

이질적 환경에서의 layering 되어진 멀티미디어 스트림 서비스

이흥기^o, 김현정, 김상형, 김현정, 신정아, 이성인, 유관중,
충남대학교 컴퓨터학과

{helius^o, heriskim, shkim, ppuyo, sinnja, shlee, kijoo}@nspplab.cs.cnu.ac.kr

Stream servicing for layered Multimedia Stream in Heterogeneous system

Lee Heung-ki^o, Kim Hyun-Jung, Kim Sang-Hyung, Kim Hyun-Jung, Shin Jung-Ah, Lee Sung-In,
Yoo Kawn-Jong
Dept. of computer science, Chung Nam University

요 약

다양한 네트워크 환경에서의 멀티미디어 서비스에 대한 연구가 많이 이루어지고 있다. 그러나, 아직까지 멀티미디어 데이터의 양을 감당할 수 있는 네트워크 환경이 구성되어 있지 않고 있다. 이러한, 가변적인 네트워크 환경에 적응하면서, 사용자의 욕구를 만족시키기 위한 연구가 활발히 이루어지고 있다. 그 중에서 layering 되어진 멀티미디어 데이터에 대한 연구가 많이 진행 중이다. 이에 본 논문은 이러한 layering data의 전송방법에 대하여 논하고자 한다.

1. 서 론¹⁾

최근 멀티미디어에 대한 관심이 커지면서, 멀티미디어의 저장과 전송에 대한 연구가 많이 이루어지고 있다. 이전 단순히 데스크탑에 유선으로 연결되어진 네트워크의 형태가 아니라, 여러 가지 다른 형태에서의 멀티미디어 서비스에 대한 연구가 이루어지고 있다. PDA, Mobile phone에서의 동영상 서비스가 그 대표적인 예라고 할 수 있다. 이러한 환경에서의 멀티미디어 서비스는 몇 가지 특수성을 갖는다. 비교적 작은 메모리 공간, 낮은 데이터 처리 능력, 무선 네트워크 환경 등이다. 이러한 환경은 멀티미디어 데이터 처리를 쉽지 않게 하고 있다. 따라서, 이러한 환경에서의 멀티미디어 서비스가 많이

연구 중이다. 이 중에서 layering method은 이러한 문제점의 해결책으로 제시되어지고 있다. 이에 본 논문은 layered 되어진 멀티미디어 데이터의 전송방법에 대해서 논하고자 한다. 2장에서는 layered coding 방식과 전송 방법에 대한 관련 연구에 대해서 알아보고, 3장에서는 layered stream을 전송방식에 대해서 논하며, 4장에서 결론을 맺는다.

2. 관련 연구

2.1 Scalable Coding

Scalable Coding이란, 하나의 영상 서비스를 하나의 stream으로 제공하는 것이 아니라, 여러 개의 나누어진 stream을 통해서 서비스를 수행하는 것이다. 서비스를 제공하기에 반드시 필요한 Base Layer와 존재할 경우에 좋은 서비스를 제공하지만 반드시 필요로 하지 않는 Enhancement Layer가 존재한다. MPEG system에서는 크게 4 가지 Scalability가 있다. Temporal Scalability,

1) 본 연구는 BK21 충남대학교 정보통신 인력 양성 사업단의 지원을 통하여 이루어 졌습니다.

Spatial Scalability, SNR Scalability 그리고 Data Partitioning이 있다. [1][3]

Temporal Scalability는 시간적 개념을 이용하여 scalable한 서비스를 제공하는 방법이다. 서비스 제공자가 단위 시간에 보여지게될 frame의 수를 조절하는 방식이다. Spatial Scalability는 두 개의 생성 stream의 크기가 다른 stream의 생성하여, 이 두 개의 Stream을 이용하여 서비스를 제공하는 것이다. SNR scalability는 양자화를 수행할 때의 계수의 값을 조절하는 것이다. 마지막으로 Data Partitioning방법은 중요한 데이터 (DCT계수에서 DC 값)를 다른 데이터와 분리하여 stream내에서 분리하여 배치하는 방법이다. 그리고 이러한 Scalability 중에 하나 이상의 서비스를 동시에 적용하는 Hybrid 방식이 있다.[3]

이러한 scalability는 한 번의 설정이 이루어지면, 계속하여 그 성질을 유지하는 것이 아니다. 가변적으로 변화하는 네트워크의 상황에 따라서 변화하는 것이 가능하다. 즉, 일정 시간동안은 base layer 만이 제공되어지며, 임의의 시간에서는 base layer 와 enhancement layer가 동시에 서비스를 제공하는 것이다. [1]

2.2 Streaming Service

Streaming서비스를 제공하는 방식에는 서비스 제공을 주도하는 쪽이 어느 쪽이냐에 따라서 크게 2가지 형태로 나누어질 수 있다. 하나는 server가 서비스를 주도하는 Push 방식과 client에서 서비스를 주도하는 Pull 방식이다.

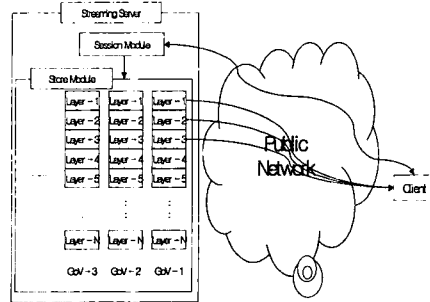
Push 방식은 client가 처음 서비스를 요청할 때, 자신의 버퍼의 크기와 디코더의 Profile을 제공한다. 이러한 자료를 가지고서, 서버는 전송할 수 있는 데이터를 결정하게 된다. 이에 반하여 Pull방식에서는 client측에서 service 목록을 요청한 다음에, 자신이 필요한 데이터를 server측에서 데이터를 전송 받는 방식이다. [3]

3. Layered 스트림 서비스

3.1 서비스 정책

하나의 Stream을 여러 개의 stream으로 나누는 목적은 QoS를 보장하는 측면이 강하다. 그러나, 이러한 QoS 측정을 서버에서 단독을 결정하기는 어려운 일이다. 따라서, 보통 그 측정이 client측에서 이루어진다. 즉, client측에서 자신이 받은 layer의 크기를 가지고서 현재의 네트워크상황을 판단하는 것이다. 이러한 것을 고려할 때, Push 방식보다는 Pull 방식의 적용이 유리하다.

이것에 대한 서비스 개요도가 다음과 같다. 즉, 클라이언트는 자신의 맞는 서비스 선택하여 서버에 요구하게 된다.



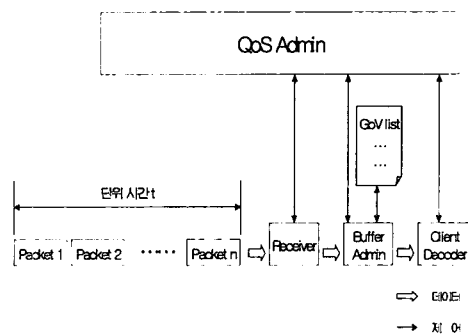
[그림 1] Stream Servicing Diagram

3.2 QoS level의 결정

QoS level의 설정은 두 단계를 통해서 이루어진다. 하나는 server측에 요청한 데이터의 level이고, 다른 하나는 실제적으로 서버 측에 도달되어진 데이터의 level이다. 이렇게 QoS level을 두 단계로 나누는 이유는 실제적으로 네트워크에서 전달 시, 데이터의 loss가 일어날 수 있기 때문이다.

우선, 서버 측에 요청하는 level은 다음과 같은 데이터를 기준으로 이루어진다.

- ① 이전 GoV영역에서의 전송되어진 Packet의 양.
- ② 클라이언트의 버퍼 영역의 크기.
- ③ 클라이언트의 처리 능력.



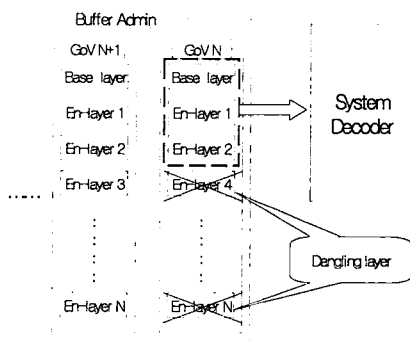
[그림 2] Demanding QoS level

전송되어진 Packet의 양은 클라이언트에서 counting을 수행함으로써 이루어질 수 있다. 버퍼 영역의 크기는 클라이언트가 처음 연결 시, 서비스에 대하여 필요한 자료를 가지고서 이루어진다. 클라이언트의 처리 능력은 디코더가 데이터의 시간 내에 디코딩이 수행되어지는가를 측정함으로써, 얻어질 수 있다.

이렇게 결정되어진 level은 서버 측에 전송되어지며, 이러한 내용을 전송 받은 서버는 다음 GoV의 전송 level을 결정하게 된다. 그러나, 실제적으로 도달한 QoS

level은 client에서 요구한 내용과 일치하지 않는다. 따라서 전송되어진 실제 QoS level을 결정하기 위해 다음의 규칙을 적용한다.

- ① Base layer가 존재할 때 그 GoV의 처리가 이루어진다.
- ② 상위 layer가 존재하는 layer중에서 가장 하위 layer가 전송 level을 결정한다.
- ③ 상위 layer가 존재하지 않는 layer는 dangling layer로 인식되어지며, 해당 데이터는 버려지게 된다.

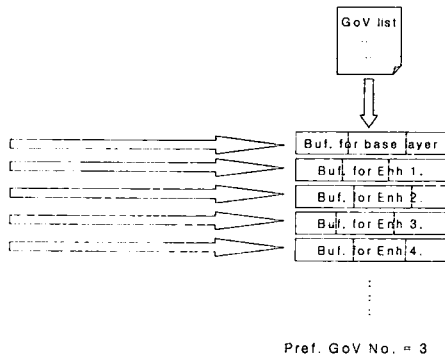


[그림 3] Receiving QoS level

이러한 기준으로 얻어진 QoS level은 디코더가 전송되어진 데이터를 디코딩을 수행하는데 적용되어지게 된다.

3.3 버퍼 관리 정책

하나의 layered stream에 하나의 버퍼 공간의 할당을 기준으로 한다. 시스템에서 허용된 버퍼의 크기를 전송받고자 하는 layered stream의 개수로 나누어서 할당하는 것이다. 그리고, Pre-fetch의 양을 정한다.

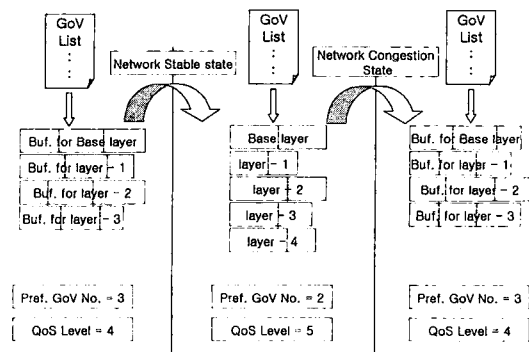


[그림 4] 초기 상태에서의 버퍼 할당

Presentation time이 되어지도록, 서버로부터 전송이

수행되어지지 않을 경우에, Jitter가 발생했음으로 인식한다. 그리고 할당되어진 버퍼의 크기에 대한 변동이 이루어진다. QoS level을 다운시키고, 그 level을 위하여 할당되어진 버퍼를 상위 layer를 위한 버퍼공간으로 재 할당이 수행되어진다.

할당되어진 layer에 대한 데이터가 전송되어질 경우에는 네트워크 상황이 호전되었다고 인식한다. 이 경우에, QoS level을 상향조정하게 된다. 그리고, 상위 layer에 할당되어진 일정 부분을 모아서 새로운 layer에 새로운 버퍼 공간이 할당되어질 수 있도록 한다.



[그림 5] 네트워크 상황에 따른 버퍼 관리

4. 결론

많은 양의 멀티미디어 데이터를 처리가 이루어지기 위해서 클라이언트에서의 QoS level의 설정과 버퍼 공간의 활용에 대해서 논하였다. PD나 기타 여러 가지의 기기에서는 네트워크 상황에 대처하는 능력에 대한 연구가 더 이루어져야 할 것이다. 무선 환경에서 서비스 제공에 대한 연구에 대하여 깊은 연구가 이루어져야 할 것이다.

5. 참조 문헌

- [1] 김현정, 손호신, 유관중, "적응형 미디어 전송을 위한 데이터 분할기술", 한국멀티미디어학회 추계 학술발표대회 제 4권 2호, 2001 11.
- [2] Jae-Wook Kim and Rhan Ha, Hojung Cha, "Bandwidth Adaptive Smoothing for multimedia Delivery", 2000 IEEE International Conference On Multimedia and Expo, August 2000.
- [3] K.R. Rao, Zoran S. Bojkovic, "Packet Video Communications ATM Networks", Prentice Hall PTR Upper Saddle River, 2000.