

논문 2008-45TC-12-11

다중 체인구조를 이용한 Peer-to-Peer 기반 IPTV 시스템 설계

(A Design of Peer-to-Peer Based IPTV System using Multiple Chain Architecture)

김지훈*, 김영한**

(Jihoon Kim and Young Han Kim)

요약

본 논문에서는 다중 체인구조를 이용한 P2P 기반 IPTV 시스템을 제안한다. 제안하는 시스템은 P2P 네트워크의 특징인 peer의 빈번한 가입과 탈퇴 상황에서 단순한 과정으로 네트워크를 재구성하는 장점이 있다. 인터넷과는 달리 ISP가 관리하는 IPTV 환경은 전송회선의 속도와 안정성이 일정수준으로 보장되어 있다. 따라서 IPTV 환경에서의 P2P 네트워크는 단순성 측면을 우선적으로 고려해야 한다. 기존에 제안되어있는 단일 체인구조는 단순성을 강조하였으나 같은 채널을 시청하는 peer의 개수가 증가하면 체인의 끝부분에 연결되어 있는 peer는 상당한 delay가 발생한다. 제안하는 시스템은 이러한 delay 문제를 해결하기 위하여 체인을 여러 개의 레벨로 분리하고 각 레벨은 다시 span으로 나누었다. 레벨과 span으로 분리를 하였지만 기본적인 구조는 체인구조이므로 peer가 join 하거나 departure 할 경우에 단순한 네트워크의 재구성 과정을 제공한다. 수치적인 해석을 통해 본 논문에서 제안한 다중 체인구조를 이용한 P2P 시스템이 단일 체인구조 방식에 비해 delay 및 신뢰도 성능이 우수하다는 것을 보여준다.

Abstract

In this paper, we propose a P2P based IPTV system using a multiple chain architecture. Proposed system is robust to the peer churn. As opposed to the internet, the IPTV network managed by a single ISP has fewer bandwidth constraints and end-to-end connectivity. So, we emphasize preferentially robustness of a P2P network in IPTV environment. A single chain structure which was proposed previously emphasizes simplicity, however there exists considerable delay time at the end part of peers as the number of peers increasing. As a solution to the problem, we propose the scheme which separates the chain into several levels and again divides each level into spans to diminish a delay time. Though the chain is separated into level and span, basic structure of proposed scheme is still a chain structure. So the scheme simplifies the recovery procedure caused by join or departure of peers. We will show the improved performance of proposed scheme rather than single chain structure with respect to the delay time and reliability.

Keywords: IPTV, P2P, peer-to-peer network, multiple chain architecture

I. 서론

실시간 방송서비스를 갖춘 IPTV 네트워크는 채널수가 증가됨에 따라 트래픽 증가가 문제시 되고 있다. 이를 해결하기 위한 방안으로 중앙의 서버 대신 직접 다룬 수신 단말로부터 스트리밍 비디오를 공급받도록 하여 부하를 보완하려는 P2P 방식이 제안되고 있다^[1~4].

P2P 구조는 peer들 간의 연결로 오버레이네트워크가 구성되므로 중간에 있던 peer가 빠져 나가게 되면 해당

* 정회원, 부천대학 e-비즈니스과
(Bucheon College)

** 정회원, 송실대학교 정보통신전자공학부
(Soongsil University)

※ 본 연구는 지식경제부 및 정보통신연구진흥원의 IT 신성장동력핵심기술개발사업[2008-F-015-01, 서비스 가용성을 위한 이동성 관리 기술 연구]과 지식경제부 및 정보통신연구진흥원의 대학 IT연구센터 지원사업의 일부로 수행하였음
[IITA-2008-(C1090-0801-0036)]

접수일자: 2008년8월20일, 수정완료일: 2008년12월10일

peer와 연결되어서 데이터를 주고받던 peer들의 연결이 끊어진다. 따라서 사용자들이 불편을 느끼지 않도록 빠져나간 peer로 인해 끊어진 연결을 빨리 복구시켜 주어야 한다. 특히 IPTV의 특성상 peer들의 가입(join)과 탈퇴(departure)가 빈번하고 예측하기 힘들기 때문에 IPTV를 위한 P2P 구조는 동적인 peer 환경에 단순하고 안정적으로 적용할 수 있어야 한다.

그동안 인터넷 망에서의 P2P 방식의 멀티미디어 스트리밍에 대한 연구는 활발히 진행되어 왔으며, CoolStreaming^[5], PPLive^[6], PPStream^[7], UUSee^[8] 등은 실제로 운영되고 있다. 대표적인 P2P 구조는 트리구조와 메시구조로 구분할 수 있다. 트리구조는 peer들 간의 데이터의 전송경로가 일정하기 때문에 각 peer에게로의 데이터 전송이 빠르다는 장점이 있는 반면 peer가 탈퇴한 경우에 구조의 재구성이 복잡하다는 단점이 있으며 이를 해결하기 위해서 다계층 트리구조등의 방법이 제안되었다^[9~11]. 메시구조는 동적인 P2P 환경에 적합한 구조이나 전체적으로 데이터 전송이 효율적이지 못하다는 단점이 있다^[5, 12]. 이러한 P2P 방식에 대한 연구는 지금까지 대부분 네트워크 환경과 단말기의 형태가 다양한 인터넷 망을 대상으로 진행되었다.

단일 ISP에 의해 관리되는 IPTV 환경은 인터넷과 비교할 때 전송회선의 속도와 안정성 측면에서 나은 품질을 제공한다. 따라서 IPTV 환경에서 운영되는 P2P 구조는 인터넷에서의 경우와는 달리 속도와 안정성이 보장된다는 가정 하에 설계 할 수 있다. P2P 구조를 설계함에 있어서 고려해야 할 중요한 요소 중 하나는 빈번한 peer의 가입과 탈퇴로 인한 네트워크 토폴로지의 동적인 변화에 대하여 얼마나 단순하고 빠르게 적응하는지의 정도이다. 제안하는 다중 체인구조는 peer의 출입이 빈번한 P2P 환경에서 단순하고 안정적인 재구성 과정을 제공한다.

IPTV 환경에서 Live TV P2P구조에 대한 연구는 체인구조를 적용한 방식이 제안되어있다^[13]. 이 논문에서는 동일 ISP가 운영하는 IPTV 환경에 P2P 방식을 적용함으로써 멀티캐스트 방식에 비해서 네트워크 비용이 줄어든다는 것을 보여주었다. 하지만 이 논문에서 제안한 단일 체인구조는 한 채널에 접속한 peer의 개수가 많아질 때 나중에 접속한 peer는 지연시간이 길어진다는 문제가 있으며, 모든 peer가 하나의 체인으로 구성되어 있으므로 중간에 위치한 peer에 문제가 생겼을 때 해당 peer의 뒤에 위치한 peer들 모두가 영향을 받게 되어 시스템의 신뢰성을 저하 시킬 수 있다.

본 논문에서는 이러한 체인구조의 문제점을 해결하기 위한 방법으로 다중 체인구조(multiple chain architecture)를 제안한다. 다중 체인구조는 하나의 체인구조로 연결된 peer들을 여러 개의 레벨 및 span으로 분리한 구조로써 체인구조의 문제점을 보완하였으며 peer의 출입이 빈번한 P2P 환경에서 단순하고 안정적인 재구성 과정을 제공한다.

본 논문의 구성은 서론에 이어 제II장에서는 본 논문에서 제안하는 다중 체인구조의 시스템과 peer 관리 알고리즘을 소개한다. 그리고 제III장에서는 제안한 구조의 특성을 몇 가지 관점에서 수학적으로 해석한다. 마지막으로 제IV장에서 결론을 맺는다.

II. 다중 체인 기반 IPTV 시스템 구조

1. 시스템 개요

본 논문에서 제안하는 다중 체인구조는 그림 1과 같다. 하나의 DSLAM 내에서 같은 채널을 시청하는 peer들은 L 개의 레벨로 구성되며 각 레벨은 다시 S 개의 span으로 나누어진다.

하나의 span에는 $(L-i)$ (i 는 1부터 L 까지의 레벨 순서번호)개의 peer들이 체인구조로 연결되어 있다. 즉, 레벨 1의 각 span에는 $L-1$ 개의 peer들이 체인형태로

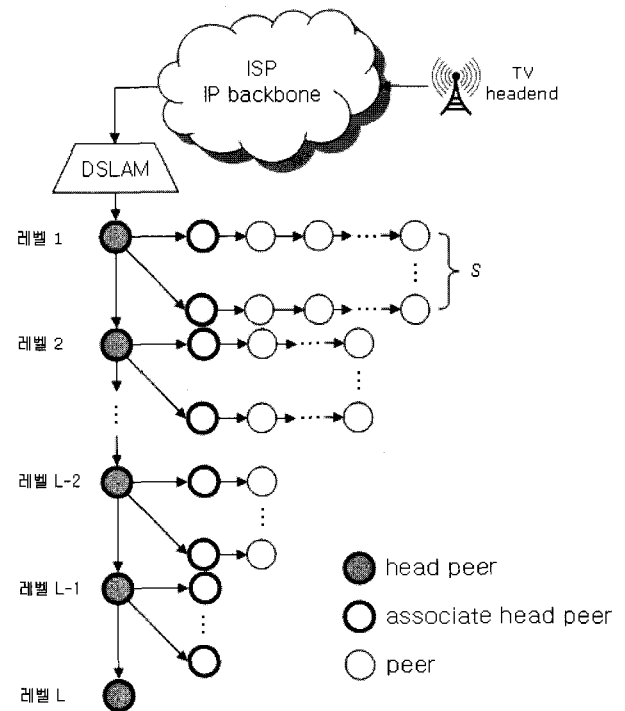


그림 1. 다중 체인구조
Fig. 1. Multiple Chain Architecture.

연결되며 레벨 2의 span에는 $L-2$ 개, 레벨 3은 $L-3$ 개와 같은 식으로 레벨이 증가할수록 span의 체인에 연결되는 peer는 1개씩 줄어든다. 이렇게 설계한 이유는 DSLAM에서의 hop 수가 레벨이 증가할수록 늘어나는 것을 보상해주기 위해서이다. 각 레벨의 맨 앞에 있는 peer를 head peer라고 정의하고 두 번째 peer를 associate head peer라고 정의한다. 각 레벨은 head peer들로 연결된다. 레벨 l 의 head peer는 레벨 l 에 속한 모든 peer 및 레벨 $l+1$ 의 head peer에 대한 정보를 가지고 있으며, 레벨 l 에 새로운 peer가 가입하거나 레벨 l 에 속한 peer가 탈퇴하면 레벨 정보를 갱신하여 현재 레벨의 각 span에 속한 peer의 개수를 파악하고 있다. 또한 모든 레벨의 head peer는 주기적으로 자신의 정보를 ISP tracker에게 제공하며 새로운 peer가 가입을 원할 때 tracker는 head peer로부터 받은 정보를 참조하여 새로운 peer의 위치를 지정하게 된다.

Associate head peer는 자신이 속한 span의 모든 peer에 대한 정보를 가진다. 만일 head peer가 갑작스런 문제로 인해 제대로 된 과정을 거치지 않고 탈퇴하게 되는 경우에 첫 번째 span의 associate head peer가 새로운 head peer가 되며, 나머지 span의 associate head peer는 새로운 head peer에게 span의 peer 정보를 제공한다.

다중 체인구조에서 생방송 TV 영상데이터가 peer들에게 전달되는 과정은 다음과 같다. Head-end system에서 전송된 생방송 TV 영상데이터는 ISP에 의해 관리되는 IP망을 거치고 DSLAM을 통해서 첫 번째 레벨의 head peer $P_{1,1}$ 로 전송된다. $P_{1,1}$ 은 같은 레벨에 속한 S 개의 span 그룹 peer에게는 체인구조로 데이터를 전송하며, 다음 레벨의 head peer에게도 데이터를 전송함으로써 모든 레벨의 peer들이 순서대로 전송받을 수 있게 된다.

본 논문에서 제안하는 다중 체인구조는 head peer를 제외한 대부분의 peer가 체인구조로 연결되어 있으므로 peer들 간의 데이터전송이 단순하며 또한 peer들의 가입 및 탈퇴가 빈번한 환경에서 전체 구조의 재구성성이 간단하다는 장점이 있다.

2. Peer 관리 알고리즘

이 절에서는 다중 체인구조에서 peer가 채널에 가입하거나 채널로부터 탈퇴하는 경우에 구조를 재구성하는 알고리즘에 대해서 기술한다. 우선 제안하는 알고리즘의 동작을 위해서 몇 가지 가정이 필요하다. 먼저 peer

표 1. 알고리즘 기술을 위한 용어 정의
Table 1. Notations for algorithm description.

PI	채널에 가입하려는 peer
LEVEL(n)	n번째 레벨
$P_{i,j,k}$	LEVEL(i)의 j번째 span에 속한 peer 중에서 k번째 위치에 있는 peer
$P_{i,1}$	LEVEL(i)의 head peer
$P_{i,j,2}$	LEVEL(i)의 j번째 span의 associate head peer
N	DSLAM내에서 같은 채널을 시청하는 peer 개수
S	하나의 레벨에 속한 span 개수
L	전체 레벨 개수

가 가입할 때 위치를 지정해주기 위해서 ISP에서 제공하는 tracker가 존재한다. 그리고 모든 레벨의 head peer는 주기적으로 레벨의 정보를 tracker에게 전송하며, 인접한 peer들 간에는 서로의 존재를 확인하기 위하여 주기적으로 'alive' 메시지를 주고받는다. 알고리즘을 기술하는데 필요한 용어는 표 1과 같다.

가. Peer 가입

새로운 peer PI가 채널에 가입하기 위하여 tracker에게 등록하면 tracker는 LEVEL(1)부터 LEVEL(L)까지 순서대로 head peer의 정보를 이용해서 PI가 가입할 수 있는지 확인한다. tracker는 우선 LEVEL(1)의 각 span에 대하여 peer 개수를 점검하여 새로운 peer를 가입시킬지 여부를 결정한다. LEVEL(1)에 PI가 가입할 수 있는 공간이 있으면 LEVEL(1)의 head peer에게 PI를 가입시킬 것을 지시하고, head peer는 해당 span의 마지막 peer 다음위치에 PI를 가입시킨 후에 레벨 정보를 갱신한다. LEVEL(1)의 어떤 span에도 PI를 가입시킬 공간이 없는 경우에는 LEVEL(2)에 대하여 가입가능여부를 확인하며 계속해서 LEVEL(L)까지 확인과정을 반복한다. 이때 가입하려는 레벨에 peer가 하나도 없으면 tracker는 PI를 head peer로 지정하고, 가입하려는 span에 peer가 없으면 head peer는 PI를 associate head peer로 지정한다. 만일, 어떤 레벨에도 PI가 가입할 수 있는 자리가 없으면 tracker는 레벨수를 1 증가시키고 PI를 LEVEL(1)의 첫 번째 span 마지막 peer 다음 위치에 가입시킨다. 레벨수가 1 증가하게 되면 모든 레벨에서 수용할 수 있는 peer의 개수가 각 span 별로 1개씩 증가하게 되므로 각 레벨의 head peer는 정보를 갱신해야 한다. Peer의 가입과정을 pseudo code로 나타내면 다음과 같다.

Algorithm 1 Peer Join

```

for i=1 to L
  j←Addable(i)
  if j≠0 then
    if no peer in LEVEL(i)
      Add(i, PI);
      Assign_head_peer(PI);
    else if no peer in span j
      AddLast(i, j, PI);
      Assign_associate_head_peer(PI);
    else
      AddLast(i, j, PI);
    end if
    Adjust(i);
    return;
  end if
end for
L ← L+1
Adjust(1);
AddLast(1, 1, PI );
for i=2 to L
  Adjust(i);
end for

Addable(i)
for k=1 to S
  if (number of peers in the span < L-i) then
    return k;
  end if
end for
return 0;
end if
    
```

나. Peer 탈퇴

채널에 속해있던 peer가 탈퇴할 때에 head peer일 경우와 아닐 경우에 따라 다른 과정을 거치게 된다. 각각의 경우를 구분하여 탈퇴 과정을 설명한다.

(1) Peer의 정상적인 탈퇴

Head peer가 아닌 peer $P_{i,j,k}(k \neq 1)$ 가 채널그룹을 탈퇴하려면 $P_{i,j,k}$ 는 $P_{i,j,k-1}$ 과 $P_{i,j,k+1}$ 및 head peer $P_{i,1}$ 에게 탈퇴신호를 전송한다. $P_{i,j,k}$ 로부터 탈퇴신호를 받은 $P_{i,j,k-1}$ 은 $P_{i,j,k}$ 에게로 데이터전송을 중단하고 $P_{i,j,k+1}$ 에게 전송한다. 또한 $P_{i,1}$ 은 레벨 목록에서 $P_{i,j,k}$ 를 삭제하고 레벨 정보를 갱신한 후에 $P_{i,j,2}$ 에게 변경된 레벨 정보를 전송한다. 단, associate head peer가 탈퇴한 경우에는 head peer는 새로운 associate head peer에게 레벨에 속한 peer들의 정보를 전송한다.

(2) Peer의 비정상적인 탈퇴

정상적으로 채널그룹을 탈퇴하지 않고 $P_{i,j,k}$ 에 문제가

생겨서 더 이상 전송을 할 수 없는 경우에는 $P_{i,j,k}$ 가 탈퇴신호를 보낼 수가 없다. 이런 문제를 해결하기 위해서 인접한 peer들은 'alive' 메시지를 주기적으로 주고받으며, 'alive' 메시지가 일정한 시간동안 전송되지 않으면 상대 peer가 탈퇴했다고 판단한다. 즉, $P_{i,j,k+1}$ 은 t_{max} 동안 $P_{i,j,k}$ 로부터 'alive' 메시지의 전송이 없으면 head peer $P_{i,1}$ 에게 상황을 알린다. 그러면 $P_{i,1}$ 은 $P_{i,j,k-1}$ 로 하여금 $P_{i,j,k}$ 에게 전송을 중단하고 $P_{i,j,k+1}$ 에게 데이터를 전송하도록 하고 레벨 정보를 갱신한다. Peer의 탈퇴 과정을 pseudo code로 나타내면 다음과 같다.

Algorithm 2 Graceful departure of peer $P_{i,j,k}(k \neq 1)$

```

send Departure_Message( $P_{i,j,k}$ ) from  $P_{i,j,k}$  to  $P_{i,j,k-1}, P_{i,j,k+1}$ ;
transmit Data( $P_{i,j,k-1}$ ) to  $P_{i,j,k+1}$ ;
Adjust(i);
send Level_Info(i) from  $P_{i,1}$  to  $P_{i,j,2}$ ;
    
```

Algorithm 3 Accidental departure of peer $P_{i,j,k}(k \neq 1)$

```

if Timer( $P_{i,j,k+1}$ ) >  $t_{max}$  without 'alive' message then
  send Departure_Message( $P_{i,j,k}$ ) from  $P_{i,j,k+1}$  to  $P_{i,j}$ ;
  transmit Data( $P_{i,j,k-1}$ ) to  $P_{i,j,k+1}$ ;
  Adjust(i);
  send Level_Info(i) from  $P_{i,1}$  to  $P_{i,j,2}$ ;
end if
    
```

(3) Head peer의 정상적인 탈퇴

Head peer가 탈퇴하는 과정은 다음과 같다. 탈퇴하려는 head peer $P_{i,1}$ 은 우선, 자신에게 데이터를 전송해 주던 이전 레벨의 head peer $P_{i-1,1}$ 과 자신이 데이터를 전송하던 associate head peer들 $P_{i,j,2}(j=1,2,\dots,S)$, 그리고 다음 레벨의 head peer $P_{i+1,1}$ 에게 탈퇴신호를 전송한다. 또한, $P_{i,1}$ 은 첫 번째 span의 associate head peer $P_{i,1,2}$ 에게 head peer가 될 것을 지시하고 i 레벨의 다른 peer에

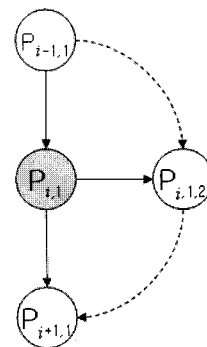


그림 2. head peer의 탈퇴로 인한 전송경로의 변화
Fig. 2. Change of transmission path due to departure of head peer.

대한 정보를 전송한다. 탈퇴신호를 받은 $P_{i-1,1}$ 은 $P_{i,1}$ 로의 전송을 중단하고 $P_{i,1,2}$ 에게 데이터를 전송한다. $P_{i,1,2}$ 는 자신이 head peer가 되었다는 사실을 tracker에게 알리고 $P_{i+1,1}$ 과 나머지 associate head peer $P_{i,j,2}(j=2,\dots,s)$ 에게 데이터를 전송한다. 위에서 설명한 과정을 도식적으로 표현하면 그림 2와 같다. 그림에서 실선화살표는 $P_{i,1}$ 이 탈퇴하기 전의 데이터 전송경로를 나타내며, $P_{i,1}$ 이 탈퇴하게 되면 전송경로가 점선화살표와 같이 바뀌게 된다.

(4) Head peer의 비정상적인 탈퇴

Head peer에 문제가 발생한 경우에도 일반 peer의 경우와 유사한 과정으로 네트워크를 재구성하게 된다. 즉, $P_{i+1,1}$ 은 t_{max} 시간동안 $P_{i,1}$ 로부터 'alive' 메시지의 전송이 없으면 $P_{i-1,1}$ 에게 상황을 알린다. $P_{i-1,1}$ 은 $P_{i+1,1}$ 로부터 보고를 받으면 $P_{i,1,2}$ 에게 데이터를 전송함과 동시에 $P_{i,1,2}$ 에게 head peer가 될 것을 지시하고 i 레벨의 peer 정보를 $P_{i,1,2}$ 에게 전송한다. $P_{i,1,2}$ 는 자신이 head peer가 되었다는 사실을 tracker에게 알리고 $P_{i+1,1}$ 과 나머지 associate head peer $P_{i,j,2}(j=2,\dots,s)$ 에게 데이터를 전송한다. head peer의 탈퇴 과정을 pseudo code로 나타내면 다음과 같다.

```
Algorithm 4 Graceful departure of head peer
send Departure_Message( $P_{i,1}$ ) from  $P_{i,1}$  to  $P_{i-1,1}$ ;
for j=1 to S
    send Departure_Message( $P_{i,1}$ ) from  $P_{i,1}$  to  $P_{i,j,2}$ ;
end for
transmit Data( $P_{i-1,1}$ ) to  $P_{i,1,2}$ ;
Assign_head_peer( $P_{i,1,2}$ );
for j=2 to S
    transmit Data( $P_{i,1,2}$ ) to  $P_{i,j,2}$ ;
end for
transmit Data( $P_{i,1,2}$ ) to  $P_{i+1,1}$ ;
```

```
Algorithm 5 Accidental departure of head peer
if Timer( $P_{i+1,1}$ ) >  $t_{max}$  without 'alive' message then
    send Departure_Message( $P_{i,1}$ ) from  $P_{i+1,1}$  to  $P_{i-1,1}$ ;
    transmit Data( $P_{i-1,1}$ ) to  $P_{i,1,2}$ ;
    Assign_head_peer( $P_{i,1,2}$ );
    for j=2 to S
        transmit Data( $P_{i,1,2}$ ) to  $P_{i,j,2}$ ;
    end for
    transmit Data( $P_{i,1,2}$ ) to  $P_{i+1,1}$ ;
end if
```

이 절에서 살펴본 바와 같이 본 논문에서 제안한 다중 체인구조는 peer의 가입 및 탈퇴로 인한 네트워크

재구성 과정이 간단하므로 peer의 출입이 빈번한 IPTV 환경에 적합한 구조라고 볼 수 있다. 다중 체인구조의 다른 특징은 peer가 탈퇴하는 경우를 처리하는 알고리즘에서 tracker의 제어가 필요없이 peer들 간의 정보와 동작으로 네트워크의 재구성이 가능하다는 점이다. 이처럼 분산 처리방식을 취함으로써 네트워크의 부하를 줄이고 신뢰성을 높일 수 있다.

III. 구조분석

이 장에서는 본 논문에서 제안한 다중 체인구조에 대하여 분석해보고자 한다. 먼저 DSLAM 내의 특정채널에 가입한 peer의 개수와 span, 그리고 레벨의 상관관계에 대하여 분석하고, 레벨과 체인구조에 대한 delay 특성에 대하여 분석한다. 마지막으로 특정 peer에 문제가 생겼을 때 그로 인해 영향을 받는 peer의 개수를 통해 시스템의 신뢰성을 분석한다.

1. peer 개수에 따른 span과 레벨의 상관관계

어떤 peer가 채널에 가입하면 2장에서 설명한 알고리즘에 따라 지정한 레벨의 span에 가입하게 된다. span 개수는 회선의 종류, peer 간의 지리적 거리, 허용 가능한 최대 지연시간 등의 요소를 고려하여 정하게 되며, 따라서 전체 peer의 개수와 span 개수를 기반으로 레벨의 개수를 구할 수 있다. 다중 체인구조의 레벨 개수 L 을 구하기 위해 먼저 S 개의 span과 L 개의 레벨로 구성되는 peer의 개수 N 을 식 (1) 과 같은 부등식으로 정의할 수 있다.

$$\sum_{i=0}^{L-2} (1 + S \times i) < N \leq \sum_{i=0}^{L-1} (1 + S \times i) \quad (1)$$

식 (1)의 부등식을 L 에 대해서 풀어 양의 정수 L 의 값을 구하면 다음과 같다.

$$L = \left\lceil \frac{(S-2) + \sqrt{(S-2)^2 + 8SN}}{2S} \right\rceil \quad (2)$$

여기서 ceiling $\lceil A \rceil$ 는 A 를 넘는 정수들 중 최소 정수를 뜻한다.

그림 3은 N 의 개수에 따른 span 개수와 레벨 개수의 상관관계를 도표로 표시한 것이다.

제안하는 다중 체인구조의 지연시간은 레벨 수에 비례하므로 peer 전체 개수 N 과 span 개수 S 가 정해진 경우에 생기는 지연을 식 (2)를 이용하여 예측해 볼 수

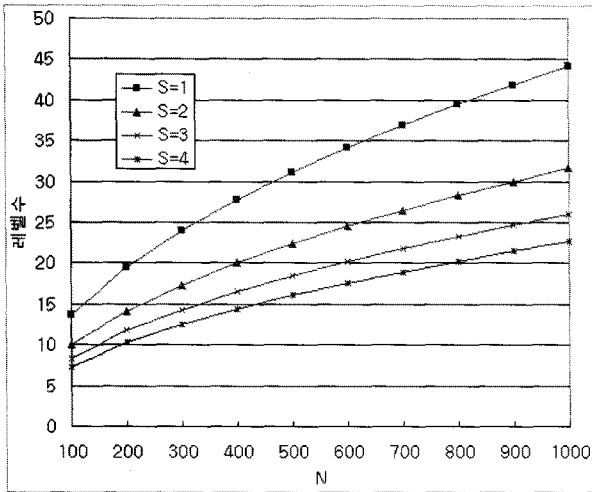


그림 3. peer 개수에 따른 레벨수
Fig. 3. Number of levels over peer size.

있다.

식 (2)를 이용하여 span 개수 S의 값을 구하면 다음과 같다.

$$S \approx \left\lceil \frac{2N}{L^2} \right\rceil \quad (3)$$

식 (2)와 (3)을 이용하면 peer 개수가 정해졌을 때 span 개수에 따른 레벨 수를 예측할 수 있다. 또한 레벨수와 span 개수는 지연과 관련이 있으므로 최대 지연 허용값이 주어지는 경우에 가능한 레벨 및 span 개수를 구할 수 있다.

2. 지연시간(delay time)

IPTV 네트워크를 운영할 때 고려해야 할 요소 중의 하나가 지연시간(delay time)이다. 특히 Live TV 방송인 경우에는 사용자들이 느끼는 체감적인 동시성이 중요하다 하겠다. 따라서 이 절에서는 본 논문에서 제안한 다중 체인구조의 지연에 대하여 분석하고 단일 체인구조의 경우와 비교해 본다. 본 논문에서 언급하는 지연은 DSLAM으로부터 peer까지의 지연을 의미한다. 각 peer에서의 지연은 전송지연(transmission delay), 전파지연(propagation delay) 및 대기지연(queueing delay)로 구분된다. 전파지연은 하나의 데이터 블록이 어떤 peer에서 인접한 peer로 전송되는데 걸리는 시간이며 여기서는 일정한 시간 T_p 로 가정한다. 전송지연은 데이터 블록이 네트워크로 보내지는데 걸리는 시간을 의미하며 peer의 상향링크 속도를 R이라 하고 데이터 블록 크기를 B라고 하면 B/R 이 된다. 대기지연은 peer와 head peer로 나누어서 생각해 볼 수 있다. 우선 레벨

내에서 peer간의 전송에서는 peer의 연결이 체인구조이므로 하향링크 속도와 상향링크 속도가 같다고 가정하면 대기지연은 발생하지 않는다. head peer의 경우에는 데이터 블록을 하나의 회선을 통해서 S+1번 보내야 하므로 대기지연이 발생하게 되며 그 값은 $S \cdot B/R$ 이 된다. 세가지 지연요소를 적용해서 한 채널에 속한 peer들 중에서 최대지연은 $P_{l,1}$ 까지의 지연이며 값을 구해보면 다음과 같다.

$$T_{Dmax-level} = (L-1)T_p + (L-1)(S+1)\frac{B}{R} \quad (4)$$

단일 체인구조에 대하여 최대 지연값은 식 (5)와 같다.

$$T_{Dmax-chain} = (N-1)\left(T_p + \frac{B}{R}\right) \quad (5)$$

지연에 관련된 또 하나의 고려사항은 동일한 채널에 속한 모든 peer들의 지연값을 peer 전체 개수 N으로 나눈 평균값이며 아래 식과 같이 구해진다.

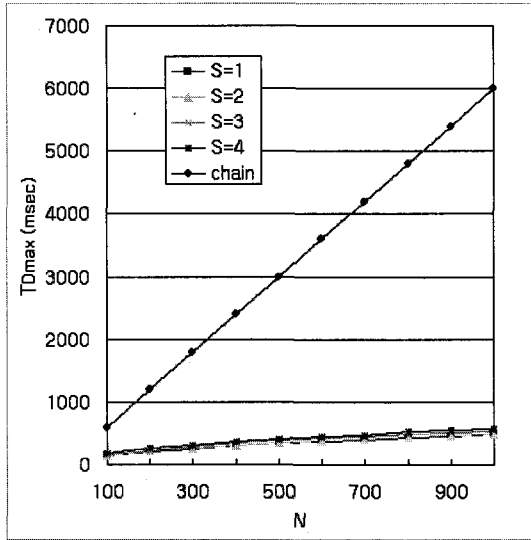
$$T_{Davg-level} = \frac{1}{N}\left(S\sum_{n=1}^{L-1}n^2 + \sum_{n=1}^{L-1}n\right)T_p + \frac{1}{2N}\left(S(S+2)\sum_{n=1}^{L-1}n^2 - (S^2-2S-2)\sum_{n=1}^{L-1}n\right)\frac{B}{R} \quad (6)$$

마찬가지로 체인구조에 대한 지연 평균값은 식 (7)과 같다.

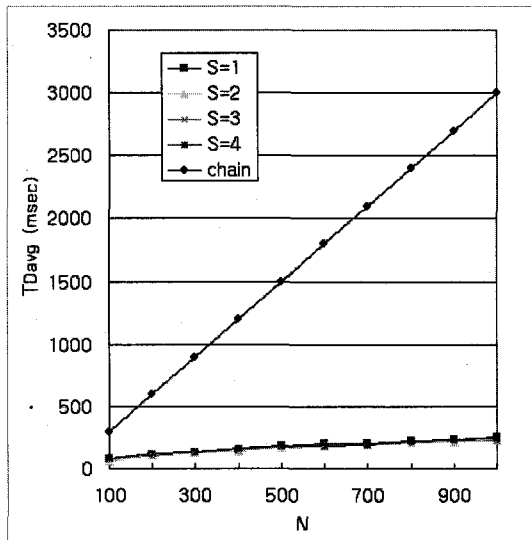
$$T_{Davg-chain} = \frac{N-1}{2}\left(T_p + \frac{B}{R}\right) \quad (7)$$

위의 수식에서 구한 최대 지연과 지연 평균값을 도표로 나타내면 그림 4와 같다. 도표에서 보는 바와 같이 단일 체인구조에 비해 다중 체인구조의 지연이 적다는 것을 알 수 있다. 또한 다중 체인구조에서 span의 개수가 커질수록 최대 지연은 증가하고, 평균 지연은 감소하는 것을 볼 수 있다.

Span의 개수가 늘어남에 따라 대기지연값은 span 개수에 비례하여 늘어나는 반면 레벨수 L 값은 span의 체공근에 반비례하여 줄어들기 때문에 최대 지연은 span 개수가 늘어나면 더 커진다. 하지만 각 레벨의 체인에 연결된 peer는 대기지연이 없으므로 span 개수가 늘어나더라도 지연이 더 커지지 않는다. 따라서 지연 평균값은 span이 늘어남에 따라 감소하게 된다.



(a) 최대 delay
(a) Maximum delay



(b) delay 평균
(b) Average of delay

그림 4. peer 개수에 따른 최대 delay와 delay 평균 (Tp=1msec, block size=50kbits, uplink bandwidth=10Mbps)

Fig. 4. Maximum and average delay time over peer size (Tp=1msec, block size=50kbits, uplink bandwidth=10Mbps).

3. 신뢰도

본 논문에서는 시스템의 신뢰도를 판단하는 요소로 어떤 peer에게 문제가 생겨서 불가피하게 지연이 발생하는 경우에 영향을 받는 peer의 개수를 고려한다. 즉, 모든 peer에게 문제가 생길 확률이 동일하다고 가정할 때 어떤 peer의 문제로 인해서 그 이후에 연결되어 있는 peer들은 서비스의 저하를 가져올 수 있다. 따라서 전체 시스템의 신뢰도가 좋으려면 어떤 peer로 인해 영

향을 받는 peer의 개수가 적어야 한다.

그림 1에서 보는 것과 같이 다중 체인구조에서 만일 레벨 1의 head peer에게 문제가 생기면 N개의 모든 peer가 영향을 받는다. head peer가 아닌 peer에게 문제가 생기면 영향을 받는 peer의 개수는 해당 peer 뒤에 연결되어 있는 peer들로 줄어든다. 이와 같이 모든 peer들에 대해서 자신에게 문제가 발생한 경우에 영향을 받는 peer의 개수를 구하고 이를 전체 peer 개수 N으로 나눈 평균값을 구하면 아래 식과 같다.

$$C_{Pavg-level} = \frac{1}{6N}L(L-1)[(2L-1) \times S + 3] \quad (8)$$

같은 방법으로 순수한 체인구조에 대하여 peer의 문제발생으로 인해 영향을 받는 peer의 개수 평균은 다음과 같다.

$$C_{Pavg-chain} = \frac{N-1}{2} \quad (9)$$

그림 5는 peer에 문제가 발생한 경우에 영향을 받는 peer의 평균 개수를 보여준다. 그림에서 보는 바와 같이 peer의 문제로 인해 발생하는 지연에 대한 신뢰도가 단일 체인구조에 비해서 다중 체인구조가 높다는 것을 알 수 있다. 또한 다중 체인구조에서 span의 개수가 커질수록 평균값이 커지는 것을 볼 수 있다. 그 이유는 span 개수가 커질수록 head peer에 연결되는 peer의 개수가 증가하므로 head peer에 문제가 발생할 경우에 영향을 받는 peer의 개수도 증가하게 된다. 따라서 peer에

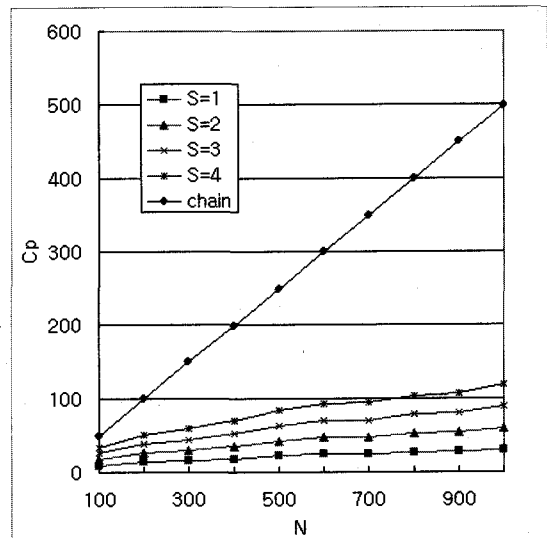


그림 5. peer에 발생한 문제로 인해 영향 받는 peer 개수의 평균값

Fig. 5. Average number of peers affected by accidental departure of peer.

문제가 발생할 확률이 동일하다고 한다면 span 개수가 커짐에 따라서 어떤 peer 때문에 영향을 받는 peer의 개수도 증가하게 된다. 따라서 span의 개수를 정할 때 지연시간과 신뢰도를 함께 고려해야 할 것이다.

IV. 결 론

본 논문에서는 ISP에 의해 관리되는 IPTV 환경에서 생방송 TV 전송을 위한 다중 체인구조의 P2P 시스템을 제안하였다. 제안하는 다중 체인구조는 peer의 빈번한 가입과 탈퇴로 인해 발생하는 네트워크의 재구성 과정이 단순할 뿐만 아니라 peer가 탈퇴하는 경우에는 tracker의 제어가 필요 없이 peer들의 정보만으로 재구성이 가능하다는 장점이 있다. 또한 다중 체인구조는 단일 체인구조에 비해서 향상된 지연 성능과 높은 신뢰도를 보여준다는 것을 수학적 분석을 통해 증명하였다. 향후에 IPTV 망에서 VOD 서비스에 다중 체인구조를 적용하는 방법과 사용자가 채널 선택을 위해 채널에 머무르는 시간이 짧은 경우에 효율적으로 재구성하는 방법에 대한 연구가 필요하다.

참 고 문 헌

- [1] Y. Chu, S. G. Rao, and H. Zhang, "A case for end system multicast," in ACM SIGMETRICS, pp. 1-12, Santa Clara, USA, June 2000.
- [2] N. Magharei, R. Rejaie, and Y. Guo, "Mesh or multiple-tree: A comparative study of live P2P streaming approaches," IEEE INFOCOM'07, pp. 101-109, Anchorage, USA, May 2007.
- [3] X. Hei, C. Liang, J. Liang, Y. Liu, and K. Ross, "A Measurement Study of a Large-Scale P2P IPTV System," IEEE Trans. on Multimedia, pp. 1672 - 1687, December 2007.
- [4] M. Zhang, J. Luo, L. Zhao, and S. Yang, "A peer-to-peer network for live media streaming using a push-pull approach," in Proc. ACM Multimedia(MM'05), pp. 287-290, Hilton, Singapore, Nov. 2005.
- [5] X. Zhang, J. Liu, B. Li, and T. S. Yum, "CoolStreaming/DONet: A data-driven Overlay Network for Live Media Streaming," IEEE INFOCOM'05, pp. 2002-2011, Miami, USA, March 2005.
- [6] PPLive. <http://www.pplive.com/>.
- [7] PPStream. <http://www.ppstream.com/>.
- [8] UUsee. <http://www.uusee.com/>.
- [9] J. Noh, A. Mavlankar, P. Baccichet, and B. Girod, "Reducing End-to-End Transmission Delay in P2P streaming systems Using Multiple Trees with Moderate Outdegree," In Proc. of International Conference on Multimedia and Expo, ICME 2008, pp. 1-4, Hannover, Germany, June 2008.
- [10] K. Park, S. Pack, and T. Kwon, "Climber: An Incentive-based Resilient Peer-to-Peer System for Live Streaming Services," In Proc. of International Workshop on Peer-to-Peer Systems(IPTPS '08), pp. 69-74, Tampa, USA, Feb. 2008.
- [11] D. A. Tran, K. A. Hua, and T. T. Do, "A peer-to-peer architecture for media streaming," IEEE J. Selected Areas in Communications, Vol. 22, Issue 1, pp. 121-133, Jan. 2004.
- [12] M. Heffeeda, A. Habib, B. Botev, D. Xu, and B. Bhargava, "PROMISE: peer-to-peer media streaming using CollectCast," in Proc. ACM Multimedia(MM'03), pp. 45-54, Berkeley, USA, Nov. 2003.
- [13] M. Cha, P. Rodriguez, S. Moon, and J. Crowcroft, "On Next-Generation Telco-Managed P2P TV Architectures," In Proc. of International Workshop on Peer-to-Peer Systems(IPTPS '08), pp. 45-50, Tampa, USA, Feb. 2008.

저 자 소 개



김 지 훈(정회원)

1987년 부산대학교 전기공학과
학사 졸업.

1990년 한국과학기술원 전기 및
전자공학과 석사 졸업.

1990년~1993년 한국통신
연구개발단.

1993년~1995년 한국휴렛팩커드 컴퓨터개발부.

1995년~현재 부천대학 e-비즈니스과 부교수.

<주관심분야 : P2P, IPTV, 멀티미디어스트리밍>



김 영 한(정회원)

1984년 서울대학교 전자공학
학사.

1986년 한국과학기술원 전기전자
공학 석사.

1990년 한국과학기술원 전기전자
공학 박사.

1990년~1994년 디지콤정보통신연구소 연구부장

2006년~현재 숭실대 정보통신공학과 교수

2006년~현재 통신학회 인터넷연구회위원장

2006년~현재 VoIP포럼 차세대분과 위원장

<주관심분야 : BcN, IMS, VoIP, QoS, MANET>