

다중 경로를 이용한 TCP/IP 기반 고품질 및 고장 감내 비디오 전송 시스템

A video transmission system for a high quality and fault tolerance based on multiple paths using TCP/IP

김 남 수¹ 이 종 열² 편 기 현^{1*}
Nam-su Kim Jong-Yeol Lee Kihyun Pyun

요 약

e-러닝의 활성화와 인터넷 비디오 서비스 증가로 인하여 대규모 사용자에 대한 비디오 전송 서비스에 대한 요구가 매년 증가하고 있다. 이러한 요구를 충족시키기 위해 기존 방식은 비디오 서버를 루트 노드로 갖는 트리 구조를 이용하는데 이 방식은 경로상의 한 노드에서만 문제가 발생해도 그 노드 하위 노드들 전체의 서비스가 단절되는 위험성을 안고 있다. 본 연구에서는 다중 경로를 활용한 주문형 비디오 전송 서비스를 제안한다. 기존의 구조에서 사용하는 경로에 백업과 속도 향상을 위한 새로운 경로를 추가함으로써 상위 노드의 문제가 하위 노드의 서비스에 미치는 영향을 방지하여 고품질의 비디오 서비스를 제공할 수 있음을 실험을 통해 보인다.

☞ 주제어 : 비디오 스트리밍, 다중 경로 전송

ABSTRACT

As the e-learning spreads widely and demands on the internet video service, transmitting video data for many users over the Internet becomes popular. To satisfy this needs, the traditional approach uses a tree structure that uses the video server as the root node. However, this approach has the danger of stopping the video service even when one of the nodes along the path has a some problem. In this paper, we propose a video-on-demand service that uses multiple paths. We add new paths for backup and speed up for transmitting the video data. We show by simulation experiments that our approach provides a high-quality of video service.

☞ keyword : video streaming, multiple path transmission

1. 서 론

e-러닝의 활성화와 인터넷 비디오 서비스 증가로 인하여 대규모 사용자에 대한 비디오 전송 서비스에 대한 요구가 매년 증가하고 있다. 이러한 요구를 충족시키기 위한 단순 클라이언트-서버 모델은 확장성에 문제가 있다. 이 문제를 완화하기 위해 오버레이 멀티캐스트(overlay multicast) 혹은 P2P (Peer-to-Peer) 멀티캐스트 방식을 사용하는데 대부분 비디오 서버를 루트 노드로 갖

는 트리 구조를 사용하고 있다[1]-[18].

하지만 비디오 서버를 루트 노드로 갖는 트리 구조는 서비스 단절의 위험성을 안고 있다. 트리 구조는 하나의 경로만을 사용하여 데이터를 전송하기 때문에 경로 상에 있는 중계 역할을 수행하는 노드가 갑자기 종료되었거나, 비디오 데이터 전송 속도에 문제가 있을 경우 중계 노드의 하부에 연결된 모든 노드들이 재생에 필요한 데이터가 부족하게 되어 재생이 중단될 수 있다.

이를 해결하기 위해 대부분의 기존 연구는 사용자들의 연결을 효율적으로 관리하기 위한 구조에 초점이 맞추어져 있다. 하지만 효율적인 관리 구조도 중요하지만 고품질의 비디오 서비스를 위해서는 노드의 비디오 재생이 중단되지 않는 것이 무엇보다 중요하다.

본 논문에서는 비디오 전송 서비스를 위해 하나의 경로만을 사용하는 것이 아니라 기본 경로 이외에 백업 경

¹ Dept. of Computer Science and Engineering, Chonbuk National Univ., 561-756, Korea.

² Department of Electronic Engineering, Chonbuk National Univ., 561-756, Korea.

* Corresponding author (khyun@chonbuk.ac.kr)

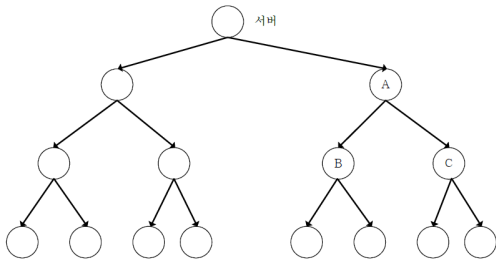
[Received 31 March 2014, Reviewed 3 April 2014, Accepted 15 September 2014]

로와 속도 향상 경로를 활용하는 다중 경로 서비스를 제안한다. 기존의 구조에서 사용하는 경로에 새로운 경로를 추가함으로써 상위 노드의 문제가 하위 노드의 서비스에 미치는 영향을 방지하여 고품질의 비디오 서비스를 제공할 수 있다.

2. 관련 연구

2.1 전통적인 비디오 전송 시스템

전통적인 비디오 전송 시스템은 대부분 트리구조를 사용한다. 그림 1은 가장 기본적인 트리구조를 사용한 비디오 서비스 전송 시스템의 한 예를 보여준다.



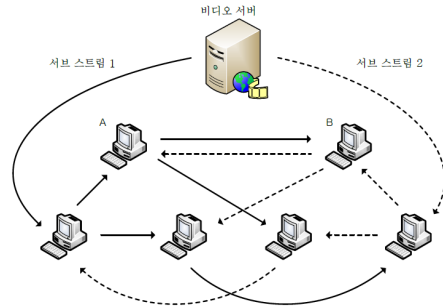
(그림 1) 완전이진 트리 구조의 비디오 전송 시스템 예
(Figure 1) An example of video transmission system using a complete binary tree

대부분의 기존 연구는 그림 1에서 노드 A가 비정상적으로 종료되었을 때 그 노드에 연결되어 있던 노드 B, 노드 C의 서비스 지속을 위해 새롭게 연결하는 방법에 초점을 맞춘다. 이 경우 새로운 연결을 위한 서비스 지연이나 서비스 품질 저하가 발생한다.

2.2 스플릿스트림

스플릿스트림(SplitStream)은 다중경로를 사용하여 비디오 서비스를 제공하는 기술이다[11]. 이 기술은 하나의 비디오 스트림을 다중표현부호화 기법을 사용하여 여러 개의 서브 스트림으로 나누고, 이를 각각 다른 경로를 통하여 전송하는 방법이다. 각각의 서브 스트림은 독립적으로 재생이 가능하며 전송받는 서브 스트림의 개수가 증가할수록 화질이 좋아진다.

그림 2는 하나의 비디오 데이터 스트림을 두 개의 서브스트림으로 나누어 전송하는 스플릿스트림의 한 예를 나타낸 것이다.



(그림 2) 스플릿스트림의 비디오 전송 예
(Figure 2) An example of SplitStream video transmission

그림 2에서 첫 번째 서브 스트림은 실선으로 표시된 경로를 통하여 전송되고, 두 번째 서브 스트림은 점선으로 표시된 경로를 통하여 전송된다. 서브 스트림 1의 전송에서는 노드 A는 노드 B의 상위 노드이나 서브 스트림 2의 전송에서는 노드 A가 노드 B의 하위 노드가 된다. 이와 같이 스플릿스트림에서는 각각의 서브 스트림마다 서로 다른 트리를 생성하기 때문에 하나의 서브 스트림에서의 상위 노드가 다른 서브 스트림에서도 상위 노드라고 말할 수 없다. 따라서 노드 A에 문제가 발생하면 노드 B는 서브 스트림 1을 전송받지 못하게 된다. 하지만 점선으로 표시된 경로를 통하여 서브 스트림 2를 전송받을 수 있기 때문에 재생이 중단되는 것을 피할 수 있다. 다만 서브 스트림 1을 받지 못하기 때문에 화질이 저하된다.

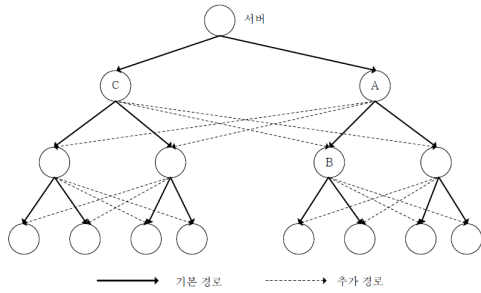
또한 스플릿스트림은 다중 스트림으로 나눌 수 없는 비디오 데이터에 적용할 수 없는 제한점을 갖는다.

3. 시스템 구조

3.1 기본 아이디어

비디오 전송 시스템은 그림 3에서 보는 바와 같이 전통적인 구조와 비슷하게 하나의 서버와 다수개의 노드들로 트리 구조를 형성한다.

그림 3에서 실선으로 연결된 트리 구조는 기존 전통 방식을 그대로 활용한다. 비디오 데이터는 서버에만 저장되며 각 노드는 서버로부터 전송받은 비디오 데이터를 재생하고, 다른 노드들에게 전송하는 중계 역할을 동시에 수행하게 된다.



(그림 3) 추가 경로를 포함한 네트워크 토폴로지 구조 예
(Figure 3) An example of topology structure including additional paths

기존의 트리 구조와 차별화되는 점은 그림 3에서 보는 바와 같이 각 노드는 자신의 형제 노드의 자식 노드들에게 점선으로 표시된 추가 경로를 가질 수 있다는 점이다. 추가 경로는 기본 경로 상의 노드에 해당되는 컴퓨터가 갑작스럽게 종료되었을 때 하위 노드들의 서비스가 단절되는 문제를 해결함과 동시에 네트워크 대역폭이 부족해서 발생하는 서비스 단절 문제를 해결할 수 있다. 예를 들어, 그림 3의 노드 A가 갑자기 종료되었다고 가정해 보자. 기본 경로만을 유지하는 경우 노드 B와 그 하위 노드들은 일정시간 뒤에 비디오 재생이 중단된다. 그러나 추가 경로가 존재하는 경우 노드 B는 비디오 재생이 중단될 위험성이 발생할 때 추가 경로 C로부터 비디오 데이터를 수신 받을 수 있다. 그 결과 노드 B와 그 하위 노드들은 서비스 단절이 발생하지 않는다.

3.2 노드간 전송방식

본 시스템은 주문형 비디오 서비스를 대상으로 하기 때문에 초기 재생 지연이 조금 발생되는 것이 큰 문제가 되지 않고 화면 깨짐이 없는 고품질 비디오 전송 서비스가 더욱 중요하다. 비디오 데이터는 만일 비디오 프레임 재생 시점에 해당 비디오 프레임의 일부분이라도 부족하게 되면 전체 비디오 프레임 재생 품질에 악영향을 끼치게 된다. 가령 1% 데이터 손실이 10% 이상의 비디오 품질 손실을 발생시킬 수도 있다.

이러한 환경에서 각 노드는 비디오 데이터를 버퍼에 저장하여 일정량을 쌓은 이후 재생을 시작하므로 비디오 데이터 전송 오류나 패킷 손실이 발생하더라도 다시 전송할 시간적 여유가 있다. 이와 같은 환경에서 비디오 패킷 손실이 없는 최고 품질을 제공하기 위해서는 UDP 방식보다 TCP를 활용한다. 이 때 하나의 노드는 몇 개의

자식 노드들에게만 비디오 데이터를 중계하면 되므로 TCP 사용으로 인한 확장성 문제는 발생하지 않는다. 또한 실시간 방송 서비스의 경우도 어느 정도의 지연을 사용자가 인내할 수 있으면 본 시스템을 적용할 수 있을 것으로 기대한다.

한편 UDP를 사용하는 경우와 달리 노드간 비디오 데이터 전송 방식으로 TCP를 사용하게 되면 상위 노드의 비디오 버퍼와 하위 노드의 비디오 버퍼간의 동기화가 추가적으로 필요하게 된다. 여기서 동기화란 상위 노드의 비디오 버퍼에 저장된 데이터는 그 노드에서 비디오 재생이 완료되어 해당 데이터가 불필요하게 되었다 할지라도 하위 노드의 비디오 버퍼에 저장될 때 까지 버리지 말고, 하위 노드들이 그 비디오 데이터를 완전히 수신할 때 까지 기다려야 한다는 것을 의미한다.

3.3 노드의 동작 방식

각 노드는 수신된 비디오 데이터를 저장하는 비디오 버퍼를 1개씩 가지고 있다. 이 비디오 버퍼는 비디오 데이터를 재생하기 위한 용도로 사용됨과 동시에 하위 노드에게 데이터를 전달하는 전송 버퍼의 역할도 수행한다. 또한 하위 노드가 이 노드를 추가 경로로 설정하는 경우에는 백업 역할 및 속도 향상의 역할도 수행한다.

이 논문에서는 편의상 하위 노드가 이 노드를 추가 경로로 설정하지 않는 경우 이 노드가 재생 및 전송 역할을 수행한다고 명명한다. 또한 이 노드가 백업 노드의 역할도 수행하는 경우 이 노드가 백업 역할을 수행한다고 명명하며, 마찬가지로 속도 향상의 역할을 수행하는 경우를 속도 향상 역할을 수행한다고 말한다. 각 노드는 이제 세 가지 역할을 독립적으로 수행하기 때문에 이 역할들은 서로 중첩될 수 있다.

3.3.1 초기 설정

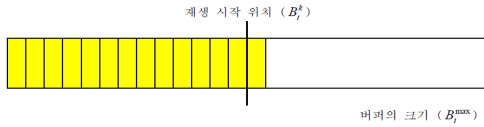
비디오 서비스를 원하는 각 노드는 비디오 서버에게 비디오 전송을 요청하게 된다. 전송 트리 구성 방식은 기존의 전통적인 방식을 따르므로 자세한 설명은 생략한다.

그러나 기존의 전통적인 방식과 달리 본 시스템은 비디오 재생을 서비스를 시작하기 전에 비디오 트레이스 파일(video trace file)을 미리 전송받는다. 비디오 트레이스 파일은 비디오 재생에 필요한 정보를 담은 파일로 초당 비디오 프레임 수를 포함하는 헤더와 비디오 프레임 크기들이 순차적인 나열한 텍스트 파일이다. 보통 수 KB 정도 크기로 전체 시스템 성능에 영향이 없다. 본 연구는

이 비디오 트레이스 파일을 이용하여 비디오 버퍼에 남은 데이터로부터 정확한 재생 시간을 계산할 수 있다.

3.3.2 재생 및 전송 역할

그림 4는 노드 i 의 비디오 버퍼 B_i 를 표현한 것이다.



(그림 4) 노드의 비디오 버퍼

(Figure 4) The video buffer of the node

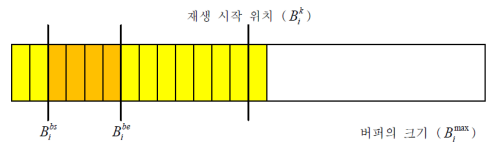
각 노드 i 의 비디오 버퍼 크기는 고정 크기 B_i^{max} 라고 가정한다. 비디오 데이터가 수신되기 시작하면 바로 재생이 시작되는 것이 아니라 고정 임계값 B_i^k 바이트를 넘을 때까지 재생을 미루게 된다. 이 임계값을 너무 크게 잡게 되면 불필요하게 비디오 재생이 지연되고, 이 값을 너무 작게 잡게 되면 비디오 데이터 전송 오류나 손실이 발생했을 때 다시 전송받을 수 있는 시간이 줄어들기 때문에 비디오 재생 중단이 발생할 위험이 커진다. 적절한 임계값을 설정하는 것은 매우 중요한 이슈이며 이 논문의 향후 연구에서 계속할 예정이다.

3.3.3 백업 역할

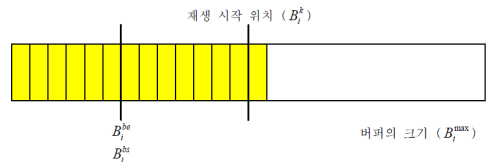
각 노드는 백업 역할을 요청받아 백업 역할을 수행할 수 있다. 하위 노드가 여러 가능한 백업 노드들 중 어떤 노드를 선정하는 지에 대한 논의는 3.3.5 절로 미루고, 이 절에서는 한 노드가 백업 노드로 선정되었을 때 그 노드의 버퍼가 백업 역할을 수행하는 방법을 설명한다.

그림 5에 보는 바와 같이 하위 노드에서 이 노드에게 백업 역할을 요청할 때 비디오 데이터의 P 지점에서 Q 지점까지 전송해 달라고 요청하는 경우를 가정하자. 이때 이 P 지점이 백업 역할의 시작점이 되고 여기에 해당하는 버퍼 지점을 B_i^{bs} 라고 표기한다. 또한 Q 지점에 해당하는 지점을 B_i^{bc} 라고 표기한다.

B_i^{bs} 는 고정된 값이 아니라 하위 노드에게 해당 데이터 전송을 완료하게 되면 점차적으로 그 값이 증가하게 된다. 그 결과 B_i^{bs} 값이 B_i^{bc} 와 동일하게 되면 이 노드의 백업 역할은 종료된다. 비디오 버퍼 내에 데이터는 B_i^{bs} 값 이전에 위치한 비디오 데이터만을 잃을 수 있다.



(a) 백업 역할 초기의 버퍼 상태



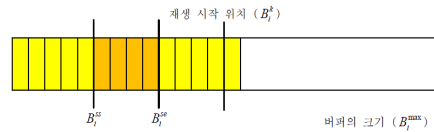
(b) 백업 역할 완료시의 버퍼 상태

(그림 5) 백업 역할 수행시의 비디오 버퍼 예

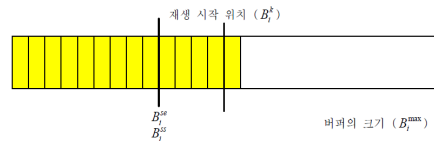
(Figure 5) An example of the video buffer as the back-up role

3.3.4 속도 향상 역할

각 노드는 백업 역할뿐만 아니라 속도 향상 역할을 수행할 수 있다. 그림 6에서 보는 바와 같이 하위 노드에서 이 노드에게 속도 향상 역할을 요청할 비디오 데이터의 X 지점에서 Y 지점까지 전송해 달라고 요청한 경우를 가정하자.



(a) 속도 향상 역할 초기의 버퍼 상태



(b) 속도 향상 역할 완료시의 버퍼 상태

(그림 6) 속도 향상 역할 수행시의 비디오 버퍼

(Figure 6) The video buffer as the speed-up role

이 X 지점이 속도 향상 역할의 시작점이 되고 여기에 해당하는 버퍼 지점을 B_i^{ss} 라고 표기하자. 또한 Y 지점에 해당하는 버퍼 지점을 B_i^{se} 라고 표기하자. B_i^{ss} 는 고정된 값이 아니며, 하위 노드에게 해당 데이터 전송을 완료하게 되면 점차적으로 그 값이 증가하게 된다. 그 결과 B_i^{ss} 값이 B_i^{se} 와 동일하게 되면 이 노드의 속도 향상

역할은 종료된다. 비디오 버퍼내의 데이터의 삭제는 B_i^{ss} 값 이전에 위치한 비디오 데이터만을 없앨 수 있다.

노드가 백업 역할과 속도 향상 역할을 동시에 수행하고 있는 경우 B_i^{ss} 와 B_i^{bs} 의 값들 중 작은 값을 기준으로 그 이전 비디오 데이터를 없앨 수 있다.

3.3.5 백업 역할 및 속도 향상 노드 선정하기

각 노드는 자신의 부모 노드에 대한 형제노드들을 백업 역할 및 속도 향상 역할에 대한 후보자들로 삼는다. 후보자 노드들을 이와 같은 방식으로 제한한 이유는 확장 문제가 발생하는 것을 막기 위한 것이다.

여러 후보자들 중에 어느 노드를 백업 또는 속도 향상 노드로 선정할 지는 비디오 데이터를 수신하는 하위 노드가 결정한다. 또한 선정된 상위 노드에서 비디오 데이터의 어느 부분을 전송해 줄지에 대한 결정도 하위 노드가 결정한다. 이와 같이 하위 노드가 추가 경로를 선정하고 데이터 할당 영역을 결정하기 때문에 수신자 주도 방식으로 볼 수 있다.

백업 역할 및 속도 향상 역할은 특정 비디오 데이터 영역에 대해서 매번 새롭게 설정함으로써 비디오 전송 관리를 세밀하게 조종할 수 있다. 그러나 만일 백업 역할이나 속도 향상 역할을 해당 비디오 데이터 전송을 완료한 이후 매번 새롭게 시작하는 방식을 사용하게 되면 TCP 전송의 slow-start 특성으로 인한 불필요한 전송 속도 저하 문제가 발생할 수 있다.

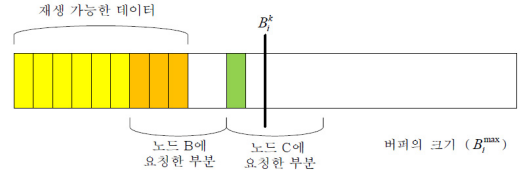
본 시스템은 이러한 전송 단절에 의한 TCP/IP 전송 속도 저하 문제를 해결하기 위해서 하위 노드가 상위 노드로부터 요청한 데이터 전송이 완료되지 않았더라도 그 이전에 “미리” 새로운 추가 경로 역할을 요청함으로써 slow-start에 의한 성능 저하 문제를 회피한다.

3.3.6 버퍼동작 : 재생중지 및 재시작

각 노드는 자신의 상위 노드에게서 전달받은 데이터를 버퍼에 저장한 후 일정량 B_i^k 가 넘었을 때 비디오 재생을 시작하며, 더 이상 재생 가능한 데이터가 없을 때 재생을 중단한다.

그런데 본 시스템에서 재생 가능한 데이터는 버퍼에 저장되어있는 데이터의 총량이 아니다. 앞에서 살펴본 바와 같이 노드의 세 가지 역할에 따라 삭제할 수 있는 데이터의 위치가 다르기 때문에 재생이 끝난 데이터일지라도 해당 역할이 수행되기 전에는 삭제할 수가 없다. 예

들 들어, 그림 7은 노드 A가 부모 노드 B에게 백업 역할을 요청하고 동시에 노드 C에게 속도 향상 요청을 한 경우 비디오 버퍼의 내용을 보인 것이다. 그림 7에서 보는 바와 같이 버퍼 내의 모든 데이터가 재생 가능한 것은 아니다.

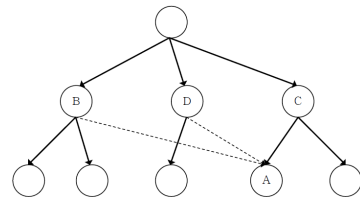


(그림 7) 노드 A의 재생 가능한 데이터 영역
(Figure 7) Data area of playable video data of node A

재생이 중단된 노드가 재생을 다시 시작하기 위해서는 기본 경로와 추가 경로에 연결된 노드들로부터 비디오 데이터를 전송받아 버퍼에 저장하고, 저장된 재생 가능한 데이터의 양이 임계값 B_i^k 를 넘을 때 재생을 다시 시작한다.

3.3.7 버퍼동작 : 속도 향상 요청과 백업 요청

속도 향상 요청은 그림 8과 같은 구조에서 노드 A의 데이터의 소모 비율이 기본 경로인 노드 C로부터의 수신율보다 떨어지는 등 노드 A의 버퍼에 저장된 재생 가능한 데이터의 양이 임계값 B_i^k 보다 작아질 때 이루어진다. 이 경우 노드 A는 부모 노드 C에게 데이터를 전송받고 있는 영역의 끝부분부터 일정량의 데이터를 속도 향상 요청으로 처리할 데이터로 결정한다. 그 뒤 결정된 부분의 정보를 추가 경로로 연결된 노드 B에게 전송하고, 노드 B는 전달 받은 정보를 자신의 B_i^{ss} 와 B_i^{sc} 에 저장하고 노드 A를 위한 속도 향상 역할을 수행한다.



(그림 8) 노드 A의 연결 구조
(Figure 8) The connection structure of node A

노드 A의 백업 요청의 시작은 다음과 같다. 노드 A에게 데이터를 전송 하는 부모노드 C가 갑자기 종료되었을

때 노드 A는 이를 인지하지 못하지만 재생 가능한 데이터의 양이 감소되는 결과를 낳는다. 그 결과 버퍼에 남은 양이 B_i^s 보다 작아질 때 노드 B에게 속도 향상 요청을 수행한다. 하지만 부모노드 C에게 요청된 부분의 전송이 완료되지 않았기 때문에 속도 향상 역할을 수행하는 노드 B로부터 받는 데이터는 재생이 불가능하게 된다. 결국 재생 가능한 데이터의 양이 B_i^b 보다 작아질 때 노드 A는 부모 노드 C가 서비스 단절 되었다고 판단하여 부모 노드 C와의 연결을 종료하고, 추가 경로로 연결된 노드 D에게 백업 역할을 요청한다. 백업 역할 요청할 때 노드 A는 전 부모 노드 C에게 요청한 데이터 부분 중에 전송이 완료되지 않은 부분을 노드 D에게 백업 요청하게 된다.

4. 실험 및 결과

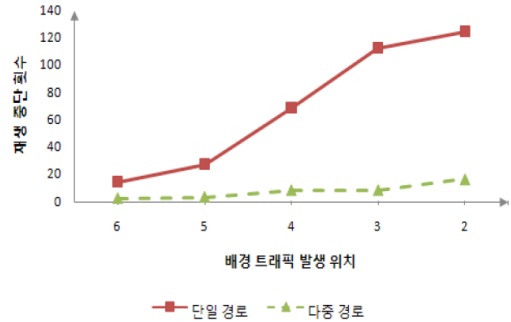
실험은 ns-2 시뮬레이터를 이용하였다[19]. 실험은 1개의 비디오 서버와 126 개의 노드로 이루어진 완전이진 트리를 사용하였으며 트리의 깊이는 6이다. 각 노드는 기본 경로이외에 추가 경로를 부모의 형제 노드에게 연결되어 있다고 가정하였다. 각 노드는 평균 1 Mbps 대역폭을 가지고 있으며, 배경 트래픽은 540 Kbps를 사용하였다. 추가 경로의 대역폭은 200 Kbps이다. 실험은 속도 향상 요청만 존재한다고 가정하였다. 서비스에 참여하는 노드의 버퍼는 1MB 크기를 가지고 있으며, 재생 시작 임계값은 그 절반인 500 KB를 가정하였다. 또 속도 향상 요청의 임계값은 360 KB로 설정하였다.

여러 개의 비디오를 사용하여 실험하였으나 결과가 비슷하게 나와 하나의 비디오 파일에 대한 결과만을 가지고 분석하였다. 실험에 사용된 비디오 파일은 평균 360 Kbps 비트율을 갖고 90000 비디오 프레임으로 구성되어 있으며 초당 프레임 수는 25이다.

실험의 목적은 추가 경로를 사용하였을 때 상위 노드에서 문제가 발생하더라도 하위 노드들에게 고수준의 서비스 품질을 제공하는지를 검증하는 것이다. 우리는 노드가 겪게 되는 재생 중단 횟수가 적으면 더 나은 서비스 품질을 제공 받았다고 정의한다. 또한, 재생 중단의 횟수가 동일한 경우 평균 서비스 단절 시간이 짧을수록 더 나은 서비스 품질을 받고 있다고 간주한다.

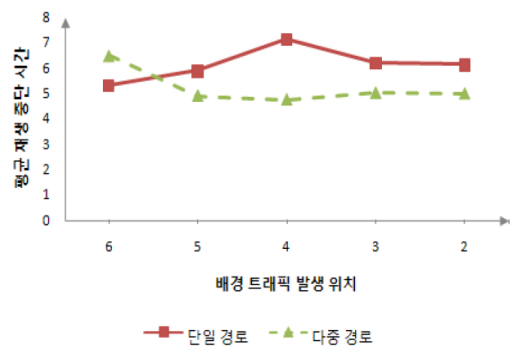
그림 9는 배경 트래픽 발생 위치에 따른 전체 재생 중단 횟수를 나타낸 것이다. 단일 경로에서 배경 트래픽의 발생 위치가 상위일수록 재생 중단 횟수가 급격히 증가하는 것에 비해서 본 연구에서 제안한 다중 경로를 사용하

였을 때에는 재생 중단 횟수가 조금씩 약간만 증가하는 것을 볼 수 있다.



(그림 9) 전체 시스템 내의 재생 중단 횟수
(Figure 9) The number of service interruption

그림 10는 그림 9의 실험을 평균 재생 중단 시간 관점에서 나타낸 것이다. 우리의 서비스 품질 관점에서는 그림 9의 서비스 중단 회수가 중요하지만 그림 10는 단일 경로와 다중 경로 방식의 차이점을 확인하는데 유용하다. 그림 10에서 대부분의 경우 다중 경로가 평균 서비스 중단 시간 측면에서도 더 우수하다. 그림 10에서 깊이가 6인 지점에서 배경 트래픽이 발생하는 경우 단일 경로 방식이 더 나은 것처럼 착각할 수 있다. 이것은 평균 비디오 재생 중단 시간을 측정할 때 전체 서비스 중단 시간을 서비스 중단된 노드의 수로 나누었기 때문이다. 다중 경로 방식의 경우 서비스 중단된 노드의 숫자가 적고 단일 경로 방식의 경우 서비스 중단 노드 숫자가 많기 때문에 발생하는 현상이다.



(그림 10) 평균 비디오 재생 중단 시간
(Figure 10) The average time interval of video playback stopping

5. 결론 및 향후 연구

본 논문은 하나의 서버와 다수개의 노드로 구성된 VOD 멀티미디어 서비스에서 고품질의 서비스를 제공하기 위한 TCP/IP 환경에서의 다중 경로를 통한 비디오 전송 시스템을 제안하였다. 제안하는 시스템은 릴레이 역할을 수행하는 노드에 문제가 발생하였을 때 그 영향이 하위에 연결된 노드에게 전달되어 재생이 중단되는 것을 방지하는데 목적이 있다. 이 목적을 위하여 백업 역할과 속도 향상 역할을 수행하는 추가 경로를 설정하는 방법을 제안하였고, 실험을 통하여 기존의 단일 경로만을 사용하여 전송하는 방법보다 제안하는 시스템을 사용하였을 때 재생 중단 횟수가 월등히 적다는 것을 보였고, 이를 통해 제안하는 시스템이 비디오 재생의 연속성에 있어서 우수함을 보였다.

향후 연구로 좀 더 다양한 환경에서 본 시스템이 갖고 있는 인자들을 최적으로 설정하는 방법에 대한 연구하고 노드들이 다양한 환경에서 비정상적인 종료를 할 때 본 연구의 성능을 평가할 예정이다.

참고 문헌 (Reference)

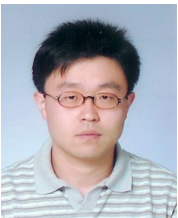
- [1] M. Hosseini, D. Ahmed, S. Shirmohammadi, and N. Georganas, "A Survey of Application-layer Multicast Protocols", IEEE Communications Surveys & Tutorials, Volume 9, Issue 3, Third Quarter 2007
- [2] D. Tran, K. Hua, and T. Do, "A peer-to-peer architecture for media streaming", IEEE Journal on Selected Areas in Communications, Jan. 2004
- [3] S. Banerjee, B. Bhattacharjee, and C. Kommareddy, "Scalable Application Layer Multicast", Proc. of the ACM SIGCOMM, Pittsburgh, PA, AUG. 2002
- [4] J. Liu, S. Rao, Li and H. Zhang, "Opportunities and Challenges of Peer-to-Peer Internet Video Broadcast", (invited) Proc. of IEEE, Special Issue on Recent Advances in Distributed multimedia Communications, 2007
- [5] W. Ken Yiu, X. Jin, and S. Gary Chan, "Challenges and Approaches in Large-Scale P2P Media Streaming", Multimedia, IEEE, April-June 2007
- [6] X. Hei, Y. Liu, and K. Ross, "IPTV over P2P Streaming Networks : The Mesh-Pull Approach", Communications Magazine, IEEE, Feb. 2008
- [7] B. Li, and H. Yin, "Peer-to-Peer Live Video Streaming on the Internet: Issues, Existing Approaches, and Challenges", Communications magazine, IEEE. June 2007
- [8] J. Moi, D. Epema, and H. Sips, "The Orchard Algorithm: Building Multicast Trees for P2P Video Multicasting Without Free-Riding", IEEE Transactions on Multimedia, Dec. 2007
- [9] T. Small, B. Li, and B. Liang, "Outreach: Peer-to-Peer Topology Construction towards Minimized Server Bandwidth Costs", IEEE Journal on Selected Areas in Communications, Jan 2007
- [10] E. Mykoniati, R. Landa, S. Spirou, R. Clegg, L. Latif, D. Griffin, and M. Rio, "Scalable Peer-to-Peer Streaming for Live Entertainment Content", Communications Magazine, IEEE, Dec. 2008
- [11] M. Castro, P. Druschel, A. Kermarrec, A. Nadi, A. Rowstron, and A. Singh, "SplitStream: high-bandwidth multicast in cooperative environments", SOSP '03, Proc. of the 9th ACM symposium on Operating systems principles(SOSP '03), Dec. 2003
- [12] D. Purandare, and R. Guha, "An Alliance Based Peering Scheme for P2P Live Media Streaming", IEEE Transactions on Multimedia, Dec. 2007
- [13] A. Habib, and J. Chuang, "Service differentiated peer selection: an incentive mechanism for peer-to-peer media streaming", IEEE Transactions on Multimedia, June 2006
- [14] X. Liao, J. Hai, Y. Liu, and L. Ni, "Scalable Live Streaming Service Based on Interoverlay Optimization", IEEE Transactions on Parallel and Distributed Systems, Dec. 2007
- [15] A. Raghuv eer, E. Kusmerek, and D. Du, "A Network-Aware Approach for Video and Metadata Streaming", IEEE Transactions on Circuits and Systems for video technology, Aug. 2007
- [16] P. Frossard, J. Martine, and M. Civanlar, "Media Streaming With Network Diversity", (invited) Proc. of the IEEE, Jan. 2008
- [17] J. Zhang, Y. Zhao, Y. Liu, C. Chen, "Enabling P2P One-View Multiparty Video Conferencing", IEEE Transactions on Parallel and Distributed Systems, Vol. 25, no. 1, pp. 73-82. 2014

- [18] W. Wu, R.T.B. Ma, J.C.S. Lui, "Distributed Caching via Rewarding: An Incentive Scheme Design in P2P-VoD Systems", IEEE Transactions on Parallel and Distributed Systems, Vol. 25, no. 3, pp. 612-621. 2014
- [19] The Network Simulator - ns-2,
<http://www.isi.edu/nsnam/ns/>

● 저 자 소 개 ●



김 남 수 (Nam-su Kim)
1998년 전북대학교 컴퓨터공학부 졸업(학사)
2010년 전북대학교 대학원 컴퓨터공학과 졸업(석사)
2010년 ~ 현재 LG전자 MC 사업부 연구원
관심분야 : Network topology, Database etc.
E-mail : whstar83@gmail.com



이 중 열 (Jong-yeol Lee)
1993년 한국과학기술원 전기및전자공학과 졸업(학사)
1996년 한국과학기술원 전기및전자공학과 졸업(석사)
2002년 한국과학기술원 전기및전자공학과 졸업(박사)
2004~현재 전북대학교 전자공학부 교수
관심분야 : Embedded software, Compiler
E-mail : jong@jbnu.ac.kr



편 기 현 (Kihyun Pyun)
1995년 인하대학교 전자계산공학과 졸업(학사)
1997년 KAIST 전산학과 졸업(석사)
2002년 KAIST 전산학과 졸업(박사)
2003년 KAIST 전기및전자공학 박사후 연구원
2004년 ~현재 전북대학교 컴퓨터공학부 부교수
관심분야 : 차세대 인터넷 서비스, 유무선 네트워크, 시스템 소프트웨어, 모바일 소프트웨어
E-mail : khpyun@chonbuk.ac.kr