

P2P 스트리밍 시스템을 위한 시뮬레이터 개발

김혜선*, 조윤식*, 황기태*

*한성대학교 컴퓨터공학과

e-mail:{happyi00, holysmoke, calafk}@hansung.ac.kr

Design and Implement of a Simulator for P2P Streaming Systems

Hye-Sun Kim*, Yoon-Sik Cho*, Kitae Hwang*

*Dept. of Computer Engineering, Hansung University

요 약

인터넷 스트리밍 시스템은 스트리밍 소스를 공급하는 미디어 서버와 이로부터 미디어 스트림을 받아 분배하는 스트리밍 서버, 그리고 스트리밍 단말기들로 구성되며, 기존에는 하나의 스트리밍 서버에 다수의 단말기들이 직접적으로 연결되는 방식을 취하고 있다. 이러한 클라이언트-서버 형태의 중앙 구조의 단점을 극복하여 확장성을 제공하기 위해 P2P 스트리밍 시스템에 대한 연구가 최근 들어 진행되고 있다. 그러나 P2P 방식을 이용한 인터넷 스트리밍 시스템을 설계, 구현, 테스트를 하기 위해서 많은 비용이 소비되는 현실적인 어려움이 있다. 그러므로 본 논문에서는 P2P 스트리밍 시스템에 대한 다양한 연구 및 실험을 지원하기 위해, P2P 스트리밍 시스템을 정의하고, NS2 시뮬레이션 라이브러리를 이용하여 P2P 스트리밍 시스템 시뮬레이터 *P2PStreamSim*을 설계 및 구현하였다. 또한 테스트 P2P 스트리밍 시스템을 사례로 적용하여 *P2PStreamSim*의 동작을 검증하고 성능을 평가하였다.

1. 서론

네트워크 속도 및 멀티미디어에 대한 컴퓨팅 처리 능력이 향상됨에 따라 인터넷 방송 등 인터넷 스트리밍 시스템들이 개발 및 활용되고 있다. 인터넷 스트리밍 시스템에 P2P 네트워크를 이용함으로써 클라이언트-서버 형태 구조의 단점을 해결하기 위한 연구들이 다소 진행되고 있다.[1,2] P2P 스트리밍 시스템이란 스트리밍 소스가 존재하며 이를 소비하고자 피어들이 직접적으로 스트리밍 소스에 접속하여 스트리밍 데이터를 받아 재생하거나, 스트리밍이 진행 중인 다른 피어에 접속하여 스트림 데이터를 중계받아 재생하는 방식의 시스템을 말한다.

P2P 스트리밍 시스템의 실제 구현을 통한 실험은 현실적인 어려움이 있다. 그러므로 P2P 스트리밍 시스템에 대한 연구, 개발을 위해 필연적으로 시뮬레이터의 개발이 필요하다.

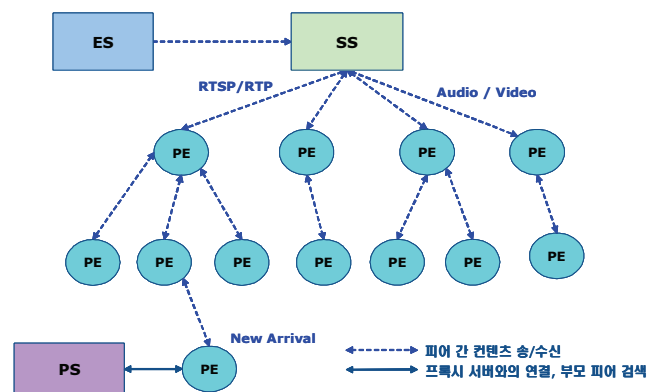
이에 본 논문에서는 P2P 스트리밍 시스템에 대한 시뮬레이터인 *P2PStreamSim*을 NS2(Network Simulator 2)[3]를 근간으로 설계 구현하였다. 또한 시뮬레이터의 동작 및 시뮬레이션의 정확성을 확인하기 위해 테스트 P2P 스트리밍 시스템을 정의하고 이를 시뮬레이션하고 성능을 분석하였다.

2. P2P 스트리밍 시스템 모델링

2.1 P2P 스트리밍 시스템 정의

본 절에서는 P2P 시스템을 정의한다. P2P 스트리밍 시스템은 구조적 측면에서 인코딩 서버(Encoding Server:

ES), 스트리밍 서버(Streaming Server: SS), 프록시 서버(Proxy Server : PS), 그리고 피어(Peer: PE)들로 구성된다. ES는 비디오 또는 오디오 데이터를 인코딩하여 SS로 RTP(Real-Time Transport Protocol)[4]를 이용하여 전송한다. 부모 피어와 자식 피어는 RTSP(Real Time Streaming Protocol)[5]로 연결되며 실시간 패킷 전송은 RTP를 이용한다. SS는 최상위 부모 피어의 역할로서, 자식 PE 들에게 수신한 패킷을 실시간으로 전달한다. PS는 피어의 도착과 이탈 시 필요에 따라 피어들에게 부모 피어를 결정하여 알려주며, 이에 따라 네트워크 토폴로지를 재구성한다. 그림 2.1은 본 논문에서 정의한 P2P 스트리밍 시스템의 한 사례이다.



(그림 2.1) P2P 스트리밍 시스템 모델 사례

2.2 행동 모델

2.2.1 실시간 릴레이

ES로부터 말단 PE까지 스트리밍 서비스가 이루어지는 과정이다. ES는 일정한 시간간격으로 RTP 패킷을 생성하여 SS로 전달하며, SS는 이를 연결된 자식 PE들에게 분배한다. 각 PE들은 SS로부터 수신한 RTP 데이터를 버퍼링한 후, 한 프레임씩 디코딩하여 재생한다. 또한 PE는 연결된 자식 PE들에게 각각 RTP 데이터를 릴레이 해주는 역할을 한다.

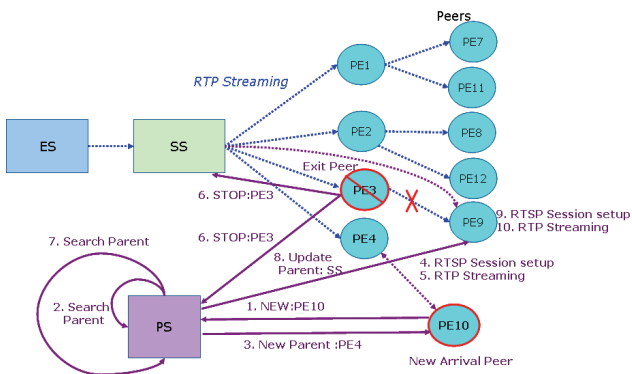
2.2.2 피어 참여

새로 도착한 피어는 PS에게 자신의 존재를 알린다. PS는 적당한 부모 피어를 탐색하여 새로 도착한 피어에게 알려준다. 새로 도착한 피어는 부모 피어로 접속하여 RTSP 세션을 설정한 후 RTP 패킷을 수신한다. 새로 도착한 피어는 수신한 패킷을 버퍼에 일시 저장하고 즉시 디코딩 후 재생한다.

2.2.3 피어 이탈 및 재배포

피어가 네트워크에서 이탈할 시, 연결된 자식 피어들이 새로운 부모 피어로 접속되는 과정이다. PE가 이탈 시, PS와 부모 피어에게 알린다. PS는 적당한 부모 피어(SS)를 검색하여 이탈 노드의 자식 피어에게 새로운 부모 피어(SS)를 알려준다. 이탈 노드의 자식 피어는 새로운 부모 피어에 접속하여 RTSP 세션을 설정한다. 이탈 노드의 자식 피어는 새로운 부모 피어로부터 RTP 패킷을 수신하여 디코딩 후 재생한다.

그림 2.2는 실시간 릴레이, 피어 참여, 피어 이탈 및 재배포 과정을 묘사한 그림이다.



(그림 2.2) P2P 스트리밍 시스템의 행동 모델

2.3 성능 모델

• 연결 지연 시간(P_{delay})

P_{delay} 는 사용자가 스트리밍 서비스를 받기 위해 기다리는 초기 연결 지연 시간을 의미한다. 이는 P2P 네트워크의 과부하 정도에 대한 지수로 이용할 수 있다.

• 재생 시간 갭($P_{playgap}$)

$P_{playgap}$ 은 동일한 프레임이 ES에서 재생 될 때와 임의의 PE에서 재생되는 시간 사이의 차이 값을 나타낸다. 동일한 조건하에서 버퍼링 알고리즘이나 PS의 배치 알고리즘의 효율성에 따라 달라질 수 있는 성능 요소이다.

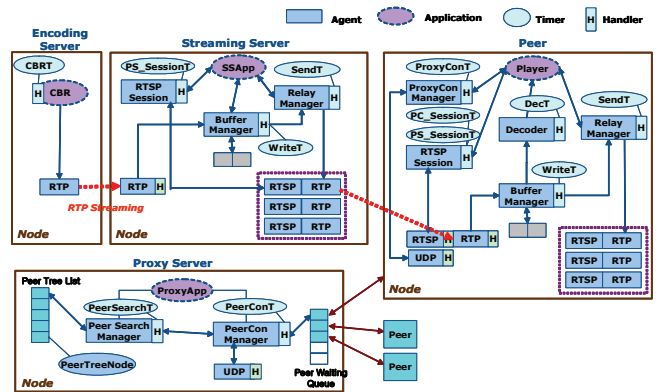
• 지터율($P_{jitterrate}$)

$P_{jitterrate}$ 은 스트리밍 시간동안 총 패킷 중 손실 패킷에 의한 지터의 비율이다. 패킷 손실은 P2P 시스템에서 PE들의 자유로운 이탈로 인하여, 스트리밍의 일시적 중단, 혹은 네트워크 트래픽의 급격한 변화로 인해 발생한다.

3. P2PStreamSim 구현

3.1 시뮬레이터 구조

NS2에서는 P2P와 같은 노드들이 동적으로 생성, 삭제가 자유롭게 일어나는 네트워크 토폴로지를 시뮬레이션 하기에 다소 부적합한 구조로 되어있다. 따라서 동적인 피어의 도착과 이탈을 시뮬레이션 하기 위해 노드 간 링크는 메시 구조로 구성하고, 동적인 연결을 위해 RTSP와 RTP 에이전트를 런타임 중에 생성하여 연결하도록 구현하였다. AgentManager는 동적으로 RTSP와 RTP를 생성하여 노드에 attach 하거나 detach 하기 위한 OTcl 클래스로, 동적인 피어의 참여와 이탈을 시뮬레이션 하기 위해 필요하다. 그림 3.1은 P2PStreamSim의 전체 내부 구조이다.



(그림 3.1) P2PStreamSim 내부 설계

3.2 ES(Encoding Server) 구현

ES는 CBR 또는 VBR 형태의 트래픽을 발생시키는 어플리케이션과 RTP 패킷을 전송하는 RTP 에이전트로 구성된다. CBR 어플리케이션은 일정한 간격으로 RTP 데이터를 생성한다. RTP 에이전트는 생성된 RTP 데이터를 패킷으로 나누어 전송한다.

3.3 SS(Streaming Server) 구현

SS는 스트림을 분배하는 노드로서 부모 피어의 역할을 수행한다. SS는 RTSP Session, Buffer Manager, Replay Manager, RTSP, RTP 에이전트로 구성된다.

임의의 PE가 RTSP 세션 연결을 요청할 경우 RTSP

Session 에이전트는 응답 메시지를 생성하여 RTSP 에이전트로 보낸다. 세션 설정 후, Buffer Manager는 RTP 에이전트로부터 수신한 RTP 패킷을 버퍼에 저장한다. Relay Manager 에이전트는 버퍼에서 한 패킷씩 읽어 RTP 에이전트를 통해 자식 PE 들에게 전송한다.

3.4 PE(Peers) 구현

PE의 구조는 SS와 유사하며 SS 구조에 Proxy Con Manager와 Decoder 에이전트가 추가된 구조이다.

임의의 PE가 네트워크에 참여할 때, Proxy Con Manager는 요청 메시지를 생성하여 UDP를 통해 PS에게 새로운 부모 피어를 요청한다. PE는 PS로부터 할당받은 부모 피어로 RTSP Session 에이전트를 이용하여 요청 메시지를 생성하고 RTSP 에이전트를 통해 부모 피어로 전송한다. 부모 피어와 세션을 설정한 후, Buffer Manager는 RTP 에이전트로부터 수신한 데이터를 버퍼에 저장한다. Decoder 에이전트는 버퍼에서 한 프레임씩 읽어 디코딩한 후 재생한다. 연결된 자식 피어가 있을 경우 Relay Manager 에이전트는 버퍼에서 한 패킷씩 읽어 RTP를 통해 연결된 피어에게 데이터를 전송한다.

3.5 PS(Proxy Server) 구현

PS는 P2P 네트워크에 존재하는 피어에 대한 정보를 관리하는 서버로서, PE가 네트워크에 참여하거나 이탈 시 부모 피어를 탐색하여 PE에게 알려주며 동적으로 네트워크 토폴로지를 재구성한다.

4. 테스트 시뮬레이션

테스트 P2P 스트리밍 시스템을 정의하고 이를 사례로 적용하여 *P2PStreamSim*의 동작을 검증하고 성능 평가를 수행한다.

4.1 테스트 P2P 스트리밍 시스템 모델 및 성능 지수

테스트 P2P 스트리밍 시스템은 한 대의 ES, SS, PS와 n (현재는 100)개의 PE로 구성되며 성능 지수는 연결 지연 시간, 재생 시간 잭, 지터율로 설정한다.

4.2 시뮬레이션 실행

시뮬레이션을 위해 표 4.1, 4.2와 같이 파라미터를 설정하였다. PE ID는 피어를 구별하는 식별자이고, P2P 네트워크에 참여하는 순서대로 ID가 하나씩 증가한다. 100개의 PE가 평균 1초 간격으로 지수 분포로 도착한다. ES의 bitRate은 Atome 사의 H.264 / AAC Kompressor 하드웨어 인코더[6]를 기준으로 해상도가 720×480 일 때 실제 측정된 값으로 설정하였다.

<표 4.1> P2P 스트리밍 시스템 모델 파라미터

요소	파라미터	값
PE	bufferCount	2
	bufferAccessDelay	1us
	RTSPStackTime	1ms
	decodingTime	1ms
PS	protocolStackTime	1ms
	algorithmTime	1ms

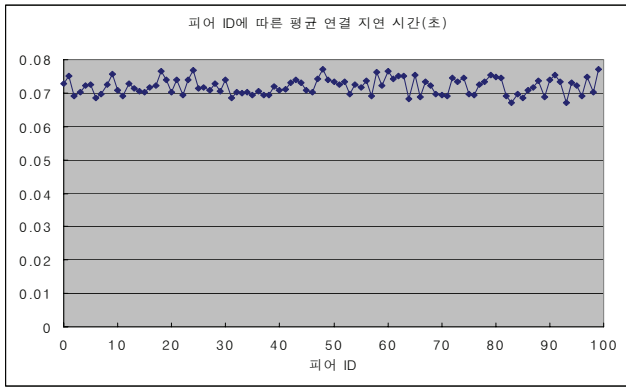
<표 4.2> 작업 부하

요소	파라미터	값
네트워크 환경	bandwidth	100Mbps
	transmissionDelay	10ms
ES	streamingTime	500s
	encodingType	CBR
	dataType	Video
	bitRate	71Kbps
	frameRate	30fps
PE	arrivalPeerCount	100
	meanInterarrivalTime	1.0s
	interarrivalDistribution	Exponential
	exitPeerPercentage	30%
	meanExitIntervalTime	1.0s
	exitIntervalDistribution	Exponential
	exitStartTime	200s
	capability	5
	waitExit	0ms, 30ms, 70ms

4.3 시뮬레이션 결과

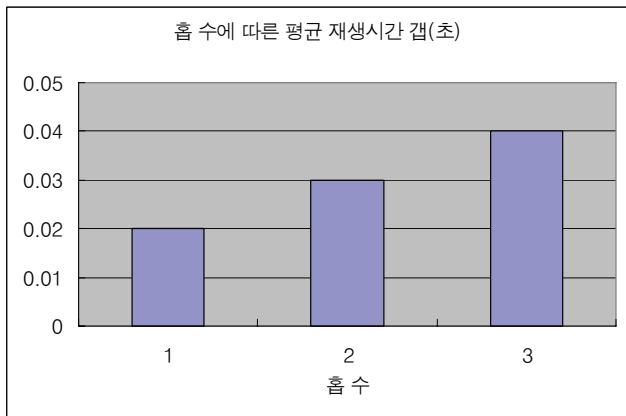
4.3.1 실험 결과 1 : 피어 참여에 따른 실험 결과

PE ID 별로 평균 연결 지연 시간(P_{delay})을 측정된 결과는 그림 4.1과 같다. 실험 결과 총 100개의 피어에 대한 평균 연결 지연시간은 약 70ms 소요됨을 알 수 있다. P_{delay} 는 PE가 PS 와 자신의 피어 서버와 통신하는 시간(약 40ms)과 각 PE들이 첫 번째 프레임을 받기 위해서 소요되는 시간(26.5ms)의 합이다.



(그림 4.1) 피어 ID에 따른 평균 연결 지연 시간(P_{delay})

그림 4.2는 홉 수에 따른 평균 재생 시간 갭($P_{playgap}$)을 측정한 결과이며, 테스트 P2P 시스템에서 PE들의 최대 홉 수는 3이다.



(그림 4.2) 홉 수에 따른 평균 재생 시간 갭($P_{playgap}$)

실험 결과 홉 수가 증가함에 따라 피어의 재생 시간 갭이 증가함을 알 수 있다. 네트워크에 참여한 피어의 가용 능력, 즉 연결 가능한 자식 피어의 수가 모두 동일하며 피어의 프로세싱 시간(버퍼링, 릴레이)이 매우 짧기 때문에 재생 시간 갭에 영향을 미치지 않는다.

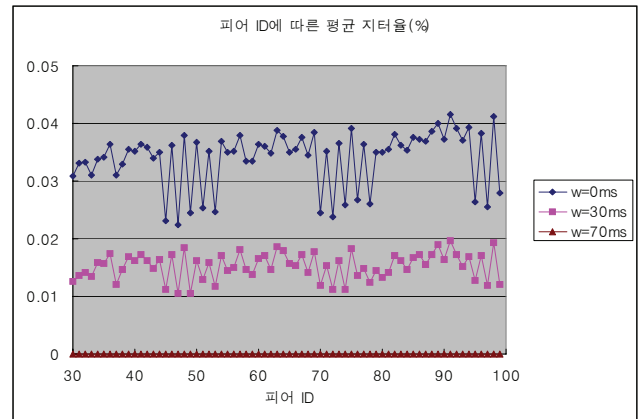
4.3.2 실험 결과 2 : 피어 이탈에 따른 실험 결과

지터의 발생은 부모 피어의 이탈로 인해 자식 피어들의 스트리밍 서비스가 일시적으로 중단되기 때문이다.

임의의 피어가 스트리밍 서비스를 중단하고 이탈할 경우 이탈하고자 한 시점에서 실제 사용자가 완전한 종료에 이루어질 때까지 기다릴 수 있다고 정의한 최대 대기 시간을 $T_{waitexit}$ 로 정의하며, 이는 시스템에 설정하는 외부 상수이다.

총 500초의 스트리밍 시간동안 100개의 피어 중 30%의 피어가 평균 1초 간격으로 이탈할 경우를 시뮬레이션 하였다. 그림 4.3은 $T_{waitexit}$ 시간이 0ms부터 70ms까지 증가할 경우 PE ID에 대한 평균 지터율($P_{jitterrate}$)을 측정한 결

과이다.



(그림 4.3) 피어 ID에 따른 평균 지터율($P_{jitterrate}$)

실험 결과 $T_{waitexit}$ 가 증가함에 따라 PE ID에 따른 평균 지터율은 감소하며, $T_{waitexit}$ 가 70ms일 때는 지터가 발생하지 않는다. 이것은 피어의 평균 연결 지연 시간이 약 70ms이므로 이 시간만큼 이탈 피어가 $T_{waitexit}$ 을 유지한다면 자식 피어들은 스트림 끊김 현상 없이 원활한 서비스를 받을 수 있다.

5. 결론

본 논문에서는 P2P 스트리밍 시스템에 대한 모델과 성능 지수를 정의하고 NS2 시뮬레이션 툴을 이용하여 **P2PStreamSim**을 설계 및 구현하였다. 또한 테스트 P2P 스트리밍 시스템에 대한 성능 평가를 수행하였다.

본 논문 연구는 P2P 스트리밍 시스템을 개발하기 전 단계에서 설계의 정확성을 검증하고 그 성능을 평가할 수 있는 도구를 제공한다는 면에서 큰 의미를 지닌다. 향후 P2P 스트리밍 시스템에 대한 실제 파라미터 값들을 적용한 실험을 수행하며 다양한 P2P 스트리밍 시스템 모델에 대한 성능 평가를 수행할 수 있도록 **P2PStreamSim**을 확장 보완하고자 한다.

참고문헌

- [1] Napster, <http://openmap.sourceforge.net/napster.txt>
- [2] M.Ripeanu, "Peer-to-Peer Architecture Case Study: Gnutella Network," *In Proceedings of IEEE 1st Int'l Conf. on Peer-to-Peer Computing*, 2001
- [3] NS2tutorial, <http://www.isi.edu/nsnam/ns/tutorial/index.html>
- [4] RTP, H. Schulzrinne, S. Casner and R. Frederick, RFC 3550
- [5] RTSP, H. Schulzrinne, A. Rao and R. Lanphier, RFC 2326
- [6] H265/ACC Kompressor http://www.atene.com/BB_kompressorAVC.php