

다중 디스크립션 코딩을 이용한 P2P 미디어 스트리밍 시스템

이승은[○], 한동윤, 박유현[‡], 김경석[†]
부산대학교 컴퓨터공학과[○], 한국전자통신연구원[‡], 부산대학교 정보컴퓨터공학부[†]
{selee[○], dyhan, gimgs0[†]}@asadal.pusan.ac.kr, bakyh[‡]@etri.re.kr

P2P Media Streaming System Using Multiple Description Coding

Seung-eun Lee[○], Dong-yun Han, Yuhyeon Bak[‡], Kyong-sok Kim[†]
Dept. of Computer Engineering[○], Electronics and Telecommunications Research Institute[‡],
Division of Computer Science and Engineering, Pusan National University[†]

요 약

최근 인터넷 접속 환경의 발달과 시스템 및 네트워크 성능 향상의 영향으로 단순한 정보의 전달뿐만 아니라 오디오, 비디오와 같은 미디어 콘텐츠(contents)의 정보 전달을 효율적으로 제공하는 방안이 연구되고 있다. 미디어 콘텐츠를 스트리밍 하는 것은 인터넷 트래픽의 아주 큰 부분을 차지하기 때문에 이러한 콘텐츠를 효율적으로 분산하기 위하여 우리는 P2P(Peer-to-Peer) 오버레이 네트워크를 이용한다. 본 논문은 Tapestry[1]를 미디어 콘텐츠 스트리밍 환경에 맞게 변형하며, 노드와 네트워크의 부하를 균등하게 분산시키기 위해 MDC(multiple description coding)[2]를 이용하여 하나의 콘텐츠를 다중 트리에 분산한다. 제안된 메커니즘으로 사용자에게 보다 강화된 서비스를 제공한다.

1. 서 론

최근 몇 년 동안 인터넷 트래픽의 아주 큰 부분을 차지하는 미디어 콘텐츠를 네트워크상에 분산하는 것이 논제가 되고 있다. 이것에 관한 연구는 전체적으로 크게 세 가지로 분류할 수 있다: (a) 서버/클라이언트 구조 (b) CDN(content distribution network) (c) P2P(Peer-to-Peer) 구조. 먼저, 서버/클라이언트 구조에서 서버는 콘텐츠를 저장, 유지 그리고 관리하게 된다. 서버가 모든 정보를 가지고 있기 때문에 클라이언트가 콘텐츠를 검색하고 이용하기는 매우 편리하다는 장점을 가지지만, 서버 과부하 발생 시 클라이언트가 받는 스트리밍 미디어 콘텐츠의 품질이 상당히 떨어질 수 있다. 두 번째, Akamai와 같은 기존의 CDN은 서버/클라이언트 구조에서 서버를 보충하며 클라이언트를 분산하고 저장하기 위해 제공된 기기들을 사용한다. 이것은 높은 수준의 성능을 보장한다. 하지만 비용이나 규모면에서 CNN 같은 영리적인 큰 사이트에 적합하며 동시에 많은 클라이언트들이 접근하여 큰 트래픽이 발생할 경우 대처하는 방법이 명확하지 못하다. 마지막으로, P2P 구조에서는 클라이언트들이 콘텐츠를 저장하고 유지하며 다른 클라이언트들에게 분산한다. 이것은 적은 비용으로 부하를 고르게 분산시킬 수 있지만 노드(node)들의 참여(join)나 이탈(leave)이 자주 발생함으로 높은 수준의 스트리밍 미디어 콘텐츠를 제공할 수 없다.

본 논문은 P2P 구조에서 분산 해시 테이블(distributed hash table : DHT)을 기반으로 구현된 Tapestry[1]를 사용한다. 이것은 P2P의 고유한 성질로 시스템을 구성하기 위한 비용이 적으며, 노드들의 빈번한 참여와 이탈에도 불구하고 높은 이용 가능성을 가지고 콘텐츠에 접근하도록 한다. 또한, MDC를 이용하여 오디오/비디오 같은 스트리밍 미디어 콘텐츠를 효율적으로 분산하여 특정 노드의 과부하를 방지하는 방안을 제시한다.

본 논문은 다음과 같이 구성된다: 2장은 본 논문과 관련이 많은 Tapestry와 MDC를 살펴보고, 3장에서는 우리가 제안하는 스트리밍 미디어 콘텐츠 분산 방법에 대하여 설명한다. 마지막으로 4장에서는 결론 및 향후 과제를 보여준다.

2. 관련 연구

2.1 Tapestry

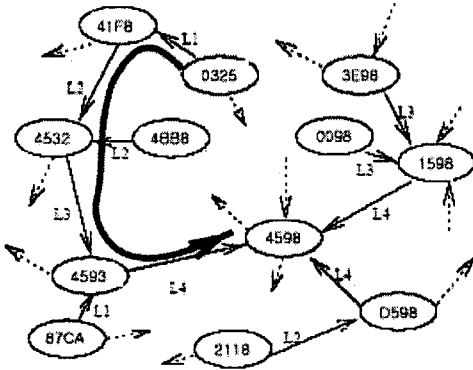
이번 장에서는 P2P 네트워크에서 Tapestry라 불리는 응용 계층 탐색 프로토콜을 소개한다. 이 연구는 Plaxton, Rajaraman과 Richa에 의해 소개된 hashed-suffix mesh[3]와 유사한 메커니즘을 사용한다. 본 논문에서는 hashed-prefix mesh를 사용하여 탐색한다. Tapestry와 같은 DHT 기반 P2P 시스템의 자원 탐색시간은 $O(\log N)$ 으로 알려져 있다.[4][5] 각 Tapestry 노드는 서버, 라우터 그리고 클라이언트 역할을 한다. 또한, 노드들과 데이터들은 시스템 상에 참여할 때 임의적이며 유일한 고정 길이의 비트를 가지고 물리적 위치와 독립된 식별자를 할당받게 된다. 우리는 이것을 노드와 데이터에 따라 NodeID와 GUID(Global Unique ID)로 표현한다. 이들 NodeID들은 이웃 링크(neighbor link)들의 망을 구성하기 위해 사용된다. 이 시스템은 SHA-1의 해시 알고리즘을 사용하여 노드들과 데이터들에 할당되는 ID들이 고르게 분산됨을 가정한다.

2.1.1 탐색 방법

Tapestry는 네트워크 지역성(locality)을 고려하여 IP를 통해 탐색하는 시간으로 측정된 IP 탐색 기반구조에서의 가장 가까운 노드들을 이웃(neighbor)이라고 하며, 각 노드는 자신의 이웃들의 정보를 유지하여 다중 레벨로 구성된 하나의 이웃 맵(neighbor map)을 가진다. 각 레벨은 목적지 ID인 GUID와 공유된 접두사(prefix)의 단계

를 나타내며, 이 탐색 맵을 사용하여 탐색한다. 이것은 요청 NodeID에서 GUID로 ID의 자릿수를 증가시키면서 메시지를 전송하는 방법으로, N번째 혹은 GUID와 적어도 N 길이의 접두사를 공유한다. GUID와 가장 많은 접두사를 공유하며 숫자상 가장 가까운 NodeID는 그 데이터의 루트 노드(root node)가 된다. 즉, Tapestry[1]는 네트워크 내의 모든 노드가 루트가 된 트리들의 하나의 큰 집합으로 볼 수 있다.

데이터 O에 대한 질의 메시지는 GUID를 가지고 O의 루트 쪽으로 순차적으로 레벨들을 탐색한다. 각 단계에서, 메시지가 O의 위치 정보를 포함하고 있는 노드를 만난다면, 그 메시지는 즉시 데이터를 포함하고 있는 서버 쪽으로 방향을 전환한다. 만약 만나지 못한다면, 메시지는 루트에 도달하게 되어 O에 대한 위치 정보 탐색을 보장한다.



[그림 1] GUID 4598 찾는 노드 0325의 Tapestry 탐색

2.1.2 데이터 위치선정

데이터 O를 가진 서버 S는 O의 루트 노드 쪽으로 메시지를 전달하여 데이터가 네트워크에 참여하여 존재하게 되었음을 알린다. 루트 노드는 O에 대한 트리의 루트가 되는 장소로 사용되는 네트워크 내의 유일한 노드이다. 메시지가 지나가는 경로의 각 홉에서, 알림 메시지는 <Object-ID(O), Server-ID(S)> 형식으로 데이터 O가 저장된 서버 S의 위치를 가리키는 간단한 포인터를 저장한다. 이것은 단순한 포인터일 뿐 데이터의 복사본이 아니다.

Tapestry는 하나의 데이터를 다중 경로를 통해 제공할 수 있기 때문에 네트워크 부하를 균등하게 하며 장애에 대한 내구성을 가진다. 또한 지역성에 근거하여 이웃 망을 구성하기 때문에 가까이에 위치한 데이터를 빨리 찾을 수 있다. 하지만 각 데이터는 단일 루트를 가지고 있으므로 이용 가능성(availability) 문제를 가지고 있으며, 많은 클라이언트들이 특정 데이터를 짧은 시간동안 동시에 요청한다면 루트 노드에 과부하가 발생할 수도 있다.

2.2 MDC (Multiple Description Coding)

MDC는 오디오/비디오 신호를 1개 이상의 스트림으로 분할하거나 n개의 디스크립션(description)으로 인코딩하여 각각을 독립적인 경로로 전송하는 메소드이다.¹⁾ 각 디스크립션은 서로 내용이 중복되기 때문에 어느 하나를 잃더라도 계속적인 스트리밍 서비스가 가능하다는 것이

가장 큰 장점이다. 반면 사용자가 작은 수의 디스크립션을 받는다면 원본 소스보다 낮은 품질로 디코딩 된다.

3. 시스템 구조

우리는 Tapestry 구조의 단일 루트로 인한 결점을 보완하고 MDC를 사용하여 스트리밍 미디어 콘텐츠를 분산하는 시스템 구조를 제안한다. Tapestry 구조의 결점을 보완하기 위해 하나의 콘텐츠를 n개의 디스크립션으로 인코딩하여 n개의 분산 트리를 가지게 된다. 이것은 몇 개의 다른 루트 노드를 생성함으로써 네트워크 부하를 분산시켜 그 콘텐츠를 가진 노드의 과부하를 미리 방지할 수 있으며 이용 가능성의 효율을 높인다.

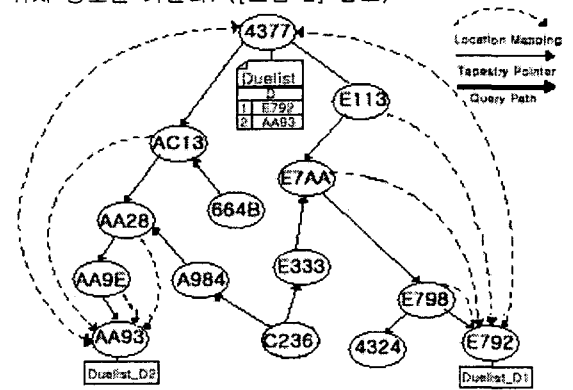
3.1 변형된 Tapestry를 이용한 미디어 스트리밍

원본 콘텐츠가 다음과 같은 디스크립션으로 구성될 때, $C = D1 + D2 + \dots + Dn$, 각 디스크립션은 하나의 파일로 관리되며 n개의 디스크립션이 P2P에 배치된다.

만일 사용자가 원본 콘텐츠와 같은 품질로 서비스를 받고자한다면 n개의 디스크립션을 전송받아야 한다.

디스크립션의 수를 '2'로 가정한 시스템의 시나리오는 다음과 같다.

(1) 노드 4377이 콘텐츠 'Duelist'를 가지고 Tapestry 네트워크에 참여할 때, 노드 4377은 콘텐츠 'Duelist'를 2개의 디스크립션으로 인코딩한다. 각 디스크립션은 'Duelist_D1', 'Duelist_D2'라는 파일로 관리되며 각 파일명을 해쉬하여 서로 다른 GUID인 'E792', 'AA93'을 할당 받게 된다. 그리고 노드 4377은 2개의 GUID의 루트 노드들을 탐색하여 각 디스크립션을 루트 노드에 분산 저장하고 노드 4377과 탐색 경로 상의 중간 노드들은 루트 노드의 위치 정보를 가진다. 또한, 노드 4377과 콘텐츠 'Duelist'의 디스크립션의 루트 노드들은 서로의 위치 정보를 가진다. ([그림 2] 참조)

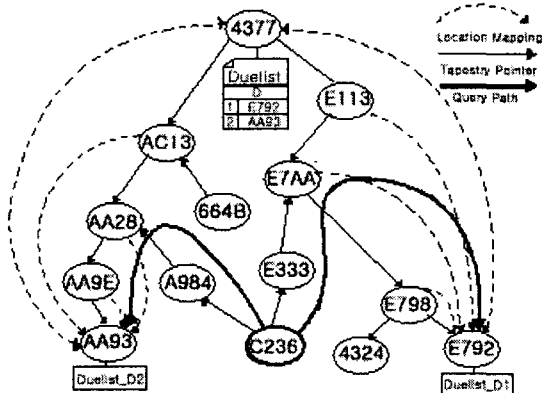


[그림 2] 콘텐츠 'Duelist'의 디스크립션 {D1, D2}를 분산 저장

(2) 콘텐츠 'Duelist'를 서비스 받기 원하는 요청 노드 C236은 각 디스크립션의 GUID인 'E792', 'AA93'을 가지고 루트 노드들을 찾아가는다. 각 루트 노드의 위치정보를 가진 중간 노드 E7AA와 AA28에 의해 직접적으로 연

1) 8~16개의 디스크립션은 충분한 서비스를 제공한다. [6][7]

결하여 서비스를 제공 받게 된다. ([그림 3] 참조)



[그림 3] 노드 'C236'에서 콘텐츠 'Duelist' 탐색

제안된 시스템 시나리오는 각 콘텐츠에 대해 다중 분산 트리로 디스크립션을 분산 저장함으로써 사용자에게 각각 다른 경로를 통하여 디스크립션을 제공한다. 이것은 끊어지지 않는 안정적인 서비스를 보장한다.

비디오 화상회의와 같이 하나의 노드가 특정 다수의 목적지에 데이터를 효과적으로 전송하기 위해 멀티캐스트가 많이 사용되고 있다. 스트리밍 미디어 콘텐츠를 안정적으로 서비스하는 본 시스템은 Tapestry 기반의 멀티캐스트 방법을 제안한 Bayeux[8]를 통하여 간편하게 멀티캐스트를 제공할 수 있다.

3.2 장애 발견 및 관리

본 시스템은 발생할 수 있는 장애를 크게 두 가지로 분류하고 그에 따른 관리방법들을 제안한다.

첫 번째로, 각 콘텐츠에 대해 n 개의 디스크립션을 유지하여 사용자에게 안정된 서비스를 제공하기 위해, 디스크립션의 이탈을 발견하고 관리하는 방법이다. 네트워크에 참여한 노드 S 가 콘텐츠를 분산한 MDC의 디스크립션을 저장하고 있는 노드들의 존재 여부를 확인하고 관리하는 것으로, 노드 S 는 디스크립션을 저장하고 있는 노드들에게 주기적으로 작은 단위의 광시 메시지를 보냄으로 그 노드들의 존재 여부를 확인하게 된다. 예측 가능한 이탈 시, 이탈하는 노드는 자신이 네트워크를 떠나게 됨을 노드 S 에게 알리고 노드 S 는 이탈한 노드가 저장하고 있던 디스크립션의 GUID를 가지고 새로운 루트 노드를 찾아가 그 디스크립션을 저장하게 된다. 새로운 루트 노드로 가는 경로 상의 중간 노드들과 노드 S 는 그 루트 노드의 위치 정보를 저장하게 되며 이탈 노드의 위치 정보를 저장하고 있는 경로 상의 중간 노드들에게 광시 메시지를 보낸다. 예측 불가능한 이탈 시, 이탈 노드는 어떠한 경고도 없이 떠나므로 노드 S 는 광시 메시지를 통해 일정 시간동안 응답이 없을 시 그 노드가 이탈했음을 알게 된다. 전자의 경우와 마찬가지로 새로운 루트 노드를 찾아서 갱신하게 된다.

두 번째로, 콘텐츠 'C'를 가진 노드 S 의 네트워크 이탈에 대비하는 방법이 있다. 노드 S 는 콘텐츠 'C'를 k 개의 다른 해쉬 함수를 사용하여 복제한다. k 의 수는

그 콘텐츠의 인기도에 따라 결정될 수 있으며 네트워크 상에 균등하게 분산되게 된다. 하나의 디스크립션에 대한 다중 복사본들의 위치 정보를 유지하고 있는 노드는 복사본들로부터의 거리에 따라 정렬한다. 그러므로 그 디스크립션을 원하는 사용자들은 가장 가까운 거리에서 빠르게 서비스를 제공받을 수 있다.

그 외의 다른 장애가 발생할 경우에는 Tapestry의 복구 방법을 이용하여 안정적으로 복구한다.

4. 결론 및 향후 연구과제

우리는 P2P 시스템에서 스트리밍 미디어 콘텐츠를 제공하기 위한 메커니즘을 제안하였다. MDC를 사용하여 스트리밍 미디어 콘텐츠를 네트워크 상에 분산 시키므로 하나의 콘텐츠는 다중 분산 트리를 갖게 된다. 이것은 Tapestry 구조의 단일 루트 노드의 문제점을 해결하며, 노드의 부하를 균등하게 만들었다. 또한, MDC의 각 디스크립션을 k 개로 복제함으로써 이용 가능성의 효율을 높였으며 사용자에게 일정 수준 이상의 서비스를 제공하게 한다.

대역폭과 노드의 능력에 따라 MDC의 디스크립션의 수를 다르게 적용하는 방안과 사용자에게 더 나은 품질의 서비스를 제공하기 위해 새로운 분산 트리를 고안 중이다. 현재 본 논문에서 제안한 초기 메커니즘의 성능을 평가하기 위하여 실험 단계에 있다.

참조문헌

- [1] B. Y. Zhao, J. Kubiatowicz, and A. D. Joseph, "Tapestry: An infrastructure for fault-tolerant wide-area location and routing", Tech. Rep. UCB/CSD-01-1141, Computer Science Division, University of California, Berkeley, Apr 2001
- [2] J. G. Apostolopoulos and S. J. Wee, "Unbalanced multiple description video communication using path diversity", In IEEE International Conference on Image Processing, Oct. 2001
- [3] PLAXTON, C. G., RAJARAMAN, R., AND RICHA, A. W. "Accessing nearby copies of replicated objects in a distributed environment", In Proceedings of the Ninth Annual ACM Symposium on Parallel Algorithms and Architectures (SPAA) (June 1997)
- [4] 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
- [5] S. Ratnasamy, P. Francis, M. Handley, R. Karp, "A Scalable Content-Addressable Network", ACM SIGCOMM'01, Aug. 2001
- [6] Aditya Ganjam and Hui Zhang, "Internet Multicast Video Delivery", In proceedings of the IEEE, VOL.93, NO.1, JANUARY 2005
- [7] Venkata N. Padmanabhan, Helen J. Wang, and Philip A. Chou, "Resilient Peer-to-Peer Streaming", In Technical Report MSR-TR-2003-11 of Microsoft Research, March 2003
- [8] Shelley Q. Zhuang, Ben Y. Zhao, Anthony D. Joseph, Randy H. Katz, John D. Kubiatowicz, "Bayeux: An Architecture for Scalable and Fault-tolerant Wide-area Data Dissemination", In Proceedings of NOSSDAV, 2001