

오버레이 멀티캐스트 환경에서 효율적인 QOS의 보장을 위한 노드 관리 스케줄링

최성욱^{1*}

¹시립 인천 전문대학 컴퓨터 정보과

A Node Management Scheduling for Efficient Guarantee of QOS based on Overlay Multicast Environment

Sung-uk Choi^{1*}

¹Dept. of Computer and Information Science, Incheon City College

요 약 최근 IP 멀티캐스팅 대신에 각광을 받고 있는 오버레이 멀티캐스팅 기법은 네트워크 응용계층의 서비스로써, 서버의 자원 활용과 네트워크의 대역폭에 대한 비용 면에서 유리하다. 본 논문에서는 효율적인 멀티미디어 서비스를 위하여 오버레이 멀티캐스팅 환경에서의 P2P 조인 스케줄링 정책을 제안한다. 본 논문의 성능 평가를 위하여 본 논문과 유사성을 가진 기법과 비교하여 멀티캐스트에 참여한 노드들의 평균노드 접속 수, 최대 대기 보장 시간의 위배 건수, 그리고 버퍼의 결함 확률을 실험을 하였다. 그 결과 약 15 ~ 20%의 성능의 향상을 보여 주었다.

Abstract The overlay multicast that has been recently presented as an alternative for the IP multicast has been getting much persuasion by the system resource and the network bandwidth and the advancement of the network cost. we propose a overlay multi_casting network policy for efficient multimedia services with P2P join scheduling. In order to evaluate the performance of proposed scheme, We test the performance of multi_casting nodes with average join count, the maximum guaranteed waiting time and buffer starvation ratio with similar techniques. The result of simulation shows improved about 15~20% of performance.

Key Words : P2P, Overlay Multicast, IP Multicast, QOS, Buffer Starvation, Session Manager

1. 서론

오버레이 멀티캐스트[1-8]는 중단 호스트에게 멀티캐스트 패킷 전달을 위해 응용 계층에서 멀티캐스트를 지원하는 기술이다. 이 기술은 현재 네트워크 기반구조의 변경이나 증설 없이 즉시로 보급을 촉진시킬 수 있다는 장점이 있다. 그림1은 오버레이 네트워크의 특징을 설명하기 위하여 두 개의 계층 즉 물리적 구조와 논리적 구조로 표현하여 보았다. 아래쪽 계층은 물리적 구조이며 트랜스포트 네트워크, 모바일 네트워크, 무선 네트워크 등 실제의 다양한 네트워크를 표현한 것이다. 그리고 위의 계층은 오버레이 네트워크로서 물리적 네트워크에 기반을 둔 논리적 네트워크이다. 세션 매니저(Session

Manager : SM)는 클라이언트 노드로부터 받은 정보를 이용하여 오버레이 토폴로지를 구성하고 멀티캐스트 에이전시(Multicast Agency : MA)들을 관리한다. 멀티캐스트 에이전시(MA)는 가입자들에게 데이터 전송을 위한 최적 경로를 설정 및 저장된 스트림을 전달한다.

VOD 서비스에서 가장 민감한 문제는 서비스 지연과 사용자 QOS의 보장이라 할 수가 있다. 서버 및 프락시의 자원 고갈에 의한 서비스 지연을 극복하기 위하여 제안된 P2P 네트워킹은 클라이언트들 간에 서로의 스트림을 공유하게 함으로써 일부의 문제를 극복하고 있지만, 반면에 서비스 중인 노드들의 떠남(departure)과 참여(join)가 빈번히 발생함으로 인한 사용자 QOS의 보장에 대한 문제가 새로운 이슈가 되고 있다[9,10]. 본 논문에서는 오버

본 논문은 2008년도 시립인천전문대학 연구비 지원으로 수행되었음.

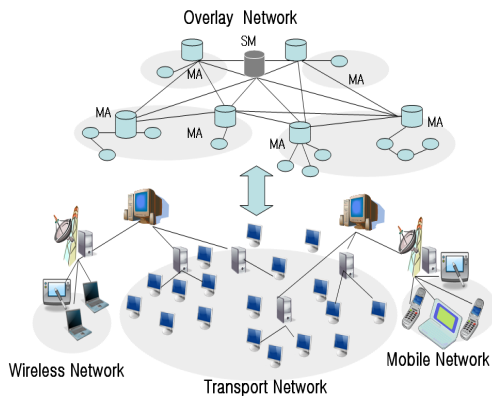
*교신저자 : 최성욱(csunguk@gmail.com)

접수일 09년 05월 10일

수정일 09년 6월 15일

재확정일 09년 07월 22일

레이 P2P 멀티캐스팅 환경에서 VOD 서비스에 참여 하는 클라이언트 노드들의 대기시간을 단축하고 부모노드들의 떠남과 참여에 의해서 하부 노드들에게서 발생할 수 있는 QOS의 저하 문제를 최소화하기 위한 조인 스케줄링 방안을 제안한다. 본 논문의 2장에서는 VOD 멀티캐스팅 서비스에 관련된 연구를 살펴보고 3장에서는 대기시간 단축 및 사용자 QOS 보장을 위한 효율적인 노드 관리 스케줄링을 제안한다. 4장에서는 본 연구에 대한 성능평가를 위해서 실험을 통하여 기존의 오버레이 멀티캐스트 기법과 비교분석을 한다. 끝으로 5장에서는 결론과 함께 향후 연구 방향에 대하여 간략히 기술한다.



[그림 1] 오버레이 네트워크 구조

2. 관련 연구

오버레이 멀티캐스트는 동적 노드 기반 오버레이 멀티캐스트(Dynamic node based overlay Multicast)[3,4]와 정적 노드 기반 오버레이 멀티캐스트(Static node based overlay Multicast)[1,2,6,7,8]로 나눌 수 있다. 동적 노드 기반 오버레이 멀티캐스트는 그룹 멤버들을 자율 구성(Self-Organization)하여 오버레이 멀티캐스트 트리로 구성하는 방식이다. 정적 노드 기반 오버레이 멀티캐스트는 전체 네트워크 주변의 중요한 위치에 오버레이 프락시라는 고정노드를 설치하여 사용자 요구가 발생하면 멀티캐스트 서비스하기 위하여 고정노드가 오버레이 트리를 형성하는 방법이다. 이 방식은 고정 노드로 인하여 안정적이고 향상된 서비스를 쉽게 제공할 수 있지만 노드 조인시의 네트워크의 불균형 문제와 인기비디오 등 서비스 요구의 폭주에 대한 유연성에 문제가 있다. 이 중에서 Overcast[6,7]는 단일 소스를 위한 작은 크기에서 중간 크기의 노드까지의 풀(pool)에 형성된 최적화 오버레이 트

리를 이용하여 멀티캐스트를 구현하였다. 그러나 "Up/Down" 프로토콜을 수행하기위한 오버헤드의 문제와 대역폭 우선의 조인정책으로, 많은 노드들이 참여할 경우 네트워크 토폴로지의 부담을 줄 수 있다는 문제가 있다. 복합형 오버레이[2]는 대역폭 우선 조인 정책을 개선한 리프 노드 우선의 조인 정책과 라이브 스트림 및 VOD 스트림을 위한 복합 오버레이 구조를 제안하여, 조인시간 및 네트워크 토폴로지의 부담을 개선하였다. 그러나 이러한 연구는 P2P에서 잦은 노드의 떠남과 참여에 의한 네트워크 토폴로지의 변경 시에 사용자 QOS 보장을 위한 방안이 미흡하였다. 본 논문에서는 이를 위하여 최대 서비스 대기시간을 보장하고 부모 노드의 떠남이 발생하였을 때, 하위 자식노드가 새로운 부모 노드를 찾을 때까지의 시간 동안에 발생할 수 있는 버퍼의 기아(Buffer Starvation) 즉 지터현상을 최소화하여 QOS의 저하를 방지하기위한 노드 관리 스케줄링을 제안한다.

3. P2P 오버레이 멀티캐스트 모델

3.1 개요

본 논문에서 제안한 오버레이 멀티캐스트 모델은 P2P 스케줄링을 기반으로 하고 있다. 그림2에서 보는 바와 같이 세션 매니저(SM)의 역할은 각 세션의 네트워크 토폴로지를 관리한다. 또한 가입자의 요청이 오면 멀티캐스트 에이전시에서 보내온 네트워크 정보를 참고하여 적절한 세션을 스케줄링 한 후, 멀티캐스트 에이전시에 해당 서비스를 요청한다. 그 밖에 서비스를 요청한 가입자의 확인, 과금, 서비스 등급 결정, QOS 품질 등 가입자 관리를 수행한다. 멀티캐스트 에이전시(MA)는 오버레이 프락시의 기능을 수행하며 각 세션내의 가입자에 대한 자원의 수락 및 제어 기능을 담당한다. 이를 위하여 자체의 스토리지 기능을 가지고 있으며, 다른 멀티캐스트 에이전시 간의 커뮤니케이션과 하위 클라이언트 노드와의 연결 스케줄링을 수행한다. 클라이언트 노드(CN)는 오버레이 네트워크를 구성하는 하위 객체이며 P2P용 스트림 블록을 저장할 버퍼를 가지고 있다. 만일 클라이언트 노드 CN_i 이 서비스 참여를 위해서 네트워크 내 임의의 노드에 조인을 하게 되면, CN_i 의 버퍼에는 현재의 세션 S_i 의 루트 노드(검정색 클라이언트 노드)가 재생 중인 비디오 스트림 블록 V_i 와 루트 노드의 버퍼에 저장된 스트림 V_1 에서 V_{i-1} 까지를 CN_i 의 버퍼 B_n 에 식1과 같이 기억시킨다. 또한 스트림 블록의 사이즈를 β , 최대 버퍼 사이

즈를 B_{max} 라 하면 클라이언트 버퍼는 식2와 같이 정의할 수가 있다. 클라이언트 노드끼리의 연결에 있어서 부모노드의 떠남은 전송 경로에 대한 결합의 발생으로 이어져서 스트림 데이터들이 모든 자식들에게 전달할 수 없는 상황이 발생하게 된다.

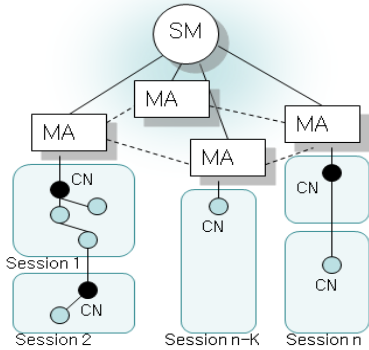
$$B_j \leftarrow V_j : \text{Current Stream}$$

$$\text{and } \sum_{n=1}^{n=j-1} (B_n \leftarrow V_n) : \text{Buffered Stream}$$

(1)

$$1 \leq \sum_{n=1}^{n=j} (V_n \cdot \beta) \leq B_{max}$$

(2)



[그림 2] 제안된 오버레이 구조

이러한 스트리밍 데이터의 손실은 부모 잃은 자식노드들이 자신과 연결된 서브 트리 전체에 영향을 주게 된다. 따라서 스스로부터 주어진 노드까지의 총 지터는 노드가 위치해 있는 트리의 깊이에 따라 선형적으로 증가된다. 이러한 문제를 거대한 버퍼 용량으로 접근한다면 네트워크자원의 낭비를 초래할 것이고, 데이터 전송을 위한 트리가 깊은 길이를 가졌다면 많은 양의 스트리밍 데이터에 대한 손실을 가지고 올 것이다. 피어 조인 스케줄링은 이러한 유동적인 노드들의 문제를 최소화하기 위하여 제안한다.

3.2 피어 조인 스케줄링

피어 조인 스케줄링(Peer Join Scheduling(PJS))은 서비스 참여를 희망하는 클라이언트 노드가 멀티캐스트 에이전시에서 직접 서비스를 받지 않고 클라이언트 노드들끼리 조인하는 방식이며, 세션 내에 있는 노드들의 버퍼에 해당 서비스 스트림이 존재할 경우에 실행한다. PJS

방식은 VOD 요구가 특정 시간 내에 서비스가 폭주하거나 인기 비디오를 서비스할 경우에 유리하다. PJS방식은 조인 시간을 단축하고 노드의 추가 시에 네트워크를 균형적으로 관리하기 때문에 어떤 위치에서 노드의 이탈이나 참여가 발생하더라도 변경되는 노드 수를 안정적으로 관리한다. 즉 동일한 차수에 속해있는 형제 노드가 N 개가 있고, 형제노드 N_i 를 루트로 하는 서브 트리의 노드수를 NT_i 라 할 때, 각 서브노드의 총 평균을 식3으로 하면 식4의 값을 최소화하는 정책이 필요하다.

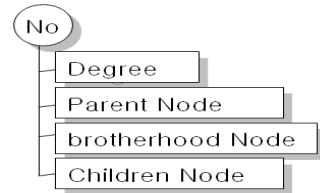
$$NT_{iAVE} = (\sum_{i=1}^n NT_i) / n$$

(3)

$$\sum_{i=1}^n (NT_i - NT_{iAVE})^2 / (N-1)$$

(4)

각 노드의 정보는 그림3과 같다. 모든 클라이언트 노드는 자신의 번호와 멀티캐스팅 에이전시와의 차수, 부모노드와 형제노드, 자식노드의 정보를 포함한다. 이러한 노드들은 참여를 신속하게하고 균형 있는 트리 네트워크를 구축하는데 도움이 된다. 어떤 시간대에 버퍼에 저장되어 있는 블록들 중에서 동일한 시작점을 버퍼에 가지고 있는 클라이언트들을 하나의 세션 S_i 으로 그룹화한다.



[그림 3] 노드 정보

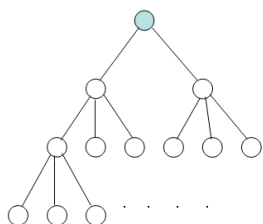
클라이언트 버퍼에 캐싱된 스트림의 크기가 B_{max} 를 초과할 경우에는 MA에 의해서 또 하나의 세션인 S_{i+1} 를 생성한다. 그리고 S_i 의 클라이언트들은 자신의 버퍼에 캐싱된 블록들을 다음 S_{i+1} 의 멤버 클라이언트에게 체인을 이용하여 스트림 서비스를 제공한다. PJS 처리단계는 다음과 같다.

단계 1 : 해당 세션 S_i 에서 루트 노드를 N_i 라 할 때, 참여하려는 노드 C 는 차수($degree_{n_i \rightarrow C}$)가 가장 짧고, 가장 넓은 대역폭($rate_{n_i \rightarrow C}$)을 가지며 형제노드 ($bro_{with n_j}$)가 적은 리프 노드 L_i 를 식 5와 같이 검색한다.

$$\max_{L_i \in S} \left(\frac{\text{rate}_{n_i \rightarrow C}}{\text{degree}_{n_i \rightarrow C}} \right) \cdot \text{AND} \cdot \min_{L_i \in S} (\text{bro with } n_j) \quad (5)$$

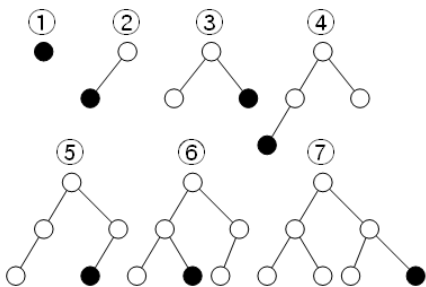
단계 2 : 검색된 노드 L_i 가 있으면 단계 3으로 가고 검색된 노드가 없으면 해당 스트림을 공유할 수 있는 또 다른 세션이 나타날 때까지 PJS 대기 시간 내에서 대기한다. PJS 대기 시간이 지나면 현재의 서비스 요구는 멀티캐스트 에이전시에 의해서 직접 조인 스케줄링 된다.

단계 3 : 검색된 노드 L_i 의 부모 노드 P_i 를 검사한다. P_i 노드가 서비스 할 수 있는 대역폭을 소유하고 있으면 L_i 의 형제노드로 참여를 허용하고, 그렇지 않을 경우는 L_i 의 자식노드로 참여시킨다.



[그림 4] 예제 토폴로지

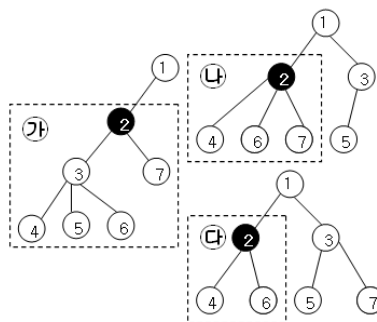
그림4의 토폴로지는 최대 가능 차수(degree)가 4이고 Root 노드는 최대2개의 자식노드를, 나머지 노드는 각각 최대 3개의 자식 노드에 서비스할 수 있는 대역폭을 가지고 있다. 그림5는 그림4의 토폴로지를 이용하여 7개의 노드를 위한 PJS 순서를 예를 나타낸다. 그림6의 ㉔㉕는 지금까지 알려진 피어 투 피어의 조인 방식이다. ㉔는 BFS(Breadth First Search)[9,10] ㉕는 DFS(Depth First Search)[2] 결과와 유사하다할 수가 있다.



[그림 5] PJS 처리 순서

㉔는 PJS방식을 나타낸다. 만일 ㉔번 노드가 네트워크에서 떠남이 발생한 경우에 ㉔의 방식은 2차수에 걸쳐 5개의 클라이언트가 영향을 받고, ㉕는 1차수 3개의 클라이언트가 영향을 받는다. 결국 한 노드가 떠남으로 영향을 받는 자식의 노드가 많아질수록 기아 현상 기아의 확률이 높아질 수가 있으며 결국은 클라이언트의 QOS에 영향을 준다. PJS의 처리시간은 식6과 같이 정의할 수가 있다.

$$PJS_{\max} = S_{sm} + S_{ns} + N \times N_T \quad (6)$$



[그림 6] 조인 스케줄링 비교

여기서 S_{sm} 은 SM의 스케줄링 시간, S_{ns} 는 네트워크 스위칭 시간, D_{sch} 는 스토리지 검색시간, N 은 경로 내 차수, N_T 는 노드와 노드간의 네트워크 전송시간이다.

3.3 서버 조인 스케줄링

서버 조인 스케줄링 (Server Join Scheduling(SJS))은 클라이언트 노드들끼리의 조인이 아닌 서버의 역할을 하는 멀티캐스트 에이전시(MA)에 직접 조인하는 방식이다. 서비스를 원하는 클라이언트 노드는 우선 PJS 큐에 대기하여 서비스 받을 세션을 기다리게 되는데, 만일 미리 정해진 PJS 대기시간을 초과할 경우에는 오버레이 프락시 기능을 하는 MA로 직접 조인하게 된다. SJS는 MA의 자원을 효율적으로 운영하기 위해서 조인을 위한 시간 간격을 포함한다. 또한 서비스 요구에 따라 조인의 간격에 차이를 둘 필요가 있다. 이를 위하여 SJS에서는 조인의 시간 간격을 결정하기 위하여 인기도 분포인 Zipf 분포를 활용하는 방안을 제안한다. I를 전체 비디오 수라 하면, 보통 비디오의 인기도를 나타내는 Zipf 분포 함수 $f(i)$ 는 식7과 같다. 식8은 인기도에 대한 접근 확률의 편중도를 높이기 위하여 식7을 약간 변형한 형태이며, θ 값은 0.271을 적용하는데, 비디오에 대한 사용자의 요청률을 나타낼

때 일반적으로 많이 사용된다[11].

$$f(i) = C / i \text{ here } C = 1/\sum (1/i) < \text{for } i = 1 \text{ to } I > \quad (7)$$

$$f(i) = C / i (1-\theta) \text{ where } C = 1/\sum (1/i (1-\theta)) < \text{for } i = 1 \text{ to } I > \quad (8)$$

이를 이용하여 SJS 대기큐의 j번째 비디오에 대한 조인 간격 T_j 는 식9로 정리될 수 있는데 여기서 T_{max} 은 SJS 조인의 최대 대기 값이며, $\sum f(n)$ 은 조인을 위한 모든 비디오의 도착률의 합이 된다.

$$T_j = T_{max} \cdot ((\sum f(n) - f(j))/\sum f(n)) \quad (9)$$

SJS의 처리 시간은 식10과 같이 나타낼 수가 있다.

$$SJS_{max} = T_j + S_{ns} + D_{sch} + N_T \quad (10)$$

여기서 T_j 은 SJS의 조인 대기 시간, S_{ns} 는 네트워크 스위칭 시간, D_{sch} 는 스토리지 검색시간, N_T 는 MA와 노드간의 네트워크 전송시간이다. 그러므로 서비스 대기 보장시간은 식11과 같다.

$$W_{max} \geq PJS_{max} + SJS_{max} \quad (11)$$

4. 시뮬레이션 및 분석

본 논문의 성능을 평가하기 위하여 프로세스 기반의 이산 사건 중심 시뮬레이션 커널(Process-based Discrete Event Oriented Kernel)인 SSFNet[12]시뮬레이션 프레임워크를 사용한다. 서버에 저장된 비디오는 100분 길이의 하나의 비디오를 사용하는 것을 가정한다. 그리고 클라이언트의 도착율은 인기도를 감안한 Zipf 분포에 따르며, 최대 서비스 보장 대기시간은 120sec로 하고, 멀티 캐스트 트리가 형성된 이후에 2분마다 클라이언트의 떠남 장애가 발생하도록 한다. 본 논문에서 제안된 스케줄링의 타당성을 실험해보기위해서 논문과 유사한 오버레이 멀티 캐스트구조를 사용하고 있는 논문[2,6]의 방식과 비교한다. 각 스케줄링의 개요를 표1에 표시한다. 우선 오버캐스트[6]는 IP_멀티캐스트를 기본으로 하고 네트워크 대역폭이 부족할 때에만 간헐적으로 오버레이 멀티캐스트를 활용하는 방식이다.

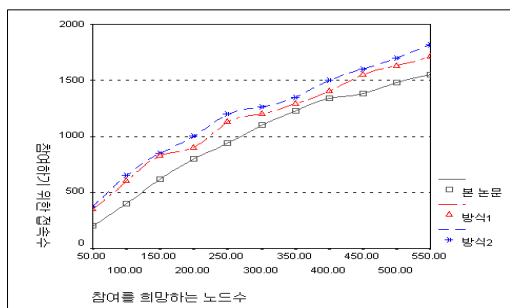
[표 1] 유사기법 비교

기법	오버캐스트[6]	복합오버레이[2]	본 논문
IP*	√	√	
OVER*	√	√	√
Main*	IP*	VOD: IP* Live: OVER*	OVER*
노드 참여방식	대역폭	대역폭 리프노드우선	대역폭 균형
경로탐색 알고리즘	대역폭 탐색	깊이 우선 탐색	균형적 탐색
대기시간 보장			√
IP*: IP_Multicast OVER* : Overlay_Multicast Main*: 기본이 되는 스케줄링			

또한 노드의 참여를 위해서 대역폭만을 감안하기 때문에 P2P 환경에서 효율적인 네트워크의 활용이 어렵다. 복합 오버레이 멀티캐스트기법[2]는 VOD 서비스일 경우에는 IP_멀티캐스트, 실시간 Live 서비스일 경우에는 오버레이 멀티캐스트를 활용한다는 특징을 가지고 있다. 노드의 참여는 가장 넓은 대역폭과 가장 적은 레벨 수의 비가 최대값을 가진 노드에 접속을 시도하기 때문에 넓고 짧은 멀티캐스트 트리를 구축 할 수 있다. 그러나 이 방식은 DFS(Depth First Search) 알고리즘을 사용하기 때문에 노드사이의 경로와 하위노드들의 접속 횟수가 일정하지 못하고 부모노드의 떠남에 의한 버퍼 관리에 불리한 점이 있다. 이후 각각의 방식을 편의상 방식1 및 방식2로 표기한다.

가. 참여를 하기위한 평균노드 접속수

참여 노드들의 증가에 따라 참여를 위해 부모노드들의 평균 접속 횟수를 측정한 결과를 그림7에 나타내었다. 세 논문 모두 참여를 희망하는 노드 수의 증가에 따라 부모노드를 찾기 위한 접속회수가 점진적으로 증가하는 양상을 보여준다. 다만, 본 논문의 부모 노드 탐색 알고리즘은, 방식1과 방식2의 비교에서는 각각 약 7 ~ 11% 정도와 20%정도의 적은 접속 횟수로 부모노드를 선택하는 결과의 우수성을 보인다. 이러한 적은 접속 횟수는 노드들에게 서비스를 보다 빠르게 제공할 뿐만 아니라 전체 네트워크 트래픽에 많은 영향을 미치게 된다.

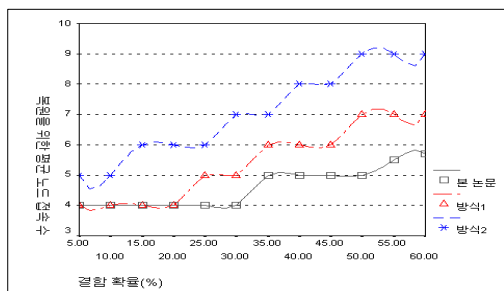


[그림 7] 참여를 하기위한 평균노드 접속수

반면에 방식2는 많은 노드들의 참여가 발생하면 제일 큰 대역폭을 찾기 위해 네트워크의 경로를 재탐색하는 프로세스로 인하여 지연이 많이 발생하는 것으로 나타난다.

나. 결함확률에 따른 복원지연

그림8은 결함 발생률에 따라 복원을 위해 부모 노드를 찾기 위해 접속한 노드 수를 나타낸다. 본 논문에서 제안된 방식과 방식1은 복원을 위한 노드 접속 수에서 보면 결함확률이 20% 이하일 경우에는 유사한 성능을 보이나 20% 이상일 경우는 본 논문의 우수성이 나타난다. 이것은 노드들의 떠남이 발생하여도 균형 트리를 구축하여 새로운 부모들을 찾는 자식노드의 수를 상대적으로 줄인 결과라 판단된다. 그리고 방식2는 전송 트리에 따라 부모 노드가 큰 대역폭을 찾기 위한 알고리즘의 영향으로 인하여 평균노드수가 상대적으로 높게 나타난 것으로 판단된다.

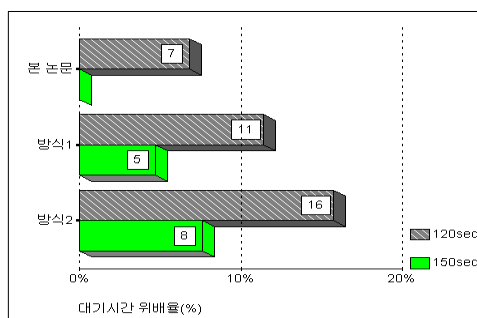


[그림 8] 결함확률에 따른 복원지연

다. 최대 서비스 대기 시간 위배

서비스 지연에서 오는 사용자의 서비스 요청 취소와 밀접한 연관이 있는 최대 서비스 대기 보장 시간의 위배율을 분석해보았다. 위배율은 현재 네트워크상에 있는 모든 노드와 서비스 대기 시간 내에 참여하지 못한 노드들

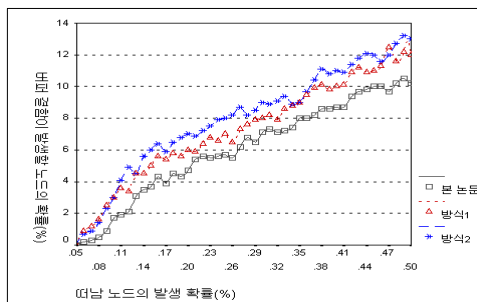
의 비율이다. 보장 시간을 120sec으로 하였을 경우 위배율은 본 논문(6.8%)이 방식1(11.4%), 방식2(15.7%) 비해 35%정도 단축하고 있으나, 모두 대기 시간을 위배하는 것으로 나타나고 있다. 그러나 시뮬레이션 된 결과를 150sec까지 연장해 보면 본 논문에서 제안한 방식의 위배율이 Zero로 되었음을 알 수 있다. 이는 제안한 방식이 최대 보장 시간(120sec)을 초과한다 하더라도, ± 20% 내에서 모두 서비스됨을 의미한다. 이러한 편차가 생기는 이유는 최대 서비스 대기 보장시간이 주어진다하더라도 식9와 같이 조인 하기위한 큐의 대기시간이 가변으로 계산됨으로 생기는 현상이라 할 수가 있다. 참고로 그림9에 나타난 각 수치는 반올림되었다.



[그림 9] 최대 대기시간 위배율

라. 버퍼 기아 현상 측정

그림10의 실험은 버퍼의 기아 즉 지터현상을 조사하는 실험으로 스트리밍 서비스의 QOS의 저하 문제를 평가하는 척도가 될 수 있다. 이 실험은 노드들의 떠남이 점차로 증가하였을 때 버퍼 기아가 발생한 노드들의 수를 비율(%)로 표시하였다. 제안한 방식이 방식1, 방식2에 비해서 유리하게 나타난 것은 균형적인 트리구축을 행하는 본 논문의 방식이 부모노드의 떠남이 발생하였을 때, 고아 노드들의 복원 시간이 상대적으로 짧다는 점이 작용하였다고 생각되어진다.



[그림 10] 기아 현상 기아 측정 실험

5. 결론

본 논문에서는 오버레이 P2P 멀티캐스팅 환경에서 VOD 서비스의 참여 하는 클라이언트 노드들의 참여 지연 시간을 단축하고 최대 서비스 대기 시간을 보장하며, 빈번한 떠남과 참여에 의해서 하부 노드들에게서 발생할 수 있는 QoS의 저하 문제를 최소화하기 위한 네트워크 노드 관리 정책을 제안하였다. 결과적으로 제안기법이 비교 연구[2,6]에 비해서 참여 지연에서 15%정도의 개선된 수치를 보여 주었으며, 임의의 노드가 떠남에서 발생하는 하위노드간의 복원을 위한 접속 수에서는 약 20% 정도 개선된 결과를 보여 주었다. 그리고 최대 서비스 대기 시간 보장율은 보장 대기시간의 20%내에서 모두 서비스됨을 보여주었는데, 이 편차는 동적으로 산출되는 조인 대기 시간의 차이라 할 수가 있다. 또한 사용자 QoS에 영향을 주는 버퍼 기아의 확률은 비교한 방식에 비해서 약 18% 정도 확률의 감소를 보여주었다. 이 연구는 IPTV 서비스 뿐 만아니라, 차세대 저장 스트림 서버 네트워크에도 활용할 수 있을 것이다. 현재 진행 중인 연구는 제안된 방식들을 더욱 다듬어 이동통신과 무선통신이 결합된 오버레이 멀티캐스팅 환경을 구현하는 것을 목표로 하고 있다.

참고 문헌

- [1] 고주영, 심재창, 김현기 “오버레이 멀티캐스팅에서의 트래픽 에이전트에 관한 고찰” 정보처리학회논문지B, 제12권, 제6호, PP 685~ 690, 10월, 2005.
- [2] 김종경, 최성욱 “P2P 환경에서 복합 스트림 서비스를 위한 효율적인 오버레이 멀티캐스팅 스케줄링” 컴퓨터 정보학회, 제13권, 제6호, pp 233~241, 11월, 2008.
- [3] Y. Chu, S. Rao and H. Zhang, "A case for end system multicast", in Proceedings of ACM SIGMETRICS, Santa Clara, CA, pp. 1~12, June 2000.
- [4] V. N. Padmanabhan, H. J. Wang, P. A. Chou and K. Sripandikulchai, "Distributing streaming media content using cooperative networking", in ACM/IEEE NOSSDAV, Miami, FL, USA, ay 2002.
- [5] De Vleeschauwer, B.; De Turck, F.; Dhoedt, B.; Demeester, P. "Online Management of QoS Enabled Overlay Multicast Services" Global Telecommunications Conference, IEEE Nov.27, 2006
- [6] J. Jannotti, D. K. Gifford, K. L. Johnson, M. F. Kaashoek and J. W. O'Toole, "Overcast: Reliable

Multicasting with an Overlay Network ", in Proceeding of the Fourth Symposium on Operating Systems Design and Implementation, pp.197~212. 2000.

- [7] John Jannotti "Network Layer Support for overlay Networks" Massachusetts Institute of technology, August 2002
- [8] Y. Chawathe, "Scattercast: An architecture for Internet Broadcast Distribution as an Infrastructure Service", ph.D Thesis, University of California, Berkeley, 2000.
- [9] T. T. Do, K. A. Hua, M. A.Tantaoui, "P2VoD: Providing Fault Tolerant Video-on-Demand Streaming in Peer-to-Peer Environment". To appear in the IEEE International Conference on Communications(ICC 2004), Paris, France, pp.20-24, June. 2004
- [10] Yang Guo, Kyungwon Suh, James F. Kurose, Donald F. Towsley, "P2Cast: peer-to-peer patching scheme for VOD service", in Proceeding of the twelfth International Conference on WWW, 2003
- [11] Asit Dan, Dinker Sitaram, "Dynamic batching policies for an on-demand video server". *Multimedia Systems*, pp 112-121, 1996
- [12] "SSF Simulator implementation", <http://www.ssfnet.org/ssfImplementations.html>

최 성 욱(Sung-uk Choi)

[정회원]



- 1987년 2월 : 경희대학교 대학원 전자공학과 (공학석사)
- 2001년 2월 : 아주대학교 대학원 컴퓨터공학과 (공학박사)
- 1989년 3월 : ~ 1992년 2월 : 충남전문대학 조교수
- 1992년 3월 ~ 현재 : 시립인천전문대학 컴퓨터정보과 교수

<관심분야>

VOD(P2P, IPTV)네트워크 스케줄링, 분산운영체제, S/W 엔지니어링