

碩 士 學 位 論 文

피어-투-피어 환경에서의  
적응적 콘텐츠 분배 기술

Adaptive Contents Distribution Techniques  
on Peer-to-Peer Environments

淑明女子大學校 大學院

컴퓨터科學科 컴퓨터科學 專攻

宋 閔 惠

碩 士 學 位 論 文

피어-투-피어 환경에서의  
적응적 콘텐츠 분배 기술

Adaptive Contents Distribution Techniques  
on Peer-to-Peer Environments

指導教授 金 倫 希

이 論文을 理學 碩士 學位 論文으로 提出함.

2007年 6月 日

淑明女子大學校 大學院

컴퓨터科學科 컴퓨터科學 專攻

宋 閔 惠

宋閨惠의 理學 碩士 請求 論文을 認准함.

피어-투-피어 환경에서의 적응적 콘텐츠 분배 기술  
Adaptive Contents Distribution Techniques  
on Peer-to-Peer Environments

2007年 6月 日

심사위원장                      教授(印)

위            원                      教授(印)

위            원                      教授(印)

淑明女子大學校 大學院

# 목 차

국문 요약 .....	V
제 1 장 서론.....	1
1.1 연구의 내용 및 방법 .....	2
1.2 논문의 구성 .....	2
제 2 장 관련연구.....	3
2.1 분산된 콘텐츠의 위치 검색과 라우팅 기법 .....	3
2.1.1 SPiDeR.....	4
2.1.2 Xeja.....	5
2.2 멀티캐스트 기술을 이용한 콘텐츠 분배 기법 .....	6
2.2.1 DOMTS.....	7
2.2.2 PROMISE .....	8
2.2.3 MMOM (Max-Min Overlay Multicast).....	9
2.3 관련 연구들과 본 연구와의 비교 .....	11
제 3 장 적응적인 콘텐츠 분배 관리 구조 설계 .....	12
3.1 적응적 분배 안정화 기법 .....	14
3.1.1 적응적 분배 트리 스케줄링 및 동적 재 분배 알고리즘 .....	14
3.1.2 적응적 분배 및 관리 모니터링.....	18
제 4 장 안정적인 콘텐츠 분배 관리 시스템 구현 .....	22

4.1	컨텐츠 분배 트리 구성 및 전송 서비스 .....	22
<b>제 5 장</b>	<b>실험 및 결과 분석.....</b>	<b>26</b>
5.1	실험 결과.....	27
5.1.1	컨텐츠 전송 완료 시간 측정 및 분석.....	28
5.1.2	컨텐츠 분배 트리 레벨 측정 및 분석.....	29
5.1.3	분배 트리 재구성 시 컨텐츠 완료 시간 측정 및 분석...31	
<b>제 6 장</b>	<b>결 론.....</b>	<b>33</b>
<b>부 록</b>	<b>.....</b>	<b>34</b>
<b>참 고 문 헌</b>	<b>.....</b>	<b>41</b>
<b>ABSTRACT</b>	<b>.....</b>	<b>44</b>

## 표 목 차

[표 1] 관련 연구 상의 논문과 본 연구와의 비교.....	11
[표 2] 콘텐츠 분배 트리 구성 알고리즘.....	15
[표 3] 콘텐츠 분배 트리 재구성 알고리즘.....	17
[표 4] 스케줄링 메타 정보.....	19
[표 5] 콘텐츠에 대한 메타 정보 표현.....	37

## 그림 목 차

[그림 1] P2P 구조 요약[2].....	4
[그림 2] SPiDeR 의 피어 광고 방법.....	5
[그림 3] Xejia 의 채널 기반 P2P 서비스.....	6
[그림 4] 멀티캐스트에서 오류 극복 방법 분류[11].....	7
[그림 5] 최적의 멀티캐스트 트리 구성.....	8
[그림 6] PROMISE 구조.....	9
[그림 7] 트리 구성 알고리즘 예.....	10
[그림 8] 적응적 콘텐츠 분배 관리 미들웨어 구조.....	12
[그림 9] 콘텐츠 분배 관리 흐름.....	14
[그림 10] 적응적 분배 및 모니터링 흐름도.....	20
[그림 11] 스케줄링 정보 XML.....	23
[그림 12] 콘텐츠 분배 트리 모니터링.....	24
[그림 13] 재분배 트리 구성 모니터링.....	25
[그림 14] 분배 트리 재 구성 흐름.....	26
[그림 15] 노드 개수의 변화에 따른 작업 완료 시간 측정.....	28
[그림 16] 5 개의 노드를 적용한 분배 트리 레벨 구성 빈도.....	29
[그림 17] 10 개의 노드를 적용한 분배 트리 레벨 구성 빈도.....	30
[그림 18] 20 개의 노드를 적용한 분배 트리 레벨 구성 빈도.....	30
[그림 19] 30 개의 노드를 적용한 분배 트리 레벨 구성 빈도.....	31
[그림 20] 피어 떠남 시 노드 개수의 변화에 따른 작업 완료 시간 측정.....	32
[그림 21] 슈퍼 피어, 헬퍼 피어, 에지 피어.....	35
[그림 22] P2P 네트워크 참여 및 채널 카탈로그 검색.....	36
[그림 23] 콘텐츠 메타 정보 XML.....	38
[그림 24] 콘텐츠 검색 화면.....	40

## 국문 요약

현재의 P2P 분산 환경에서 P2P 네트워크에 참여하는 디바이스의 종류가 점차 다양해지고 이러한 디바이스를 포함한 네트워크 구조도 다양해지고 있다. 그러나 이종의 네트워크에서 자원을 효과적으로 공유하고 원하는 자원을 쉽게 찾을 수 있기 위한 표준화가 이뤄지지 않고 있으며 다양한 네트워크 환경에 적용할 수 있는 서비스에 대한 고려가 부족하다. 또한 기존의 콘텐츠 공유 응용프로그램은 콘텐츠를 제공한 피어의 성능이나 대역폭에 따라 데이터 전송 속도가 급격히 떨어지는 단점이 있다. 이러한 유동적인 피어 환경에 적응하여 콘텐츠의 안정적인 분배를 지원하는 환경이 필요하다. 그러므로 개방형 표준을 지원하고 콘텐츠 분배의 연속성, 완결성을 보장하여 콘텐츠를 지속적이고 효율적으로 제공하기 위한 연구가 선행되어야 한다.

본 논문에서는 이러한 P2P 환경에서 웹 서비스 기술을 이용하여 개방형 표준을 고려한 콘텐츠 관리 기능과 유동적인 P2P 환경에서 콘텐츠의 안정적인 분배를 지원하기 위해 피어의 성능을 고려한 분배 트리를 작성하는 알고리즘과 피어의 참여, 떠남과 같은 환경변화에 적응하여 분배 트리를 재 작성하는 알고리즘을 적용한 분배 관리 기능을 지원하는 시스템을 설계하였다. 이를 본 연구의 프로토타입인 PACDS(P2P based Contents Distribution System)을 P2P 표준 플랫폼인 JXTA 프로토콜을 이용하여 구현하였고 실제 환경에서 실험을 통해 검증하였다.



## 제 1 장 서론

초기 인터넷에는 독립적인 서버나 서버를 엮은 클러스터들이 제공하는 서비스를 PC 와 같은 단말기가 접속해서 이용하는 서버 위주로 서비스가 전형적인 형태였다. 단말기의 컴퓨팅 파워가 증가하고 접속되는 단말기의 수가 거대해짐에 따라 이들을 효율적으로 이용할 수 있는 방법을 연구하게 되었다. 그 결과 P2P, Overlay, Grid 와 같이 발전된 분산 미들웨어 기술이 제시되었다. 이 중 P2P(Peer To Peer)는 서버가 아닌 인터넷상의 수많은 개인용 컴퓨터들을 직접 연결하여 데이터를 공유하게 되었다.

P2P 기술은 분산 네트워크 환경을 효율적으로 이용할 수 있는 기술을 제공한다. 이는 소수의 서버에 집중하기 보다는 망 구성에 참여하는 기계들의 계산과 대역폭 성능에 의존하여 통신망에 참여한 피어(Peer)는 자신의 자원(스토리지, 콘텐츠, 컴퓨팅 자원)등을 DNS(Domain Name Service)를 이용하지 않고 다른 피어와 통신하고 자원을 공유할 수 있다. 피어는 서버와 클라이언트 역할을 모두 수행할 수 있어 자신의 자원을 다른 피어와 공유할 수 있게 한다.

한편 유동적인 P2P 네트워크 환경에서 콘텐츠를 효율적이고 안정적으로 제공하기 위해 다양한 콘텐츠 분배 기술이 필요하게 되었다. 이들 기술은 네트워크를 구성하고 있는 피어 시스템간의 거리를 측정하고 데이터의 유효성과 유사성을 검사하며 유무선 네트워크 트래픽 양과 특정 시스템에 집중되는 부하를 분산시키거나 감소하는 방법을 제시한다. 대표적인 콘텐츠 분배 기술로 실제 콘텐츠는 피어 시스템에 나눠 저장하고 그에 대한 효율적인 탐색 방법을 제공하는 기술과 콘텐츠를 제공하는 주체가 수시로 변하는 환경에서 효과적으로 콘텐츠를 분배하는 기술을 들 수 있다. 또한 이 기종의 네트워크 환경을 지원하기 위해 웹 서비스 표준과 결합하여 통합된 기술 표준 형태로 서비스를 생성하고 지원하는 방안이 연구 중에 있다.

그러나 현재의 P2P 분산 환경에서는 통합된 표준 없이 다양한 네트워크 환경에 적응할 수 있는 서비스에 대한 고려가 부족하고 유동적인 피어 환경에 적응하여 콘텐츠의 안정적인 분배 환경 지원이 어렵다.

그러므로 개방형 표준을 지원하고 콘텐츠 분배의 연속성, 완결성을 보장하여 콘텐츠를 지속적이고 효율적으로 제공하기 위한 연구가 선행되어야 한다.

본 논문에서는 P2P 환경에서 웹 서비스 기술을 적용하여 콘텐츠 정보를 관리하고 피어를 기능별로 분류하여 P2P 네트워크를 구성하는 구조를 제시한다. 또한 콘텐츠 별로 채널 그룹을 생성하고 콘텐츠 분배 트리를 적용하여 효율적이고 안정적인 콘텐츠 분배 기술을 제안한다.

## 1.1 연구의 내용 및 방법

구체적인 연구의 내용과 방법을 제시하면, 다음과 같다.

첫째, 이 연구를 수행하기 위하여 P2P 환경을 지원하는 다양한 시스템에 관한 관련 문서를 통해 P2P 기반 분산 환경 모델의 기능에 대한 요구사항을 분석한다.

둘째, 개방형 표준을 고려하고 P2P 기반 분산 환경에서 콘텐츠 이동의 안정화를 지원하는 미들웨어 구조를 제안한다.

셋째, P2P 환경을 지원하는 시스템을 설계하고 프로토타입 시스템인 PACDS(P2P based Adaptive Contents Distribution System) 구현 및 실험을 통해 검증한다.

## 1.2 논문의 구성

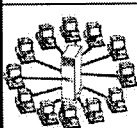
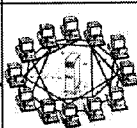
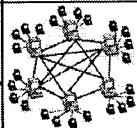
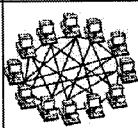
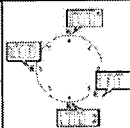
1 장은 서론으로 연구의 필요성과 목적, 연구 내용과 방법, 논문구성 등에 대해 기술하였다. 2 장에서는 기존의 콘텐츠 분배의 안정성을 지원하는 기술들을 살펴보고 P2P 기반에서 웹 서비스를 지원하기 위한 방법들에 대해 알아본다. 3 장과 4 장에서는 본 연구에서 제안하는 콘텐츠의 안정화된 분배 관리 기법에 대한 설계 및 구조를 소개하고 5 장에서 실험 후 그 결과를 분석해 제안한 방법의 성능을 검증하며, 6 장에서 결론을 정리한다.

## 제 2 장 관련연구

본 장에서는 P2P 환경에서 콘텐츠 분배의 효율성과 안정성 지원을 고려한 다양한 기법에 대해 살펴보고 이와 관련된 연구에 대해 살펴본다. 또한 이들 관련 연구와 본 논문과 비교 분석한다.

### 2.1 분산된 콘텐츠의 위치 검색과 라우팅 기법

P2P 네트워크 인프라는 [그림 1]와 같이 그 구성 방식에 따라 Structured 방식과 Unstructured 방식으로 구분할 수 있다. 또 자원 검색 방식에 따라 중앙집중 형 구조와 분산 구조로 구분이 되기도 한다. P2P 네트워크에서는 기존의 클라이언트/서버 모델과는 달리 자원이 다수의 피어에 분산되어 있기 때문에 여러 자원을 관리하고 검색하는 기술이 필수적이다. Unstructured P2P 를 기반으로 하는 디스커버리(Discovery) 기법으로는 중앙집중 형 P2P(Centralized P2P), 분산 형 P2P(Pure P2P), 계층형 P2P(Hybrid P2P)가 있다. [1] 반면 Structured P2P 오버레이 네트워크 구축 기술은 Unstructured 방식에 비해 비교적 최근에 제안된 P2P 프레임 구축 기술로 DHT(Distributed Hash Table)를 기반으로 구축되는 추세로 대표적인 프로토콜로 Chord[3], Pastry[4], Tapestry[5], CAN(Content Addressable Network) [6] 등이 있다. DHT 는 일반 해쉬 테이블을 응용 프로그램 계층에서 논리적으로 생성되어 물리적인 변화에도 상관 없이 동작하는 오버레이 네트워크에 적용한 것으로 DHT 기반의 오버레이 네트워크에 참여한 피어들은 <key, value> 쌍의 일부를 저장하고 있어 전체 네트워크는 커다란 해쉬 테이블로 표현될 수 있다. [7]

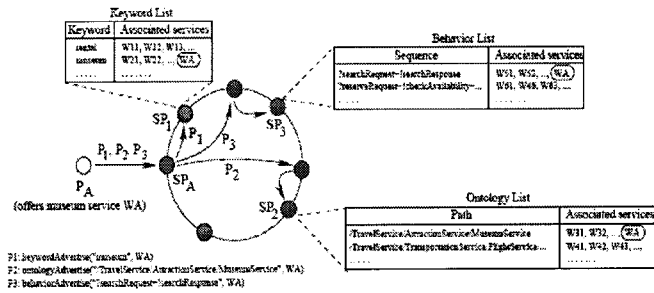
<b>Client-Server</b>	<b>Peer-to-Peer</b>			
1. Server is the central entity and only provider of service and content → Network managed by the Server 2. Server as the higher performance system. 3. Clients as the lower performance system Example: WWW	1. Resources are shared between the peers 2. Resources can be accessed directly from other peers 3. Peer is provider and requestor (Servernt concept)			
	<b>Unstructured P2P</b>			<b>Structured P2P</b>
	<b>Centralized P2P</b>	<b>Hybrid P2P</b>	<b>Pure P2P</b>	<b>DHT-Based</b>
	1. All features of Peer-to-Peer included 2. Central entity is necessary to provide the service 3. Central entity is some kind of index/group database Example: Napster	1. All features of Peer-to-Peer included 2. Any terminal entity can be removed without loss of functionality 3. → dynamic central entities Example: Gnutella 0.6, JXTA	1. All features of Peer-to-Peer included 2. Any terminal entity can be removed without loss of functionality 3. → No central entities Examples: Gnutella 0.4, Freenet	1. All features of Peer-to-Peer included 2. Any terminal entity can be removed without loss of functionality 3. → No central entities 4. Connections in the overlay are "fixed" Examples: Chord, CAN
				

[그림 1] P2P 구조 요약[2]

### 2.1.1 SPiDeR

SpiDeR[8]은 피어 사이에 제공하는 서비스를 발견하고 P2P 환경에서 실행할 수 있는 기초적인 미들웨어를 지원하는 시스템으로 다양한 서비스 프로바이더(Provider)들로 구성된다. 이들은 다른 피어가 제공한 광고를 저장하고 동적이면서 분산된 룩업(Lookup)서비스를 통해 서비스를 검색한다. SPiDeR 는 다음과 같은 다양한 검색 방법을 제공한다. 첫째, P2P 네트워크에 참여한 피어가 제공한 광고 검색, 둘째, 서비스 프로바이더에 의한 키워드 중심 검색, 셋째, 온톨로지(Ontology)로부터 분류하는 온톨로지 중심 검색, 넷째, 경로 행동에 기초한 검색 등을 통해 사용자가 원하는 서비스를 검색할 수 있다.

SPiDeR 는 [그림 2]에서 보는 바와 같이 코드 링 방식을 이용하여 서비스 프로바이더를 위치 시킨다. 이 중 일부 프로바이더는 슈퍼 피어로 대체되어 참여한 다른 프로바이더가 제공한 콘텐츠에 대한 데이터를 소유하고 관리한다.

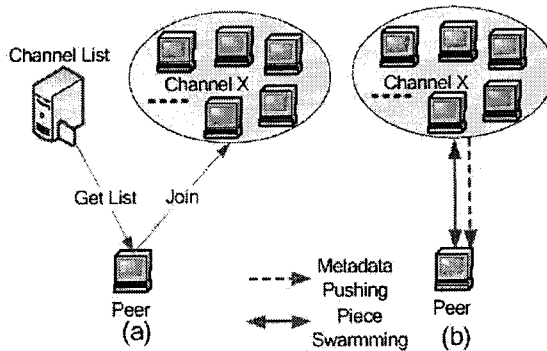


[그림 2] SPiDeR의 피어 광고 방법

### 2.1.2 Xeja

현재의 인터넷 환경은 개방적이고 전세계에 걸쳐 접근 가능하다. 이러한 환경에서 산재되어 있는 방대한 데이터를 빠르고 쉽게 공유하기 위한 분배 네트워크가 필요로 하게 되었다. Xeja[9] 시스템은 전송속도와 검색의 성능을 향상시키기 위한 서비스를 제공한다. 즉, 콘텐츠에 대한 디렉터리 서비스와 콘텐츠 조각에 대한 관리를 통해 빠른 검색 서비스와 콘텐츠 전송 서비스를 제공한다. Xeja는 P2P 멀티 소스 콘텐츠 분배 시스템으로 유연한 채널 기반의 분배된 디렉터리 서비스를 지원한다.

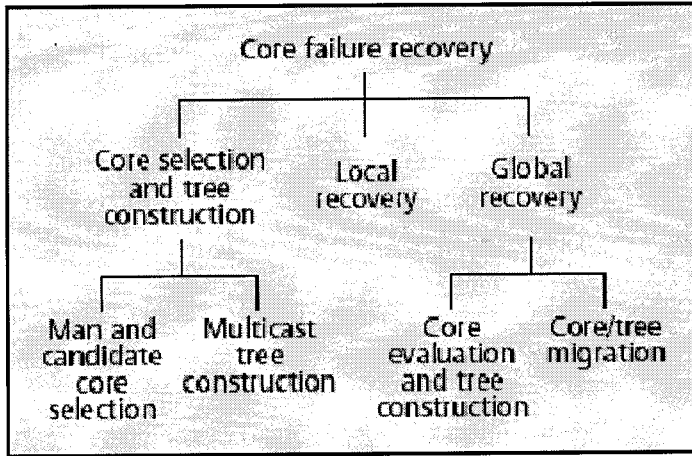
이는 자동적으로 사용자에게 대한 정보를 모으고 [그림 3]와 같이 참여 피어들 사이에 채널을 생성하고 이를 계층적으로 관리한다. 공통의 관심을 갖는 피어끼리 채널을 생성하고 콘텐츠를 제공 받는다. 이때 모든 피어 사이에 콘텐츠 조각을 분배함으로써 전체 콘텐츠에 대한 가용성을 증가시켰다. 또한 멀티 소스 피어로부터 병렬적으로 파일을 다운로드 할 수 있다.



[그림 3] Xeja의 채널 기반 P2P 서비스

## 2.2 멀티캐스트 기술을 이용한 콘텐츠 분배 기법

인터넷 환경 상에서 그룹 간 통신의 기하급수적인 증가와 통신량의 증가로 인한 네트워크 부하를 감소시키기 위해 멀티캐스트 기술을 등장하게 되었다. 멀티캐스트는 같은 내용의 데이터를 여러 명의 특정한 그룹의 수신자들에게 동시에 전송하는 방식으로 데이터 중복 전송으로 인한 네트워크의 부하를 줄일 수 있다. [11]에서는 멀티캐스트의 생명 주기와 오류 발생 시 이를 극복할 수 있는 오류 방법에 대해 설명되어 있다. 멀티캐스트는 특정 그룹을 생성하고 그 그룹에 있는 피어들에게 최적의 분배 방식을 적용하여 데이터 전달 후 멀티캐스트 그룹을 소멸하는 것만을 생명 주기로 정의하였다. 또한 유동적인 P2P 네트워크 환경에서 피어의 참가와 떠남이 발생하였을 경우 현재 진행 중인 멀티캐스트의 효율성을 유지할 수 있는 오류 극복 방법을 [그림 4]와 같이 분류하였다. 이러한 멀티캐스트 방식은 다자간 화상회의와 같은 음성 및 화상 데이터의 실시간 전송이 필요한 환경에서 꼭 필요한 방식으로 이러한 멀티캐스트 방식을 이용한 연구는 다음과 같다.

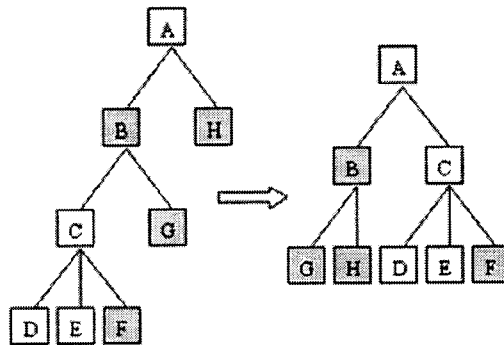


[그림 4] 멀티캐스트에서 오류 극복 방법 분류[11]

### 2.2.1 DOMTS

DOMTS(Dynamically Optimize Multicast Tree Structure)[12]는 Tsinghua University 에서 진행된 P2P 환경 지원 멀티캐스트 연구로 소스-투-엔드 (Source-To-End) 지연이나 네트워크 대역폭의 낭비와 같은 오버레이 멀티캐스트의 문제점을 극복하고자 동적으로 최적화된 멀티캐스트 트리 구성 방법을 제안한다. DOMT(Dynamically Optimize Multicast Tree) 프로토콜은 동적인 네트워크 환경에서 물리적인 네트워크 계층의 대역폭에 대한 낭비와 피어 간 지연 시간을 줄이는 방안을 고려한다. DOMT 는 가십 (Gossip)기반의 오버레이 네트워크를 구성하고 이러한 오버레이 네트워크 상에 가십에서 제공하는 멀티캐스트 트리 구성 방법을 이용하여 멀티캐스트 트리를 작성한다. 그러나 일정 시간을 두고 피어의 시스템 기여도를 측정하여 최적화된 멀티캐스트 트리를 재 구성한다. 또한 동일한 네트워크를 사용하고 있는 피어를 검사하여 트리의 반대편에 존재하게 하여 전송 중 피어의 떠남과 같은 오류 발생 시 전체 트리의 손상을 최소화 하도록 한다. 즉, [그림 5]에서 보는 바와 같이 A, C, D, E를 네트워크 a 을 사용한다 정의하고 B, F, G, H를 네트워크 b를 사용한다고 정의하였을 경우, 왼쪽의 트리는 네트워크 b 의 오류 발생시 전체 트리에 영향이 미

치지만 오른쪽 트리는 최적의 멀티캐스트 트리 구성 방법을 적용하여 네트워크 b 오류에도 불구하고 네트워크 a 상에 존재하는 피어는 안정적으로 콘텐츠를 분배할 수 있다.

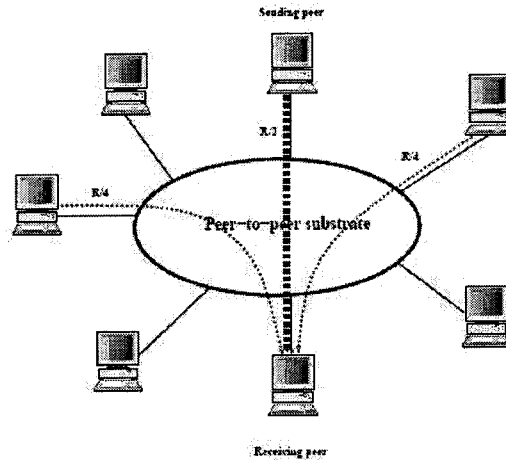


[그림 5] 최적의 멀티캐스트 트리 구성

### 2.2.2 PROMISE

PROMISE[13]는 Purdue University에서 연구 중인 프로젝트로 피어 록업, 피어 기반의 모아진 스트리밍, 네트워크와 피어 상황에 동적인 적응 등이 주요 기능을 갖는 새로운 피어 투 피어 스트리밍 시스템이다. 특히, CollectCast 라고 불리는 새로운 어플리케이션 레벨의 p2p 서비스를 기반으로 하고 있다. CollectCast 는 네트워크 토폴로지와 성능 정보를 기본으로 공급자를 위한 선택 추론과 피어와 연결 상태에 대한 모니터링과 피어와 연결 실패에 대한 적은 오버헤드로 재 시도하고 동적으로 활성화된 공급자 피어와 동적인 스위칭이 가능하여 네트워크에 대한 성능을 만족시키는 기능들을 제공한다.



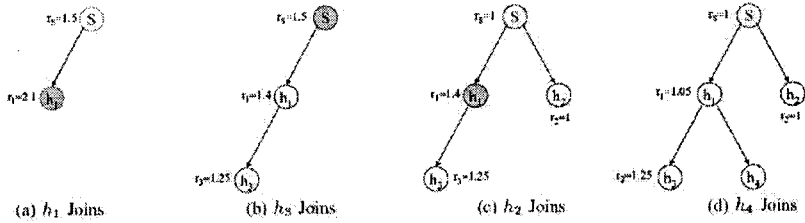


[그림 6] PROMISE 구조

PROMISE 구조는 [그림 6]과 같이 P2P 기판을 통하여 상호 연결된 피어들의 집합으로 구성된다. 엄밀히 PROMISE 는 콘텐츠를 전달하기 위하여 멀티캐스트 분배 트리를 구축하지 않는다. 대신, 어떤 피어가 어떤 콘텐츠를 요청할 때, 먼저 하부 P2P 기판에 검색 요청을 제출하고, 그 콘텐츠를 소유하고 있는 후보 피어들의 집합을 반환 받는다 그 중 최적 부분집합인 활동적인 송신자 집합을 결정하기 위하여 위상-인식 선택 알고리즘(topology-aware selection algorithm)을 사용한다. 그 후, 수신자는 집합 내의 모든 피어들과 병렬로 연결을 시도하고 연결이 성공하면, 수신자는 송신자들에게 송신율과 콘텐츠 할당을 배정하고 전송을 시작한다. PROMISE 에서, 노드 결합과 이탈은 하부 P2P 기판에 의해 처리된다.

### 2.2.3 MMOM (Max-Min Overlay Multicast)

Max-Min Overlay multicast[14]는 Illinois University 에서 연구 된 프로젝트로 피어의 성능을 기반으로 트리를 구성한다. 어플리케이션 레벨의 멀티캐스트 트리를 구성하여 보다 유연하고 빠른 콘텐츠 전송을 지원한다. 이 때 피어의 분배 트리를 구성하고 최적의 비율을 피어 마다 적용하도록 한다. 또한 트리를 구성함에 있어 전송률에 대한 계산을 적용하여 분배 트리를 생성한다.



[그림 7] 트리 구성 알고리즘 예

[그림 7]에서와 같이 피어의 참여가 있을 때 동적으로 분배 트리를 구성하며 계산식에 의해 가장 성능이 좋은 피어를 미리 선택한 후 다음 피어가 참여를 요청하였을 경우 부모 노드로 선택하게 한다. 즉,  $h_3$  이 참여하였을 경우  $h_1$  이 참여한 후 선택된  $h_1$  을  $h_3$  의 부모 피어로 선택하고  $h_2$  는  $S$  를 부모 피어로 선택하며  $h_4$  는  $h_1$  을 부모 피어로 선택하여 트리를 구성한다. 이때 피어의 요청순서에 따라 분배 트리가 구성되며 피어의 떠남이나 참여가 없을 경우 분배 트리 모양은 변하지 않는다. 다만 부모 - 자식 교환에 의해 성능의 차이가 큰 부모 피어와 자식 피어의 위치만을 변화시킨다. 피어의 떠남이 감지 되었을 경우 상위의 부모 피어에게 콘텐츠를 재 요청하게 되며 피어의 참여는 기존과 동일한 방식으로 트리를 구성하게 된다.

### 2.3 관련 연구들과 본 연구와의 비교

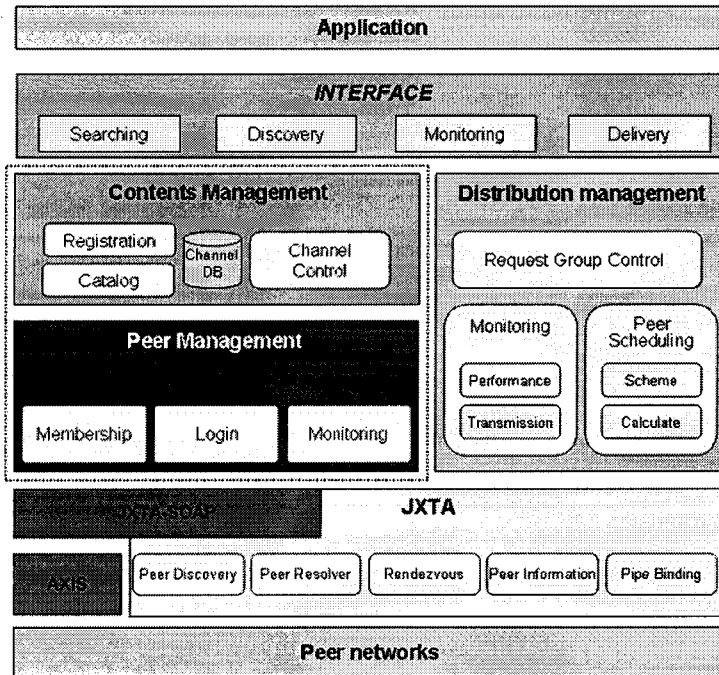
[표 1] 관련 연구 상의 논문과 본 연구와의 비교

	SPiDeR	Xeja	DOMT	PROMISE	MMOM	PACDS
컨텐츠 관리 지원	o	o	x	o	x	o
분배 안정화 지원	x	o	o	x	o	o
자동적 재분배 실행 지원	x	x	o	o	o	o

[표 1]은 기존에 개발되어 P2P 환경에서 컨텐츠 분배를 제공하는 미들웨어와 본 논문의 프로토타입인 PACDS 을 비교한 것이다. 제시되어있는 기존 미들웨어들은 각각 컨텐츠에 대해 검색하고 이를 효율적으로 분배하기 위한 다양한 방법들을 적용하였다. 그러나 적응적 컨텐츠 분배 안정화에 대한 부분이 결여되어 있다. 또한 컨텐츠 분배 안정화와 자동적 재 분배를 실행하는 MMOM 의 경우 효율적인 분배를 지원하지 못한다. 이 논문에서 제안하는 미들웨어 구조는 P2P 환경에서 적응적 분배 트리를 구성하고 오류에 대해 자동적으로 재분배를 실행하여 기존 연구들보다 사용자에게 유연하고 안정적인 환경을 제공한다.

### 제 3 장 적응적인 콘텐츠 분배 관리 구조 설계

유동적인 P2P 환경에서 콘텐츠의 안정적이고 지속적인 제공을 위해 이 종의 환경에서 적용 가능한 콘텐츠 관리 서비스 제공과 동적인 환경에 적응적인 콘텐츠 분배 관리 서비스를 지원하는 미들웨어가 필수적으로 필요하다. 이에 따라 [그림 8]과 같이 개방형 표준을 고려하고 P2P 기반 환경에서 콘텐츠 이동의 적응적 안정화를 지원하는 미들웨어를 설계하였다.



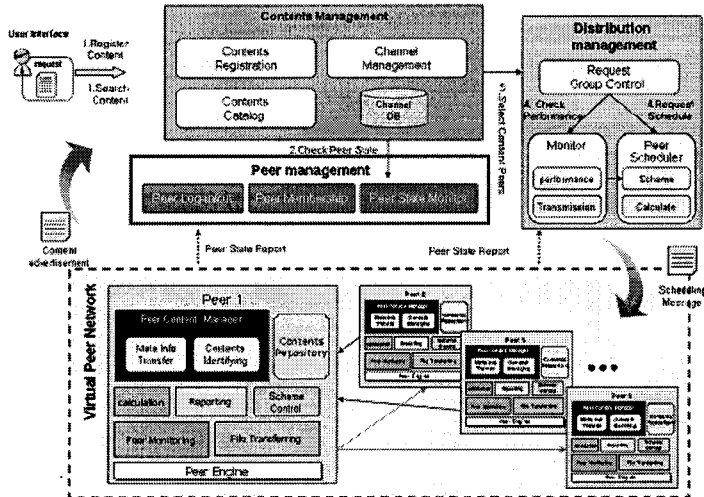
[그림 8] 적응적 콘텐츠 분배 관리 미들웨어 구조

콘텐츠 분배 안정화를 지원해주는 미들웨어는 콘텐츠 관리 부분과 P2P 네트워크에 참여한 피어 관리 부분, 콘텐츠 분배 관리 부분으로 구

성된다. 피어 관리 부분은 P2P 오버레이 네트워크에 참여한 피어에 대한 관리 부분으로 피어에 대한 멤버십과 로그인, 모니터링 서비스를 제공하며 콘텐츠 관리는 사용자로부터 요청되는 일련의 콘텐츠에 대해 웹 서비스 형태로 제공한다. 즉, 방대한 콘텐츠 정보를 채널 관리 서비스를 통해 채널 별로 관리하여 검색을 요청한 사용자에게 쉽게 검색 가능하도록 지원한다. 또한, 콘텐츠의 상태를 확인하여 콘텐츠 요청자에 맞는 최적의 콘텐츠를 선택하여 제공한다. 콘텐츠 분배 안정화 지원 미들웨어의 콘텐츠 관리 부분과 피어 관리 부분의 설계에 대한 자세한 내용은 부록 A 를 참고한다.

분배 관리 부분은 콘텐츠를 요청한 다수의 사용자들에게 안정적으로 콘텐츠를 제공한다. 이는 유동적인 피어에 대해 끊임없이 콘텐츠를 제공 받을 수 있도록 피어의 상태에 따라 콘텐츠를 제공받은 순서를 스케줄링 하고 지속적으로 요청한 피어들에 대한 상태를 모니터링 하여 콘텐츠를 제공받는 중 피어의 떠남이 감지되면 적응적으로 콘텐츠를 가지고 있는 새로운 피어에 대한 정보를 전달함으로써 지속적으로 콘텐츠를 제공받게 한다.

### 3.1 적응적 분배 안정화 기법



[그림 9] 콘텐츠 분배 관리 흐름

P2P 기반 분산 환경에서 콘텐츠를 안정적으로 제공하기 위해서는 요청한 콘텐츠를 되도록 빠른 시간 내에 받을 수 있도록 제공하고 그 시간 동안에는 제공하는 피어가 P2P 네트워크에 지속적으로 참여하고 있어야 한다. 이러한 한계를 극복하기 위해 본 연구에서는 [그림 9]와 같이 콘텐츠를 제공하는 피어의 선택에 있어 전체 콘텐츠를 제공할 수 있으면서 피어가 제공받은 동안 지속적으로 네트워크에 있을 수 있는 피어를 시스템 적으로 선택하게 하고 해당 콘텐츠를 요청하는 피어에게 빠르게 전달할 수 있는 콘텐츠 분배 트리를 작성하고 지속적으로 모니터 하여 피어의 떠남을 감지하면 적응적으로 콘텐츠 분배 트리를 재 구성함으로써 콘텐츠를 안정적으로 전달할 수 있도록 하였다.

#### 3.1.1 적응적 분배 트리 스케줄링 및 동적 재 분배 알고리즘

소수의 콘텐츠를 요청하는 다수의 피어가 존재 하였을 경우 이들에게 안정적으로 콘텐츠를 분배하기 위해 콘텐츠 요청이 들어왔을 경우 이

를 바로 제공하지 않고 일정 시간 동안 요청된 동일한 요청에 대해 분배 트리를 작성하는 방법에 대해 고려하였다. 이를 위해 해당 피어 정보를 관리하는 서비스가 필요하다. 피어는 각자 자신의 대역폭을 체크하고 요청 그룹 관리 서비스에 의해 해당 정보가 관리된다. 콘텐츠를 처음 요청한 피어의 요청 시간부터 정해진 시간까지 다른 피어의 요청을 기다리고 해당 시간이 완료되면서 콘텐츠 분배 트리를 구성한다. 분배 트리 구성 알고리즘은 [14]의 멀티캐스트 트리 알고리즘을 기반으로 적용하였다. [표 2]는 요청 피어들을 이용한 콘텐츠 분배 트리 구성 알고리즘으로 내용은 다음과 같다.

[표 2] 콘텐츠 분배 트리 구성 알고리즘

```

/*Initialization*/
1. 콘텐츠 분배 트리 T
2. 콘텐츠 요청 피어 셋  $H = \{h_1, \dots, h_N\}$ , 대역폭이 큰 순으로 정렬
3. k 번째 요청 피어의 대역폭,  $R_k$ 
4. k 번째 요청 피어의 Depth,  $D_k$ 
5. 소스 S의 대역폭  $R_s$ 
/*Tree Construct */
6. for i = 0 to N
7.    $h_k = \underset{k=1, \dots, i-1}{\operatorname{argmax}}(R_k) \ \&\&\underset{k=1, \dots, i-1}{\operatorname{argmin}}(D_k)$ 
8.   if  $R_s > R_k$ 
9.     Set S as  $h_i$ 's Parent
10.  else
11.    Set  $h_k$  as  $h_i$ 's Parent
12.  T = T U  $h_i$ 
13.   $R_s = R_s / \text{Linknum} + 1$ 
14.  for j = 1 to i
15.     $R_j = R_j / \text{Linknum} + 1$ 

```

1~5 는 안정적인 콘텐츠 분배 알고리즘에 사용된 변수들에 대한 내용으로 T 는 알고리즘을 통해 작성된 분배 트리를 나타낸다. H 는 콘텐츠를 요청한 피어들의 집합으로 피어의 대역폭이 큰 순으로 정렬된다. k 번째 요청한 피어의 대역폭을  $R_k$ , k 번째 요청한 피어의 트리 레벨을  $D_k$ , 콘텐츠를 소유한 소스 피어의 대역폭에 대해  $R_s$  로 표현한다.

6~15 는 실제 분배 트리를 작성하는 부분으로 14~15 는 트리에 참여한 피어의 대역폭을 계산하는 부분으로 자식 피어가 생길 때 마다 그에 해당 하는 수만큼 대역폭 값이 줄어든다. 6~12 에서는 콘텐츠를 요청한 피어들에 대해 대역폭 계산 값과 피어의 분배 트리 레벨을 비교하여 다음에 올 피어의 부모 피어를 선택하는 방법으로 소스 피어와 비교하고 그 외 참여하고 있는 피어들과 비교하여 피어들을 분배 트리로 위치시킨다. 이때 대역폭이 같은 피어들에 대해서는 분배 트리의 레벨을 고려하여 레벨이 낮은 피어를 우선적으로 선택하게 된다. 대역폭과 분배 트리의 레벨 고려는 기존의 알고리즘에서 추가로 작성한 부분으로 분배 트리 구성에 있어 좀 더 안정된 트리 구성을 위해 추가하였다. 또한 동적인 P2P 환경을 고려하여 새로운 피어가 콘텐츠를 요청하였을 경우 이미 작성된 분배 트리가 있는지 확인하고 계산된 대역폭 중 가장 큰 피어의 자식 피어로 스케줄링 한다.

[표 3]는 동적인 P2P 환경에서 피어의 떠남과 같은 오류가 발행하였을 경우 분배 트리를 재 구성함으로써 안정화된 콘텐츠 분배 트리를 재구성하는 알고리즘으로 내용은 다음과 같다.



[표 3] 콘텐츠 분배 트리 재구성 알고리즘

```

/*Initialization*/
1. 콘텐츠 분배 트리 T
2. 콘텐츠 요청 피어 셋  $H = \{h_1, \dots, h_N\}$ , 대역폭이 큰 순으로 정렬
3. k 번째 요청 피어의 대역폭,  $R_k$ 
4. k 번째 요청 피어의 Depth,  $D_k$ 
5. 소스 S의 대역폭  $R_s$ 
6. 다운로드 완료 피어  $D = \{d_1, \dots, d_n\}$ 
7. 고아가 된 피어  $Op$ 의 대역폭  $Op.r$ 
8. 콘텐츠 재정의 요청 피어 셋  $E = H - D - Op = \{e_1, \dots, e_N\}$ 
/*Tree Construct */
9. for i = 0 to N
10.  $h_k = \operatorname{argmax}(R_k | k=1, \dots, i-1) \&\&\operatorname{argmin}(D_k | k=1, \dots, i-1)$ 
11. if  $R_k > Op.r$ 
12. Set  $R_k$  as  $e_i$ 's Parent
13.  $T = T \cup e_i$ 
14. for j = 1 to i
15.  $R_j = R_j / \text{Linknum} + 1$ 

```

1 ~ 8은 콘텐츠 재 분배 알고리즘에 사용된 변수들에 대한 내용으로 T는 콘텐츠 재 분배 트리를 나타낸다. H는 콘텐츠 분배 알고리즘과 같은 콘텐츠 요청 피어로 대역폭에 대해 정렬되어 있고 k 번째 요청한 피어의 대역폭을  $R_k$ , k 번째 요청한 피어의 트리 레벨을  $D_k$ , 콘텐츠를 소유한 소스 피어의 대역폭에 대해  $R_s$ 로 표현한다. 추가로 다운로드 완료된 피어들의 집합을 D로 표현하였고 부모를 잃은 피어와 그의 대역폭에 대한 표현으로 Op와 Op.r로 정의하였다. 또한 분배 트리에 참여한 다른 피어 셋을 재 정의하였다. 이때 콘텐츠 전송이 완료된 피어들에 대해 재 분배 트리에 참여시키지 않으므로 피어의 자원 낭비를 지양하였다.

9 ~ 15는 콘텐츠 재 분배 트리를 작성하는 부분으로 모든 요청 피어

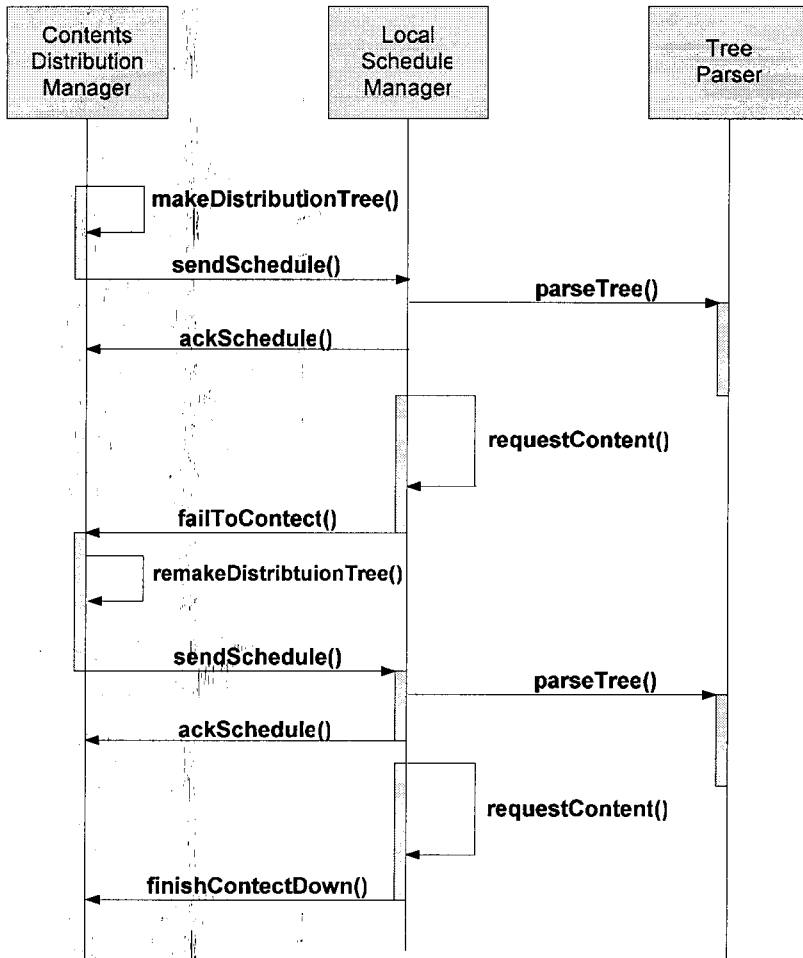
들에 대해 분배 트리와 동일한 방식으로 14~15 는 트리에 참여한 피어의 대역폭을 계산하는 부분이고 자식 피어가 생길 때 마다 그에 해당 하는 수만큼 대역폭 값이 줄어든다. 9. ~ 13 에서는 콘텐츠를 요청한 피어들에 대해 대역폭 계산 값과 피어의 분배 트리 레벨을 고려하여 최적의 부모 피어를 선택한다. 이를 통해 최적화 된 재 분배 트리를 구성할 수 있다. 본 알고리즘에서는 부모를 잃은 피어에 대해 콘텐츠 분배 트리를 재 작성함으로써 분배 트리에 대한 부하를 최소화 할 수 있다.

### 3.1.2 적응적 분배 및 관리 모니터링

콘텐츠 분배 알고리즘을 통해 생성된 분배 트리 정보를 요청한 피어들에게 전달하고 전송 상태 등에 대한 모니터링이 필요하다. 적응적 분배 및 관리 모니터링 서비스는 피어의 전송 상태 모니터링을 통해 콘텐츠가 지속적이고 안정적으로 전송되고 있는지를 확인한다. 콘텐츠 분배 트리에 참여한 피어들은 스케줄링 정보를 받음과 동시에 자신의 전송 상태에 대한 모니터링이 시작된다. 이때 전달 받는 스케줄링에 대한 메타 정보는 [표 4]와 같다.

[표 4] 스케줄링 메타 정보

속성	설명
nodeid	스케줄링에 참여한 피어들에게 부여되는 아이디로 스케줄링 정보 표시에 있어 노드아이디를 통해 제공하는 피어 혹은 제공해줘야 하는 피어의 정보를 알 수 있다.
pname	스케줄링 상에 위치한 피어 이름으로 이정보를 통해 자신이 어느 위치에 스케줄링 되었는지를 파악한다.
cname	제공받을 컨텐츠의 정보를 나타낸다.
status	모니터링을 통해 계속 변화는 값으로 전송 중 오류가 발생하면 이 속성에 오류라고 표시하여 슈퍼 피어에게 전달한다.
from	컨텐츠를 제공하는 피어의 노드 아이디
to	컨텐츠를 제공해줘야 하는 피어의 노드 아이디



[그림 10] 적응적 분배 및 모니터링 흐름도

각 피어들은 자신의 대역폭과 함께 P2P 네트워크에 참여하게 되며 이 정보를 통해 콘텐츠 분배 트리를 구성하게 된다. 위와 같은 스케줄 메타 정보는 XML 형태로 각 피어들에게 전달된다.

[그림 10]의 적응적 분배 및 모니터링에 대한 흐름도에서와 같이 Contents Distribution Manager(CDM)를 통해 스케줄 된 분배 트리 정보를 Local Schedule Manager(LSM)에게 sendSchedule()을 통해 전달하고 자신의 Tree Parser(TP)에 전달 받은 스케줄 정보를 분석하여 자신의 부모 피어와

자식 피어에 대한 정보를 확인하게 된다. 이와 동시에 LSM 은 CDM 에 스케줄 정보를 받았다는 확인 메시지를 `ackSchedul()`을 통해 전달하게 된다. LSM 은 스케줄 정보 분석을 통해 알게 된 부모 피어에게 `requestContent()`를 이용하여 콘텐츠를 요청하고 응답을 기다린다. 만약 일정 시간 부모 피어에게 확인 응답을 받지 못하면 `failToContect()`가 호출되어 CDM 에 자신의 정보와 부모의 정보를 전달하게 된다. CDM 은 즉시 분배 트리를 재 구성하게 되고 새롭게 스케줄 된 정보를 LSM 에게 전달한다. 이 후의 진행은 이전과 동일하며 콘텐츠 전송이 완료되면 `finishContectDown()`을 호출하여 CDM 에게 전송 완료를 알린다. 이러한 일련의 과정으로 인해 콘텐츠를 요청한 피어는 부모의 떠남이 발생하였을지라도 이를 적응적으로 극복하고 재 전송하여 콘텐츠 분배를 완료할 수 있다.

## 제 4 장 안정적인 콘텐츠 분배 관리 시스템 구현

앞서 콘텐츠 분배 안정화를 지원하는 미들웨어의 구조와 필요한 기능에 대해 살펴보았다. 이러한 환경의 적용 가능성을 확인하기 위해 본 논문에서는 프로토타입 PACDS 을 구현하여 이를 검증하였다. PACDS 는 크게 콘텐츠를 요청하고 해당 콘텐츠가 있는지 파악하고 선택하는 서비스와 콘텐츠를 요청한 피어를 가지고 분배 트리를 구성하는 서비스로 나누어 살펴 볼 수 있다. PACDS 의 콘텐츠 관리에 관한 설명은 부록 B 를 참고한다.

### 4.1 콘텐츠 분배 트리 구성 및 전송 서비스

콘텐츠 분배서비스는 콘텐츠를 요청한 피어의 상태를 파악하여 가장 안정적인 분배 스케줄을 작성하게 된다. 전송 스케줄 작성은 앞서 설명한 콘텐츠 검색 서비스가 먼저 제공 된 후 선택된 콘텐츠를 소유한 피어에 대해 피어의 대역폭을 파악하여 트리를 구성한다. 스케줄 된 XML 파일은 요청한 피어에게 JXTA 의 파이프를 통해 전달된다. 또한 스케줄 되는 상태를 표현하기 위해 프리뷰즈 [15]를 사용하였다. 프리뷰즈는 데이터 모델링, 시각화, 그리고 상호작용을 제공한다. 테이블, 그래프, 트리의 구조를 가장 이상적인 모습을 보이도록 해준다. 애니메이션, 다양한 쿼리, 통합적인 검색 그리고 데이터베이스와의 연결성을 제공하는 툴로 자바로 쓰여 졌으며 쉽게 Java Swing 애플리케이션이나 웹 애플릿으로 통합 가능하다. 프리뷰즈는 소프트웨어 개발자들이 Java 프로그래밍 언어를 사용하면서 상호적으로 정보를 시각화하는 애플리케이션을 생성하는데 도움을 주는 확장성이 있는 소프트웨어로 다른 애플리케이션과의 연동이 필요 없는 독립적인 애플리케이션을 가능하게 하고 데이터를 다루고 맵핑하면서 표현하는 것의 과정을 공간적 위치, 크기, 모양, 색을 사용하여 매우 간단하게 보여준다. 간단하게 말해서, 프리뷰즈는 컴포넌트의 생성, 처리, 렌더링을 위해 개발자가 쓰기 편하도록 API 를 제공해주는 툴이다.

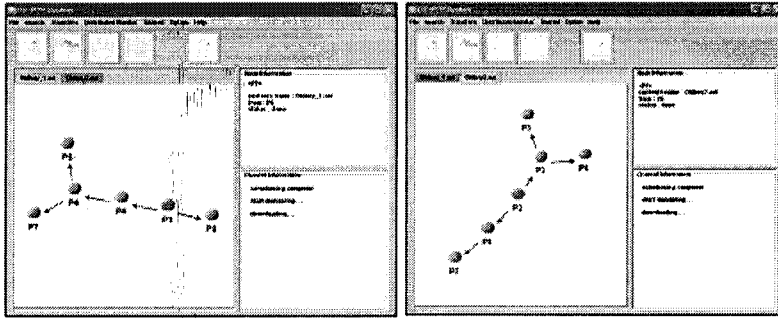
```

<key id="bandwidth" from="edge" attr name="bandwidth" attr type="string" />
<!-- nodes -->
<node id="1">
  <data key="pname">P1</data>
  <data key="cname">oldboy-01.avi</data>
  <data key="status">run</data>
  <data key="from">-1</data>
  <data key="to">P3,P4</data>
  <data key="img">img/super.gif</data>
</node>
<node id="2">
  <data key="pname">P3</data>
  <data key="cname">oldboy-01.avi</data>
  <data key="status">ready</data>
  <data key="from">P1</data>
  <data key="to">-1</data>
  <data key="img">img/ready.gif</data>
</node>
<node id="3">
  <data key="pname">P4</data>
  <data key="cname">oldboy-01.avi</data>
  <data key="status">ready</data>
  <data key="from">P1</data>
  <data key="to">P6</data>
  <data key="img">img/ready.gif</data>
</node>
<node id="4">
  <data key="pname">P6</data>
  <data key="cname">oldboy-01.avi</data>
  <data key="status">ready</data>
  <data key="from">P4</data>
  <data key="to">P5,P7</data>
  <data key="img">img/ready.gif</data>
</node>
<node id="5">
  <data key="pname">P5</data>
  <data key="cname">oldboy-01.avi</data>
  <data key="status">ready</data>
  <data key="from">P6</data>
  <data key="to">-1</data>
  <data key="img">img/ready.gif</data>
</node>
<node id="6">
  <data key="pname">P7</data>
  <data key="cname">oldboy-01.avi</data>
  <data key="status">ready</data>
  <data key="from">P6</data>
  <data key="to">-1</data>
  <data key="img">img/ready.gif</data>
</node>
<!-- edges -->
<edge source="1" target="2">
  <data key="bandwidth">101</data>
</edge>
<edge source="1" target="3">
  <data key="bandwidth">150</data>
</edge>

```

[그림 11] 스케줄링 정보 XML

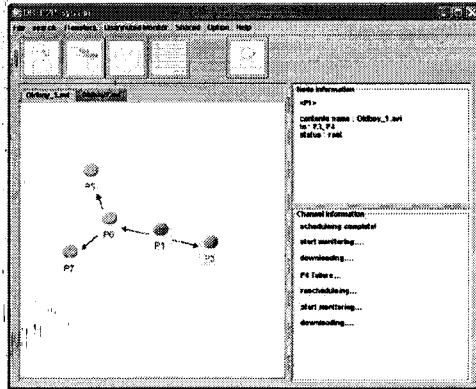
본 논문에서는 [그림 11]과 같이 프리뷰즈 그래프 XML 파일에 스케줄링 된 정보를 하나씩 추가함으로 사용자에게 스케줄링 되고 있는 상태를 동적으로 표현하였고 해당 XML 파일을 피어에게 전달하고 이를 분석하여 실제 스케줄링 된 정보로 콘텐츠 분배가 실행되도록 구현하였다.



[그림 12] 콘텐츠 분배 트리 모니터링

[그림 12]은 콘텐츠를 요청한 다수의 피어에 대한 스케줄링 상태를 모니터링하고 실제 콘텐츠 전송을 시작 하였을 경우 전송에 대한 모니터링을 GUI 형태로 제공한 화면이다. 이를 통해 현재 스케줄링 되고 있는 상태 정보를 그래프와 텍스트 형태로 확인할 수 있다. GUI의 왼쪽은 소스에 대한 스케줄링을 그래프로 나타내는 부분이고 오른쪽 상단은 왼쪽 그래프의 피어 중 하나를 선택하였을 경우 피어에 해당하는 정보를 표시 해주며 오른쪽 하단은 현재 스케줄링 되는 일련의 과정을 텍스트로 나타내 준다. 제공하는 콘텐츠가 하나의 피어에게서 제공되는 것이 아니라면 조각에 대해 각각 스케줄링이 되며 동시에 전송이 진행된다. 소스 피어 하나에 대해 스케줄 그래프가 하나 만들어 지는 것이므로 소스 피어의 개수가 하나 이상일 경우 새로운 탭에 해당 스케줄링 정보를 표현한다.

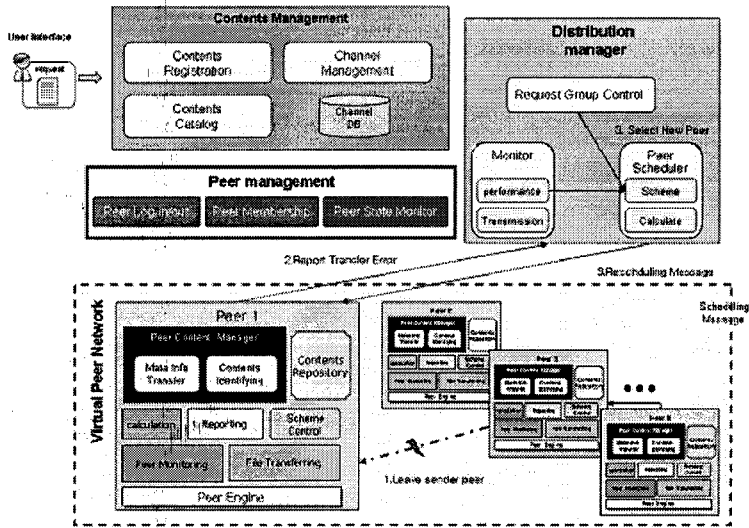




[그림 13] 재분배 트리 구성 모니터링

컨텐츠의 전송이 시작되면 오른쪽 채널 정보에서 현재 진행 상태에 대한 메시지가 출력되며 왼쪽 그래프 피어의 색 변화를 통해 가시적으로 컨텐츠가 전송되고 있다는 것을 확인할 수 있다. 컨텐츠 분배가 진행되면서 피어의 떠남이 감지되면 해당 피어와 자식 피어의 정보를 전달하고 이를 통해 재분배 트리를 작성한다. 이때 스케줄링 되는 상태는 [그림 13]에서 보이는 바와 같이 오른쪽 텍스트 박스에 해당 메시지가 출력되게 되고 왼쪽의 그래프 틀에서 이를 가시적으로 표현한다. 이를 통해 현재 어느 부분에서 전송이 지연되고 있는 지 혹은 전송 중 피어의 접속 끊김과 같은 상태를 파악할 수 있다.

## 제 5 장 실험 및 결과 분석



[그림 14] 분배 트리 재 구성 흐름

[그림 9]와 [그림 14]는 피어가 P2P 오버레이 네트워크에 참여하여 콘텐츠를 요청하고 스케줄링을 통해 콘텐츠를 제공받던 중 피어의 떠남으로 전체 콘텐츠에 대해 콘텐츠를 받을 수 없을 경우 이에 대해 재 스케줄링을 후 콘텐츠를 지속적으로 분배 받는 흐름을 나타낸다.

① 로컬 피어의 사용자는 시스템을 시작하게 되면 기본 P2P 네트워크 환경에 접속하게 되고 원하는 콘텐츠에 대한 콘텐츠 검색을 요청한다.

② 로컬 피어의 검색 요청을 받아 콘텐츠 관리자는 자신의 로컬 저장소를 검색하여 생성된 채널이 있는지 검사한다. 콘텐츠에 대한 채널이 존재한다면 채널에 대한 정보를 로컬 피어에게 전달한다.

③ 로컬 피어는 채널의 디스크립션 정보를 보고 원하는 채널을 선택하여 조인하게 되고 해당 채널 안에 있는 콘텐츠 중 완전한 콘텐츠를 선택하고 동일한 콘텐츠에 대해 새로운 콘텐츠 요청이 있는지 일정 시간 기다린다.

④ 일정 시간 후 콘텐츠를 요청한 피어들이 모이면 피어들의 대역폭

을 확인하여 콘텐츠 트리 구성 알고리즘을 적용한 스케줄 정보를 요청한 피어들에게 전달한다.

⑤ 스케줄링 정보를 받은 로컬 피어는 메시지를 받았다는 사실을 슈퍼 피어에게 알리고 자신의 전송 모니터링을 시작한다. 스케줄링 정보를 분석하여 자신이 콘텐츠를 받아야 하는 소스 피어에게 콘텐츠를 요청한다.

⑥ 콘텐츠 전송에 대한 상태가 변화될 때 마다 해당 정보를 슈퍼 피어에게 알린다. 이때 콘텐츠 전송 중 해당 부모 피어가 네트워크를 떠났을 경우 이에 대해 자식 피어는 이를 감지하고 슈퍼 피어에게 알린다.

⑦ 슈퍼 피어는 해당 정보를 보고 스케줄링 된 트리를 재 구성하여 부모를 잃은 피어에게 재 스케줄링 된 정보를 전달하고 자신은 스케줄링 정보를 업데이트 한다.

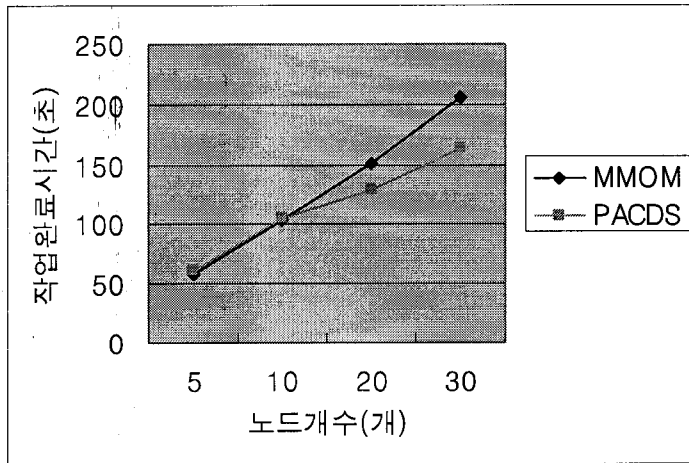
## 5.1 실험 결과

이 장에서는 4.장의 분배 트리 알고리즘을 토대로 실제 P2P 환경에서 콘텐츠를 주고 받는 시간을 측정하고, 그 결과를 분석하였다. 실험환경으로 자바 버전 1.5 이상의 리눅스 노드 30 개에서 걸쳐 실시하였다. 각 노드는 512M 이상의 메모리와 1.8Ghz 듀얼 코어 CPU를 제공한다. 또한 실험에 참여한 피어의 대역폭은 500 ~ 800Kbps 사이의 값을 갖는다. 실험 시간은 시스템의 성능이나 실험 시의 네트워크 상황을 감안한다.

노드는 무작위로 JXTA 네트워크에 조인하게 되며 JXTA의 랭데부 피어를 슈퍼 피어로 설정하여 실험한다. PACDS 네트워크 상의 각 피어는 125M의 콘텐츠를 요청하게 되며 전송 상태를 체크한다. 요청 피어의 개수를 늘려가며 실험을 진행하였다. 실험은 MMOM(Max-Min Overlay Multicast) 시스템과 본 논문의 PACDS의 트리 구성 방식 및 콘텐츠 분배 시간을 비교한다. 이를 위해 MMOM 시스템의 트리 구성 및 콘텐츠 분배를 추가로 구현하였다.

### 5.1.1 콘텐츠 전송 완료 시간 측정 및 분석

본 절에서는 콘텐츠 노드 변화에 따른 콘텐츠 전송 완료 시간을 측정하고 이를 분석하였다. 이 실험은 콘텐츠를 요청하는 피어를 5 개, 10 개, 20 개, 30 개로 늘려가면서 모든 요청 피어가 콘텐츠를 전송 받은 시간을 측정하였다.



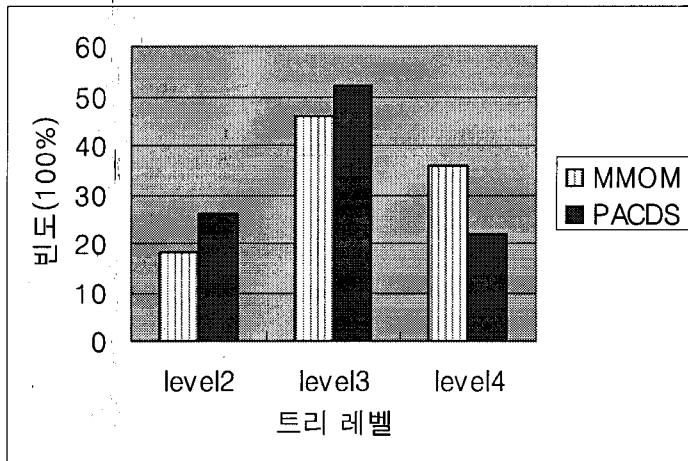
[그림 15] 노드 개수의 변화에 따른 작업 완료 시간 측정

[그림 15]는 콘텐츠 요청 피어의 개수를 늘려가며 콘텐츠를 요청했을 시의 콘텐츠 전송 완료 시간을 MMOM의 알고리즘을 적용했을 경우와 PACDS의 콘텐츠 분배 알고리즘을 적용했을 경우의 시간을 비교한 그래프이다. 피어의 요청 순서와 대역폭을 일정하게 유지하고 피어의 떠남이 없을 경우를 측정한 것으로 콘텐츠를 요청하는 피어의 개수가 적었을 경우는 두 시스템 모두 콘텐츠 전송 시간의 차이가 없었으나 피어의 개수가 많아짐에 따라 PACDS에서 작업완료시간이 줄어드는 양상을 띠고 있다. 즉, 요청 피어의 개수가 5개인 경우 MMOM의 알고리즘을 적용했을 경우 평균 58.2초가 소요되는 반면 PACDS에서는 평균 61.07초로 콘텐츠를 요청하는 노드가 적었을 경우는 좀더 시간이 소요되나 피어의 개수가 늘어날수록 PACDS에서 콘텐츠 전송 완료 시간이 단축되는 것을

확인하였다.

### 5.1.2 콘텐츠 분배 트리 레벨 측정 및 분석

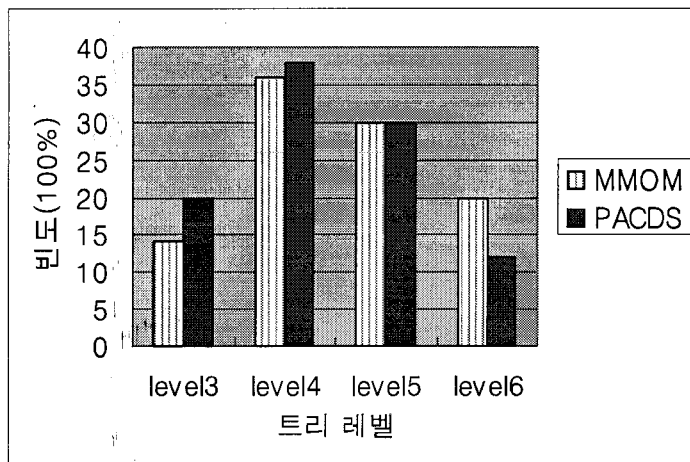
본 절에서는 노드 개수 별 콘텐츠 분배 트리 작성 시 트리 레벨의 발생 빈도를 측정하고 이를 분석하였다. 이 실험은 콘텐츠를 요청하는 피어를 5 개, 10 개, 20 개, 30 개로 늘려가면서 생성된 분배 트리의 레벨에 대한 빈도를 측정하였다. 이때 트리 구성 중 새로운 피어의 참여는 고려하지 않고 이전에 요청한 피어만을 이용하여 트리를 구성하는 것을 가정으로 한다.



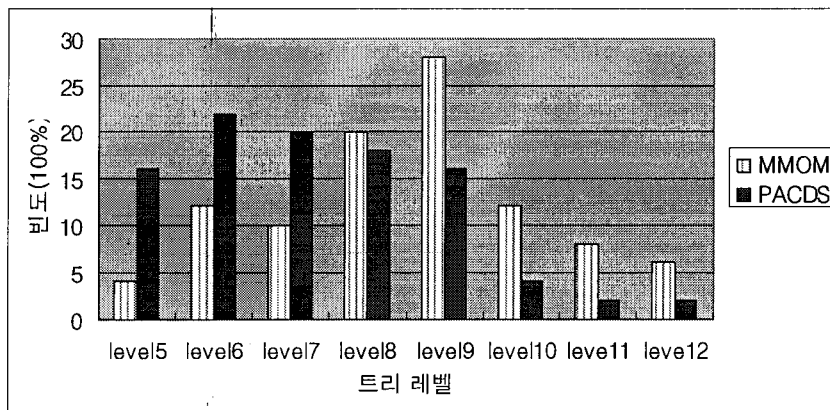
[그림 16] 5 개의 노드를 적용한 분배 트리 레벨 구성 빈도

[그림 16]은 콘텐츠를 요청한 피어가 5 개일 경우 생성되는 분배 트리의 빈도를 측정한 것으로 분배 트리 구성 시 요구되는 대역폭은 무작위로 선택 된다. 레벨은 분배 트리의 깊이를 나타내는 단위로 레벨 0 은 소스 피어, 레벨 1 은 소스와 연결된 피어를 나타낸다. MMOM 시스템의 경우 트리 구성 시 피어의 대역폭 만을 고려하여 트리를 구성한다. 이에 비해 PACDS 는 피어의 대역폭과 피어의 레벨을 고려하여 트리를 구성한다. 이러한 MMOM 의 알고리즘과 PACDS 의 알고리즘을 적용한 분배 트

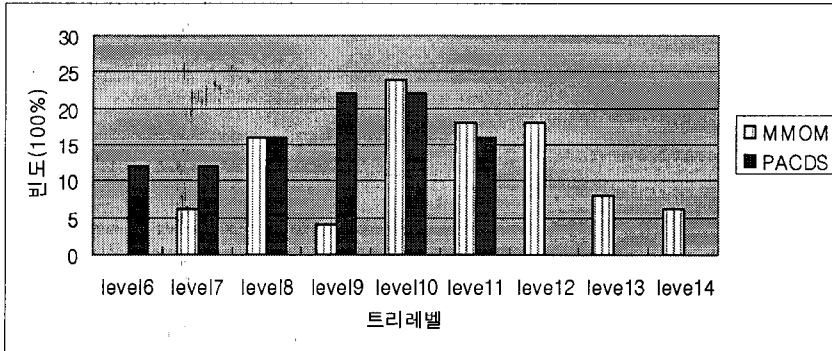
리의 레벨 빈도를 측정한 결과는 다음과 같다. MMOM 의 알고리즘과 PACDS 의 알고리즘 모두 레벨 3 까지 생성된 트리가 가장 많이 생성되었다. 또한 레벨 2, 레벨 3, 레벨 4 까지 적용된 분배 트리가 각각 생성된 것을 확인할 수 있다. 그러나 세부적으로 살펴보면 PACDS 에 의해 레벨이 낮은 분배 트리 생성 빈도가 높음을 확인할 수 있다.



[그림 17] 10 개의 노드를 적용한 분배 트리 레벨 구성 빈도



[그림 18] 20 개의 노드를 적용한 분배 트리 레벨 구성 빈도

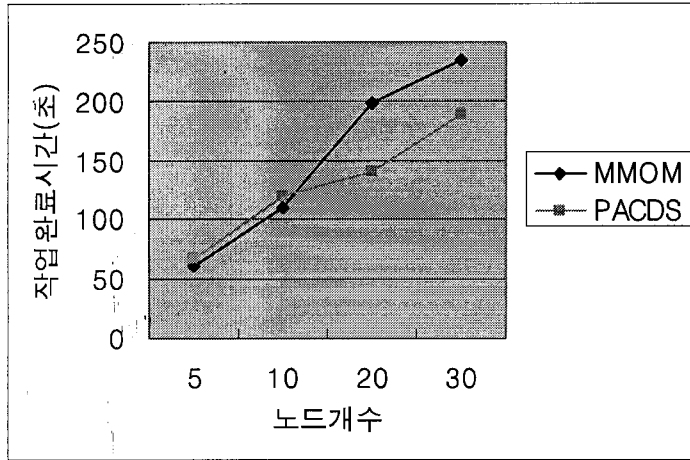


[그림 19] 30 개의 노드를 적용한 분배 트리 레벨 구성 빈도

[그림 17], [그림 18], [그림 19]은 각각 노드의 개수를 10 개, 20 개, 30 개로 늘리며 측정한 결과의 그래프로 노드의 개수가 늘어나면서 MMOM의 알고리즘을 적용했을 경우의 분배 트리 보다 PACDS의 분배 트리의 레벨이 낮음을 확인할 수 있다. 노드가 30 개일 경우 PACDS는 레벨 5 부터 레벨 10 사이의 트리가 생성되는 반면 MMOM의 시스템은 레벨 7부터 레벨 14 사이의 트리가 생성된다. 이는 분배 트리 구성 시 트리의 레벨을 고려함으로써 좀 더 안정적인 분배 트리를 생성하기 때문이며 효과적인 콘텐츠 분배에 용이하다.

### 5.1.3 분배 트리 재구성 시 콘텐츠 완료 시간 측정 및 분석

본 절에서는 콘텐츠 노드 변화에 따른 콘텐츠 재 전송 완료 시간을 측정하고 이를 분석하였다. 이 실험은 피어를 5 개, 10 개, 20 개, 30 개로 늘려가면서 콘텐츠를 요청하였을 경우 무작위로 하나의 피어를 오버레이 네트워크에 떠났을 경우 분배 트리를 재 구성하고 콘텐츠 전송을 완료하는 시간을 측정하였다. MMOM의 경우 부모 상위의 피어를 찾아 콘텐츠를 재 전송 하는 데 반해 PACDS는 성능 좋은 부모 피어를 찾아 콘텐츠를 재 전송 할 수 있도록 한다. 이때 콘텐츠 분배가 시작되면 새로운 노드가 콘텐츠를 요청하는 것에 대한 고려를 배제하였다.



[그림 20] 피어 떠남 시 노드 개수의 변화에 따른 작업 완료 시간 측정

[그림 20]는 기존의 트리 구성 후 피어의 떠남이 발생하였을 경우 재 구성하여 컨텐츠 전송을 완료하는 평균 시간을 나타낸 그래프이다. 실제 노드 환경에서 실험을 적용한 것으로 네트워크 상황에 따라 작업 완료 시간의 편차가 발생하였다. 노드의 개수가 20 개 이상일 경우 (+)11.25 초의 편차가 있음을 확인하였다. 실험 결과, 노드의 개수가 작았을 경우는 MMOM 시스템과 PACDS 시스템 간의 작업 완료 시간의 차이가 근소한 것을 확인할 수 있다. 이는 피어의 개수가 적을 경우 새로운 노드를 찾아 주는 것 보다 상위 피어에 바로 컨텐츠를 요청하는 것이 효율적일 수 있다. 그러나 노드의 개수가 늘어남에 따라 작업 완료 시간이 MMOM 시스템에 비해 줄어드는 모습을 볼 수 있다. 이는 피어의 개수가 많은 경우 상위 피어를 찾고 실제 요청하는데 소요되는 시간과 상위 피어의 상황을 고려하지 않기 때문에 PACDS 에서 적용한 성능 좋은 피어를 찾아서 바로 컨텐츠를 요청하게 함으로 실제 컨텐츠 완료 시간이 MMOM 에 비해 줄어든 것이다. 결과적으로, 피어의 개수가 많아 질수록 MMOM 시스템 보다 PACDS 가 효율적이고 안정적으로 컨텐츠 전송을 완료함을 확인할 수 있다.



## 제 6 장 결 론

P2P 기술은 콘텐츠 공유 응용프로그램에 긴밀히 유용되나 기존의 콘텐츠 공유 응용프로그램은 콘텐츠를 제공한 피어의 성능이나 대역폭에 따라 데이터 전송 속도가 급격히 떨어지는 단점이 있었다. 이러한 유동적인 P2P 환경에서 안정적으로 콘텐츠를 분배하기 위해서는 피어와 콘텐츠를 관리하고 최적의 전송 방법을 지원하는 미들웨어가 필수적으로 필요하다. 이에 따라 본 논문에서는 P2P 기반 분산 환경에서 콘텐츠 이동의 안정화를 지원하는 콘텐츠 선택 방법과 분배 방법을 제시하였고 피어 상태 감지를 통한 분배 트리 재구성 기법을 제안하였다. 또한 P2P 표준인 JXTA 를 이용하여 프로토타입을 구현하고 실험을 통해 검증했다.

P2P 네트워크를 구성하는 피어의 상태 등을 관리하는 피어 관리, 콘텐츠의 효율적인 분배 관리와 오류에 대한 적응적 재 분배 관리 기능으로 구성되는 서비스 미들웨어는 P2P 네트워크에 참여하여 원하는 콘텐츠를 검색하고 반환된 콘텐츠 채널에 참여한 후 다수의 피어와 안정적으로 콘텐츠를 제공받을 수 있다. 콘텐츠의 분배 안정화를 위해 동적으로 분배 트리를 생성하고 실제 피어 사이에 적용함으로써 전송의 효율성을 높였으며 피어 떠남을 감지한 후 적응적으로 분배 트리를 재 구성하여 콘텐츠 전송의 지속성과 안정성을 제공하였다. 따라서 본 논문에서 제시된 적응적 콘텐츠 분배 기법은 향후 비슷한 연구의 선례가 될 것이다.

차후에는 다양한 기기들을 포함하여 동적으로 구성된 네트워크 구조 아래 콘텐츠의 안정적인 분배를 지원하기 위한 폭넓은 방안을 제시하고 콘텐츠 분배뿐 아니라 서비스 관리에 있어 일관된 서비스 이용 방법을 제공해야 할 것이다. 또한 특정 P2P 응용에 국한된 프로토콜 대신 표준화된 프로토콜을 정의하여 이질적인 자원들을 일관성 있게 사용하고 관리할 수 있는 자원 관리 서비스가 필요하며 무선 환경과 결합하여 피어의 이동성 및 계속성, 유연성에 대한 QoS(Quality of Service) 지원 방법을 제시하여야 할 것이다.

## 부 록

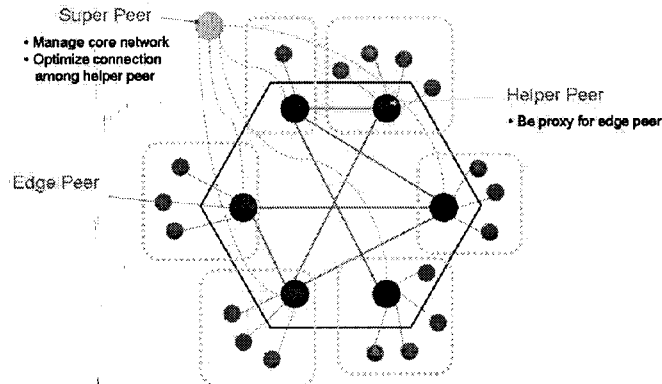
### A. 웹 서비스 기술을 이용한 콘텐츠 관리

안정적인 콘텐츠 분배를 위해서는 제공한 콘텐츠에 대한 관리가 필수적이다. 또한 P2P 환경에서 다양한 디바이스가 참여할 수 있는 환경으로 나아가는 현 시점에서 개방형 표준을 고려한 콘텐츠 관리 부분을 고려하게 되었다. 또한 웹 서비스 기술을 적용함으로써 P2P의 단점인 피어 검색에 대한 부하를 웹 서비스 개념과 접목함으로써 나아가 분산된 디렉터리 서비스를 제공할 수 있는 발판이 될 것으로 사료된다.

콘텐츠와 피어에 대한 관리는 P2P 네트워크를 구성하고 생성된 채널 그룹별로 관리하는 기능을 제공한다. 즉, 사용자는 피어 관리를 통해 P2P 네트워크에 참여하고 자신의 상태를 P2P 네트워크를 통해 다른 피어에게 알린다. 콘텐츠 관리는 피어가 소유하고 있는 콘텐츠를 관리하고 콘텐츠 요청에 대해 최적의 콘텐츠를 선택한다. 이는 네트워크에 참여하는 피어의 수가 방대해짐에 따라 빠른 검색 서비스를 제공할 수 있다. 이러한 기능을 웹 서비스 기술을 적용함으로써 이 기종의 환경에서도 모두 콘텐츠를 검색하고 제공받을 수 있는 시스템을 구성하였다.

## 1. P2P 오버레이 네트워크 구성 및 피어 정의

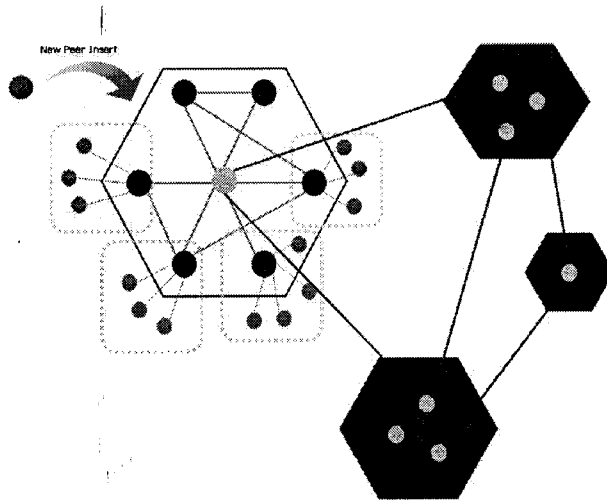
P2P 오버레이 네트워크 상에는 [그림 21]와 같이 세가지 종류의 피어가 존재하게 된다.



[그림 21] 슈퍼 피어, 헬퍼 피어, 에지 피어

슈퍼 피어 (Super peer)는 콘텐츠 검색 웹 서비스를 제공하는 피어로 P2P 네트워크 상에 존재하는 콘텐츠에 대한 정보를 수집하고 검색할 수 있는 환경을 제공한다. 즉, 자신만의 콘텐츠 저장소를 보유하고 있어 피어가 제공한 다양한 콘텐츠 광고에 대한 정보를 관리한다. 헬퍼 피어 (Helper peer)는 실제 콘텐츠를 소유한 피어로 자신의 콘텐츠 정보를 메타 데이터로 표현하여 슈퍼 피어에 등록하여 다른 피어가 자신의 콘텐츠에 접근할 수 있도록 지원한다. 콘텐츠들은 피어 콘텐츠 관리자를 통해 저장된 콘텐츠를 식별하고 각각의 콘텐츠에 대한 메타 데이터를 정의하게 된다. 에지 피어 (Edge peer)는 콘텐츠를 요청하는 피어로 원하는 콘텐츠를 슈퍼 피어가 제공하는 콘텐츠 검색 웹 서비스를 이용하여 콘텐츠의 위치를 파악할 수 있다. 네트워크에 참여한 모든 피어는 헬퍼 피어, 에지 피어가 될 수 있다.

## 2. 콘텐츠 등록 및 채널 카탈로그 검색 서비스



[그림 22] P2P 네트워크 참여 및 채널 카탈로그 검색

피어는 P2P 네트워크에 자신을 광고하며 참여하고 이때 자신의 콘텐츠를 등록하게 된다. 이때 선택 되는 슈퍼 피어는 자신과 가까운 슈퍼 피어를 선택한다. 각각의 슈퍼 피어는 자신의 이웃 슈퍼 피어에 대한 정보를 가지고 있어 콘텐츠 등록에 있어 채널이 존재한다면 해당 채널 그룹을 반환한다.

콘텐츠는 일련의 메타 정보에 의해 표현되며 그에 대한 내용은 [표 5] 과 같다. `cname`, `cmsid`, `length`, `pid` 는 JXTA[10]의 CMS 서비스의 콘텐츠 정의를 빌어 표현하였다. 일련의 과정은 피어 어플리케이션 상에 존재하는 피어 콘텐츠 관리자를 통해 메타 데이터를 생성하며 이를 슈퍼 피어에게 제공함으로써 유일하게 식별할 수 있게 된다. 실제 콘텐츠는 제공은 헬퍼 피어를 통해 이뤄지지만 콘텐츠 및 소유주에 대한 정보는 슈퍼 피어에 저장되게 된다. 또한 콘텐츠에 변화가 생길 때 마다 이를 감지하고 새롭게 업데이트된 정보를 전달한다.

[표 5] 콘텐츠에 대한 메타 정보 표현

속성	설명
cname	콘텐츠 이름으로 파일명
cmsid	피어 콘텐츠 매니저를 이용해 적용된 콘텐츠의 고유한 아이디
cid	동일한 콘텐츠인지를 식별하기 위한 콘텐츠 하나에 대한 아이디
length	콘텐츠가 가지고 있는 파일 사이즈
type	콘텐츠의 타입 예)avi, mp3, zip
part	조각난 콘텐츠에 대해 몇 번째 콘텐츠인지를 확인하기 위한 속성
pid	콘텐츠를 소유하고 있는 피어의 아이디
pname	콘텐츠를 소유하고 있는 피어가 정한 이름

슈퍼 피어는 콘텐츠 이름이 바뀌더라도 콘텐츠 아이디를 통해 식별 가능하며 분배된 콘텐츠의 위치 정보를 확인 할 수 없으므로 콘텐츠에 대한 익명성을 보장할 수 있다. P2P 오버레이 네트워크 환경에서는 서비스에 대한 공유도 가능하며 이는 웹 서비스 측면에서도 동일한 개념으로 인식될 수 있다. 즉, 피어는 자신이 가지고 있는 서비스에 대해 광고할 수 있고 서비스를 사용 하고자 하는 피어는 서비스를 제공하는 피어가 광고했던 내용을 기반으로 서비스를 요청할 수 있다. 이와 동일하게 웹 서비스는 WSDL 을 클라이언트에게 제공함으로써 서비스를 이용할 수 있도록 지원한다. 이러한 개념을 기반으로 본 연구에서는 콘텐츠를 검색하는 웹 서비스를 생성하고 생성된 검색 서비스에 대한 광고에 WSDL 을 첨가함으로써 콘텐츠에 대한 검색 웹 서비스를 지원한다.

## B. 콘텐츠 관리 서비스

콘텐츠 분배 안정화를 지원하는 미들웨어가 적용된 P2P 환경에서 사용자는 자신이 원하는 콘텐츠를 빠른 시간에 정확하게 받기를 원한다. 이를 충족시키기 위해 미들웨어는 사용자의 요청과 실제 검색되는 콘텐츠에 대한 정보를 제공하기 위한 서비스가 마련되어야 한다. 이는 콘텐츠의 검색과 완전한 콘텐츠를 제공하는 상태 정보를 제시해준다. 뿐만 아니라 콘텐츠 요청 서비스는 실시간으로 작업에 대한 정보를 수집하고, 현재 어떤 작업이 수행되고 있는지에 대한 정보도 보여준다. 콘텐츠를 검색하기 위해서는 콘텐츠를 제공하는 피어가 존재해야 하며 콘텐츠를 제공하는 피어는 콘텐츠에 대한 정보를 XML 형식으로 작성하여 콘텐츠 관리 서비스에 등록하게 된다. 콘텐츠 검색에 있어 요청한 피어에게 최적의 콘텐츠를 찾아주기 위해 콘텐츠의 완전함을 확인하고 그렇지 않을 경우 완전한 콘텐츠를 제공하기 위해 조각을 소유하고 있는 피어는 조합하여 최적의 피어 셋을 생성한다. 콘텐츠의 메타 정보는 JXTA 의 CMS 서비스에 속성을 첨가하여 구현하였다. [그림 23]는 콘텐츠의 메타 정보를 이용하여 실제 콘텐츠를 표현한 XML 파일로 콘텐츠 XML 파일을 분석하여 관리한다.

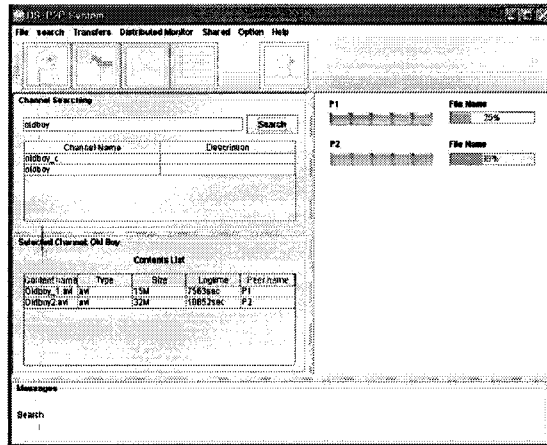
```
<?xml version="1.0" encoding="UTF-8" ?>
- <provide>
  <cname>OldBoy-03.avi</cname>
  <cmsid>md5:619e858596337f3bf7ed2558c0e4b8c8</cmsid>
  <cid>MM100001</cid>
  <length>200000</length>
  <ctype>.zip</ctype>
  <part>00100</part>
  <pid>urn:jxta:uid-
    59616261646162614E50472050325030FF8118D0FED41D4AEB8FA218307D53D03</pid>
  <pname>P1</pname>
</provide>
```

[그림 23] 콘텐츠 메타 정보 XML

본 논문에서 구현한 PACDS 는 P2P 네트워크 환경에서 콘텐츠 요청 작업에 초점을 두고 이를 해결하기 위해 실시간으로 작업에 대한 메시지를 주고 받는다. 전송 모니터링 서비스를 통해 콘텐츠에 대한 검색부터 스케줄링 후 전송을 완료 될 때까지의 상태를 보인다. 콘텐츠에 대한 ID

는 유일하기 때문에 해당 콘텐츠의 전송 변화에 대해 동적으로 콘텐츠 ID 와 로그파일에 해당 작업에 대한 정보를 적어 이를 매칭시키는 작업을 통해 로그파일의 변화를 동적으로 탐지한다. 즉, 로그파일의 사이즈가 변화할 때마다 작업의 상태 변화를 가정하여 이벤트를 발생시킴으로써 실시간으로 읽어 들이고 로그파일에 쓰여진 작업의 상태를 UI 에 반영한다. 사용자의 GUI 에서 가시적인 효과를 위해 서로 다른 색의 형태로 해당 작업의 상태를 보여주고 작업이 진행됨에 따라 상태의 변화를 파악할 수 있다. 다운로드가 완료되면 사용자는 다운이 된 해당 디렉터리로 이동하여 다운이 완료된 작업 결과를 확인할 수 있다.

[그림 24]는 이러한 일련의 과정을 수행할 수 있는 사용자 GUI 로 사용자가 원하는 콘텐츠를 찾기 위해 왼쪽 상단의 채널 검색 창에서 찾기 원하는 콘텐츠의 이름을 입력하고 search 버튼을 누르면 콘텐츠 검색 웹 서비스를 호출하게 되고 콘텐츠를 소유하고 있는 채널을 검색하여 그 결과를 하단의 테이블에 보인다. 사용자는 자신이 원하는 콘텐츠가 속해 있는 채널을 선택할 수 있으며 이때 하단의 selected Channel 부분을 통해 현재 내가 속한 채널의 정보를 확인할 수 있다. 사용자는 전송될 콘텐츠 정보를 오른쪽 그림을 통해 확인할 수 있다. 즉, 현재 선택된 채널은 완전한 콘텐츠를 가지고 있는 피어는 존재하지 않으며 조각을 가지고 있는 콘텐츠의 조합을 통해 완전한 하나의 콘텐츠를 제공할 수 있다는 것을 확인할 수 있다. 또한 하위의 Message Box 를 통해 현재 어떠한 과정을 거쳐 P1 과 P2 가 선택 되는지 일련의 과정에 대해 텍스트 형태로 사용자에게 제공된다. 이렇게 선택된 콘텐츠에 대해 현재 제공하는 콘텐츠 중 어떠한 피어가 얼마큼 제공하는 지에 대한 정보를 Progress Bar 형태로 표현한다.



[그림 24] 컨텐츠 검색 화면



## 참 고 문 헌

- [1] Stephanos Androutsellis-Theotokis and Diomidis Spinellis. "A survey of peer-to-peer content distribution technologies". ACM Computing Surveys, December 2004
- [2] Rüdiger Schollmeier "Neue Wege in der Musikindustrie P2P mit Skype, Kazza, ... Oder: Was kommt nach Napster & Co?" Vortrag im Rahmen der Vortragsreihe des VDE/VDI-Arbeitskreis Informationstechnik October 21st 2004, München
- [3] I.Stoca, R.Morris, D. Karger, F.Kaashoer and H. Balakrishnan, "Chord: A scalable peer-to-per lookup service for Internet application", in Proc. ACM SIGCOMM 2001
- [4] A. Rowstron and P. Druschel, "Pastry: Scalable, decentralized object location and routing for large-scale peer-to-peer systems". IFIP/ACM International Conference on Distributed Systems Platforms (Middleware), Heidelberg, Germany, pages 329-350, November, 2001.
- [5] Ben Y. Zhao, Ling Huang, Jeremy Stribling, Sean C. Rhea, Anthony D. Joseph, and John Kubiatowicz "Tapestry: A Resilient Global-scale Overlay for Service Deployment "IEEE Journal on Selected Areas in Communications, January 2004, Vol. 22, No. 1.
- [6] Sylvia Ratnasamy,Paul Francis, Mark Handley ,Richard Karp ,Scott Schenker "A scalable content-addressable network" Applications, Technologies, Architectures, and Protocols for Computer Communication archive Proceedings of the 2001 conference on Applications, technologies, architectures, and protocols for computer communications

- [7] U. Wieder, M. Dahlin, "A Simple Fault tolerant distributed Hash Table", IPTPS 2003, Berkeley CA, February 2003
- [8] Ozgur D. Sahin, Cagdas E. Gerede, Divyakant Agrawal, Amr El Abbadi, Oscar Ibarra, Jianwen Su: SPiDeR: P2P-Based Web Service Discovery. The Third International Conference on Service Oriented Computing (ICSOC 2005), Netherlands, 2005.
- [9] Pei Zheng, Chen Wang Xejia: A Scalable Channel-Based Multi-source Content Distribution System. WCW 2004: 106-115
- [10] JXTA v2.0 Protocols Specification, Sun Microsystems Inc., <http://spec.jxta.org/nonav/v1.0/docbook/JXTAProtocols.pdf>
- [11] A. Striegel, G. Manimaran "A Survey of QoS Multicasting Issues " IEEE Communications Magazine, 40(6):82--87, June 2002.
- [12] Ming Zeng, Hongliang Yu, Feng Yang, Weimin Zheng " Dynamically Optimize Multicast Tree Structure" In Proceedings of the 2006 Workshop on NGI and P2P Systems (INPS 06, in Conjunction with GCC 2006), Changsha, China, October 2006
- [13] Mohamed Hefeeda, Ahsan Habib, Boyan Botev, Dongyan Xu, Bharat Bhargava "PROMISE: Peer-to-Peer Media Streaming Using CollectCast" MM'03 2003
- [14] Yi Cui, Yuan Xue and Klara Nahrstedt, Max-min Overlay Multicast: Rate Allocation and Tree Construction, in Proc. of IEEE International

Workshop on QoS (IwQoS '04).

[15] 프리퓨즈 homepage <http://www.prefuse.org/>

## ABSTRACT

A Peer to Peer (P2P) network is a heterogeneous network with various types of devices and is used for connecting nodes. A participant in a P2P network can share content files, computing resource, and data storage. But current P2P applications can not support resource search dynamically in a heterogeneous network because they have not standardized protocols for P2P. And they have a weak point of decreasing content transmission rate because they do not consider performance or bandwidth state of peers. Therefore, a P2P distribution environment needs a middleware that supports stable and complete contents distribution in dynamically moving environment.

In this paper, I propose the middleware architecture to support an adaptive contents distribution service and a content management service using open standard techniques. I suggest bandwidth-aware tree-based distribution algorithm during tree construction. Moreover, adaptive redistribution algorithm is supplied whenever detects tree separation is detected. P2P based Contents Distribution System (PACDS) is implemented using JXTA protocol which is a set of XML based protocols that allow any devices connected to a network to exchange messages. Finally, I verify the performance of PACDS through experiments over P2P environment.