

P2P 네트워크에서 연속형 미디어 콘텐츠의 분산형 배포 기법

권진백* · 엄현영** · 이정배***

요약

P2P(peer-to-peer) 모델은 클라이언트-서버 모델의 서버 링크 병목현상을 해결하기 위한 대안으로 고려되고 있다. 본 연구에서, 우리는 P2P 네트워크에서 연속형 미디어의 배포 문제 중 두 가지를 다루었다. 첫째는 다중-소스 스트리밍 세션을 위한 미디어 데이터의 전송 스케줄링 문제이다. 버퍼링 지연을 최소화시키는 정교한 분산 전송 스케줄링으로서 FSS(fixed-length slotted scheduling)라는 기법을 제안한다. 두 번째 문제는 스스로 성장하는 P2P 시스템에서 미디어 콘텐츠의 배포 속도 향상 문제이다. 이를 위해 우리는 시스템의 스트리밍 용량이 증가하는 속도를 가속시키는 FAST라 불리는 기술을 제안한다.

A Distribution Scheme for Continuous Media Contents over Peer-to-Peer Networks

Jin Baek Kwon* · Heon Young Yeom** · Jeong Bae Lee***

ABSTRACT

A peer-to-peer model is very useful in solving the server link bottleneck problem of a client-server model. In this work, we discuss the problems of distributing multimedia content over peer-to-peer network. We focus on two problems in peer-to-peer media content distribution systems. The first is the transmission scheduling of the media data for a multi-source streaming session. We present a sophisticated scheduling scheme called fixed-length slotted scheduling, which results in minimum buffering delay. The second problem is on the fast distribution of media content in the peer-to-peer system that is self-growing. We propose a mechanism accelerating the speed at which the system's streaming capacity increases, called FAST.

키워드 : P2P 시스템(P2P Systems), 미디어 스트리밍(Media Streaming), 전송 스케줄링(Transmission Scheduling), 분산 스트리밍(Distributed Streaming)

1. 서론

연속형 미디어가 가까운 미래에 인터넷 트래픽의 상당한 부분을 차지하게 될 것으로 예견되고 있다. 연속형 미디어 스트리밍에 대한 기존 연구의 대부분이 클라이언트-서버 모델에 기반을 두고 있다. 미디어 스트리밍은 많은 대역폭을 요구하기 때문에 전통적인 유니캐스트(unicast) 클라이언트-서버 모델이 사용된다면 서버 네트워크 대역폭의 부족현상이 심각해진다. 따라서, 이러한 모델에 기초한 스트리밍 서비스는 낮은 확장성(scalability) 때문에 비용이 비싸지게 된다. 단일 소스 멀티캐스트는 하나의 스트림이 여러 클라이언트에게 전달되는 솔루션으로서 IP 멀티캐스트가 대표적이다. 하지만, IP 멀티캐스트가 안고 있는 난해한 문제들과 혼잡 제

어와 신뢰성 같은 상위 계층의 기능 지원 문제로 인해 실제 적용이 늦어지고 있다. 이러한 이유로 멀티캐스트에 대한 수많은 연구들이 이루어졌음에도 아직도 전통적인 유니캐스트 스트리밍이 보편적으로 사용되고 있다[12, 13]. 그러므로, 클라이언트 수에 대한 확장성을 가지면서 기존의 IP 프로토콜에 적용될 수 있는 저비용의 미디어 콘텐츠 배포 시스템의 설계는 도전할 만한 연구 주제이다.

P2P 모델은 서버 링크 병목현상을 해결하기 위한 이상적인 대안이다. P2P 모델에서, 미디어 콘텐츠는 클라이언트 자신들의 대역폭을 이용하여 배포된다. 다시 말하면, 클라이언트들은 자신의 자원을 전체 P2P 시스템에 제공하고, 다른 클라이언트들에 의해 제공된 자원을 이용한다. 더 구체적으로 말하자면, 특정 미디어 파일을 소유한 *제공 피어*(supplying peer)는 그 파일을 원하는 *요청 피어들*(requesting peers)에게 그 파일을 스트리밍해줄 수 있다. 피어들이 콘텐츠의 공유를 위해 협력하기 때문에, 데이터 트래픽은 어떤 특정 사이트에 집중되지 않는다. P2P 시스템에서는 콘텐츠를 소유

* 이 논문은 2003년도 서울대학교 교내학술연구비 지원을 받았음.

† 정 회원 : 서울대학교 컴퓨터정보학부 교수

** 정 회원 : 서울대학교 컴퓨터공학부 교수

*** 종신회원 : 서울대학교 컴퓨터정보학부 교수

논문접수 : 2004년 4월 21일, 심사완료 : 2004년 11월 9일

하고 서비스하는 중앙 서버가 존재하지 않고 피어들은 평등한 입장에서 동작하는 것이 일반적이다. 콘텐츠들이 최초에 어떻게 P2P 시스템에 존재하는지는 논의로 하기로 하고, 본 논문에서는 언급하지 않는다. 최초에 콘텐츠를 보유한 종자 피어들(seed peers)이 있다고 가정한다. P2P 콘텐츠 배포 시스템의 예에는 Napster[2], Gnutella[1] 등이 있다.

최근 몇 년간 P2P 시스템에 대한 많은 연구가 있어 왔다 [3, 7-9, 11]. 그런 연구들은 주로 범용 P2P 시스템의 데이터 파일 검색과 저장장치 관리 기술에 초점을 맞추고 있다. P2P 네트워크 위에서 연속형 미디어의 배포에 대한 문제는 최근 2년 간 [4-6, 10]에서 최근에 연구되었다. 하지만, P2P 스트리밍 시스템에 대한 연구들은 아직 초기 단계에 있고 성능 향상과 모델의 일반화에 있어 연구의 여지가 남아있다.

본 논문에서, P2P 미디어 콘텐츠 배포 시스템의 두 가지 문제를 해결하기 위한 기술을 제안하고 있다. 먼저, 다중-소스 P2P 스트리밍 세션을 위한 데이터 전송 스케줄링 문제를 다루었다. 다시 말하면, 하나의 요청 피어와 다양한 송신 대역폭을 갖는 제공 피어들의 집합이 주어졌을 때, 전송 스케줄링 문제는 각 제공 피어와 연결된 다중 채널로 미디어 데이터 세그먼트를 전송하는 방법이다. 첫 번째 세그먼트의 스트리밍의 시작과 재생 시작 사이의 시간 간격을 버퍼링 지연이라고 하는데, 이 것은 데이터 세그먼트의 전송 스케줄에 의해 결정된다. 다중 채널 스트리밍에서 전송 스케줄에 대한 문제는 Xu 등에 의해 [10]에서 이미 다루졌다. 여기서, Xu 등이 해결 방법으로서 제안한 OTS는 세그먼트 길이가 동일한 경우에는 최적이다. 우리는 버퍼링 지연을 더 줄인 FSS(fixed-length slotted scheduling)라는 또 다른 스케줄링 기법을 제안한다. FSS는 OTS와는 달리 일정한 길이의 슬롯에 기초해 각 채널 대역폭에 따라 미디어 파일을 다양한 길이의 세그먼트들로 나누는데, 길이는 그 세그먼트가 전송되는 채널의 대역폭과 슬롯 길이에 의해 결정된다. 이런 방법은 미디어 데이터를 가능한 순서대로 받아 버퍼링 지연을 줄이는 것을 목적으로 한다. 두 번째 문제는 미디어 콘텐츠를 전체 시스템에 빠르게 배포하는 방법에 관한 것이다. 최초에, 어떤 콘텐츠를 가진 단 몇 개의 종자 피어들이 제공 피어들로서 존재하고, 비종자 피어들은 요청 피어들로서 그 콘텐츠를 요청한다. 요청 피어는 그 콘텐츠의 수신에 끝난 후에는 제공 피어가 될 수 있기 때문에, P2P 시스템의 서비스 용량은 시간이 지남에 따라 커지게 된다. 하지만, 초기에는 적은 수의 제공 피어들이 존재하기 때문에, 그 시스템이 낮은 요청률을 감당할 수 있다. 우리는 본 논문에서 P2P 시스템 용량이 증가하는 속도를 가속시키는 FAST라는 기술을 제안한다. FAST는 콘텐츠를 제공하기 위한 세밀한 조건을 제시하여, 콘텐츠 제공 가능한 피어의 개수를 증가시킴으로써 배포의 속도를 가속시킨다.

이 후 논문은 다음과 같이 구성된다. 2장에서 본 논문에서 가정하는 P2P 스트리밍 모델을 제시하고, 3장에서 전송 스케

줄링 기법이 소개된다. 4장에서는 배포 가속 기법을 제안하고, 5장에서 제안된 두 기법의 성능을 보여준다. 마지막으로 6장에서 본 논문을 결론짓는다.

2. P2P 스트리밍 시스템 모델

이 절에서는 본 논문에서 가정하는 P2P 미디어 스트리밍 모델을 제시하기로 한다. 우리의 모델은 기존 연구들의 가정 모델에 비해 더 현실적이고 일반적이다. 하나의 요청 피어는 제공 피어 후보 집합으로부터 자신을 위한 제공 피어들을 선택하고, 선택된 각각의 제공 피어와 통신 채널을 설정한 후 전송 스케줄링 기법에 따라 각 제공 피어에게 데이터 세그먼트를 요청한다. 그 요청 피어는 다중 채널로부터 데이터를 수신 받은 후에 그 콘텐츠의 제공 피어가 될 수 있다. 또한, 하나의 제공 피어는 동시에 여러 요청 피어들을 서비스할 수 있다고 가정한다. P2P 시스템에서 콘텐츠를 검색하는 문제는 또 하나의 연구 주제이므로, 우리는 적당한 콘텐츠 검색 기술을 이용해 제공 피어의 후보 집합을 얻을 수 있고 각 피어의 대역폭 사용량에 대한 정보를 알 수 있다고 가정한다.

r 는 미디어 콘텐츠의 재생 비트율을 나타내고, 각 요청 피어 P_r 은 $R_{in}(r)$ 의 수신 대역폭과 $R_{out}(r)$ 의 송신 대역폭을 갖는다. 본 논문에서는 서술의 용이성을 위해 재생율이 r 인 하나의 미디어 콘텐츠의 배포 문제를 다루지만, 재생율이 각각 다른 여러 콘텐츠의 문제로 그대로 확장할 수 있다. 피어들의 수신 대역폭과 송신 대역폭은 각각 다르고, $0 < R_{in}(r) \leq \gamma$ 이고 $R_{out}(r) > 0$ 이라 가정한다. 이러한 가정들은 [10]의 가정보다 더 약하다. 피어들 간 데이터 전송 중에 발생할 수 있는 끊임(jitter)과 지연(delay)에 따른 전송계층 수준의 QoS 문제는 멀티미디어 스트리밍의 초기 연구의 주요한 주제로서 많은 연구가 이루어졌고[14-17], 이에 대한 연구는 본 논문의 범위를 벗어난다. 따라서, 본 연구에서는 네트워크 수준에서 QoS가 제공된다는 가정 하에, 피어 간 형성된 연결은 일정한 대역폭을 보장한다고 가정한다. 또한, 미디어 파일은 CBR(constant-bit-rate)으로 인코딩되었다고 가정한다.

3. 전송 스케줄

이 절에서는 미디어 데이터의 전송 스케줄 문제를 다룬다. 문제는 다음과 같이 기술된다:

하나의 요청 피어 P_r 와 n 개의 채널에 대해서, P_r 의 연속 재생을 보장하면서 버퍼링 지연을 최소화시키기 위한 콘텐츠의 각 데이터 세그먼트를 전송할 채널 선택과 세그먼트들의 전송 순서를 찾는 문제

P_r 을 위한 제공 피어의 선택 문제는 3장에서 다루기로 한

다. 대역폭이 콘텐츠의 재생을 r 인 한 개의 채널에서 스트리밍을 받는 단순한 경우를 고려해 보자. 만약 데이터가 콘텐츠 파일의 처음부터 끝까지 순서대로 전송된다면, 이론적인 버퍼링 지연은 0일 될 것이다. 하지만, 데이터가 r 보다 작은 대역폭의 여러 채널로 전송된다면, 문제는 복잡해지고 버퍼링 지연을 수반하게 된다. 왜냐하면 이 경우 데이터의 전송 순서가 재생순서와 일치하지 않기 때문이다. 따라서 버퍼링 지연을 줄이기 위해 잘 설계된 미디어 전송 스케줄이 필요하게 된다.

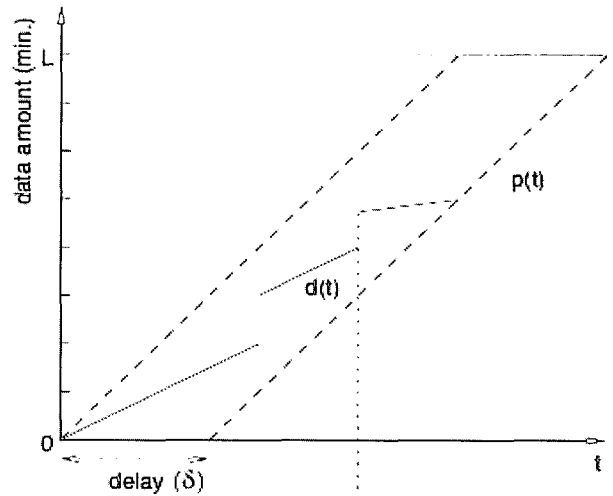
함수 $p(t)$ 는 재생이 시작된 후 t 초 동안 재생된 데이터량으로 정의하고, $d(t)$ 는 스트리밍이 시작된 후 t 초 동안 수신된 미디어 파일의 처음부터 연속된 데이터량으로 정의하자. 미디어 파일이 CBR로 인코딩되었다고 가정했으므로, 본문에서 데이터량은 시간(초)로 표현된다. 즉, 데이터량 k 는 k 초 동안 재생되는 데이터량을 의미한다. 미디어 콘텐츠의 길이가 L 초일 때, 대역폭이 $r/2$ 인 두개의 채널로 길이가 $L/4$ 인 네 개의 세그먼트를 전송하는 문제를 고려해 보자. 첫 번째 채널로 첫 번째 세그먼트 S_1 과 두 번째 세그먼트 S_2 가 차례로 전송되고, 두 번째 채널로 세 번째와 네 번째 세그먼트가 차례로 전송되는 스케줄을 가정하자. 그러면, 요청 피어가 S_1 의 다운로드를 완료하는 시점에 S_3 의 다운로드도 완료된다. S_2 가 아직 다운로드되지 않았으므로, S_1 과 S_3 는 연속된 데이터가 아니다. 따라서, 실제로 수신된 총 데이터량은 $L/2$ 이지만, $d(t)$ 는 S_1 의 데이터만 포함하기 때문에 정의에 의해 $L/4$ 가 된다. 결국, 단일 채널 스트리밍의 경우 $d(t)$ 는 1의 기울기로 직선으로 증가하겠지만, 다중 채널 스트리밍의 경우 $d(t)$ 는 불연속적으로 증가한다.

연속 재생을 보장하기 위해, 다음 조건이 만족되어야 한다:

$$\forall t \geq 0, d(t) \geq p(t) \tag{1}$$

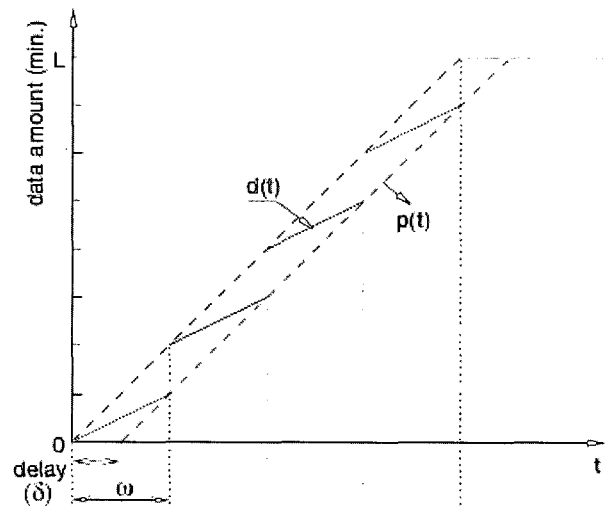
(그림 1)(a)은 OTS[10]의 전송 스케줄 예에 대한 $p(t)$ 와 $d(t)$ 의 증가 형태를 보여주고 있다. 여기서, 재생 중이라면 $p(t)$ 는 기울기가 1인 직선이 되고, 재생이 시작되기 전이거나 끝난 후에는 기울기가 0인 직선이 된다. 반면, $d(t)$ 는 수신 스케줄에 따라 가변적인 함수이다. 그림에서 대역폭이 각각 $r/2, r/4, r/8, r/8$ 인 네 개의 채널이 있고, 요청 피어는 여덟 개의 $L/8$ 초-세그먼트들의 전송을 OTS에 따라 스케줄한다. 그림에서 보듯이, 연속 재생을 보장하기 위한 식 (1)의 조건을 만족시키기 위해 δ 초의 버퍼링 지연이 요구된다. 이 예에서 OTS의 δ 는 $3L/8$ 이다. 그러므로, $p(t)$ 는 다음과 같이 표현될 수 있다.

$$p(t) = \min\{L, \max\{t - \delta, 0\}\}, \text{ for } t \geq 0, \tag{2}$$



ch.1	1	2	4	8
ch.2	3		7	
ch.3	6			
ch.4	5			

(a) OTS[10]



ch.1	1	5	9	13
ch.2	2	6	10	14
ch.3	3	7	11	15
ch.4	4	8	12	16

(b) FFS

(그림 1) 전송 스케줄

여기서 δ 는 버퍼링 지연으로서 직선 점선($y=t$)과 $d(t)$ 의 최대 수직 거리이다. $d(t)$ 가 직선의 형태에 근접할수록, 버퍼링 지연은 줄어든다는 사실을 알 수 있다. 잘 설계된 스케줄링 기법은 $d(t)$ 를 직선에 가깝게 증가시키는 기법이다. 그래서, 우리는 FSS(fixed-length slotted scheduling)이라는 전송 스케줄링 기법을 제안한다.

$d(t)$ 를 직선으로 증가시키기 위해서, 요청 피어 P_r 은 데이터를 재생순서대로 수신해야 한다. 이 원칙에 바탕을 두고, FSS는 각 채널을 대역폭에 상관없이 동일한 길이의 슬롯으로 나누고 데이터를 슬롯에 할당한다. 하나의 슬롯 길이의 데이터 블록을 FSS의 세그먼트라 한다. 각 채널의 대역폭은 다양하고 슬롯 길이는 일정하므로, 세그먼트의 길이는 그 세그먼트가 할당된 채널 대역폭에 따라 다양하다. 세그먼트들은 라운드-로빈 방식으로 채널들에 순서대로 할당된다. 슬롯의 길이가 w 이고 채널 i 의 대역폭이 B_i 라 하면, 채널 i 에 할당되는 세그먼트의 길이는 wB_i 가 된다. (그림 1)(b)가 FSS의 개념을 보여주고 있다. 여기서 채널의 개수와 각 채널의 대역폭은 (그림 1)(a)와 동일하다. 그림의 예에서 슬롯 길이 w 는 $L/4$ 이고, FSS의 버퍼링 지연은 $L/8$ 로서 (그림 1)(a)의 OTS의 버퍼링 지연의 1/3에 불과하다.

FSS의 버퍼링 지연 δ 을 구해보자. 채널의 개수가 n 일 때, 총 수신 채널 대역폭 $B^* = \sum_{i=1}^n B_i$ 로 정의된다. 그러면, $d(t)$ 는

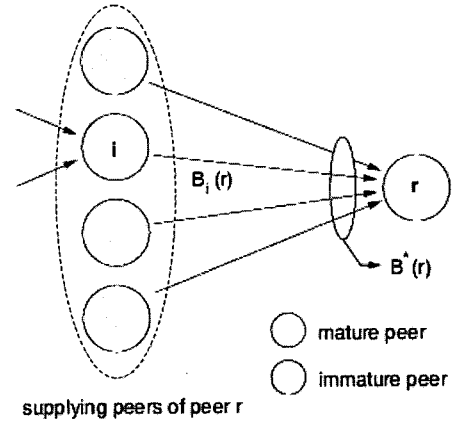
$$d(t) = \min \left\{ \frac{B_1}{\gamma} t + \frac{B^* - B_1}{\gamma} \left\lfloor \frac{t}{\omega} \right\rfloor \omega, L \right\} \quad (3)$$

식 (2)의 $p(t)$ 와 식 (3)의 $d(t)$ 로 식 (1)을 확장하여 δ 에 대해 정리하면, 최소 버퍼링 지연을 구할 수 있다. 상세한 유도 과정은 부록에 수록하였다. 최소 버퍼링 지연 δ 는

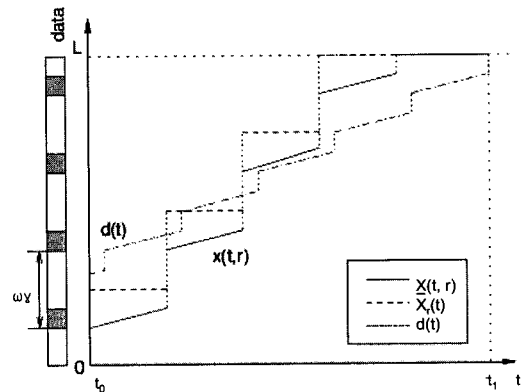
$$\delta = \left(\frac{\gamma}{B^*} - 1 \right) L + \left(\frac{B^* - B_1}{\gamma} \right) \omega, \text{ if } B^* \leq \gamma \quad (4)$$

이 된다. 위 수식으로부터, δ 는 B_1 과 w 에 의존한다는 것을 알 수 있고, B_1 이 클수록, w 가 작을수록 δ 는 작아진다. 그러므로, FSS는 버퍼링 지연의 최소화를 위해 대역폭이 가장 큰 채널을 멀티-채널 세션의 첫 번째 채널로서 선택한다. 버퍼링 지연이 슬롯 길이(w)에 비례하므로, 이론적으로 슬롯 길이가 아주 짧다면 다중 채널들은 마치 B^* 의 대역폭을 갖는 단일 채널처럼 동작하게 된다. 이것이 버퍼링 지연이 0인 다중-채널 스트리밍 문제의 최적 해법이다. 하지만, 슬롯의 길이는 인터넷의 대역폭 변동과 전송 처리 등의 오버헤드를 감당할 수 있을 정도로 길어야 한다. 또한, 슬롯 길이가 하부

물리적 네트워크의 최대 패킷 크기보다 너무 작으면, 낮은 네트워크 활용율로 인해 대역폭의 낭비를 초래하게 된다. 따라서, 슬롯 길이는 신중하게 결정되어야 할 시스템 파라미터이다.



(a) Semi-mature Peer



(b) Data Growth of Peer

(그림 2) 후보 피어와 요청 피어의 수신 데이터

4. 배포 가속 기법

[10]에서, 어떤 미디어 콘텐츠의 제공 피어의 후보 집합은 그 미디어 파일 전체를 가지고 피어들로만 구성된다. 하지만, 미디어 파일을 다운로드 중인 피어들도 다른 요청 피어들에게 그 콘텐츠를 제공할 수 있다. 시스템 용량은 후보 집합의 크기에 비례하므로, 후보 집합을 가능한 한 크게 만드는 것이 유리하다. 어느 미디어 콘텐츠의 후보 집합이 크다면 그 콘텐츠의 요청 피어가 충분한 수신 대역폭을 확보하기가 쉬울 것이고 또한 그 요청 피어는 상대적으로 빠른 시간 안에 후보 집합에 포함될 것이다. 그러므로, 배포 가속 문제의 목적은 후보 집합의 크기 증가 속도를 가속시키는 것이고, 본 논문에서는 후보 피어가 될 수 있는 피어들을 가능한 한 많

이 발굴함으로써 그 목적을 달성하고자 한다.

미디어 파일 전체를 가진 피어를 *완숙(mature) 피어*, 미디어 파일을 다운로드 중인 피어를 *미숙(immature) 피어*라고 정의한다. (그림 2)(a)는 다중 채널 스트리밍 세션을 구성하는 요청 피어 P_r 과 제공 피어들을 보여주고 있다. 회색 원은 완숙 피어를, 흰 원은 미숙 피어를 나타낸다(위에서 언급했듯이 경우에 따라 미숙 피어도 제공 피어가 될 수 있다). $d_i(t)$ 를 피어 P_i 의 $d(t)$ 라 정의하고, 또 하나의 함수를 다음과 같이 정의하자.

$x_i(t, r)$: P_i 가 요청 피어 P_r 의 제공 피어로 선택되었다고 할 때, 시간 t 에 P_r 로부터 전송 요청을 받는 데이터의 미디어 파일 내에서의 절대 위치

FSS에 따르면, 제공 피어는 미디어 데이터를 연속적으로 전송하지 않고, 미리 정해진 패턴으로 일정한 크기의 데이터를 건너뛰면서 전송한다. (그림 1)(b)의 예를 고려하면, 두 번째 채널은 $\gamma/4$ 길이의 네 개의 세그먼트, S_2, S_6, S_{10}, S_{14} 를 전송한다. 따라서, (그림 2)(b)에서 보이는 것처럼, $x_i(t, r)$ 은 기운 계단형 증가 함수의 형태를 보이고, 실선의 증가율은 $B_i(r)/\gamma$ 가 된다. 여기서 $B_i(r)$ 은 P_r 의 i 번째 채널 대역폭이고, 세로 축의 직사각형의 회색 부분은 콘텐츠 파일에서 해당 세그먼트의 위치를 보여준다. 또한, 그 함수에서 계단 단단의 넓이는 슬롯 길이와 같다. P_i 가 완숙 피어라면 $d_i(t) = L$ 이고, 아니라면 식 (3)에 의해 결정된다. (그림 2)(b)는 P_i 가 미숙 피어일 때의 $d_i(t)$ 도 함께 보여주고 있다. $x_i(t, r)$ 이 $d_i(t)$ 를 가로지르는 것은 P_r 이 P_i 가 아직 수신하지 못한 데이터를 요청한다는 것을 의미한다. 그러므로, 이 경우 P_i 는 P_r 의 제공 피어가 될 수 없다. 미숙 피어 P_i 가 P_r 의 제공 피어가 되기 위해서는 다음 조건이 만족되어야 한다. t_0 는 현재 시간을 의미한다.

$$\forall t \geq t_0, d_i(t) \geq x_i(t, r) \quad (5)$$

이 조건을 만족하는 미숙 피어들을 P_r 의 *준완숙 피어*라고 한다. 결과적으로, P_r 의 후보 집합은 P_r 의 완숙 피어들과 준완숙 피어들로 구성된다.

$d_i(t)$ 는 세션을 구성하는 시점에서 결정되지만, $x_i(t, r)$ 은 결정될 수 없다. 왜냐하면, P_r 이 그 세션의 제공 피어들의 선택 작업이 완료되기 전에는 $x_i(t, r)$ 이 결정될 수 없기 때문이다. 이런 이유로, $x_i(t, r)$ 의 상한(upper-bound) 함수 $\bar{x}_r(t)$ 를 대신 사용한다. $B(r)$ 을 P_r 의 스트리밍 세션의 총 수신 대역폭이라고 하면, P_r 은 매 슬롯마다 $\omega B^*(r)$ 크

기의 데이터를 수신할 것이다. 즉, 시점 t_k 가 현재 시간 t_0 으로부터 k 번째 슬롯에 위치하고 있을 때,

$$(k-1)B^*(r) \leq x_i(t_k, r) < kB^*(r) \leq kR_{in}(r)$$

이 성립한다. 여기서 $R_{in}(r)$ 은 P_r 의 사용가능한 수신 대역폭이다. $B^*(r)$ 또한 미리 결정할 수 없기 때문에 더 느슨한 상한 $kR_{in}(r)$ 을 사용한다. 따라서, 상한 함수 $\bar{x}_r(t)$ 는 다음과 같다.

$$\bar{x}_r(t) = R_{in}(r) \cdot \lfloor (t-t_0)/\omega \rfloor \omega$$

(그림 2)(b)에서, 계단형 점선이 $\bar{x}_r(t)$ 를 나타낸다. 그러므로, 식 (4)를 만족하기 위한 충분조건은 $\forall t \geq t_0, d_i(t) \geq \bar{x}_r(t)$ 가 된다. 이 조건이 미숙 피어 P_i 의 준완숙 피어 여부를 판단하는 기준으로 사용될 수 있다. 이 조건은 충분조건이므로, 이 것이 만족되지 않았다고 해서 P_i 가 준완숙 피어가 아니라는 것을 의미하지는 않는다. 다만, 이 조건이 만족되면 P_i 가 준완숙 피어라는 것을 보장할 수 있다.

제공 피어 선택 과정은 다음과 같다. P_i 은 요청한 콘텐츠의 후보 집합 C_r 을 결정한다. 이 C_r 의 각 피어는 완숙 피어 또는 준완숙 피어이고 사용가능한 송신 대역폭을 가지고 있다¹⁾. C_r 를 결정하기 위해 P_r 은 조건 $\forall t \geq t_0, d_i(t) \geq \bar{x}_r(t)$ 의 만족 여부를 검증함으로써 각 미숙 피어 P_i 가 준완숙 피어인지를 알아낸다. 첫 번째 채널 대역폭이 클수록 FSS의 버퍼링 지연이 짧아지므로(식 (4)), C_r 중에서 사용가능한 송신 대역폭이 가장 큰 피어를 첫 번째 제공 피어로 선택한다. 이 과정을 총 수신 대역폭 $B^*(r)$ 이 $R_{in}(r)$ 과 같아질 때까지 반복한다. 정형화된 알고리즘은 (그림 3)에 제시되어 있다.

P2P 시스템이 감당할 수 없을 정도로 많은 요청 피어들이 있는 경우에 요청 피어들은 충분한 대역폭을 확보할 수 없을 수도 있다. 이러한 경우에, P_r 이 택할 수 있는 정책은 세 가지가 있을 수 있다. 첫 번째 정책은 일단 획득한 채널로 스트리밍을 받는 것(FAST1)이고, 두 번째는 요청을 철회하고 σ 시간 후에 재시도하는 것(FAST2)이고, 마지막 정책은 획득한 채널로 스트리밍을 일단 받고 σ 시간 마다 채널의 추가 획득을 시도하는 것(FAST3)이다. FAST1에서, P_r 은 낮은 대역폭으로 스트리밍을 받기 때문에, 버퍼링 지연이 길 뿐만 아니라 P_r 이 준완숙 피어가 되는 데 걸리는 시간이 길어진다. 준완숙 피어가 아닌 미숙 피어들은 시스템의 대역폭을

1) P_r 은 각 미숙 피어 P_i 에 대해 $d_i(t)$ 를 결정하기 위해 필요한 정보를 얻을 수 있다고 가정한다.

```

var
  vout(i) : available out-bound bandwidth of peer  $p_i$ ;
  C : set of mature peers and semi-mature peers ;
begin
  if C =  $\phi$  then exit ;
   $P_1^s \leftarrow \max_{p_i \in C} v_{out}(i)$  ;
  if  $v_{out}(i) \leq R_m$  then
     $B_1 \leftarrow v_{out}(i)$  ;
  else  $B_1 \leftarrow R_m$  ;
   $B^* \leftarrow B_1$  ;  $n \leftarrow 1$  ;
  C  $\leftarrow$  C - {  $P_1^s$  } ;
  while (  $P_i \in C$  ) and (  $B^* < R_m$  ) do
     $n \leftarrow n + 1$  ;
     $P_n^s \leftarrow P_i$  ;
    if  $B^* + v_{out}(i) \leq R_m$  then
       $B_n \leftarrow v_{out}(i)$  ;
    else  $B_n \leftarrow R_m - B^*$  ;
       $B^* \leftarrow B^* + B_n$  ;
  done
end

```

(그림 3) 제공 피어 선택 알고리즘

점유하면서도 시스템 용량에 기여하지 못한다. 이러한 미숙 피어들이 FAST1의 경우 상대적으로 많이 존재하므로 FAST1은 좋은 선택이 되지 못한다. 반면, FAST2는 요청 피어가 스트리밍 서비스를 받기 위한 대기 시간이 FAST1보다 길지만, 버퍼링 지연은 더 짧고 준완속 피어가 되는 시간도 짧다. 또한, 대기하는 피어들은 대역폭을 점유하지 않는다. 따라서, FAST2가 FAST1보다 시스템 용량을 증가시키는 데 더 효과적이다. 그러나, FAST2는 작업 보존형(work-conserving)이 아니므로, 대기하는 피어가 있음에도 사용되지 않는 대역폭이 존재한다. FAST3은 FAST1과 FAST2의 단점을 극복하는 정책이다. 왜냐하면, FAST3에서 P_i 은 불충분한 대역폭으로 일단 스트리밍을 시작하고, 주기적으로 추가 대역폭을 얻기 위해 시도하기 때문이다. 즉 FAST3가 최선의 정책이라 할 수 있을 것이다.

FAST3가 최선이긴 하지만 복잡성으로 인해 해결해야 할 문제를 갖는다. 다른 정책과는 달리 수신 채널의 개수가 세션 중에 변경될 수 있다. 이것은 $d_r(t)$ 의 변경을 의미하므로 전송 스케줄의 변경과 준완속 피어 판단 조건의 변경을 초래하게 된다. 다행히도, 채널의 개수가 감소하지는 않기 때문에 채널 설정 변화에 따른 서비스 불능 현상은 발생하지 않는다. 하지만, 전송 스케줄은 새로운 채널 설정에 적용하여 세션 중에 수정되어야 한다. FSS는 슬롯 단위로 이루어지므로, 현재 슬롯이 끝난 뒤에 새 채널을 추가해 스케줄링이 가능하다. 단순히 이미 전송된 데이터를 제외한 나머지 부분에

대해 FSS를 적용하면 된다. 현재 시간을 t_0 라 하면, 스트리밍이 끝나는 시간 t_f 는

$$t_f = t_0 + \left\lceil \frac{L - d_r(t_0)}{\frac{B^*(r)\omega}{\gamma}} \right\rceil \omega$$

이다. 위 식에서 $L - d_r(t_0)$ 는 수신해야 할 남은 데이터량을 뜻하고, $B^*(r)\omega/\gamma$ 는 다중 채널로 한 슬롯 동안 보낼 수 있는 데이터량을 뜻한다. 따라서 올림(ceiling) 기호로 묶인 식은 남은 데이터를 전송하기 위해 요구되는 슬롯 개수를 의미한다. 만약 $d_r(t)$ 가 직선이라고 가정하면, $p(t) = t - \delta$ 이므로, t 와 $p(t)$ 에 각각 t_f 와 L 을 대입하면, 새로운 재생 지연 시간은 $\delta = t_f - L$ 이 된다. 하지만, 다중 채널에서 $d_r(t)$ 는 기온 계단형 증가함수이므로, 조정이 필요하다. 앞에서 언급했듯이, 버퍼링 지연은 식 (3)의 $d(t)$ 와 t 의 최대 수직 거리이다. 이 최대 거리는 $d_r(t)$ 의 마지막 점프 직전에 발생하고 점프 높이는 $(B^* - B_1)\omega/\gamma$ 이다. 따라서, $p(t) \leq d_r(t)$ 를 만족시키기 위해 마지막 점프 높이가 버퍼링 지연에 더해져야 한다. 즉, 다음과 같은 새로운 버퍼링 지연을 구할 수 있다.

$$\delta = (t_f - L) + \frac{B^*(r) - B_1(r)}{\gamma} \cdot \omega$$

5. 성능 평가

5.1 버퍼링 지연

먼저, FSS의 버퍼링 지연을 OTS의 버퍼링 지연과 비교했다. OTS는 채널 대역폭에 대한 제약이 있고 FSS는 슬롯-기반 프로토콜이므로, 이 두 기법을 직접 비교하는 것은 불가능하다. 따라서 공정한 비교를 위해, 슬롯 길이에 대한 기준을 정하고, OTS의 제약조건을 만족시키는 환경을 설정한 후 두 기법을 비교했다. OTS의 제약 조건은 다음과 같다.

- γ 가 미디어 콘텐츠의 재생율이라 할 때, 하나의 채널은 $\gamma/2, \gamma/4, \gamma/8, \dots, \gamma/2^N$ 중 하나의 대역폭을 갖는다.
- 한 세션의 총 대역폭 $B^*(r) = \gamma$ 이다.

세그먼트의 개수를 m 이고 세그먼트가 다운로드되면서 재생될 수 있다고 가정할 때, n 개의 제공 피어와 한 개의 요청 피어에 대해 OTS의 버퍼링 지연은:

$$\delta_{OTS} = \frac{(n-1)L}{m}$$

이다[10]. 같은 조건 하에서, FSS의 버퍼링 지연은 식 (4)에 따라

$$\delta_{FSS} = \frac{(1 - B_1(r))}{\gamma} \cdot \omega$$

이다. 여기서 $B_1(r)$ 은 요청 피어의 다중 채널 중 대역폭이 가장 큰 채널의 대역폭이다. 위 두 식에서 보듯이, δ_{OTS} 과 δ_{FSS} 은 아무런 연결고리가 없으므로 사실 명확한 비교가 어렵다, 하지만, 굳이 비교를 위해 어떻게 든 두 식 간의 연관성을 이끌어내야 한다. 이를 위해 가장 큰 세그먼트를 전송하는 데 걸리는 시간을 기준 인자로 설정하였다. OTS의 경우에는 세그먼트의 길이가 모두 동일하므로 가장 큰 채널 대역폭 B_1 으로 하나의 세그먼트를 전송하는 데 걸리는 시간이고, FSS의 경우는 단순히 슬롯 길이 w 이다. 예를 들어, (그림 1)(a)에서 세그먼트 s_1, s_2, s_4, s_8 을 전송하는 채널이 B_1 이고, OTS에서 세그먼트의 길이는 모두 같다. 따라서 기준 인자는 S_1 을 전송하는 데 걸리는 시간(즉, 1이라고 쓰여진 사각형의 가로 길이)이고, 이는 (그림 1)(b)의 w 와 같다. OTS의 세그먼트의 길이는 L/m 이고, 크기는 $\gamma L/m$ 이므로, 아래와 같은 식을 만들 수 있다.

$$\frac{\gamma}{B_1(r)} \cdot \frac{L}{m} = \omega$$

이것을 m 에 대해 정리한 후 δ_{OTS} 에 대입하면:

$$\delta_{OTS} = (n-1) \frac{B_1(r)}{\gamma} \cdot \omega$$

가 된다. 여기서 $c = B_1(r)/\gamma$ 라 하자. 그러면, $B_1(r)$ 이 그 세션의 가장 큰 대역폭이고 $B^*(r) = \gamma$ 이므로 n 은 $1/c$ 보다 크거나 같다. 즉, $n \geq 1/c$. 예를 들어 $B_1(r)$ 이 $\gamma/4$ 라 하면 그 세션의 제공 피어의 개수 n 은 4보다 크거나 같아야 한다는 것이다. 따라서

$$\delta_{OTS} = (n-1)c \cdot \omega \geq \left(\frac{1}{c} - 1\right)c \cdot \omega = \delta_{FSS}$$

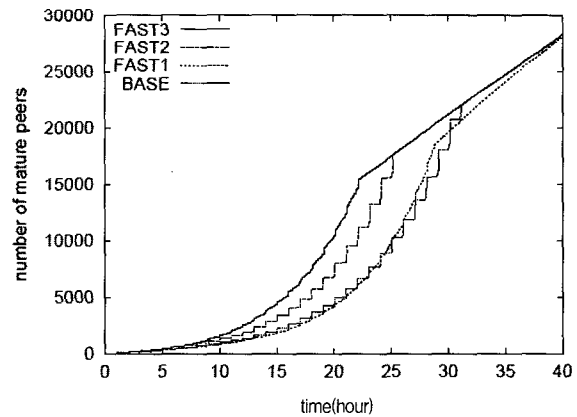
이다. 즉, FSS의 버퍼링 지연은 OTS보다 작거나 같다. (그림 1)의 예에서 $\delta_{OTS} = 3\omega/2$ 이고, $\delta_{FSS} = \omega/2$ 로 FSS의 버퍼링 지연이 OTS의 1/3에 불과하다는 것을 알 수 있다.

5.2 배포 가속

이 절에서 우리는 시뮬레이션을 통해 FAST 기법의 배포 가속 효과를 보여준다. 전체 50,100개의 피어를 가지고 있는 P2P 미디어 스트리밍 시스템을 실험했다. 최초로 콘텐츠를 가지고 있는 100개의 종자 피어가 있고, 다른 50,000개의 피어들은 시나리오에 따라 콘텐츠를 요청하게 된다. 모든 피어

의 수신 대역폭은 재생율과 같은 γ 이고, 종자 피어의 송신 대역폭은 모두 $\gamma/2$ 이고, 나머지 피어들은 $\gamma/2, \gamma/4, \gamma/8, \gamma/16$ 네 개 중 하나를 갖도록 했다. 각 대역폭의 분포는 10%, 10%, 40%, 40%이다. 채널 획득을 실패하거나 충분한 대역폭이 없을 때 재시도하는 주기 σ 와 슬롯 길이 ω 는 10분으로 정했고, 비디오의 길이는 60분이다. 또한 요청 피어의 요청 발생은 포아송(Poisson) 분포를 따른다고 가정했다. 비교대상이 되는 BASE 기법은 준완속 피어들을 제공 피어로 고려하지 않고, FAST2처럼 세션의 총 수신 대역폭이 요청 피어의 최대 수신 대역폭과 같은 경우($B^*(r) = R_m(r)$)에만 스트리밍을 시작하는 기법이다.

P2P 시스템에서는 배포가 진행됨에 따라 시스템의 용량이 증가한다는 특성을 갖는다. 따라서 요청이 평균 $1/\theta$ 의 포아송 분포에 따라 발생한다면, 처음에 완속 피어가 적은 상태에서는 그 요청을 완전히 수용할 수가 없다. 초기에는 이러한 혼잡 상황이 지속되다가 배포가 진행됨에 따라 완속 피어수의 증가로 어느 순간에는 모든 요청을 모두 수용할 수 있는 상태가 된다. 우리는 $1/\theta$ 의 요청 발생률을 모두 수용할 수 있는 용량에 도달하는 데 걸리는 시간을 θ -용량 도달 시간이라고 정의한다. 이 것이 작을수록 배포 속도가 빠르다고 할 수 있다.



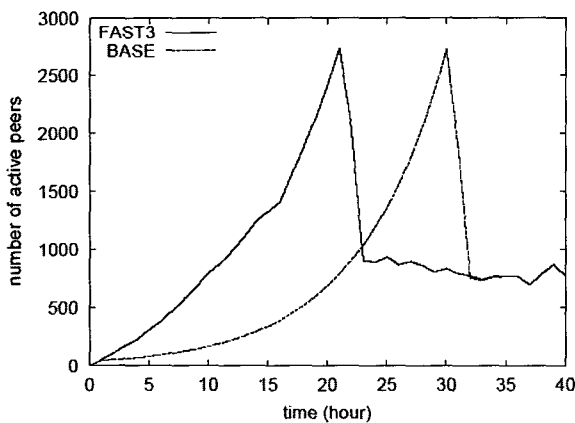
(그림 4) 배포 속도 ($\theta = 5$ 초)

(그림 4)는 θ 가 5초일 때, BASE, FAST1, FAST2, FAST3의 시간에 따른 완속 피어 개수의 증가를 보여준다. 5초의 요청 발생률을 수용할 수 있게 되면, 그림에서 mature peer의 개수는 직선으로 증가하게 된다. 하지만, 초반에 오목하게 증가하다가 약 20시간이 지난 이후에 직선으로 증가하는 것을 볼 수 있다. 이 오목한 부분이 혼잡 상태를 의미한다. FAST3의 5초-용량 도달 시간은 약 21시간으로, FAST2의 24시간, FAST1의 28시간, BASE의 30시간 보다 짧은 것을 알 수 있다. 또한 FAST3은 혼잡 상황에서 BASE에 비

해 완속 피어의 개수가 거의 두 배이다. 정확한 성능 향상은 <표 1>에서 제시되어 있다. FAST1이 30~40시간 근처에서 길게 늘어지는 것은 FAST1의 특성상 혼잡 상황에서의 낮은 수신 대역폭의 세션에서 서비스 중인 미속피어들이 잔존해 있기 때문이다. 즉, 높은 수신 대역폭의 새로운 요청 피어들과 낮은 수신 대역폭의 오래된 미속 피어들이 비슷한 시간에 완속 피어가 됨으로써 마치 요청 발생률이 높은 것처럼 직선의 기울기가 가파르게 보이는 것이다. FAST2와 BASE가 계단식으로 증가하는 것은 이들이 $B^*(r) = R_{in}(r) = \gamma$ 일 때만 스트리밍을 시작하기 때문인데, 스트리밍의 종료와 시작이 동시에 다발적으로 일어나기 때문이다. 다시 말하면, 최초에 송신 대역폭이 $\gamma/2$ 인 100개의 종자 피어가 있으므로 동시에 50개의 요청 피어가 스트리밍을 시작하고, 1시간 후에 동시에 끝나게 된다. 그러면 또 새로운 요청 피어들이 동시에 시작하고, 이런 과정이 반복되면서 계단형의 증가 곡선을 만들게 된다.

<표 1> BASE와 FAST3의 성능 비교

시간	완속피어의 수			활성피어의 수		
	BASE	FAST3	성능향상	BASE	FAST3	성능향상
5	245	351	143%	86	309	359%
10	822	1503	183%	167	797	477%
15	1963	4325	220%	340	1329	391%
20	4267	10378	243%	688	2413	351%

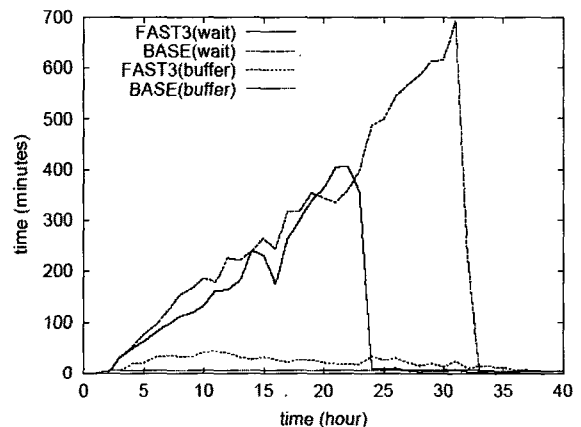


(그림 5) 활성 피어의 개수

요청 후 스트리밍을 시작하는데 걸리는 시간을 대기 시간이라고 정의하고, 스트리밍 시작 후 재생하는데 걸리는 시간을 버퍼링 지연이라고 정의한다. 혼잡 상황에서는 요청 피어들이 채널을 확보하기 어렵기 때문에 대기 시간이 0보다 크고, 혼잡 상황이 아니라면 대기 시간이 0이 된다. 버퍼링 지연은 요청 피어의 수신 채널 대역폭과 식 (4)에 의해 결정된다. 따라서 요청 피어가 재생을 시작할 때까지 걸리는 시간은 대기

시간과 버퍼링 지연을 합한 값이다. (그림 5)는 FAST3과 BASE 기법 하에서 서비스 받고 있는 활성 피어들의 개수의 변화를 보여준다. 혼잡 상황에서, 증가 곡선은 지수적으로 증가하고 시스템이 5초-용량에 도달한 후에는 평균적으로 710개(60분×60초/5초)의 활성 피어를 유지한다. FAST3은 BASE에 비해 혼잡상황(처음 20시간)에서 약 3배에서 약 5배 정도 많은 활성 피어를 갖는다. 정확한 성능 향상은 <표 1>에서 제시되어 있다.

(그림 6)은 θ 가 5초일 때 FAST3과 BASE의 대기 시간과 버퍼링 지연의 시간에 따른 변화를 보여준다. 혼잡 상황에서 대기시간이 버퍼링 지연보다 훨씬 큰 것을 볼 수 있다. 따라서, 혼잡상황에서는 대기시간을 줄이는 것이 중요하다. 24시부터 FAST3은 5초-용량에 도달하여 대기시간이 0이 된다. 따라서 24시 후 약 9시간 동안 BASE는 FAST3 비해 상당한 긴 대기시간을 가진다.



(그림 6) 대기 시간과 버퍼링 지연 ($\theta = 5$ 초)

6. 결 론

P2P 모델은 클라이언트-서버 모델의 서버 링크 병목 현상을 해결하는 이상적인 모델이다. 본 논문에서, P2P 네트워크에서 연속형 미디어 콘텐츠의 배포에 따르는 두 가지 문제를 다루었다. 첫째는 다중-소스 스트리밍 세션에서 미디어 데이터의 전송 스케줄링 문제이다. 이 문제에 대한 해법으로 버퍼링 지연을 최소화시키는 FSS라는 정밀한 스케줄링 기법을 제안했다. 이 기법은 데이터 세그먼트가 전송되는 채널의 대역폭에 따라 세그먼트의 길이를 변화시킴으로써 버퍼링 지연을 최소화할 수 있었다. 둘째는 스스로 용량이 증가하는 P2P 시스템에서 미디어 콘텐츠의 배포 속도 문제이다. 이 문제에 대한 해법으로 시스템 용량의 증가 속도를 가속화시키는 FAST라는 기법을 제안했다. 이 기법은 콘텐츠를 가지고

있는 피어 뿐 아니라 콘텐츠를 수신 중인 피어를 제공 피어로 이용함으로써 전체 시스템의 용량을 확장시켜 배포 속도를 가속시켰다. 마지막으로, 분석적 모델과 시뮬레이션을 통해 FSS와 FAST의 성능을 측정했다.

참고 문헌

- [1] Gnutella. <http://gnutella.wego.com>.
- [2] Napster. <http://www.napster.com>.
- [3] I. Clake, O. Sandberg, B. Wiley and T. Hong. Freenet : A Distributed Anonymous Information Storage and Retrieval System, In Proc. of Workshop on Design Issues in Anonymous and Unobservability, July, 2000.
- [4] Hrishikesh Deshpande, Mayank Bawa and Hector Garcia-Molina. Streaming Live Media over a Peer-to-Peer Network, Technical report, Stanford University, August, 2001.
- [5] T. Nguyen and A. Zakhor. Distributed Video Streaming Over Internet, In Proc. of Multimedia Computing and Systems, San Jose, California, January, 2002.
- [6] Venkata N. Padmanabhan, Helen J. Wang, Philip A. Chou and Kunwadee Sripanidkulchai, Distributing Streaming Media Content Using Cooperative Network-ing. In Int. Workshop on Network and Operating Systems Support for Digital Audio and Video, Miami Beach, FL, May, 2002.
- [7] S. Ratnasamy, P. Francis, M. Handley, R. Karp and S. Shenker. A Scalable Content-Addressable Network, In ACM SIGCOMM, August, 2001.
- [8] A Rowstron and P. Druschel. Pastry : Scalable Distributed Object Location and Routing for Large-Scale Peer-to-Peer Systems, In Proc. of IFIP/ACM Middleware, November, 2001.
- [9] I. Stoica, R. Morris, D. Karger, F. Kaashoek and H. Balakrishnan. Chord : A Scalable Peer-to-Peer Lookup Service for Internet Applications, In Proc. of ACM SIGCOMM, August, 2001.
- [10] Dongyan Xu, Mohamed Hefeeda, Susanne Hambrusch and Bharat Bhargava, On Peer-to-Peer Media Streaming. In Proc. of Int. Conf. on Distributed Computing Systems 2002, Austria, July, 2002.
- [11] B. Y. Zhao, J. Kubiatowicz and A. D. Joseph. Tapestry : An Infrastructure for Fault-Resilient Wide-Area Location and Routing, Technical report, University of California at Berkeley, April, 2001.
- [12] C. Diot, B. N. Levine, B. Lyles, H. Kassem and D. Balensiefen, Deployment issues for the IP multicast service and architecture, IEEE Network magazine special issue on Multicasting, pp.78-88, January/February, 2000.
- [13] Yang-Hua Chu, Sanjay G. Rao and Hui Zhang, A case for end system multicast, in ACM SIGMETRICS, pp.1-12, 2000.
- [14] L. Zhang, S. Deering, D. Estrin, S. Shenker and D. Zappala, "RSVP : A New Resource ReSerVation Protocol," IEEE Network, September, 1993.
- [15] D. Clark, S. Shenker and L. Zhang, "Supporting Real-Time Applications in an Integrated Services Packet Network : Architecture and Mechanism," In Proceedings of ACM SIGCOMM, pp.14-26, 1992.
- [16] K. Nahrstedt and R. Steinmetz. Resource Management in Networked Multimedia Systems. IEEE Computer, May, 1995.
- [17] D. Ferrari, A. Banerjea and H. Zhang, "Network Support for Multimedia: A Discussion of the Tenet Approach," Computer Networks and ISDN Systems 26, special issue on Multimedia Networking, 1994.

부록 (식 (4)의 증명)

정의에 의해 $0 \leq p(t) \leq L$ 와 $0 \leq d(t) \leq L$ 이 성립하므로, $d(t) = L$ 이면 식 (1)은 δ 에 상관없이 항상 성립한다. 따라서, $d(t) = L$ 인 경우를 제외하면

$$d(t) = \frac{B_1}{\gamma} t + \frac{B^* - B_1}{\gamma} \left\lfloor \frac{t}{\omega} \right\rfloor \omega$$

이고, 식 (1)에 위 $d(t)$ 를 대입하면 아래와 같이 쓸 수 있다.

$$\frac{B_1}{\gamma} t + \frac{B^* - B_1}{\gamma} \left\lfloor \frac{t}{\omega} \right\rfloor \omega \geq p(t).$$

$d(t) < L$ 이므로, 위 부등식이 만족되기 위해서는 $p(t) < L$ 이다. 즉

$$p(t) = \max\{t - \delta, 0\}$$

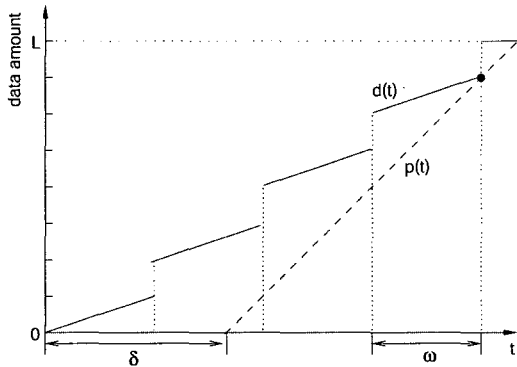
이어야 한다. 또한, $p(t) = 0$ 이면 위 부등식은 항상 성립하므로, 위 부등식이 만족하기 위해서는 $t - \delta > 0$ 인 t 와 δ 에 대해서 아래 부등식이 만족되어야 한다.

$$\frac{B_1}{\gamma} t + B^* - B_1 \frac{t}{\gamma} \left\lfloor \frac{t}{\omega} \right\rfloor \omega \geq t - \delta.$$

좌변은 (그림 1)(b)의 원점에서 시작되는 기울어진 계단형 함수이고, 우변은 기울기가 1이고 x-절편이 δ 인 함수이다. 단, (그림 1)(b)는 $B^* = \gamma$ 인 특수한 경우의 예이므로 전체

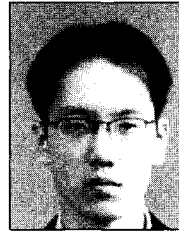
적인 기울기가 1이지만, 일반적인 가정은 $B^* \leq \gamma$ 이므로 아래 그림처럼 계단형 함수의 전체적인 기울기는 1보다 작을 수 있다. 아래 그림을 보면 좌변 함수가 우변 함수보다 항상 크거나 같기 위해서는 우변 함수가 계단형 함수의 마지막 점프 직전의 값을 지나거나 아래에 있어야 한다. 즉, 버퍼링 지연 δ 를 최소로 하기 위해서는 우변함수 $t - \delta$ 가 우변함수의 마지막 점프 직전 좌표(그림에서 검은 점)를 지나야 한다. 그림에서 그 위치 좌표를 $(t, d(t))$ 라 할 때, $t = \gamma L / B^*$ 이고, $d(t)$ 은 L 에서 점프 높이를 뺀 값이 된다. 즉,

$$d(t) = L - \left(\frac{B^* - B_1}{\gamma} \right) \cdot \omega$$



이다. 이 좌표를 $p(t) = t - \delta$ 에 대입하면, 최소 버퍼링 지연 δ 를 아래와 같이 구할 수 있다.

$$\delta = \left(\frac{\gamma}{B^*} - 1 \right) L + \left(\frac{B^* - B_1}{\gamma} \right) \omega, \text{ if } B^* \leq \gamma$$



권진백

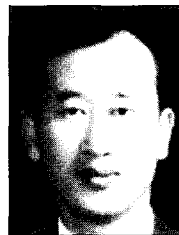
e-mail : jbkwon@sunmoon.ac.kr
 1998년 한국외국어대학교 정보통계학 이
 학사
 2000년 서울대학교 전산과학 이학석사
 2003년 서울대학교 컴퓨터공학 공학박사
 2003년~현재 선문대학교 컴퓨터정보학부
 전임강사

관심분야: 멀티미디어시스템, 임베디드시스템, 분산시스템



염헌영

e-mail : yeom@snu.ac.kr
 1984년 서울대학교 계산통계학 이학사
 1986년 Texas A&M 전산학 이학석사
 1992년 Texas A&M 전산학 이학박사
 1993년~현재 서울대학교 컴퓨터공학부 교수
 관심분야: 분산시스템, 결합내성시스템, 멀티
 미디어시스템



이정배

e-mail : jblee@sunmoon.ac.kr
 1981년 경북대학교 전자공학과 전산전공
 공학사
 1983년 경북대학교 대학원 전산 전공
 공학석사
 1995년 한양대학교 대학원 전자공학과
 공학박사

1982년~1991년 한국전자통신연구원 선임연구원
 1991년~2002년 부산외국어대학교 컴퓨터공학과 부교수
 2002년~현재 선문대학교 컴퓨터정보학부 부교수
 관심분야: 임베디드 시스템, 실시간 운영체제, 임베디드 프로토타
 이핑, 이동멀티미디어방송, 디지털방송, 데이터방송, 등