Journal of Korea Multimedia Society Vol. 8, No. 4, April 2005(pp. 502-508)

VOD 서버의 네트워크 입출력 대역폭의 효율적인 사용을 위한 동적 혼성 패칭

하숙정[†], 이경숙[††], 김진규[†††]

요 약

본 논문에서는 VOD 비디오 서버의 제한된 네트워크-입출력 대역폭을 효율적으로 사용하기 위해 탐욕 패칭과 점진 패칭의 장점을 취하여 VOD 시스템의 성능을 향상시킬 수 있는 동적 혼성 패칭 방법을 제안한다. 제안하는 방법은 요청 도착 간격이 패칭 윈도우의 크기보다 작은 비디오 요청은 점진 패칭을 사용하며 나머지 비디오 요청은 탐욕 패칭을 사용한다. 더욱이 제안한 패칭 기법은 탐욕 패칭을 개선시키기 위해 최근의 특정 패칭 멀티캐스트를 정규 멀티캐스트로 확장한다. 그 결과 새로운 요청을 위한 패칭 멀티캐스트 데이터가 비디오의 시작부터 스큐 지점까지로 되어 디스패치된 채널의 사용 시간이 대폭 감소된다. 시뮬레이션 결과는 제안한 기법이 점진 패칭보다 이탈율과 평균 서비스 지연시간에 있어서 우수함을 보여준다.

Dynamic Hybrid Patching for the Efficient Use of VOD Server's Network-I/O Bandwidth

Sook-Jeong Ha[†], Kyung-Sook Lee[††], Jin-Gyu Kim[†††]

ABSTRACT

This paper proposes a dynamic hybrid patching technique that can improve the performance of VOD systems by adopting the advantages of both greedy patching and grace patching to use a VOD server's limited network-I/O bandwidth. The proposed technique uses grace patching to the requests for the videos, arrival intervals of which are smaller than the size of patching window, and uses greedy patching to the rest requests. In addition, proposed patching technique expands the latest particular patching multicast into a regular multicast for a new request. In result, the patching multicast data for the new request can be the data from the beginning to the skew point of the video and the holding time of a dispatched channel can exceedingly decrease. Simulation results confirm that the proposed technique is better than grace patching in terms of defection rate and average service latency.

Key words: Grace Patching(점진 패칭), Greedy Patching(탐욕 패칭), Hybrid Patching(혼성 패칭), Multicast(멀티캐스트), Arrival Interval(도착 간격), VOD(주문형 비디오)

1. 서 론

인터넷을 통한 교육과 엔터테인먼트를 위해 주문형 비디오(Video-on-Demand) 서비스가 많이 이용되고 있다. VOD 서비스는 가장 각광받고 있는 멀티미디어 서비스 중 하나로 디지털 비디오 도서관, 주문형 뉴스, 원거리 교육과 같은 많은 멀티미디어 응용을 위한 중요한 기술이다[1]. 하나의 비디오 스트

※ 교신저자(Corresponding Author) : 김진규, 주소 : 경상북도 상주시 가장동(742-711), 전화 : 054)530-5326, FAX : 054)530-5329, E-mail : kjg@sangju.ac.kr
접수일 : 2004년 4월 12일, 완료일 : 2004년 11월 8일
[†] 경북대학교 전자전기컴퓨터학부 초빙교수
(E-mail : sjha55@kornet.net)
[††] 준회원, 대구가톨릭대학교 컴퓨터정보통신공학부 강의 전담 교수
(E-mail : lks8682@hotmail.com)
[†††] 상주대학교 전자전기공학부 조교수

림을 유지하는데 필요한 VOD 서버의 자원을 비디오 채널이라고 하는데 비디오 서버의 가용 채널 수는 통신 대역폭에 의해 결정되므로 한정되어 있다. 이러한 채널을 효율적으로 사용하기 위한 멀티캐스트 기법 중에서도 패칭[2,3]은 True VOD를 지원하기 위해 새롭게 서비스할 사용자를 이미 진행 중인 멀티캐스트에 합병시켜 비디오 스트림의 공유를 증가시킴으로써 비디오 서버가 새로운 채널을 할당받아 새로 전송해야할 비디오 데이터를 감소시킨다. 패칭은 정규 멀티캐스트의 수행 여부를 결정하는 방법에 따라 탐욕(greedy) 패칭과 점진(grace) 패칭으로 나눠진다. 본 논문에서는 VOD 서버의 제한된 네트워크 입출력 대역폭을 효율적으로 사용할 수 있도록 두 패칭 방법의 장점을 취할 수 있는 동적 혼성 패칭을 제안한다. 2장에서는 탐욕 패칭과 점진 패칭을 소개하고, 3장에서는 탐욕 패칭과 점진 패칭을 혼합한 혼성 패칭을 제안하고, 4장에서는 시뮬레이션을 통한 성능 평가를 한 후 5장에서 결론을 맺는다.

2. 관련 연구

비디오 서버의 네트워크 입출력 대역폭 요구를 감소시켜 VOD 서비스의 성능을 개선하기 위해 비디오 스트림을 한 사용자에게 유니캐스트하는 것이 아니라 여러 사용자가 공유하는 배칭[4], 피기백킹(piggybacking)[5], 패칭과 같은 멀티캐스트 기법이 제안되었다. 배칭은 비디오 스트림의 공유를 증가시킬 수 있지만 일정 기간동안 동일 비디오의 요청이 모일 때까지 기다렸다가 한꺼번에 서비스하기 때문에 사용자의 서비스 지연시간이 길다. 피기백킹은 배칭보다 서비스 지연시간이 짧지만 피기백 율의 변화가 ±5% 범위에 있어야 하므로 합병할 수 있는 스트림의 수가 제한되어 있으며 특별한 하드웨어 환경을 요구한다. 패칭은 서비스 지연 시간과 멀티캐스트 스트림 공유간의 상반관계를 해결하기 위해 제안되었다.

패칭에서 서버의 채널은 비디오 데이터 전체를 멀티캐스트하는 정규 채널 또는 비디오의 처음부터 일부분만 멀티캐스트하는 패칭 채널로 사용될 수 있다. 비디오 v의 요청을 위해 정규 멀티캐스트가 시작된 시간이 t_r이라고 할 때, t_r이후 특정 기간 안에 새로 서비스되는 동일 비디오의 요청에 대해서는 패칭 멀티캐스트가 수행된다. 즉 새 요청의 서비스 시작 시점이 t라고 할 때, 서버는 이미 정규 채널로 전송되어 공유할 수 없는 비디오의 시작부터 $(t-t_r)$만큼의 패칭 스트림만 새로운 채널로 전송하며, 사용자 스테이션에서는 이 데이터를 수신함과 동시에 t_r부터 이미 진행 중인 최근의 정규 멀티캐스트 스트림을 현재 시점 t부터 $(t-t_r)$ 동안 버퍼링한다. 사용자 스테이션에서는 수신한 패칭 스트림을 모두 재생한 후 버퍼링한 정규 스트림을 이용하여 비디오를 연속으로 재생한다. 이와 같이 패칭은 동일 비디오에 대한 새로운 요청과 최근의 정규 멀티캐스트간의 차이 시간 즉 스큐[2]인 $(t-t_r)$이 특정 기간보다 작으면 패칭 멀티캐스트를 수행하여 채널의 사용 시간을 단축시킨다. 이 특정 기간을 사용자 스테이션에서 정규 스트림을 버퍼링할 수 있는 정규 버퍼의 크기로 사용하는 방법을 점진 패칭이라고 한다. 정규 버퍼의 크기는 패칭 윈도우의 크기라고도 한다. Cai[3]는 점진 패칭의 성능을 최적화하기 위해서는 고정된 패칭 윈도우 크기를 사용하는 것이 바람직함을 보였다. 반면에 정규 멀티캐스트 데이터의 공유를 최대화하기 위해 진행 중인 정규 멀티캐스트가 존재하는 한 패칭 멀티캐스트를 수행하는 방법, 즉 위의 특정 기간이 비디오의 전체 재생기간인 방법을 탐욕 패칭이라고 한다.

그림 1은 VOD 서버의 알고리즘[2] 중에서 서버가 새로 서비스할 요청에 대해 정규 멀티캐스트가 아닌 패칭 멀티캐스트가 가능한 지를 판단하고 그 결과에 따라 새로 디스패치된 채널 *New_Ch*상으로 전송할 비디오 데이터를 결정하는 부분을 보여준다. 사용자가 VOD 서버에게 특정 비디오 v에 대한 서비스를 요청하면 서버는 이 요청을 대기 큐에 저장한다. 서버는 FIFO와 같은 특정 정책에 따라 새로 디스패치된 채널로 서비스할 요청을 대기 큐에서 선택한다. *GreedyPatching()*은 탐욕 패칭인 경우 *GracePatching()*은 점진 패칭인 경우 채널 *New_Ch* 상으로 전송할 비디오 데이터 D를 결정하는 함수이다. 두 알고리즘 모두 현재 요청과 최근의 정규 멀티캐스트와의 스큐가 패칭 윈도우의 크기 PW보다 작은 경우에는 비디오의 처음부터 스큐까지의 데이터만 패칭 멀티캐스트를 수행한다. 그러나 스큐가 PW보다 큰 경우 *GracePatching()*에서는 언제나 새로운 정규 멀티캐스트가 수행되는 반면에 *GreedyPatching()*에서는 진행 중인 멀티캐스트가 존재하는 한 패칭 멀티캐스트가 수행된다. *GreedyPatching()*에서의 단계 2는 스큐가 패칭 윈도우의 크기인 PW보다 커서 사용자 버퍼에 스큐만큼의 정규 스트림의 시작부

Algorithm: Server Main Routine
v: Multicast video on channel *R_Ch*
t: Current time
t_r: Start time of the latest regular multicast in progress on regular channel *R_Ch of* video v
PW: Size of patching window
D: Portion of video data which should be multicast on channel *New_Ch*
l: Playback duration of video v
$v[t_d]$: Video data of video v from the beginning to time t_d

1. Initialize the service token as *(PID = null, RID=null)*
2. If there is no regular multicast of video v in progress
   set *RID=New_Ch*
3. else
   -Modify the service token as *(PID=New_Ch, RID=R_Ch)*
   -Call *GreedyPatching()* or *GracePatching()* to determine the portion of video data, *D*, which should be multicast on channel *New_Ch*
4. Send the service token to the user

GreedyPatching()
1. if $(t-t_r \le PW)$ $D = v[t-t_r]$
2. else $D = v[l-Min(PW, l-(t-t_r))]$

GracePatching()
1. if $(t-t_r \le PW)$ $D = v[t-t_r]$
2. else
   -$D = v[l]$
   -Modify the service token as (*PID=null, RID=New_Ch*) so that *New_Ch* is designated to start a new regular multicast

그림 1. 채널 New_Ch로 전송할 데이터 결정 알고리즘(2)

분을 버퍼링할 수는 없지만 정규 스트림의 끝에서부터 최대 *PW*만큼은 사용자 버퍼에 저장할 수 있는 경우이다. 현재 시점부터 최근 정규 멀티캐스트가 종료될 때까지 남은 기간이 *PW* 이상이라면 *PW* 만큼의 정규 스트림의 끝부분을 공유할 수 있다. 그러므로 새로운 채널 *Free_Ch*로 전송할 패칭 스트림은 비디오의 시작부터 $(l-PW)$까지의 데이터가 된다. 그러나 현재 시점부터 최근 정규 멀티캐스트가 종료되기까지의 기간이 *PW*보다 작다면 현재 시점부터 정규 멀티캐스트의 종료까지 남은 시간에 해당하는 정규 스트림의 끝부분만 공유할 수 있다. 그러므로 *Free_Ch*로 전송할 패칭 스트림은 비디오 시작부터 $(t-t_r)$까지의 데이터이다.

P2Cast[6]는 유니캐스트 연결에만 의존하는 동등 계층(peer-to-peer) 접근에 기반을 두고 있다. P2Cast는 패칭과 달리 사용자가 패칭 서버의 역할을 하기 위해 비디오 데이터를 캐시하고 있어야 하며, 한 세션 안에 새로운 사용자가 추가될 때마다 멀티캐스트 트리 형성을 위한 처리과정을 거쳐야 한다. 사용자가 비디오 데이터의 시작 부분을 10% 이상 버퍼링할 수 있으며 서버의 부하가 높을 때 패칭보다 요청 거절율이 조금 우수하지만, 한 세션 안의 모든 사용자 스테이션에서 결함을 복구할 수 있는 능력이 요구된다.

3. 동적 흔성 패칭

탐욕 패칭과 점진 패칭을 비교하기 위해 다음과 같은 예를 고려해보자. 비디오 v의 길이 l은 90분, 각 사용자의 패칭 윈도우 *PW*는 5분, 요청 도착간격은 6분으로 R_0부터 R_9까지 10개의 요청이 0분부터 차례로 도착한다고 가정할 때 서버에서 전송할 총 데이터는 다음과 같다.

- 탐욕 패칭: R_0는 첫 요청이므로 정규 멀티캐스트로 서비스되며 l분 동안의 비디오가 전송된다. 나머지 요청 R_i $(1 \le i \le 9)$는 모두 R_0를 위한 정규 멀티캐스트가 진행되고 있는 상태에서 서비스되므로 패칭 멀티캐스트로 서비스된다. 요청의 도착간격이 6분으로 패칭 윈도우 크기보다 크므로 최근 정규 멀티캐스트의 마지막 5분 동안의 비디오만 공유할 수 있으므로 처음부터 $l-PW$인 85분에 해당하는 데이터를 새로 전송한다. 그러므로 $n=10$ 일 때 서버가 전송할 총 데이터는 다음과 같다.

$$D_{greedy} = 90 + \sum_{i=0}^{9} (90-5) = 855$$

- 점진 패칭: R_0는 정규 멀티캐스트로 서비스되며 l분 동안의 비디오가 전송된다. 나머지 요청 R_i $(1 \le i \le 9)$들은 모두 도착간격이 6분으로 R_{i-1}을 위한 최근의 정규 멀티캐스트의 패칭 윈도우를 벗어났으므로 새로운 정규 멀티캐스트로 서비

스된다. 그러므로 서버가 전송할 총 데이터는 다음과 같다.

$$D_{grace} = \sum_{i=0}^{9} 90 = 900$$

요청 도착간격이 PW보다 큰 경우 l분 동안 도착한 동일 비디오 v에 대한 n개의 요청에 대해 탐욕 패칭을 사용했을 때 점진 패칭보다 감소되는 데이터 전송량은 다음과 같다.

$$\begin{aligned} D_{grace} - D_{greedy} &= \sum_{i=0}^{n-1} l - (l + \sum_{i=1}^{n-1}(l - PW)) \\ &= (n \times l) - (l + (n-1)(l - PW)) \\ &= (n-1)l - (n-1)(l - PW) \\ &= (n-1)(l - (l - PW)) = (n-1)PW \end{aligned}$$

이와 같이 비디오 v에 대한 요청 도착 간격이 PW보다 큰 경우엔 마지막 일부 비디오 데이터만이라도 공유하는 탐욕 패칭이 우수함을 알 수 있다. 그러므로 비디오 서버에 도착한 요청들을 주기적으로 분석한 후 요청 도착 간격이 PW보다 큰 비디오에 대해서는 탐욕 패칭을, 그렇지 않은 비디오에 대해서는 점진 패칭을 적용하는 것이 효과적일 것이다.

탐욕 패칭을 분석하기 위해 다음과 같은 예를 고려해보자. 비디오 v에 대한 3개의 요청 R_0, R_1, R_2가 차례대로 비디오 서버에 도착하며, 비디오 v의 길이 l은 10분, 패칭 윈도우의 크기 PW는 2분이라고 가정하자. $skew(i, j)$는 현재 비디오 v에 대한 최근의 정규 멀티캐스트가 요청 R_i를 위해 진행되고 있을 때, 이 정규 멀티캐스트와 요청 R_j의 서비스 시작 시간과의 차이 즉 스큐를 나타낸다. 그림 2는 기존의 탐욕 패칭을 적용했을 때 채널을 통해 전송되는 데이터를 보여준다. 요청 R_1과 R_2는 요청 R_0를 위한 최근의 정규 멀티캐스트가 진행 중이며 스큐가 PW보다 크므로 비디오의 시작부터 $(l-PW)$까지의 데이터가 패칭 멀티캐스트된다. 그러나 그림 3과 같이 R_2가 스케줄될 때 R_1에 대한 요청이 정규 멀티캐스트로 서비스되고 있다고 가정한다면, R_2는 단지 스큐만큼의 패칭 멀티캐스트로 서비스될 수 있다. 그림 2에 비해 그림 3에서 증가된 전송량은 2분의 데이터이지만 새 요청에 대해 감소된 전송량은 7분 동안의 데이터로, 전체 감소량은 5분에 해당하는 데이터임을 알 수 있다. 새로운 요청 R_2를 서비스할 때 그림 2가 아닌 그림 3과 같은 결과를 얻기 위해서 서버는 요청 R_1이 전송할

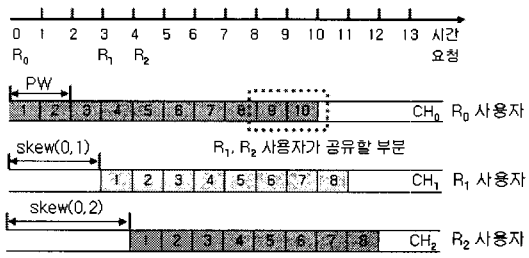

그림 2. 탐욕 패칭

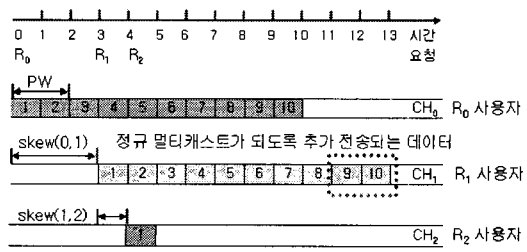

그림 3. 정규 멀티캐스트로의 확장 예

데이터를 $(l-PW)$까지가 아닌 l까지로 확장하면 될 것이다. 최근 정규 멀티캐스트의 시작 시간은 t_r, 최근 정규 멀티캐스트와의 스큐가 PW보다 커서 비디오의 시작부터 l-$Min(PW, l-(t_p-t_r))$까지의 데이터를 전송하는 가장 최근에 발생한 패칭 멀티캐스트의 시작 시간은 t_p, 새로운 요청의 스케줄링 시점 t와 최근 패칭 멀티캐스트와의 스큐는 $(t-t_p) \le PW$ 라고 가정하자. 패칭 멀티캐스트를 정규 멀티캐스트로 확장한다면 비디오의 끝까지 데이터를 전송하기 위해 $Min(PW, l-(t_p-t_r))$ 만큼 데이터 전송량이 증가된다. 그러나 새로운 요청을 서비스하기 위해 디스패치된 채널은 비디오의 시작부터 l-$Min(PW, l-(t-t_r))$까지가 아닌 방금 패칭 멀티캐스트에서 정규 멀티캐스트로 확장된 새로운 최근의 정규 멀티캐스트와의 스큐인 비디오의 시작부터 $(t-t_p)$까지만 전송하면 된다. $(t-t_p) \le PW$ 이므로 새 채널로 전송할 데이터는 최소 $(l-2\times PW)$ 만큼 감소된다. 일반 영화 비디오의 길이가 90분이고 패칭 윈도우가 5분이라고 가정한다면 정규 멀티캐스트로 확장됨으로써 증가되는 채널 사용시간은 최대 5분이며 새 요청으로 인해 감소되는 새 채널의 사용 시간은 최소 80분으로 상당히 감소된다.

그러므로 본 논문에서는 VOD 서버가 주기적으로 서버에 도착한 요청을 분석하여 비디오 별로 다음

주기 동안 적용할 패칭 기법을 동적으로 결정하며, 탐욕 패칭을 개선시키기 위해 기존의 패칭 멀티캐스트를 정규 멀티캐스트가 되도록 확장하는 동적 혼성 패칭을 제안한다. 서버는 탐욕 패칭을 적용하여 새로운 비디오 요청을 스케줄할 때 패칭 멀티캐스트의 확장이 필요한 지를 판단해야 한다. 이를 위해 VOD 서버는 비디오의 시작부터 $l\text{-}Min(PW,\ l\text{-}(t_p\text{-}t_r))$까지의 데이터가 패칭 멀티캐스트되는 가장 최근의 패칭 멀티캐스트의 시작 시간 t_p를 유지한다. 새로운 요청의 스큐 $(t\text{-}t_p)$가 PW보다 작다면 이 최근의 패칭 멀티캐스트를 정규 멀티캐스트로 확장한다. 그렇다면 이렇게 확장된 새로운 정규 멀티캐스트의 시작 시간 t_p와 새 요청간의 스큐 $(t\text{-}t_p)$는 PW보다 작게 되고, 결국 서버는 새 요청에 대해 이 스큐만큼의 비디오 시작부분만 패칭 멀티캐스트로 서비스할 수 있다.

4. 시뮬레이션 및 성능 평가

3장에서 제안한 혼성 패칭의 성능을 평가하기 위해 시뮬레이션을 수행하여 [2]에서 탐욕 패칭보다 우수한 것으로 나타난 점진 패칭과 성능을 비교한다. 성능 평가에서 고려된 평가 항목은 VOD 서버의 성능 평가에 주로 사용되는 비디오 요청 도착율에 따른 사용자의 이탈율, 평균 서비스 지연시간, 공평성이다. 비디오의 길이는 [2]와 같이 일반적인 영화 길이인 90분으로, 패칭 윈도우의 크기는 5분으로, 비디오의 개수 N은 100으로 설정하였다. 사용자가 임의의 비디오 i를 요청할 확률 p_i는 $Zipf$ 법칙을 따라 $1/(i^z \sum_{j=1}^{N} \frac{1}{j^z})$ 로 설정하였으며, 스큐 인자인 z는 VOD 응용에서 일반적으로 나타나는 0.7을 사용하였다[3,7]. 일단 사용자가 비디오를 보게 되면 끝까지 순차적으로 본다고 가정하며, 사용자의 이탈 시간은 1분에서 5분까지로 랜덤 분포로 설정하였다. 서버는 시스템 초기에는 모든 비디오에 대해 점진 패칭을 적용하되 이후로는 30분을 주기로 이전 30분 동안 도착한 요청에 대해 각 비디오 별로 평균 요청 도착 간격을 계산하고 이를 근거로 점진 패칭과 탐욕 패칭 대상을 동적으로 결정하여 시스템의 최근 상황에 적응적이 되도록 한다.

비디오 서버가 1200 개의 비디오 스트림을 동시에 전송할 수 있을 때 서버의 부하에 따른 성능을 비교하기 위해 포아송 분포로 도착한 평균 사용자 요청 도착율이 분당 10에서 90일 경우 각각에 대해 360분 동안 도착한 사용자 요청을 대상으로 시뮬레이션을 수행하였다. 성능은 시뮬레이션 중단으로 인해 서비스되지 못한 사용자들을 제외시키기 위해 시작부터 350분 동안 도착한 사용자 요청에 대해서만 비교 평가하였다.

그림 4는 점진 패칭과 혼성 패칭 두 기법의 요청 도착율에 따른 이탈율을 보여준다. 이탈율은 시뮬레이션 기간동안 도착한 전체 비디오 요청 중에서 대기 큐에서 서비스를 기다리다가 이탈한 사용자 요청이 차지하는 비율로서 이탈한 사용자 요청 수를 전체 요청 수로 나눈 값이다. 그림 4에서 두 가지 기법 모두 이탈율이 거의 0인 경우를 제외했을 때, 동적 혼성 패칭의 이탈율이 점진 패칭보다 평균 30.7% 정도 개선되었다.

그림 5는 두 패칭 기법의 비디오 요청 도착율에 따른 평균 서비스 지연시간을 보여준다. 평균 서비스

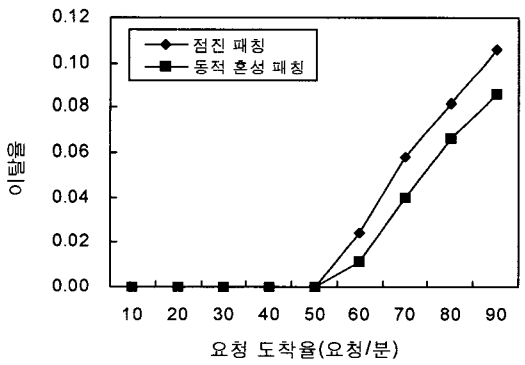

그림 4. 요청 도착율과 이탈율

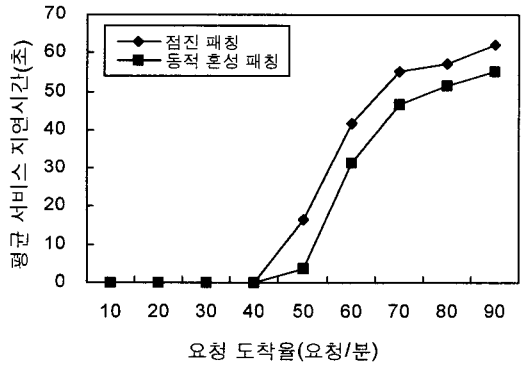

그림 5. 요청 도착율과 평균 서비스 지연시간

지연시간은 서비스를 받은 전체 사용자들이 비디오를 요청한 후 서비스를 받을 때까지 대기 큐에서 기다린 평균 시간이다. 그림 5에서 서비스 지연 시간이 0인 경우를 제외했을 때, 제안한 동적 혼성 패칭의 평균 서비스 지연시간이 점진 패칭보다 평균 27.8% 정도 개선되었다. 그림 4와 그림 5에서 제안한 동적 혼성 패칭이 점진 패칭보다 우수한 것으로 나타난 것은 주기적으로 서버에 도착한 요청들의 평균 도착 시간을 근거로 다음 주기 동안 각 비디오 요청에 대해 적용할 패칭 방법을 선택하는 동적 방법을 사용함으로써 비디오별로 적절한 패칭 방법을 적용할 수 있었기 때문이다. 그리고 제안한 패칭 기법에서는 탐욕 패칭을 적용하는 경우 이미 비디오의 시작부터 l-$Min(PW, l-(t_p-t_r))$까지의 패칭 스트림을 전송하고 있는 최근의 패칭 멀티캐스트와 새 요청과의 스큐를 분석하였다. 이 스큐가 PW 보다 작은 경우에는 최근 패칭 멀티캐스트를 최근 정규 멀티캐스트로 확장하며, 이로 인해 새 요청이 확장된 최근의 정규 멀티캐스트의 패칭 윈도우 안에 있게 되어 새로 전송해야 하는 비디오 데이터가 PW 이하인 비디오의 처음부터 스큐까지로 대폭 감소되었기 때문이다.

그림 6은 비디오 요청 도착율에 따른 두 가지 기법의 공평성(*fairness*)을 보여준다. 공평성은 모든 비디오 요청에 대한 평균 이탈율이 d이고 비디오 i에 대한 요청 이탈율이 d_i라고 할 때 $1-\sqrt{\frac{\sum_i^N (d_i-d)^2}{N-1}}$ 와 같이 계산된다[2]. 공평성은 두 가지 패칭 기법이 거의 비슷한데 이는 점진 패칭과 혼성 패칭 모두 가용 채널이 디스패치되었을 때 요청이 도착한 순서대로 스케줄링을 했으며 특정 요청에 대해 우선적인 서비스를 제공하지 않는 기법이기 때문이다.

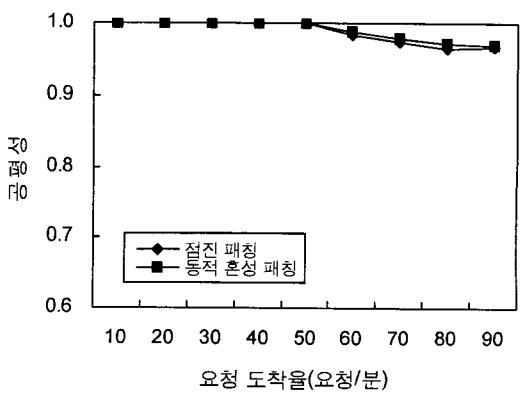

그림 6. 요청 도착율과 공평성

시뮬레이션 결과 제안한 동적 혼성 패칭은 VOD 서버의 부하에 따라 비디오별로 적절한 패칭 기법을 적용하고 최근의 정규 멀티캐스트뿐만 아니라 최근의 패칭 멀티캐스트에 대한 정보를 유지하고 분석하여 탐욕 패칭에서 전송될 패칭 스트림의 양을 감소시킴으로써, 사용자의 이탈율을 감소시켜 서버의 처리량을 증가시킬 뿐만 아니라 사용자에게 보다 빠른 응답 시간을 제공할 수 있는 기법으로 나타났다.

5. 결 론

비디오 서버의 네트워크 입출력 대역폭을 효율적으로 사용하여 많은 사용자들에게 양질의 VOD 서비스를 제공하기 위한 대표적인 멀티캐스트 기법으로 패칭이 있다. 본 논문에서는 VOD 서버의 제한된 네트워크 입출력 대역폭을 보다 효율적으로 사용하도록 서버의 최근 상태에 동적으로 적응할 수 있는 동적 혼성 패칭 기법을 제안하였다. 제안한 동적 혼성 패칭 기법에서는 주기적으로 서버에 도착한 요청들을 분석한 후 다음 주기 동안 각 비디오에 적용할 패칭 기법을 점진 또는 탐욕 패칭 중 하나로 결정하였다. 탐욕 패칭을 개선시키기 위해 새로 서비스할 요청과 최근 패칭 멀티캐스트와의 스큐가 패칭 윈도우 크기보다 작은 경우 진행 중인 패칭 멀티캐스트를 정규 멀티캐스트로 확장하였다. 시뮬레이션 결과 공평성은 두 패칭이 거의 동일하였으나 제안한 동적 혼성 패칭은 최근의 비디오 요청에 대한 분석을 근거로 각 비디오마다 적절한 패칭 기법을 선택하는 동적 기법을 사용함으로써 이탈율과 평균 서비스 지연시간에 있어서 점진 패칭보다 우수함을 알 수 있었다. 향후 탐욕과 점진 패칭을 적용하기 위한 보다 좋은 기준을 찾아서 혼성 패칭의 성능을 향상시키고자 한다.

참 고 문 헌

[1] Jani Huoponen and Thorsten Wagner, "Video on Demand: A Survey," Telecommunication Networks Project, 1, http://fiddle.visc.vt.edu/courses/ee4984/Projects1996/huoponen_wagner/huoponen_wagner. html, 1996.

[2] K. Hua, Y. Cai, and S. Sheu, "Patching: A Multicast Technique for True Video-on-Demand Services," In Proc. ACM Multimedia, pp. 191-200, 1998.

[3] Y. Cai, K. Hua, and K. Vu, "Optimizing Patching Performance," In Proc. SPIE/ACM Conference on Multimedia Computing and Networking, pp. 204-215, 1999.

[4] A. Dan, D. Sitaram, and P. Shahabuddin, "Scheduling Polices for an On-Demand Video Server with Batching," In Proc. of the 2nd ACM Multimedia Conference, pp. 25-32, 1994.

[5] L. Golubchik, J. Lui, and R. Muntz, "Adaptive Piggybacking: Arrival Technique for Data Sharing in Video-on-Demand Service," ACM Multimedia Systems, Vol.4, No.3, pp.140-155, 1996.

[6] A.Yang Guo, Hyoungwon Suh, Jim Kurose, and Don Towsley, "P2Cast: Peer-to-peer Patching Scheme VoD Service," In Proceedings of the 12th International Conference of World Wide Web, pp. 301-309, 2003.

[7] A.Chervenak, D. Patterson, and R. Katz, "Choosing the Best Storage System for Video Service," In Proc. of ACM Multimedia 95, pp. 109-119, Aug. 1995.

하 숙 정

1988년 계명대학교 전자계산학과(학사)
1990년 중앙대학교 전자계산학과(석사)
1998년 대구가톨릭대학교 전산통계학과 전산전공(박사)
2001년~현재 경북대학교 전자전기컴퓨터학부 초빙교수
관심분야 : 모바일 컴퓨팅, 분산 컴퓨팅, 멀티미디어 시스템 등

이 경 숙

1990년 대구가톨릭대학교 수학교육학과(학사)
1993년 대구가톨릭대학교 전산통계학과(석사)
2000년 대구가톨릭대학교 전산통계학과 전산전공(박사)
2002년~현재 경북대학교 전자전기컴퓨터학부 초빙교수
관심분야 : 모바일 컴퓨팅, 분산 컴퓨팅, 멀티미디어 시스템 등

김 진 규

1990년 경일대학교 공과대학 전기공학과(학사)
1994년 경북대학교 대학원 전기공학과(석사)
1998년 경북대학교 대학원 전기공학과(박사)
2000년 경북대학교 전자전기컴퓨터학부 BK21 조교수
2001년~현재 상주대학교 전자전기공학부 조교수
관심분야 : 임베디드 시스템, 모바일 인터넷 등