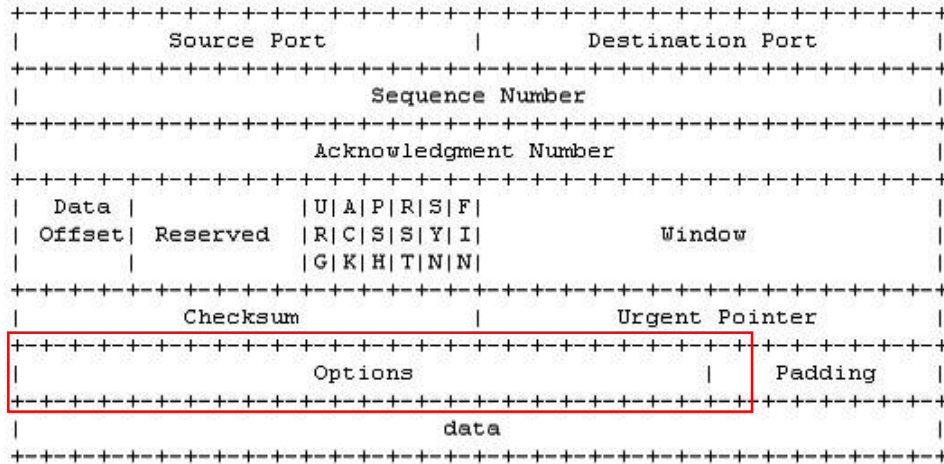


그림 6. TCP 패킷의 구조
Fig. 6. TCP packet structure

정보의 포함을 위해서 Options 필드의 활용을 통해 최대 32 바이트까지 헤더 정보를 추가할 수 있다. Options 필드의 추가 여부 및 크기는 Data offset (Header length를 나타냄) 값을 통해 지정할 수 있다. 본 논문에서는 식 (4) 를 통해 계산된 TCP 타임스탬프 값을 그림 6의 Options 필드에 추가하게 된다.

TCP 패킷의 Options 필드는 크게 Kind, Length, Data 등의 3가지 필드로 구성된다. 1바이트 크기의 Kind 필드는 용도를 나타내고, 1바이트 크기의 Length 필드는 현재의 Kind 필드에 해당하는 Options 필드의 전체 크기를 나타내며, Data 필드는 해당되는 Kind의 목적을 위해 필요한 실제 데이터를 포함하게 된다. 한편, TCP 페이로드에 포함되는 모든 TS 패킷들의 헤더에 PCR 정보가 기록되어 있지 않는 TCP 패킷에 대해서는 TCP 타임스탬프 계산 및 Options 필드 기록을 생략하게 된다.

3. de-jitter 버퍼링 시간 추정 기법

3.2 절에서 제시된 TCP 패킷에 실려서 전송되는 TCP 타임스탬프 값을 활용하여 수신 측에서는 이론적인 수신 측 도착 시간에 해당하는 TCP 타임스탬프 값과 실제 T-STD 도착 순간의 수신측의 STC 클럭의 카운트값 사이의 차이를 통해 IP 망을 통해 TCP 패킷이 전송되면서 겪게 되는

도착 시간의 지연 변이를 추정할 수 있다. 즉, (n+1)번째 TCP 패킷이 경험하는 네트워크 지터인 $N(n+1)$ 는 다음 식과 같이 표현된다.

$$N(n+1) = \frac{TCPt(n+1) - TCPa(n+1)}{27MHz} \quad (5)$$

식 (5)에서 TCPa(n+1)은 (n+1)번째 TCP 패킷이 수신 측에 도착하는 순간에서의 27MHz 기반의 수신측 STC 클럭의 카운트 값을 나타낸다. 한편, 90KHz 정밀도의 33비트 길이의 TCPT_base(n+1)의 타임스탬프 값을 TCP 헤더의 Options 필드에 기록한 경우에 계산되는 네트워크 지터인 $N_{base}(n+1)$ 는 다음 식과 같이 계산된다.

$$N_{base}(n+1) = \frac{TCPT_base(n+1) - TCPa_base(n+1)}{90kHz} \quad (6)$$

식 (6)에서 TCPa_base(n+1)은 90KHz 단위를 갖는 TCPa(n+1)의 base 클럭을 의미한다. 식 (6)에 의해 계산되는 네트워크 지터인 $N_{base}(n+1)$ 는 식 (5)의 결과인 $N(n+1)$ 과 비교하였을 때 다음과 같은 범위의 오차를 갖게 된다.

$$0 \leq N(n+1) - N_{base}(n+1) < \frac{1}{90kHz} (\cong 11\mu s) \quad (7)$$

따라서, 11 μ s 정도의 오차가 크게 중요하지 않은 응용 분야에서는 90 KHz 정밀도인 base 클럭만 사용해도 문제가 없다.

27 MHz 정밀도의 42 비트 길이의 TCP 타임스탬프를 적용할 경우 이웃하는 두 패킷인 TCP(n)과 TCP(n+1) 간의 도착 시간 간격 (inter packet spacing) $D(n+1)$ 은 다음 식 (8)과 같이 계산되어 진다.

한편, 90 KHz 정밀도의 33비트 길이의 TCPT_base(n+1)의 타임스탬프 값을 적용하는 경우 이웃하는 두 패킷 간의 도착 시간 간격 (inter packet spacing) $D_{base}(n+1)$ 은 다음 식 (9)와 같이 계산되어 진다.

식 (8)과 (9)에 의해 얻어진 두 패킷 간의 도착 시간 간격을 공통적으로 $D(n+1)$ 로 표시할 경우 연속적으로 도착하는 패킷들 간의 도착 간격 지연 변이인 inter-arrival jitter 는 이동평균 (moving average)에 의해 다음과 같이 계산된다.

$$J(n+1) = (1-\alpha) \cdot J(n) + \alpha \cdot |D(n+1)| \quad (10)$$

식 (10)에서 α ($0 < \alpha < 1$)는 스무딩 팩터 (smoothing factor)로서 최근에 새롭게 업데이트된 네트워크 지터의 반영 비율을 조절할 수 있다.

$J(n+1)$ 에 대한 분산 (variance)인 $v(n+1)$ 도 이동평균 (moving average)에 의해 다음과 같이 계산된다.

$$v(n+1) = (1-\beta) \cdot v(n) + \beta \left(|D(n+1)| - J(n) \right)^2 \quad (11)$$

식 (11)에서 β ($0 < \beta < 1$)는 스무딩 팩터 (smoothing factor)로서 최근에 새롭게 업데이트된 네트워크 지터와 기)의 평균적인 지터 사이의 변이에 대한 반응 속도를 조절할 수 있다.

식 (10)과 (11)의 결과를 바탕으로 $J(n+1)$ 을 효과적으로 흡수하기 위한 de-jitter 버퍼링 시간 $B(n+1)$ 은 다음과 같이 계산될 수 있다.

$$B(n+1) = J(n+1) + K \cdot v(n+1) \quad (12)$$

식 (12)에서 K 값은 de-jitter 버퍼링 시간의 결정에 있어서 평균적인 inter-arrival jitter 값을 벗어나는 변이인 분산 $v(n+1)$ 의 변동폭을 반영하기 위한 파라미터로써 K 값이 커지면 망에서 특수한 상황에서 발생 가능한 일반적이지 못한 큰 폭의 변동폭을 de-jitter 버퍼링에 의해 흡수할 수 있는 확률이 높아진다.

식 (12)를 통해 수신 측에 첫 번째로 도착하게 되는 패킷이 de-jitter 버퍼링을 통해 겪게 되는 초기 재생 지연 (initial playout delay) 시간이 결정되고, 이를 바탕으로 IP 망의 전송 도중에 발생한 네트워크 지터의 흡수를 고려한 프로그레시브 스트리밍 서비스의 초기 재생 시점이 결정 된다.

수신되는 TCP 패킷의 타임스탬프를 활용하여 적절한 de-jitter 버퍼링 시간을 추정하기 위해서는 어느 정도의 충분한 시간 동안 TCP 패킷을 수신해서 분석해야 하는데, 이 분석 시간은 TCP 패킷의 전송이 시작되어 안정적인 패킷 송수신 상태에 이르는데 충분한 시간인 대략 3~4초 정도 사이면 충분하다. 이 분석 시간을 T 라고 표시하면 식 (12)에 의해 얻어진 $B(n+1)$ 과 T의 대소관계를 고려하여 최종적

$$D(n+1) = \frac{1}{27MHz} \left\{ (TCPt(n+1) - TCPT(n)) - (TCPa(n+1) - TCPa(n)) \right\} \\ = \frac{1}{27MHz} (TCPt(n+1) - TCPa(n+1)) - \frac{1}{27MHz} (TCPT(n) - TCPa(n)) = N(n+1) - N(n) \quad (8)$$

$$D_{base}(n+1) = \frac{1}{90KHz} \left\{ (TCPT_base(n+1) - TCPT_base(n)) - (TCPa_base(n+1) - TCPa_base(n)) \right\} \\ = \frac{1}{90KHz} (TCPT_base(n+1) - TCPa_base(n+1)) - \frac{1}{90KHz} (TCPT_base(n) - TCPa_base(n)) \\ = N_{base}(n+1) - N_{base}(n) \quad (9)$$

인 de-jitter 버퍼링 시간 t_{dj} 를 다음과 같이 결정한다.

$$t_{dj} = \begin{cases} T, & \text{if } B(n+1) < T \\ B(n+1), & \text{otherwise.} \end{cases} \quad (13)$$

그림 7은 IP망을 통한 프로그레시브 스트리밍 서비스를 실시할 때 본 연구에 의해 도출된 inter-arrival jitter를 바탕으로 지터를 흡수하기 위한 de-jitter 버퍼링이 포함된 새로

운 T-STD 버퍼 모델을 나타낸다. 네트워크 지터를 흡수하기 위한 “IP network de-jitter buffering” 이 수신 측 파일 버퍼 (file buffer) 내부에서 이루어지고, de-jitter 버퍼링 후에 기존의 T-STD 버퍼 모델에 전달되는 TS 패킷들은 기존의 T-STD 버퍼 모델의 작동 원리를 그대로 따르게 된다.

그림 8은 파일 버퍼에 도착한 첫 번째 TCP 패킷이 본 연구에 의해 결정된 de-jitter 버퍼링 시간동안 파일 버퍼에 머무르면서 재생 시간이 지연 된 후에 설정된 de-jitter 버퍼

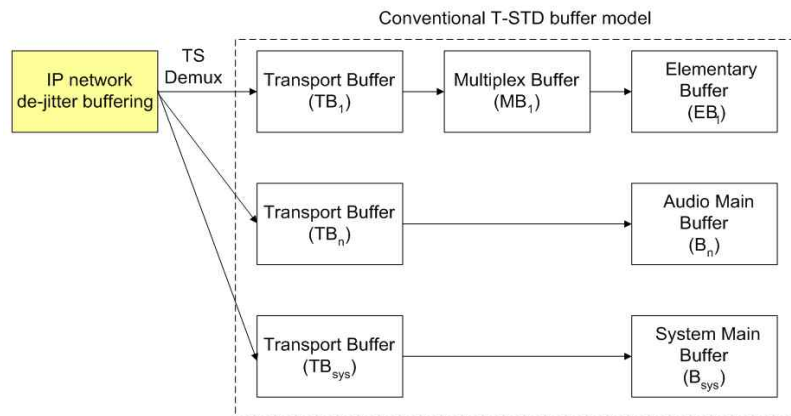


그림 7. 프로그레시브 스트리밍 서비스를 위해서 de-jitter 버퍼링이 포함된 T-STD 버퍼 모델
Fig. 7. T-STD buffer model including de-jitter buffering for progressive streaming service

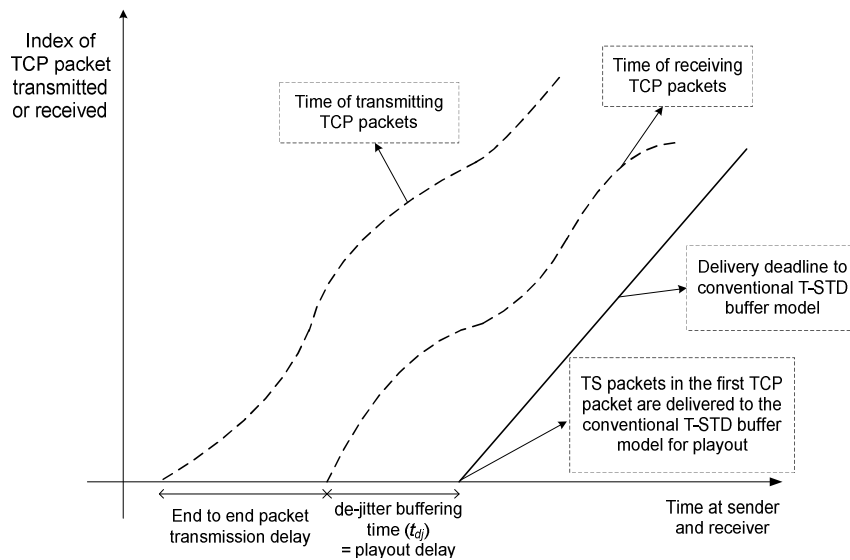


그림 8. 설정된 de-jitter 버퍼링에 의한 미디어 재생 시간 지연 (playout delay)
Fig. 8. Media playout delay caused by de-jitter buffering

링 시간 직후에 TCP 패킷에 포함된 TS 패킷들을 기존의 T-STD 버퍼 모델로 전달하는 시간적 모델링을 나타낸다.

그림 9는 본 연구의 결과를 실제적으로 프로그레시브 스트리밍 서비스에 적용하기 위한 송신 측 및 수신 측의 작업

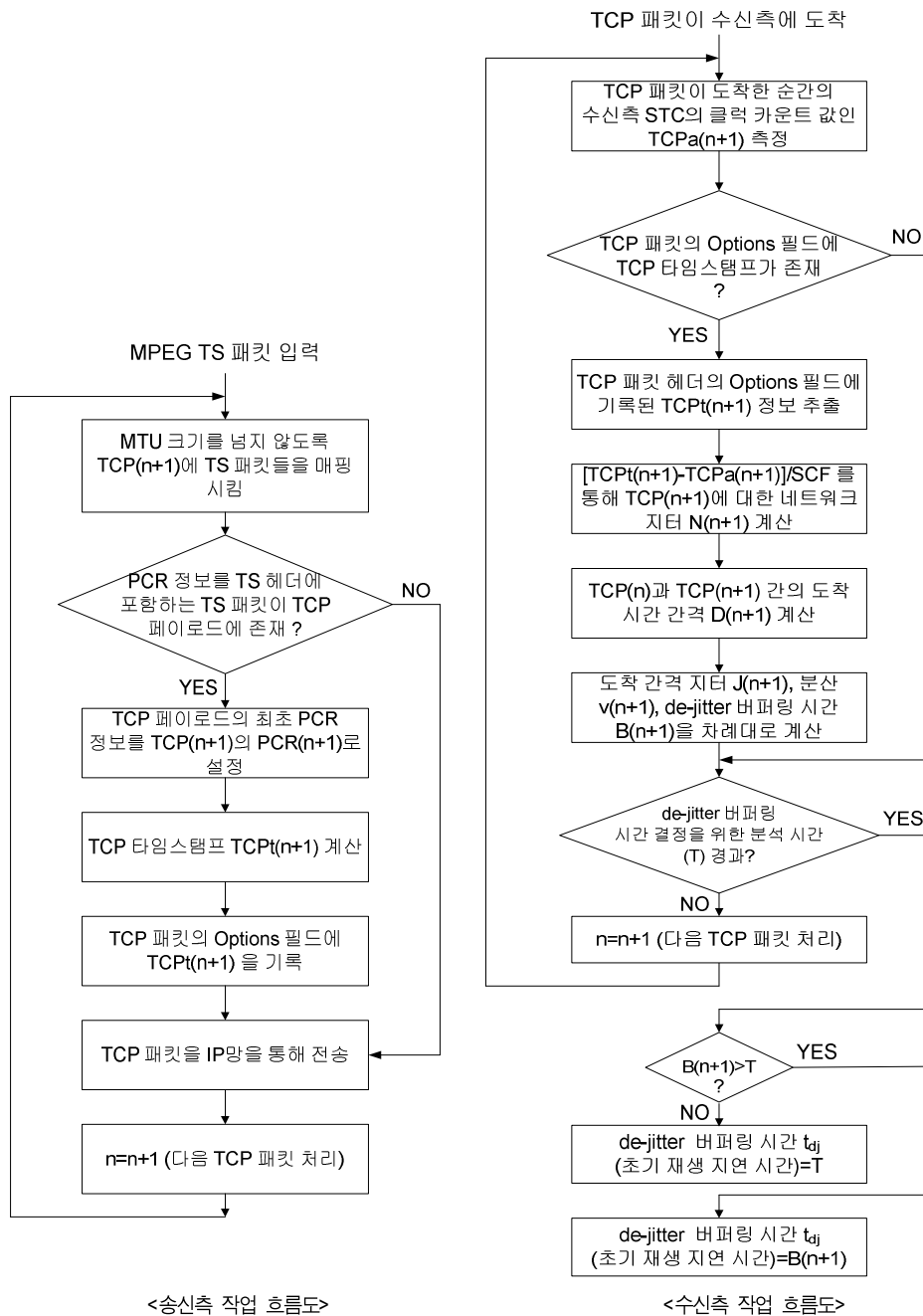


그림 9. 제안된 기법의 적용을 위한 송신 및 수신 측의 작업 흐름도
 Fig. 9. Flowchart of the proposed method at the sender and receiver sides

흐름도 (work flow)를 나타낸다. 그림 9의 과정은 스트리밍 서비스 도중에 발생할 수 있는 수신 측의 파일 버퍼 언더플로우가 발생할 경우 수신 측에서 리버퍼링을 수행할 경우에 대해서도 동일하게 적용된다.

IV. 실험 결과

de-jitter 버퍼링 시간은 수신측이 실제 프로그래시브 스트리밍 서비스를 경험하기 이전까지 기다려야 하는 서비스 지연시간과 패킷 손실률에 직접적인 영향을 미치게 된다. 그런데, 이 서비스 지연시간과 패킷 손실률 간에는 상호 상충 관계 (trade-off)가 존재한다. 즉, de-jitter 버퍼링 시간을 줄이게 되면 서비스 지연시간을 감축시킬 수 있지만, 네트워크 jitter 흡수에 불충분한 버퍼링 시간으로 인해 수신측에서의 패킷 손실률은 증가하게 된다. 반대로, de-jitter 버퍼링 시간을 늘리게 되면 서비스 지연시간은 증가하지만, jitter 흡수에 충분한 버퍼링 시간으로 인해 수신측에서의 패킷 손실률은 감소하게 된다. 따라서, 제안된 기법의 적용을 통해 추정된 de-jitter 버퍼링 시간은 서비스 지연시간과 패킷 손실률 측면에서 중요한 QoS (quality of service) 파라미터가 된다.

실제 인터넷을 통한 전송 환경에서 서비스 되는 프로그

래시브 스트리밍의 성능을 분석하기 위하여 이더넷 기반의 IP 망을 통해 프로그래시브 스트리밍 되는 CIF 규격의 다양한 MPEG-4 및 H.264 콘텐츠를 사용하여 성능을 비교하였다. 표 1은 실험에 사용된 콘텐츠 파일에 대한 주요 정보를 나타낸다. MPEG-4 비디오를 포함한 콘텐츠와 H.264를 포함한 콘텐츠는 모두 188 바이트 규격의 MPEG-2 TS로 제작되어 있으며, MPEG 시스템의 규격에 맞게 100 ms 이내에 최소한 한번 씩의 PCR 값이 MPEG-2 TS의 헤더에 기록되도록 하였다.

표 1. 실험에 사용된 MPEG-2 TS 스트림에 대한 주요 정보

Table 1. Parameters of MPEG-2 TS streams used in the experiment

	압축 부호화 방식	화면율 (fps)	부호화 율 (kbits/sec)	총 재생 시간 (sec)
해리포터 (TS-1)	MPEG-4	15	384	144
트리플 엑스 (TS-2)	MPEG-4	12	288	120
보아 (TS-3)	H.264	15	512	130
맨인블랙 (TS-4)	H.264	12	450	126

그림 10은 표 1의 MPEG-2 TS 스트림 파일에 대한 프로그래시브 스트리밍 서비스를 실시 할 때 그림 7의 de-jitter 버퍼링 시간을 일정하게 고정시킨 후에 관찰된 수신측에서의 패킷 손실률을 나타낸다. 그림 10에서의 결과는 각 MPEG-2 TS 스트림 마다 총 30회의 반복적인 실험을 수행

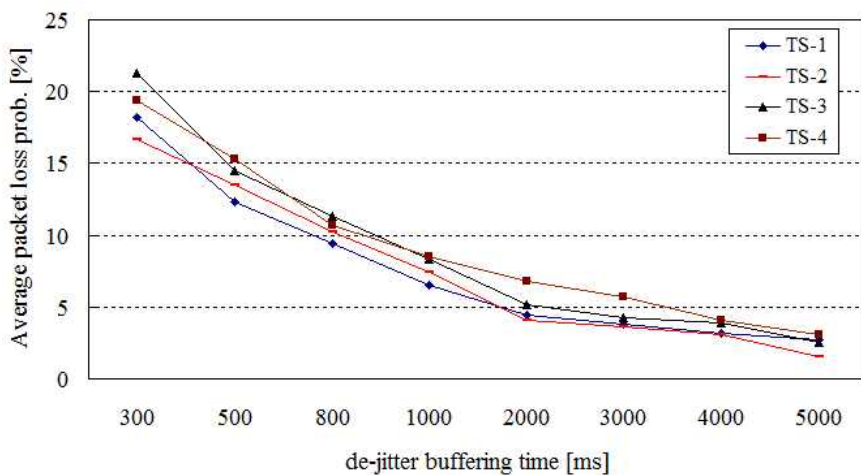


그림 10. 고정된 de-jitter 버퍼링 시간에 대한 평균 패킷 손실률

Fig. 10. Average packet loss ratio for fixed de-jitter buffering time

한 후에 평균적으로 얻어지는 패킷 손실률을 나타낸다. 전체적으로 de-jitter 버퍼링 시간이 증가할수록 평균 패킷 손실률은 점차적으로 낮아짐을 관찰할 수 있지만, 테스트 MPEG-2 TS 간의 패킷 손실 확률은 전송 시의 네트워크 상태에 의존하여 많은 차이를 나타낸다. 그림 11은 각각의 테스트 MPEG-2 TS 스트림에 대한 프로그래시브 스트리밍 서비스에서 수신측이 복호화를 시작하기 직전까지 겪게 되는 단대단 지연시간 (end-to-end delay time)을 나타낸다. 전

체적으로 de-jitter 버퍼링 시간이 증가할수록 단대단 지연 시간이 선형적으로 비례하여 증가함을 알 수 있다.

그림 12는 de-jitter 버퍼링 시간을 고정 시켰을 경우에 동일한 MPEG-2 TS 스트림에 대해 조차도 패킷 손실률의 편차가 매우 큼을 나타낸다. 그림 10의 ‘트리플 엑스’ (TS-2) MPEG-2 TS 스트림에 대해 총 30회의 독립적인 실험을 통해 얻어진 평균 패킷 손실률 (그림에서 Average로 표시)은 최대 패킷 손실률 (Maximum 으로 표시)과 최소

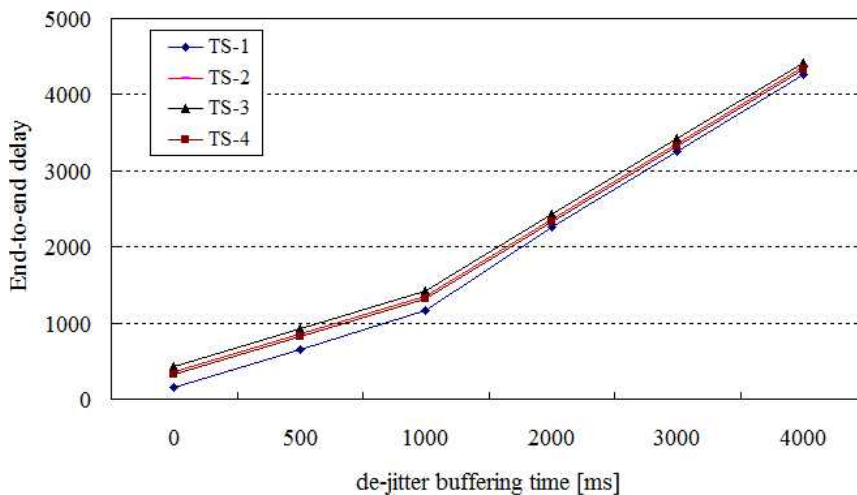


그림 11. 고정된 de-jitter 버퍼링 시간에 대한 단대단 지연 시간
Fig. 11. End-to-end delay for fixed de-jitter buffering time

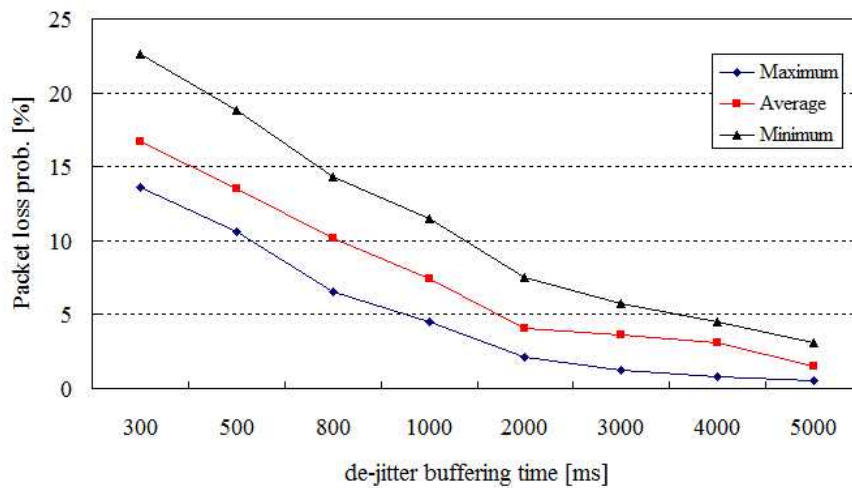


그림 12. TS-2 스트림에 대한 총 30회의 독립적인 실험에서 나타난 패킷 손실률의 변이
Fig. 12. Variation of packets loss ratio from 30 times of experiment for the TS-2 stream

패킷 손실률 (Minimum 으로 표시) 사이에 그 값의 범위가 위치하며, 최대 패킷 손실률과 최소 패킷 손실률 사이의 패킷 손실률 편차는 상당히 큰 편이다. 이는 채널에서의 전송 품질 상황을 전혀 고려하지 않은 채로 de-jitter 버퍼링 시간을 일률적으로 고정시켰기 때문이다.

그림 13은 패킷 전달 도중 네트워크 노드에서 인위적으로 추가적인 지연시간 100ms를 발생 시켰을 때 고정된 de-jitter 버퍼링 시간에 대한 평균적 패킷 손실 확률을 나타낸다. 이 실험에 적용된 인위적인 추가적 지연시간을 발생시키기 위해서 IP 네트워크 에뮬레이터인 NIST-Net^[11]을

활용하였고, ‘보아’ MPEG-2 TS 스트림 (TS-3)을 사용하여 테스트하였다.

그림 14는 제안된 de-jitter 버퍼링 시간 결정 기법을 적용하였을 경우에 각 MPEG-2 TS 스트림에 대한 평균 패킷 손실 확률을 20회 테스트한 결과를 나타낸다. 그림 14에 보이듯이 제안된 de-jitter 버퍼링 시간 추정 기법의 적용으로 네트워크 상태에 적합한 de-jitter 버퍼링 시간을 적응적으로 결정함으로써 패킷 손실률을 최소화 하고 있다. 전체적으로 패킷 손실 확률의 범위가 0.1~0.4 % 사이에 한정된다.

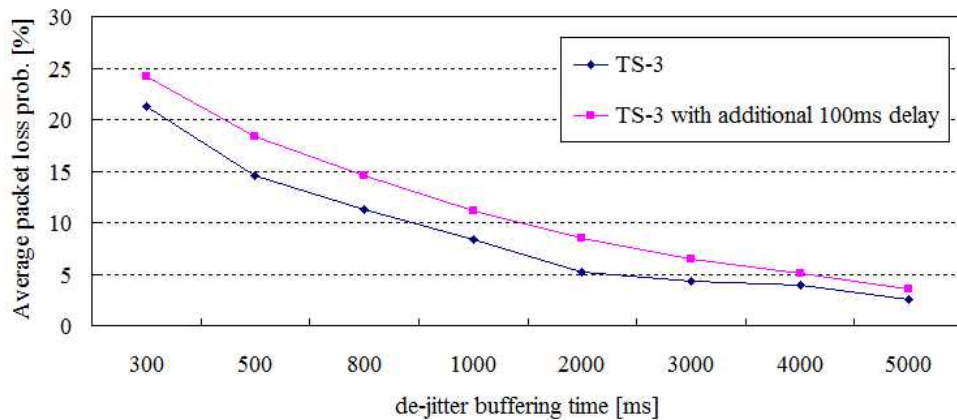


그림 13. TS-3 스트림에 대한 추가적인 100ms 네트워크 지연에 의한 패킷 손실률 상승
 Fig. 13. Increase in packet loss ratio for additional 100ms network delay for TS-3 stream

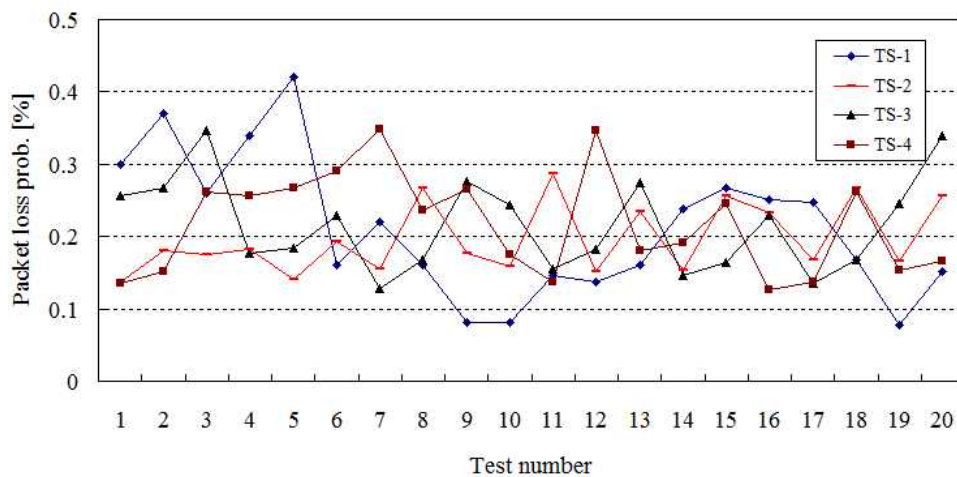


그림 14. 제안된 de-jitter 버퍼링 시간 추정에 의한 평균 패킷 손실률
 Fig. 14. Average packet loss ratio by the proposed estimation on de-jitter buffering time

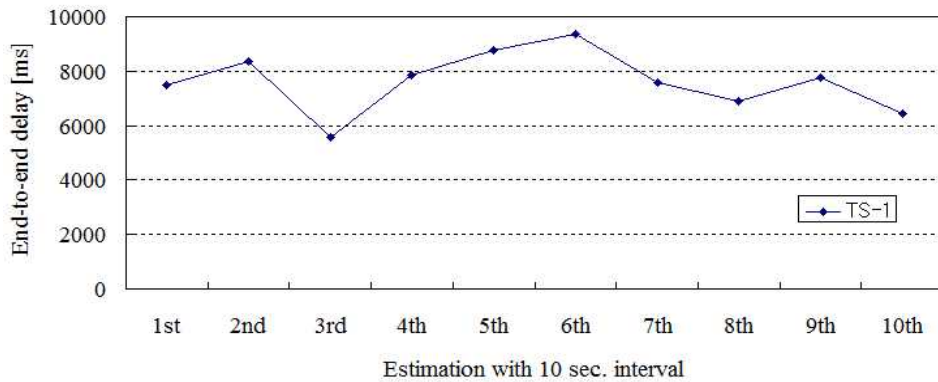


그림 15. 제안된 기법의 적용에 의한 단대단 지연 시간
 Fig. 15. End-to-end delay time by the proposed method

한편, 이 실험에서는 식 (12)에 사용된 변이를 흡수하기 위한 파라미터인 K 값을 1로 고정시켜서 실험하였다. 만약 K 값을 좀더 유연하게 확대한다면 단대단 지연시간을 증가시키는 비용을 통해 패킷 손실 확률을 감소시킬 수 있다.

그림 15는 ‘해리포터’ MPEG-2 TS (TS-1) 스트림에 대해 제안된 기법을 적용하였을 경우에 나타나게 되는 단대단 지연 시간의 변동을 나타낸다. 제안된 기법에서는 10초 주기로 de-jitter 버퍼링 시간을 갱신하여 네트워크의 지터를 흡수하도록 하였다. 이 실험에서는 NIST-NET을 활용하여 네트워크 노드에서 100~200 ms의 추가적인 지연이 빈번하게 발생하도록 하였다. 그림 15에서 보이듯이 전송 도중에 겪게 되는 다양한 지연 요소들의 영향을 고려하여 적응적으로 de-jitter 버퍼링 시간을 추정해 낼 수 있다. 제안된 기법의 적용으로 얻게 되는 단대단 지연시간은 대략 6000 ms~10000 ms 사이에 분포한다.

V. 결론

본 논문에서는 MPEG-2 TS를 유무선 인터넷 환경에서 프로그레시브 스트리밍 서비스를 통해 클라이언트로 전달할 때 발생하게 되는 TCP 패킷들 간의 도착 지연 시간 변이인 네트워크 jitter를 추정하는 방법을 제안 하였다. 추정된 네트워크 jitter 크기를 바탕으로 TS 패킷들이 수신측 T-STD 버퍼 모델에 입력 되기 이전에 지터를 흡수하기 위

해 필요한 de-jitter 버퍼링 시간의 적절한 크기를 효율적으로 결정할 수 있다. 본 연구를 통해 도출된 de-jitter 버퍼링 메커니즘을 프로그레시브 스트리밍 서비스에 적용할 경우 새롭게 추가되는 de-jitter 버퍼링 과정을 제외하고는 기존의 T-STD 버퍼 모델의 동작 원리를 그대로 활용할 수 있어서 기존의 MPEG 시스템의 버퍼 모델과 관련된 표준 기술을 수정 없이 이용할 수 있다. 제안된 기법은 최근에 ISO MPEG에서 개발 중인 DASH 서비스에 응용될 수 있다.

참고 문헌

- [1] IETF STD 0007: "Transmission Control Protocol," J. Postel, Sept. 1981.
- [2] K. Ma, R. Bartos, S. Bhatia, R. Nair, "Mobile video delivery with HTTP," IEEE Communications Magazine, vol.49, no.4, pp.166-175, Apr.2011.
- [3] C. Krasic, K. Li, J. Walpole, "The case for streaming multimedia with TCP," Proc. 8th Int. Workshop on Interactive Distributed Multimedia Systems, Berlin, pp.213-218, 2001.
- [4] ISO/IEC 14496-12, Information technology-Coding of audio-visual objects-Part 12: ISO base media file format, Third edition, Oct. 2008.
- [5] F. Nikolaus, D. Stefan, and I. Jochen, "Adaptive progressive download based on the MPEG-4 file format," Journal of Zhejiang University, vol.7, no.1, pp.106-111, 2006.
- [6] ISO/IEC JTC1/SC29/WG11 13818-1, Information technology - Generic coding of moving pictures and associated audio information: Systems, 1994.
- [7] IETF RFC 2616: "Hypertext Transfer Protocol - HTTP/1.1," R. Fielding et al., June 1999.
- [8] X. Qiu, H. Liu, D. Li, S. Zhang, D. Ghosal, "Optimizing HTTP-based

adaptive video streaming for wireless access networks,”
IEEE Int. Conf. on Broadband Network and Multimedia Technology, pp. 8
38-845, 2010.

[9] ISO/IEC JT C1/SC 9/WG 11, HTTP streaming of MPEG media con-
text and objectives, N11337, Dresden, April 2010.

[10] D. Wu, Y. Hou, and Y. Zhang, “Transporting real-time video over the
Internet: Challenges and approaches,” Proceedings of the IEEE,
Vol. 88, No. 12, pp. 1855-1877, Dec. 2000.

[11] NIST-Net, Software provided by National Institute of Standards and
Technology (NIST), <http://www.ntd.nist.gov/nistnet>.

저 자 소 개



서 광 덕

- 1996년 2월 : KAIST 전기및전자공학과 학사
- 1998년 2월 : KAIST 전기및전자공학과 석사
- 2002년 8월 : KAIST 전기및전자공학과 박사
- 2002년 8월 ~ 2005년 2월 : LG전자 단말연구소 선임연구원
- 2005년 3월 ~ 현재 : 연세대학교 컴퓨터정보통신공학부 부교수
- 주관심분야 : 영상부호화, 영상통신, 디지털 방송, DASH, MMT



김 현 정

- 2010년 8월 : 연세대학교 컴퓨터정보통신공학부 학사
- 2010년 9월 ~ 현재 : 연세대학교 전산학과 석사과정
- 주관심분야 : 영상부호화, 영상통신, 디지털 방송



김 진 수

- 1991년 2월 : 경북대학교 전자공학과 학사
- 1993년 2월 : KAIST 전기및전자공학과 석사
- 1998년 8월 : KAIST 전기및전자공학과 박사
- 1997년 ~ 2000년 : 삼성전자 선임연구원
- 2000년 ~ 현재 : 한밭대학교 정보통신컴퓨터공학부 교수
- 주관심분야 : 멀티미디어 스트리밍, SVC, Distributed Video Coding, 디지털 방송



정 순 흥

- 2001년 2월 : 부산대학교 전자공학과 학사
- 2003년 2월 : KAIST 전자전산학과 석사
- 2003년 3월 ~ 2005년 3월 : LG전자 주임연구원
- 2005년 4월 ~ 현재 : ETRI 방통융합미디어연구부 선임연구원
- 2010년 9월 ~ 현재 : KAIST 전기및전자공학과 박사과정
- 주관심분야 : 영상부호화, 멀티미디어방송, HEVC, 영상통신

저 자 소 개



유 정 주

- 1982년 2월 : 광운대학교 전자통신공학과 학사
- 1984년 2월 : 광운대학교 전자통신공학과 석사
- 2001년 7월 : 영국 Lancaster 대학교 컴퓨터공학과 박사
- 2007년 ~ 2009년 : MPEG 한국대표단 단장
- 1984년 3월 ~ 현재 : ETRI 스마트TV사업단 책임연구원
- 주관심분야 : 스마트TV, 멀티미디어 스트리밍, MMT, DASH



정 영 호

- 1992년 2월 : 전북대학교 전자공학과 학사
- 1994년 2월 : 전북대학교 전자공학과 석사
- 2006년 8월 : 충남대학교 전자공학과 박사
- 1994년 3월 ~ 현재 : 한국전자통신연구원 스마트TV시스템연구팀 팀장
- 2011년 3월 ~ 현재 : 과학기술연합대학원대학교(UST) 겸임교수
- 주관심분야 : 스마트TV, 중강방송, 디지털방송시스템/서비스, DCAS, DMB, TPEG, OFDM