키워드 검색 지원을 위한 확장 CAN 메커니즘

이명훈* · 박정수* · 조인준*

Extended-CAN Mechanism to Support Keyword Search

Myoung-Hoon Lee* · Jung-Soo Park* · In-June Cho*

요 약

분산 해쉬 테이블 기반의 구조적 P2P 시스템은 확장성이 우수하며 체계적인 검색과 라우팅을 수행하기 때문에 효율적인 검색이 가능하여 주목을 받고 있다. 그러나 이러한 장점에도 불구하고 공유파일 검색이 파일 식별자의 정확한 일치를 통해서만 가능하다. 즉 키워드 검색을 지원하지 못함으로써 P2P 응용에 있어 커다란 걸림돌이 되고 있다.

본 논문은 분산 해쉬 테이블 기반의 구조적 P2P 시스템에서 공유파일의 컨텐츠 기반 키워드 추출 및 파일 식별자를 생성하고 PLS의 확장을 통해 키워드 사전인 KID와 CKD를 작성하여 피어에서 키워드 검색을 지원하는 확장된 CAN 메커니즘을 제안하였다.

ABSTRACT

Recently, DHT-based Structured P2P System have a attention to scalability and providing efficient lookup by routing. Retrieving content of DHT-based P2P require knowledge of the exact identifier of sharing file. But user may wish to search for sharing file using descriptive keyword or content.

To resolve the problem, this paper propose Extended-CAN mechanism to support keyword search. We defined content-based keyword and identifier of sharing file, and PLS extended to KID and CKD for keyword, common keyword processing. As a result, Extended-CAN mechanism provide efficient keyword search for DHT-based Structured P2P System.

키워드

P2P, Peer-to-Peer, Keyword Search, CAN, DHT

* 배재대학교

접수일자 : 2006. 1. 11

Ⅰ. 서 론

P2P(Peer-to-Peer) 시스템은 응용계층에서 생성되는 오버레이 네트워크 구조를 통해 피어들의 자원(파일, 디스크, CPU 등)의 일부를 공유하여 사용하는 분산 시스템으로, P2P 네트워크에 참여하는 피어들은 서버와 클라이언트의 역할을 동시에 수행하는 서번트 형태로 동작한다.

초기 P2P 시스템에 대한 연구는 중앙 서버에 피어들이 연결된 형태로 오버레이 네트워크가 구성되는 중앙집중식 P2P 시스템에 초점이 맞추어 졌으나 중앙 서버의 트래픽 집중화 및 고장에 따른 전체 네트워크 단절과 확장성 제한이라는 문제점으로 인해, 현재는 중앙 서버의 개입 없이 피어들로 오버레이 네트워크를 구성하는 비중앙집중식 P2P 시스템의 연구가 활발하게 이루어지고 있다 [1][2].

특히 비중앙집중식 P2P 시스템 중 분산 해쉬 테이블(Distributed Hash Table, DHT) 기반의 구조적 P2P 시스템은 구조화된 오버레이 네트워크 위상에 위치한 피어들에게 피어의 노드 식별자(node-id)와 공유파일의 파일 식별자(file-id)를 담고 있는 해쉬 테이블의 버킷을 분산 적용한 개념이다. 분산 해쉬 테이블을 적용한 구조적 P2P 시스템은 확장성이 우수하며, 체계적인 검색 및 라우팅 수행으로 인한 높은 검색 효율성으로 인해 주목을 받고 있다 [3][4].

그러나 이러한 장점에도 불구하고 공유파일 검색은 파일 식별자의 정확한 일치를 통해서만 가능하며 키워드 검색을 지원하지 못함으로써 P2P 응용에 있어 커다란 걸림돌이 되고 있다.

이러한 문제점을 해결하고자 본 논문에서는 효율적인 키워드검색이 가능한 컨텐츠 기반의 확장 CAN 메커니즘을 제안하였다. 명확한 키워드 선정을 위해 공유 파일의 컨텐츠 기반 키워드 추출 및 파일 식별자 생성 메커니즘을 제안 하고, 피어 목록 서버인 PLS(Peer List Server)의 확장을 통해 키워드 사전인 KID(Keyword Index Dictionary)와 공통 키워드 사전인 CKD(Common Keyword Dictionary)를 작성하여 피어에서 키워드 검색을 지원하는 확장된 CAN(Content Addressable Network) 메커니즘을 제안한 것이다.

본 논문의 구성은 다음과 같다. Ⅱ장에서는 분산 해쉬 테이블 기반 P2P 시스템에서 키워드 검색 적용을 위한 관련연구에 대해 살펴보고, Ⅲ장에서는 키워드 검색을 지원하기 위한 컨텐츠 기반의 확장 CAN 메커니즘을 제안 하였다. Ⅳ장에서는 제안 시스템의 검토 및 고찰을 다루고, 마지막으로 Ⅴ장에서는 결론과 향후 연구 과제를 제시한다.

Ⅱ. 관련연구

본 장에서는 분산 해쉬 테이블 기반 P2P 시스템에서 키워드 검색 적용을 위한 관련연구에 대해 살펴본다.

2.1. 역 분산 해쉬 테이블

참고문헌 [5]는 키워드검색 문제해결을 위해 역 분산 해쉬 테이블(Inverted DHT) 기반의 확장된 Chord를 제안하였다.

역 분산 해쉬 테이블 기반의 P2P 시스템에서 공유 파일 데이터는 {keyword, list of values} 형태로 표현된다. 'keyword'는 공유 파일을 지칭하는 키워드들로 구성되며 각 키워드들을 SHA-1과 같은 해쉬 함수를 사용하여 생성한다. 'list of values' 는 키워드에 해당하는 공유 파일을 실제 소유하고 있는 피어의 위치정보 즉, IP주소이다.

파일 공유는 키워드 해쉬 값을 기준으로 m-bit 원형 식별자 공간에 위치한 피어들 중에 키워드 해쉬 값과 가장 유사한 노드 식별자를 갖는 피어에게 {keyword, list of values} 쌍이 저장된다.

파일의 검색 질의는 해쉬된 키워드 기반으로 이루어지며 키워드 해쉬 값에 대응되는 피어로 질의를 전달하여 파일의 위치정보를 얻는다. 다중 키워드의 검색에 있어서는 각 키워드에 대한 결과를 교집합 하여 최종 결과를 얻는다.

참고문헌 [5]의 아이디어는 분산 해쉬 테이블 기반 데이터 표현인 {key, value} 형태의 구성을 {keyword, list of values} 형태의 키워드 기반 구성을 통해 키워드 질의를 지원할 수 있으나 다음과 같은 문제점을 갖고 있다.

첫째, 공통키워드 처리의 문제점이다. 공통키워드란 파일을 의미하는 키워드 중 대다수의 파일에서 자주 등장하는 매우 일반적인 키워드로, 공통키워드가 많은 수의 파일들에 포함될 경우 공통키워드를 담당하는 피어는 과도한 저장 공간 및 질의를 요청받게 된다.

둘째, 다중 키워드 질의는 키워드 수만큼의 검색 질의문을 생성하기 때문에 전체 네트워크 트래픽양이 증가되는 문제점이 있다.

2.2. 키워드 융합

참고문헌 [6]은 공통키워드 처리의 문제점 해결을 목적으로 하는 역 분산 해쉬 테이블 기반의 확장된 Chord를 제안하였다.

키워드 융합 메커니즘은 참고문헌 [5]와 마찬가지로 공유 파일 데이터를 {keyword, list of values} 형태로 표현한다. 또한 공통키워드의 처리를 위해 FD(Fusion Dictionary)라는 공통 키워드 사전을 피어들이 유지한다. 공통키워드의 선별은 피어에게 요청된 키워드 검색질의를 바탕으로 키워드를 누적하여 작성이 되며 임계값 이상의 키워드를 공통키워드로 판단한다. 이렇게 작성된 FD는 일정 시간마다 피어들이 서로 갱신하여 동일한 FD를 유지하도록 한다.

파일의 질의는 해쉬된 키워드 값에 대응되는 피어로 질의를 전달하여 파일의 위치정보를 얻는다. 특히 다중 키워드의 검색에 있어서는 FD를 이용하여 공통키워드를 제외한 새로운 질의문을 만든다. 공통키워드를 제외한 각 키워드 질의에 대한 교집합 결과를 얻고 마지막으로 공통 키워드 부분을 추가시킨 교집합으로 최종 결과를 얻는다.

참고문헌 [6]의 키워드 융합 메커니즘은 공통키워드 처리로 인한 문제점을 해결하였지만 다음과 같은 문제점을 갖고 있다.

첫째, 공통 키워드 목록인 FD를 일관되게 유지하는 문제점이다. 즉, 키워드 융합 메커니즘은 FD 목록을 피어들 간에 동일하게 유지해야 하지만 P2P 네트워크가 커질수록 피어들 간의 계산양이 과도하게 증가하며 FD 갱신의 동기화가 사실상 불가능하다.

둘째, 키워드 융합 메커니즘은 공통 키워드 선정을 피어들의 검색 질의를 기반으로 하여 이루어지기 때문에 공통 키워드 선정의 모호성을 갖고 있다. 검색 질의 기반의 공통키워드 선정은 인기도에 따른 공통키워드 선정으로 볼 수 있으며, 실제 공유파일을 의미하는 컨텐츠 기반의 키워드들 중 공통키워드를 선정하여 FD 목록을 작성하는 것보다 키워드 수 및 공통 키워드수가 증가하는 문제점이 있다.

Ⅲ. 컨텐츠 기반 확장 CAN 메커니즘[4]

본 장에서는 분산 해쉬 테이블 기반의 P2P 시스템에서 키워드 검색 지원을 위해 컨텐츠 기반의 키워드 추출 및 파일 식별자 생성 메커니즘을 제안 하고, PLS의 확장을 통해 키워드 사전인 KID와 공통키워드 사전인 CKD를 작성하여 피어에서 키워드 검색을 지원하는 확장된 CAN 메커니즘을 제안한다.

3.1. 확장 CAN 오버레이 네트워크 위상

컨텐츠 기반 확장 CAN 메커니즘의 오버레이 네트워크는 그림 1과 같이 {x, y}의 공간 좌표를 갖는 기본 CAN 메커니즘의 2-차원 다면체 공간을 활용한다.

각 피어는 랜덤 함수를 통해 생성한 노드 식별자, 이웃 피어의 관리 공간영역 및 라우팅 테이블, 키워드 검색을 위한 KID와 CKD를 소유하고 있다.

새로운 피어의 네트워크 가입은 PLS의 접속을 통해 이미 네트워크에 가입되어 있는 활성 피어의 IP주소를 얻게 되며 활성피어와 피어삽입 메커니즘을 수행한다.

확장 CAN에서 피어 및 데이터는 두개의 해쉬 함수를 이용하여 키{x, y} 형태로 표현되고, 각각의 피어는 임의의 한 점 {x, y}을 선정하여 그 점을 담당하고 있는 피어의 담당 구역을 반으로 나누어 담당하게 된다.

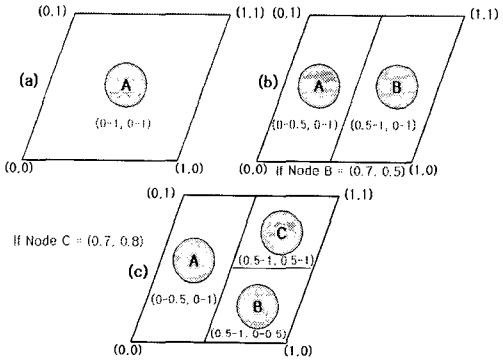

그림1. 확장 CAN 오버레이 네트워크 위상
Fig. 1. Overlay Network Topology of Extension CAN

그림 1.(b)는 두개의 피어로 구성된 확장 CAN 오버레이 네트워크의 구성을 보이고 있다. 만약 그림 1.(b)와 같이 구성된 네트워크에서 새로운 피어 C가 네트워크에 참여하고자 자신의 노드 식별자를 두개의 해쉬 함수를 사용하여 공간좌표 {0.7, 0.8}를 생성했다면, 피어 B는 이전의 자신의 담당구역 {0.5-1, 0-1}을 반으로 나누어 피어 C에게는 {0.5-1, 0.5-1}를 담당하게 하고 피어 B는 {0.5-1, 0-0.5} 공간 영역을 담당한다.

3.2. PLS

PLS는 P2P 오버레이 네트워크에 참여하고 있는 활성 피어들에 관한 정보를 담고 있는 초기 접속 서버이다. 중앙집중식 P2P에서의 중앙서버는 전체 네트워크를 구성하고 피어의 접속 목록, 피어의 공유파일 목록 관리, 질의 처리 등에 직접적인 간섭을 한다. 반면 비중앙집중식 P2P의 PLS는 현재 네트워크에 참여하고 있는 몇 개의 활성 피어의 IP주소 목록만을 유지하여 새로운 피어가 네트워크에 접속을 원할 때 IP주소 목록을 전송하여 네트워크 참여를 돕는다.

컨텐츠 기반 확장 CAN에서는 키워드 검색 지원을 위해 PLS의 기능을 다음과 같이 정의하였다.

3.2.1. 활성 피어 접속 목록 유지

PLS는 새로운 피어의 참여를 위해 현재 네트워크에 참여하고 있는 몇 개의 활성 피어의 IP주소를 유지한다. 이때 피어의 IP주소 목록은 KID와 CKD를 갱신하는 피어 중 컴퓨팅 능력이 높은 피어로 선정하여 IP주소 목록을 유지한다.

3.2.2. KID 생성 및 배포

KID는 키워드 인덱스사전으로 피어들이 공유하는 파일들의 키워드와 파일 식별자를 맵핑한 {file-id, list of keyword} 형태의 인덱스사전이다. KID는 피어가 공유 파일을 검색하기 위해 키워드 형태의 질의문을 작성했을 때 키워드들을 KID에서 검색하여 파일 식별자를 찾는 역할을 한다.

KID 생성 및 갱신, 배포는 PLS에서 담당하며 그 절차는 다음과 같다.

Step1) 피어는 파일 공유를 위해 컨텐츠 기반의 KI (Keyword Index)를 {file-id, list of keyword} 형태로 생성한다. 키워드 KI의 생성과정은 3.3 절차에 따른다.

Step2) 피어가 PLS 에 최초 접속 했을 때 자신이 소유한 KI를 전송한다. PLS는 KID와 비교하여 피어가 전송한 KI 목록 중 새로운 파일 식별자가 발견되면 KID 해당 목록을 추가한다.

Step3) PLS는 KI를 전송한 피어에게 갱신된 KID를 전송한다. KID는 CKD와 함께 피어의 키워드 검색 시 파일 식별자를 찾는 역할을 한다. 또한 피어의 KID 갱신은 피어의 KI가 수정되거나 일정한 시간 간격에 따라 PLS와 주기적으로 이루어진다.

3.2.3. CKD 생성 및 배포

CKD는 공통 키워드 사전으로 키워드 검색 시 공통키워드 처리를 위해 사용된다. CKD는 KID를 기반으로 {Common keyword, Hit} 형태로 생성된다. 'Common keyword'는 공통 키워드, 'Hit'는 KID를 기반으로 한 빈도수를 의미한다.

CKD 생성 및 갱신 및 배포는 PLS에서 담당하며 그 절차는 다음과 같다.

Step1) 피어에 의해 PLS의 KID가 갱신 되었을 때 KID에 갱신된 각각의 키워드를 CKD의 'Common keyword' 에서 검색한다.

Step2) 만약 이전에 존재하는 키워드일 경우 CKD의 'Hit'를 1증가 시키고, 존재하지 않는다면 CKD에 키워드를 등록한다. 이렇게 CKD에 등록된 키워드 중 'Hit'수가 높은 키워드는 공통 키워드로 선정이 된다.

Step3) CKD의 배포는 KID의 배포와 마찬가지로 피어의 초기 접속 또는 일정한 시간 간격에 따라 PLS와 주기적으로 이루어진다.

3.3. 컨텐츠 기반의 키워드 및 파일 식별자 생성

본 절에서는 공유파일에 대한 명확한 키워드 선정을 위해 컨텐츠 내용 기반의 키워드를 선정하고, 키워드의 조합을 통한 파일 식별자를 생성한다. 이를 통해 각 피어는 KI를 생성하여 파일 식별자는 네트워크의 피어들에게 분산하며 KI는 PLS에 전송하여 KID와 CKD의 생성을 돕는다. 이를 위해 본 논문에서는 오디오 파일을 대상으로 하였으며 ID3 태그 정보를 활용하였다.

표 1. ID3 태그 헤더
Table 1. Tag Header of ID3

위 치 (Bytes)	길 이 (Bytes)	설 명
00 ~ 02	03	TAG ID
03 ~ 32	30	Title(노래 제목)
33 ~ 62	30	Artist(연주자,가수)
63 ~ 92	30	Album(앨범명)
93 ~ 96	04	Year(발매년도)
97 ~ 126	30	Comment
127	01	Genre(장르)

ID3는 MPEG Layer 3(MP3) 오디오 파일에서 오디오 파일의 내용(곡명이나 저작자, 음악 채널 등)을 확인하기 위한 정보를 담고 있는 표준으로 사용되고 있는 태그 형식이다. ID3v1 태그는 MP3파일의 앞부분에 128bytes로 표 1과 같이 구성되어 있다.

오디오 파일의 키워드 및 파일 식별자 생성은 ID3v1 태그를 이용하며 KI 생성 절차는 다음과 같다.

Step1) 오디오 파일의 키워드는 ID3v1 태그를 활용하여 Title, Artist 정보를 추출한다.

Step2) 추출된 Title, Artist 정보를 공백단위로 분리한다. 이는 공백 단위의 분리된 정보들을 키워드로 사용하기 위함이며 본 논문에서는 Title의 공백은 10, Artist의 공백단위는 5로 제안하였다.

Step3) 공백단위로 분리된 키워드를 SHA-1 해쉬 함수를 사용하여 해쉬된 키워드 값으로 생성한다.

keyword1 = SHA-1(Title1)
keyword2 = SHA-1(Title2)
……
keyword10 = SHA-1(Title10)
keyword11 = SHA-1(Artist1)
……
keyword15 = SHA-1(Artist5)
list of keyword = (keyword1, keyword2, ……, keyword15)

Step4) 세 번째 단계에서 생성된 'list of keyword'를 조합하여 해쉬된 'file-id'를 생성한다.

$$file\text{-}id = SHA\text{-}1\{\sum_{n=1}^{10} SHA\text{-}1(Titlen) + \sum_{n=1}^{5} SHA\text{-}1(Artistn)\}$$

Step5) 상기의 단계를 통해 생성된 'list of keyword'와 'file-id'를 바탕으로 표 2와 같이 KI를 생성한다.

표 2. 피어의 키워드 인덱스
Table. 2. Keyword Index of Peer

file-id	list of keywords
file-id1	keyword1, … keyword15
:	:
file-idn	keyword1, … keyword15

컨텐츠 기반의 키워드 생성방법의 가장 큰 목적은 키워드 선정 방법을 명확히 하고 공통키워드수를 줄이기 위함이다. 기존의 키워드 선정 방법은 사용자의 임의대로 공유 파일에 대한 키워드들을 선정하기 때문에 동일한 공유 파일임에도 불구하고 키워드들이 서로 일치하지 않고, 공통키워드가 다수 발생하는 문제가 있다. 본 논문에서 제안하는 컨텐츠 기반의 키워드 생성방법은 공유 파일 컨텐츠의 메타정보에 따라 키워드가 생성되기 때문에 동일한 파일은 항상 같은 키워드들로 구성이 되어 키워드 선정의 명확성과 공통키워드 수의 감소를 갖게 된다.

3.4. 라우팅 메커니즘

컨텐츠 기반 확장 CAN 메커니즘은 분산 해쉬 테이블을 적용한 기본 CAN 라우팅 메커니즘을 활용한다. 즉, 모든 피어 및 데이터는 {key, value} 쌍의 형태로 표현이 되며 피어는 전체 네트워크의 해쉬 테이블 중 일부를 담당하게 된다. 네트워크상의 피어는 자신이 가지고 있는 노드 식별자 또는 파일 식별자를 해쉬하여 키를 구하고 오버레이 네트워크에서 해당 키가 속한 구역을 담당하는 피어에게 전달한다. 데이터를 검색할 때 역시 키 값을 구하여 같은 방식으로 검색 메시지를 전달한다.

컨텐츠 기반 확장 CAN 메커니즘에서 피어의 등록, 공유 파일의 등록, 검색 메시지를 특정 피어에게 전달하기 위하여 오버레이 네트워크 상의 라우팅을 위한 논리적 라우팅 테이블이 존재한다. 각 피어가 관리하는 논리적 라우팅 테이블은 이웃 피어의 노드 식별자(neighbor node-id)와 관리영역을 나타내는 'zone'으로 구성된 {neighbor node-id, zone} 형태의 테이블과 피어의 영역에서 관리하는 공유 파일정보인 {key, value} 형태의 테이블이 존재한다.

상기의 라우팅 테이블을 바탕으로 동작하는 라우팅 메커니즘 절차는 다음과 같다.

Step1) 피어는 특정 피어의 노드 식별자나 공유 파일을 검색하기 위한 파일 식별자를 생성하기 위해 두 개의 해쉬 함수를 사용하여 키를 생성한다.

Step2) 생성한 키 값이 피어 자신이 관리하는 영역이 아닐 경우 이웃 피어에게 키 값을 근거로 하여 검색 메시지를 전송한다. 이때 이웃 피어의 선정은 키 값과 가장 가까운 영역을 관리하는 피어를 선택한다.

Step3) 검색 메시지를 전송받은 이웃피어가 자신의 담당구역에 해당 키 값을 포함하지 않는 경우 라우팅 테이블의 이웃피어 중 키 값과 가까운 영역을 담당하는 피어에게 메시지가 전달되며, 이러한 과정이 반복되어 키 값을 담당하는 피어에게 최종 전달된다.

Step4) 키 값을 담당하는 피어에게 검색 메시지가 전달되었을 경우, 공유 파일을 찾기 위한 검색 메시지라면 피어가 담당하는 공유 파일정보인 {key, value} 형태의 테이블에서 키 값에 해당하는 'value' 값(실제 공유 파일을 소유한 피어의 IP주소)을 초기 검색 요청을 수행한 피어에게 직접 연결하여 전송한다. 그리고 피어의 참여를 위한 활성 피어의 검색 메시지였다면 3.5 피어의 참여 메커니즘을 수행한다.

결론적으로 라우팅 메커니즘은 한 홉씩 메시지를 전달할 때마다 요청한 키와의 거리를 좁혀가서 주어진 키가 속한 구역을 담당하는 피어에게 전달된다. 이러한 라우팅 메커니즘의 성능은 O(N1/d) (N : 전체 피어의 수, d : 차원수) 이며 차원이 증가함에 따라 성능이 개선된다. 구조적 P2P 시스템 중 O(log n)의 라우팅 테이블을 유지하는 Chord, Pastry등은 네트워크에 참여한 피어수가 많을수록 주기적인 메시지 전송이 많아진다. 그러나 CAN은 라우팅 테이블의 크기는 O(d)의 가상공간의 차원 값이므로 라우팅 테이블의 크기 측면에서 보다 좋은 확장성을 가진다고 할 수 있다. 즉, d를 작은 수로 유지한다면 적은 양의 이웃 피어 정보만을 관리하며 주기적인 메시지 전송 횟수를 고정시킬 수 있는 장점이 있다. 또한 유지 보수성을 볼 때 동적인 피어의 추가 및 삭제가 일어날 경우 라우팅 테이블의 수에만 영향을 준다.

3.5. 피어의 참여

컨텐츠 기반 확장 CAN 메커니즘의 오버레이 네트워크는 2-차원의 다면체 공간에 피어들이 구성되며 그 절차는 다음과 같다.

Step1) 확장 CAN 오버레이 네트워크에 참여를 원하는 피어는 랜덤 함수를 사용하여 'node-id'를 생성하고 3.3의 절차에 따라 KI를 생성한다.

Step2) 새로운 피어는 네트워크에 참여를 시도하기 위해 현재의 활성 피어들에 관한 정보를 담고 있는 PLS에 연결을 요청한다.

Step3) 피어의 연결을 수락한 PLS는 피어에게 노드 식별자와 KI를 요구한다. 이는 기존의 KID, CKD에 현재 접속을 요구하는 피어의 정보를 갱신하기 위함이다.

Step4) 피어는 PLS의 요구에 따라 'node-id' 와 KI를 전송하며 추가적으로 노드 식별자를 두개의 해쉬 함수를 사용하여 공간좌표 노드 식별자{x, y}를 생성하여 전송한다.

Step5) PLS는 새로운 피어의 네트워크 참여를 위해 활성피어 중 '노드 식별자' 기반의 공간좌표 노드 식별자{x, y}와 가장 유사한 피어를 선정하여 IP 주소를 피어에게 전송한다. 이는 피어의 초기 접속 시 공간할당을 위한 라우팅 수를 줄이기 위함이다. 또한 전송받은 KI를 기반으로 3.2의 KID 및 CKD 의 메커니즘에 따라 각 키워드 사전들의 갱신을 수행한 후 피어에게 전송한다.

Step6) 새로운 피어는 PLS에서 전송받은 정보를 바탕으로 활성 피어의 IP주소로 노드 식별자, 공간좌표 노드 식별자{x, y}를 전송하여 네트워크 참여를 요구한다.

Step7) 새로운 피어로부터 네트워크 참여를 요청받은 활성 피어가 공간좌표 노드 식별자{x, y}를 관리하고 있다면, 자신의 공간 영역의 반을 나누어 새로운 피어에게 담당하게 한다. 그렇지 않을 경우 공간좌표 노드 식별자{x, y}를 관리하는 피어를 3.4 라우팅 메커니즘에 네트워크 참여 요구 메시지를 라우팅 한다.

Step8) 새로운 피어에게 공간영역을 할당해준 활성 피어는 기존의 라우팅 테이블과 이웃 피어의 존 정보에 새로운 피어가 포함되도록 수정하고, 이를 새로운 피어와 이웃 피어들에게 갱신하도록 요구한다.

3.6. 공유데이터 삽입

컨텐츠 기반 확장 CAN 메커니즘에서의 공유 파일 삽입은 피어의 초기 네트워크 접속 시 또는 공유 파일의 갱신이 발생했을 때 수행되며 절차는 다음과 같다.

Step1) 피어가 공유 파일을 오버레이 네트워크에 삽입

하기 위해 공유파일의 파일 식별자를 생성한다. 컨텐츠 기반 확장 CAN 메커니즘에서는 파일 식별자가 공유 파일의 컨텐츠 기반 키워드의 조합으로 이루어지며 해쉬된 키워드 값은 피어의 키워드 검색에 활용된다. 따라서 3.3 컨텐츠 기반의 파일 식별자 생성 절차에 KI를 생성한다.

Step2) 피어는 PLS 접속하여 KI를 전송한다. 3.2의 절차에 따라 피어의 KI를 기반으로 PLS의 KID와 CKD 갱신하고 이를 피어에게 전송한다.

Step3) 오버레이 네트워크의 파일 식별자를 담당하는 피어에게 {file-id, value}를 삽입한다. 컨텐츠 기반 확장 CAN 메커니즘은 분산 해쉬 테이블 기반의 2차원 다면체 구조를 갖는다. 따라서 파일 식별자를 두개의 해쉬 함수를 사용하여 공유파일의 키를 생성하고, 키를 담당하는 피어에게 {key, value}쌍을 삽입한다. 공유 파일의 {key, value} 삽입과정은 키를 기반으로 3.4의 라우팅 메커니즘을 수행한다.

3.7. 공유데이터 검색

컨텐츠 기반 확장 CAN 메커니즘에서의 공유 파일의 검색은 키워드를 이용한 검색방법과 파일 식별자의 정확한 일치를 통한 검색방법으로 나누어진다.

3.7.1. 다중 키워드 검색

컨텐츠 기반 확장 CAN 메커니즘에서는 공유 파일에 대해 다중 키워드검색이 가능하다. 다중 키워드 검색방법은 피어가 PLS로부터 전송받은 KID와 CKD를 이용하며 다중키워드 검색 메커니즘의 절차는 다음과 같다.

Step1) 사용자는 요청하고자 하는 공유파일을 유추할 수 있는 키워드들을 선정한다.

Step2) 선정된 키워드들을 불리언 연산자가 포함된 검색 메시지를 작성하여 질의를 요청한다. 불리언 연산자란 사용자가 검색어간의 관계성을 정의하는데 사용되며 본 제안 방안에서는 OR, AND, NOT의 세 가지 연산자가 사용된다.

Step3) 불리언 연산자가 포함된 검색메시지 질의가 요청되면 사용자의 요청과 가장 유사한 파일 식별자 목록을 추출하기위해 다음의 네 단계를 수행한다.

Step3.1) 다중키워드 검색 질의문에 포함된 각각의 키워드를 불리언 연산자 단위로 분리하고 각각을 SHA-1 해쉬 함수를 사용하여 해쉬한다. 그리고 불리언 연산자가 포함된 해쉬된 다중 키워드 형태의 검색질의를 생성한다.

Step3.2) 각각의 해쉬된 형태의 다중 키워드 중 CKD를 검색하여 공통키워드를 추출하고, 공통키워드를 제외한 형태로 검색질의를 재 작성한다. 공통키워드를 제외한 질의문의 작성은 키워드를 기반으로 KID에서 파일 식별자를 추출할 때의 처리수를 줄이기 위함이다.

Step3.3) 재 작성된 검색질의를 바탕으로 불리언 연산자 룰에 따라 KID를 검색하여 일치하는 {file-id, list of keyword}들을 추출한다.

Step3.4) 최종적으로 제외 되었던 공통키워드를 포함하여 KID를 통해 추출되었던 {file-id, list of keyword}와 비교하여 최종 파일 식별자 목록을 추출한다.

Step4) 상기의 절차를 통해 추출한 공유 파일의 파일 식별자를 두개의 해쉬 함수를 사용하여 공유파일의 키를 생성한다. 그리고 공유파일의 키를 기반으로 3.4의 라우팅 메커니즘을 수행하여 해당키를 담당하는 피어에게 검색 질의문을 라우팅한다.

Step5) 검색 질의문을 전송 받은 담당 피어는 공유 파일 해쉬 테이블의 {key, value}에서 키에 해당하는 실제 파일을 소유한 피어의 'value'(IP주소)를 전송한다.

Step6) 검색 요청피어는 실제 파일을 소유한 피어에게 파일의 송신을 요청한다. 파일의 송수신은 HTTP 프로토콜을 이용하여 1:1로 수행된다.

3.7.2. 파일 식별자를 통한 검색

파일 식별자를 통한 검색은 사용자가 요청하고자 하는 공유 파일의 파일 식별자를 정확히 알고 있거나 완전한 키워드 목록을 알고 있을 경우 사용되며 검색 메커니즘의 절차는 다음과 같다.

Step1) 사용자가 요청하고자하는 공유파일의 완전한 키워드 목록을 알고 있을 경우 더블쿼테이션마크(" ")안에 키워드를 나열하여 검색 메시지를 작성한다. 만약 요청하고자하는 공유 파일의

파일 식별자를 알고 있다면 쿼테이션마크(' ') 안에 파일 식별자를 입력하여 검색 메시지를 작성하여 질의를 요청한다.

Step2) 더블쿼테이션마크를 사용한 검색 메시지는 완전한 키워드 목록을 알고 있은 경우라 판단하여 3.3 컨텐츠 기반 파일 식별자 생성 메커니즘에 따라 각각의 키워드들을 SHA-1 해쉬 함수를 사용하여 이를 조합한 파일 식별자를 생성한다. 쿼테이션마크를 사용한 검색 메시지는 파일 식별자를 이미 알고 있으므로 두 번째 단계는 생략한다.

Step3) 상기 절차를 통해 획득한 파일 식별자를 기반으로 3.7.1의 다중키워드 검색의 Step4 절차 이하를 수행한다.

Ⅳ. 제안 시스템 검토 및 고찰

컨텐츠 기반 확장 CAN은 분산 해쉬 테이블 기반의 P2P 시스템은 4가지 측면에서 검토 및 고찰이 될 수 있다.

첫째, 명확한 키워드 선정을 위한 컨텐츠 기반의 키워드 및 파일 식별자 생성한다. 기존의 키워드 검색을 위한 역 분산 해쉬 테이블 기반의 메커니즘에서는 공유 파일의 키워드 선정 방법이 모호하여 해당 파일을 유추할 수 있는 몇 개의 단어들로 키워드를 선정 하였다. 이러한 키워드의 선정 방법은 동일파일에 대해 사용자에 따라 서로 다른 키워드 선정이 이루어지므로 키워드 및 공통 키워드의 증가를 가중시킨다. 따라서 본 논문에서는 공유파일에 대한 명확한 키워드 선정을 위해 컨텐츠 내용 기반의 키워드를 선정하고, 키워드의 조합을 통한 파일 식별자를 생성한다. 따라서 공유 파일 컨텐츠의 메타 정보에 따라 키워드가 생성되기 때문에 동일한 파일은 항상 같은 키워드들로 구성이 되어 키워드 선정의 명확성과 공통키워드 수의 감소를 갖게 된다.

둘째, 키워드를 통한 파일 식별자 검색 및 공통 키워드 처리를 위해 키워드 사전인 KID와 공통키워드 사전인 CKD를 사용한다. CKD는 공통키워드를 담고 있는 사전으로 PLS에서 분배 및 관리 한다. KID는 오버레이 네트워크에 존재하는 피어가 소유한 공유 파일들의 {file-id, list of keyword} 형태의 사전으로 키워드를 통한 파일 식별자 검색을 위해 사용된다. CKD는 키워드 기반 검색 질의 시 공통 키워드처리를 위한 사전으로 공통 키워드를 제외한 새로운 검색질의 작성을 위해 사용된다. CKD를 사용한 공통 키워드 처리를 통해 공통키워드를 담당하는 피어에게 과도한 질의를 요청받게 되는 문제를 해결할 수 있다.

셋째, KID, CKD, 피어 목록 분배 및 관리를 위해 PLS을 사용한다. 네트워크 참여를 위한 초기 접속서버인 PLS를 확장하여 KID와 CKD의 계산 및 분배, 관리를 위임함으로써 피어의 계산양을 감소시키고 피어간의 질의응답수를 줄일 수 있으며 KID와 CKD의 배포가 효율적이다.

넷째, 기본 CAN 메커니즘의 오버레이 네트워크 위상 및 기본 메커니즘을 그대로 활용하기 때문에 CAN의 장점이 그대로 수용되고, 제안 메커니즘의 적용이 용이하다.

Ⅴ. 결 론

본 논문에서는 효율적인 키워드검색이 가능한 컨텐츠 기반의 확장 CAN 메커니즘을 제안하였다. 이를 위해 컨텐츠 기반의 키워드 추출 및 파일 식별자 생성 메커니즘을 제안 하였고, 키워드 질의 및 공통키워드 처리를 위해 PLS를 확장하여 KID와 CKD 관리를 담당하게 하였다. 또한 피어의 키워드 기반 검색 시 검색 요청 피어 자신이 소유한 KID와 CKD를 기반으로 키워드 검색을 통해 공유 파일이 검색 가능하도록 하였다.

본 논문이 제안한 컨텐츠 기반 확장 CAN 메커니즘은 CAN 메커니즘뿐만 아니라 다른 분산 해쉬 테이블 기반의 구조적 P2P에서 적용하기 용이하여 이로 인해 구조적 P2P 응용 활성화에 도움이 될 것으로 본다.

향후 연구과제는 컨텐츠 기반 확장 CAN 메커니즘의 시뮬레이션을 통해 KID 및 CKD의 인덱스 비용을 줄이기 위한 임계값 선정과 공유 파일의 인기도를 키워드 검색에 포함시켜 검색속도를 향상시키는 방안을 제시하는 것이다.

참고문헌

[1] Napster, "http://www.napster.com"

[2] Gnutella, "http://genutella.wego.com"

[3] I. Stoica, R. Morris, D.Liben-Nowell, D.R.Karger, M. F. Kaashoek, F. Dabek, and H. Balakrishnan, "Chord: A Scalable Peer-to-peer Lookup Protocol for Internet Applications," In Proc. of SIGCOMM 2001, San Diego, CA, 2001.

[4] S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Shenker, "A Scalable Content Addressable Network," In Proc. of ACM SIGCOMM 2001, San Diego, CA, 2001.

[5] P. Reynolds, A. Vahdat, "Efficient Peer-to-Peer Keyword Searching," In Proc. of Middleware 2003, Rio de Janeiro, Brazil, 2003.

[6] Lintao Liu, Kyung Dong Ryu, Kang-Won Lee, "Keyword Fusion to Support Efficient Keyword-based Search in Peer-to-Peer File Sharing," In 4th Init'l Workshop on Global and P2P Computing, Chicago IL, April 2004.

저자소개

이 명 훈(Myoung-Hoon Lee)

2001년 배재대학교 컴퓨터공학과 학사
2003년 배재대학교 컴퓨터공학과 석사
2005년 ~ 현재 배재대학교 컴퓨터공학과 박사과정

※관심분야: 네트워크 보안, 차세대 네트워크

박 정 수(Myoung-Hoon Lee)

2004년 2월 배재대학교 정보통신공학부 공학사
2004년 3월~현재 배재대학교 컴퓨터공학과 석사과정

※관심분야: 정보보호, 컴퓨터네트워크

조 인 준(In-June Jo)

1982년 전남대학교 계산통계학과 공학사
1985년 전남대학교 전자계산학과 공학석사
1999년 아주대학교 컴퓨터공학과 공학박사
1983년~1994년 한국전자통신연구원 선임연구원
1994년~현재 배재대학교 컴퓨터공학과 교수

※관심분야: 정보보호, 컴퓨터네트워크, 전산조직응용