

P2P 환경에서의 콘텐츠 분배 시스템 동향

김 윤 화*

송 은 혜**

◆ 목 차 ◆

1. 서 론
2. P2P 환경에서의 콘텐츠 분배 관리 기술
3. P2P와 결합된 웹 서비스와 스트리밍 서비스
4. 웹 서비스를 활용한 P2P 기반 콘텐츠 분배에 대한 제언
5. 결 론

1. 서 론

인터넷은 3가지 기본적인 요소인 정보, 대역폭, 컴퓨팅 자원을 기반으로 활성화 되어 왔다. 현대에 접어들면서 통신 기술의 발달로 초고속 네트워크 망이 형성되어 다수의 사용자에게 대용량의 정보를 빠른 속도로 제공하게 되었다. 그러나 데이터 정보의 양이 방대해짐에 따라 기존의 서버-클라이언트 컴퓨팅 모델 기반에서는 매시간 생성되는 수 많은 정보를 분류하고 찾아 낼 수 없고 서버의 부하와 CPU, 스토리지 부족과 같은 문제가 발생하게 되었다. 또한 개인 PC의 기능이 서버와 맞먹는 수준으로 발전하여 서버-클라이언트 구조의 한계를 클라이언트 상호간 분산·협력을 통해 해결하려는 욕구로 인해 Peer-to-Peer(P2P)가 등장하게 되었다. P2P는 소수의 서버에 집중하기 보다는 망 구성에 참여하는 기계들의 계산과 대역폭 성능에 의존하여 구성되는 통신망이다. 통신망에 참여한 피어(Peer)는 자신의 자원(스토리지, 콘텐츠, 컴퓨팅 자원)등을 DNS를 이용하지 않고 다른 피어와 통신하고 자원을 공유할 수 있다. 피어는 서버와 클라이언트 역할을 모두 수행할 수 있어 개인이 소유하고 있는 영화 파일이나 MP3 파일 등 수많은 파일들을 P2P응용 어플리케이션을 통해 자유롭게 주고 받을 수 있게 되었다.

었다.

또한 P2P 네트워크에 참여하는 디바이스의 종류도 점차 다양해지고 이러한 디바이스를 포함한 네트워크 구조도 다양해짐에 따라 이종의 네트워크에서 자원을 효과적으로 공유하고 원하는 자원을 쉽게 찾을 수 있는 규약(protocol)이 필요하게 되었다. 현재 사용되고 있는 P2P 응용프로그램들은 자기 정의된 규약을 사용하고 있어 이러한 요구 조건을 충족시키기 어려웠다. 이러한 요구 조건을 충족시키는 노력으로 썬 마이크로시스템즈에서 개발한 JXTA를 들 수 있다. JXTA는 P2P를 가능하게 하는 작은 일반 프로토콜 셋을 정의함으로써 P2P 응용프로그램 사이에 특정 언어나 플랫폼에 상관없이 기존의 모든 플랫폼과 응용 프로그램에 접속할 수 있다. 기본적으로 JXTA 프로토콜을 사용하며 PC, PDA, 서버 등 다양한 디바이스간의 통신을 메시지 형태로 가능하게 하여 컴퓨팅 자원을 활용할 수 있는 자원 분배한 플랫폼이다.

한편 유동적인 P2P 네트워크 환경에서 영화와 같은 멀티미디어 데이터의 대형화로 인해 효율적이고 안정적인 콘텐츠 제공을 위한 다양한 콘텐츠 분배 기술이 필요하게 되었다. 이들 기술은 네트워크를 구성하고 있는 피어 시스템간의 거리를 측정하고 데이터의 유효성과 유사성을 검사하며 무무선 네트워크에 존재하는 트래픽 양과 특정 시스템에 집중되는 부하를 분산시키거나 감소시킬 수 있어야 한다. 대표적인 콘텐츠 분배 기술로 실제 서비스의 내용은 피어 시스

* 숙명여자대학교 부교수

** 숙명여자대학교 컴퓨터과학과 석사과정

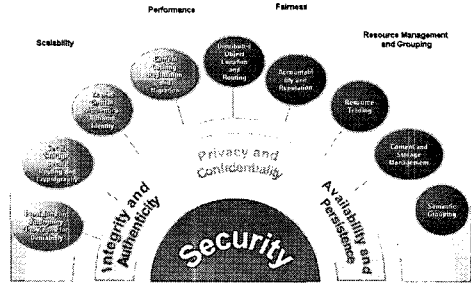
템에 나눠 저장하고 그에 대한 효율적인 탐색 방법을 제공하는 기술과 멀티미디어 데이터에 대한 수요가 증가함에 따라 콘텐츠를 제공하는 주체가 수시로 변하는 환경에서 효과적으로 콘텐츠를 분배하는 기술을 들 수 있다. 콘텐츠 검색에 있어 메시지 전달 때문에 발생하는 메시지 수가 많아 검색 비용과 네트워크의 부하 증가를 해결하기 위해 해쉬 테이블(DHT: Distributed Hash Table)을 이용한 다양한 검색 방법이 제시되었다. 이와 같은 검색 방법을 이용하여 네트워크에 산재되어 있는 피어 서비스를 빠르게 검색할 수 있게 되었다. 앞으로 이 기종의 네트워크 환경에서의 서비스 검색 제공을 위해 웹 서비스 표준과 결합하여 통합된 기술 표준 형태로 서비스를 생성하고 지원하는 방안이 연구 중에 있다. 또한 멀티미디어 데이터를 이용한 미디어스트리밍 서비스를 이용하여 콘텐츠를 요청한 피어의 대역폭과 성능을 고려하거나, 피어의 관심 별로 관리하여 다수의 피어들에게 효과적으로 콘텐츠를 분배하는 방법에 대한 연구가 진행 중이다.

본 고에서는 P2P 환경에서 콘텐츠 분배와 관련된 여러 인접 기술들과 이를 이용한 시스템들에 대해 기술하였고, P2P와 웹 서비스, 미디어 스트리밍 서비스와 결합된 다양한 시스템에 대해 소개하였다. 또한 P2P에서 피어 사이의 안정적인 콘텐츠 분배에 대해 제안하였다.

2. P2P 환경에서의 콘텐츠 분배 관리 기술

P2P 환경에서 각각의 피어는 스토리지와 같은 리소스를 네트워크를 통해 제공한다. 이러한 방법은 적은 비용으로 데이터를 공유할 수 있고, 대표 서버가 필요하지 않으며 네트워크 유지에 대한 비용이 적어 진다는 장점이 있으나 단일 피어에 대한 콘텐츠 제공 실패 위험의 증가로 QoS를 보장할 수 없고, topology를 알지 못한다는 단점이 있다. 이를 위해 그림 1에서와 같이 P2P에서 콘텐츠 분배에 대한 다양한 고려 사항이 충족 되어야 한다. 즉, 분산 서비스의 가용성은 각 피어의 신뢰성의 의해 결정되지만, 피어들은 서비스의 지속성을 보장할 수 없다는 단점을 보완하기 위해 가까운 호스트나 피어끼리 어플리케이션 그룹 내에서 클러스터화 되는 topology-aware 오버레이를 구성

하여 콘텐츠 전송 지연에 대한 QoS를 지원하거나 콘텐츠의 패킷을 다중의 콘텐츠 피어로부터 병렬적으로 전송하는 등의 기술이 필요하다.



〈그림 1〉 P2P환경에서 주요한 콘텐츠 분배 특징

(1) 분산된 콘텐츠의 위치와 라우팅

P2P에서는 자신이 원하는 자원을 가진 피어까지 어떻게 찾아가는가 가장 큰 관심사가 된다. 이를 라우팅(routing)이라고 한다. 따라서 콘텐츠의 위치정보와 공간질의를 처리를 위해 위치 정보 기반의 라우팅 방법이 필요하다. P2P에서는 인프라 구성 방식에 따라 Structured 방식과 Unstructured 방식으로 구분할 수 있다. 또 자원 검색 방식에 따라 Centralized 구조와 Distributed 구조로 구분이 되기도 한다. 기존의 클라이언트/서버 모델은 자원에 대해 서버에 질의를 요청하면 서버는 그 질의를 처리하여 결과를 클라이언트에게 전달해줌으로써 수행된다. 그러나 다수의 클라이언트가 동시에 서버에 질의를 요청하게 되면 서버는 이를 처리하는 과정에서 병목 현상이 발생하고 부하는 커지게 된다는 단점이 있다.

Client-Server	Peer-to-Peer											
<ol style="list-style-type: none"> 1. Server is the central entity and only provider of services and content. → Network managed by the server 2. Server as the higher performance system 3. Clients are the lower performance system <p>Example: WWW</p>	<ol style="list-style-type: none"> 1. Resources are shared between the peers 2. Resources can be accessed directly from other peers 3. Peer is provider and requestor (server overlay) 	<p style="text-align: center;">Unstructured P2P</p> <table border="1"> <thead> <tr> <th>Centralized P2P</th> <th>Hybrid P2P</th> </tr> </thead> <tbody> <tr> <td> <ol style="list-style-type: none"> 1. All features of Peer-to-Peer included 2. Central entity is necessary to provide the services of functionality 3. Central entity is some kind of indexing database Example: Register </td> <td> <ol style="list-style-type: none"> 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 0.4, JXTA </td> </tr> </tbody> </table>		Centralized P2P	Hybrid P2P	<ol style="list-style-type: none"> 1. All features of Peer-to-Peer included 2. Central entity is necessary to provide the services of functionality 3. Central entity is some kind of indexing database Example: Register 	<ol style="list-style-type: none"> 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 0.4, JXTA 	<table border="1"> <thead> <tr> <th>Pure P2P</th> <th>DHT-Based</th> </tr> </thead> <tbody> <tr> <td> <ol style="list-style-type: none"> 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 </td> <td> <ol style="list-style-type: none"> 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 "bids" Examples: Chord, CAN </td> </tr> </tbody> </table>	Pure P2P	DHT-Based	<ol style="list-style-type: none"> 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 	<ol style="list-style-type: none"> 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 "bids" Examples: Chord, CAN
Centralized P2P	Hybrid P2P											
<ol style="list-style-type: none"> 1. All features of Peer-to-Peer included 2. Central entity is necessary to provide the services of functionality 3. Central entity is some kind of indexing database Example: Register 	<ol style="list-style-type: none"> 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 0.4, JXTA 											
Pure P2P	DHT-Based											
<ol style="list-style-type: none"> 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 	<ol style="list-style-type: none"> 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 "bids" Examples: Chord, CAN 											

〈그림 2〉 P2P환경에서의 라우팅 기술(2)

P2P 네트워크에서는 기존의 클라이언트/서버 모델과는 달리 자원이 다수의 피어에 분산되어 있기 때문에 여러 자원을 관리하고 검색하는 기술이 필수적이다. Unstructured P2P를 기반으로 하는 피어 검색 기법으로는 중앙집중형 P2P(Centralized P2P), 분산형 P2P(Pure P2P), 계층형 P2P(Hybrid P2P)가 있다. 반면 Structured P2P기반의 네트워크 구축기술은 Unstructured 방식에 비해 비교적 최근에 제안된 P2P 프레임 기술로 DHT(Distributed Hash Table)를 기반으로 구축되는 추세이다. DHT기반 오버레이 네트워크는 검색의 효율성을 높이고자 P2P 오버레이 네트워크에 구조적인 특성을 부여한 것이다. [3]

Unstructured P2P에서 가장 대표적인 라우팅 방법에는 Gnutella[4]에서 사용하는 플러딩(flooding)기법이 있다. 이는 자신에 연결되어 있는 모든 피어에게 메시지를 흘려 보내는 방법을 말한다. 이때 메시지가 모든 피어에 전달되는 것을 막기 위해 TTL(Time To Life)을 두어 TTL 수만큼만 메시지를 전달한다. 그러나 자신의 이웃피어에게 모두 메시지를 전달하기 때문에 발생하는 메시지가 많아져 검색비용과 네트워크의 부하가 증가한다. 이를 해결하기 위해 DHT를 이용한 Structured P2P 방법이 제시되었다.

대표적인 DHT 방법으로는 CAN[5]과 Chord[6], Tapestry[7]가 있다. CAN은 해쉬 함수를 통해 모든 피어를 d차원의 지표 공간으로 매핑(mapping)시키고 자신의 영역에 인접한 영역의 피어를 이웃피어로 등록한다. 질의 피어와 목적피어 역시 해쉬 함수에 의해 지표 공간으로 매핑되고 각 피어의 이웃피어 정보를 통해 질의 피어에게서 목적 피어로 메시지가 전달되어 피어가 검색된다. Chord는 파일의 키 값과 피어 식별자의 키 값이 같은 주소공간을 가지도록 정의한다. 참여 피어들은 이 주소 공간을 분할하여 각 영역의 식별자에 해당하는 파일들을 관리하고 각 피어들은 핑커테이블(finger table)이라는 경로 테이블을 유지해 피어의 링크 정보를 유지한다. Tapestry는 피어ID의 suffix를 이용하여 라우팅하는 기법이다. 각 피어와 콘텐츠에 160비트 식별자를 해쉬 함수를 통해 할당하며 콘텐츠에는 추가로 GUID를 할당한 후 Tapestry의 neighbor map을 이용하여 원하는 key 검색 시에 key와 가장 긴 침자를 공유하는 피어로 경로 설정이 이루어진다.

(2) 콘텐츠 저장, 복사, 이동

P2P에서는 하나 이상의 피어에 콘텐츠를 복사함으로써 콘텐츠에 대한 이용도를 향상시키고 성능을 강화하며 익명의 피어에 콘텐츠를 저장하여 콘텐츠의 유효성을 향상시킨다. 이러한 콘텐츠 복사 방법에는 수동 복사와 캐쉬 기반의 복사로 나뉜다. 수동 복사란 콘텐츠를 필요로 하는 피어가 콘텐츠를 소유하고 있는 피어에게 콘텐츠를 제공받는 방법으로 P2P 시스템 내에서 자연적으로 발생한다고 볼 수 있다. 캐쉬 기반의 복사는 Freenet[8], OceanStore[9], MojoNation[10]에서 사용하는 복사 방식으로 P2P 오버레이 네트워크 상의 피어들을 통과하며 콘텐츠가 전달될 때 통과되는 피어에 콘텐츠의 복사본을 저장하는 방식이다. Freenet에서는 검색 명령이 콘텐츠를 찾는데 성공하면 파일은 네트워크를 통해 명령의 초기 소유자에게 옮겨진 후 파일의 복사본은 콘텐츠를 검색하는 과정에서 거친 모든 중간 피어에 저장된다. 이 방식은 그 후에 있을 명령에 대해 이용성과 위치 특성화가 증가하고 콘텐츠의 이용성과 성능뿐 아니라 공간성을 향상시킨다. OceanStore는 콘텐츠를 요청하는 메시지의 트래픽을 관리하여 자주 사용되는 콘텐츠에 대해 복사본을 생성한다. 광범위한 저장소 중 사용자가 많은 지역으로 콘텐츠를 복사함으로써 충분히 높은 용량도 수준을 유지할 수 있다. MojoNation에서도 유사한 방법을 지원한다. 즉, 자주 사용되는 콘텐츠 정보를 관리하고 이에 대한 트래픽과 추가적인 복사본을 관리하는 중앙화된 서비스 브로커를 통해 복사된 콘텐츠를 관리한다.

더불어 동적인 복사본 관리 알고리즘을 적용하여 콘텐츠를 분배하는 방법도 제시되었다. 동적 복사본 관리 알고리즘은 최소량의 복사본을 동적으로 배치하는 알고리즘으로 클라이언트 대기시간에 만족하는 기존의 복사본을 먼저 찾고 찾을 수 없다면 새로운 복사본을 만든다. 이와 같은 동적 복사본 관리 알고리즘은 Scan[11] 시스템에서 사용되고 있다. 콘텐츠 분배에 있어 저장, 복사, 이동은 중요한 기술 중에 하나이나 콘텐츠에 대한 지속성, 동기화와 관련해 고려해야 할 문제들이 발생한다. 특히 콘텐츠 삭제와 업데이트가 허용되는 시스템에서 더욱 콘텐츠 유효성에 대한

검증은 반드시 필요할 것이다.

(3) P2P 환경에서의 보안

P2P는 분산환경으로 분산처리의 자치적 성격을 갖는 컴퓨팅 환경이다. 이를 위해 보안의 취약성을 분석하고 기존의 보안 메커니즘에 추가적으로 필요한 사항들이 발생한다. 또한 콘텐츠 분배 시스템 구조에서는 콘텐츠에 대한 가용성, 프라이버시, 비밀성, 보전성, 신뢰성 등이 제공하여야 한다. 이는 네트워크에 참여한 피어들은 자신이 속한 그룹에 대한 신뢰가 필요하기 때문이다. 먼저 P2P 환경에서의 보안 문제점들을 살펴 본 후 이에 대한 보안 이슈들을 정리한다.

온라인 상에서 주민등록번호로 인증을 하고 ID를 발급하면서 본인의 동의 없이 ID를 생성하고 악의적으로 이용하는 등의 문제점이 있다. 이는 P2P 환경에서도 ID를 사용함에 있어서 실체의 검증을 할 수 없는 경우에 더욱 큰 어려움을 야기시킬 수 있는 취약점이다. 또한 ID 발급이 자유롭게 허용되어 있기 때문에 ID에 대한 추적이 어렵다. 즉 아무런 규칙 없이 ID 발행을 허용한다면 ID가 도용되었을 때 허가된 영역, 권한 외에서의 사용에 대한 능동적 대처가 어렵게 된다. 그러므로 ID 발급에 대한 최소한의 인증 서버가 있어서 ID에 대한 명확성과 제한성을 부여하여 ID 검증성을 강화하여야 한다. P2P는 개인의 정보가 더 위협을 받는 환경이 되었다. 누가 누구의 정보를 사용·공유 할 수 있는 지에 대한 정의도 없으며 누가 관리를 하느냐 하는 것조차 규정된 것이 없다. 이러한 환경에서 개인의 정보가 불필요하게 네트워크나 서비스 공급자에게 제공되거나 관리되어서는 안 된다.

P2P에서의 보안은 신뢰정보(Credential)를 관리하는 주체가 누구냐에 따라 중앙 집중형과 분산형으로 구분할 수 있다. 중앙 집중형의 신뢰정보 관리 방식은 신뢰할 수 있는 서버가 정보의 생성/분배/인증/폐기 등 보안의 제반 과정에 개입하는 형태를 말한다. 분산형은 이러한 서버의 도움 없이 네트워크 상에 분산되어 있는 피어 간에 또는 그들 간의 협업에 의해서 신뢰정보를 관리하는 경우를 의미한다. 중앙 집중형은 이미 국내외의 인터넷 환경에서 매우 효과적으로 사용되던 방식으로 PKI기반의 인증 기법 등을 예로 들

수 있다. 분산형은 P2P에 참여하는 피어들간에 그룹을 생성하고 그룹 생성자가 인증기관의 역할을 수행하는 방식으로 JXTA에서 사용되고 있다.

(4) 콘텐츠에 대한 익명성 보장

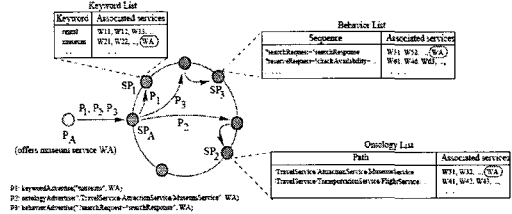
인터넷이 일반화되면서 개인과 개인이 디지털 콘텐츠 등의 정보를 주고 받는 P2P 구조가 확산되고 있다. 이러한 P2P 네트워크에서는 자유롭게 콘텐츠를 이용할 수 있고 특정한 경우를 제외하고 거의 무료로 이용할 수 있을 뿐 아니라 자신의 PC에 저장할 수 있다는 장점이 있지만 콘텐츠에 대한 소유권 등 법적 분쟁이 발생하면서 피어의 익명성 보장에 대한 측면이 부각되었다.

Freenet은 기초부터 완전히 익명성을 보장하여 절대 검열이 되지 않도록 설계 시스템이다. Freenet은 Gnutella처럼 상대방의 IP를 통해 정보를 주고 받는 기반이나 정보가 이동하면서 그 흔적을 가린다는 점에서 Gnutella와 차이점이라 할 수 있다. 각각의 피어는 자신이 소유하고 있는 문서를 파악하고 다른 피어에 저장된 문서에 대한 정보만을 갖는다. 자신과 인접한 피어들은 정기적으로 서로 통신하면서 네트워크 상에서 서로 업데이트 함으로 전체 네트워크의 일부 분 이상만을 알 수 있게 구성된다.

FreeHaven[12]은 콘텐츠에 대한 분할을 통해 익명성을 지원한다. 이는 콘텐츠를 저장하고 있는 피어를 찾거나 피어가 소유한 콘텐츠를 파괴할 수 있는 강력한 광고 기능을 지원함으로써 가능하다. 이는 콘텐츠를 제공하는 공급자나 사용하는 수요자에 대해 모두 익명성을 제공한다. 즉, 공급자는 자신의 콘텐츠를 FreeHaven 시스템에 제공하게 되고 시스템은 FreeHaven에 참여하고 있는 여러 피어들에게 공급된 콘텐츠를 분할하여 저장하게 한다. 이를 위해 FreeHaven에 참여한 피어들의 저장 공간을 요구하게 되며 저장되는 콘텐츠의 정도에 따라 인센티브를 준다. FreeHaven 시스템을 통해 콘텐츠를 공급하는 피어나 제공받는 피어 모두 콘텐츠의 정확한 위치를 알 수 없으며 오직 FreeHaven 시스템만이 전체 콘텐츠를 구성할 수 있는 방법을 알게 된다. 또한 참여자에게 제공되었던 콘텐츠에 대한 라이프타임을 적용하

여 일정 시간이 지나면 제공된 콘텐츠는 파괴된다. 나눠진 콘텐츠는 참여자의 콘텐츠 요청이 있을 때 조각을 모아서 하나의 전체 콘텐츠를 구성한다. Freenet과 비슷하게 사용자의 요청이 늘어날수록 저장된 콘텐츠의 량도 늘어나게 된다. 그러나 이러한 동적이고 익명성이 있는 환경에서 사용자에 의해 콘텐츠가 언제 지워질 지 간음하기 쉽지 않고 동일한 데이터에 대해 여러 피어가 나눠져 저장하고 있으므로 그에 따른 저장공간의 낭비도 발생한다.

의 데이터에 대한 검색과 빠른 전송 속도를 기대하기 어렵다는 단점이 있다.



〈그림 3〉 SPiDeR 검색 구조

3. P2P와 결합된 웹 서비스와 스트리밍 서비스

(1) P2P와 웹 서비스 동향

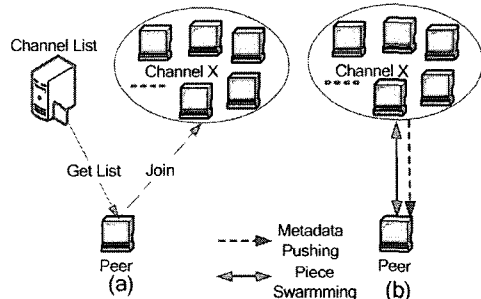
초기 웹 기술은 HTML, HTTP, URL 등이 개발된 이후에 사용자 측면에서 정보를 제공하고 이용하기 위한 기술들 위주로 개발되어 왔다. 웹 기술과 XML 관련 표준 개발은 단순히 사용자에게 웹 페이지를 보여주는 수준에서 벗어나 웹 서비스 미들웨어로 발전되었다. 웹 서비스 미들웨어는 하나의 서비스나 사업 모델이 웹을 이용하는 다른 서비스들에게 부분적인 계산의 수행을 위탁하거나 일부 서비스의 수행을 위탁하고 그 결과를 제공받는 것을 일컫는다. 이러한 웹 서비스 기술과 P2P 장점이 결합되어 확장성과 상호운영성을 지원하는 연구가 진행되고 있다.

SPiDeR[13] 다양한 웹 서비스를 발견하고 P2P 환경에서 실행할 수 있는 기초적이거나 틀을 제공하는 시스템이다. P2P 오버레이 네트워크 상에 다양한 서비스 프로바이더들로 구성되며 다른 피어들이 제공한 광고를 저장하고 서비스 검색에 있어 동적이면서 분산된 목록 서비스를 지원한다. 그림 3에서 보는 바와 같이 코드 링 방식을 이용하여 위치시킨 서비스 프로바이더에 의해 키워드 중심으로 검색 서비스가 지원되며 웹 서비스와 연관되어 서비스에 쉽게 접근할 수 있다. 기존의 인터넷은 공개되고 큰 스케일에 다양하고 많은 양의 데이터가 산재되어 있는 분배 네트워크로 구성되어 있다. 이러한 환경에서 어떤 사용자도 공유되어 있는 콘텐츠를 제공받을 수 있다. 그러나 많은 양

의 데이터에 대한 검색과 빠른 전송 속도를 기대하기 어렵다는 단점이 있다.

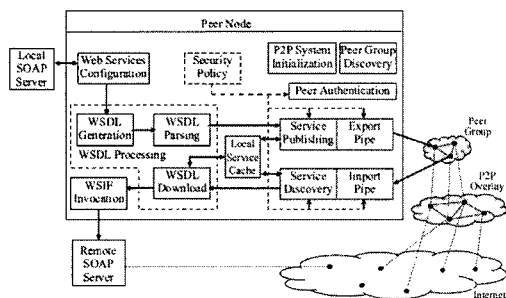
Xeja[14] 시스템은 P2P 멀티 소스 콘텐츠 분배 시스템으로 유연한 채널 기반의 분배된 디렉터리 서비스를 지원한다. 자동적으로 사용자가 어떤 공동의 관심을 가지고 있는 지에 대한 정보를 모으고 참여 피어들 사이에 채널에 대한 계층 구조를 활용하여 관리한다. 그림 4와 같이 공동관심을 갖는 피어들끼리 채널을 형성하고 콘텐츠를 주고 받음으로써 콘텐츠의 전체 조각에 대한 가용성을 증가시켰다. Xeja는 다운받는 콘텐츠에 대한 전체 콘텐츠를 보장한다. 즉, 모든 피어들 사이에 콘텐츠 조각을 가지고 있어 결국 전체 콘텐츠를 확보할 수 있다.

Xeja는 채널 기반의 콘텐츠 분배 오버레이 네트워크를 구성하고 있어 많은 소스 피어로부터 병렬적으로 파일을 다운로드 할 수 있고 이러한 상태를 유지하기 위해 자동적으로 채널을 재구성한다. 콘텐츠 디렉터리 서비스는 콘텐츠에 대한 메타 정보를 가지고 있고 인기 있는 파일에 대해서는 채널이 생성되어 있다. 채널은 온라인 커뮤니티와 비슷한 개념으로 관심 있는 피어들에 의해 유지된다.



〈그림 4〉 Xeja의 채널 기반의 웹 서비스

WebPeer[15]는 공유된 서비스 인프라 안에서 웹 서비스에 대한 상호가용성을 지원하는 환경을 지원한다. 이러한 환경은 서비스 공급과 자원의 위치 검색, 공유에 대한 기본적인 요구사항이 충족 되어 한다. WebPeer는 Web Services oriented peer-to-peer (WSOP)를 통해 서비스 공급과 검색 메커니즘을 제공하는 중앙화된 구조를 제공한다. 이는 분산된 컴퓨팅 환경에서 서버가 서비스를 UDDI에 등록하고 클라이언트는 레지스터에 접속하여 서비스의 위치를 파악하여 접속하는 것으로 그림 5와 같은 웹 서비스 기반의 P2P 구조로 도식화 할 수 있다.



<그림 5> WSOP 구조상에서 서비스 제공과 검색 모델

WSOP 구조는 중앙화와 분산된 특징을 모두 가지고 있는 구조이다. WSOP 기반의 피어들은 프로바이더를 하거나 서비스를 요청하는 피어가 될 수 있다. 일반적으로 피어는 서비스를 제공하거나 서비스를 소비하고, 서비스를 등록하는 3가지 권한을 가지고 있다. 웹 서비스와 비슷하게 WSOP 시스템에 있는 피어들은 검색된 서비스에 대하여 필히 제공해야 하며 네트워크에 참여한 모든 피어들이 접근 가능해야 한다. WSOP 시스템 구조에서 데이터에 대한 관리를 다음과 같이 수행하고 있다. 시스템은 초기화 될 때 피어 이름과 로그인, 비밀번호, HTTP 주소, TCP 포트, 랑데부 피어와 relay 피어에 대해 초기화 한다. 그 후 디폴트 피어 그룹에 속하게 되고 피어의 관심사에 따라 정의된 피어 그룹을 검색할 수 있으며 자동적으로 참여하고 관심 그룹이 없을 경우 독립적으로 구성할 수 있다.

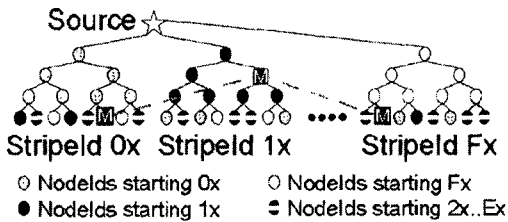
(2) P2P와 스트리밍 서비스 동향

기존의 미디어 스트리밍 서비스는 CDN (Contents Delivery Network)을 구성하고 미디어 서버를 통해 콘텐츠를 제공하였다. 그러나 순수 P2P 네트워크에서는 콘텐츠를 제공하는 주체가 고정되어있지 않으므로 콘텐츠의 안정적인 제공을 위해서는 피어간 콘텐츠 분배는 매우 중요한 기술이라 할 수 있다. 따라서 P2P 미디어스트리밍에서 요청 피어는 재생과 동시에 미디어데이터를 저장하고 스트리밍이 끝나면 다른 피어에게 미디어 파일을 제공하는 새로운 소스 피어가 되어야 한다. 그리고 특정 미디어파일을 소유한 피어의 수가 적으면 멀티소스 미디어스트리밍에 참여할 수 있는 소스 피어의 수도 소수이므로 특정미디어에 대한 요청 수를 충족시키지 못할 수 있다. 이러한 P2P 환경에서 효과적인 스트리밍 서비스를 제공하기 위해 다양한 연구가 진행 중에 있다.

SplitStream[16]은 Pastry와 scribe을 기반으로 하는 대표적인 스트리밍 시스템으로 마이크로 소프트의 연구 분야 중 한 분야이다. SplitStream은 큰 대역폭을 요하는 콘텐츠에 대한 스트리밍과 분배를 도와주는 시스템으로 P2P 환경에서 미디어 전송과 파일분배 모두에 사용될 수 있다. SplitStream은 보내고자 하는 데이터를 스트라이프(Stripe)로 명명한 세그먼트 k개로 나누고, 나눈 스트라이프 개수만큼 서로 다른 내부 피어를 갖는 멀티캐스트 트리를 생성한다. 각 스트라이프는 해당 트리를 통해 전달된다. 하나의 스트라이프는 스트라이프 식별자(stripe id)로 한다. 그림 6은 데이터를 16개의 스트라이프로 나누어 다중 멀티캐스트 트리를 구성한 예이다.

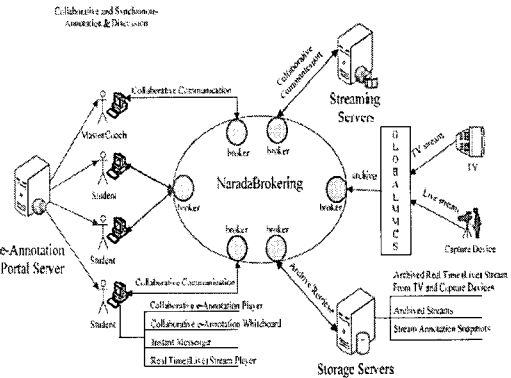
SplitStream은 부하를 분산시키기 위해서 자식 연결의 개수를 제한한다. 만일 자식 연결 개수를 제한하지 않는다면 자식 피어의 개수가 그 피어의 용량을 넘어서 병목현상을 유발할 수 있다. 따라서 SplitStream에서는 자식 연결 개수가 한계에 다다른 피어가 다른 피어로부터 연결 요청을 받으면 일단 연결을 요청한 피어를 자식피어로 받아들이고, 새로운 자식 피어 집합들을 평가하여 추방할 자식피어를 선택한다. 먼저 스트라이프 식별자와 피어 식별자가 같은 접두사를 공유하지 않는 자식 피어를 선택하여 추방한다. 만일

연결을 요청했던 피어가 선택된다면 바로 추방되고, 아니면 해당하는 자식 피어 중 임의의 피어가 선택되어 추방한다. 만일 그런 피어가 없다면 자식 피어 중 해당 스트라이프와 가장 짧은 접두어 공유를 가진 피어를 선택한다. 만일 요청을 회망한 피어가 여기에 해당된다면 추방하고, 아니면 해당하는 자식 피어들 중 임의의 피어를 선택해 추방한다. 이때 추방되도록 선택된 피어는 해당 스트라이프 트리에서 고아가 된다. 고아가 된 피어는 다시 자신의 부모 피어를 형제 피어들 중 스트라이프 식별자와 같은 접두어 공유를 갖는 임의의 형제피어에게 연결 요청을 보내는 방법과 여분의 능력그룹을 이용하여 구축된 스트라이프 멀티캐스트 트리들 중에서 현재 자신의 전달 능력보다 적은 개수의 자식 피어를 갖고 있는 그룹에 합류를 요청하는 방법으로 새로운 그룹에 속하게 한다. [17]



<그림 6> SplitStream의 다중 멀티캐스트 트리 구성

Community Grids Lab에서 진행중인 NaradaBrokering [18] 프로젝트는 분산 미들웨어 시스템 환경에서 웹 서비스, 그리드 서비스, peer-to-peer 상호작용, 일반적인 미들웨어 함수를 통합한 단일 메시지 환경에서의 상호 메시지 교환에 대한 연구를 진행 중이다. NaradaBrokering이란 이름은 설득의 힘을 가지고 있고 신의 메시지 역할을 하였던 Narada라는 고대 인도 신화에서 유래된 인물이다. NaradaBrokering은 분산된 메시지 하부구조로 두 종류의 밀접한 능력을 제공한다. 첫 번째는 메시지 기반 미들웨어로 이는 사용자, 자원, 서비스, 프록시 등으로 표현되는 엔티티들의 메시지 교환을 통한 통신을 쉽게 도와준다. 두 번째는 메시지의 소비자로서 등록된 사용자에게만 효과적인 라우팅 메시지를 이용하여 통지 프레임워크를 제공한다.



<그림 7> Time Streaming Data Grid Application 구조

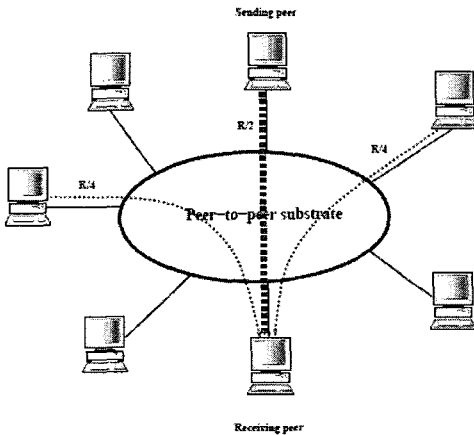
NaradaBrokering은 그림 7과 같이 실시간 스트리밍 데이터 그리드 (Real-time Streaming Data Grid) 미들웨어를 제공한다. 이는 시간이 진행됨에 따라 RTSP(Real Time Streaming Protocol)을 이용하여 비디오 스트리밍을 지원한다. RTSP는 HTTP/1.1의 함수나 구문이 유사하고 HTTP에 의해 확장된 프로토콜로 기본적으로 상태를 유지한다. 상태에는 초기화, 준비, 플레이/레코딩이 있고 기본적인 RTSP 조정 기능으로는 시작, 멈춤, 찾기 등이 있다.

또한 웹 서비스 기반의 스트리밍 서버를 제공한다. 이는 RTSP 상위에서 고려되는 제어 기능들과 함께 RTSP의 미들웨어를 제공한다. 또한, 스트리밍 서버는 라이브 스트리밍의 찾기도 제공하는데 이는 as RealNetworks Helix Server와 QuickTime Streaming Server 과 같은 현재의 RTSP 구현물들에서는 제공하고 있지 않는 기능이다. NaradaBrokering 미들웨어를 이용한 스트리밍 서비스는 네트워크를 가로질러 데이터들을 전송한다. 스트리밍 클라이언트는 NaradaBrokering 피어를 통해 데이터를 주고 받는다. 오디오, 비디오, 스틸 이미지, 문자 메시지 등과 같은 다양한 타입을 포장하여 브로커 네트워크를 통해 전송할 수 있다. 스트림 메타 정보를 기초로 사용자는 비디오인지 오디오인지 혹은 다른 데이터 타입인지 인지하고 처리한다.

NaradaBrokering 에서 모든 멤버는 그 그룹 내의 모든 다른 멤버들의 리스트를 관리한다. 그리고 그 당사

내의 그것의 이웃들과 그룹 멤버십 지식을 주기적으로 교환한다. 어떤 피어가 그 그룹에 결합하기를 원할 때, 그것은 out-of-band bootstrap 메커니즘으로 그룹 멤버의 리스트를 얻어야 하고, 그것은 어떤 이웃으로 추가되기 위하여 요청하는 메시지를 전송한다. 어떤 피어가 그 그룹을 이탈하기를 원한다면, 그것은 그것의 이웃에게 공지하고, 그리고 이 정보는 그 망사에 따라 그 그룹의 다른 멤버들에게 전달된다. 이것 때문에, NaradaBrokering 은 대형 네트워크를 지원할 수 없고, 그것은 단지 작은 P2P 네트워크에 역점을 둔다.

PROMISE[19]는 Purdue University에서 연구 중인 프로젝트로 피어 룩업, 피어 기반의 모아진 스트리밍, 네트워크와 피어 상황에 동적인 적응 등이 주요 기능을 갖는 새로운 피어 투 피어 스트리밍 시스템이다. 특히, CollectCast라고 불리는 새로운 어플리케이션 레벨의 p2p 서비스를 기반으로 하고 있다. CollectCast는 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 8〉 PROMISE 구조

PROMISE 구조는 그림 8에서 도식한 것과 같이 P2P 기판(substrate)을 통해 상호 연결된 피어간의 집

합으로 구성된다. P2P 기판은 피어 간의 연결성, 피어 멤버십 관리, 콘텐츠에 대한 룩업(look up) 성능 등을 유지한다. 미디어 스트림 데이터를 여러 송신 피어로부터 전달받을 수 있다. 이러한 PROMISE의 기능은 P2P 기판아래 독립적으로 동작한다. 즉, Pastry, Chord, CAN과 같은 P2P 기판 위에 PROMISE가 배치될 수 있다. PROMISE는 콘텐츠 요청이 들어오면 하부 P2P 기판을 통해 검색되고, 검색된 피어들 중 요청 피어에게 최적으로 전달할 수 있는 콘텐츠 피어 들을 선택한다. 요청 피어는 선택된 피어들과 일대일 연결을 맺고 송신 피어들에게 송신율과 데이터 할당을 배정하여 콘텐츠 전송을 시작한다.

4. 웹 서비스를 활용한 P2P기반 콘텐츠 분배에 대한 제안

P2P 통신 기술을 이용해 특정한 서버 없이도 자원과 콘텐츠를 공유할 수 있으며 여러 분야와 접목하여 응용될 것이다. 그러나 P2P 분산 환경에서 서비스의 이동성 지원과 컨텍스트 관리를 통해 동적으로 자원을 이용하여 분산 환경의 안정화 지원에 어려움이 있기 때문에 피어 시스템의 변화에 따라 서비스의 상태 변화 또는 추후 서비스의 생성 등을 가능 하도록 하는 환경을 구축할 필요가 있다. 이를 바탕으로 맞춤형 P2P 분산 환경에서 안정화를 지원하는 서비스를 제공하기 위해서 서비스의 생성은 물론 사용하고자 할 서비스의 합성 및 검색을 통해 여러 가지 형태로 만들어 내는 기술이 필요하고, 각 피어의 컨텍스트 정보를 통해 피어 시스템간의 안정화를 이끌고 콘텐츠를 효율적으로 분배하기 위한 기술과 사용량을 모니터링 할 수 있는 표준화된 인터페이스를 가지는 서비스 관리기술의 개발은 개방형 표준을 고려하고 P2P 기반 분산 환경에서 콘텐츠 이동의 안정화가 필요하다.

본 고에서는 P2P 환경에서 콘텐츠 정보를 관리하고 콘텐츠를 소유하고 있는 피어의 위치 정보를 제시하는 방법과 개방형 표준을 고려하고 스트리밍 데이터를 안정적이고 효율적으로 제공하기 위한 다양한 콘텐츠 구성 방법과 전송 스케줄을 제공함으로써 콘텐츠 분배에 대해 제안한다. 이는 피어/그룹 관리를 통

해 P2P 오버레이 네트워크를 구성하는 피어를 기능별로 분류하고 콘텐츠 별로 채널을 생성하여 콘텐츠 관리를 통해 콘텐츠의 완전성을 보증하고 피어의 성능을 고려한 분배 트리 알고리즘을 적용하여 피어간 콘텐츠의 최적 분배를 지원한다. 또한 JXTA 프로토콜을 이용하여 피어 간의 메시지를 관리함으로써 이 기종의 플랫폼에도 적용할 수 있으며 콘텐츠 전송 및 피어의 상태 모니터링 서비스를 이용하여 안정화된 서비스를 제공할 수 있다.

5. 결론

P2P 기술은 여러 분야에 응용되고 있고, 특히 콘텐츠 공유 응용프로그램은 큰 인기를 누리고 있다. 그러나 기존의 콘텐츠 공유 응용프로그램은 자료는 공유하도록 제공한 피어의 성능이나 서버의 성능과 대역폭에 따라 데이터 전송 속도가 급격히 떨어지는 단점이 있으며 미디어 데이터에 대해 다운로드가 끝난 후에야 재생을 할 수 있는 등 응답시간이 길고 원하는 콘텐츠인지 미리 확인할 수 없는 단점을 가지고 있다. 이러한 단점을 보완하기 위한 다양한 기술들을 살펴 보았다. 또한 앞으로 P2P 기술 결합하여 새로운 서비스를 제공과 웹 서비스가 결합하여 개발 된다면 사용자의 QoS를 만족시킬 수 있는 스트리밍 데이터 및 대용량의 데이터 서비스를 제공할 수 있다. 즉, 웹 서비스 서버가 피어 사이의 중개자 역할을 하여 접속 로드를 조절하여 이와 같은 단점이 개선될 수 있다.

또한 앞으로의 유비쿼터스 환경 안에서 다양한 기기 (device)들은 동적으로 네트워크를 구성하고 구성된 네트워크 구조 아래 서비스 관리에 있어 일관된 서비스 이용 방법을 제공하고 다종의 기기 간 서비스 이동성을 지원하는 구조를 제공하기 위하여 기존의 패러다임을 벗어난 서비스 지향 구조(SOA)로 발전해 나갈 것이다. 위의 요구조건을 모두 충족시키기 위해 서비스 지향 구조로 각광받고 있는 웹 서비스 지원과 이러한 서비스 관리를 동시에 제공하는 통합 미들웨어가 출현할 것이다. 이 경우 분산된 네트워크 환경의 안정성이 우선적으로 확보되어야 한다. 이러한 환경에서는 사용자의 요구사항과 Context에 부합하는 서비스들

을 선택하여 응용프로그램을 조직하여 보다 다양한 유비쿼터스 서비스를 제공할 수 있게 될 것이다.

참고 문헌

- [1] A. Oram, editor. Peer-to-Peer: Harnessing the Power of Disruptive Technologies. O'Reilly & Associates, March 2001.
- [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] Stephanos Androutsellis-Theotokis and Diomidis Spinellis. "A survey of peer-to-peer content distribution technologies". ACM Computing Surveys, December 2004
- [4] Gnutella homepage. <http://www.gnutella.com/>
- [5] S.Ratnasamy, P.Francis, M.Handley, R.Karp, and S.Shenker, "A Scalable content-addressable network", Proc. of ACM SIGCOMM, 2001
- [6] I.Stoca, R.Morris, D. Karger, F.Kaashoer and H. Balakrishnan, "Chord: A scalable peer-to-peer lookup service for Internet application", in Proc. ACM SIGCOMM 2001
- [7] Kirsten Hildrum, John Kubiawicz, Satish Rao and Ben Y. Zhao, "Distributed Object Location in a Dynamic Network " Theory of Computing Systems, March 2004, No. 37, Pgs. 405-440, Springer-Verlag Expanded journal version of SPAA 2002 paper.
- [8] Freenet homepage. <http://freenet.sourceforge.net/>
- [9] OceanStore homepage <http://oceanstore.cs.berkeley.edu/>
- [10] mojonation homepage <http://www.mojonation.net/>
- [11] Scan homepage <http://3d2f.com/tags/scan/p2p/>
- [12] FreeHaven homepage <http://freehaven.net/>
- [13] 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.

- [14] Pei Zheng, Chen Wang Xejia: A Scalable Channel-Based Multi-source Content Distribution System. WCW 2004: 106-115
- [15] Ruixuan Li, Zhi Zhang, Zhigang Wang, Wei Song, Zhengding Lu, "WebPeer: A P2P-based System for Publishing and Discovering Web Services", Proceedings of the 2005 IEEE International Conference on Services Computing - Volume 02
- [16] M. Castro, P. Druschel, A-M. Kermarrec, A. Nandi, A. Rowstron and A. Singh, "SplitStream: High-대역 폭multicast in a cooperative environment", SOSp'03, Lake Bolton, New York, October, 2003.
- [17] 김은석, 장지용, 박성용, "R-CAT : 피어능력을 고려한 내구적 멀티캐스트 트리 생성기법", 2006 한국컴퓨터종합학술대회 P.147~156
- [18] Shrideep Pallickara and Geoffrey Fox "NaradaBrokering: A Distributed Middleware Framework and Architecture for Enabling Durable Peer-to-Peer Grids" in Proceedings of ACM/IFIP/USENIX International Middleware Conference Middleware-2003, Rio Janeiro, Brazil June 2003
- [19] Mohamed Hefeeda, Ahsan Habib, Boyan Botev, Dongyan Xu, Bharat Bhargava "PROMISE: Peer-to-Peer Media Streaming Using CollectCast" MM'03 2003

◎ 저 자 소 개 ◎



김 윤 희

1991년 숙명여자대학교 전산학과 학사
 1996년 Syracuse University 전산학과 석사
 2000년 Syracuse University 전산학과 박사
 1991년~1994년 한국전자통신연구소 연구원
 2000년~2001년 Rochester Institute of Technology, 컴퓨터공학과 조교수
 2001년~2004년 숙명여자대학교 조교수
 2004년~현재 숙명여자대학교 부교수
 관심분야: 문제풀이환경, 자율컴퓨팅, 분산 관리
 E-mail: yulan@sm.ac.kr



송 은 혜

2005년 숙명여자대학교 컴퓨터공학과 학사
 2005년~현재 숙명여자대학교 컴퓨터공학과 석사과정
 관심분야: 그리드 컴퓨팅, P2P 컴퓨팅
 E-mail: grace@sm.ac.kr