다중 소스로부터 다운로드가 가능한 P2P 시스템에서 고의적 변조 파일의 배제 기법°

류중희, 김유나, 김 종
포항공과대학교 컴퓨터공학과
e-mail : {foresee, existion, jkim}@postech.ac.kr

Excluding Forged Files from Multi-Source Downloadable P2P Systems°

Junghei You, Yuna Kim, Jong Kim
Dept. of Computer Science and Engineering
Pohang University of Science and Technology (POSTECH)

요 약

최근 P2P 파일 공유 시스템은 다운로드 속도를 높이기 위해 한 파일을 블록 단위로 나누어 여러 피어로부터 동시에 내려 받는다. 그러나 악의적인 피어가 고의적으로 변조한 파일을 전송 받게 되면 해당 블록의 재전송으로 네트워크 자원이 소모되고, 블록들을 재 조합하여 파일을 구성하는데 걸리는 시간이 증가한다. 본 연구에서는 자원 절약을 위해 고의적 변조 파일을 P2P 시스템에서 배제시키는 방법을 제안하고자 한다. 제안하는 방법은 디렉토리 서비스를 담당하는 피어가 변조 블록에 대한 신고를 받아 그 신고 횟수가 임계치에 도달하면, 해당 변조 블록을 포함한 파일을 검색 결과에서 배제시킴으로써 P2P 상에 공유되는 것을 막는다. 또한 제안하는 시스템의 성능을 모의 실험을 통해 평가한 결과, 일반적인 P2P 시스템 및 피어 평판 관리 시스템을 적용한 P2P 시스템보다 제안한 시스템의 고의적 변조 파일의 공유 비율이 각각 22 배, 4 배 낮고, 올바른 파일은 항상 공유되는 것을 확인하였다. 그러므로 제안하는 방법을 기존 P2P 시스템에 적용할 경우, 고의적 변조 파일의 다운로딩에서 발생하는 네트워크 자원 낭비를 줄일 수 있고 피어의 다운로드 속도도 증가할 것이다.

1. 서론

최근에는 다운로드 속도를 높이기 위해, 하나의 파일을 여러 개의 블록으로 나누어 서로 다른 피어로부터 동시에 내려 받는 P2P 시스템이 널리 이용되고 있다[1-4]. 이러한 P2P 시스템에서는 여러 소스로부터 블록 단위로 내려 받은 후에 블록들을 재 조합하여 파일을 완성한다. 이 과정에서 변조된 블록을 다운로드 하였을 경우에는 파일 재조합 과정이 완료되지 않고, 완성이 될 때까지 해당 블록을 재전송 받는다.

블록의 변조가 네트워크 전송 중에 발생한 일시적인 오류 때문이라면 쉽게 복구할 수 있다. 그러나 악의적인 피어가 고의적으로 변조하여 공유한 블록이라면, 변조 블록을 지속적으로 재전송 받아야 하기 때문에 네트워크 자원의 낭비와 클라이언트 시스템 사용량 증가를 초래하고 더불어 파일 다운로드 시간이 연장된다. 여기서 *고의적 변조 블록*이란, 블록의 내용을 변경 하였지만 블록의 무결성을 검사를 위한 검사합(checksum)은 변경되기 전 상태의 해쉬 값으로 설정되어 있는 블록이다. *고의적 변조 파일*은 고의적 변조 블록을 포함한 파일을 뜻한다.

한편 악의적인 피어를 구별하고 품질이 낮은 파일 공유를 막기 위해, 평판 관리(reputation management) 시스템에 관한 연구가 활발히 진행되고 있다[5-8]. 평판 관리 시스템은 크게 피어 평판 관리와 파일 평판 관리의 두 가지로 나눌 수 있다. 피어 평판 관리 시스템[5,6,8]은 피어 단위로 평판을 관리하는 것이고, 파일 평판 관리 시스템[7]은 피어 단위 및 파일 단위로 평판을 관리하는 것이다. 그러나 파일 평판 관리 시스템에서는 평판도가 높은 파일을 변조한다면 이를 탐지하기 어렵다. 그리고 악의적인 피어는 고의적 변조 파일 뿐 아니라 올바른 파일도 공유하고 있기 때문에, 피어 평판 관리 시스템에서는 악의적인 피어를 탐지하는 것이 어렵고 탐지하더라도 해당 피어가 제공하는 변조 파일만 배제시키는 것이 어렵다. 그러므로 고의적 변조 파일을 탐지하고 이를 P2P 파일 공유 시스템으로부터 배제시키는 새로운 방법이 필요하다.

본 논문에서는 네트워크 및 피어 시스템의 자원 절약을 위해, P2P 파일 공유 시스템에서 고의적 변조 파

° 본 연구는 정보통신부 및 정보통신연구진흥원의 대학 IT연구센터 지원사업의 연구결과로 수행되었음. (IITA-2007-(C1090-0701-0045))

일을 배제시키는 방법을 제안하고자 한다. 제안하는 방법은 탐지된 고의적 변조 파일을 검색 결과에서 배제시킴으로써 P2P 상에서 공유되는 것을 막는다. 또한 악의적인 피어가 공유하는 파일이라도 올바른 파일이라면 배제되지 않음을 보장한다.

본 논문의 구성은 다음과 같다. 2 장은 관련 연구를 살펴보고, 3 장에서는 제안하는 고의적 변조 파일을 배제하는 방법을 설명하고, 4 장에서는 성능평가를 위한 모의 실험의 결과를 보인다. 마지막으로 5 장에서는 결론을 맺고 향후 연구방향을 제시한다.

2. 관련 연구

2.1. eMule

eMule[1]은 2002 년부터 공개 프로젝트로 시작하여 사용자가 다종의 공유 네트워크를 연결하여 이용할 수 있도록 지원하는 대표적인 P2P 파일 공유 클라이언트 프로그램이다. eMule 은 여러 피어로부터 하나의 파일을 동시에 내려 받는 것이 가능하고, 다운로드 중에도 파일의 일부를 공유할 수 있으며, 해쉬 알고리즘을 이용하여 파일의 변조를 탐지할 수 있다.

공유되는 파일은 9.28MB 크기의 파트들로 나눠지며 하나의 파트는 다시 180KB 크기의 블록들로 나눠진다. 파일은 파트단위로 동시에 다운로드 되고, 다운로드가 완료된 파트는 파트해쉬와 비교하여 변조 유무를 검사한다. 만일 파트가 변조 되었다면 AICH (Advanced Intelligent Corruption Handling) 알고리즘[9]을 이용하여 그 파트 중 변조된 블록들을 찾아내어 그 블록들만 다시 다운로드 하여, 파일을 복구한다.

2.2. 평판 관리 시스템

P2P 상에서 피어의 악의성 여부를 판단하기 위해 다른 피어들의 경험과 피드백을 바탕으로 피어의 평판도 (reputation, 피어가 공유하는 파일의 품질 만족도를 나타냄)와 신뢰도(credibility, 피어의 피드백의 신뢰성을 나타냄)를 관리하는 피어 평판 시스템[5,6]이 제시되었다.

Mekouar et al.[5]가 제시한 시스템에서는 업로딩 피어 P_j 의 담당 수퍼피어가 P_j 의 평판도 AB_j (수식 1)를 계산하고 관리한다. 수식 (1)에서 $D_{*,j}^{+}$는 P_j 가 업로드한 파일중에 긍정적인 평가를 받은 파일의 크기, $D_{*,j}^{-}$는 P_j 가 업로드한 파일중에 부정적인 평가를 받은 파일의 크기이다. 다운로딩 피어는 업로딩 피어의 평판도에 근거하여 다운로드할 파일을 결정한다.

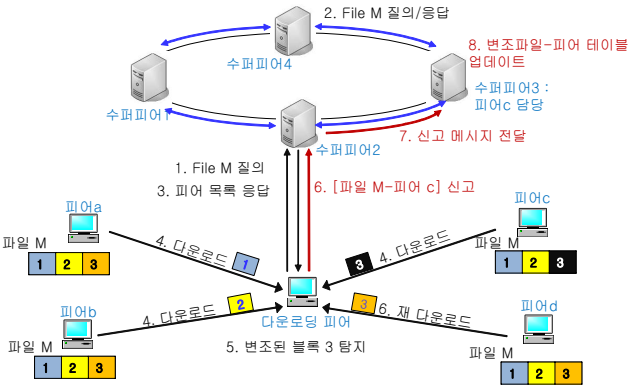
$$AB_j = \frac{D_{*,j}^{+} - D_{*,j}^{-}}{D_{*,j}^{+} + D_{*,j}^{-}} \quad \text{if } D_{*,j} \neq 0, \qquad (1)$$

$$AB_j = 0 \quad \text{otherwise}$$

EigenTrust[6]는 각 피어는 자신이 경험한 피어들의 평판도를 관리하고, 파일 다운로드시 업로딩 피어를 선택할 때 주변 피어들에게 평판도를 질의하여 결정한다. 한편 XRep[7]은 파일의 평판도와 피어의 평판도를 함께 고려하여 업로딩 피어를 결정한다.

3. 제안하는 시스템

본 연구에서 고려하는 P2P 파일 공유 시스템은 2 단계 계층 구조를 가진 오버레이 네트워크를 이룬다. 첫 번째 계층은 디렉토리 서비스를 제공하는 수퍼피어들로 이루어져 있고, 두 번째 계층은 파일을 검색 및 업/다운로드를 하는 일반 피어들로 이루어져 있다. 파일은 여러 개의 파트로 나누어져 동시에 다운로드 되며, 받은 파트의 해쉬값이 다를 경우 다운로딩 피어는 AICH 알고리즘[9]을 이용하여 변조 블록을 탐지하고 해당 블록을 재전송 받는다. 또한 재전송 받는 동시에 수퍼피어에게 변조된 파일을 신고하고, 신고를 받은 수퍼피어는 고의적 변조 파일을 탐지하여 디렉토리 서비스를 중지한다 (그림 1).



(그림 1) 고의적 변조 파일 배제방법의 개략적인 구조

고의적 변조 파일을 배제하는 방법은 총 세 단계로 이루어진다.

【단계 1: 변조된 파일 신고】

다운로딩 피어는 받는 블록들의 변조 상태를 검사하고, 만일 변조된 블록이 있다면 그 블록을 포함한 파일(*corrupt_file*)의 ID($ID_{corrupt\_file}$)와 그 블록을 공유한 피어(*corrupt_peer*)의 ID($ID_{corrupt\_peer}$)를 쌍으로 수퍼피어에게 신고한다. 신고 메시지는 위조될 수 없도록 신고하는 피어(*claim_peer*)의 개인키로 암호화된 전자 서명($SIGN_{claim\_peer}$)이 포함된다. 즉, 신고 메시지는 $[ID_{corrupt\_peer} \| ID_{corrupt\_file} \| SIGN_{claim\_peer}]$ 이다. 신고 메시지를 받은 수퍼피어는 메시지에 표기된 *corrupt_peer* 의 디렉토리 서비스를 담당하는 수퍼피어 ($SUP_{corrupt\_peer}$)에게 이 메시지를 전달한다.

【단계 2: 고의적 변조 파일 탐지】

수퍼피어는 신고 메시지를 받을 때마다, 메시지의 전자서명을 검증한 후 변조 파일-피어 테이블 <표 1> 에 레코드를 갱신한다. ($ID_{corrupt\_peer}$, $ID_{corrupt\_file}$)–속성은 각각 변조된 블록을 제공한 피어의 ID 와 그 블록을 포함한 파일의 ID 를 나타낸다. *Claim#*-속성은 해당 ($ID_{corrupt\_peer}$, $ID_{corrupt\_file}$)-속성값이 몇 번 신고되었는지를 나타낸다. 즉, 신고될 때마다 이 *Claim#*-속성값이 1 씩 증가된다. 이 *Claim#*-속성값이 고의적 변조 파일 임계치 ($TH_{forgedfile}$)보다 같거나 커지면, 해당 ($ID_{corrupt\_peer}$, $ID_{corrupt\_file}$)-속성값은 고의적 변조 파일로 판단한다. 이는 *ForgedFlag*-속성의 속성값 *Yes* 또는 *No* 로 표현된다.

【단계 3: 디렉토리 서비스 중지】

이전 단계에서 고의적 변조 파일로 탐지된 파일은

더 이상 P2P 시스템에 공유되지 않도록 디렉토리 서비스를 중지한다.

<표 1> 고의적 변조 파일-피어 테이블

$ID_{corrupt\_peer}$	$ID_{corrupt\_file}$	*Claim#*	*ForgedFlag*
3242189	8927382	3	No
2234123	4562432	10	Yes

4. 모의 실험을 통한 성능 평가

4.1. 평가 요소

제안하는 시스템을 평가하는 요소는 고의적 변조 파일의 검색률 (*FSR*), 고의적 변조 파일 탐지 방법의 거짓양성 (false positive) 오판율 (*FFPR*), 두 가지이다.

$$FSR = \frac{\sum_{i}^{total\_req} \frac{N(PRF_i)}{N(PR_i)}}{total\_req} \qquad FFPR = \frac{N(HFX)}{N(HF)}$$

FSR 에서 $N(PR_i)$는 i 번째 요청된 파일을 가지고 있다고 검색된 피어의 개수, $N(PRF_i)$는 i 번째 요청된 파일을 가지고 있다고 검색되었지만 실제로는 고의적 변조 파일로 공유하고 있는 피어의 개수, *total_req* 는 파일이 요청된 총 횟수를 의미한다. 즉, *FSR* 은 P2P 시스템 내에서 고의적 변조 파일의 탐지가 얼마나 정확하며 또 얼만큼 충분히 배제되고 있는지를 나타내는 척도이며, 이 값이 0 에 가까울수록 고의적 변조 파일이 유포될 확률은 낮다는 것을 의미한다.

FFPR 에서 *N(HF)*는 고의적 변조 블록을 포함하지 않은 파일, 즉 올바른 파일의 개수이고, *N(HFX)*는 올바른 파일이지만 잘못된 탐지로 인해 P2P 시스템에서 배제된 파일의 개수이다. 즉, *FFPR* 은 올바른 파일이 배제되지 않고 충분히 공유되는지를 나타내는 척도이며, 이 값이 0 에 가깝게 낮다면 대부분의 올바른 파일이 충분히 공유 및 배포 되고 있음을 뜻한다.

4.2. 실험 환경 및 방법

성능 평가를 위해 2 가지 실험을 수행하였고, 이에 사용된 매개 변수와 그 값은 <표 2>와 같다. 악의적인 피어는 고의적 변조 파일을 공유하는 피어이다.

<표 2> 실험 매개 변수와 설정 값

실험 매개 변수	값
일반피어의 총 개수	1000
수퍼피어의 총 개수	10
초기에 공유된 파일 개수	1000
초기에 공유된 파일 인스턴스의 개수	4000
파일의 요청 총 횟수	1000000
악의적인 피어의 비율	10,30,50%
악의적인 피어가 공유하고 있는 파일 중에 고의적 변조 파일의 비율	50% 이상
고의적 변조 파일 임계치 ($TH_{forgedfile}$)	5

【실험 1】 고의적 변조 파일의 검색률(*FSR*)에 대해 <표 3>에 기재된 3 가지 시스템을 비교하였다. 실험 대상 ③은 Mekouar et al. [5]가 제안한 피어 평판 시스템을 이용하였고, 평판도(수식 1)가 0.5 미만이면 디렉토리 서비스를 중단하였다. 악의적인 신고를 하는 피어는 없다고 가정하였다. *FSR* 값은 파일 요청 횟수가 5,000 번일 때마다 측정하였다.

【실험 2】 거짓양성 오판율 (FFPR)에 대해 <표 3>의 ①과 ②를 비교하였다. 악의적인 신고를 하는 피어는 없다고 가정하였다. *FSR* 값은 파일 요청 횟수가 5,000 번일 때마다 측정하였다.

<표 3> 실험 대상

①	eMule 만 사용한 경우 (*eMule*)
②	eMule 에 제안하는 시스템을 함께 사용한 경우 (*eMule+eX-FB*)
③	eMule 에 피어 평판 시스템을 함께 사용한 경우 (*eMule+peerRP*)

4.3. 실험 결과 및 분석

【실험 1】 악의적인 피어의 비율에 상관없이 파일의 요청 횟수가 높아질수록 제안하는 시스템(*eMule+eX-FB*)의 고의적 변조 파일의 검색율(*FSR*)이 다른 시스템에 비해 현저하게 감소한다 (그림 2). 그러나 요청 횟수가 작을 때에는 *eMule+peerRP* 가 좀더 낮은 *FSR* 을 나타낸다. *eMule+peerRP* 에서는 악의적인 피어의 평판도가 낮아지면 그 피어가 제공하는 모든 파일을 배제시키기 때문에, 아직 탐지되지 않은 다른 고의적 변조 파일도 함께 배제되기 때문이다.

실험에서 측정된 *FSR* 값들의 평균값으로 계산했을 때, *eMule+eX-FB* 의 *FSR* 은 *eMule* 보다 약 22 배 감소했고, *eMule+peerRP* 보다 약 4 배 감소하였다. 이 실험 결과는 제안하는 시스템이 타 시스템보다 더 많고 더 정확하게 고의적 변조 파일을 배제시키고 있음을 의미한다.

【실험 2】 제안하는 시스템(*eMule+eX-FB*)은 거짓양성 오판율(*FFPR*)이 항상 0 인 반면, *eMule+peerRP* 는 *FFPR* 값이 0 이상이며, 그 값은 파일의 요청 횟수가 증가함에 따라 감소한다 (그림 3). *eMule+peerRP* 의 *FFPR* 이 0 보다 큰 값을 갖는 원인은 낮은 평판도를 가진 피어가 공유하는 모든 파일은 고의적 변조 파일로 판단되어 올바른 파일도 함께 배제 되기 때문이다. 하지만 요청 횟수 증가에 따라 피어의 평판도가 평준화 되기 때문에 *FFPR* 이 점진적으로 감소하게 된다.

따라서 실험 결과는 올바른 파일 공유에 있어서 제안하는 시스템이 기존의 평판 관리 시스템에 비해 더 나은 성능을 보여주고 있다.

5. 결론 및 향후 계획

본 논문에서는 다중 소스로부터 다운로드가 가능한 P2P 파일 공유 시스템에서 네트워크 자원 낭비와 클라이언트 시스템 사용량이 증가하는 것을 막기 위해, 고의적 변조 파일을 배제시키는 방법을 제안하였다. 제안한 방법은 디렉토리 서비스를 담당하는 수퍼피어가 변조파일-피어 테이블을 관리함으로써, 파일 검색 요청 시 고의적 변조 파일에 대해 디렉토리 서비스를 중단함으로써 고의적 변조 파일의 공유를 배제시켰다.

모의 실험을 통해 고의적 변조 파일이 검색되는 비율이 타 시스템에 비해 최대 22 배까지 감소하는 것을 알 수 있었다. 또한 거짓양성 (false positive) 오판율은 악의적인 신고를 감안하더라도 0.05 를 넘지 않았다. 그러므로 제안하는 방법을 기존 P2P 시스템에 적용할 경우, 고의적 변조 파일의 다운로딩에서 발생하는 네트워크 자원 낭비를 줄일 수 있고 피어의 다운로드 속도도 증가할 것이다.

향후 연구로는 수퍼피어에서 관리하는 테이블로 인한 메모리 소비와 신고 메시지 처리과정 오버헤드를 분석해야 할 것이고, 오래된 신고 기록을 삭제하는데 효과적인 알고리즘이 필요할 것이다. 또한 기존 평판 관리 시스템과 연동하게 되었을 때의 효과에 대한 분석이 필요하다.

참고문헌

[1] eMule, http://www.emule.org/.

[2] BitTorrent, http://www.bittorrent.com/.

[3] Gnutella, http://www.gnutella.com/.

[4] KaZaA, http://www.kazaa.com/.

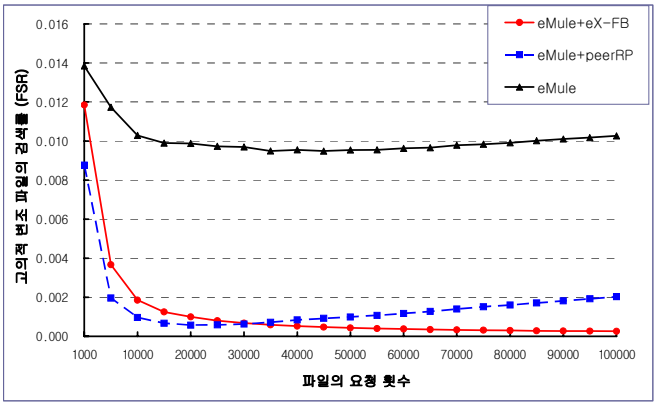
[5] Loubna Mekouar, Youssef Iraqi and Raouf Boutaba, Peer-to-Peer's Most Wanted: Malicious Peers. *Computer Networks, Special Issue on Management in Peer-to-Peer Systems: Trust, Reputation and Security*, Vol. 50, no. 4, pp. 545-562, 2006.

[6] Sepandar D. Kamvar, Mario T. Schlosser, and Hector Carcia-Molina, The EigenTrust Algorithm for Reputation Management in P2P Networks. In *Proc. the 12th international World Wide Web Conference (WWW'03)*, 2003.

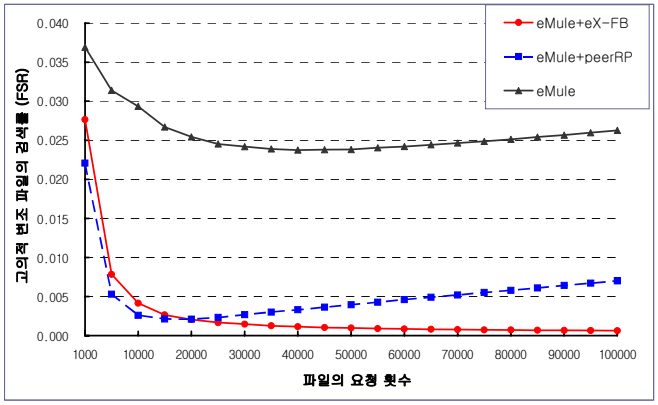
[7] E. Damiani, D. C. di Vimercati, S. Paraboschi, P. Samarati, and F. Violante, Reputation-Based Approach for Choosing Reliable Resources in Peer-to-Peer Networks. In *Proc. the 9th ACM Conference on Computer and Communications Security (CCS'02),* 2002.

[8] Anurag Garg and Roberto Cascella, Reputation Management for Collaborative Content Distribution. In *Proc. the 6th IEEE International Symposium on a World of Wireless Mobile and Multimedia Networks (WoWMoM '05)*, 2005.

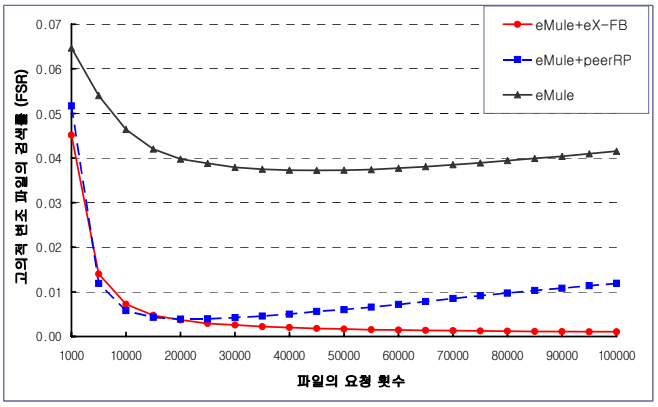
[9] Advanced Intelligent Corruption Handling (AICH), http://www.emule-project.net/



(a) 악의적인 피어의 비율이 10 % 일 때

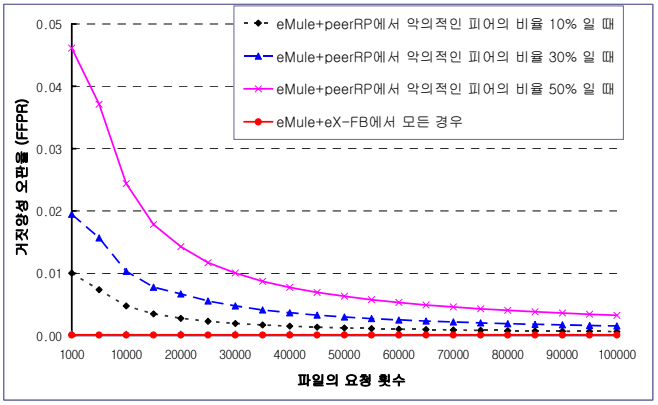


(b) 악의적인 피어의 비율이 30% 일 때



(c) 악의적인 피어의 비율이 50% 일 때

(그림 2) 고의적 변조 파일의 검색율(*FSR*) 비교



(그림 3) eMule 에 피어 평판 시스템을 함께 사용한 경우 (*eMule+peerRP*)의 *FFPR*