

P2P 네트워크에서의 오디오 스트리밍 시스템

¹정의현, ²김성진⁰

¹안양대학교 이공대학 디지털미디어공학전공

¹jung@anyang.ac.kr

²한양여자대학 인터넷정보과

²sjkim@hywoman.ac.kr

An Audio Streaming System on the P2P Network

¹Eui-Hyun Jung and ²Sung-Jin Kim⁰

¹Dept. of Digital Media, Anyang University

²Dept. of Internet Information, Hanyang Women's College

요 약

P2P 컴퓨팅은 자원 공유와 분산 컴퓨팅의 대안적 구조로 많은 주목을 받아 왔으며, 여러 분야에 응용되고 있다. 그러나, P2P 환경에서의 멀티미디어 스트리밍은 피어간의 낮은 대역폭과 불안정한 연결이라는 구조적인 문제 때문에 상대적으로 주목받지 못하였다. 본 논문에서는 이의 해결을 위해서 다중 피어 결합 방식을 제안하고, 이를 이용한 오디오 스트리밍 시스템의 설계 및 구현에 관하여 논한다. 다중 피어 결합 방식은 데이터의 부분들을 여러 피어로부터 동시에 전송받아 가상적으로 높은 대역폭을 확보하는 방식이며, 이 방식을 이용하여 기존 P2P 환경하에서 실현 가능한 오디오 스트리밍 시스템의 프로토타입을 설계하였다. 설계된 프로토타입은 JXTA 기반하에서 구현되었으며, JavaSound를 이용하여 오디오 브라우징을 처리하였다.

1. 서 론

인터넷이 멀티미디어 데이터의 주요한 매체로 각광받으면서, 인터넷상에서 효율적으로 멀티미디어를 재생 및 분배하는 것에 대한 연구들이 활발하다[1,2]. 네트워크를 통한 멀티미디어의 분배(Multimedia Retrieval)는 크게 다운로드 모드(Download)와 스트리밍(Streaming) 모드로 나뉘어질 수 있다. 다운로드 모드는 원격(remote) 머신에 위치한 데이터 전체를 모두 전송받은 후에 로컬(local) 머신에서 재생하는 방식이다. 이 방식은 대역폭의 제한 없이 안전한 데이터 전송을 보장하는데 비해, 사용자가 원하는 시점에 해당 멀티미디어 데이터를 사용할 수 없는 단점을 갖고 있다.

이에 비해 스트리밍 모드는 원격 머신의 일부분의 데이터를 전송받은 후에 로컬 머신에서 바로 재생을 하고, 필요한 데이터를 원격 머신에서 계속 전송받는 방식이다. 스트리밍 모드는 데이터의 즉각적인 사용이 가능하다는 장점 때문에 다운로드 모드에 비해서 높은 선호도를 갖는다[4]. 그러나, 이 모드는 데이터에 대한 안정적인 접근과 전송 세션이 높은 대역폭을 요구하게 되며, 기술적으로 해결해야 하는 문제들이 상당히 많은 편이다[3].

이러한 요구조건 때문에 현재 대부분의 스트리밍 분배는 클라이언트/서버 방식을 이용하고 있으며, 높은 대역폭을 제공하는 전용의 미디어 서버(Media Server)를 이용하는 경우가 대부분이다. 그러나, 이러한 클라이언트/서버 구조는 기존의 P2P 커뮤니티에서 지적하는 문제를 그대로 안고 있다. 첫째, 미디어 서버의 저장 용량에 한계가 있기 때문에, 어떤 미디어 서버도 사용자가 원하는 데이터를 모두 제공할 수 없다. 둘째, 여러 사용자들이 동시에 하나의 서버에 접속하는 경우, 부하 집중의 문제가 발생하게 된다.

그러나 이러한 문제를 해결하기 위해서 P2P 구조를 단순히 적용하는 것은 다른 문제를 야기하게 된다. 스트리밍 분배는 실시간 재생을 위해 시간당 정해진 대역폭이 제공되어야 한다. 그러나, P2P에서의 피어들은 다른 피어들에게 서비스를 제공하기 위해 인터넷을 사용하는 것이 아니므로, 한정된 대역폭을 가지게 된다. 또한, 세션중에 데이터를 전송하는 피어가 임의로 접속을 해제하는 경우, 스트리밍 분배를 받는 피어는 더 이상 미디어 재생을 할 수 없게 된다. 따라서, 단순히 피어간의 단단단(End-to-End) 접속을 통한 스트리밍 분배는 효율적인 대안이 될 수 없다.

본 논문에서는 이를 해결하기 위하여 다중 피어 결합(Multi-Peer Binding) 방식을 이용한 오디오 스트리밍 시

시스템을 제안한다. 다중 피어 결합은 기존의 단대단 접속과 달리 데이터를 부분별로 여러 피어로부터 동시에 전송받는 방식이다. 이 방식을 이용하면 가상적으로 대역폭의 확장이 가능하며, 결합된 일부 피어가 접속을 해제해도 해당 데이터의 중복(replication)을 통해서 안정적인 스트리밍을 제공할 수 있게 된다.

제안된 시스템은 다중 피어 결합을 제공하기 위해, 미디어 데이터를 정해진 단위 크기로 청크(chunk)로 분할하고, 청크의 유일성을 보장하기 위해 데이터를 MD5[4] 알고리즘으로 128비트의 해쉬를 제공하게 된다. 이렇게 만들어진 해쉬를 이용하여 여러 피어에 분할된 청크들을 유일한 값으로 접근할 수 있게 된다. 이러한 방식을 이용하여, P2P상에서도 효과적인 스트리밍이 가능함을 보였다.

본 논문의 구성은 다음과 같다. 2장에서는 다중 피어 결합의 구조에 대해서 설명한다. 3장에서는 다중 피어 결합 구조를 적용한 오디오 스트리밍 시스템의 구조와 구현 방식에 대해서 논한다. 4장에서는 실제 구현과 테스트에 관해 설명하고, 5장에서 결론을 맺는다.

2. 다중 피어 결합

2.1 다중 피어 결합의 구조

다중 피어 결합은 스트리밍에 필요한 데이터를 단대단으로 접속된 1개의 종단 시스템으로부터 데이터를 받는 것이 아니라, 그림 1처럼 데이터를 여러 조각으로 나누어 여러 호스트에게 동시에 전송받는 것이다.

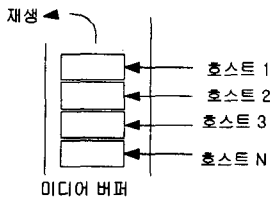


그림 1. 다중 피어로부터의 데이터 전송

단대단 접속의 대역폭은 양단간의 세션 대역폭에 종속되어 있지만, 다중 피어 결합을 이용하면, 여러 종단 시스템으로부터 데이터를 받기 때문에, 가상적인 대역폭의 확장을 할 수 있게 된다. 이러한 구조는 e-donkey[7]와 같은 P2P 자원 공유 응용에서 사용되고 있으나, 자원 공유에 효율적인 방식으로 구성되어 있다는 문제점을 갖고 있다. 이것은 자원 공유 입장에서 실시간적인 데이터의 요구사항을 고려하지 않아서 설계되었기 때문에, 스트리밍에 적절한 데이터의 크기와 실시간적으로 필요한 데이터의 우선 순위를 고려한 알고리즘이 결여되어 있는 상황이라, 직접적인 적용은 불가능하다.

2.2 데이터 분할방식

하나의 미디어 데이터를 여러 개의 조각인 청크로 분할하였을 때, 이 청크들은 여러 시스템에 분산되어 수용되기 때문에, 청크들이 해당 데이터의 일부라는 것을 확인하는 방법이 필요하다. 본 논문에서는 청크를 해쉬하는 방법이용하여 이를 해결하였다. MD5 해쉬 방식은 일방

향 함수를 이용하여 데이터로부터 유일한 값을 만들어 낸다[6]. 따라서 청크가 여러 피어들에 분산되어 있다하더라도, 청크의 해쉬를 키로 넘겨줌으로써, 원하는 청크를 얻을 수 있게 된다. 또한, 파일 이름을 변경하는 상황이 있어도, 해당 파일의 데이터를 변경하지 않는 한, 해쉬가 변화되지 않으므로 이름에 종속적인 문제를 해결할 수 있게 된다. 본 논문에서는 파일을 지시하기 위한 파일해쉬와 조각들을 지시하기 위한 청크해쉬로 나눈다.

이 구조를 구현하기 위해, 그림 2에서 볼 수 있는 것처럼, 미디어 데이터를 64Kbyte의 크기로 분할하였다. 분할된 파일들은 각각 MD5 알고리즘을 이용하여 청크해쉬를 얻게 된다. 이렇게 얻어진 청크해쉬를 다시 데이터를 이용해서, 해당 미디어 데이터의 파일 해쉬를 얻는 구조를 택하였다.

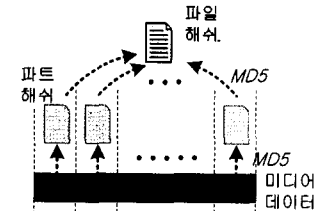


그림 2. 해쉬 생성의 구조

3. 시스템 설계

3.1 시스템 구조

다중 피어 결합을 이용한 오디오 스트리밍 시스템은 P2P 네트워크에 두 개의 네트워크 연결과 다운로드 매니저를 갖는 응용 형태이다. 그림 3에서 볼 수 있는 것처럼, 브로드캐스트(broadcast) 기능을 제공하는 Propagate Pipe연결로 필요한 미디어 파일의 검색을 요구하게 된다. 이 요구에 대한 응답은 유니캐스트(unicast) 연결을 제공하는 Unicast Pipe를 통해 넘겨받게 되며, 실제적인 데이터의 전송은 다운로드 매니저를 통한 소켓 접속을 통해서 수행되게 된다.

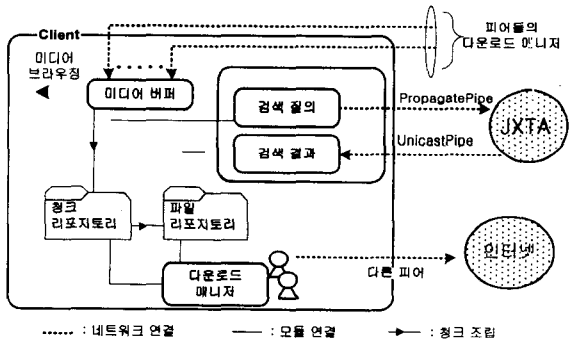


그림 3. 클라이언트의 구조

3.2 세션 바인딩 프로토콜

사용자가 필요한 오디오 데이터에 접근하기 위해서는 JXTA 네트워크에 접속된 피어의 Broadcast Pipe를 이용해서 오디오 데이터의 이름과 관련된 키워드를 보낸다. 이 키워드는 접속한 모든 피어에게 전달된다. 이 키워드

를 넘겨받은 피어들은 키워드를 포함한 파일 이름과 파일 해쉬를 보내게 된다. 이렇게 보내진 파일 이름들중에서, 사용자가 선택을 하게 되면, 이번에는 파일 해쉬를 다시 브로드캐스트하게 된다. 이것은 다른 이름을 가진 같은 파일을 지정하기 위해서이다. 이 요구에 대해서는 해당 파일이나 해당 파일의 청크를 보유한 피어들이 가지고 있는 청크 해쉬들을 응답으로 보내게 된다. 이 청크 해쉬들은 앞에서 설명한 것처럼, 청크를 지정하는 유일한 값을 갖게 된다. 이 흐름도는 그림 4에 설명되어 있다.

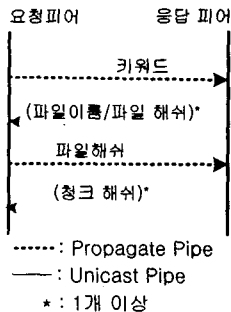


그림 4. 청크 해쉬를 얻는 프로토콜 흐름도

이렇게 얻어진 청크들은 각 청크의 순서별(order)로 파일을 요청한 피어에서 연결 리스트로 저장된다. 이 연결 리스트는 미디어 브라우징에 필요한 순서별로 필요한 청크들의 정보를 저장하고, 노드별로 청크를 갖고 있는 모든 피어에 대한 정보를 보유하게 된다. 이런 구조를 갖는 이유는 청크의 전송 도중에 해당 피어의 접속이 해제되면, 같은 청크를 갖고 있는 피어로 접속하여 청크의 재전송을 하기 위해서이다. 이러한 청크의 중복 구조를 이용함으로써, 잦은 접속 해제가 일어나는 피어 환경에서 안정적인 데이터 전송을 보장받게 된다.

4. 구현 및 평가

4.1. 시스템 구현

설계된 시스템은 JXTA를 하부 네트워크로 구현되었다. 본 논문의 주요한 목표가 P2P 환경에서의 오디오 스트리밍에 관한 내용으로 하부 네트워크를 새로 구현하는 것 보다는 기존의 P2P를 제공하는 JXTA[5]가 프로토타입의 테스트에 적합하다고 판단하였다. 오디오 브라우징은 JavaSound 기능을 이용하였으며, JavaSound 하에서 MP3의 지원이 가능하도록 모듈을 추가하였으며, 미디어 버퍼의 구현은 내부에서는 분할된 링버퍼의 형태를 갖되, 외부에는 직렬적인 접근이 가능하도록 입력 스트림(Input Stream) 구조를 갖도록 구현하였다. 실제 구현된 응용은 그림 5와 같다.

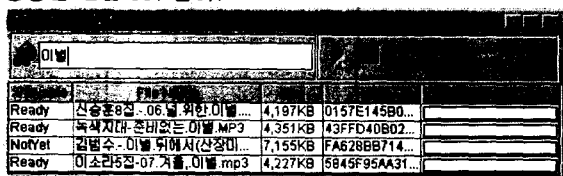


그림 5. 응용의 모습

4.2. 실험 및 평가

구현된 시스템을 실험하기 위하여, LAN 환경에서 5대의 컴퓨터에 해당 시스템을 설치하였다. 설치된 시스템의 두 지점에서 4.2MByte의 같은 오디오 샘플을 위치하되, 이름을 달리하였다. 두 지점에서 위치한 오디오 샘플을 다른 두 지점에서 처음에 요구하였고, 일정 시간 경과후 나머지 지점에서 해당 데이터를 요구하였다. 이렇게 되었을 때, 마지막으로 데이터를 요구한 지점에서의 청크에 대한 연결 리스트는 표 1과 같은 구조를 나타내었다.

표 1. 연결 리스트

청크 ID	청크 보유 호스트	청크의 우선순위 호스트
1	4	203.229.203.8
2	4	203.229.203.8
40	2	203.229.203.9
41	2	203.229.203.9

표 1에서 볼 수 있는 것은 앞부분의 청크들은 이미 전송되어 4개의 호스트가 모두 가지고 있으나, 중간 부분들은 첫번째로 전송을 요구한 피어들이 아직 보유하고 있지 않으므로, 청크 보유 호스트의 수가 2개로 나왔다. 그리고, 청크의 우선순위 호스트는 청크에 대한 응답이 오는 순위를 우선순위 호스트로 지정하였다. 따라서, 파일 청크를 보유한 모든 호스트들에서 청크 정보를 얻어올 수 있음을 확인할 수 있었다. 이 결과는 논문에서 설계된 대로 여러 피어가 결합하여 해당 피어에게 데이터를 제공할 수 있음을 보여준다.

5. 결론

기존의 P2P 환경에서의 스트리밍은 여러 장점에도 불구하고 불안정한 피어간의 접속과 낮은 대역폭으로 인하여 현실화되기 어려웠다. 본 논문에서는 다중 피어 결합 방식을 이용해서, 가상적으로 대역폭을 확장하고 전송의 안정성을 높이는 방식을 제안하였다. 이 방식을 적용한 오디오 스트리밍 시스템은 기존의 P2P 환경의 수정없이 효율적이고 안정적인 스트리밍 기능을 제공하는 구조로 기대된다. 향후에는 다른 미디어 스트리밍에 대한 연구가 진행될 예정이다.

참고문헌

- [1]Abdel-Baki.N, Aumann.B, Grossman.H.P. "Analyzing Multimedia Streaming in a Distributed Environment," ECUMN 2002, pp.56-63, Apr. 2002
- [2]Gebhard.H, Lindner.L, "Virtual Interface Broadcasting," IEEE Communication Magazine, vol.39, no.6, pp.182-188, Jun. 2001
- [3]Yuqing.S, Mielke.M, Aidong.Z, "NetMedia: synchronized streaming of multimedia presentations in distributed environments," Multimedia Computing and Systems, vol.2, pp.585-590, Jun.1999
- [4]Rivest, R.L., "The MD5 message-digest algorithm," RFC-1321, 1992