

## 사이버 교육을 위한 P2P 기반 콘텐츠 전송시스템에서 효율적인 부하 분산 정책

김진일\*, 황윤철\*\*

### An Efficient Load Balancing Policy in P2P Contents Delivery System for Cyber Education

Jin-Il Kim\*, Yoon-Cheol Hwang\*\*

#### 요약

온라인 사이버 교육 시스템에서 수강생들에게 파일을 안정적으로 전송하는 것이 필수적이지만, 사용자의 요구가 일시적으로 많아질 경우, 처리부하가 집중되는 과부하 현상으로 시스템이 중지될 수도 있다. 이를 해결하기 위한 방법으로 P2P 기반의 소프트웨어적 해결책을 사용한다. 하지만, 이러한 해결책은 피어의 자율성을 고려할 경우, 서비스 지연 현상을 초래하거나 심지어는 부분적으로 서비스를 할 수 없는 결과를 초래하여 오히려 시스템의 효율성이 떨어지는 문제점이 있다. 본 논문에서는 P2P 네트워크 환경 하에서 피어의 자율성이 고려한 콘텐츠 전송 시스템에 적합한 효율적인 부하 분산 정책을 제안한다. 제안된 방법은 인접한 피어들의 서브네트워크를 구성하여, 서버의 부하를 분산시켰다. 또한 서브네트워크 내에서 서버 역할을 수행할 피어들이 다수일 경우, 각 피어들의 성능과 사용 패턴 정보를 기반으로 피어의 자율성으로 인한 서비스 지연을 최소화하는 서버 선정기법을 제안하고 기존의 방법에 비해 우수함을 시뮬레이션을 통하여 보였다.

#### Abstract

To perform on-line cyber education, it is indispensable to delivery contents file to participant users. When users' requests occur simultaneously in a shot period, server falls into overload phase. A common to avoid overload is a software solution based on P2P, which does no cost any additional expense. But, in this solution, it yields many problems such as service delay and system failure because of peer self-control. In this paper, we propose an efficient load balancing policy based on P2P system, which is the contents delivery system for cyber education system using idle computing power. In the proposed solution, we present server selection scheme called P2P-Pattern method, which results in minimum service delay. we can see that our scheme outperforms previous scheme.

▶ Keyword : 피어투피어투(Peer-To-Peer), 디지털 콘텐츠(Digital Contents), 전송(Delivery), 사이버교육(Cyber Education)

• 제1저자 : 김진일, • 교신저자 : 황윤철

• 접수일 : 2007.10.24, 심사일 : 2007.11.5, 심사완료일 : 2007.11.16.

\* 배재대학교 교양교육지원센터 교수 \*\* 충북대학교 전자계산학과 박사과정

## I. 서론

멀티미디어 기술의 급격한 발달로 인해 오디오, 영상, 그리고 비디오와 같은 다양한 디지털 콘텐츠가 증가하고 있다 [1]. 이와 더불어 교육의 형태도 캠퍼스를 벗어난 또는 지정된 장소와의 시간을 초월한 사이버 교육을 통한 열린교육의 형태가 추구하고 있다[2,3]. 특히, 사이버 교육은 컴퓨터를 이용한 그래픽 및 가상 기술과 빠른 정보 통신망으로 개인의 특성에 맞는 개별교육 및 대면교육과 비슷한 효과를 얻을 수 있는 형태로 교수와 학생들 간의 상호 작용을 지원해야 한다[2]. 즉, 앞으로는 단순한 지식전달 형태가 아니라 수강생 개개인의 감성적인 교감을 가질 수 있는 기능이 포함된 콘텐츠가 만들어져야 하기 때문에 단위 시간당 전송해야 할 데이터의 양이 점점 증가할 수 밖에 없다[4]. 그러므로 이러한 새로운 요구와 외적인 환경 변화에 적용할 수 있는 콘텐츠 전송 시스템이 필요성이 제기되고 있다.

최근에 이와 같은 문제를 해결하기 위해 P2P 네트워크를 활용한 다양한 연구들이 진행되고 있는데, 이러한 P2P 방식은 콘텐츠의 관리 부담을 클라이언트 PC에 분산시키므로 컴퓨터 시스템의 효율성을 극대화할 수 있고 사용자의 트래픽 또한 중앙 서버에 집중하지 않고 사용자 회선으로 분산하기 때문에 과도한 트래픽을 예방한다.

P2P 관련 연구들 중에 유휴컴퓨팅 자원을 이용한 콘텐츠 전송 시스템에 관한 연구가 진행되고 있지만, 대부분 P2P 네트워크에 참여하는 피어들이 탈퇴(leave)없이 계속 동작한다[5]는 가정 하에 이루어지고 있다. 하지만, 인터넷을 기반으로 하는 컴퓨팅 환경에서는 각 피어들이 자유롭게 참여 할 수 있고 언제라도 사용자가 임의로 콘텐츠를 서비스하는 도중에 탈퇴를 할 수 있기 때문에 피어의 자율성이 고려되어야 한다. 하지만 피어의 자율성을 고려할 경우, 서비스 지연 현상을 초래하거나 심지어는 부분적으로 서비스를 할 수 없는 결과를 초래하여 오히려 시스템의 효율성이 떨어지는 문제점이 있다[3].

따라서, 본 논문에서는 이러한 문제점을 해결하기 위해 피어의 자율성이 고려되더라도 콘텐츠 전송 시스템의 효율성을 높일 수 있는 부하 분산 정책을 제안한다. 제안된 방법은 인접한 피어들의 서브네트워크 구성 및 그룹 분할을 통하여 서브네트워크 내의 피어들에게 서비스를 제공함으로써, 서버의 부하를 분산시켰다. 또한 서브네트워크 내에서 서버 역할을 수행할 피어들이 다수일 경우, 각 피어들의 성능과 사용 패턴 정보를 이용하여 피어의 자율성으로 인해

발생하는 서비스 지연 및 결함을 예방하기 위한 효과적인 코스서버 선정기법을 제안한다.

## II. 관련 연구

콘텐츠 전송 시스템에는 웹기반 클라이언트/서버, 콘텐츠 전송 네트워크(Content Delivery Networks, CDN) 그리고 P2P 파일 공유 시스템(Peer-to-Peer file sharing system) 등으로 나누어질 수 있다. 사용자 관점에서 볼 때, 이 전송 시스템들은 동일하게 사용자에게 콘텐츠를 분배하는 역할을 제공하지만, 시스템들의 구조는 서로 다르고, 이러한 차이점은 시스템의 성능이나 작업부하에 영향을 미친다.

### 2.1 웹기반 클라이언트/서버

웹 기반 클라이언트/서버 방식에 대한 연구는 주로 웹 작업 부하[6], 웹 오브젝트의 특성[7], 그리고 웹의 하이퍼링크 구조 모델링 문제를 다루고 있다. 하지만 이러한 연구들은 주로 웹 오브젝트의 크기가 작은 경우에 한해서 진행되었다.

### 2.2 콘텐츠 전송 네트워크

콘텐츠 전송 네트워크는 서버와 클라이언트 사이에 전송할 콘텐츠를 캐시하고 캐시 서버를 이용하여, 클라이언트의 요청이 있으면 가까운 위치의 캐시 서버로부터 콘텐츠를 전송받도록 해서 전송비용을 감소시킨다.

콘텐츠 전송 네트워크에 대한 연구는 시스템의 독립적이고 폐쇄적인 특성으로 인해 연구가 활성화 되지는 않았지만 최근 연구에서 콘텐츠 전송 네트워크가 평균 다운로드 시간을 줄일 수 있음이 알려졌다[8,9]. 그러나 DNS 재설정 기술은 DNS 지연 때문에 상당한 오버헤드가 추가된다. 또 다른 연구에서는 콘텐츠 전송 네트워크의 중요한 장점으로 최적의 중복서버로 클라이언트를 라우팅하는 것이 아니라 클라이언트가 최악의 경우를 피할 수 있도록 도움을 줄 수 있다는 것이다[10]. 콘텐츠 전송 네트워크가 서버에 집중되는 부하를 캐시 서버로 분산시킬 수 있지만, 캐시 서버가 서비스할 수 있는 클라이언트의 수가 제한되어 있어 결국 클라이언트의 수가 증가할수록 캐시 서버를 확장해야만 한다.

### 2.3 P2P 파일 공유 시스템

P2P는 개별적인 작업처리 단위를 클라이언트와 서버로 구분하지 않고, P2P 네트워크에 참여하는 피어(peer)들이

작업 요청 및 작업 처리를 동시에 수행할 수 있게 함으로써 클라이언트/서버 방식에서 발생하는 문제점들을 개선하였다. P2P 기반 컴퓨팅 방식은 일반적으로 논리적인 네트워크 구조에 따라 하이브리드형과 순수형으로 분류할 수 있다. 순수형 P2P기반 방식은 중앙에 서버없이 피어들끼리 자기조직화를 통해 가상의 논리적 네트워크를 구성한다. 원하는 피어를 찾기 위해 이웃하는 피어들에게 질의를 보내어 피어의 위치를 찾는 완전 분산형으로 확장성이 뛰어나지만 P2P 고유 작업들 때문에 실제 작업처리를 위해 필요한 네트워크 트래픽보다 더 많은 양의 네트워크 트래픽을 유발할 수 있으며, 보안체계 구축, 관리 및 제어에도 어려움이 있다. 하이브리드형 P2P기반 방식에서는 디렉터리(directory) 서버가 각 피어의 위치정보를 보관하는 역할을 수행하여 임의의 피어가 다른 피어를 찾고자 할 때, 디렉터리 서버에 단순히 쿼리를 보내어 그 응답으로 원하는 피어를 찾을 수 있도록 한다. 또한 디렉터리 서버에는 피어들의 검색기능이외에도 인증 기능 등 일부 기능만을 중앙에서 관리하는 최소한의 서버 역할만 수행함으로써 피어들에게 강력한 보안체계를 구축할 수 있고, 필요에 따라 확장성과 제어의 수준조절이 가능함으로 공동 목적의 접속자를 갖는 범용 서비스 구현에 적합하다.

최근에는 유휴컴퓨팅 자원을 이용한 콘텐츠 전송 시스템에 관한 연구가 진행되었지만, 고정된 컴퓨팅 자원을 활용하는 데 초점이 맞추어져 있으며, P2P 네트워크에 참여하는 피어들이 탈퇴 없이 계속 동작한다(5)는 가정 하에 이루어지고 있다. 그러나 인터넷을 기반으로 하는 컴퓨팅 환경에서는 각 피어들이 자유롭게 참여 할 수 있고 언제라도 사용자가 임의로 콘텐츠를 서비스하는 도중에 탈퇴를 할 수 있기 때문에 기존의 방법을 그대로 사용하기는 어렵다. 즉, 작업이 임의적으로 시작되고 중지되는 현상이 발생할 수도 있고 유휴 컴퓨팅 자원을 활용하는 경우에는 각 피어의 자원이 콘텐츠 전송 서비스를 위해서만 사용하는 것이 아니기 때문에 개인적인 용도로의 사용으로 인해 일시적으로 중지되는 경우가 발생할 수도 있다. 이와 같은 피어의 자율성은 서비스 지연 현상을 초래하거나 심지어는 부분적으로 서비스를 할 수 없는 결과를 초래한다. 이것은 선정되는 특정 피어의 성능과 자원에 대한 고려가 없기 때문에 서로 다른 성능과 자원을 가지는 피어들로 구성된 시스템에서는 서비스 가용 시간을 예측할 수 없어 시스템의 효율성이 떨어지는 문제점이 있다. 그러므로 요청된 서비스에 대해 가상서버 역할을 하는 피어가 종료되지 않고 해당 서버 네트워크 그룹에서 지속적으로 서비스를 제공할 수 있는 시간을 예측하는 것이 중요하다.

### III. P2P 기반 콘텐츠 전송 시스템

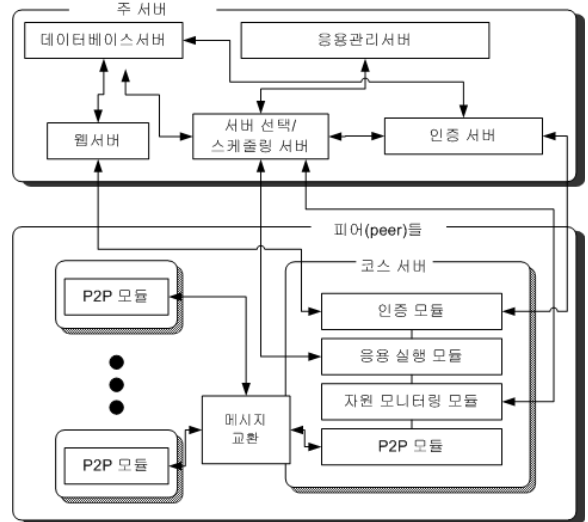


그림 1. P2P 콘텐츠 전송시스템의 개념  
Fig. 1 Concept of P2P Contents Delivery Systems

그림 1은 P2P 네트워크 환경에서 유휴 컴퓨팅 자원을 활용한 콘텐츠 전송 시스템의 개략적인 개념도이다(3). 주 서버(Main Server)는 피어들 및 회원관리, 콘텐츠 관리, DB 구축 및 운영, 홈페이지운영 및 관리 등의 기능을 수행하는, 중앙 관리 서버로 웹 서버(Web Server), 스케줄링 서버(Scheduling Server), 인증서버(Authentication Server), 데이터베이스 서버(Database Server), 응용 관리서버(Application Management Server)들의 집합체이다. 웹 서버는 사이버 강의에 관한 일반 정보를 홈페이지를 통해서 제공한다. 즉, 강의 수강을 위한 회원가입, 에이전트 프로그램의 다운로드 서비스 그리고 수강 통계정보 등을 제공한다. 스케줄링 서버는 활성중인 피어들로부터 서비스 요청이 있으면 콘텐츠를 제공하며 서비스 결과의 회수를 담당한다. 인증서버는 피어에게 인증서 발급/조회 서비스를 제공하여 올바른 피어로부터 전송된 데이터 및 결과, 그리고 메시지인지를 판별한다. 데이터베이스 서버는 피어들의 시스템 정보와 작업 정보, 그리고 회원정보의 관리를 담당한다. 응용관리 서버는 코스서버에 보내줄 데이터를 저장한다. 피어들은 사이버 콘텐츠 전송시스템에 회원으로 등록하고 자신의 PC를 제공하기로 동의한 PC들로, 필요에 따라 주

서버로부터 전송된 콘텐츠를 서브네트워크의 다른 피어들에게 제공하고 그 결과를 다시 주 서버로 제출하는 역할을 담당한다. 이러한 역할, 즉, 강의를 제공하기 위해 서브네트워크에서 선택된 피어를 코스 서버라고 한다.

### 3.1 전송 방법

과부하 발생으로 인해 주 서버의 응답 시간 지연, 서버 다운 현상이 나타나는 것을 예방하기 위한 전송 서비스 과정은 그림 2와 같다[11].

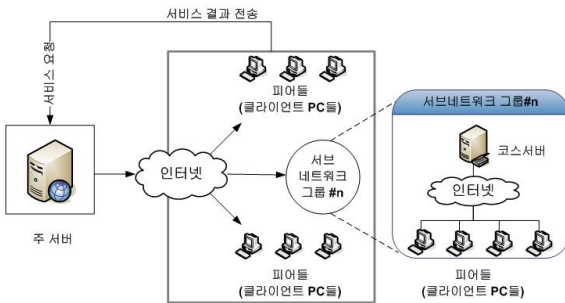


그림 2. 전송 서비스 과정  
Fig. 2. Process of Delivery Service

주 서버의 부하 상태를 정상 부하와 과부하 두 개의 상태로 구분하는 데, 부하 표현 방법은 표 1과 같다. 각 상태를 구분한 기준은 주 서버에 설정한 대기 큐의 길이(Main Server Queue Length, MSQL)를 기준으로 하여 임계값 정책(Threshold Policy)을 기반으로 한다. 정상 부하 상태는 임계값이 대기 큐의 길이보다 작은 경우( $MSQL \leq T$ )이며, 과부하는 임계값이 대기 큐의 길이보다 크거나 같은 경우( $MSQL > T$ )이다. 만약에 서비스 요청이 들어오면 부하 측정 프리시저가 호출되어서 자신의 부하를 측정한다. 만일 주 서버가 정상 부하 상태라면 주 서버는 추가적인 요청에 대해 서비스를 제공할 수 있지만, 만일 과부하 상태라면, 주 서버는 서비스를 요청한 피어(클라이언트)와 지리적으로 인접한 서브네트워크 내의 다른 피어들에게 요청에 대한 서비스를 해줄 수 있는지 질의한다. 여기서, 서브네트워크(subnetwork)란 IP 주소의 프리픽스가 동일한, 즉, 지리적으로 인접한 노드들의 집합을 말한다. 또한 정상 부하 상태라고 하더라도 프리픽스 체크 프리시저를 호출하여 등록된 프리픽스 그룹(Subnetwork\_Group(i))에 존재하면 프리픽스가 동일한 서브네트워크 내의 피어들에게 서비스를 해줄 수 있는지를 질의한다.

만약 서브네트워크 내에 지정된 코스 서버가 없다면, 주 서버에 있는 서버 선택/스케줄링 서버를 이용하여 서브네트워크 그룹 내에서 가장 적합한 피어를 서브네트워크 그룹에 대한 코스 서버로 선택한 다음, 필요한 세그먼트의 전송한다. 이 코스 서버는 서브네트워크 내의 피어들에게 직접 데이터 전송을 함으로써, 주 서버에 집중되는 부하를 분산시킬 뿐만 아니라 전체 네트워크상의 트래픽 발생을 줄일 수 있다. 이러한 방법으로 구성된 서브네트워크 그룹은 P2P 기반 콘텐츠 전송시스템을 구성하는 단위모듈이 된다.

표 1. 부하의 표현  
Table 1. Overload Roles

기준	의미
$MSQL \leq T$	정상부하 상태
$Subnetwork\_Group(i) \ni Peer(Prefix)$	과부하 상태로 처리
$MSQL > T$	과부하 상태

서버로부터 코스서버들에게 전송할 하나 이상의 서비스할 콘텐츠들이 존재할 때, 서비스할 콘텐츠들은 전송을 위해 하나로 통합되는 데, 이것을 코스 패키지(course package)라 한다. 코스 패키지란 교육에 필요한 콘텐츠로 관리되는 최소 단위를 의미한다. 하나의 서버에 다수의 코스 패키지들이 존재할 수 있으므로, 각 코스에는 고유의 식별자 CID(Course Identifier)를 갖는다. 그리고 주 서버로부터 코스 서버로 전송되는 코스패키지는 SCO(Shareable Content Object)라는 학습객체 개념의 학습단위 조각들로 구성되어 있으므로, 각 학습단위는 고유의 식별자 SCOID(Shareable Content Object Identifier)를 갖는다. 그림 3은 서비스할 내용들, 코스패키지, SCO와의 관계를 보여준다.

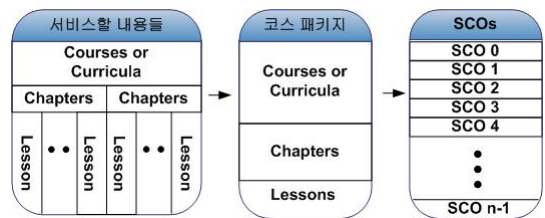


그림 3. 서비스 내용들, 코스 패키지, SCOs  
Fig. 3 Service Contents, Course package, SCOs

서비스할 코스 패키지를 분할 전송하지 않는다면, 해당 코스패키지가 완전히 전송될 때까지는 선택된 코스 서버가 클라이언트의 역할만을 수행할 뿐 서버로서의 역할을 수행

할 수 없다. 하지만 코스 패키지를 분할 전송하면, 코스 서버는 수신이 완료된 코스 패키지의 조각 즉, SCO를 이용하여 피어(클라이언트 PC들)의 요청에 서비스를 제공하는 서버의 역할을 수행할 수 있다. 현재 서브네트워크 그룹 내의 피어들의 수는 코스 서버의 과부하를 방지하도록 하기 위하여 50개로 제한하고 50개의 클라이언트 PC들이 접속하게 되면 주 서버는 해당 그룹을 분할하도록 한다.

서비스 요청에 따라, 주 서버가 코스패키지를 서브네트워크 그룹에서 선택된 코스 서버로 전송하게 되는 데, n개의 세그먼트로 분할된 코스패키지를 관리하기 위해, 두 개의 비트스트링 MSB(MainServer Bit string)과 CSB(Course Server Bit string)를 사용한다.

### 3.2 피어 ID 및 서브네트워크 그룹 구성

기존의 스케줄링 기법에서는 피어 자율성 결함과 피어 간섭의 개인작업 및 예상치 못한 서비스 중단과 같은 결함을 발생하게 되면 일반적으로 이에 대한 처리를 수행하는 스케줄링 기법을 제안하고 있다. 하지만 이러한 문제를 미리 예방하기 위한 방법을 고려하지 않았기 때문에 재처리 문제가 발생하게 된다. 즉, 더 이상 서비스가 진행되지 않아서 전체 서비스가 지연되고 봉쇄되는 현상이 발생하게 되어 정해진 시간 내에 교육이 진행되지 못하고 시간이 지연되는 문제점이 있었다. 그러므로 피어들 중에서 선정된 코스 서버가 클라이언트 PC가 요청한 서비스가 완료될 때까지 중지되지 않고 해당 서브 네트워크 그룹 내에서 지속적으로 서비스를 제공하는 것은 중요하다. 하지만, 피어들의 성능과 상태에 대한 고려가 없이 코스 서버를 선정하기 때문에, 서비스 지속 시간을 정확하게 예측할 수 없어 시스템의 효율성이 떨어지는 문제점이 있다.

따라서, 본 논문에서는 각 노드의 성능과 자원에 대한 정보를 기반으로 피어의 서비스 중단 결함을 예방하기 위한 방법을 제안한다. 이를 위해 먼저 피어 자원의 가용길이, 신뢰성, 사용자의 특성 등이 반영된 피어 ID 기법을 사용한다. 피어 ID는 각 피어들과 접속하는 사용자를 구분하기 위해 사용되며 요청되는 서비스의 특성에 따라 서비스를 하기 위한 서브네트워크 그룹을 구성한다. 서비스를 하는 과정에서 피어의 특성을 고려하여 피어 ID를 부여함으로써 서비스의 요구에 맞는 서비스 기능을 수행할 수 있다. 피어 ID는 피어 등록 이름으로 시스템에 자신을 등록한 후, 피어가 작업에 참여하는 경우에 부여되는 식별자를 나타낸다. 피어 ID는 피어의 특성, 즉 성능, 위치, 신뢰도, 사용패턴(PAT × Prediction Strategy) 등을 바탕으로 피어 ID 부여 함수

에 의해 새로운 피어 ID를 부여 받는다.

$$m\_nPeerId = SelectPeerId(PeerList, Method)$$

$$Method = (Performance, Location, P_{Trust}, UsagePattern)$$

여기에서, PeerList는 피어의 목록을 나타내고, Method는 피어의 성능, 위치, 신뢰도, 사용 패턴에 대한 선택 정책을 나타낸다. 피어가 정상적으로 동작하면서 주어진 서비스 작업을 수행할 수 있는 확률을  $P_{Trust}$  라고 정의한다.

$$P_{Trust} = \frac{MAT}{MAT + MFT}$$

여기에서, MAT는 평균 자원제공 시간(Mean Available Time)을 의미하고, MFT(Mean Failure Time)는 평균 자원제공 실패시간을 의미한다.

요청된 작업을 수행하는 경우, 피어 ID 함수는 먼저 피어의 위치에 따라 지리적으로 같은 지역이나 서로 인접한 지역에 속하는 피어들을 하나의 그룹으로 묶은 다음에 피어의 성능, 위치, 신뢰도, 사용패턴에 따라 피어 ID를 부여하도록 한다. 피어의 성능은 피어의 상태와 관련이 있고, 피어의 위치는 참여하는 피어의 위치에 따라 작업을 할당함으로써 통신비용의 절감 효과와 지역별로 작업을 할당할 수도 있다. 피어의 신뢰도는 피어가 결함이나 서비스 중단을 고려하여 얼마나 많은 시간 동안에 주어진 서비스 수행에 참여하는 가를 나타내는 기여도를 말한다. 사용 패턴은 피어가 일정한 시간대에 반복적으로 사용되는 유형을 의미한다.

### 3.3 사용 패턴

시스템 자원을 최대한 활용하기 위한, 효율적인 스케줄러는 서비스 요청이 있을 때, 서버 다운현상이 나타나지 않도록 가능한 예방해야하고, 또한 재서비스도 고려해야 한다. 이상적인 스케줄러는 서비스를 요청한 클라이언트 PC의 서비스 시간보다 큰 성능을 가지는 서버에 서비스를 할당하는 것이다. 이것은 서비스에 대한 재처리 발생하지 않고 주어진 시간 내에, 서비스를 제공하는 것을 의미한다.

피어의 사용 패턴을 예상하기 위한 방법은 가장 단순하면서도 효과적인 것이 관찰 방법이다[11]. 피어는 하루 중 예상할 수 있는 시간에 유희상태가 되는 경향이 있다. 일반적으로 사용자는 피어 즉, 자원제공 PC를 밤새도록 거의

사용하지 않을 뿐만 아니라 점심시간 동안에도 사용하는 경우가 드물다. 그리고 어떤 사용자는 매일 같은 시간대에 컴퓨터를 사용하는 경향이 있고, 특히, 주말과 휴일에 유희상태가 되는 경우가 많다. 또한, 강의를 듣고 있는 동안에도 유희상태(여기서는, CPU 사용률 10% 미만인 경우)인 경우가 많다. 만약, 어떤 사용자가 유희상태로 예정된 시간이 다 되었음에도 서비스 작업을 계속하고 있다면, 그 사용자는 갑자기 작업을 멈추는 것이 아니라 계속해서 서비스 작업을 하도록 한다. 이러한 관찰을 이용하여, 분 단위, 시간 단위, 하루 단위, 일주일 단위로부터 피어 활동성의 상호 연관성을 찾을 수 있다. 이러한 과정 속에서 예전의 사용 패턴에 대한 중요한 상호 연관성을 발견할 수 있다.

서비스 작업을 스케줄링하기 위해서, 다음  $n$  분에 대한 피어 가용길이를 예상하는 것이 중요하다. 여기서,  $n$ 은 스케줄러의 기간을 나타낸다. 다음에 소개되는 3 가지 예상 기법들은 다음에 수행 가능한 피어 가용길이  $n$  분은 이전에 사용한  $n$  분과 정확하게 일치할 것이라고 예상하는 것이다. 먼저, 최근 사용패턴(RECENT\_Pattern) 기법은 다음에 수행 가능한 피어 가용 길이  $n$  분이 마지막에 수행한 평균 피어 가용 길이  $n$  분과 정확하게 같다고 예상한다. 예를 들면 만약 이전의 40분에서, 평균 피어 가용시간이 25분이라면, 최근 사용 패턴은 다음 25분 동안 평균 피어 가용길이가 25분일 것이라고 예상하는 것이다. 일 단위 사용패턴(DAY\_Pattern) 기법과 주 단위 사용패턴(WEEK\_Pattern) 기법은 예상을 위하여 일 단위, 주 단위를 사용한다는 것을 제외하고는 최근 사용 패턴 기법과 비슷하다. 주 단위 사용패턴은 다음  $n$  분이 전날 같은 시간대의  $n$  분과 정확하게 같다고 예상한다. 주 단위 사용패턴은 다음  $n$  분이 지난 주 같은 시간대의  $n$  분과 정확하게 같다고 예상한다.

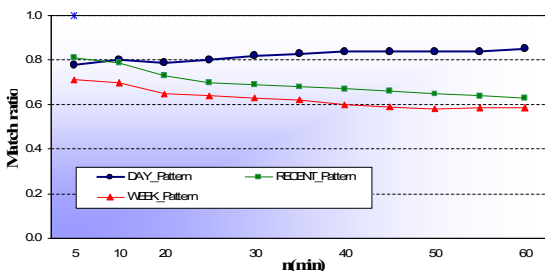


그림 4. 피어 사용패턴 예상 기법  
Fig. 4 Peer Usage Pattern Prediction Strategy

그림 4와 같이, 다음  $n$  분의 피어 가용 시간을 예상하는

가장 좋은 방법은 10분이하일 경우에는 최근 이전의  $n$  분을 예상하는 최근 사용패턴(RECENT\_Pattern)기법의 경우가 가장 좋은 결과를 나타내었고, 10분 이상일 경우에는 피어 가용길이가 전날 같은 시간대를 이용하는 일 단위 사용패턴(DAY\_Pattern)의 경우가 최상의 결과를 나타냈었다. 가장 최악의 예상 방법은 주 단위 사용패턴(WEEK\_Pattern) 기법이다. 피어 가용길이를 예상할 필요가 있는 스케줄링 알고리즘은  $n$ 이 작을 때는 최근 정보를 활용하고  $n$ 이 클 경우에는 전날 정보를 이용하면 우수한 알고리즘이 될 수 있음을 알 수 있다.

일반적으로, 콘텐츠를 서비스하는 경우에 한 강좌의 최소 크기가 20분 정도이므로, 10분미만으로 접속하는 경우는 드물다.

## IV. 실험결과 및 고찰

### 4.1 실험 환경

본 논문에서 제안하는 서버 환경은 다음을 만족한다. 첫째, 사용자가 요구하는 작업들은 모든 서버에 의하여 수행이 가능하다. 즉, 대상이 되는 모든 서버에 개설된 과목 대한 디지털 콘텐츠가 저장되어있고, 서비스가 가능하다. 둘째, 유희 컴퓨팅 파워를 이용하였으므로, 각 서버의 처리능력은 서비스 요청이 없더라도 시간에 따라 변화한다. 셋째, 사용자가 요구하는 서비스들의 예상 수행 시간은 예측이 가능하다.

시스템의 환경은 크게 하드웨어 및 소프트웨어 환경으로 구분할 수 있다. 시스템의 하드웨어 구현 환경은 주 서버 안정적으로 다수의 코스 서버 접속을 유지 관리하면서 콘텐츠를 전송하기 위한 시스템 사양이 필요하다. CPU는 셀러론 1.2 GHz \* 2, 메모리는 512 MByte, HDD : 100 GByte, NIC은 100 Mbps급 그리고 운영체제는 Windows 2000 서비스 팩 3이상으로 구성한다.

또한, 코스 서버는 일반적으로 PC 정도의 시스템에 하드디스크 용량과 램을 보강한 정도의 시스템으로 구성한다. CPU는 셀러론 1.2 GHz, 메모리는 512 MByte 정도, 하드디스크는 100 GByte, NIC은 100 Mbps급, SNIC 그리고 운영체제는 Windows 2000 또는 Windows XP Professional 이상으로 구성한다.

#### 4.2 실험결과 및 분석

본 논문에서 제안한 방법과 기존의 단일서버 및 P2P-First를 이용한 방식과의 성능 비교를 위해서 시뮬레이션을 수행하였다. 평균 대기시간과 평균 시스템 이용률을 평가 척도로 한다. 먼저, 성능 평가를 하기위해 주 서버의 2주 분량의 파일 전송 요청 로그를 분석하였고, 피어의 키보드와 마우스는 적어도 5분 동안 유휴상태에 놓이게 된다고 가정한다. 시뮬레이션 환경은 다음과 같다. 사이버 강의 과목이 5 곳의 실습실에서 디지털 콘텐츠 형태로 수업이 진행 되므로, 서버는 5 개의 서브네트워크에 속한 피어들의 요청을 처리한다. 각 서브네트워크에는 약 50개의 피어들이 존재하며, 서버는 총 250개 피어들의 요청을 처리한다. 그리고 서버와 피어사이의 통신 비용과 통신 오류에 의한 지체는 고려하지 않았다. 사이버 강좌를 실시하는 과목의 특성에 따라 콘텐츠 구성요소의 비율이 다르기 때문에 크기가 다양하지만, 서비스할 코스 패키지의 크기를 100M로 하며, 이것을 1MB의 크기를 갖는 100개의 SCO로 되어있다고 가정한다. 학습 요청의 발생 분포는 포아송 분포를 따르는 것으로 하였다. 초당 발생하는 요청 수를 5 건으로 하고 이를 증가시켜 가면서 실험했으며, 수 차례에 걸쳐 반복하였다.

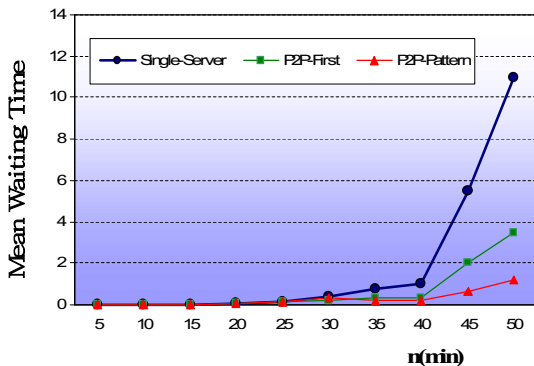


그림 5. 평균 대기시간 비교  
Fig. 5 Mean Waiting time

그림 5는 단일 서버, P2P-First 기법, P2P-Pattern 기법의 평균 응답 시간을 보여주고 있다. 여기서, P2P-First 방법은 유휴 상태에 놓인 피어들 중에서 처음 만나는 피어를 서버로 선정하는 기법이다. 피어로부터 서버로 전달되는 초당 요청 수가 약 25이하의 경우에는 전송 방법들 사이의 평균 대

기시간은 큰 차이를 보이지 않았지만 약 40 이상이 되면, 단일 서버의 평균 대기시간이 급속히 증가한다. 하지만 P2P를 이용한 전송방법의 경우에는 상대적으로 대기시간이 적게 증가한다. 이것은 피어가 학습 내용(SCO)을 받자마자 서버의 역할을 수행하기 때문에 서버의 수가 증가하는 효과가 있다. 즉, 유휴 자원을 이용하여 다른 피어들의 요청에 대한 서비스를 제공함으로써 서버의 부하를 분산시키는 효과가 있다. 또한, 일반적으로 상용하는 P2P-First 방법보다는 제안된 P2P-Pattern 방법이 더 우수함을 알 수 있다. 이것은 피어들의 성능과 사용패턴을 고려하지 않고 서버를 선정하는 경우에는 예상치 못한 서비스 중단으로 인한 재처리 시간으로 서비스 시간이 지연되는 반면에, 서비스 지속 시간을 예측하는 경우에는 시스템의 효율성이 높아진다는 것을 의미한다.

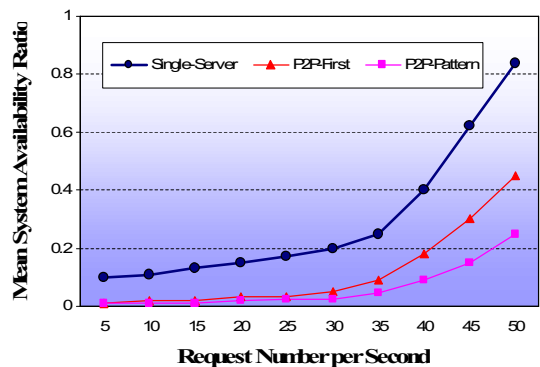


그림 6. 평균 시스템 이용률  
Fig. 6 Mean System Availability Ratio

그림 6은 단일 서버, P2P-First 방법, P2P-Pattern 방법에서의 초당 요청 수의 증가에 따른 평균 시스템 이용률의 변화를 보여준다. 전송방법에 관계없이 요청 수가 증가할수록 서버의 이용률이 증가했다. 특히, 단일 서버를 이용하는 경우에는 초당 요청수가 40을 넘으면 시스템 이용률이 급격하게 증가하는 것을 볼 수 있다. 하지만 P2P-First 방법과 제안된 P2P-Pattern 방법은 서비스 요청의 수가 증가할수록 서버의 역할을 하는 피어들의 수도 함께 증가해서 P2P 처리 능력이 향상된다.

#### V. 결 론

유휴자원을 활용한 P2P 기반 콘텐츠 전송 시스템은 각 피어들이 자유롭게 참여 할 수 있고 언제라도 사용자가 임

의로 콘텐츠를 서비스하는 도중에 탈퇴를 할 수 있기 때문에 피어의 자율성이 고려해야 한다. 하지만 피어의 자율성을 고려할 경우, 서비스 지연 현상을 초래하거나 심지어는 부분적으로 서비스를 할 수 없는 결과를 초래하여 오히려 시스템의 효율성이 떨어지는 문제점이 있다.

본 논문에서는 이러한 문제점을 해결하기 위해 피어의 자율성이 고려되더라도 콘텐츠 전송 시스템의 효율성을 높일 수 있는 부하 분산 정책을 제안했다. 첫째, 인접한 피어들의 서브네트워크 구성을 통하여, 서버의 부하를 분산시켰다. 만약, 서비스 요청이 들어오면 부하측정 프리시저가 호출되어서 주 서버의 부하 상태를 측정하여 정상 부하 상태라면 주 서버는 추가적인 요청에 대해 서비스를 제공하고 만일 과부하 상태라면, 인접한 피어들의 서브네트워크 구성 및 그룹 분할을 통하여 서브네트워크 내의 피어들에게 서비스를 제공함으로써, 전체 네트워크 상에 트래픽 발생을 줄였다. 그리고 정상 부하 상태라고 하더라도 프리픽스 체크 프리시저를 호출하여 등록된 프리픽스 그룹에 존재하면 과부하 경우와 동일하게 처리한다.

둘째, 각 피어들의 성능과 사용 패턴에 대한 정보를 이용하여 피어의 자율성으로 인해 발생하는 서비스 지연 및 결함을 예방하기위한 효과적인 코스서버 선정기법을 제안하였다. 이 기법은 피어의 자율성으로 인해 서비스 작업이 중지될 가능성이 있는 피어들은 회피하고 사용 패턴을 조사하여 지속적인 서비스가 가능한 피어를 선택함으로써 재처리 기회의 최소화하여 서버의 부하 분산에 기여한다. 즉, 실험 결과에서도 알 수 있듯이 유휴 상태에 놓인 피어들 중에서 처음 만나는 피어를 서버로 선정하는 P2P-First 방법 보다는 사용 패턴을 이용한 제안된 P2P-Pattern 방법이 더 우수함을 알 수 있었다. 이것은 피어들의 성능과 사용패턴을 고려하지 않고 서버를 선정하는 경우에는 피어 간섭의 개인작업 및 예상치 못한 서비스 중단으로 인한 재처리 시간으로 서비스 시간이 지연되는 반면에, 서버의 서비스 지속 시간을 예측하는 경우에 시스템의 효율성이 높아질 수 있음을 의미한다.

그리고 제안된 기법은 교육기관이 사이버교육을 점차 확대하거나 전면적으로 도입을 고려할 경우 추가적인 비용없이 기존의 구축된 인프라 및 자원을 활용하여 콘텐츠 전송 시스템을 구축할 수 있으므로 특히, 국제화 시대의 무한 경쟁에 내몰리고 있는 대학의 경쟁력 강화에 기반이 될 수 있으리라 사료된다.

향후 연구 과제로는 서브네트워크 내에 클라이언트의 수가 한계수량을 초과하였을 때, 해당 그룹을 분할하는 방법에 관한 연구가 필요하며, 세그먼트가 서버가 아닌 다른 피

어로부터 전송될 때, 세그먼트의 위/변조 여부를 확인할 수 있는 보안 기술에 대한 연구가 필요하다.

## 참고문헌

- [1] 이재혁의 3인, "영상의 예지 특성을 고려한 웨이블릿 기반의 적응적인 워터마킹 기법", 한국컴퓨터정보학회 논문지, Vol. 11, No. 2, pp.53-63, 2006.5.
- [2] 황대준, "사이버 스페이스상의 상호참여형 실시간 원격 교육시스템에 관한 연구", 정보처리 제4권 제3호, pp.29-40, 1997.5.
- [3] Dae J. Hwang, CBM based Integrated Multimedia Distance Education System, In Proceedings of International Conference on Online EDUCA, May 1996, Seoul, Korea.
- [4] 김진일, "인터넷기반 P2P컴퓨팅 환경에서의 콘텐츠 전송 서비스 시스템에 관한 연구", 인터넷정보학회 제 6권 제6호, pp.1-12, 2005.12.
- [5] 손세일, 이석균, "P2P를 이용한 배포 서버의 부하 분산", 정보처리학회논문지 A 제13권-A권 제 1호, pp.45-52, 2006.2.
- [6] V.N. Padmanabhan and L. Qiu. The content and access dynamics of a busy web site: Findings and implications. In Proc. of ACM SIGCOMM 2000, August 2000.
- [7] L. Breslau, P. Cao, L. Fan, G. Phillips, and S. Shenker. Web caching and zipf-like distributions: Evidence and implications. In Proc. of IEEE INFOCOM 1999, March 1999.
- [8] M. Koletsou and G. M. Voelker. The Medusa proxy: A tool for exploring user-perceived web performance. In Proc. of the Sixth Int. Workshop on Web Caching and Content Distribution, June 2001.
- [9] B. Krishnamurthy, C. Wills, and Y. Zhang. On the use and performance of content distribution networks. In Proc. of SIGCOMM IMW 2001, Nov. 2001.
- [10] Stefan Saroiu, An Analysis of Internet Content Delivery Systems, In Proc. of the Fifth Symposium on Operating Systems Design and Implementation, OSDI 2002.



- [11] Jinil kim, Sang-Gu Lee, Hee-Hyol Lee, Design of an Efficient Scheduling Policy in P2P Contents Delivery System for Cyber Education, Proceeding of Electronics, Information and Systems Conference Electronics, Information and Systems Society, I.E.E. of Japan, 2007.

## 저 자 소 개



### 김 진 일

한남대학교 컴퓨터공학과 공학박사  
고려대학교 교육학과 박사과정  
배재대학교 IT센터 책임강사  
(주)블루베리소프트 개발팀장  
현재, 배재대 교양교육지원센터 교수  
〈관심분야〉 병렬처리 및 네트워크, 퍼지이론, 교육 콘텐츠



### 황 윤 철

1994년: 한남대학교 전자계산공학과 졸업(공학사)  
1996년: 한남대학교 대학원 전자계산공학과졸업(MS)  
1999년~현재: 충북대학교 대학원 전자계산학과 박사 과정수료  
〈관심분야〉 네트워크, IDS, ITS, Ad Hoc 및 센서네트워크 보안