

다중 피어 및 공유 자원의 신뢰성 향상을 위한 P2P 네트워크의 평판 관리

신정화[†], 김태훈^{**}, 탁성우^{***}

요 약

P2P 네트워크가 가진 오픈성과 익명성 때문에 P2P 참여자들은 손쉽게 신뢰할 수 없는 자원들을 확산할 수 있다. 신뢰할 수 없는 자원의 확산을 감소시키고 동시에 이러한 자원을 제공하는 악의적인 피어들이 P2P 네트워크에 참여하는 것을 제한하기 위하여 피어들의 과거 행동을 나타내는 평판을 이용할 수 있으며, 이 정보는 피어들의 미래 행동을 예측하는데 도움이 된다. 한편 평판을 참조하는데 있어 몇몇 피어는 특정 피어의 평판을 증가 또는 감소시키기 위하여 고의적으로 거짓 평가(false feedback)를 제공할 수 있다. 이에 본 논문에서는 거짓 평가를 주는 피어들을 식별함으로써 P2P 네트워크의 신뢰성과 효율성을 향상시킬 수 있는 새로운 평판 관리 기법, TrustRRep(Trustable Resource sharing service using Reputation)를 제안하고자 한다. 제안 기법은 악의적인 피어들에 의한 자원 공유를 식별함으로써 공유 자원에 대한 신뢰성 또한 제공할 수 있다. 본 논문에서는 NS-2 시뮬레이터를 이용하여 제안 기법과 기존 평판 관리 기법의 효율성을 비교하고 분석하였다. 제안 기법은 신뢰할 수 없는 자원에 대한 다운로드 횟수를 감소시켜 신뢰할 수 없는 자원의 확산을 막고, 거짓 평가를 주는 피어의 식별 및 피어들의 평판에 신뢰성을 제공한다. 또한 악의적인 피어들의 신뢰 값을 감소시킴으로써 P2P 네트워크의 참여에 제약을 가한다.

A Reputation Management Scheme Improving the Trustworthiness of Multi-peers and Shared Resources in P2P Networks

Jung Hwa Shin[†], Tae Hoon Kim^{**}, Sung Woo Tak^{***}

ABSTRACT

Inauthentic resources can be easily spread by P2P (Peer-to-Peer) participants due to the openness and anonymity of P2P networks. A possible way to restrict the distribution of inauthentic resources and prevent malicious peers from joining P2P networks is to exploit peers' reputation which reflect their past behaviors and are also helpful to predict peers' future behaviors. There is a possibility that some peers intentionally plays along with other peers in order to increase/decrease its reputation through false feedback exchanges. Therefore, we propose a new reputation management scheme, called TrustRRep (Trustable Resource sharing service using Reputation) scheme, which improves the trustworthiness and efficiency of P2P networks by identifying peers who give false feedback. The TrustRRep scheme is also capable of providing peers with the trustworthiness of shared resources by discriminating resources distributed by malicious peers. We implement the proposed TrustRRep scheme on the NS-2 simulator for evaluating its performance compared to the recent reputation management work available in literature. A case study on simulations shows that the proposed reputation management scheme yields efficient performance in terms of the minimal download ratio and dissemination of inauthentic resources, the efficient identification of peers who give false feedback, and the provisioning of the trustworthiness of peers' reputation. It also shows that the proposed TrustRRep scheme imposes the restrictions of participating P2P networks on a malicious peers by diminishing its trust value.

Key words: P2P(Peer-to-Peer), Reputation(평판), False Feedback(거짓 평가)

* 교신저자(Corresponding Author): 탁성우, 주소: 부산시 금정구 장전동 산30번지(609-735), 전화: 051)510-2387, FAX: 051)515-2208, E-mail: swtak@pusan.ac.kr
접수일: 2008년 4월 16일, 완료일: 2008년 7월 4일

[†] 준회원, 부산대학교 U-Port정보기술산학공동사업단 전임연구원

(E-mail: shinjh@pknu.ac.kr)

^{**} 준회원, 부산대학교 컴퓨터공학과 박사과정
(E-mail: ninkth@hotmail.com)

^{***} 부산대학교 정보컴퓨터공학부 부교수

* 이 논문은 2008년 정부(국토해양부)의 재원으로 지능형 국토정보기술혁신사업단의 지원을 받아 수행된 연구임

1. 서 론

피어 간에 다양한 자원의 공유가 가능한 P2P 네트워크 환경에서는 피어간의 트랜잭션 생성 및 자원 공유에 대하여 특별한 제약이 없다. 주로 익명의 피어 간에 발생하는 트랜잭션에서는 악의적인 목적을 가진 피어에 의해 신뢰할 수 없는 자원이나 질이 낮은 자원이 손쉽게 확산될 수 있다[1]. 따라서 악의적인 피어의 참여 제한 및 신뢰할 수 없는 자원의 확산을 감소시키기 위해서는 공유 자원에 대한 평판과 개별 피어의 이전 행동을 나타내는 평판을 사용할 수 있다. 트랜잭션을 수행한 피어들이 제공한 평가(feedback) 값을 이용하여 계산된 평판은 자원과 피어의 신뢰성을 제공할 수 있지만, 피어들이 제공한 평가 값의 정직성을 고려하지 않고 계산된다.

그러므로 악의적인 피어는 특정 피어의 평판을 조작하기 위해 고의적으로 거짓 평가(false feedback)를 제공할 수 있으며, 또한 피어들의 거짓 평가는 공유 자원 및 피어의 평판에 대한 신뢰성을 감소시킨다[2,3]. 평판 시스템의 효율성은 평가의 질(quality)에 의존하며, 평판 시스템에서 중요한 요소는 거짓 평가를 식별하는 것이다. 현재 P2P 어플리케이션을 위한 대부분의 평판 시스템들은 서비스와 평가(feedback) 사이에 상관 관계를 가정한다. 좋은 서비스를 제공하는 피어들은 신뢰할 수 있는 평가를 제공하고 질이 낮은 서비스를 제공하는 피어는 거짓 평가를 제공하는 경향이 있다.

이에 본 논문에서는 공유 자원과 피어의 평판에 대한 신뢰성 및 거짓 평가를 주는 피어를 식별하여 P2P 네트워크의 신뢰성 및 효율성을 향상시키기 위한 새로운 평판 관리 기법인 TrustRRep(Trustable Resource sharing service using Reputation)를 제안하였다. 본 논문에서는 NS-2 시뮬레이터[4]를 이용하여 TrustRRep 기법과 기존 평판 관리 기법의 효율성을 비교하고 분석하였다. TrustRRep 기법은 신뢰할 수 없는 자원에 대한 다운로드 횟수를 감소시켜 신뢰할 수 없는 자원의 확산을 막고, 거짓 평가를 주는 피어의 식별 및 피어의 평가 값에 신뢰성을 제공한다. 또한 악의적인 피어의 신뢰 값을 감소시킴으로써 악의적인 피어의 트랜잭션 참여에 제약을 가할 수 있다.

본 논문의 구성은 다음과 같다. 2장에서는 P2P 네트워크에서 평판을 관리하는 기법에 대한 기존 연구

를 분석하였으며, 3장에서는 공유 자원과 피어의 신뢰성을 제공하고 liar를 식별 할 수 있는 TrustRRep 기법을 설명하였다. 4장에서는 TrustRRep 기법의 성능을 분석하였으며, 마지막으로 5장에서는 결론을 기술하였다.

2. 관련 연구

P2P 네트워크 환경에서는 공유 자원 및 개별 피어의 신뢰성을 제공하기 위하여 평판이 주로 사용되고 있으며 평판의 관리를 위해 많은 기법들이 제안되어져 왔다. 가장 잘 알려진 평판 관리 기법은 온라인 경매 사이트인 eBay에서 사용하는 방법이다[5]. eBay에서는 트랜잭션이 완료된 후에 피어들이 서로를 평가하고, 이러한 평가 값의 합으로 피어의 평판을 계산한다. 계산된 평판은 피어가 향후 트랜잭션 대상을 선택할 때 사용한다. 현재 대부분의 P2P 네트워크 환경에서는 파일 공유 서비스가 가장 많이 사용되며, 이러한 파일 공유 서비스에서 신뢰할 수 없는 자원의 다운로드를 감소시킬 수 있는 평판 관리 기법이 많이 연구되고 있다. 이와 관련된 기존의 연구를 살펴보면 다음과 같다.

EigenTrust 기법은 신뢰할 수 없는 자원을 공유하는 악의적인 피어를 감소시키기 위한 기법으로 트랜잭션 대상을 선택할 때 피어의 평판을 참조하여 악의적인 피어의 참여에 제약을 가한다[6]. 반면 EigenTrust 기법은 트랜잭션 결과에 거짓 평가를 주는 피어를 식별할 수 있는 기능은 제공하지 않는다. 이와 달리 PeerTrust 기법은 피어 간에 교환되는 평가 값의 유사도를 계산하여 상대적인 피어의 정직성을 계산한다[7]. 그리고 PeerTrust 기법은 상대 피어에 평가를 주는 피어에게 인센티브를 줌으로써 피어가 평가에 활발하게 참여하도록 하지만, 평가 값 자체에 대한 신뢰성과 상관없이 인센티브를 주는 문제점이 있다. EigenTrust 기법 및 PeerTrust 기법은 공유 자원과 개별 피어에 대한 평판을 구분하지 않고 공유 자원의 평판을 개별 피어의 평판 계산에 활용한다.

반면 FileTrust 기법은 공유 자원과 개별 피어에 대한 평판을 따로 구분하여 개별 피어뿐만 아니라 공유 자원에 대한 평판을 관리한다[8]. 그리고 FileTrust 기법은 트랜잭션 대상을 선택할 때 가장 높은 평판을 가지는 자원을 선택하고, 자원을 제공하

는 피어 중 가장 낮은 기여도를 가지는 피어를 선택하여 특정 자원을 가진 피어에게 트래픽이 집중되는 현상을 차단한다. Iguchi 기법은 공유 자원과 개별 피어에 대한 평판을 따로 구분하여 관리하며, 가장 높은 평판을 가지는 자원을 트랜잭션 대상으로 선택한 후에 자원을 제공하는 피어 중 가장 높은 기여 점수를 가지는 피어를 선택하여 트랜잭션을 수행한다[9]. 그러므로 FileTrust 기법과 달리 특정 피어에 트래픽이 집중되는 경우가 발생할 수 있다. 한편, FileTrust 기법과 Iguchi 기법은 피어의 거짓 평가를 식별하기 위해 동일한 자원을 다운로드 한 피어들이 제공한 평가 값을 참조하여 피어의 정직성을 계산하지만 동일한 자원에 대한 평가 값이 없을 경우 정직성의 계산이 어렵다.

지금까지 살펴본 바와 같이 기존의 연구는 신뢰할 수 없는 자원을 공유하는 피어를 식별하는 것을 목적으로 하며, 평가 값 자체에 대한 정직성과 거짓 평가를 주는 피어의 식별을 고려하지 않거나 식별에 필요한 값의 계산이 어려운 단점을 가지고 있다[10]. 분석된 기존의 관련 연구를 기반으로 본 논문에서 접근하고자 하는 방향은 다음과 같다. 다중 피어 및 공유 자원의 신뢰성을 제공할 뿐만 아니라 거짓 평가를 주는 피어를 식별하고 악의적인 피어의 참여를 제한함으로써 P2P 네트워크의 신뢰성 및 효율성을 향상시킬 수 있는 TrustRRep 기법을 제안하고자 한다.

TrustRRep 기법은 비구조화된 P2P 네트워크에서 다중으로 분산되어 있는 신뢰 피어들을 이용하여 공유 자원과 개별 피어의 평판을 따로 구분하여 관리한다. 그리고 개별 피어의 평가 값에 대한 정직성을 확인하여 거짓 평가 여부를 식별하기 위한 정직성 비율과 피어의 다운로드 요청을 허가하기 위해 참조하는 피어의 신뢰 값을 계산하기 위한 수식을 제시한다. 다음 장에서는 TrustRRep 기법에 대하여 상세히 기술하였다.

3. 다중 피어 및 공유 자원의 신뢰성 향상을 위한 P2P 네트워크의 평판 관리

본 장에서는 다중 피어 및 공유 자원의 신뢰성을 제공하고 거짓 평가를 주는 피어를 식별함으로써 P2P 네트워크의 신뢰성과 효율성을 향상시킬 수 있는 새로운 TrustRRep 기법에 대해 설명한다.

3.1 TrustRRep 기법의 특징

기존의 평판 관리 기법은 대부분 공유 자원과 피어에 대하여 신뢰성을 제공하는 것을 목적으로 하며, 거짓 평가를 주는 피어에 대한 식별은 고려하지 않는다. 따라서 자원을 요청한 피어는 거짓 평가를 주는 피어에 의해서 올바르게 실행되지 않거나 요청한 자원과 다른 자원을 다운로드 받을 수 있다. 자원을 요청하고자 하는 피어는 해당 자원의 다운로드를 수행한 피어들에 의해 제공된 평가 값의 합으로 계산된 평판을 참조하여 업로드 대상의 자원을 선택하기 때문에 피어가 제공하는 평가 값의 신뢰성은 매우 중요하다. 그러나 일부 피어는 신뢰할 수 있는 피어의 자원을 다운로드 하더라도 대상 피어의 평판을 감소시키기 위해 다운로드 받은 자원에 대하여 거짓 평가를 주거나, 악의적인 피어의 자원을 다운로드 하더라도 악의적인 피어의 평판을 증가시키기 위하여 악의적인 피어가 제공하는 자원에 대해 좋은 평가를 줄 수도 있다[11].

이와 같이 거짓 평가를 주는 피어를 일반적으로 'liar'라 하고, 피어의 거짓 평가는 악의적인 피어의 평판을 증가시키고 정직한 피어의 평판을 감소시켜 다른 피어의 평판에 부정하게 영향을 미칠 수 있다. 그러므로 평판 관리 기법의 설계에서는 공유 자원과 피어의 신뢰성을 제공하고 거짓 평가를 주는 피어의 식별을 고려해야 한다. 공유 자원 및 다중 피어의 신뢰성을 제공함과 동시에 liar의 식별 및 liar의 서비스 참여와 이용에 제약을 가하며 자원과 피어의 평판을 구분하여 관리하는 본 논문에서 제안한 TrustRRep 기법의 특징을 살펴보면 다음과 같다.

자원 평판은 피어가 공유하는 개별 자원에 대한 신뢰성을 나타내고, 피어 평판은 피어가 신뢰할 수 있는 자원을 공유하고 있는가를 나타낸다. 피어가 공유하는 전체 자원의 평판 합을 참조하여 피어 평판을 계산한다. 그리고 피어의 정직한 평가 유무를 나타내는 정직성 비율 및 다운로드 요청을 허가하기 위한 피어의 신뢰 값도 계산한다. 또 이러한 정보들을 관리하는 다수의 신뢰 피어가 존재하며, 신뢰 피어는 피어의 공유 자원 목록과 자원 및 피어의 평판, 그리고 피어의 신뢰 값과 정직성 비율을 관리한다. 피어들은 다운로드 대상 자원을 선택하고, 자원의 다운로드 요청을 수락하기 위해 필요한 정보를 신뢰 피어에게 요청하고, 해당 트랜잭션이 완료된 후에 다운로드

받은 자원의 평가 값을 신뢰 피어로 전송한다.

제안 기법은 부트스트래핑을 위해 초기에 잘 알려진 신뢰 피어가 있다고 가정한다. 그리고 트랜잭션의 증가로 초기에 알려진 신뢰 피어에게 트래픽이 집중할 경우 트랜잭션에 참여중인 피어 중 높은 신뢰 값을 가지는 피어를 신뢰 피어로 추가로 지정하여 트랜잭션 참여 피어들에게 해당 피어에 관한 정보를 알려준다. 제안 기법의 동작 절차는 다음과 같이 4개의 절차로 구성 된다: (절차 I) 참여, (절차 II) 검색 질의와 응답, (절차 III) 선택과 다운로드, (절차 IV) 평가와 업데이트. 다음 절에서는 각 절차별 세부 동작을 기술하였다.

3.2 TrustRRep 기법의 동작

표 1은 TrustRRep 기법의 세부적인 동작을 설명하기 위해 사용하는 용어들을 나타낸다.

표 1. 용어

기 호	설 명
P_i, P_j	피어 아이디, $i, j = \{1, 2, \dots, n\}$
TP_i	신뢰 피어 아이디, $i = \{1, 2, \dots, n\}$
$Neighbors(P_i)$	피어 P_i 의 이웃 피어의 집합, $i = \{1, 2, \dots, n\}$
P_A	자원 요청 피어
P_B	자원 제공 피어
$r^k_{P_i}$	피어 P_i 의 공유 자원 r 의 아이디, $i = \{1, 2, \dots, n\}, k = \{1, 2, \dots, m\}$
$RR(r^k_{P_i})$	피어 P_i 의 자원 r^k 에 대한 평판, $i = \{1, 2, \dots, n\}, k = \{1, 2, \dots, m\}$
$PR(P_i)$	피어 P_i 의 평판, $i = \{1, 2, \dots, n\}$
$\delta(P_i)$	피어 P_i 의 정직성 비율, $i = \{1, 2, \dots, n\}$
$T(P_i)$	피어 P_i 의 신뢰 값, $i = \{1, 2, \dots, n\}$
KEY	검색 키워드
TTL_{init}	초기 TTL 값
$FV_{P_i, P_j}(r^k_{P_i})$	피어 P_j 의 자원 r^k 에 대하여 P_i 가 제공한 평가, $i, j = \{1, 2, \dots, n\}, k = \{1, 2, \dots, m\}$
$\alpha(r^k_{P_i})$	피어 P_i 의 자원 r^k 에 대한 좋은 평가 값의 비율, $i = \{1, 2, \dots, n\}, k = \{1, 2, \dots, m\}$
$\beta(r^k_{P_i})$	피어 P_i 의 자원 r^k 에 대한 나쁜 평가 값의 비율, $i = \{1, 2, \dots, n\}, k = \{1, 2, \dots, m\}$
$CFN(P_i)$	P_i 가 제공한 올바른 평가 횟수 $i = \{1, 2, \dots, n\}$
$TFN(P_i)$	피어 P_i 가 제출한 전체 평가 횟수
$TDN(r^k_{P_i})$	피어 P_i 의 자원 r^k 에 대한 전체 다운로드 횟수, $i = \{1, 2, \dots, n\}, k = \{1, 2, \dots, m\}$
$TN(P_i)$	피어 P_i 가 수행한 전체 트랜잭션 횟수, $i = \{1, 2, \dots, n\}$

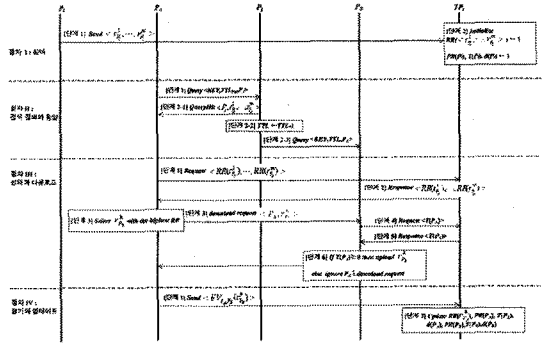


그림 1. TrustRRep 기법의 전체 동작

그림 1은 TrustRRep 기법의 전체 동작을 보여주며, 각 절차에 대한 세부 설명은 다음과 같다.

절차 I. 참여 (Join)

(단계 1) Each peer P_i : Join

for all $i = 1, \dots, n$ do

send $\langle r^1_{P_i}, \dots, r^m_{P_i} \rangle, \dots, \langle r^1_{P_n}, \dots, r^m_{P_n} \rangle$ to TP_i

end-loop

(단계 2) all $TP_i (i = 1, \dots, n)$: Initialize

$RR(\langle r^1_{P_1}, \dots, r^m_{P_1} \rangle), \dots, RR(\langle r^1_{P_n}, \dots, r^m_{P_n} \rangle) \leftarrow 1,$

$PR(P_1), \dots, PR(P_n) \leftarrow 1, T(P_1), \dots, T(P_n) \leftarrow 1,$

$\delta(P_1), \dots, \delta(P_n) \leftarrow 1$

절차 I은 피어들이 P2P 네트워크에 참여하고 신뢰 피어가 초기화를 수행하는 과정을 보여준다. 각 동작 과정을 살펴보면 다음과 같다. (단계 1)에서 각각의 피어(P_1, \dots, P_n)들은 파일 공유 서비스를 이용하기 위해 P2P 네트워크에 참여하고, 자신이 공유하고자 하는 자원 목록 $\langle r^1_{P_1}, \dots, r^m_{P_1} \rangle, \dots, \langle r^1_{P_n}, \dots, r^m_{P_n} \rangle$ 을 신뢰 피어 TP_i 에게 전송한다. 신뢰 피어 TP_i 는 각각의 피어로부터 받은 공유 자원 목록의 자원 평판(RR)과 피어 P_1 부터 피어 P_n 까지의 피어 평판(PR) 및 피어의 신뢰 값(T) 그리고 정직성 비율(δ)의 값을 1로 초기화 한다(단계 2).

절차 II. 검색 질의와 응답 (Query and QueryHit)

(단계 1) P_A : Query

for all $Neighbors(P_A)$ do

send Query $\langle KEY, TTL_{init}, P_A \rangle$ to

$Neighbors(P_A)$

end-loop

```

(단계 2)  $P_j \in \text{Neighbors}(P_A) : \text{QueryHit}$ 
  if find(KEY) != NULL then
    for all  $j=1, \dots, n$  do
      send QueryHit  $\langle P_j, r_{P_j}^1, \dots, r_{P_j}^m \rangle$  to  $P_A$ 
      (단계 2-1)
    end-loop
  else
    TTL  $\leftarrow$  TTL-1
    (단계 2-2)
    if TTL > 0 then
      for all Neighbors( $P_j$ ) do
        send Query  $\langle \text{KEY}, \text{TTL}, P_A \rangle$  to
        Neighbors( $P_j$ )
        (단계 2-3)
      end-loop
    end-if
  end-if
(단계 3)  $P_A : \text{Receive}$ 
  for all  $j=1, \dots, n$  do
    receive QueryHit  $\langle P_j, r_{P_j}^1, \dots, r_{P_j}^m \rangle$ 
  end-loop
  
```

절차 II는 피어들이 원하는 자원을 다운로드 하기 위해 자원에 관한 검색을 요청하고 여러 피어들로부터 응답을 받는 과정을 보여준다. 각 동작 과정을 살펴보면 다음과 같다. (단계 1)은 각각의 피어가 필요로 하는 자원을 찾기 위해 검색 질의를 전송하는 단계이다. 자원 요청 피어인 피어 P_A 는 자신의 아이디(P_A)와 검색 질의가 전달될 이웃 피어의 수(TTL_{init}), 그리고 찾고자 하는 자원과 관련된 검색어(KEY)를 포함한 질의 메시지 (Query $\langle \text{KEY}, TTL_{init}, P_A \rangle$)를 자신의 이웃 피어(Neighbors(P_A))들에게 전송한다. TTL_{init} 은 질의 메시지가 전달될 이웃 피어의 수를 나타내는 값으로 이웃 피어를 통과할 때마다 1씩 감소하며 0이 되면 더 이상 메시지를 전달하지 않으므로써 메시지가 네트워크 내에서 계속적으로 전달되는 것을 제한한다.

(단계 2)는 각각의 피어로부터 질의를 받은 이웃 피어들의 동작을 설명한다. 피어 P_A 로부터 질의 메시지를 받은 이웃 피어($P_j, j=1, \dots, n$)들은 검색어에 일치하는 자원을 가지고 있다면, 피어 P_A 에게 자신의 아이디($P_j, j=1, \dots, n$)와 해당 자원의 아이디($r_{P_j}^1, \dots, r_{P_j}^m$)를 함께 전송한다(단계 2-1). 피어 P_A 의 이웃 피어들이 검색어에 일치하는 자원을 가지고 있지 않다면,

TTL 값을 1만큼 감소시키고 절차 II의 (단계 1)에서 받은 질의 메시지를 자신의 이웃 피어(Neighbors(P_j))들에게 전송한다(단계 2-2와 2-3). 피어 P_A 는 검색어에 일치하는 자원을 가진 피어들로부터 응답 메시지(QueryHit $\langle P_j, r_{P_j}^1, \dots, r_{P_j}^m \rangle$)를 받는다(단계 3).

절차 III. 선택과 다운로드 (Select and Download)

```

(단계1)  $P_A : \text{Request}$ 
  for all  $i=1, \dots, n$  do
    send request  $\langle RR(r_{P_i}^1), \dots, RR(r_{P_i}^m) \rangle$  to  $TP_i$ 
  end-loop
(단계 2) all  $TP_i \in TP$ 
  send response  $\langle RR(r_{P_i}^1), \dots, RR(r_{P_i}^m) \rangle$  to  $P_A$ 
(단계3)  $P_A : \text{select}$ 
  select  $r_{P_B}^k$  with the highest RR among
   $\{RR(r_{P_i}^1), \dots, RR(r_{P_i}^m)\}$ 
  send download request  $\langle P_A, r_{P_B}^k \rangle$  to  $P_B$ 
(단계 4)  $P_B : \text{Request}$ 
  for all  $i=1, \dots, n$  do
    send request  $\langle T(P_A) \rangle$  to  $TP_i$ 
  end-loop
(단계5) all  $TP_i \in TP$ 
  send response  $T(P_A)$  to  $P_B$ 
(단계6)  $P_B : \text{Receive}$ 
  receive response  $T(P_A)$ 
  if  $T(P_A) \geq 0$  then
    upload  $r_{P_B}^k$  to  $P_A$ 
  else
    ignore  $P_A$ 's download request
  end-if
  
```

절차 III은 피어들이 다운로드 대상 자원과 피어를 선택하고 자원의 다운로드를 요청하는 과정을 보여준다. 각 동작 과정을 살펴보면 다음과 같다. (단계 1)에서 피어 P_A 는 절차 II의 (단계 3)에서 받은 메시지(QueryHit $\langle P_j, r_{P_j}^1, \dots, r_{P_j}^m \rangle$)를 참조하여 다운로드 대상 자원과 피어의 선택을 위해 신뢰 피어 TP_i 에게 자원의 평판 정보($\langle RR(r_{P_i}^1), \dots, RR(r_{P_i}^m) \rangle$)를 요청한다. 신뢰 피어 TP_i 는 피어 P_A 에게 자원의 평판 정보를 전송한다(단계 2).

(단계 3)은 피어가 다운로드 대상 자원이나 피어를 선택하고 다운로드를 요청하는 단계로 피어 P_A 는

신뢰 피어 TP_i 로부터 받은 자원의 평판 정보 ($\{RR(r_{P_j}^1), \dots, RR(r_{P_j}^m)\}$)를 확인하여 가장 높은 평판 정보를 가지는 자원($r_{P_B}^k$)을 선택하고, 해당 자원을 가진 피어 P_B 에게 자원($r_{P_B}^k$)의 다운로드를 요청한다. 자원의 다운로드 요청을 받은 피어 P_B 는 요청에 응답하기 전 P_A 의 신뢰 값($T(P_A)$)을 확인하기 위해 신뢰 피어 TP_i 에게 요청한다(단계 4). 신뢰 피어 TP_i 는 피어 P_B 에게 피어 P_A 의 신뢰 값을 전송한다(단계 5). 피어 P_B 는 피어 P_A 의 신뢰 값이 0 이상이면 피어 P_A 가 요청한 자원($r_{P_B}^k$)을 P_A 로 전송하고, 그렇지 않은 경우 요청에 응답하지 않는다(단계 6).

절차 IV. 평가와 업데이트 (Evaluate and Update)

(단계 1) P_A : Send

for all $i=1, \dots, n$ do

send feedback $FV_{P_A, P_B}(r_{P_B}^k)$ to TP_i

(feedback value : 1 or -1)

end-loop

(단계 2) all $TP_i \in TP$: Update

receive $FV_{P_A, P_B}(r_{P_B}^k)$

update δ, RR, PR, T

$$\delta(P_A) = \frac{CFN(P_A)}{TFN(P_A)}, \quad 0 \leq \delta \leq 1 \quad (1)$$

$$RR(r_{P_B}^k) = \frac{1}{TDN(r_{P_B}^k)} \sum_{k=1}^{TDN(r_{P_B}^k)} FV_{P_A, P_B}(r_{P_B}^k) \quad (2)$$

$$PR(P_A) = \frac{1}{TN(P_A)} \sum_{k=1}^{TN(P_A)} RR(r_{P_A}^k) \quad (3)$$

$$T(P_A) = PR(P_A) \times \delta(P_A) \quad (4)$$

절차 IV는 자원 요청 피어가 다운로드 받은 자원의 평가 값을 전송하고 신뢰 피어가 자원 및 피어의 평판과 신뢰 값, 정직성 비율을 업데이트하는 과정을 보여준다. 각 동작 과정을 살펴보면 다음과 같다. 자원을 다운로드 받은 피어 P_A 는 신뢰 피어 TP_i 에게 해당 자원의 평가 값($FV_{P_A, P_B}(r_{P_B}^k)$)을 전송한다(단계 1). 자원이 올바르게 실행되고 자신이 요청한 자원이 맞을 경우 1을 평가 값으로 전송하고, 그렇지 않은 경우 -1을 평가 값으로 전송한다. 신뢰 피어 TP_i 는 피어 P_A 의 평가가 정직한지 확인하여 식 (1)을 사용하여 정직성 비율($\delta(P_A)$)을 계산하고 liar 여부를 판단한다. 제안 기법에서 평가 값을 전송한 피어 P_A 의 liar 여부를 판단하는 세부 과정은 다음과 같다.

(단계 1)에서 피어 P_A 가 신뢰 피어 TP_i 에게 다운로드 받은 자원($r_{P_B}^k$)의 평가 값을 전송하면 신뢰 피어 TP_i 는 해당 자원이 최근 n ($n \geq 2$)번의 업로드에서 받은 전체 평가에서 좋은 평가 값의 비율($\alpha(r_{P_B}^k)$)과 나쁜 평가 값의 비율($\beta(r_{P_B}^k)$)을 계산한다. 그리고 피어 P_A 가 전송한 평가 값의 정직성을 확인하기 위하여 식 (5)와 같이 0과 1사이의 값을 가지는 확률 변수 X 를 허용 기준(acceptance criteria)으로 사용하였다. 이때 확률 변수 X 는 균일 분포(uniform distribution)를 따른다.

$$\alpha(r_{P_B}^k) + \beta(r_{P_B}^k) = 1, \quad X \sim [0, 1] \quad (5)$$

피어 P_A 가 자원($r_{P_B}^k$)에 대하여 좋은 평가 값(1)을 준 경우 식 (5)의 확률 변수 X 를 참조하여 식 (6)을 사용하여 평가의 정직성을 판단한다. 식 (7)은 피어 P_A 가 자원($r_{P_B}^k$)에 대하여 나쁜 평가 값(-1)을 준 경우 평가의 정직성을 판단하는 식으로, 식 (5)의 확률 변수 X 를 참조하여 판단한다. 식 (6)의 $FV(1)$ 과 식 (7)의 $FV(-1)$ 은 피어 P_A 로부터 받은 평가 값을 나타낸다.

$$FV(1) = \begin{cases} TRUE, & \text{if } 0 \leq X \leq \alpha(r_{P_B}^k) \\ FALSE, & \text{if } \alpha(r_{P_B}^k) < X \leq \alpha(r_{P_B}^k) + \beta(r_{P_B}^k) \end{cases} \quad (6)$$

$$FV(-1) = \begin{cases} FALSE, & \text{if } 0 \leq X \leq \alpha(r_{P_B}^k) \\ TRUE, & \text{if } \alpha(r_{P_B}^k) < X \leq \alpha(r_{P_B}^k) + \beta(r_{P_B}^k) \end{cases} \quad (7)$$

자원을 제공한 피어 P_B 의 좋은 평가 비율($\alpha(r_{P_B}^k)$)이 높아지면 평가 값을 전송한 피어 P_A 의 평가 값도 정직한 평가로 판단될 가능성이 높아지고, 반면 자원을 제공한 피어 P_B 의 나쁜 평가 비율($\beta(r_{P_B}^k)$)이 높아지면 피어 P_A 가 전송한 평가 값도 거짓 평가로 판단될 가능성이 높아진다.

예를 들어, 피어 P_B 가 최근 10번의 자원 업로드에서 받은 좋은 평가 값의 비율이 0.6, 나쁜 평가 값의 비율이 0.4이고, 피어 P_A 가 신뢰 피어 TP_i 로 평가 값을 전송할 때 발생된 확률 변수 X 의 값이 0.2라고 가정하자. 피어 P_A 가 좋은 평가 값을 준 경우 식 (6)을 참조하며, 확률 변수 X 의 값이 $TRUE$ 의 범위에 해당하므로 정직한 평가라고 판단한다. 만약 확률 변수 X 의 값이 0.8이라면 $FALSE$ 의 범위에 해당하므로 거짓 평가라고 판단한다. 이와 같이 피어 P_B 의 자원에 대하여 다수의 피어들의 평가 값을 참조하여 피어 P_A 의 평가에 대한 정직성을 확인한다. 이를 바

따라서 피어 P_A 가 제공한 정직한 평가 횟수 ($CFN(P_A)$)와 전체 평가 횟수($TFN(P_A)$)를 이용하여 식 (1)과 같이 피어 P_A 의 정직성 비율(δ)을 계산하고 liar 여부를 판단한다. 본 논문의 실험에서는 식 (1)에 의해 계산된 정직성 비율의 값이 0.5 미만이면 liar로 판단한다.

신뢰 피어 TP_i 는 피어 P_A 의 liar 여부를 판단한 다음, 그 결과를 바탕으로 식 (2)와 식(3), 식 (4)를 사용하여 자원 평판(RR)과 피어 평판(PR) 및 신뢰 값(T)에 대한 업데이트를 수행한다(단계 2). 식 (2)는 다운로드 대상 자원($r_{P_B}^k$)의 평판을 업데이트하는 식으로 자원의 전체 다운로드 횟수($TDN(r_{P_B}^k)$)와 평가 값($FV_{P_A, P_B}(r_{P_B}^k)$)의 합을 이용하여 계산한다. 이때, 식 (1)의 계산 결과에 의해 liar로 판단된 피어로부터 받은 평가 값은 자원($r_{P_B}^k$)의 평판 계산에 반영하지 않으므로써 자원 평판의 신뢰성을 향상시킨다.

식 (3)은 피어 평판(PR)을 계산하는 식으로 피어 P_A 가 수행한 전체 트랜잭션 횟수($TN(P_A)$)와 피어 P_A 의 공유 자원에 대한 평판의 합($RR(r_{P_A}^k)$)을 참조하여 계산한다. 식 (4)는 피어의 신뢰 값을 계산하는 식으로 식 (1)에 의해 계산된 피어의 정직성 비율과 식 (3)을 이용하여 계산된 피어 평판을 참조하여 계산한다. 그러므로 식 (1)의 정직성 비율 계산 결과에 의해 liar로 판단된 피어는 신뢰할 수 있는 자원을 제공하더라도 거짓 평가로 인해 낮은 정직성 비율을 가지기 때문에 피어의 신뢰 값이 감소할 수 있다. 또 식 (3)에 의해 계산된 피어의 평판 역시 신뢰 값 계산에 영향을 미치기 때문에 피어의 평판이 감소하는 경우 피어의 신뢰 값까지 감소시킨다.

4. 실험 및 분석

본 논문에서는 제안 기법인 TrustRRep의 효율성을 검증하기 위하여 다음과 같은 실험 환경 하에서 기존에 제안된 평판 관리 기법(EigenTrust, PeerTrust, FileTrust, Iguchi)들과 제안 기법인 TrustRRep에 대하여 4가지 실험 환경(Case I, Case II, Case II, Case IV)을 이용하여 3가지 실험을 수행하였다. 실험을 위한 오버레이 네트워크로 비구조적 P2P 네트워크인 그누텔라[12] 환경을 확장하였으며, 실험 도구로 오버레이 네트워크에서 피어들의 올바른 동작을 실험하기에 적합한 NS-2 시뮬레이터[4]를 사용

표 2. 매개변수

	설 명	기본 값
네트워크	네트워크에 참여하는 전체 피어 수	100
	초기에 배포된 자원 수	400
	각 피어가 가지고 있는 자원 수	$2 \leq n \leq 8$
	메시지의 TTL	5
	피어별 최대 in-degree 수	4
	피어별 최대 out-degree 수	4
피어 동작	신뢰할 수 없는 자원을 배포하는 피어 비율	20%
	거짓 평가를 주는 피어 비율	20%
	악의적인 피어가 주기적으로 배포하는 새로운 자원의 수	2
	피어가 질의를 받았을 때 응답하는 비율	100%
시뮬레이션	전체 트랜잭션 횟수	5,000
	평균 실험 횟수	10

하였다. 본 논문에서는 실험을 위해 표 2와 같은 매개변수를 사용한다.

실험에서 공유 자원과 피어의 평판, 피어의 신뢰 값과 정직성 비율의 초기 값은 1로 설정하고, 공유 자원에 대한 평가 값은 1 또는 -1을 사용한다. 표 3은 악의적인 피어의 행동과 거짓 평가를 주는 liar의 존재 여부, 피어의 재참여 허용 여부를 구분한 4가지 실험을 환경을 보여준다.

첫 번째 실험에서는 트랜잭션의 증가에 따라 신뢰할 수 없는 자원의 다운로드 비율의 변화를 확인하였다. 그림 2는 Case I의 실험 결과로 악의적인 피어는 신뢰할 수 없는 자원을 제공하고, liar는 없으며, 피어의 재참여를 허용하지 않는 환경을 가정하여 실험하

표 3. 실험 환경

실험 환경	악의적인 피어의 행동	Liar	재참여
Case I	신뢰할 수 없는 자원의 공유	liar가 없음	허용하지 않음
Case II		liar가 없음	허용
Case III		악의적인 피어가 liar로 동작	허용하지 않음
Case IV		몇몇 피어들이 liar로 동작 (악의적인 피어와는 독립적인)	허용

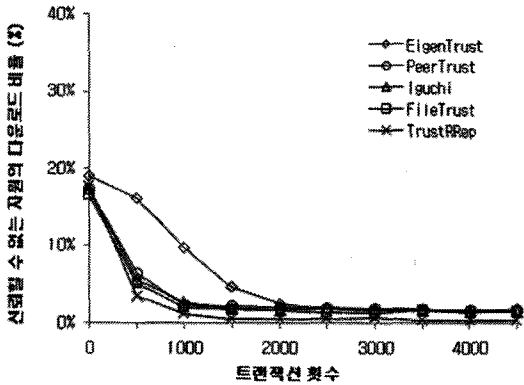


그림 2. Case I

었다.

EigenTrust 기법은 다른 기법(PeerTrust, FileTrust, Iguchi 및 TrustRRep)들과 달리 다운로드 대상을 선택할 때 확률적인 선택 방법을 이용하기 때문에 상대적으로 낮은 신뢰 값을 가지는 피어를 다운로드 대상으로 선택할 수 있다. 그러므로 초기에 신뢰할 수 없는 자원의 다운로드가 증가할 수 있기 때문에 다른 기법들보다 낮은 성능을 보인다. 반면 TrustRRep 기법을 포함한 다른 기법들은 가장 높은 평판이나 신뢰 값을 가지는 자원이나 피어를 다운로드 대상으로 선택하므로 비슷한 성능을 보인다.

그림 3은 Case II의 실험 결과로 악의적인 피어는 신뢰할 수 없는 자원을 제공하고, liar는 없으며, 악의적인 피어들이 신뢰할 수 없는 자원의 제공으로 평판이 낮아질 때 서비스를 종료하고 재참여를 할 수 있는 환경을 가정하였다. 이와 같은 환경에서는 평판 정보의 초기화로 신뢰할 수 없는 자원에 대한 사전

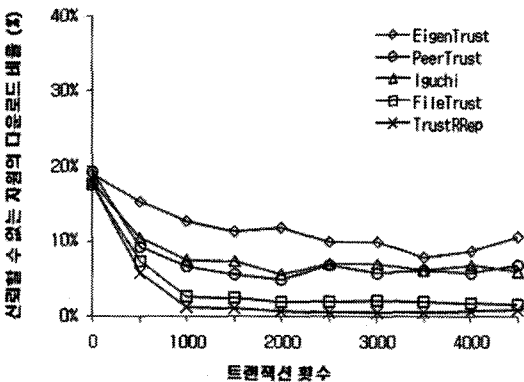


그림 3. Case II

정보가 없기 때문에 피어들은 신뢰할 수 없는 자원을 다운로드 할 수 있다. EigenTrust와 PeerTrust 및 Iguchi 기법은 피어들이 재참여를 할 경우 피어의 평판을 시스템에서 정한 기본 값으로 초기화하므로 성능이 감소한다. 반면 TrustRRep 기법은 자원과 피어의 평판을 구분하여 참조하고 피어가 재참여를 할 경우 식 (3)에 의해 피어가 공유하는 자원의 평판을 참조하여 피어의 평판을 계산하기 때문에 EigenTrust와 PeerTrust 및 Iguchi 기법 보다 좋은 성능을 보인다. FileTrust 기법 역시 피어가 공유하는 자원 평판을 참조하여 피어의 평판을 계산하기 때문에 제안 기법과 비슷한 성능을 보인다.

그림 4는 악의적인 피어가 신뢰할 수 없는 자원을 제공하고, 거짓 평가를 주는 liar로도 동작하며, 피어의 재참여는 허용하지 않는 환경을 가정한 Case III의 실험 결과를 보여준다. EigenTrust 기법은 liar를 식별할 수 있는 방법을 제공하지 않기 때문에 성능이 크게 감소한다. FileTrust와 Iguchi 기법은 동일한 자원을 다운로드 한 피어들이 제공한 평가 값을 참조하여 liar 식별을 위한 정직성(credibility)을 계산하기 때문에 동일한 자원에 대한 평가 값이 없다면 정직성의 계산이 어렵다. PeerTrust 기법은 피어의 평판 정보를 이용하여 악의적인 피어와 liar를 식별한다. 제안 기법인 TrustRRep은 각 피어들의 평가 값과 피어가 평가 값을 제출할 때 발생된 난수 값을 이용하여 평가의 정직성을 확률적으로 확인하고, 식 (1)을 사용하여 피어의 정직성 비율을 계산하여 liar를 판단한다.

그림 5는 Case IV의 실험 결과로 신뢰할 수 없는

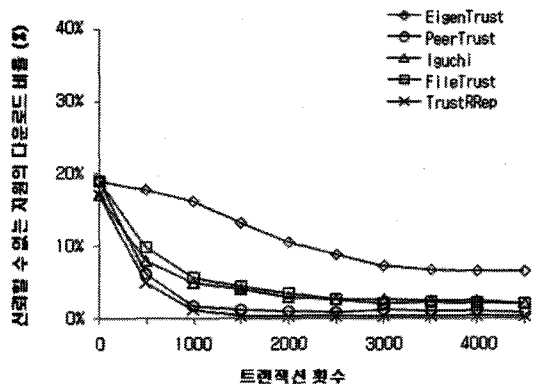


그림 4. Case III

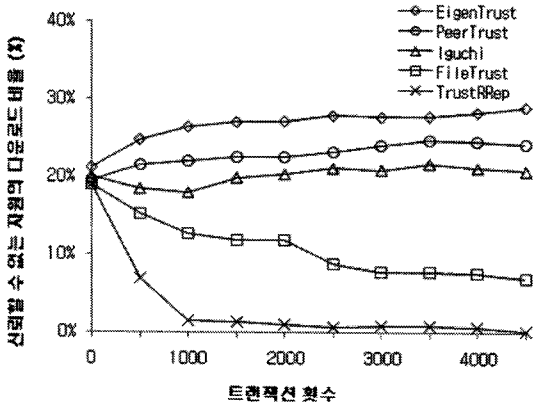


그림 5. Case IV

자원을 제공하는 악의적인 피어와 거짓 평가를 주는 liar가 각각 존재한다. 또 악의적인 피어들은 자신의 평판이 낮아질 때 서비스를 종료하고 재참여를 할 수 있는 환경을 가정하였다. EigenTrust 기법은 악의적인 피어와 liar를 식별하고 제약을 가하는 방법을 제공하지 않기 때문에 낮은 성능을 보인다. PeerTrust 기법은 재참여를 한 피어의 평판을 시스템에서 정한 초기 값으로 설정하므로 재참여를 한 피어의 평판이 상대적으로 높아질 수 있기 때문에 낮은 성능을 보인다. FileTrust 기법은 liar를 식별할 수 있는 정직성(credibility)의 계산 시간이 오래 걸리기 때문에 낮은 성능을 보인다. 또한 기존의 기법들은 신뢰할 수 없는 자원을 제공하는 악의적인 피어의 참여에 제약을 가하지 않기 때문에 악의적인 피어들은 계속적으로 자원을 다운로드 할 수 있다. 반면, TrustRRep 기법은 자원과 피어의 평판을 이용하여 계산되는 신뢰 값을 참조하여 피어의 신뢰 값이 0보다 낮은 경우 다운로드 요청을 허가하지 않기 때문에 악의적인 피어의 참여를 제한할 수 있다.

두 번째 실험은 트랜잭션 증가에 따라 피어의 신뢰 값 변화가 악의적인 피어에서 행해지는 업로드와 다운로드 요청 횟수의 변화에 미치는 영향을 확인하기 위해 Case IV의 환경에서 수행하였다. 그림 6은 악의적인 피어의 in-degree 수에 대한 변화를 보여준다. In-degree는 피어가 다른 피어에게 자원의 다운로드를 요청하는 횟수를 의미한다. 제안 기법에서는 악의적인 피어의 신뢰 값이 0보다 작은 경우 자원의 다운로드 요청을 거부하므로 트랜잭션이 증가할수록 in-degree의 수가 감소한다. 반면 다른 기법

(EigenTrust, PeerTrust, FileTrust 및 Iguchi) 들은 피어의 평판이나 신뢰 값이 낮아지더라도 특별한 제약을 가하지 않기 때문에 비슷한 성능을 보인다.

그림 7은 악의적인 피어의 out-degree 수에 대한 변화를 보여준다. Out-degree는 피어가 다른 피어에게 자원을 제공(업로드)하는 횟수를 의미한다.

EigenTrust 기법은 악의적인 피어를 식별하고 제약을 가하는 방법을 제공하지 않기 때문에 out-degree 수의 변화가 거의 없다. PeerTrust와 Iguchi 및 FileTrust는 피어의 평판 정보를 이용하여 악의적인 피어를 처리하므로 트랜잭션에 따라 out-degree의 수가 감소한다. 제안 기법 역시 악의적인 피어의 신뢰할 수 없는 자원의 제공으로 식 (2)에 의해 자원 평판이 감소하고, 이는 식 (3)에 의한 피어 평판 계산과 식 (4)에 의한 피어 신뢰 값 계산까지 영향을 미친다. 그러므로 트랜잭션이 증가할수록 신뢰 값의 감소로 out-degree의 수가 감소한다. 이는 악의적인 피어가 가진 자원을 요청하는 피어의 수가 점차적으로

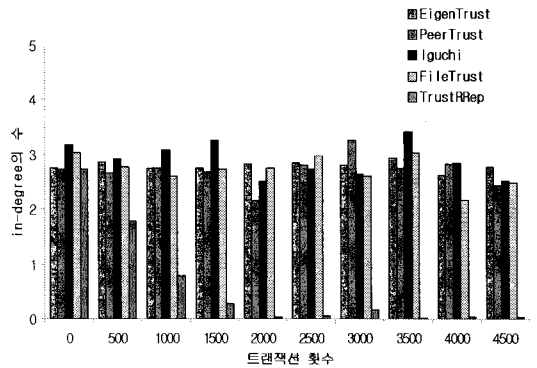


그림 6. 악의적인 피어의 in-degree 수

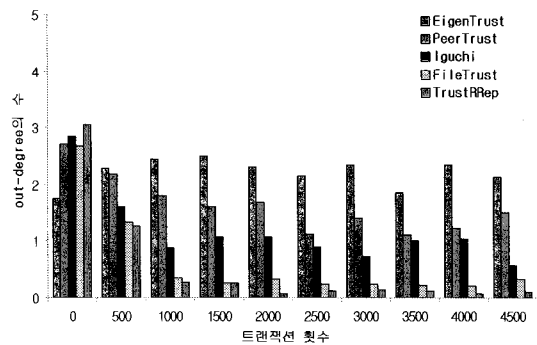


그림 7. 악의적인 피어의 out-degree 수

감소하는 것을 의미한다.

한편, 악의적인 피어가 올바른 피어보다 높은 신뢰 값을 가질 경우 자원에 대한 다운로드 요청을 받을 확률이 높아지고, 이러한 피어가 많아질수록 시스템 내에서 신뢰할 수 없는 자원의 확산이 빠르게 일어난다. 이에 Case IV의 환경에서 올바른 피어보다 높은 신뢰 값을 가지는 악의적인 피어의 수를 살펴보았다. 그림 8은 가장 낮은 신뢰 값을 가지는 올바른 피어보다 높은 신뢰 값을 가지는 악의적인 피어의 수를 보여준다.

EigenTrust와 PeerTrust 기법은 피어의 재참여를 고려하지 않기 때문에 악의적인 피어들의 신뢰 값은 시스템에서 정한 값으로 초기화된다. 그러므로 많은 수의 악의적인 피어들이 초기 트랜잭션에서 올바른 피어보다 높은 신뢰 값을 가지는 경우가 발생할 수 있다. FileTrust 기법은 liar의 식별을 위한 정직성을 계산할 때 특정 자원에 대해 자신의 평가 값과 다른 피어의 평가 값을 참조한다. 그러므로 동일한 자원에 대한 평가가 없을 경우 정직성 계산에 오류가 발생할 수 있고, 이는 자원의 평판뿐만 아니라 피어의 신뢰 값 계산에도 영향을 미친다. 반면 TrustRRep 기법은 다수의 피어들의 자원 평가 값을 참조하고 식 (5)에 의해 생성된 값을 참조하여 식(6)과 (7)을 사용하여 자원 평가의 정직성을 확인한다. 그리고 식 (1)을 사용하여 정직성 비율을 계산하여 liar를 판단한다. 그리고 liar로 판단된 피어로부터 받은 평가 값은 식 (2)의 자원 평판 계산에 반영하지 않기 때문에 이 값을 이용하여 계산되는 식 (3)의 피어 평판과 식 (4)의 신뢰 값에 대한 정확도를 향상시킬 수 있다.

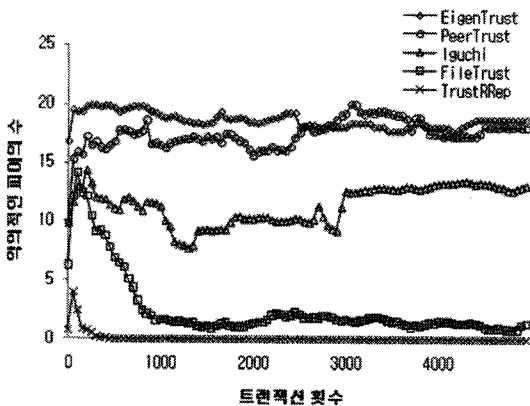


그림 8. 악의적인 피어의 계산 오류

5. 결 론

평판은 특정 사용자와 과거에 트랜잭션을 수행한 경험이 있는 사용자들에 의해 제공되는 정보로 P2P 서비스와 같은 분산 환경에서 사전 트랜잭션 경험이 없는 사용자들에게 믿을 수 있는 트랜잭션 대상자를 선택할 수 있도록 도움을 준다. 이러한 도움을 주기 위해서는 P2P 서비스에서 평판을 참조할 때 트랜잭션 결과에 대해 거짓 평가(false feedback)를 주거나 특정 사용자에 대한 평판의 증가 및 감소를 공모하는 사용자들의 악의적인 행동에 효과적으로 대응할 수 있는 기법이 필요하다.

본 논문에서는 공유 자원과 피어의 평판, 그리고 피어의 신뢰 값을 참조하여 자원과 피어에 대한 신뢰성을 제공하고, 자원 요청 피어의 평가의 정직성 및 liar의 식별을 제공할 수 있는 TrustRRep 기법을 제안하였다. NS-2 시뮬레이터를 이용한 실험을 통하여 TrustRRep 기법에서 신뢰할 수 없는 자원의 다운로드 감소와 악의적인 피어의 신뢰 값 계산이 올바르게 수행되어 신뢰할 수 없는 자원을 제공하는 악의적인 피어의 서비스 참여가 제한되는 것을 확인하였다. 따라서 제안한 TrustRRep 기법은 악의적인 피어와 liar의 참여를 제한함으로써 신뢰할 수 있는 자원의 다운로드 증가와 피어들이 제공하는 평판의 신뢰성을 제공함으로써 P2P 네트워크의 신뢰성과 효율성을 향상시킬 수 있다.

참 고 문 헌

- [1] Y. Wang and J. Vassileva, "Bayesian Network Trust Model in Peer-to-Peer Networks," *Proceedings of the 2nd International Workshop*, Springer-Verlag, Lecture Notes in Computer Science 2872, pp. 23-34, 2005.
- [2] Y. Zou, L. Gu, G. Li, B. Xie, and H. Mei, "Rectifying Prejudicial Feedback Ratings in Reputation based Trust Management," *IEEE International Conference on Services Computing*, pp. 530-535, 2007.
- [3] J. Chang, H. Wang, G. Yin, and Y. Tang, "A New Reputation Mechanism Against Dishonest Recommendations in P2P Systems," *Pro-*

ceedings of the 8th International Conference on Web Information Systems Engineering Nancy, Springer-Verlag, Lecture Notes in Computer Science 4831, pp. 449-460, 2007.

- [4] K. Fall and K. Varadhan, "The ns-2 manual," Technical Report, The VINT Project, UC Berkeley, LBL, and Xerox RARC, 2002.
- [5] S. S. Standifird, "Reputation and e-Commerce: eBay Auctions and the Asymmetrical Impact of Positive and Negative Ratings," *Journal of Management*, Vol.27, No.3, pp. 279-295, 2001.
- [6] S.D. Kamvar, M.T. Schlosser, and H. Garcia-Molina, "The Eeigentrust Algorithm for Reputation Management in P2P Networks," *Proceedings of the 12th International Conference on World Wide Web*, pp. 640-651, 2003.
- [7] L. Xiong and L. Liu, "PeerTrust: Supporting Reputation-based Trust for Peer-to-Peer Electronic Communities," *IEEE Transactions on Knowledge and Data Engineering*, Vol.16, No.7, pp. 843-857, 2004.
- [8] O.H. Kwon, S.Y. Lee, and J. Kim, "FileTrust: Reputation Management for Reliable Resource Sharing in Structured Peer-to-Peer Networks," *IEICE Transactions on Communications*, Vol.E90-B, No.4, pp. 826-835, 2007.
- [9] M. Iguchi, M. Terada, and K. Fujimura, "Managing Resource and Servent Reputation in P2P Networks," *Proceedings of the 37th Hawaii International Conference on System Sciences*, Track 7, Vol.7, 2004.
- [10] L. Mekouar, Y. Iraqi, and R. Boutaba, "Peer-to-Peer's most wanted: Malicious peers," *In International Computer Networks Journal, Special issue on management in Peer-to-Peer Systems: Trust, Reputation and Security*, Vol.50, No.4, pp. 545-562, 2006.
- [11] Y. Jin, Z. Gu, J. Gu, and H. Zhao, "A New Reputation-Based Trust Management Mechanism Against False Feedbacks in Peer-

to-Peer Systems," *Proceedings of the 8th International Conference on Web Information Systems Engineering Nancy*, Springer-Verlag, Lecture Notes in Computer Science 4831, pp. 62-73, 2007.

- [12] E. Adar and B. A. Huberman, "Free Riding on Gnutella," http://www.firstmonday.dk/issues/issue5_10/adar, 2000.



신 정 화

1997년 2월 한국방송통신대학교 컴퓨터과학과 졸업
 2000년 8월 부경대학교 전산정보학과 석사
 2006년 8월 부경대학교 전자계산학과 박사

2007년 9월~현재 부산대학교 U-Port정보기술산학공동사업단 전임연구원

관심분야 : 암호이론, P2P Security, Reputation Management System, Incentive Mechanism



김 태 훈

2006년 2월 부산대학교 정보컴퓨터공학부 졸업
 2008년 2월 부산대학교 컴퓨터공학과 석사
 2008년 3월~현재 부산대학교 컴퓨터공학과 박사과정

관심분야 : 위치인식, P2P, 토폴로지 제어



탁 성 우

1995년 2월 부산대학교 컴퓨터공학과 졸업
 1997년 2월 부산대학교 컴퓨터공학과 석사
 2003년 2월 미국미주리주립대학교 ComputerScience 박사
 2004년~현재 부산대학교 정보컴퓨터공학부 부교수

2004년~현재 부산대학교 컴퓨터 및 정보통신 연구소 겸임 연구원

관심분야 : 유무선 네트워크, Soc 설계, 실시간 시스템, 위치인식, 최적화 기법, 그래프 이론, 큐잉이론