

P2P 환경에서의 이동 에이전트 기반 자원 검색 기법

신중수^o, 김문정, 엄영익

성균관대학교 정보통신공학부 분산시스템연구실

jongsoo77^o@hanmail.net, tops@ece.skku.ac.kr, yeom@ece.skku.ac.kr

A Mobile Agent Based Resource Discovery Mechanism in P2P Environments

Jong-soo Shin^o, Moon Jeong Kim, Young Ik Eom

School of Information and Communication Engineering, Sungkyunkwan University

요 약

최근 인터넷의 사용자 증가와 함께 사용자들의 요구는 다양해지고 있다. 또한 멀티미디어 콘텐츠도 사용자의 요구에 따라 점차 대용량화되고 있다. 멀티미디어 서비스 제공자는 사용자의 다양한 요구를 충족시키기 위해서 대용량 고품질의 콘텐츠를 많이 보유하고 있어야 한다. 그러나 클라이언트-서버 환경에서 대용량의 멀티미디어 콘텐츠 서비스를 제공하기 위해서는 고성능의 컴퓨터가 필요하다. 게다가 콘텐츠의 양과 사용자의 수가 증가함에 따라 중앙 서버로 부하가 집중되고, 신뢰성이 떨어진다는 문제점이 있다. 본 논문에서는 중앙으로 집중되는 요청을 분산시키고 대용량의 멀티미디어 콘텐츠를 저장하기 위해서 각 컴퓨터들을 P2P 환경으로 구축하고, 이동 에이전트를 이용하여 분산된 멀티미디어 콘텐츠를 검색하는 기법을 제시한다.

1. 서론

인터넷의 발전과 함께 사용자의 요구는 복잡하고 다양해지고 있다. 사용자는 서비스의 속도와 서비스를 통해 제공받는 콘텐츠의 질도 중요한 것으로 인식하고 있다. 이러한 사용자의 요구를 만족시키기 위해서 서비스 제공자는 대용량의 콘텐츠를 다양하게 보유하고 있어야 한다. 그러나 클라이언트-서버 환경에서 대용량의 콘텐츠들을 다양하게 보유하기 위해서는 서버 비용에 대한 부담이 따르게 된다. 게다가 사용자가 늘어나면 중앙 서버에 부하가 집중되어 서비스의 속도가 낮아지게 된다. 이러한 문제점을 해결하기 위해서는 다양한 멀티미디어 콘텐츠를 여러 개의 서버에 분산시켜 저장하고 분산된 콘텐츠를 관리하고 검색할 수 있는 기법이 필요하다.

본 논문에서는 이동 에이전트가 사용자의 요청에 따라 생성되어 이동하면서 콘텐츠를 검색하는 방식의 검색기법과 분산된 멀티미디어 콘텐츠를 관리하는 기법을 제안한다.

본 논문은 다음과 같이 구성된다. 2장에서는 제안기법에 관련된 연구내용을 소개한다. 3장에서는 제안하는 기법이 동작될 시스템 구조, 자료구조, 동작과정 그리고 알고리즘을 기술한다. 4장에서는 본 논문에서 제안한 기법을 평가한다. 5장에서는 본 논문의 결론과 보완점에 대해 기술한다.

2. 관련연구

2.1 이동 에이전트(mobile agent)

이동 에이전트란 지능을 가지고 자율적으로 이동하면서 노드에 독립적으로 작업을 수행할 수 있는 프로세스이다[1]. 또한 이동 에이전트는 특정 노드의 상태에 영향 받지 않으며, 통신상태에 관계없이 작업을 수행할 수 있다[2, 4].

2.2 P2P 환경(peer-to-peer environment)

P2P 환경이란 각 컴퓨터가 동등한 능력을 가지고 있어서 어떤 컴퓨터에서라도 통신 세션을 시작할 수 있는 통신 모델을 말한다. 또한 P2P 환경은 동등 계층 통신이라고도 하며 네트워크에 연결되어 있는 모든 컴퓨터들이 서로 대등한 관계에서 데이터나 주변장치 등을 공유할 수 있다[3].

P2P 환경의 종류에는 중앙관리 서버를 두어 자원을 관리하는 하이브리드 P2P(hybrid P2P) 방식과 중앙서버 없이 순수하게 노드들 간의 자원관리를 하는 순수 P2P(pure P2P) 방식이 있다[5].

2.3 P2P 환경에서의 자원 검색 기법

P2P 환경에서의 자원 검색 기법으로는 자신의 리스트에 있는 모든

피어로 질의를 하는 브로드캐스트(flooding broadcast of queries) 방식과, 자원이 존재할 가능성이 있는 특정 피어로만 질의를 하는 선택적 포워딩/라우팅(selective forwarding/routing) 방식이 있다.

다음으로는 냅스터(napster)와 같이 자원의 정보를 저장한 서버를 통해 검색하는 집중화된 인덱스와 저장소(centralized indexes and repositories) 방식과 중앙 서버가 없이 각 피어가 가진 자원의 정보를 공유하여 중앙 서버를 제거한 분산된 인덱스와 저장소(distributed indexes and repositories) 방식이 있다. 또한 모든 자원이 고유한 ID로 저장되어 검색속도가 매우 빠른 분산된 해시 테이블 네트워크(decentralized hash table network) 방식도 있으나 이 방식이 있다 [6, 7].

3. 제안기법

본 장에서는 이동 에이전트를 이용한 검색 기법과 콘텐츠의 추가 및 삭제기법을 제안한다. 먼저 본 제안 기법의 시스템 구성과 자료 구조에 대해서 설명한다. 다음으로 요청된 콘텐츠의 검색과 콘텐츠의 추가 및 삭제의 동작과정에 대해 기술한다. 마지막으로 동작과정에서 기술한 내용을 알고리즘으로 표현한다.

본 논문에서의 네트워크 구성은 이미 완료된 상태라고 가정한다. 또한 각 그룹을 구성하는 피어(peer)는 변하더라도, 그룹을 대표하는 피어는 변하지 않는다고 가정한다.

3.1 시스템 구성

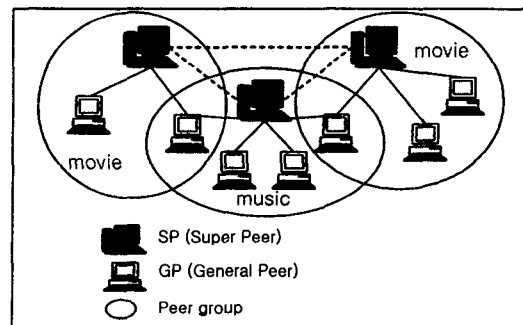


그림 1. 전체 시스템 구조

그림 1에서와 같이, 멀티미디어 서버는 콘텐츠의 타입에 따라 논리적인 그룹을 이루고 있다. 또한 각 피어는 하나 이상의 피어 그룹에 포함되어 있다. 피어는 SP(Super Peer)와 GP(General Peer)로 나눈다. 각 GP는 자신이 가지고 있는 콘텐츠에 대한 정보를 저장한 RPT(Resource Profile Table)와 RPT에 저장된 콘텐츠의 타입과 속한 그룹의 SP정보를 가진 SPI를 가지고 있다. 또한 GP는 자신의 RPT를 관리하는 RMA(Resource Management Agent)를 가지고 있다.

SP는 각 그룹 당 하나만 존재하는 피어로서, 그룹 내의 콘텐츠 정보를 가진 RPI(Resource Profile Index)와 다른 그룹에 대한 정보를 저장한 SPT(Super Peer Table)를 가지고 있다. GP는 그룹 내의 SP가 아닌 다른 피어를 말한다. QPA(Query Processing Agent)는 전체 시스템 안에서 콘텐츠를 검색하여 사용자에게 결과를 돌려주는 이종 에이전트로서, 사용자의 요청이 발생할 경우에 요청메시지를 받은 피어에서 생성된다.

3.2 자료구조

그림 2에서 제안기법에서 사용되는 테이블의 구조를 보인다.

(a) RPT	C_Name	F_name	Owner	Type	Author	Size	Pt	U
(b) SPI	Type	SP_Name	Count					
(c) RPI	Name	Author	Owner					
(d) SPT	SP_Name	Type						

그림 2. 테이블 구조

(a)의 RPT는 각 피어가 가지고 있는 콘텐츠의 정보를 유지한다. 검색이 일어날 경우 각 피어는 콘텐츠 종류별로 피어가 속해있는 그룹의 SP정보를 SPT에 유지하게 된다. U필드는 업데이트 유무를 나타낸다. 그룹 내의 피어가 가지고 있는 모든 콘텐츠의 정보는 RPI에 저장하며 RPI의 내용은 업데이트 메시지에 의해 갱신된다. SP는 다른 그룹의 정보를 SPT에 저장한다.

그림 3에서 제안기법에서 사용되는 메시지의 구조를 보인다.

Type	Size	Keyword1	Keyword2	...	
(e) 요청 메시지 (query message)					
Total	C_Name	F_Name	Owner	Author	Size
(f) 결과 메시지 (result message)					
Total	C_Name	Author	Owner	Type	U
(g) 업데이트 메시지 (update message)					
Q/R	Type	SP_Name			
(h) 그룹 요청/응답 메시지 (group query/response message)					

그림 3. 메시지 구조

클라이언트의 요청이 생기면 요청 메시지가 임의의 피어로 전달된다. 요청에 맞는 콘텐츠를 찾으면 결과 메시지를 클라이언트로 즉시 전송한다. 각 피어에서 콘텐츠가 변경되는 경우 GP는 변경사항을 SP에 알려주어야 한다. 변경사항은 업데이트 메시지로 RPI에 반영된다. 새로운 타입의 콘텐츠가 생기는 경우 GP는 해당 타입의 SP를 알고 있어야 한다.

(e)와 (f) 메시지의 Total필드는 복수개의 정보를 전송할 때 엔트리의 개수를 지정하는 필드이다. (h)에서 Q/R필드는 그룹 요청 메시지/그룹 응답 메시지를 구분하는 필드이다.

3.3 동작과정

3.3.1 요청된 콘텐츠의 검색

사용자는 검색하기 전에 검색하고자 하는 콘텐츠의 종류를 반드시 선택해야 한다.

사용자의 요청에 의한 검색과정은 그림4에서 보인다.

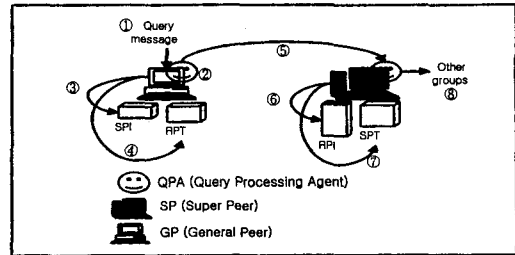


그림 4. 요청된 콘텐츠의 검색과정

그림 4와 같이, 사용자가 콘텐츠의 검색을 시작하면, 요청 메시지가 임의의 피어로 전달된다. 요청 메시지를 받은 피어에서는 QPA가 생성된다. 생성된 QPA는 현재 자신의 피어에 있는 SPI를 검색하여 해당 콘텐츠의 타입을 가지고 있는지 확인한다. SPI에 요청 메시지와 같은 타입의 엔트리가 없다면, 현재 QPA가 생성된 피어에는 사용자가 요청한 타입의 콘텐츠가 없다. 그러므로 사용자가 요청한 타입과 같은 그룹을 찾기 위해서 SPI의 가장 처음 엔트리의 SP로 이동하여 SPT를 검색한다. SPT의 검색결과는 사용자가 요청한 콘텐츠 타입과 같은 그룹들의 정보이므로 QPA는 SPT의 검색결과를 가지고 각 그룹을 이동하면서 검색한다.

QPA가 생성된 피어의 SPI에 요청한 콘텐츠 타입이 있다면 현재 피어에는 사용자가 요청한 타입과 같은 콘텐츠가 존재한다. 그러므로 QPA는 현재 피어의 RPT를 검색한다. 다음으로 QPA가 생성된 그룹 내의 다른 피어를 검색해야 한다. QPA는 SPI를 검색하여 요청메시지의 타입과 같은 그룹의 SP로 이동한다. SP로 이동한 QPA는 RPI를 검색하여 요청한 콘텐츠를 가진 피어로 이동하면서 각 피어의 RPT를 검색한다. 검색한 결과는 결과메시지로 사용자에게 즉시 보내준다. 그룹 내의 검색이 끝나면 사용자의 요청과 같은 다른 그룹이 있는지 확인해야 한다. 만약 다른 그룹이 존재한다면 다른 그룹도 검색해야 하기 때문이다. QPA는 검색하던 그룹의 SP로 돌아와서 SPT를 검색한다. QPA는 같은 타입의 다른 그룹에 대한 정보를 얻어서 다른 그룹의 SP로 이동한다. QPA가 이동하는 그룹은 사용자가 요청한 그룹이므로, RPI를 검색하여 그룹 내에서 사용자의 요청에 맞는 콘텐츠를 찾는다.

3.3.2 콘텐츠의 추가 삭제

새로운 콘텐츠의 추가와 삭제에 대한 동작과정은 그림 3에서 보인다.

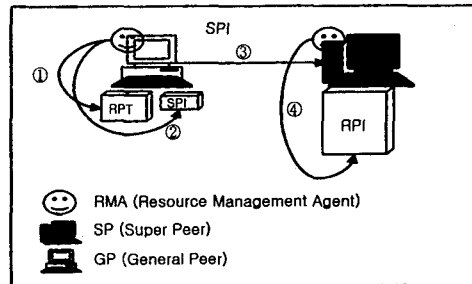


그림 5. 콘텐츠의 추가 삭제과정

그림 5과 같이 새로운 콘텐츠가 생기면 RMA는 새로 생기는 엔트리의 U필드를 1로 설정한다. 다음으로 RMA는 SPI에서 같은 타입에 해당하는 엔트리의 Count필드에 1을 더해준다. 엔트리가 없는 경우에는 새로운 엔트리를 만들고 Count필드에 1을 넣는다. 기존에 존재했던 콘텐츠가 삭제되는 경우 U필드를 2로 설정한다. SPI의 같은 타입에 해당하는 엔트리의 Count필드에서 1을 빼준다. 1을 뺀 후에 0이 되는 엔트리는 SPI에서 삭제한다. 새로 생성된 타입의 콘텐츠는 SP정보가 없다. 따라서 SPI의 가장 위의 엔트리로 그룹요청 메시지를 보내어 새로 생긴 콘텐츠 타입에 해당하는 SP정보를 얻는다. RMA는 RPT의 U필드가 1 또는 2로 설정된 엔트리의 변경사항을 SP에 전송한다. 이때, 변경된 콘텐츠의 종류에 따라 해당 SP로 그 내용을 전송한다. Update필드

가 1인 엔트리는 SP의 RPI에 추가되고, U필드가 2인 엔트리는 SP의 RPI를 검색하여 삭제시킨다.

3.4 알고리즘

3.4.1 요청된 콘텐츠의 검색과정

사용자로부터 요청 메시지를 받은 피어가 콘텐츠를 검색하는 과정을 알고리즘 1에서 보인다.

알고리즘 1. 요청된 콘텐츠 검색 알고리즘

```

if ( Type in SPI = Type in query message. ) {
  while ( !End of RPT )
    if ( Keyword compare with RPT ) send result messages to client;
  QPA moves to SP which is queried type;
  while ( !End of RPI )
    if ( Keyword compare with RPI ) save the peer names to P_info;
  run SearchInGroup routine;
  QPA go back to SP;
  while ( !End of SPT )
    if ( Type in query message = Type in SPT ) save SP_Names to S_info;
  run SearchOtherGroup routine;
} else {
  QPA moves to first entry in SPI;
  while ( !End of SPT )
    if ( Type in query message = Type in SPT ) save SP_Name in S_info;
  run SearchOtherGroup routine;
}
    
```

P_info는 QPA가 그룹 내에서 이동하게 될 피어들의 경로를 저장하고 S_info는 QPA가 이동하게 될 SP들의 경로를 저장하는 자료구조이다. S_info는 QPA가 가지고 있다. 알고리즘1의 요청된 콘텐츠의 검색 과정에서 QPA가 그룹 내의 피어들을 검색하는 과정을 알고리즘 2에서 보인다.

알고리즘 2. "SearchInGroup" 루틴

```

while ( !End of P_info ) {
  QPA moves to the peer that has the contents;
  while ( !End of RPT )
    if ( Keyword compare with RPT ) send result message to client;
  QPA moves other peer in group;
}
    
```

알고리즘 2와 같이 QPA는 RPI의 검색결과를 가지고 그룹 내의 피어들로 이동하면서 검색을 수행한다. 알고리즘 1에서 QPA가 다른 그룹을 검색하는 과정을 알고리즘 3에서 보인다.

알고리즘 3. "SearchOtherGroup" 루틴

```

while ( !End of SPT in QPA ) {
  QPA moves to SP
  while ( !End of RPI ) {
    if ( Keyword compare with RPI )
      run SearchInGroup routine;
  }
  QPA moves to other SP;
}
    
```

알고리즘 3과 같이 QPA는 SP에서 다른 그룹의 SP정보를 얻어서 이동한다. 다른 그룹에서의 검색과정은 SearchInGroup 루틴을 따른다.

3.4.2 새로운 콘텐츠의 생성

새로운 콘텐츠가 생성될 경우의 동작과정을 알고리즘 4에서 보인다.

알고리즘 4. 새로운 콘텐츠의 생성

```

set U in RPT to 1;
if ( SPI entry = type of new contents ) Count++;
else {
  make new entry in SPI;
  send group query message to first entry of SP in SPI;
  if ( check Type in SPT ) save SP_Name field;
  send back group response message to the primarily peer;
  put the SP information into the SP_Name;
}
send update message to the new SP
add contents information to RPI;
    
```

3.4.3 콘텐츠의 삭제

존재하던 콘텐츠가 삭제되었을 경우의 동작과정을 알고리즘 5에서 보인다.

알고리즘 5. 콘텐츠의 삭제

```

set U in RPT to 2;
Count-- in SPI;
if ( Count = 0 ) delete the entry in SPI;
send the update messages to the SP;
check Update field;
delete entry in RPI;
    
```

4. 평가

본 논문에서는 콘텐츠의 타입에 따라 각 피어들을 그룹화 하였다. 따라서 SP는 SP간의 정보만 주고받게 되고 GP는 그룹이 변경될 경우에만 SP와 메시지를 교환하게 된다.

각 그룹의 피어들은 서로 같은 타입의 콘텐츠를 가지게 되고, 검색 과정에서 QPA는 각 그룹을 대표하는 SP의 정보만을 가지고 모든 피어를 검색할 수 있다.

이동 에이전트는 사용자가 선택한 타입의 콘텐츠를 가진 그룹만을 이동한다. 이동 에이전트가 SP에 도착하면 RPI를 검색하여 콘텐츠를 가진 피어들을 이동하면서 결과를 사용자에게 보내준다. 따라서 에이전트의 불필요한 이동 없이 결과 메시지를 즉시 전송함으로써 검색속도의 향상을 기대할 수 있다.

5. 결론 및 향후 과제

본 논문은 P2P환경에서 이동에이전트를 이용하는 검색기법을 제안하였다. 제안기법은 콘텐츠의 종류별로 논리적인 그룹을 만들어서 네트워크를 구성하는 패킷의 수를 줄였다. 각 SP에 콘텐츠의 인덱스를 두어 에이전트가 이동하는 과정을 최적화하고 결과 값을 즉시 전송하도록 하여 사용자가 볼 수 있도록 했다. 그리고 검색결과는 최대한 많이 보여주고 사용자가 선택할 수 있도록 했다.

향후 연구과제로는 시뮬레이션을 통한 동작속도의 확실한 검증과정이 필요하다. 또한 이미 완료되어 있다고 가정된 시스템성이 동적으로 구성될 수 있도록 피어의 등록 및 삭제과정에 대한 연구도 병행되어야 할 것이다.

6. 참고문헌

- [1] V. A. Pham and A. Karmouch, "Mobile software Agents : An overview," IEEE Communications Magazine , Volume:36, Issue:7, Page(s) 26-37, July. 1998.
- [2] G. Cabri, L. Leonardi and F. Zambonelli, "Mobile agent coordination models for Internet applications," Computer , Volume: 33, Issue: 2, Page(s): 82-89, Feb. 2000.
- [3] M. Ripeanu, "Peer-to-Peer Architecture Case study: Gnutella Network," Peer-to-Peer Computing 2001, Page(s): 99-100, Aug. 2001.
- [4] "Mobile Agent Computing : A White Paper," Mitsubishi Electric ITA, Horizon System Lab, Jan. 1998.
- [5] J. Bligh, O. Callaghan, D. Corcoran, D. Aitken and J. Tobin, "Peer-to-Peer Technologieks and Protocols, 2001.
- [6] A. Oram, "Peer-to-Peer: Harnessing the Benefits of a Disruptive Technology," March. 2001.
- [7] <http://cubicmetercrystal.com/alpine/discovery.html>