

Resilient Video Streaming을 위한 향상된 혼잡 제어 기법

진현석⁰, 김광훈, 박종민, 이창환, 이동만
 한국정보통신대학교 공학부
 {hsjin⁰, nuly17, alrog, leech, dlee}@icu.ac.kr

Enhanced Congestion Control for Resilient Video Streaming

Hyunseok Jin⁰, Gwanhun Kim, Jongmin Park, Changhwan Lee, Dongman Lee
 School of Engineering, Information and Communications University

요약

멀티미디어 데이터를 이종의 수신자에게 안정적으로 전달하고자 하는 것은 인터넷에서 중요한 주제 중의 하나이다. 특히, 최근에 소개된 계층형 MDC(Layered Multiple Description Coding) 기법은 오버레이 멀티캐스트에서 이 문제를 효율적으로 해결하는데 중요한 접근방법이다. 그러나, 이 과정에서 저속 수렴(slow convergence)과 참가 실험(join experiment) 동안에의 손실과 같은 문제가 새롭게 발생하게 되었다. 본 논문에서는 비디오 스트리밍 서비스에서 위의 문제를 해결할 수 있는 효율적인 계층형 멀티캐스트 혼잡 제어 기법을 소개한다. 여기서 제시하는 기법의 가장 특징적인 점은 사용자가 수신할 계층(layer)의 수를 결정하기 위하여 패킷페어(packet-pair)방식에 기반한 수신률 조절 메커니즘을 사용하는 것이다. 결과적으로 본 논문에서는 수신자가 최적의 전송률에 빠르게 수렴하면서도 손실을 최소화할 수 있는 종단-대-종단간 혼잡 제어 기법을 제시한다.

1. 서론

인터넷을 통한 비디오 스트리밍 서비스는 네트워크 대역폭의 증가와 사용자의 다양한 요구에 의해 15년 이상 흥미로운 주제로 다루어져 왔다. 그와 관련된 많은 연구들이 진행되었지만, 특히 멀티캐스트 통신은 현재의 유니캐스트 방식보다 비디오 콘텐츠를 복수의 수신자에게 효율적으로 전송할 수 있는 충분한 잠재력을 보여주었다. 멀티캐스트 라우팅을 지원하기 위해 제안된 방법은 크게 IP 멀티캐스트와 오버레이 멀티캐스트[1]가 있다. 최초에는 IP 멀티캐스트에 관한 연구가 많이 행해졌지만 그것은 확장성과 보급문제, 그리고 상위 계층 기능에 대한 지원에 있어서 문제점을 드러냈다. 반면에 오버레이 멀티캐스트는 현재의 인터넷 기반 환경을 그대로 이용하여 점차적으로 보급될 수 있는 특징을 갖추고 있다. 오버레이 멀티캐스트는 멀티캐스트의 기능을 수행하기 위하여 오직 네트워크에 연결된 종단 호스트만을 사용한다. 종단 호스트는 각자가 자신이 참여하는 그룹의 상태를 유지함으로써 기존에 발생했던 그룹 확장성에 관한 문제를 해결할 수 있다.

전통적인 오버레이 멀티캐스트에서는 데이터 전달을 위해 오직 하나의 전송 트리를 사용하였다. 그러나, 그 방법은 참가 노드들의 빈번한 가입/해지(join/leave)와 실패(failure)에 취약한 점이 있다. 따라서 만약 불특정 다수의 멤버십 변동이 많은 집단에 대해 비디오 스트리밍 서비스를 지원하고자 한다면, 핵심이 되는 사항은 사용자들의 잦은 참가 및 종료에 대

해서도 콘텐츠의 분배를 견고하게 유지하는 것이다. 이를 해결하기 위해 Resilient Peer-to-Peer Streaming[2] 기법이 소개되었다. 이 연구에서는 다수의 트리와 MDC(Multiple Description Coding)를 사용하여 중복(redundancy)을 통한 견고성(resilience)의 효과를 얻는 것을 기반으로 한다. MDC는 하나의 스트리밍 데이터를 여러 개의 독립적인 조각(description)으로 나누어 보내는 코딩 기법이다. MDC로 보내지는 데이터는 계층형 코딩 기법(Layered Coding)[3]과는 달리 어떤 조각(description)의 조합에 대해서도 받을 수 만큼에 해당하는 품질의 스트리밍을 복원해낼 수 있다. 결과적으로 다수의 계층으로 나누어진 비디오 스트림의 조각들을 각각의 다른 전송 경로로 전송함으로써 참가 노드의 트리 이탈에 따라 일부 전송 경로에서 전송 실패가 발생하더라도 다른 경로로 전달되는 데이터로 비디오의 재생이 가능하므로 사용자에게 견고성을 제공해 줄 수 있다.

오버레이 멀티캐스트에서 또 하나의 중요한 문제는 대역폭의 다양성(Heterogeneity)을 지원하는 것이다. 현재의 네트워크는 서로 다른 종류의 네트워크망과 단말들이 혼합된 상태로 존재하며, 이를 위해서는 위에서 설명된 기법에 혼잡제어와 이질적인 수신자를 고려하는 방법을 통합하는 것이 바람직하다. [4]에서는 다양성(heterogeneity)과 견고성을 지원하기 위한 디자인을 MDC를 이용한 다중 전송 방식[2]에 결합하여 사용하며, 부모 노드와 자식 노드의 혼합형 적응방법을 사용한다. 그러나 그 과정에서 또 다른 문제가 생겨나게 되었다.

그것은 저속수렴(slow convergence: 계층을 단계적으로 추가하여 최적의 전송률에 접근하는 시간이 많이 걸리는 현상)과 안정성(stability: 최적의 전송률에서 비디오의 품질변화가 심하지 않아 안정적으로 시청할 수 있는 상태)의 문제이다. [4]에서 제안한 기법은 계층들을 단계적으로 예약함에 따라 최적의 전송률에 느리게 수렴하고, 참가 실험(join experiment: 일정기간의 평형기간 뒤에 자신이 사용할 수 있는 더 많은 대역폭을 발견하기 위해 시험적으로 트리에 가입하여 데이터를 받아보는 것) 과정에서 실패할 경우 불필요한 혼잡을 유발할 수도 있다.

따라서, 본 논문에서는 Resilient Peer-to-Peer Streaming [2]에서 발생하는 위와 같은 문제들을 해결할 수 있는 향상된 혼잡 제어 기법에 대해 논의한다. 여기서 사용자는 현재의 상황에서 최대한 수신할 수 있는 계층(layer)의 수를 결정하기 위하여 패킷 페어(packet-pair)방식[5]에 기반한 수신을 조절 기법을 사용한다. 결과적으로 최적의 전송률에 빠르게 수렴하면서도 손실을 최소화할 수 있는 중단-대-중단간 혼잡 제어 기법을 제시한다.

본 논문은 다음과 같이 구성되어 있다. 2절에서는 관련연구를 소개하고, 3절과 4절은 각각 비디오 스트리밍에서 혼잡제어를 위한 고려사항과 우리가 제안하는 기법을 소개한다. 5절에서 시뮬레이션 결과를 분석하고, 6절에서는 결론과 향후 과제에 대해 기술하도록 한다.

2. 관련 연구

본 장에서는 비디오 스트리밍을 위한 효율적인 오버레이 멀티캐스트 시스템을 위해서, 먼저 다수의 트리 또는 계층형 코딩 기법을 사용한 몇 가지의 관련된 연구를 소개 한다.

Resilient Peer-to-Peer Streaming [2]: 이 접근방법은 단말들의 잦은 멤버십 변동에도 스트리밍 콘텐츠의 분배가 견고하도록 하는 데에 집중을 하고 있다. 견고성(resilience)을 위해, 여기서는 두 가지의 중복(redundancy)을 제안한다. 첫째로 저자는 다수의 분배 트리를 사용하고 있다. 전체 노드를 '풍족한'(fertile, 내부 노드로서 자식 노드를 가질 수 있는) 것과 '황폐한'(sterile, 중단 노드로서 자식 노드를 가질 수 없는) 것의 두 그룹으로 나누어 그것을 노드의 위치 결정에 반영하고 있다. 따라서 트리는 짧고 퍼지는 형태의 모습으로 만들어질 수 있으며 견고함을 보장한다. 둘째로 여기서는 콘텐츠 전송을 위해 MDC를 사용하고 있다. MDC는 오디오나 비디오의 신호를 다수의 분리된 스트림으로 쪼개어 보내는 방식이다. 이때 수신자는 이들 스트림 중의 어떤 집합을 받아 조합하더라도 독립적으로 해석이 가능하여 에러에 강하고, 선택적인 품질 조절이 가능해진다. 중복으로 인한 어느 정도의 오버헤드를 감수한다면, 이 방법은 노드의 잦은 멤버십 변동에 유연하게 대처하고, 사용자에게 맞는 대역폭의 할당과 견고한 스트리밍을 가능하게 해준다.

Resilient P2P using Layered MDC [4]: 이 연구는 기존의 Resilient Peer-to-Peer Streaming에 새로운 적응적(adaptation) 프로토콜을 적용한 것이다. 저자들은 이전에 사용했던 MDC 기법에서 생기는 네트워크의 이질성 문제를 다루기 위해서 계층형 MDC (Layered MDC) 기법을 적용한다. 계층형 MDC 기법은 기본적으로 RLM[3]에서 사용했던 수신을 조절 기법과 비슷하지만 각 계층의 중복 생성을 통해서 이질성 처리와 동시에 견고성을 이루고자 하는 점에서 다르다. 계층형 MDC는 기본 계층들(Base Layers)을 적은 대역폭을 가진 클라이언트들에게 보내고, 기본 계층들과 상위 계층들(Enhancement Layers)을 높은 대역폭을 가진 클라이언트들에게 보냄으로써 이질적인 호스트들의 공정성을 해결한다.

PLM [6]: PLM은 계층형 코딩 기법에 기반한 멀티미디어 전송 어플리케이션의 적응적 수신을 조절 기법이다. 이 방법은 효과적인 계층 추가를 위해 수신자 주도형의 대역폭 측정 기법을 사용하고 있다. 여기서는 모든 라우터가 fair queuing scheduling[7]을 구현하고 있다고 가정하고, 또한 모든 사용자가 독립적이고 비협동적이라고 가정한다. PLM은 IP 멀티캐스트 영역에서 흥미롭고 가능성 있는 방법을 제시하지만 그것을 오버레이 네트워크에 적용하기 위해서는 다른 환경을 고려하여야 한다.

Network Bandwidth Measurement [8]: 네트워크 링크에서 대역폭을 측정하는 방법은 크게 두 종류로 나눌 수 있다: 그것은 수동적(passive) 측정방법과 능동적(active) 측정방법이다. 수동적 측정법은 존재하는 데이터의 전달 결과를 추적하여 값을 측정하는 방법이다. 이것은 주로 트래픽이 적은 무선 네트워크에서의 대역폭 측정을 위해 쓰인다. 반면에 능동적 측정법은 즉각적인 링크의 수용능력과 가용 대역폭을 측정하는 데에 사용된다. 여러 가지 기법들이 소개되었지만 그 중에 우리는 패킷 페어(packet-pair)[5] 기법에 집중을 하도록 하겠다. 패킷 페어 기법은 간단하고, 빠르며, 네트워크에 최소한의 부담을 주면서 측정이 가능한 방법으로 소개되고 있다. 만약 출발지에서 두 개의 패킷을 연달아 보내면 도착지에서는 그 두 패킷 간의 벌어진 간격을 통하여 네트워크의 상태를 추론할 수 있다.

3. 혼잡제어를 위한 고려사항

3.1. 혼잡제어의 정의

일반적으로 혼잡은 송신자가 네트워크로 데이터를 전송하는데 있어, 수신자가 처리할 수 있는 속도 이상으로 보낼 때 발생한다. 혼잡이 발생하면 라우터의 큐가 가득 차기 때문에, 해당 라우터는 어떤 패킷도 처리 하지 못하게 된다. 혼잡 문제를 해결하기 위해서는 송신자가 네트워크 상태에 맞게 송신율을 조절할 수 있어야 하고, 수신자는 혼잡을 알아채고 알려줄 수 있어야 한다.

Transport 계층의 TCP는 'Slow Start'와 'AIMD'같은 혼잡제어 기법을 가지고 있다. 그러나 비디오 스트리밍이 주로 사용하는 UDP는 혼잡제어를 위한 어떤 메커니즘도 갖고 있지 않기 때문에 TCP트래픽과의 형평성(fairness)을 위해서는 혼잡제어에 대한 고려가 필요하다.

3.2. Resilient P2P using Layered MDC에서 혼잡제어를 위한 고려사항

기본적으로 Resilient P2P Streaming에서의 혼잡제어는 다음과 같은 사항을 요구한다. 먼저, 송신자는 혼잡이 일어난 링크의 위치를 알아낼 수 있어야 한다. 실제로, 혼잡은 노드의 외부방향(outgoing) 링크에서 발생할 수도 있고, 내부방향(incoming) 링크에서 발생할 수도 있다. 따라서, 혼잡의 위치에 따라 대응하는 방법도 달라져야 하는 것이다. 둘째로 혼잡제어의 과정에서 자식노드에게 미치는 혼잡의 영향이 최소화 되도록 해야 한다. 오버레이 네트워크에서는 트리를 따라 네트워크의 상태가 전파될 수 있으므로 최대한 자식노드에게 혼잡으로 인한 영향을 덜 끼치는 방법을 연구하는 것이 필요하다.

여기에 더하여, Resilient P2P using Layered MDC에서는 누적형태의 스트림을 사용하는데 따른 수렴시간과 안정성(stability)에 대한 고려가 절실하다. 여기서 수렴시간은 수신자가 단계적으로 계층을 추가하여 혼잡을 일으키지 않는 최적의 상태를 찾아가는 시간을 말하며, 안정성은 수신자가 받아보는 비디오의 품질이 계층의 추가와 삭제에 따른 영향을 최대한 받지 않고 고르게 나타낼 수 있도록 하는 것이다.

4. 제안된 방법

우리가 제안하는 방법은 누적형태(cumulative)의 계층형 코딩 기법과 패킷 페어(packet-pair)를 사용한 가용 대역폭의 측정을 기반으로 한다. 이 방법은 완전한 수신자 주도형의 방식이므로, 혼잡제어를 위한 모든 부담은 수신자 쪽에 있게 된다. 송신자 쪽에서 필요한 가정은 송신자가 누적형태의 계층형 데이터를 보내고 패킷 페어를 통한 측정이 가능하다는 것뿐이다. 이어지는 절에서, 우리는 핵심 메커니즘을 소개하고 그것에 기반한 새로운 기법을 제안하겠다.

4.1. 패킷페어(packet-pair)를 사용한 가용 대역폭 측정

네트워크 링크의 수용능력(capacity)과 가용 대역폭(available bandwidth)을 측정하는 것은 스트리밍 어플리케이션에 있어서 매우 중요한 문제이다. 특히 이것은 서버의 서비스 조정과 혼잡제어에 있어서 더욱 중요성을 가진다. 예를 들어, 실시간 비디오 스트리밍에서 수신자의 가입을 결정하는 허가제어와 같은 순간에, 송신자는 확실한 결정을 위한 사전 지식으로써 위와 같은 정보를 필요로 할 수 있다. 하지만 차별화되고 확실한 서비스를 보장하면서 링크의 수용능력과 가용 대역폭을 측정하는 방법을 개선시키는 것은 매우 힘든 일이다.

측정을 위해 제안된 여러 방법들 중에 패킷페어(packet-pair)는 가용 대역폭 측정을 위한 우수한 방법 중의 하나이다. 패킷페어를 위해서 네트워크의 모든 라우터는 fair scheduler를 구현할 수 있다고 가정한다. 이때 만약 송신자가 두 개의 패킷을 연달아 전송한다면, 수신자는 도착한 패킷 쌍의 간격과 패킷 크기를 이용해서 가용 대역폭을 추론할 수 있다. 주기적으로 패킷 쌍을 전송함으로써, 수신자는 가용 대역폭을 추적할 수 있다[5].

패킷페어 대역폭 측정법의 주요 특징은 크게 두 부분으로 나눌 수 있다. 첫째, 이 방법은TCP와 달리 손실을 요구하지 않는다는 것이다. 패킷페어 대역폭 추론 방법은 손실 측정을 기본으로 하지 않고, 전송 간격 측정을 기본으로 한다. 둘째, 수신자는 병목지점의 큐(bottleneck queue)가 형성되고 과잉(overflow)되기 전에 혼잡을 발견할 수 있다. 가장 간단한 경우에, 혼잡이 발생한 후에 큐를 떠나는 첫 번째 패킷 쌍이 혼잡을 가리키는 신호가 된다.

4.2. 패킷페어를 이용한 빠른 수렴

본 논문에서 제시하는 기법에서 송신자는 기본적으로 누적형태의 계층들을 송신하고, 각각의 계층에 대해서 다중의 경로를 통해 병렬로 패킷페어를 전송하여 대역폭을 측정한다고 가정한다. 즉 모든 패킷페어는 동시에 전송된다. 이때 수신자 측에서 이루어지는 빠른 수렴 과정에 대해서 설명하도록 하겠다.

수신자는 스트리밍의 세션에 참여하자마자, 각각의 계층이 사용중인 대역폭을 이미 안다고 가정한다. 그리고 우리는 능력을 위해 최종 송신자와 수신자 사이가 아닌, 부모와 자식 사이의 가용 대역폭 측정만을 고려한다. L_i 를 i 계층의 데이터 대역폭, B_i 를 i 개의 누적된 계층이 필요로 하는 대역폭이라고 하면 다음이 성립한다.

$$B_i = \sum_{j=1}^i L_j$$

일단 수신자가 스트리밍 서비스에 가입하기 위한 요청을 보내게 되면, 송신자는 수신자에게 각각의 트리에서 연결 가능한 부모노드를 알려주고, 결정적 트리 알고리즘을 이용해서 새로운 노드를 모든 다중 트리에 소속시킨다. 세션의 시작 단계에서 수신자는 단지 기본 계층에만 가입하고 송신자로부터 첫 패킷페어가 오기를 기다린다. 첫 패킷페어를 받을 때, 수신노드는 각각의 다중 경로들로부터 가용 대역폭을 알 수 있다. 이 가용 대역폭 정보를 가지고, 송신자는 각각의 누적 계층 대역폭과 비교한다. 만약 가용 대역폭이 각 트리의 현재 예약된 계층보다 작으면, 수신자는 그 계층을 버린다. 반대의 경우라면 수신자는 그 계층을 추가한다. PP_i 를 i 계층의 가용 대역폭이라고 했을 때, 위의 알고리즘을 표현하면 그림 1과 같다.

```

L1 = 10k, L2 = 20k, L3 = 40k, L4 = 80k
(Li is streaming bandwidth in i layer)

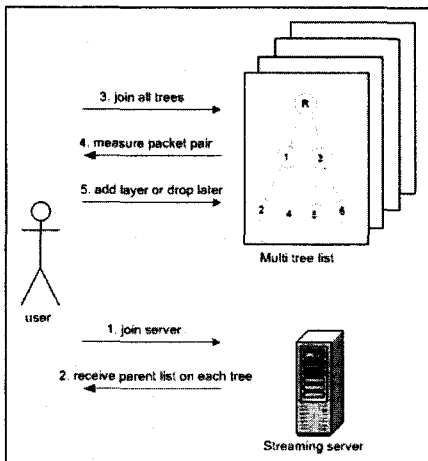

$$B_i = \sum_{j=1}^i L_j$$


(Bi is required bandwidth in i cumulative layers)
PPI is available bandwidth in i layer

WHILE all of layers are checked
{
  IF PPI < Bi THEN
    DropLayer(all);
    BREAK;
  ELSE
    AddLayer(i);
}
    
```

[그림 1] 가용 대역폭 측정을 위한 알고리즘

구체적인 설명을 위해서 간단한 예제를 보도록 하자. 간단한 계산을 위해서 서로 독립적인 10k, 20k, 40k, 80k의 4종류의 description이 있다고 가정한다. 그리고, 총 100k대역폭을 수용할 수 있는 새 노드가 있다. RLM의 경우, 향상된 계층(enhancement layer)에 가입하려고 할 때 마다, 적어도 한번의 시도와 실패가 발생할 수 있다. 따라서 10k의 기본 계층에서 10k, 40k, 80k로 계층을 추가함에 따라 각각의 경우에 참가의 성공 여부에 따라 손실이 발생할 수 있는 것이다. 그러나, 여기서 제안하는 기법에서는 10k의 최초 계층에서부터 가용 대역폭의 범위 안인 20k, 40k까지는 패킷페어의 결과에 따라 손실 없이 바로 수신이 가능해지는 것이다. 참가실험(join experiment)과정에서 손실을 발생시키는 RLM과는 달리, 여기서는 가용 대역폭을 측정하고 참가 여부를 결정할 수 있으므로 최적의 전송률에 수렴하는 데에 있어 손실이 발생되지 않는 것이다.



[그림 2] 빠른 수렴을 위한 계층 추가 절차

4.3. 효율적인 혼잡해결기법

일반적으로 효과적인 혼잡 제어 기법에는 두 가지의 고려사항이 있다. 그것은 혼잡의 탐지(congestion detection)와 혼잡의 해결(congestion resolution)이다.

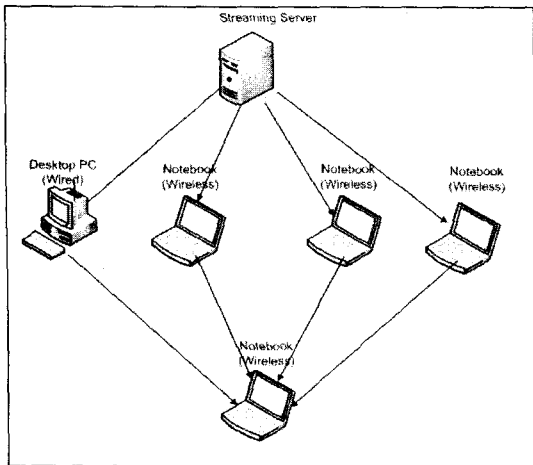
현재 사용하고 있는 시스템에서는 혼잡링크의 정확한 위치를 파악하기 위해서 network tomography[9] 방법을 사용하고 있다. 그것은 혼잡을 세 가지의 경우로 나누고, 해결기법을 적용하고 있다. 여기서 우리가 집중해야 할 부분은 혼잡해결기법을 얼마나 개선할 수 있는가 하는 것이다. 보다 효율적인 계층관리와 서버의 부담을 줄이기 위해 우리는 shedding-layer 기법[4] 대신에 suspending-layer 기법[10]을 제안한다. suspending-layer 기법은 혼잡이 발생했을 때, 기존의 트리 구조를 유지하면서 부모 노드가 자식 노드로 가는 데이터 전송을 일시적으로 멈춘 상태로 유지하는 것을 말한다. 그 후에 만약 나머지 스트리밍 데이터에 손실이 발생하지 않는다면 자식 노드는 바로 adding-layer 기법을 적용할 수 있다. suspending 이후에 이어지는 adding에서는 먼저 suspending-layer의 리스트를 살펴봐야 한다. 만약 거기에 노드 항목이 존재한다면, 자식 노드는 그 노드에 대하여 패킷페어 측정을 하고 가입절차를 밟게 된다. 만약 suspending-layer의 리스트에 항목이 존재하지 않는다면, 일반적인 adding-layer 절차를 수행하게 한다. Adding-layer 기법은 RLM[3]의 join experiment와 흡사한데, 우리는 기존의 adding-layer 기법에 패킷페어 메커니즘을 도입하였다. 패킷페어로부터 추론된 가용 대역폭의 측정값은 자식 노드가 더 많은 스트림을 받을 것인지 말 것인지를 결정할 수 있게 한다. 이전의 join experiment 기법에서는 자식 노드가 부모 노드가 될 후보들을 건네받았을 때, 바로 계층을 추가하고, 그 후에 스트림 데이터의 손실이나 혼잡을 탐지했다. 그러나 우리의 기법에서는 가용 대역폭 측정을 통하여 자식 노드가 새로운 계층을 추가하기 전에 트리에 가입을 할 것인지 말 것인지를 확실히 결정할 수 있다. 결과적으로 우리는 suspending-layer를 통하여 실행 오버헤드를 줄이고 adding-layer에서 발생하는 불필요한 손실을 줄임으로써 더 효율적인 혼잡해결기법을 수행한다.

5. 시뮬레이션

제안된 알고리즘의 테스트를 위해서 우리는 Java 기반의 시뮬레이터를 만들었다. 시뮬레이션에 적용된 특징적인 인자들이 다음의 표에 나열되어 있다.

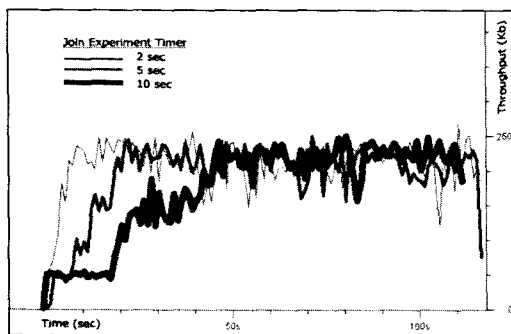
[표 1] 시뮬레이션 환경 인자

Number of nodes	6
Simulation Time	120 sec
Tree Count	4
Each Stripe's Bandwidth	50KB/sec, 50, 100, 100
Total Bandwidth	300KB/sec



[그림 3] 시뮬레이션 구조(Simulation topology)

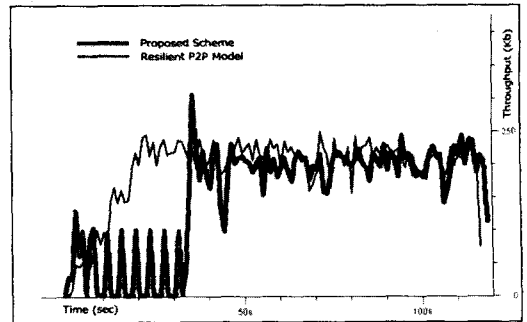
그림 3은 시뮬레이션에서 사용된 네트워크의 topology이다. 전체 환경은 6개의 노드로 이루어져 있다. 시스템은 가변적인 환경을 반영하기 위하여 무선 노드를 포함하고 있다. 가장 아래에 있는 노드가 새롭게 도착하여 가입하려고 하는 노드로 쓰이게 된다. 그것은 상위의 4개의 노드로부터 서로다른 경로로 스트리밍을 받게 된다. 가입과 동시에 제안된 기법이 수행되고 스트리밍을 받게 되면 새롭게 가입한 노드에서 트래픽이 측정되게 된다.



[그림 4] Join Experiment Timer에 따른 수렴시간의 변화

Join experiment timer란 계층형 코딩에서 다음 단계의 계층을 수신하기 위하여 시형적으로 데이터를 받아보는 주기를 말한다. 그림 4에서 보듯이, 우리는 Resilient P2P Streaming using Layered MDC에서 서로 다른 join experiment timer를 사용하였을 때 발생하는 수렴속도를 측정하여 보았다. 결과와 같이 더 짧은 join experiment timer를 사용하였을 때, 최적의 전송률에 대한 수렴속도도 빨라진다는 것을 알 수 있었다. 하지만 실제로 짧은 timer를 사용하면 현재 네트워크의 상황에

반응하기가 어렵고, 반면에 긴 timer를 사용하면 성능을 떨어뜨리는 결과를 초래하기가 쉽다. 그래서, 우리는 비교를 위해 RLM에서 최소값으로 사용하고 있는 5초의 timer를 사용하였다.



[그림 5] Resilient P2P Model과 제안된 기법 간의 수렴시간 비교

그림 5는 Resilient P2P Model과 우리가 제안한 기법 간의 수렴속도 차이를 보여준다. Resilient 모델의 join experiment timer는 5초이므로 4개의 전송경로로부터 스트리밍을 추가하여 수렴지점에 다다르기 위해 총 20초의 시간이 걸린다. 그리고 우리가 제시한 모델은 수렴지점에 이리는데 약 30초가 걸리는 것을 알 수 있다. 여기서 먼저, 이번 모델에서 가용 대역폭을 측정하기 위해 우리는 오픈 소스의 대역폭 측정 도구를 사용하였다. 여러 번의 비교와 실험을 통하여 선택한 결과였지만 실제로 이 방법은 대역폭을 측정하는데 있어서 20초 가량의 시간이 소비되었다. 결과적으로는 resilient 모델보다 전체 수렴시간이 더 길게 나온 결과가 확인되고 있다. 하지만 전체 수렴과정 중 가용 대역폭의 측정시간이 대부분을 차지하고, 그 측정이 끝난 이후에는 바로 사용 가능한 만큼의 예약이 동시에 이루어지는 것을 확인할 수가 있다. 따라서 원래의 의도대로 좀더 빠르고 효율적으로 가용 대역폭을 측정할 수 있는 방법을 찾아내 도입하거나 현재의 기법을 개선할 수 있다면, 대역폭 측정을 통한 스트리밍의 추가와 삭제가 분명 효과가 있다는 것을 확인할 수 있었다.

6. 결론 및 향후 과제

우리는 비디오 스트리밍에서의 혼잡제어로부터 연구를 시작했다. 그리고 거기에서의 향상된 혼잡제어 기법을 디자인하였다. 그것은 기본적으로 누적형태의 계층형 멀티캐스트 기법과 가용 대역폭의 측정을 위한 패킷페어 기법을 사용하고 있다. 이 방법들은 수신자의 이질성을 고려함과 동시에 RLM의 join experiment와 같은 절차를 없애으로써 혼잡제어에 도움을 줄 수 있다. 결론적으로 우리는 기존의 Resilient P2P using Layered MDC에서 사용되는 혼잡제어의 기법을 향상시키는

방법을 제안하였고, 이것을 통하여 보다 나은 비디오 스트리밍을 구현할 수 있다.

무엇보다도 우리의 접근 방법은 가용 대역폭의 측정결과에 의존적이므로, 혼잡제어를 위한 더 효과적인 지원을 위해서는 패킷페어 또는 다른 대역폭 측정 기법을 보다 정확하고 빠르게 개선할 필요가 있다. 그리고 시뮬레이션에서는 테스트를 위한 가정과 제약사항이 있었는데, 실제의 스트리밍 서비스에 대한 영향을 고려하기 위해서는 단순 트래픽이 아닌 비디오 데이터를 사용한 전체 시나리오의 구상과 구현이 필요하다. 또한 adding/shading layer 단계의 더욱 구체화된 구현이 혼잡제어를 하는 데 도움을 줄 수 있을 것이다.

참고 문헌

- [1] Y. Chu, S. G. Rao, and H. Zhang, "A Case for End System Multicast," In Proc. ACM SIGMETRICS, June 2000.
- [2] V. Padmanabhan, H. Wang, and P. Chou, "Resilient Peer-to-Peer Streaming," in Proc. IEEE ICNP, Nov 2003
- [3] S. McCanne and V. Jacobson, "Receiver-Driven Layered Multicast," Proc. of ACM SIGCOMM, Sep 1996, pp.117-30.
- [4] V. N. Padmanabhan, H. J. Wang, and P. A. Chou, "Supporting Heterogeneity and Congestion Control in Peer-to-Peer Multicast Streaming," In IPTPS, February 2004.
- [5] S. Keshav, "Congestion Control in Computer Networks," PhD thesis, EECS, University of Berkeley, CA 94720, USA, September 1991.
- [6] A. Legout and E. Biersack, "PLM: Fast Convergence for Cumulative Layered Multicast Transmission Schemes," Proc. of SIGMETRICS, 2000, pp. 13-22.
- [7] J. Bennett and H. Zhang, "Wf2q: Worst-case fair weighted fair queueing," In Proceedings of IEEE INFOCOM'96, San Francisco, CA, USA, March 1996.
- [8] Fang Qi, Jin Zheng, Weijia Jia, and Guojun Wang "Available Bandwidth Measurement Schemes over Networks", ICCM C 2005, LNCS 3619, pp931-940
- [9] T. Bu, N. Duffield, F. L. Presti, and D. Towsley, "Network Tomography on General Topologies," In Proc. ACM SIGMETRICS (June 2002).
- [10] Z. Zhang and V. O. K. Li, "Router-Assisted Layered Multicast," Proc. of IEEE ICC, 2002.
- [11] V. K. Goyal, "Multiple Description Coding: Compression Meets the Network," IEEE Signal Processing Magazine, pages 74-93, September 2001
- [12] A. Ganjam and H. Zhang, "Internet Multicast Video Delivery," Invited Paper, Proc. of the IEEE, VOL. 93, NO. 1, January 2005.
- [13] P. A. Chou, H. J. Wang, and V. N. Padmanabhan, "Layered multiple description coding," in Proc. Packet Video Workshop, Apr. 2003.
- [14] A. Matrawy and I. Lambadaris, "A Survey of Congestion Control Schemes for Multicast Video Applications," IEEE Communications Surveys & Tutorials, Fourth Quarter 2003.
- [15] R. Gopalakrishnan et al., "A Simple Loss Differentiation Approach to Layered Multicast," Proc. of IEEE INFOCOM, 2000.