

라이브 미디어 스트리밍 서비스를 위한 P2P기반 데이터 분할 전송 시스템

최 순, 변혜선, 이미정
이화여자대학교 컴퓨터공학과

e-mail: suni@ewhain.net, ladybhs@ewhain.net, lmj@ewha.ac.kr

P2P-based divisional data transmission system for live media streaming service

Sun Choi, Heasun Byun, Meejeong Lee
Dept. of Computer Science and Engineering, Ewha Womans University

요 약

최근 인터넷 사용자들의 요구는 멀티미디어로 집중되고 있으며 그중 라이브 미디어 스트리밍 서비스에 대한 요구가 증가하고 있다. 라이브 서비스에서는 적절한 시간에 사용자에게 데이터가 도착하는 것이 중요하다. 따라서 라이브의 시간적절성을 충족시켜 줄 효율적이고 신속한 데이터 전달구조와 전송 기법이 요구된다. 이에 본 논문에서는 트리와 메시 구조를 혼합한 하이브리드 방식으로 네트워크 자원을 효율적으로 사용하면서 빠른 데이터 전송으로 라이브의 시간적절성을 충족시킬 수 있는 데이터 분할 전송 방식의 P2P(Peer-to-Peer) 오버레이 구조를 제안한다. 제안하는 ToG(Tree of Groups)는 n개의 피어들이 메시로 그룹을 형성하고, 그렇게 형성된 그룹들이 트리를 이루는 구조이다. ToG에서 그룹 내의 각 피어들은 상위그룹의 피어 한 개와 부모-자식으로 연결되어 있어서 그룹 사이에 여러 개의 연결이 존재하게 된다. 따라서 그룹 내에서 어느 한 피어가 그룹을 빠져 나가더라도 상위그룹과의 여러 연결에 의해서 서비스 지속성을 보장 할 수 있다. ToG는 그룹단위로 트리가 형성되기 때문에 피어의 개수가 같을 때 피어단위로 트리를 형성하는 구조보다 트리의 깊이가 얕아진다. 그에 따라 말단에 있는 피어들에게까지 빠른 시간에 데이터가 전달 될 수 있다. ToG의 데이터 전달은 소스로부터 세그먼트가 일정한 값 n으로 나뉘어져 각 피어들에게 전달된다. 세그먼트 조각은 소스로부터 나뉘어져 전송 될 때 책임적으로 전달해야할 피어와 전달 순서가 정해져있고, 데이터 전송 스케줄링을 위한 버퍼 맵 교환은 필요하지 않다.

1. 서론

개개인의 PC의 성능이 좋아지고, 자원이 풍부해짐에 따라 사용자들의 멀티미디어에 대한 요구도 높아지고 있다. 멀티미디어 데이터 전송은 기존의 텍스트 기반의 데이터 전송보다 네트워크 자원이 더욱 많이 필요하다. 그에 따라 현존하는 네트워크에서 자원을 효율적으로 사용하기 위한 방안으로 각 피어들이 서로 오버레이를 구성하여 데이터를 전달하는 P2P 전송방식이 제안되었다. P2P 방식은 확장성이 좋고, 분산된 네트워크 환경을 제공하지만 피어들의 접속이 유동적이기 때문에 구조를 안정적으로 유지하기 위한 방안이 필요하다.

P2P 구조는 크게 두 가지로 분류할 수 있다. 하나는

트리 기반 구조이고, 다른 하나는 메시 기반의 구조이다. 트리 기반 구조는 피어들이 부모-자식으로 연결되어 있다. PUSH로 데이터를 전달하기 때문에 버퍼 맵 교환이나 데이터 요청 없이 데이터가 빠르게 전달될 수 있으나, 트리 상단의 피어가 갑작스럽게 오버레이를 떠나는 경우 하단 피어들의 서비스 지속성에 큰 영향을 미칠 수 있다는 단점을 지닌다. 메시 기반 구조는 여러 피어들과 이웃을 맺고, 그 이웃들과 버퍼 맵 교환을 통하여 PULL로 데이터를 전달하는 방식이다. 이웃 중 하나의 피어가 빠져나가더라도 다른 피어들에게 서비스를 지속해서 받을 수 있는 장점을 지닌다. 하지만 주기적인 버퍼 맵 교환을 하기 때문에 트리에 비해 추가적으로 많은 제어메시지가 발생하는 단점을 지닌다.

사용자들의 요구는 이제 단순히 멀티미디어를 전송만 고자 하는데서 나아가 멀티미디어를 라이브로 서비스 받기를 원하고 있다. 라이브 서비스는 적절한 시간 안에 데이터가 사용자에게 전달되는 것이 중요하다. 따라서 시간적절성을 충족시켜 줄 수 있는 구조와 전송기법이 요구된다.

"이 논문은 2008년도 정부(교육과학기술부)의 재원으로 한국과학재단의 지원을 받아 수행된 연구임(No. R01-2008-000-12015-0)."

"본 연구는 지식경제부 및 정보통신연구진흥원의 대학 IT 연구센터 지원사업의 연구결과로 수행되었음."

(IITA-2008-(C1090-0801-0036))

이 논문에서 제안하는 ToG(Tree of Groups)는 라이브 스트리밍 서비스에 중요한 데이터의 시간적질성을 만족시키는 하이브리드 P2P 오버레이 구조이다. ToG에서는 일정 시간(One Frame Time: OFT)에 전송되어야 할 세그먼트가 작은 조각들로 나뉘어져 PUSH로 전송된다. OFT는 사용자가 일정시간 플레이할 수 있는 데이터의 양을 의미한다. 즉, 다음 플레이를 위하여 준비되어야 하는 일정한 스트리밍 데이터의 양이다. ToG에서 세그먼트 조각은 소스에서 나누어질 때 이미 조각을 전송할 피어와 전송경로가 책임적으로 정해져 있다. 따라서 한 피어에게 조각이 중복되어 전송되는 일이 없고, 이를 통해 네트워크 자원을 효율적으로 사용할 수 있다.

기존에 제안된 하이브리드 구조인 mTreebone[1]은 트리로 데이터를 전송하고 메시로는 손실된 데이터만을 받고, Anysee2[2]에서는 컨트롤 메시지는 트리를 통해서 전달하며 데이터는 메시로 전달한다. 이 두 구조와는 다르게, ToG는 트리와 메시에서 모두 데이터가 전달된다. 따라서 기존 연구들에 비하여 각 피어들에게 데이터가 전송되는 hop수를 줄여 전체적인 데이터 전달 속도를 향상시킬 수 있다.

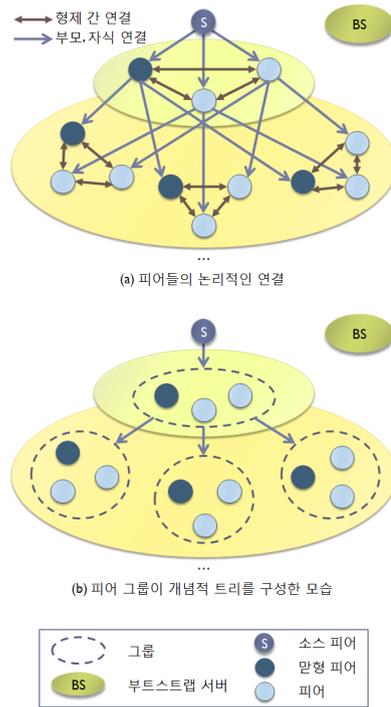
ToG와 비슷하게 데이터를 분할하여 전송하는 HCPS[3]는 Cluster의 Header들이 소스에 연결되어 있고 Cluster 내에서 피어들이 메시로 연결된 계층적인 P2P 구조이다. ToG와 유사하게 HCPS도 Cluster Header들이 소스에 연결되어 데이터 전달 hop수가 짧아 각 피어들에게 데이터가 빠르게 전달되는 장점을 가진다. 그러나 HCPS는 Cluster내에서 Cluster Header만이 소스와 연결되어 있기 때문에 Cluster Header가 갑작스럽게 오버레이를 떠날 경우 Cluster 내의 다른 피어들의 서비스 지속성이 심각하게 영향을 받는다. 이에 반해 ToG는 어느 한 피어가 오버레이를 떠난다 하더라도 나머지 피어들은 부모그룹의 피어들과 계속해서 연결되어 있기 때문에 서비스 지속성에 큰 영향을 받지 않는다.

본 논문의 구성은 다음과 같다. 1장 서론에 이어, 2장에서는 제안하는 ToG의 구조와 구성요소, 데이터 전송 방법에 대하여 이야기하고 3장에서 피어의 가입과 탈퇴에 대하여 설명한다. 마지막으로 4장에서 결론을 내리고 앞으로의 연구방향에 대하여 이야기한다.

2. ToG

2.1. ToG 구조

그림 1은 $n=3$ (n : 그룹 내 최대 피어 개수)인 ToG구조의 예를 보여준다. 그림 1(a)는 오버레이에서 피어들의 연결을 보여준다. 그림 1(b)는 그룹들이 서로 트리처럼 연결되어 있는 모습을 보여준다. ToG는 그룹 내에서는 메시로 연결되어있고, 그룹 간에는 트리로 연결되어 있다. 즉, 한 개의 피어는 한 개의 부모피어와 $n-1$ 개의 형제피어, m 개(m : 자식 그룹의 수)의 자식피어와 연결을 맺고 있고, 각 그룹들은 n 개의 링크로 상위 부모 그룹과 연결되어있다.



(그림 1) $n=3$ 인 ToG구조의 예

m 은 그룹 내의 만형피어가 자신의 그룹이 서비스할 수 있는 능력을 고려하여 적당한 값을 결정하게 된다.

이러한 연결에서 세그먼트는 n 개의 조각으로 나뉘어져 PUSH로 부모피어에서 자식피어로 전달된다. 각각의 피어들은 자신의 부모에게 받은 조각만을 형제피어들과 자식 피어들에게 전달한다. 피어는 형제피어들과 메시처럼 연결되어 있지만 자신이 부모에게 받은 조각만을 전달하기 때문에 버퍼 맵 교환을 하거나 데이터 요청을 하지 않는다. 예외적으로 데이터 손실이나 자신과 연결된 피어의 갑작스런 탈퇴로 인하여 데이터를 받지 못했을 경우에만 부모나 형제피어에게 손실 데이터를 요청하게 된다.

2.2. ToG 구성요소

그림 1에서 보면 ToG는 소스(S)와 부트스트랩 서버(BS), 만형피어, 피어 등의 구성요소로 이루어져 있다. 소스는 OFT에 전송되어야 할 세그먼트를 n 개의 조각으로 나누어 전송하는 데이터의 근원지이다. 부트스트랩 서버는 새로운 피어가 ToG에 가입하도록 도와주는 서버이다. 만형피어는 그룹의 관리를 맡은 피어로서, 그룹에 가장 먼저 속하게 되는 피어가 만형피어의 역할을 하게 된다. 만형피어가 오버레이를 떠났을 경우에는 그룹에 참여한 피어 순으로 만형피어를 맡게 된다. 만형피어는 그룹을 떠나는 피어가 발생했을 경우 그 피어의 역할을 대신할 피어를 그룹 내에서 선택하고, 응답조직 프로세스를 시작한다. 책임

피어는 부트스트랩 서버를 도와 현재 새로운 피어의 가입을 도와주는 만형피어이다.

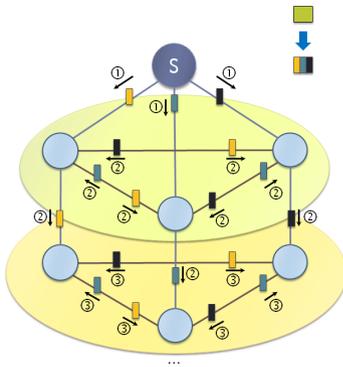
ToG에서는 두 가지 리스트가 필요하다. 하나는 각각의 피어들이 가지고 있는 자식리스트이고 다른 하나는 부트스트랩 서버가 관리하고 있는 책임리스트이다. 자식리스트는 자식그룹에 속한 n*m개 피어에 대한 정보를 담고 있다. 자식리스트는 책임리스트를 업데이트하기 위한 용도로 사용되고, 자식그룹 피어의 탈퇴에 대처하기 위한 용도로 사용된다. 책임리스트에는 책임피어가 될 자격이 있는 피어들, 즉 자식그룹에 새로운 피어들이 들어올 수 있는 그룹의 만형피어들이 들어가 있다. 부트스트랩 서버는 책임리스트의 가장 상단에 있는 책임피어를 새로운 피어에게 알려줌으로서, 책임피어의 자식그룹에 새로운 피어가 들어갈 수 있도록 한다.

2.3. 데이터 전송 방법

그림 2는 n=3일 때, 세그먼트 조각이 전달되는 과정을 보여준다. OFT에 전송되어야할 세그먼트는 n개의 조각으로 나누어져 소스와 연결된 하위그룹 각각의 피어들에게 전달된다(그림2-①). 소스는 n개의 피어에게 각각 다른 n개의 조각을 보내고, 조각 n개중 1개의 조각을 받은 피어는 자신의 형제피어들과 자식피어들에게 자신이 받은 1개의 조각을 전달한다(그림2-②,③). 이와 같은 과정을 통해 모든 피어들은 부모에게 1개, 각각의 형제에게 1개씩 n-1개 조각을 받아서 n개의 조각을 모두 받게 된다.

그룹에 n개의 피어가 다 채워지지 않았을 경우, 그룹에 존재하는 피어들 사이에서는 서로 데이터를 교환하고 받지 못한 조각은 부모피어들이 그 조각을 전달할 피어가 생기기 전까지 각자의 자식피어에게 전달하여 준다.

앞장에서 언급한 바와 같이 ToG는 손실된 데이터 조각이 발생했을 경우에만 PULL을 이용하여 자신의 부모피어나 형제 피어에게 데이터 조각을 요청한다. 부모피어에게 데이터 조각을 받지 못한 피어는 형제피어들에게 자신이 데이터 조각을 받지 못했음을 알리고, 각각의 피어들은 받지 못한 조각을 자신의 부모피어에게 요청하여 받게 된다. 이때, 만형피어가 부모에게 조각을 받지 못했음을



(그림 2) 데이터 전송

알린 피어에게 자신이 전달해야하는 조각과 함께 그 조각을 전달해 준다.

3. ToG 프로세스

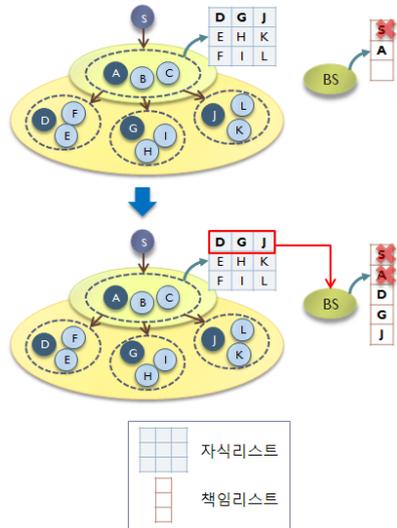
3.1 가입

A. 가입

새롭게 가입하고자 하는 피어(N)는 부트스트랩 서버에게 가입 요청 메시지를 보낸다. 부트스트랩 서버는 피어 N에게 책임리스트 상단에 있는 책임피어를 알려준다. 피어 N은 책임피어에게 접속하여 가입하고 싶다는 것을 알리고, 책임피어는 피어 N에게 피어 N이 속하게 될 그룹의 형제피어들과 연결될 부모피어를 알려 준다. 그러면 피어 N은 자신의 부모, 형제피어들과 연결을 맺는다.

B. 책임리스트 업데이트

그림 3에서 책임피어 A는 자식리스트가 꽉 차게 되어 더 이상 책임피어의 역할을 수행할 수 없게 되었다. 피어 A는 BS에게 자신이 가지고 있는 자식리스트의 만형피어들을 알려주고, 자식그룹에게는 자식그룹의 만형피어들이 책임리스트에 속하게 되었음을 알려준다. BS는 책임리스트에서 피어 A를 삭제하고 새롭게 받은 자식리스트를 가지고 책임리스트를 업데이트 한다.

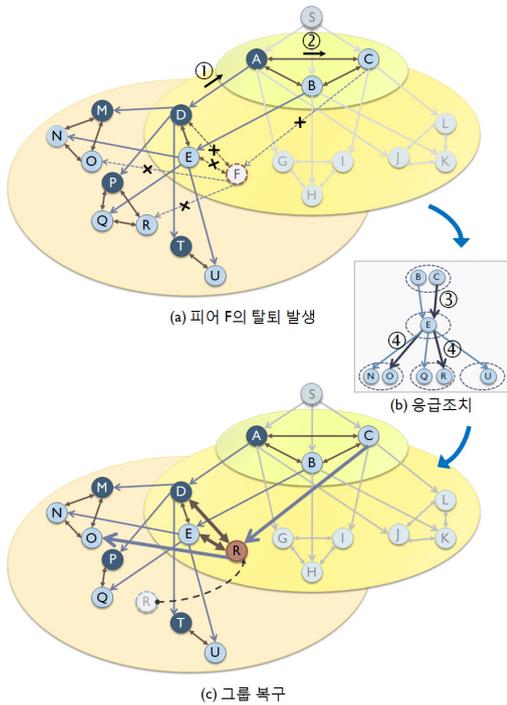


(그림 3) 책임리스트 업데이트

3.2 탈퇴

A. 데이터 전달을 위한 응급조치

형제피어가 갑작스럽게 빠져 나가 데이터 조각을 받을 수 없게 된 피어들은 자신의 부모피어에게 받지 못한 조각을 요청한다. 형제피어가 P2P 오버레이를 떠난 것인지 확인하는 방법으로, 빈번하게 데이터 조각을 보내지 않는 피어에 대하여 일정 시간(Threshold Th1)을 초과하게 되



(그림 4) 피어 탈퇴 시 응답조치 및 그룹 복구

면 형제피어는 살아있는지 여부를 확인하기 위한 메시지를 보낸다. 응답이 없을 경우 형제피어가 떠난 것으로 간주하고 떠난 피어를 대신하여 탈퇴 프로세스를 시작한다.

그림 4는 피어 F가 떠났을 때 탈퇴 프로세스가 어떻게 동작되는지 보여준다. 자신의 그룹 내에서 피어 F가 떠났다는 사실을 인지하면 만형 피어 D는 자신의 부모 피어 A에게 피어 F가 떠났음을 알리는 탈퇴알림 메시지를 보낸다(그림4-①). 그러면 피어 A는 피어 F와 부모-자식으로 연결되어 있던 피어 C에게 만형피어 D가 지목한 피어 E와 연결하여 데이터 조각을 전달하도록 지시한다(그림4-②). 지시를 받은 C는 피어 E와 연결하여 F에게 보내던 것과 마찬가지로 E에게 데이터 조각을 보낸다(그림4-③). 피어 E는 피어 F의 자식피어였던 피어 O, 피어 R과도 연결을 맺어 피어 C에게 받은 조각을 전달해 준다(그림4-④). 피어 E는 피어 B를 부모로 두고 N, Q, U를 자식으로 둔 피어로서 동작하고, 피어 C를 부모로 두고 O와 R을 자식으로 둔 피어로서도 동작한다(이하, 이 논문에서는 대리피어로 부름). 만약 탈퇴한 피어가 만형피어이면서 동시에 책임리스트에 속하는 피어라면 새롭게 만형피어를 맡게 된 피어가 부트스트랩 서버에게 책임리스트를 갱신해 줄 것을 요청한다.

B. 그룹 복구

시스템 전체의 균형과 그룹 내의 안정성을 높이기 위해 떠난 피어의 자리를 복구하는 동작을 수행한다. 그림

4(b)의 대리피어 E는 떠난 피어와 연결되어 있던 피어들에게 그룹복구 메시지가 말단의 피어들에게 전달되도록 메시지 전달을 부탁한다. 이때, 그룹복구 메시지는 세그먼트 조각과 함께 전달된다. 말단에 존재하면서 접속시간이 일정한 값(Threshold Th2) 이상 된 안정적인 피어는 대리피어 E에게 응답 메시지를 보낸다. 이 때 자신의 조건정보(위치, 대역폭 등)를 함께 보낸다. 그러면 대리피어 E는 응답을 보낸 피어들 중 가장 적합한 피어를 자신의 그룹에 참여하도록 선택한다. 대리피어 E는 자신이 선택한 피어 R에게 선택되었음을 알리는 메시지와 피어 R이 연결되어야 할 피어(C, D, E, O)에 대한 정보를 보낸다. 그림 4(c)에서 선택받은 피어 R은 만형피어에게 탈퇴 메시지를 보낸 후 새로운 그룹에 참여하게 된다.

말단에 있는 피어가 탈퇴할 때, 탈퇴한 그룹의 부모그룹의 만형피어가 책임피어가 아니라면 탈퇴한 피어로 인해 생긴 Hole을 새로운 피어로 채울 수가 없다. 따라서 부모그룹의 만형피어는 부트스트랩 서버에게 자식그룹의 Hole을 알리는 메시지를 보낸다. Hole 메시지를 받은 부트스트랩 서버는 새로운 피어의 가입요청 메시지가 들어왔을 경우, 책임피어 대신 Hole 메시지를 보낸 피어를 새로운 피어에게 우선적으로 알려준다. 새로운 피어가 들어옴으로써, 그룹 내에 Hole은 사라지게 된다.

3. 결론

본 논문에서는 라이브 미디어 스트리밍 서비스에 적합한 P2P 오버레이 구조를 제안하였다. 제안한 ToG는 그룹단위의 트리를 형성함으로써 트리의 약점인 churn에 대한 취약성을 극복 하고자 하였고, 트리의 깊이를 줄여 말단에 위치한 피어들에게까지 빠르게 데이터가 전송 될 수 있도록 하였다. 앞으로 ToG를 더욱 스마트한 구조로 만들기 위하여, 물리적 특성(지역성, 대역폭 등)을 고려한 구조설립 방안을 연구할 계획이며, 이 구조에 알맞은 데이터 전달과 데이터 분할 기법에 대해서도 연구 할 예정이다. 또한 OFT, Th1, Th2, 그리고 n의 최적 값을 찾기 위한 시뮬레이션을 계획하고 있으며, 이를 통하여 ToG의 성능을 분석하고자 한다.

참고문헌

[1] F. Wang, Y. Xiong, and Liu, "mTreebone: A Hybrid Tree/Mesh Overlay for Application-Layer Live video Multicast," Proceeding of IEEE ICDCS'07, May, 2007
 [2] Q. Huang, H. Jin, X. Liao, "P2P Live Streaming with Tree-Mesh based Hybrid Overlay," Proceeding of IEEE ICPPW'07, September, 2007
 [3] Chao Liang, Yang Guo, Yong Liu, "Hierarchically Clustered P2P Streaming System," Proceeding of IEEE GLOBECOM'07, November, 2007