韓國컴퓨터情報學會 論文誌
第16卷 第5號, 2011. 5.

2011-16-5-4-1

P2P 환경을 위한 허위 데이터 감축 정책

김 승 연*, 이 원 주**, 전 창 호***

False Data Reduction Strategy for P2P Environment

Seung Yun Kim*, Won Joo Lee**, Chang Ho Jeon***

요 약

본 논문에서는 P2P 환경에서 허위 데이터를 감축할 수 있는 FDR(False Data Reduction) 정책을 제안한다. 이 정책의 특징은 사용자가 허위 데이터를 인지하게 되면 그 파일을 다운로드 중인 다른 피어들에게 그 데이터 삭제를 요청한다. 또한 허위 데이터를 다운로드 중인 피어들에게 이를 통보함으로써 허위 데이터를 다운로드하지 않도록 하고, P2P환경에 퍼지지 않도록 허위 데이터를 삭제한다. 또한 이러한 모든 과정은 어떠한 검색 서버도 요구 되지 않고 오직 피어들 간에 정보 교환으로 이루어지기 때문에 검색 서버가 필요하지 않은 순수 P2P모델에 적용할 수 있다는 장점이 있다. 본 논문에서는 시뮬레이션을 통하여 FDR 정책이 네트워크 트래픽을 줄이는데 효과적임을 보인다. 임으로써 유효 데이터의 평균전송시간을 단축할 수 있음을 보인다. 그 결과 허위 데이터 비율에 따른 유효 데이터의 평균전송시간을 9.78~16.84% 단축함을 알 수 있었다.

▸ Keyword : 허위 데이터 감축, 피어, P2P 모델.

Abstract

In this paper, we propose a FDR(False Data Reduction) strategy for P2P environment that reduces false data. The key idea of our strategy is that we use FDR algorithm to stop transmitting of false data and to delete that. If a user recognizes false data in downloaded-data and the user's peer requests the others to stop the transmission of the false data immediately. Also, the FDR algorithm notifies the other peers to prohibit spreading of the false data in the environment. All this procedure is possible to be executed in each peer without any lookup server. The FDR algorithm needs only a little data exchange among peers. Through simulation, we show that it is more effective to reduce the network traffic than the previous P2P strategy. We also show that the proposed strategy improves the performance of network compared to previous P2P strategy. As a result, The FDR strategy is decreased 9.78 ~ 16.84% of mean true data transmission time.

▸ Keyword : FDR(False Data Reduction), Peer, P2P(Peer-to-Peer) model.

• 제1저자 : 김승연 교신저자 : 이원주
• 투고일 : 2011. 02. 07, 심사일 : 2011. 02. 19, 게재확정일 : 2011. 02. 21.
* 삼성전자반도체 연구소 CAE Team ICAD Group 선임연구원 (Samsung Electronics ICAD Group, CAE Team, Semiconductor R&D Center)
**인하공업전문대학 컴퓨터정보과 부교수(Dept. of Computer Science, Inha Technical College)
***한양대 전자컴퓨터공학부 교수(School of Electrical and Computer Engineering, Hanyang University)

Ⅰ. 서 론

초고속 인터넷의 보급과 스마트 폰, iPad, 갤럭시 탭 등과 같은 스마트 디바이스 사용자가 증가하면서 다양한 미디어 콘텐츠가 제작되고 있다. 이러한 미디어 콘텐츠는 P2P(Peer-to-Peer) 환경을 통하여 많은 사용자들에게 활용되고 있다. 하지만 악의적인 사용자들이 P2P 응용 프로그램이 가지고 있는 허위 데이터에 대한 취약점을 이용하여 허위 데이터의 이름이나 설명을 변경하여 인기 있는 콘텐츠에 대한 허위 데이터들을 생성하기도 한다. 또한 콘텐츠 저작권을 가진 업체에서 저작권을 보호하기 위한 하나의 방법으로 허위 데이터를 생성하는 정책을 사용하기도 한다. 위조된 허위 데이터들은 유효한 콘텐츠를 사용하는 P2P 사용자들에게 혼란을 주며, 불필요한 네트워크 트래픽을 증가시키는 원인이 된다.

P2P 환경에서는 콘텐츠의 진위 여부를 검증하는 것이 쉽지 않기 때문에 기존의 클라이언트-서버 환경에 비해 콘텐츠 관리에 어려운 점이 있다[1][2][3]. 또한, 위조된 콘텐츠는 다운로드를 완료하기 전에 P2P 사용자가 해당 콘텐츠의 진위 여부를 검증 할 수 없으며, 다운로드 받는 동안 다른 사용자에게 전송될 수 있다. 이러한 위조된 콘텐츠의 확산은 네트워크 트래픽 증가 원인이 되어 네트워크의 성능을 저하시킨다.

이러한 문제점을 해결하기 위해 본 논문에서는 P2P 환경에서 허위 데이터 확산에 따른 네트워크 트래픽 증가를 최소화 할 수 있는 알고리즘을 제안한다. 이 알고리즘의 특징은 새로운 데이터를 생성할 때마다 고유 ID를 부여하여 허위 데이터 생성을 막는다. 또한 각 피어들이 다운로드 받은 콘텐츠에 대한 경로를 관리함으로써 해당 콘텐츠가 허위 데이터로 판정되면 그 경로를 이용하여 허위 데이터를 삭제한다.

본 논문의 구성은 다음과 같다. 2장에서는 기존의 P2P 서비스에 대하여 소개하고 문제점을 설명한다. 3장에서는 제안한 FDR(False Data Reduction) 알고리즘에 대하여 자세히 설명한다. 그리고 4장에서 성능평가에 대하여 설명하고, 5장에서 결론을 맺는다.

Ⅱ. 관련 연구

2.1 P2P 서비스 종류

파일공유 서비스는 일반적으로 가장 많이 알려진 P2P 서비스로 사용자간 파일을 공유한다. 이 서비스는 사용자들이 공유한 파일을 검색하고, 검색된 파일 중에서 사용자가 원하는 파일을 특정 피어에서 다운로드한다[4]. 대표적인 예로는 냅스터(Naster)[5], 그누텔라(Gnutella)[6], 소리바다[7], 당나귀(eDonkey)[8], 카자(KaZaa)등이 있다. 파일공유 P2P의 시스템 구조는 다양하다. 순수한 P2P 파일 공유와 혼합형(Hybrid), 실시간 커뮤니케이션 P2P로 분류할 수 있다.

혼합형 P2P는 서버 기능 및 수에 따라 분류가 가능하다[9]. 실시간 커뮤니케이션 P2P는 커뮤니케이션에 초점을 둔 모델이다. 이 모델은 사용자들이 특정 서버에 로그인하여 커뮤니티를 형성하고, 온라인상에서 실시간으로 메시지를 교환한다. 서버에 등록된 메시지를 주고받을 때는 서버를 통하지 않고 클라이언트 간의 메시지 교환이 가능하다. 이러한 실시간 커뮤니케이션 P2P 서비스는 현재 메시지 교환, 전자우편, 파일전송 및 음성 채팅 서비스 등을 제공하고 있다. 대표적인 예로는 Microsoft사의 MSN messenger[10]와 Mirabilis사의 ICQ[11]가 있다.

2.2 파일 공유 P2P 분류

순수 P2P 모델은 그림 1과 같이 중앙 서버가 존재 하지 않기 때문에 모든 피어(Peer)가 클라이언트와 서버가 된다.

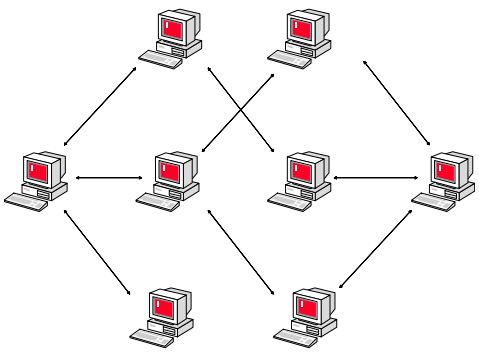

그림 1. 순수 P2P 모델
Fig. 1. Pure P2P model

순수 P2P 방식은 인터넷 접속과 동시에 Plug-and-Play와 같은 기능을 제공하여 인터넷 환경에서 효율적으로 작동한다. 장점은 모든 피어가 동등한 권한을 가지고 있기 때문에 단일 지점의 장애에 무관하다. 특정 서버에 의한 검색 방법이 아니기 때문에 P2P 시스템의 입장에서는 외부로부터의 특정 지점 공격에 대응이 쉽다. 하지만 데이터 검색을 위해 데이터의 생성과 색인 정보를 각 피어들이 유지 관리해야 하는 단점이 있다. 또한 데이터에 대한 검색을 제공하는 서버가 없기

때문에 각 피어에 직접 질의를 하는 방법을 사용하므로 응답 속도가 느리다.

순수 P2P 모델의 예로는 그누텔라와 프리넷(Freenet)이 있다[12]. 그누텔라의 경우 검색에는 플러딩(Flooding) 방식을 이용하기 때문에 네트워크에 부하를 많이 준다. 프리넷은 주변의 피어에게 질의를 하고 찾는 데이터가 없는 경우 다른 피어에게 질의를 한다. 검색에 의해 발생하는 트래픽을 막기 위해 횟수를 정해놓는다. 이러한 방법은 사용자가 찾는 데이터가 프리넷 환경에 존재하여도 검색에 실패할 수 있다.

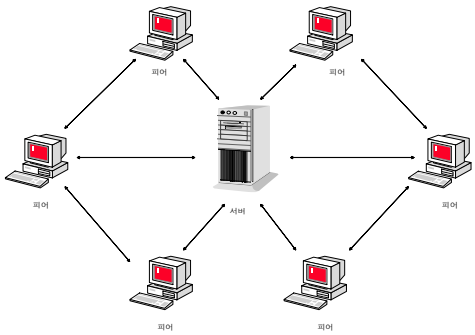

그림 2. 혼합형 P2P 모델
Fig. 2. Hybrid P2P model

그림 2와 같은 혼합형 모델은 서버가 제공하는 서비스와 서버의 수에 따라 3 가지로 분류 할 수 있다.

첫 번째 모델은 간단한 조회 기능 서버를 가진 P2P가 있다. 서버는 접속하는 피어에 이미 접속된 피어들의 이름을 알려준다. 피어는 서버에 로그인하면서 피어에 대한 존재를 서버의 피어 리스트에 등록하게 된다. 검색을 하거나 데이터를 전송하는 몫은 모두 피어들에게 있다.

두 번째 모델은 조회와 검색의 기능을 가진 서버를 사용하는 경우이다. 이 모델은 서버에서 조회 서비스와 동시에 검색 서비스를 제공한다. 검색 서버는 각각의 피어들이 가지고 있는 데이터의 목록을 가지고 다른 피어들에게 데이터의 검색에 도움을 준다. 피어들은 필요한 데이터를 검색할 때 다른 피어에 접속하지 않고 검색 서버를 이용하기 때문에 빠른 검색이 가능하다. 서버에서 검색된 내용을 바탕으로 피어 간의 데이터 전송이 이루어지게 된다. 현재 상용화된 파일공유 P2P 시스템 중에 가장 보편화 되어 있다.

세 번째 모델은 조회, 검색, 콘텐츠 제공 기능을 제공하는 서버를 가진 P2P 모델이다. 기존의 P2P 모델과는 달리 모든 P2P 시스템 관리를 서버에서 한다. 모든 자원이 중앙에 위치한 서버의 데이터베이스에 저장되어 있다. 특정 피어가 데이터를 요구 하게 되는 경우 다른 피어와 통신을 하는 것이 아니라 서버에게 그 데이터에 대한 요구를 한다. 즉, 피어가 필요로 하는 데이터 전송은 서버에서 하게 된다. 이 모델의 경우 서버에 많은 부담을 준다는 단점이 있다. 또한 중앙 서버에 전적으로 의존하기 때문에 단일 지점의 장애로 인한 P2P 시스템 전체의 기능이 마비되는 단점이 있다. 그림 2와 같은 혼합형 모델의 예로는 냅스터(Napster)가 있다. 냅스터는 중앙 서버에서 음악파일의 검색에 대한 정보를 제공한다.

2.3 기존의 P2P 응응 프로그램의 문제점

일반적으로 P2P 응용 프로그램은 그림 3과 같이 데이터를 전송 할 때 다운로드를 종료하지 않아도 다른 피어(peer)에 전송할 수 있도록 설계되어 있다.

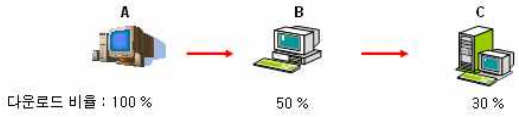

그림 3. P2P 환경의 데이터 전송 모델
Fig. 3. P2P Environment of Data Transfer Model

즉, 피어 B는 피어 A로부터 데이터를 50% 다운로드 받은 상태에서 받은 데이터의 진위 여부를 판정하지 않고 피어 C로 재전송한다. 피어 B에서 데이터를 100% 다운로드한 후에 해당 데이터가 허위로 판정되면 서버에 보고한다. 기존의 P2P 응용 프로그램에서는 피어 C로 전송중인 데이터를 중단할 수 없다. 이러한 P2P 환경의 데이터 전송 모델의 문제로 인해 허위 데이터의 확산은 증가한다. 또한 불필요한 네트워크 트래픽이 증가한다.

기존의 P2P 응용 프로그램은 허위 데이터에 대한 대응책이 부족하다. 특히 Pruna[13]는 허위 데이터에 대한 대응전략으로 허위 데이터 신고 기능을 사용하고 있다. 즉, P2P 사용자가 다운로드 받은 데이터가 허위 데이터로 판정되면 서버에 허위 데이터임을 신고하고, 다른 사용자들로 하여금 해당 데이터에 대한 허위 데이터 신고 수를 참조하게 함으로써 허위 데이터의 확산을 방지한다. 하지만 허위 데이터에 대한 정보가 사용자에게 실시간으로 제공되지 않기 때문에 사용자가 데이터를 검색한 이후에 허위 데이터로 판정되는 경우에는 허위 데이터 확산을 방지 할 수 없다는 단점이 있다. 그리고 허위 데이터로 확인되어도 그 데이터들은 P2P 시스템에 존재함에 따라 허위 데이터의 확산은 계속되어 네트워크 트래픽을 증가시킨다.

III. 허위 데이터 감축 알고리즘

본 논문에서는 P2P 환경에서 허위 데이터 확산으로 인한 네트워크 트래픽 증가를 최소화 할 수 있는 FDR(False Data Reduction) 알고리즘을 제안한다. 이 알고리즘은 먼저 P2P 환경에서 생성되는 데이터에 고유 ID를 부여하여 허위 데이터의 생성을 방지한다. 또한 각 피어들이 다운로드 받은 콘텐츠에 대한 경로를 저장함으로써 해당 콘텐츠가 허위 데이터로 판정되면 그 경로를 이용하여 허위 데이터를 삭제한다.

3.1 고유 ID 부여

기존의 파일명 중심의 검색 방법을 사용하는 P2P 시스템에서는 사용자가 파일명을 변경하는 것이 매우 쉽기 때문에 허위 데이터를 생성하기가 용이하다. 따라서 기존의 파일명 중심의 검색 방법보다는 미디어 콘텐츠의 파일 헤더에 저장되어 있는 제목과 저자, 파일 크기 등의 정보를 이용하여 새로운 ID를 부여하고 그 ID를 검색하는 방법을 택한다. 이때 악의적인 사용자가 허위 데이터로 판명된 파일에 대해 다른 이름으로 다시 공유하는 것을 방지하기 위해 같은 파일에 대하여 동일한 ID를 부여하여 사용한다.

3.2 FDR(False Data Reduction) 알고리즘

FDR 알고리즘에서는 여러 단계로 전송 중인 데이터에 대해서 허위로 확인된 데이터를 삭제하기 위해 데이터의 전송 경로를 유지한다.

각 피어는 데이터 전송 경로를 유지하기 위해 데이터 전송에 참여하는 피어들 간에 데이터 전송 경로 정보를 교환하여 저장한다. 데이터 전송에 참여하는 각 피어는 송신 피어의 주소와 해당 파일을 요청한 수신 피어의 주소를 순서대로 저장한다. 그리고 송신 피어는 수신 피어에게 데이터 전송을 시작하기 전에 생성된 데이터 전송 경로를 알려준다. 데이터를 받는 수신 피어는 그 정보를 통해 데이터가 어떤 피어로부터 전송되었는지 알 수 있다. 데이터를 받고 있는 수신 피어에 또 다른 새로운 피어가 그 파일에 대해 전송 요청을 하게 되면 데이터를 받고 있는 피어에 저장된 데이터 전송 경로에 새로운 피어의 주소를 추가한다. 그리고 그 추가된 경로를 데이터 요청 피어에 전송하고, 실제 유효한 데이터를 전송한다. 데이터 경로가 완성되면 마지막 피어는 자신의 데이터 경로에 있는 첫 번째 피어에게 데이터 경로를 복사한다. 마지막 피어가 자신의 데이터 경로를 복사하는 이유는 피어가 데이터 전송 경로상의 피어가 다운되었거나 네트워크 접속에 실패할 경우 데이터 경로를 따라 제어를 하는 것이 불가능하기 때문이다.

데이터 전송 경로 생성 예제는 그림 4와 같다.

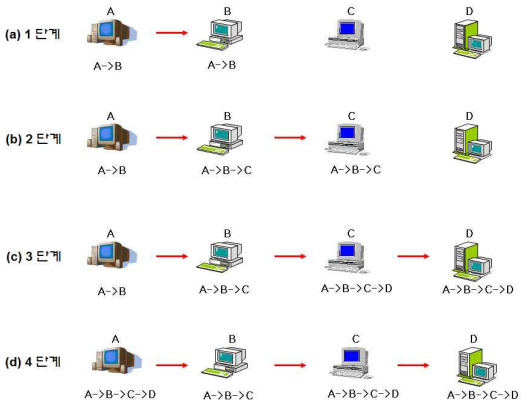

그림 4. 데이터 전송 경로 생성 예
Fig. 4. Example of data transfer path creation.

그림 4에서 4 개의 피어 A, B, C, D가 존재한다고 가정하고 데이터는 A->B->C->D로 전송된다고 가정한다. (a)1단계에서 A->B로 데이터 전송이 시작되면 전송중인 데이터에 대해 고유 ID와 함께 전송 경로를 A와 B에 저장한다. 그리고 (b)2단계에서 C가 B에게 데이터 전송요청을 하면 송신하는 B와 수신하는 C 의 경로를 서로 수정한다. (c)3단계에서도 (b)2단계와 같이 전송 경로를 수정하고 다시 D가 C에게 전송을 요청하는 경우 각 C와 D의 전송경로를 수정한다. (d)4단계에서는 노드 실패에 대응할 수 있게 마지막 피어 D가 가지고 있는 전체 데이터 전송경로를 초기 피어 A에 복사함으로써 데이터 전송 경로는 완성된다.

그림 4와 같이 파일 전송할 때는 먼저 데이터 전송 경로를 생성한 후 파일을 전송한다. 파일 송신 알고리즘은 그림 5와 같다.

```
//파일 송신 (FID : File ID, PID : Peer ID)
void File_Sending (FID, receiver_PID)
{
   // 데이터 경로(data path) 설정
   Make_DataPath (FID, receiver_PID);
   Send_DataPath (FID, receiver_PID);
   //유효 데이터 전송
   Send_Data (FID, receiver_PID);
}
```

그림 5. 파일 송신 알고리즘
Fig. 5. File sending algorithm

그림 5의 Make_DataPath(FID, receiver_PID)에서는 파일 전송 전에 파일 전송을 요청한 수신 피어의 ID를 데이터 경로에 추가 한다. Send_DataPath (FID, receiver_PID)는 완성된 데이터 경로를 파일 수신 피어에 전달하고, Send_Data(FID, receiver_PID)에서 요청한 유효 데이터를 전송한다.

```
//파일 수신 (FID : File ID, PID : Peer ID)
void File_Receiving (FID, sender_PID)
{
  Request_file(FID, receiver_PID); //파일 요청
  //데이터 전송 요청 피어의 데이터 경로 수신
  Receiving_sender_DataPath(FID, sender_PID);
  //피어의 데이터 경로 수정
  Update_DataPath(FID, PID, data_path);
  if (isLastPeer) {  //마지막 피어인 경우
    //자신의 전송경로를 최초의 peer에 전송
    send_DataPath(FID, get_source_peer());
  }
  //유효 데이터 수신
  receive_data (FID, receiver_PID);
}
```

그림 6. 파일 수신 알고리즘
Fig. 6. File receiving algorithm

그림 6은 파일을 요청한 수신 피어의 파일 수신 알고리즘이다. Request_file(FID, receiver_PID)은 파일을 요청할 때 요청하는 피어의 ID와 원하는 file ID를 보낸다. Receiving_sender_DataPath (FID, sender_PID)는 파일 송신 피어에서 데이터 경로를 완성하여 파일 전송을 요청한 수신 피어에 전달한다. Update_DataPath(FID, PID, data_path)에서는 데이터 경로를 전달받은 피어의 데이터 경로를 수정한다. 그리고 파일 수신 피어가 경로의 마지막 피어인 경우, send_DataPath(FID, get_source_peer())에서 그 피어에 저장된 데이터 경로와 파일 전송중인 피어에 대한 모든 정보를 최초 피어에 전송한다. receive_data (FID, receiver_PID)에서는 유효한 데이터를 수신한다.

이러한 데이터 전송 경로 유지 방법은 각 피어에 데이터 전송 경로 정보를 저장 관리함으로써 전송중인 데이터가 허위 데이터로 확인되면 확인된 위치에서 데이터 전송 경로의 역방향에 따라 각 피어에서 해당 데이터를 삭제할 수 있다. 이때 허위 데이터를 삭제하는 알고리즘은 그림 7과 같다.

```
// FID : File ID, PID : Peer ID
void  Delete_Faike_File(FID)
{
  //데이터 경로상의 피어들의 존재 유무 확인
  if(data_path_valid (FID, PID, data_path)){
     if(is_last_peer){ //마지막 피어인지 확인
       delete_DataPath(file_ID); //데이터 경로 삭제
     }else{
       abort_sending(FID);        //전송 취소
       delete_file(FID);          //파일 삭제
       //앞 Peer에 허위 데이터 삭제 요청 신호 전달
       send_false_data_signal(previous_PID);
       delete_data_path(FID);  //데이터 경로 삭제
     }
  }
}
```

그림 7. 허위 데이터 삭제 알고리즘
Fig. 7. False Data Reduction Algorithm

그림 7에서는 먼저 데이터 전송 경로가 유효한 경로인지 파악한다. 유효한 경로이면 데이터 전송 경로의 마지막 피어인지 확인한다. 만약 마지막 피어이면 delete_DataPath (file_ID)에서 데이터 경로를 삭제한다. 하지만 마지막 피어가 아니면 abort_sending(FID)에서 데이터 전송을 취소하고, delete_file(FID)에서 파일을 삭제한다. 그리고 send_false_data_signal(previous_PID)에서 이전 피어에게 데이터 삭제 신호를 보내고, delete_data_path(FID)에서 데이터 경로를 삭제함으로써 역방향으로 데이터를 삭제한다.

그림 4를 예로 들어 허위 데이터 삭제 과정을 설명한다. (d)4 단계의 피어 C에서 다운로드 받은 데이터가 허위 데이터로 확인되면 검색 서버에 신고하고, 허위 데이터를 삭제한다. 그리고 데이터 전송경로에 따라 피어 C로 데이터를 전송한 피어 B에 허위 데이터임을 알린다. 따라서 피어 B는 자신의 허위 데이터를 삭제한 후 데이터 전송경로에 따라 피어 A에게 알린다. 이러한 과정으로 데이터 전송 경로 상에 나타난 모든 피어에 존재하는 허위 데이터를 삭제한다.

IV. 성능 평가

P2P 환경에서 허위 데이터 증가는 네트워크 트래픽을 증가시키기 때문에 각 피어의 유효 콘텐츠 전송시간을 증가시킨다. 성능 평가에서는 FDR 알고리즘이 허위 데이터를 삭제하여 네트워크 트래픽 증가를 최소화함으로써 각 피어의 유효 콘텐츠 전송시간을 감소시키는 것을 보인다. 본 논문에서는 FDR 알고리즘의 성능 평가를 위해 시뮬레이터로 NS-2를 사용한다. 성능평가를 위한 시뮬레이션 환경은 표 1과 같다.

표 1. 시뮬레이션 환경
Table 1. Simulation environment

하드웨어 사양	• CPU : Pentium4 2.0 • 메모리 : 768 MB
소프트웨어 사양	• 운영체제 : Red Hat 9 • 리눅스 커널 버전 : 2.4.2 • NS-2 버전 : 2.27 • TCL 버전 : 8.4.5 • Xgraph 버전 : 12.1

본 논문에서는 평균전송시간을 성능 평가 척도로 사용한다. 평균전송시간은 유효 데이터를 전송하는데 소요되는 평균 시간이다.

그림 8. FDR 알고리즘 미적용의 경우 5개 파일 전송시간
Fig. 8. Transmission time of 5 files without false data reduction algorithm

먼저 5개의 데이터에서 유효 데이터 1개(허위 데이터 80%)일 때 FDR 알고리즘을 적용하지 않고 유효 데이터를 다운로드하는데 소요되는 전송시간을 측정한 결과는 그림 8과 같다.

첫 번째 파일이 유효 데이터 인 경우 전송시간은 244초, 두 번째 파일이 유효 데이터 인 경우 전송시간은 222초, 세 번째 파일이 유효 데이터 인 경우 전송시간은 215초 네 번째 파일이 유효 데이터 인 경우 전송시간은 223초, 다섯 번째 파일이 유효 데이터인 경우 전송시간은 236초이다. 따라서 유효 데이터를 다운로드하는데 소요되는 평균전송시간은 228초임을 알 수 있다.

FDR 알고리즘을 적용하여 허위 데이터 비율에 따른 평균전송시간을 측정한 결과는 그림 9와 같다.

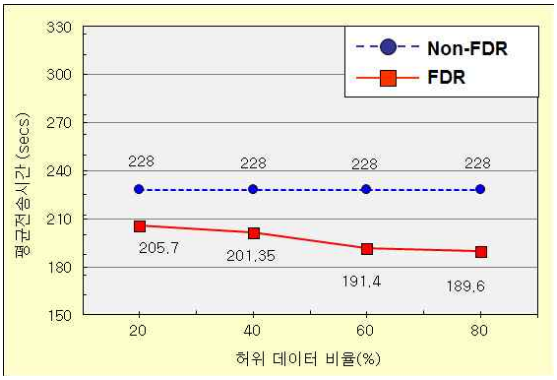

그림 9. 허위 데이터 비율에 따른 평균전송시간
Fig. 9. False data rate vs. Mean transmission time

그림 9에서 FDR은 FDR 알고리즘을 적용한 결과이고 Non-FDR은 FDR 알고리즘을 적용하지 않은 결과이다. 그림 9에서 Non-FDR은 허위 데이터 비율에 관계없이 일정함을 보인다. 하지만 FDR은 허위 데이터 비율이 증가할수록 평균전송시간이 감소함을 보인다. 따라서 허위 데이터 비율이 증가할수록 평균전송시간 면에서 FDR 알고리즘의 성능이 향상됨을 알 수 있다.

다음은 허위 데이터 비율에 따른 평균전송시간 감소율을 구한 결과는 그림 10과 같다.

그림 10을 살펴보면 허위 데이터 비율이 20인 경우 FDR의 평균전송시간은 Non-FDR에 비해 9.78% 감소함을 알 수 있다. 또한 허위 데이터 비율이 80인 경우 FDR의 평균전송시간은 Non-FDR에 비해 16.84% 감소했음을 알 수 있다. 따라서 허위 데이터 비율이 증가할수록 평균전송시간 면에서 FDR 알고리즘의 성능이 향상됨을 알 수 있다.

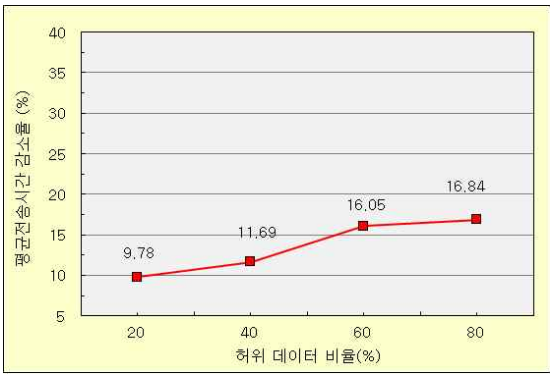

그림 10. 허위 데이터 비율에 따른 평균전송시간 감소율
Fig. 10. False data rate vs. Mean transmission time reduction rate

V. 결론

본 논문에서는 P2P환경에서 허위 데이터를 삭제하여 네트워크 트래픽의 증가를 최소화 하는 FDR 알고리즘을 제안한다. 이 알고리즘의 특징은 여러 피어에 걸쳐 전송 중인 데이터가 허위 데이터로 확인되는 경우 각 피어를 경유하는 전송을 취소함으로써 불필요하게 발생하는 네트워크 트래픽을 최소화한다. 또한 피어들 간의 데이터 전송 경로를 유지하고 허위 데이터를 삭제함으로써 특정 검색 서버에 어떠한 접근도 없기 때문에 검색 서버가 없는 파일공유 P2P에 적용할 수 있다. 본 논문에서는 시뮬레이션을 통하여 FDR 알고리즘의 성능을 검증하였다. 시뮬레이션에서는 FDR 알고리즘이 허위 데이터를 삭제하여 네트워크 트래픽의 증가를 방지함으로써 유효 데이터의 평균전송시간을 단축할 수 있음을 보였다. 시뮬레이션 결과 유효 데이터의 평균전송시간이 9.78~16.84% 단축됨을 알 수 있었다. 또한, 전체 데이터에서 유효 데이터의 비율이 높을수록 FDR 알고리즘의 성능이 향상된다는 것을 알 수 있었다.

향후 연구에서는 악의적인 사용자에 의해 유효 데이터가 삭제될 수도 있기 때문에 파일 전송 경로 상의 피어 수, 허위 데이터 신고 횟수, 파일 등록일, 검색횟수, 다운로드횟수 등의 정보를 고려한 허위 데이터 감축 알고리즘을 개발하고자 한다.

참고문헌

[1] Young-Jin Kim, Young Ik Eom, "Discovery Mechanisms for P2P Computing Environments," Communications of Korean Institute of Information Scientists and Engineers, Vol. 22, No. 3, pp. 6-17, Mar. 2004.

[2] Lan Quan, Kyung Geun Lee, "File Sharing in Unstructured Peer-to-Peer Net works," Communications of Korean Institute of Information Scientists and Engineers, Vol. 22, No. 3, pp. 35-44, Mar. 2004.

[3] Minho Shin, William A. Arbaugh, "Efficient Peer-to-Peer Lookup in Multi-hop Wireless Networks," KSII Transactions on Internet and Information Systems, Vol. 3, No. 1, pp. 5-25, Feb. 2009.

[4] Sang Won Oh, Won Joo Lee, Chang Ho Jeon, "An Efficient Peer-to-Peer Based Replication Strategy for Data Grid," Journal of the Institute Electronics Engineers of Korea, Vol. 45, Computer and Information, No. 2, pp. 10-17, Mar. 2008.

[5] *http://www.napster.com/*

[6] Ripeanu, M., "Peer-to-peer architecture case study: Gnutella network," Peer-to-Peer Computing, 2001. Proceedings First International Conference on 27-29 Aug. 2001.

[7] *http://soribada.com/*

[8] *http://www.edonkey2000.com/*

[9] Dreamtech Software Team "Cracking the Code : Peer to Peer Application Development," 2001 Hungry Minds Inc.

[10] *http://messenger.msn.com/*

[11] *http://www.icq.com/*

[12] Clarke, I., Sandberg, O., Wiley, B., and Hong, T.W., "Freenet: A distributed anonymous information storage and retrieval system," in International Workshop on Designing Privacy Enhancing Technologies, Berkeley, CA. Springer-verlag, Heidelberg, 2000.

[13] *http://www.pruna.com/*

저 자 소 개

김 승 연

2003: 한양대학교
전자 컴퓨터 공학부 공학사.
2005: 한양대학교
컴퓨터공학과 공학석사.
2005. 2-2010. 7: Dupont
Photomask Data
Preparation Engineer
현 재: 삼성전자 반도체 연구소 CAE
team ICAD Group 선임 연구원
관심분야 : 클라우드컴퓨팅, Grid컴퓨팅,
성능분석
Email: becrazy@gmail.com

이 원 주

1989: 한양대학교
전자계산학과 공학사.
1991: 한양대학교
컴퓨터공학과 공학석사.
2004: 한양대학교
컴퓨터공학과 공학박사.
현 재: 인하공업전문대학 컴퓨터정보과
부교수.
관심분야: 병렬처리시스템, 성능분석,
Grid컴퓨팅, 클라우드컴퓨팅
Email: wonjoo2@inhatc.ac.kr

전 창 호

1977: 한양대학교
전자공학과 학사.
1982: Cornell University
컴퓨터공학과 석사.
1986: Cornell University
컴퓨터공학과 박사.
1977-1979: 전자통신연구소
연구원.
현 재 : 한양대학교 전자컴퓨터공학부
교수.
관심분야: 병렬처리시스템, 성능분석,
Grid컴퓨팅, 클라우드컴퓨팅
Email: chj5193@hanyang.ac.kr