

물리적 토폴로지를 고려한 P2P환경에서 다중 디스크립션 코딩을 이용한 미디어 스트리밍 기법

강동훈[○], 박성환, 한동윤, 차봉관, 김경석[†]
 부산대학교 컴퓨터공학과[○], 부산대학교 정보컴퓨터공학부[†]
 {dhkang[○], shpark, dyhan, bgcha, gimgs0[†]}@asadal.pusan.ac.kr

Media Streaming Method that Use Multiple Description Coding in P2P Environment that Consider Physical Topology

Donghun Kang[○], Sunghwan Park, Dongyun Han, Bonggwon Cha, Kyongsok Kim[†]
 Dept. of Computer Engineering, Pusan National University,
 Dept. of Computer Science & Engineering, Pusan National University[†]

요 약

최근 초고속 인터넷이 급속히 보급되고 시스템 및 네트워크 성능의 향상으로 미디어 콘텐츠에 대한 전달을 효율적으로 제공하는 방안이 연구되고 있다. 네트워크의 성능이 향상되었다고 하더라도 미디어 콘텐츠를 스트리밍 하는 것은 인터넷 트래픽의 큰 부분을 차지한다. 그래서 이러한 콘텐츠를 효율적으로 분산시키고 서비스하기 위해서 논문에서는 P2P(Peer-to-Peer) 오버레이 네트워크를 이용한다. 스트리밍 서비스에서 반드시 보장해야 하는 것은 서비스가 끊김없이 이루어져야 한다는 것이다. 이 논문에서는 하나의 노드가 콘텐츠를 전체를 스트리밍 서비스를 하는 것이 아니라 다중 디스크립션 코딩을 활용하여 여러 노드가 스트리밍 서비스를 제공함으로써 특정 노드에 문제가 발생하더라도 스트리밍 서비스가 계속해서 이루어지는 방법을 제안하였다.

1. 서 론

최근 몇 년 동안 네트워크 트래픽의 아주 큰 부분을 차지하는 미디어 파일을 분산시키는 연구가 활발히 진행되고 있다. 네트워크의 성능이 많이 향상되었다고는 하지만 아직도 미디어 콘텐츠에 대한 트래픽은 여전히 부담이 되고 있다. 앞으로 더욱더 많은 사람들이 대용량 미디어 콘텐츠에 대한 서비스를 요구할 것이다. 우리는 이러한 문제를 해결하기 위해서 물리적 토폴로지를 고려한 P2P 환경을 기반으로 다중 디스크립션 코딩(MDC:Multiple Description Coding)을 이용한 미디어 스트리밍 서비스를 제안한다. 물리적으로 가까운 노드로부터 서비스를 받게 되었을 경우 빠른 시간에 받을 수 있으며 전체 네트워크의 트래픽을 줄일 수 있다. 그리고 다중 디스크립션 코딩을 이용함으로써 일부 디스크립션에 대한 서비스가 중단되더라도 서비스를 계속 할 수 있다.

본 논문은 다음과 같이 구성된다. 2장은 본 논문의 관련 연구인 Chord와 TB-Chord, 그리고 MDC에 대해서 살펴보고, 3장에서는 제안하는 스트리밍 기법에 대해서 설명한다. 4장에서는 스트리밍 기법이 실제로 어떻게 동작하는지에 대해서 설명하고, 마지막으로 5장에서는 결론과 향후 과제를 보여준다.

2. 관련 연구

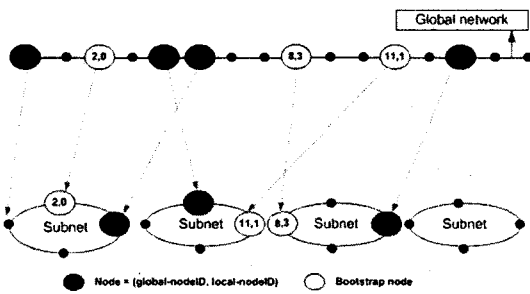
2.1 TB-Chord(Topology Based Chord)

TB-Chord[1]는 시스템 성능향상을 위한 토폴로지 기반의 P2P 시스템이다. P2P 오버레이 네트워크에서는

노드의 IP주소를 해쉬 함수를 이용하여 해쉬하여 나온 결과 값에 따라 시스템에 참여되므로 노드들 사이의 물리적 거리를 고려하지 않는다. 즉 P2P 시스템에서 서로 이웃한 노드라도 실제 물리적 네트워크에서는 여러 네트워크를 경유할 수 있기 때문에 물리적인 레이턴시(Latency)가 클 수 있다. 그래서 실제 탐색 과정에서는 홉(hop)을 몇 번 뛰지 않더라도 레이턴시가 클 수 있기 때문에 시스템 성능 저하 원인이 될 수 있다. TB-Chord에서는 이런 문제점을 해결하기 위해 토폴로지(Topology)를 고려한 계층구조인 TB-Chord를 제안하고 효율적인 저장 공간(Storage)을 위한 복제(Replication) 전략을 제안하였다. 물리적으로 가까운 노드들을 하나의 서브넷(subnet)으로 구성하고 데이터의 복제본(replication)을 자신의 서브넷에 두어 탐색할 때 물리적인 레이턴시를 줄이며 효율적으로 탐색을 할 수 있게 만들었다.

2.1.1 TB-Chord의 시스템 구조

(그림 1)은 TB-Chord의 시스템 구조이다. TB-Chord는 모든 노드로 구성된 글로벌 네트워크와 토폴로지 정보를 이용하여 구성된 여러 개의 서브넷으로 구성된 계층적 오버레이 네트워크이다. 각 네트워크는 Chord 시스템으로 구성되어 있다. 각 노드는 IP주소를 2개의 해쉬 함수(h1, h2)를 이용하여 Global-nodeID와 Local-nodeID를 구한다. Global-nodeID는 글로벌 네트워크에서 자원 탐색하거나 노드가 참여(join)될 때 사용되는 식별자이다. Local-nodeID는 서브넷에서 자원 탐색하거나 노드가 참여될 때 사용되는 식별자이다. 각 서브넷에서는 자신의



[그림 1] TR-Chord의 구조

서브넷을 관리하는 슈퍼피어가 반드시 한 개 존재한다. 각 노드는 글로벌 네트워크에서 효율적으로 자원을 탐색하기 위해 자신의 서브넷에 존재하는 노드들의 Global-nodeID를 유지한다.

2.1.2 노드의 참여

노드가 참여할 때, 먼저 Global-nodeID를 가지고 글로벌에 참여한 후 서브넷의 슈퍼피어에게 ping을 보내 거리가 임계값(threshold)이하 일 때 해당 서브넷에 참여하고, 그렇지 않을 경우 바로 옆의 슈퍼피어로 이동하면서 임계값 이하인 슈퍼피어를 찾아 해당 서브넷에 참여하는 구조이다.

2.1.3 데이터 탐색

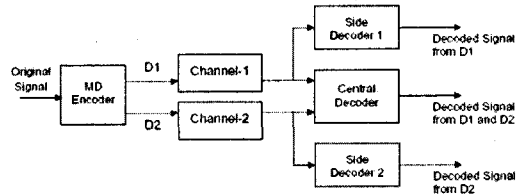
먼저 Local-nodeID를 가지고 자신의 서브넷에서 찾은 다음 없을 경우 Global-nodeID를 가지고 글로벌에서 찾는다. 글로벌에서 찾은 콘텐츠의 경우 검색의 효율을 위해 해당 콘텐츠를 자신의 서브넷의 노드에게 복제한다.

2.2 MDC(Mutiple Description Coding)

일반적인 미디어 코더가 하나의 비트 스트림을 생성하는 것과는 달리 계층화 코딩(LC : Layered Coding)과 MDC는 하나의 미디어 소스를 두 개 이상의 비트 스트림으로 만들어 피할 수 없는 전송에러에 강인하게 하는 소스코딩 방법이다[2].

MDC는 벨 연구소에서 최초로 개발된 방법이다[3]. 그 후에 소스 신호에 대한 MDC는 정보 이론 분야와 신호 처리 분야에서 널리 연구되어 왔다[4,5]. 신호 처리 분야에서 MDC의 개념은 음성 압축에 차례로 적용되었으며[6] 그 후에 영상과 동영상 압축에 차례로 적용되었다[7,8]. MDC는 하나의 스트림을 여러 개의 분리된 비트 스트림으로 코딩하는 것을 의미한다. 즉, 하나의 스트림을 여러 k개의 스트림으로 분할하고 각 스트림에는 자신의 정보와 다른 스트림에 관한 부가 정보를 추가하여 어느 하나의 스트림을 잃더라도 나머지 스트림으로 복구할 수 있고 모두를 수신하게 되면 송신 시와 똑같은 품질의 콘텐츠를 생성할 수 있는 코딩 기술이다[9]. 이것은 계층 코딩과는 달리 각 비트 스트림은 서로 독립적이며 어느 것이 더 중요한 것이 없이 각각 똑같은 중요도를 가지며 어떤 스트림을 수신하는 재생이 가능하고 수신되는 스트림이 많을수록 재생 품질이 좋아지는 장점이 있다[10].

(그림 2)는 2개의 디스크립션으로 코딩되는 시스템을 나타낸다. MDC 인코더는 2개의 디스크립션을 생성하고 이들을 두 개의 독립적인 채널을 통해서 전송한다. 만일



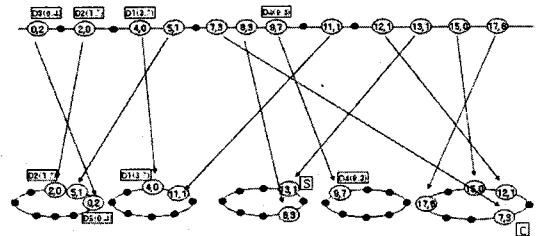
[그림 2] 2개의 디스크립션으로 구성된 MDC시스템[11]

두 채널 모두에서 에러가 발생하지 않은 경우 중앙 디코더(central decoder)는 손상되지 않은 두 개의 디스크립션을 전송받고 이를 하나의 고품질 신호로 재구성하게 된다. 만일 하나의 채널만이 에러가 발생하지 않은 경우에는 주변 디코더(side decoder)가 하나의 저품질이지만 수용가능한 신호로 재구성하게 된다[11].

3. 제안하는 스트리밍 기법

3.1 스트리밍 서비스의 기본 동작

- 각 콘텐츠는 n개의 디스크립션으로 나누어져서 해쉬가 되어 해당하는 노드에서 관리되어 진다.
- 해당 서브넷에서 어떤 노드가 콘텐츠에 대해서 스트리밍 서비스를 요청했을 때, 3.3절의 정책에 따라 프리패치나 복제시킨다.
- 초기에 글로벌 네트워크로부터 가장 느리게 다운로드되는 디스크립션은 서브넷에 무조건 복제시킨다.
- 각 서브넷에서 각 디스크립션을 관리하는 노드들은 자신이 관리하는 디스크립션의 인기도를 체크해서 추가적으로 복제시킨다.



[그림 3] 시스템 구조

3.2 참여와 이탈시의 스트리밍 서비스

3.2.1 참여

- 노드가 참여하면 자신에 관리해야하는 디스크립션을 자신의 successor에게 요청한다. successor는 자신이 앞 노드를 말한다. (그림 3)에서 (7,3)노드의 successor는 (17,6)노드이다.
- 새로 참여한 노드가 자신이 관리해야하는 디스크립션에 대해 successor로부터 다운로드 받는 도중 해당 디스크립션에 대한 서비스 요청이 왔을 시에는 successor가 서비스를 하고 다운로드를 완료한 이후의 시점부터는 자신이 서비스를 한다.

3.2.2 이탈

- ① 이탈 사실을 알리고 떠날 때
 - 자신이 관리하는 디스크립션들을 successor에게 넘겨주고 떠난다.
- ② 이탈 사실을 알리지 않고 떠날 때
 - 서버넷 안의 노드에게서 서비스를 받고 있는 경우 글로벌로 넘어가서 서비스를 받는다. 본 논문에서는 successor에 복제 정책을 사용하지 않는다. 그렇기 때문에 글로벌에서 다운로드 받는 도중 이런 일이 생기면 원본 콘텐츠가 없어진 것이므로 나머지 디스크립션을 다운로드 받는 것으로 만족해야 한다.

3.3 각 디스크립션의 도착시간에 따른 동기화 처리

- ① 스트리밍 서비스를 받고 있을 때, 만약 특정 디스크립션이 외부로부터 다운로드 되는 시간이 너무 느려 의미가 없다고 생각 할 때는 일부분을 서버넷에 프리패치 하거나 복제시킨다. 이렇게 함으로써 다음번에 서버넷 안의 노드로 부터 요청이 왔을 때, 글로벌로부터 받지 않고 로컬에서 서비스를 받게 된다.
- ② 초기에 다운로드 받을 때, 각 디스크립션 별로 실행 가능한 최소의 비트율과 실제 네트워크 비트율을 계산해서 실행이 가능하지 않은 서비스에 대해서 프리패치와 복제 정책을 사용하여 해당 서버넷안에서 다음 서비스에서는 정상적인 서비스가 가능하도록 한다.

4. 동작 시나리오

지금부터는 우리가 제안한 시스템이 실제로 어떻게 동작하는지에 대해서 살펴보겠다. (그림 3)는 우리가 제안한 시스템의 전체적인 구조이다. 시나리오에서는 디스크립션의 개수를 4개로 하였다.

4.1 서비스 시나리오

(그림 3)에서 사용하는 용어에 대한 정의를 하겠다.

- S : 콘텐츠의 원본을 가지고 있는 노드
- C : 콘텐츠에 대한 스트리밍 서비스를 원하는 노드
- D1~D4 : S가 가지고 있는 콘텐츠를 4개의 디스크립션으로 인코딩한 것

① S노드가 '수퍼맨'이라는 콘텐츠를 가지고 참여한다. S노드는 '수퍼맨'을 4개의 디스크립션으로 인코딩한다. 각 디스크립션은 '수퍼맨_1', '수퍼맨_2', '수퍼맨_3', '수퍼맨_4'로 관리되며 각 파일명을 두 개의 해쉬 함수인 h1과 h2로 해쉬하여 글로벌에서 관리되어 진다. (그림 3)에서 D1은 '수퍼맨_1', D2는 '수퍼맨_2', D3는 '수퍼맨_3', D4는 '수퍼맨_4'로 가정하겠다.

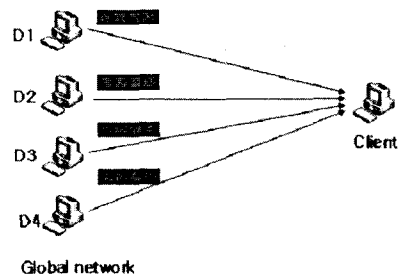
② C노드는 '수퍼맨' 콘텐츠를 스트리밍 서비스 받기 원

한다. 4개의 디스크립션을 h1과 h2로 해쉬한 다음 자신의 서버넷에서 찾는다. 서버넷에 해당 콘텐츠에 대한 디스크립션을 관리하는 노드가 모두 없기 때문에 글로벌에서 찾는다. 글로벌에서 디스크립션을 관리하는 노드를 찾은 후 스트리밍 서비스를 받는다.

③ 스트리밍 서비스를 받고 나서 4개의 디스크립션 중에서 전송 속도가 가장 낮은 것에 대해서는 서버넷에 저장 시켜놓는다. 만약 D3가 가장 다운로드 시간이 오래 걸린다면 해당 디스크립션을 자신의 서버넷에 저장시킨다. D3의 로컬 해쉬 값이 40이므로 로컬에서는 (17,6)노드가 관리할 것이다.

④ 시간이 지난 후 만약 (12,1)노드가 '수퍼맨'에 대해서 스트리밍 서비스를 받기 원할 경우 로컬에서 D3를 받고 D1, D2, D4에 대해서는 4.2의 정책에 따라 프리패치 후 스트리밍 서비스를 받는다.

⑤ 인기도에 따라서 추가적으로 D1, D2, D4도 로컬로 복제한다. 인기도는 로컬에서 디스크립션을 관리하는 각 노드들이 체크한다. 인기도가 높아지는 콘텐츠에 대해서는 D1, D2, D4 순으로 로컬에 복제되어 진다.



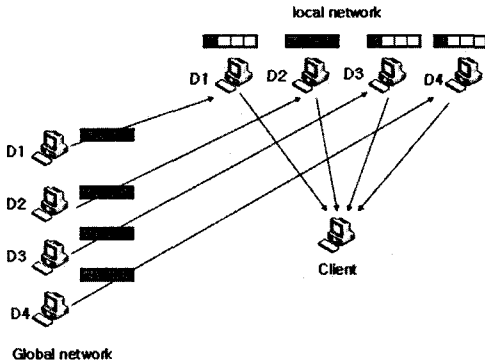
[그림4]

4.2 제안한 시스템의 문제점에 따른 해결 방안

(그림 4)과 같이 클라이언트는 글로벌로부터 모든 디스크립션을 받을 경우 발생하는 문제점을 살펴보자. D1~D4들 중 일부의 디스크립션의 다운로드 받는 속도가 느려 제대로 된 서비스를 받기 힘들 경우를 고려해보자. 만약 4개의 디스크립션의 비트율을 계산해서 가장 속도가 좋은 디스크립션은 일단 다운로드 받아 스트리밍 서비스를 한다. 그리고 다운로드 속도가 느린 디스크립션에 대해서 비트율에 따라 디스크립션의 일부분을 로컬에 프리패치시켜 놓는다. 만약 모든 디스크립션이 느리게 다운로드 될 경우 현재 클라이언트에게는 정상적인 서비스가 어려울 것이다. 가장 느리게 도착하는 디스크립션의 전체를 로컬에 복사하고 나머지 3개의 디스크립션은 비트율에 따라 프리패치시킨다. 결국 (그림 5)와 같이 된다. 하지만 모든 디스크립션이 느릴 가능성은 높지 않다. 또한, 이런 경우가 발생한다고 하더라도 서버넷 안의 다음 요청 노드에 대해서 제대로 된 서비스를 할 수 있다.

4.3 프리패치 사이즈 결정

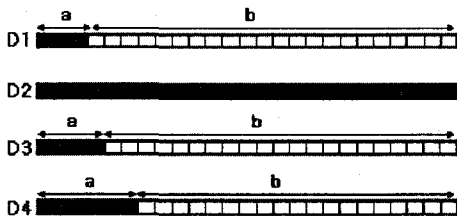
디스크립션 별로 실행 가능한 최소의 비트율이 있을 것이다. 하지만 실제 네트워크 속도가 이것에 미치지 못할



[그림5]

때, 로컬에 디스크립션의 일부분을 프리패치시키는 전략을 사용할 것이다. 그러면 프리패치 사이즈를 어떻게 결정할지에 대해서 살펴보자. 먼저 용어에 대한 정의부터 하겠다.

- C : 원본 콘텐츠
- CS : 원본 콘텐츠 사이즈 (Mbyte)
- CT : 콘텐츠의 실행시간 (초)
- CD_{Sn} : 콘텐츠에 대한 n번째 디스크립션의 사이즈 (Mbyte)
- CD_{bn} : 콘텐츠에 대한 n번째 디스크립션에 대해서 서비스 가능한 최소 비트율 (b/s)
- CD_{Rbn} : 콘텐츠에 대한 n번째 디스크립션이 노드에 전송하는 실제 네트워크 비트율 (b/s)
- $CPfSn$: 콘텐츠에 대한 n번째 디스크립션을 CD_{bn} 과 CD_{Rbn} 을 고려해서 로컬노드에게 프리패치시켜야 할 사이즈(Mbyte)
- us : 윈도우 사이즈는 (본 논문에서는 55600byte로 고정)



[그림6]

콘텐츠에 대한 n번째 디스크립션에 대한 최소 비트율은 아래와 같이 계산이 된다.

$$CD_{bn} = \frac{(10^3 \times 10^3 \times 8 \times CD_{Sn})}{CT}$$

그리고 만약 $CD_{bn} > CD_{Rbn}$ 이면 서버넷에 프리패치

시켜야 할 것이다. 서버넷에서 관리해야 하는 노드가 가져야 할 양은 CT시간 동안 $CD_{bn} - CD_{Rbn}$ 의 속도로 다운로드 되는 양을 미리 프리패치시켜 놓으면 된다. 예를 들어 실행시간이 60분인 콘텐츠에 대해서 100Mbyte의 용량을 가지는 디스크립션이 있을 때, 실제로 끊임 없이 보려면 200kbps의 속도가 나와야한다. 하지만 150kbps의 속도만 나온다면 50kbps가 60분 동안 받아야 하는 양을 미리 프리패치시켜 놓으면 된다. 클라이언트에게는 27Mbyte로 서비스 하면서 글로벌로부터 나머지를 받아 오면서 클라이언트에게 서비스를 하면 된다. $CPfSn$ 은 아래와 같이 계산된다.

$$CPfSn = (CD_{bn} - CD_{Rbn}) \times CT$$

(그림 6)은 D2가 4개의 디스크립션 비트율이 가장 느려 서버넷에 디스크립션 전체를 복제했다. 그리고 나머지에 대해서는 해당 디스크립션에 실제 네트워크 비트율과 실행 가능한 최소의 비트율에 따라서 프리패치해 놓았다. 프리패치한 노드들은 클라이언트에게 'a'로서 서비스 해주면서 글로벌로부터 'b'를 받아와 계속해서 서비스를 한다.

5. 결론 및 향후 연구과제

우리는 물리적 토폴로지를 고려한 P2P 환경을 기반으로 MDC를 이용한 미디어 스트리밍 서비스를 제안하였다. 우리가 제안한 논문은 스트리밍 서비스를 받을 때 물리적으로 가까운 노드로부터 서비스를 받게 되며, MDC를 이용함으로써 일부 노드가 서비스를 못하는 경우가 발생하더라도 계속해서 서비스할 수 있는 장점을 가졌다.

향후에 콘텐츠의 인기도 따라 복제 정책을 세워야하고 콘텐츠의 특정 부분에 대해서 인기도가 높아질 경우에 해당 부분만을 미리 복제하는 정책 또한 세워야한다.

참고문헌

- [1] 차봉관, 한동윤, 손영성, 김경석, "Lookup 효율성을 위해 Topology를 고려한 Chord 시스템", 한국컴퓨터 종합학술대회 2005 논문집 Vol.32, pp.589-591 2005. 7.
- [2] I. Stoica, R. Morris, D. Karger, M.F Kaashoek, and H. Balakrishnan, "Chord: A Scalable Peer-to-Peer Lookup Service for Internet Applications", ACM SIGCOMM'01, Aug. 2001.
- [3] Yen-Chi Lee, Joohee Kim, Yuce!Altunbasak, Russell M. Mercereau, "Layerd coded vs. multiple description coded video over error-prone networks", EURASIP signal processing : image communication, Vol.18, pp.337-356, May, 2003
- [4] Michael Zink, Andreas Mauthe, "P2P Streaming using Multiple Description Coded Video", Proceedings of EuroMicro2004, Sep., 2004
- [5] L.Ozarow, "On a source coding problem with two channels and three receivers," Bell Syst. Tech. J., Vol.59, pp.1909-1921, Dec., 1980.

- [6] V.A. Vaishampayan, "Design of multiple description scalar quantizer," IEEE Trans. Inform. Theory, Vol.39, pp.821-834, May, 1993.
- [7] A. Ingle and V. A. Vaishampayan, DPCM system design for diversity systems with applications to packetized speech, IEEE Trans. Speech and Audio Processing, Vol.3, pp.48-57, Jan., 1995
- [8] M. Orchard, Y. Wang, V. A. Vaishampayan, and A. Reibman, "Redundancy rate distortion analysis of multiple description image coding using pairwise correlating transforms," Proc. Int. Conf. Image Processing, pp.608-611, Oct.,1997
- [9] Y. Wang, M. Orchard, V. A. Vaishampayan, and A. Reibman, "Multiple description coding using pairwise correlating transforms," IEEE Trans. Image Processing, vol.10, pp.351-366, Mar., 2001.
- [10] Apostolopoulos, J.G., "Error-resilient Video Compression via Multiple State Streams", In Processings of VLBV'99, Oct., 1999.
- [11] 김선호, 송병호, "멀티미디어 콘텐츠의 QoS를 개선한 전송 메커니즘", 정보처리학회 논문지 B, 제12-B 권 제2호, 2005. 4.