DPay: 피어-투-피어 환경을 위한 분산 해시 테이블 기반의 소액 지불 시스템

(DPay: Distributed-Hash-Table-based Micropayment System for Peer-to-Peer Environments)

서 대 일 † 김 수 현 †† 송 규 원 †
(Daeil Seo) (Suhyun Kim) (Gyuwon Song)

요 약 피어-투-피어(P2P) 시스템은 참여하는 사용자가 소유하고 있는 리소스를 서로 간에 공유하여 많은 이득을 얻을 수 있게 해준다. 그러나 사용자들이 악의적인 행동을 하거나 오프라인일 때, P2P 시스템이나 응용 프로그램들은 정상적인 서비스를 제공하는데 문제가 발생 할 수 있다. 소액 지불 시스템을 이용하여 서비스 제공에 대해 보상을 해준다면 이러한 문제를 해결하는데 도움이 될 수 있다. 지금까지 대부분의 소액 지불 시스템은 중앙 집중화 된 브로커를 사용하는데, 브로커에 많은 부하를 발생시키는 문제점을 가지고 있다. 예를 들어, 코인의 소유자가 오프라인인 경우 브로커가 소유자를 대신하여 지불 정보를 처리하는데 이는 브로커의 부하를 증가시키는 주요 원인이며 P2P 시스템 특성상 매우 빈번하게 발생 할 수밖에 없다. 본 논문에서 제안하는 DPay는 P2P 환경을 위한 소액 지불 시스템으로, 분산 해시 테이블을 이용하여 모든 암호화된 지불 정보를 안전하게 기록하고 다운타임 프로토콜을 사용하지 않음으로써 브로커의 부하를 획기적으로 감소시켜 시스템의 확장성을 크게 향상 시킨 시스템이다. 또한 실시간 중복 결제 검출 방법을 제안하고, DPay와 기존 지불 시스템 간의 비교와 실험 결과를 제시한다. 실험결과 DPay는 브로커의 부하가 기존 시스템 대비 평균 30%로 줄어든 것으로 나타났으며, 실시간 중복 결제 검출과 보다 안전한 지불 정보 기록을 가능하게 하여 다양한 P2P 시스템에 적용할 수 있을 것으로 기대된다.

키워드 : 소액 지불 시스템, 피어-투-피어, 분산 해시 테이블

Abstract Emerging peer-to-peer systems benefit from the large amount of resources provided by many peers. However, many peer-to-peer systems or applications suffer from malicious peers and it is not guaranteed that peers are always online. Micropayment systems are accounting and charging mechanism for buying services, so we can apply them to solve these problems. In the past the majority of micropayment system uses a centralized broker but the problem with most existing micropayment system is a heavy load on the broker. For instance, when an owner of the coin is offline, the broker delegates the owner and handles payment messages. It occurs frequently because of characteristic of peer-to-peer system and is another load of the broker. In this paper we introduce DPay, a peer-to-peer micropayment system that uses distributed hash table (DHT) for storing encrypted payment messages and increases scalability and reduces the load of broker by removing downtime protocol. We show the idea of real-time double spending detection in DPay and report the results of several evaluations in order to compare DPay and other payment scheme. In simulation result, the load

· 본 연구는 지식경제부 및 한국산업기술평가관리원의 산업원천기술개발사업의 일환으로 수행하였음[2009-S-036-01, 고성능 가상머신 규격 및 기술 개발]

† 정 회 원 : 과학기술연합대학원대학교 HCI 및 로봇응용공학
xdesktop@imrc.kist.re.kr
sharp81@imrc.kist.re.kr
†† 정 회 원 : 한국과학기술연구원 영상미디어연구센터 선임연구원
dr.suhyun.kim@gmail.com

논문접수 : 2009년 6월 22일
심사완료 : 2009년 8월 2일

정보과학회논문지: 컴퓨팅의 실제 및 레터 제15권 제10호(2009.10)

of broker in DPay is reduced by 30% on average of other previous payment scheme. We expect that DPay can apply various peer-to-peer systems because it provides a real-time double spending detection and stores more secure payment messages.

Key words : Micropayment system, Peer-to-Peer, Distributed Hash Table (DHT)

1. 서 론

Peer-to-peer(P2P) 시스템은 시스템에 참여하는 사용자로부터 제공되는 많은 양의 리소스를 통해 큰 이득을 얻고 있다. P2P 시스템은 컴퓨터의 저장 공간, 프로세싱 사이클 등의 리소스들을 사용자간에 공유할 수 있는 방법을 제공한다. 예를 들어, BitTorrent[1]와 eMule Project[2]는 신뢰할 수 있는 방법을 통하여 다수의 데이터를 서로 간에 공유하고 있다. 이에 더해 P2P 시스템은 전통적인 server-client 구조에 비해 더 높은 확장성, 고성능을 제공한다. P2P 시스템은 자동적으로 복사본을 생성하여 다른 참여자에게 동일한 정보를 기록하고, Chord[3], Pastry[4]와 같은 오버레이를 통해 효율적으로 데이터에 접근할 수 있는 방법을 제공한다. 그러나 이러한 P2P 시스템과 응용 프로그램들은 free-riding과 악의적인 행동을 하는 참여자에 의해 문제가 발생하고 있는데, 문제를 해결하기 위해서는 참여하는 사용자간에 제공하는 서비스에 대한 비용을 서로 간에 지불하고 받을 수 있는 방법이 필요하다.

소액 지불 시스템은 빈번히 일어나는 작은 금액의 거래를 효율적으로 하기 위해 디자인 된 시스템으로 대략 $10 이하의 거래를 대상으로 한다. 뉴스, 논문 음악 등의 디지털 콘텐츠의 구매에 적합하며 P2P 시스템에서 서비스의 구매에도 사용될 수 있을 것이다. 소액 지불 시스템은 대상 상품 가격이 높지 않으므로 지불 처리에 사용되는 비용이 너무 높아서는 곤란하다. 소액 지불은 작은 금액에 대한 지불이므로 최대한의 보안이 요구되지 않는다는 점을 이용하여 경량화된 지불 메커니즘이 사용된다.

PPay[5]와 같은 P2P 환경에서의 소액 지불 시스템은 시스템에 참여하고 있는 사용자들이 판매자와 구매자가 모두 될 수 있다는 특징을 지닌다. 이런 상황에서는 P2P 시스템의 특징을 살려서 참여하고 있는 사용자들의 남는 리소스를 활용하여 브로커의 부하를 나눌 수 있다면 보다 높은 확장성과 성능을 가지는 소액 지불 시스템을 구축할 수 있게 된다.

하지만 P2P 시스템은 사용자가 시스템에 빈번하게 출입을 반복하며, 네트워크의 문제로 일시적인 통신장애가 발생하여 기본 프로토콜을 적용할 수 없는 경우가 자주 발생한다. 위와 같은 원인에 의해 코인의 소유자가 오프라인일 때 PPay는 다운타임 프로토콜을 사용한다. 다운타임 프로토콜이 수행되면 브로커가 소유자를 대신하여 코인을 할당하기 위한 메시지를 처리하고, 이 메시지들을 기록한다. 위 방법은 브로커에게 또 다른 부하를 발생시키는 원인이 된다.

본 논문에서는 새로운 P2P 기반 소액 지불 시스템인 DPay를 제안한다. DPay는 PPay와 비교하여 분산 해시 테이블에 사용자 간의 거래 정보를 저장하고 다운타임 프로토콜의 사용을 제거함으로써 브로커의 부하를 좀 더 줄여 보다 나은 확장성을 제공할 수 있다. 또한 코인의 소유자가 오프라인일 때 기존 방법은 브로커가 코인의 소유자를 대신하여 지불 과정에 참여하지만, 본 논문에서는 브로커가 지불과정에 참여하지 않고 피어들이 분산 해시 테이블에 있는 이전 지불 정보를 직접 확인하여 처리한다. 이에 더하여 분산 해시 테이블에 저장되어 있는 지불 정보를 이용한 실시간 중복 결제 검출 방법을 제공한다. 본 논문은 다음과 같은 구성을 가진다. 다음 장은 본 논문과 관련된 관련 연구에 대해 살펴보고 3장은 DPay를 이해하기 위한 배경지식을 제공하고, 4장은 DPay의 알고리즘을 제시한다. 5장은 DPay와 다른 소액 지불 시스템의 비교를 위한 실험과 평가를 보여주고 6장에서 결론을 제시한다.

표기법: 본 논문에서는 아래 표 1과 같은 기호를 사용하기로 한다.

표 1 프로토콜에 사용되는 기호

기호	설명
B, U, V, W	브로커(B) 및 사용자(U,V,W)의 식별자
SK_X, PK_X	X의 비밀 키와 공개 키
SK_{BC}, PK_{BC}	코인 서명을 위한 비밀 키와 공개 키
$\{msg\}K_X$	키 K_X를 사용하여 서명된 메시지
<k, v>	분산 해시 테이블에 저장되는 키와 값

2. 관련 연구

소액 지불 시스템은 작은 금액의 지불을 처리하기 위하여 고안되었으며, 예전부터 많은 종류의 소액 지불 시스템이 제안되었다. Millicent[7]는 Digital Equipment에 의해 제안된 방법으로 scrip이라는 특정 판매자에 종속적인 전자 화폐를 이용하는 방법으로 사용에 까다로움이 있었다. NetBill[8]은 e-commerce 프로토콜로 익명성을 제공하는 소액 지불 시스템이다. 그러나 처리 속도가 늦다는 단점을 가지고 있다. 위에 제시된 시스템들은 지불 과정에 매번 브로커가 참여하게 되는데 이러한

이유로 총 지불 횟수가 n번인 경우 O(n)의 브로커의 부하가 발생된다.

PayWord[9]는 신용 기반의 오프라인 지불 시스템으로 특정 사용자와 판매자에 종속적인 payword의 해시 체인을 이용하는 방법이다. PayWord는 공개키 기반의 암호화보다 비용이 적은 해시 함수를 사용하여 비용을 줄이는 방법을 사용하고 있다. 이러한 PayWord는 기존의 방법에 비하여 브로커의 부하를 획기적으로 줄였으나 각각의 사용자는 판매자로부터 서비스를 구매할 때마다 각각의 판매자에 맞는 다른 해시 체인을 만들어 내야하는 문제를 가지고 있다. NetPay[10]는 PayWord에 기반을 둔 안전하고 비용이 적게 드는 오프라인 지불 시스템이다. NetPay는 사용자가 사용하는 해시 체인이 브로커에 의하여 생성되기 때문에 payword가 특정 판매자에 종속적이지 않게 된다. 즉, 생성된 해시 체인을 다른 판매자에게도 사용할 수 있다.

PPay[5]와 WhoPay[11]는 P2P환경을 위한 소액 지불 시스템으로 공개키 기반의 방법을 사용하고 있다. PPay는 유동적이고 스스로 관리하는 화폐를 사용하고 있다. 유동적이라는 의미는 다른 사용자에게 받은 전자화폐를 중앙화된 브로커를 거치지 않고 다른 사람에게 지불 할 수 있다는 것을 말한다. 스스로 관리되는 화폐는 코인에 관련된 메시지가 모두 처음에 코인을 발급받은 소유자를 통해서 이루어지기 때문이다. WhoPay는 그룹 서명 방법을 이용하여 공평성을 취하고 있다. 이러한 시스템들의 주요 목적은 브로커의 부하를 줄여 확장성을 높이는데 있다. 브로커는 코인을 발급하거나 코인의 유효기간 갱신, 환전에만 참여하도록 하고, 다운타임 프로토콜을 이용하여 소유자가 오프라인인 경우에만 코인의 할당 메시지를 처리한다. 그러나 이러한 다운타임 프로토콜에 브로커가 참여하여 추가적인 부하가 발생된다. 또한 피어가 다시 네트워크에 들어오는 경우 그동안 발생했던 지불 정보를 동기화하기 위한 추가 비용이 발생된다. 그 이후에는 다시 코인의 소유자가 지불 정보를 처리한다.

3. 배경 지식

3.1 PPay

PPay[5]는 peer-to-peer 환경을 위한 소액 지불 시스템이다. 이러한 PPay는 브로커와 사용자, 두 가지의 행위자가 있다. 브로커는 코인을 사용자에게 제공하는 일을 하고, 브로커로부터 코인을 산 사용자는 그 코인의 소유자(owner)가 된다. 처음에 결정된 코인의 소유자는 거래가 일어나는 동안 변하지 않는다. 브로커 B로부터 코인 C를 구입한 사용자 U는 다른 사용자 V에게 서비스를 구입하면 코인 C를 이용하여 대가를 지불한다. 코인 C를 받은 사용자 V는 그 코인의 보유자(holder)가 되지만, 코인의 소유자는 변경되지 않는다. 이후 코인 C에 대한 거래는 모두 소유자인 사용자 U를 통해 이루어진다. 이후 사용자 V가 다른 사용자 W로부터 서비스를 구매하면, 사용자 V는 코인 C의 소유자인 사용자 U에게 코인의 보유자 변경을 요청하고, 사용자 U는 이 요청을 처리하여 코인의 보유자를 사용자 V에서 W로 변경한다. PPay는 소액 지불은 작은 금액에 대한 지불이므로 최대한의 보안이 요구되지 않으며, 지불 메커니즘은 반드시 경량이어야 하고 지불하고자하는 가치를 넘어서면 안된다고 정의하고 있다. 다음은 PPay에 대한 상세 프로토콜이다.

사용자 U가 브로커 B로부터 코인을 구입하면, 브로커로부터 전송되는 코인은 다음과 같다.

$$C = \{U, sn\}_{SKB}$$

sn은 코인의 순차 번호로 코인을 구별하기 위한 식별자이다. 사용자 U가 사용자 V로부터 서비스를 구입하는 경우 코인 C는 다음과 같은 과정을 거쳐 사용자 V에게 할당된다.

$$A_{UV} = \{V, seq_1, C\}_{SKU}$$

seq_1는 코인이 할당된 순차번호이고 사용자 V는 코인의 새로운 보유자이다. 사용자 V는 이 코인을 다른 사용자와의 거래에 사용할 수 있는데, 사용자 W로부터 서비스를 구입한 후 코인 C를 이용하기 위해서는 사용자 U에게 재할당 요청 메시지를 보낸다.

$$R_{UVW} = \{W, A_{UV}\}_{SKV}$$

코인의 소유자인 사용자 U는 사용자 V가 코인을 양도한 것을 추후에 증명하기 위하여 재할당 메시지를 보유하고, 새로운 할당 메시지를 사용자 V와 W에게 전송한다.

$$A_{UW} = \{W, seq_2, C\}_{SKU}$$

seq_2는 반드시 seq_1보다 반드시 큰 값을 가져야 한다. 사용자 W가 메시지를 받은 후 사용자 W가 코인의 새로운 보유자가 된다. 코인을 관리하기 위해서는 반드시 코인의 소유자가 온라인이어야 하는데, 만약 코인의 소유자인 사용자 U가 오프라인인 경우, PPay는 다른 사용자로부터 받은 코인을 사용하기 위하여 다운타임 프로토콜을 사용하게 된다. 이때 에는 브로커가 코인의 소유자를 대신하여 코인 전달을 위한 메시지들을 처리하고 보관한다. 즉, 다운타임 프로토콜에서는 코인의 소유자 역할을 대신하여 브로커가 코인 전달 메시지를 처리한다. 이를 통하여 사용자는 다른 사용자로부터 받은 코인을 코인의 소유자가 아닌 브로커를 통하여 재할당 할 수 있다. 사용자 W는 코인의 소유자 U가 오프라인인 것을 확인하면, 코인의 보유자 변경을 위한 재할당 메시지를 브로커에 보낸다. 이를 받은 브로커는 재할당 메시

지를 코인의 소유자가 처리한 것과 동일하게 처리하며 메시지의 서명만 자신의 키를 이용한다. 이렇게 받은 재할당 메시지는 코인의 소유자 U가 온라인이 될 때까지 브로커가 보관한다. 코인의 소유자 U가 다시 시스템에 들어오면, 소유자는 반드시 소유자가 오프라인 동안 발생했던 코인 재할당 메시지에 대하여 브로커와 동기화를 한다.

3.2 분산 해시 테이블(DHT)

분산 해시 테이블은 비구조적인 peer-to-peer 시스템이 브로드캐스팅을 통해 피어들의 데이터를 검색하면서 발생하는 비효율성의 문제를 개선하기 위해 등장하였다. 검색의 효율성을 높이기 위해 분산 해시 테이블을 이용하여 각각의 피어가 라우팅 테이블을 가지고 이를 통하여 정형화된 검색방법을 제공한다. 이러한 DHT 시스템으로는 Chord[3], Pastry[4] 등의 시스템이 연구되었다. 이러한 시스템은 SHA-1과 같은 해시 함수를 사용하여 각각의 노드와 키에 m-bit의 식별자를 할당한다. 할당된 식별자는 해시 성질에 의해 오버레이 네트워크에 pseudo random한 위치에 분포하며 노드들은 노드의 식별자를 이용하여 자신이 관리해야 할 오브젝트의 키를 할당받고 이에 따라 노드들 간에 로드 밸런싱이 이루어진다. 분산 해시 테이블은 <k, v>의 쌍으로 이루어진 key, value를 이용하여 오브젝트에 대한 정보를 저장한다.

분산 해시 테이블은 분산, 확장성, 내결함성을 특징으로 한다. 분산 해시 테이블을 이용하면 시스템에 참여하는 노드들은 중앙에서 조정을 담당하는 노드가 필요하지 않고, 수십만에서 수백만의 노드가 시스템에 참여하더라도 효율적으로 처리할 수 있는 확장성을 가진다. 대부분의 N개의 노드를 가지는 분산 해시 테이블은 테이블에 있는 데이터를 조회하는데 걸리는 시간이 $O(logN)$으로 결정된다. 또한 노드가 빈번하게 네트워크에 계속적으로 출입을 반복하여 node churn이 발생하게 되면 P2P 시스템에 악영향을 주는데 이러한 node churn이 자주 발생하더라도 저장된 데이터를 복제하거나 정보를 퍼트리게 함으로서 시스템은 신뢰성을 제공한다. 이 논문에서는 소액 지불 시스템에서 브로커의 확장성을 높이기 위해 분산 해시 테이블을 사용하는 방법을 사용한다.

4. DPay 디자인

4.1 DPay 개요

DPay의 가장 중요한 특징은 브로커가 기존 방법은 코인의 소유자가 오프라인일 때 다운타임 프로토콜이 사용되어 브로커의 부하가 발생하였는데, DPay는 이러한 다운타임 프로토콜을 제거함으로써 브로커의 부하를 줄인다. DPay는 PPay에서 사용하는 프로토콜을 기반으로 구현된 소액 지불 시스템으로, PPay와 유사하게 DPay는 코인을 사용하며, 이 코인은 다른 사용자에게 재할당 할 수 있다. 사용자는 브로커로부터 코인을 구입하여 사용하고, 사용자는 이를 이용하여 다른 사용자로부터 서비스의 구입에 이용한다. DPay와 PPay의 가장 중요한 차이점은 브로커가 사용자간의 거래에 얼마나 참여하는지에 있다. DPay는 분산 해시 테이블을 사용하여 코인의 소유자가 오프라인인 경우에 다운타임 프로토콜을 사용하지 않고, 브로커의 참여 없이 사용자들이 코인을 사용할 수 있다. 사용자 간에 코인이 할당된 정보들은 분산 해시 테이블에 기록되며 자동적으로 사본을 생성되어 다른 피어들에게 복제된다. 사용자는 언제나 이전의 코인 할당 정보를 분산 해시 테이블을 조회함으로써 알 수 있다. 또 다른 차이점은 DPay는 실시간 중복 결제 검출 방법을 제공한다. 사용자는 코인의 지불이 끝난 이후에 그 정보를 분산 해시 테이블에 기록한다. 이 정보를 이용하여 사용자는 필요할 때마다 코인이 사용된 정보들을 추적할 수 있다. 그림 1은 DPay에 대한 전체 흐름도이다.

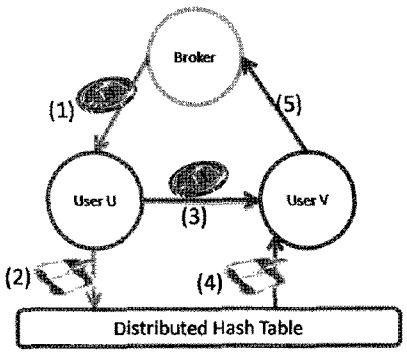

그림 1 DPay의 전체 흐름도. (1) 사용자 U가 코인 구입 (2) 사용자 U가 코인 정보를 DHT에 기록 (3) 사용자 U가 코인을 지불 (4) 사용자 V가 DHT를 조회해 지불 정보를 확인 (5) 사용자 V가 코인 환전 요청

4.2 DPay 프로토콜

DPay 프로토콜은 다음과 같이 구성된다. 사용자가 다른 사용자로부터 서비스를 제공받은 후 지불을 하기 위해서는 코인이 필요하다. 이에 따라 사용자 U는 브로커 B로부터 코인 C를 구입한다. 브로커는 코인 구입 요청에 따라 코인을 발급한다. 브로커는 발급을 요청한 사용자를 구별하기 위한 식별자 값인 U와 코인의 식별값으로 사용할 수 있는 순차번호 sn, 코인의 유효기간을 나타내는 $expired$를 이용하여 코인을 발급한다. 코인은 유효기간이 만료되면 환전되거나 유효기간을 갱신해야 한다.

$$C = \{U, sn, expired\}_{SKBC}$$

생성된 코인은 브로커의 비밀 키에 의하여 서명되고 난 후 사용자 U에게 전송된다. seq_0는 지불의 순차번호 값으로 초기 값은 0을 사용한다. seq는 매번 거래마다 1씩 증가한다.

$$T_B = \{C, seq_0\}_{SKB}$$

브로커로부터 코인을 받은 사용자 U는 코인 발급 정보를 분산 해시 테이블에 기록한다. 키 값으로는 코인과 지불의 순차번호를 해시한 값을 사용하며 사용자의 비밀 키로 받은 코인 정보와 코인의 발급자인 브로커의 식별자 값 B를 서명하여 저장한다.

$$<h(C, seq_0), T_{BU} = \{T_B, B\}_{SKU}>$$

사용자 U가 사용자 V로부터 서비스를 구매하고 코인을 지불하기 위해서는 사용자 U는 자신의 비밀 키를 이용하여 코인을 서명한 후 사용자 V에게 전송한다. 지불 순차번호 seq_1는 seq_0 보다 1 큰 값으로 설정된다.

$$T_U = \{C, seq_1\}_{SKU}$$

사용자 V가 사용자 U로부터 지불 정보를 받으면 코인의 정보와 현재 지불 순차번호를 알 수 있다. 이 정보를 이용하여 사용자 V는 분산 해시 테이블에서 정보를 조회한다. 이를 통해 실시간 중복 결제 검출이 가능하다.

$$<h(C, seq_1), T_{UV} = \{T_U, U\}_{SKV}>$$

사용자 V가 사용자 W로부터 서비스를 구입하면 사용자 V는 사용자 U로부터 받은 코인을 사용한다. 이때 코인은 유효기간이 만료되지 않아야 한다. 사용자 V는 지불 순차번호를 1만큼 증가시키고 자신의 비밀 키로 서명한 후 사용자 W에게 전송한다.

$$T_V = \{C, seq_2\}_{SKV}$$

사용자 W가 사용자 V로부터 지불 정보를 받으면, 사용자 W는 분산 해시 테이블을 조회하여 현재 지불 정보의 이상여부를 확인한다. 받은 지불 정보가 이상이 없으면 사용자 W는 현재 지불 정보를 분산 해시 테이블에 기록한다.

$$<h(C, seq_2), T_{VW} = \{T_V, U\}_{SKW}>$$

코인의 유효기간이 만료되면 사용자는 코인의 유효기간의 갱신을 요청하거나 코인을 현금으로 환전해야 한다. 사용자가 코인을 현금으로 환전을 요청하면 브로커는 코인의 순차 번호와 최종 지불 순차번호를 이용하여 코인이 사용된 정보를 분산 해시 테이블에서 조회하여 확인한다. 이상이 없는 경우 사용자에게 코인을 현금으로 환전해 준다. 사용자가 브로커에게 코인의 유효기간 연장을 요청하는 경우, 브로커는 이전의 지불 정보를 확인한 후 이상이 없는 경우에 코인에 새로운 유효기간을 정하여 사용자에게 코인을 돌려준다.

일반적으로 코인의 유효기간이 지난 경우, 사용자는 코인의 유효기간을 연장하기 위하여 브로커에게 갱신을 요청한다. 사용자 W는 코인의 유효기간을 갱신하기 위해 가지고 있는 코인을 브로커에게 보낸다.

$$T_W = \{C, seq_i\}_{SKW}$$

코인을 받은 브로커는 이전 지불 순차번호인 seq_{i-1}에서 0까지의 지불 정보를 분산 해시 테이블에서 조회한다. 브로커는 조회된 지불 정보를 확인하여 이상이 없는 경우에는 새로운 유효기간을 가지는 새로운 코인 C'을 생성한다.

$$C' = \{U, sn, expired'\}_{SKBC}$$

U는 처음 코인을 브로커로부터 구입한 사용자의 식별자이고, sn은 코인의 순차 번호이다. 코인의 유효기간이 갱신되면 sn은 0으로 초기화되고 $expired'$는 새로운 유효기간으로 설정된다. 브로커는 코인의 유효기간의 갱신이 완료되면 코인 C에 대한 지불 정보를 분산 해시 테이블에서 삭제를 요청한다. 사용자는 브로커로 받은 유효기간이 갱신된 코인을 다른 코인과 동일하게 사용한다.

4.3 실시간 중복 결제 검출

사용자가 다른 사용자로부터 코인을 받으면, 분산 해시 테이블을 이용하여 이전의 지불 정보에 대해서 조회한다. 이 정보를 이용하여 사용자는 지불 정보를 기록한 사람을 확인할 수 있고, 지불 정보가 올바르게 기록되었는지 확인할 수 있다. 사용자가 잘못된 코인을 사용한 경우 코인을 받은 사용자에 의해 검출이 가능하다. 잘못된 코인의 경우 이전 지불 정보가 분산 해시 테이블에 없거나 잘못된 지불 정보가 기록되어 있다. 사용자가 지불 정보를 조회하고 나면 코인의 서명을 확인한다. 만약 코인이 브로커의 비밀 키에 의해서 생성된 것이 아니라면 지불 정보는 잘못된 것이 된다. 그리고 분산 해시 테이블에서 현재 지불 순차번호인 seq_i가 조회되거나 이전 지불 순차번호인 seq_{i-1}이 조회되지 않으면 올바른 지불이 아니다. 정상적인 지불 과정은 지불 순차번호는 1씩 증가한다. 현재 지불 순차번호가 조회된다는 것은 이미 이전에 누군가에게 지불되었다는 의미이고, 이전의 순차번호가 조회되지 않으면 순차번호가 비정상적으로 증가하였거나 잘못된 코인을 이용한 것이기 때문이다.

5. 실험 및 평가

이전 장에서는 DPay 알고리즘에 대하여 살펴보았다. 이 장은 다음과 같은 질문의 답을 구하기 위해 실험을 수행하였다. (1) DPay는 분산 해시 테이블을 이용하여 지불 정보를 기록하면 PPay에 비하여 브로커의 부하를 줄일 수 있는가? (2) 노드의 수가 증가함에 따라 브로커의 부하는 어떠한 변화를 보이는가? (3) 노드의 lifetime과 downtime 시간에 따라 브로커의 부하는 어떻게 되는가?

5.1 실험 설계

본 논문에서는 OMNeT++에 기반을 둔 오버레이 네트워크 시뮬레이터인 OverSim[6]을 이용하여 다양한 매개변수들을 이용하여 실험을 수행하였다. 브로커와 각각의 피어의 부하가 피어의 숫자, lifetime, downtime의 변화에 따라 어떻게 변화하는가를 측정하기 위하여 실험을 수행하였다. 실험에 사용된 기본 매개변수 값은 다음과 같다. 시스템에는 총 1000개의 피어가 참여하고, 지불 정보를 기록하기 위한 분산 해시 테이블로 Chord[3]를 사용하였다. 코인의 발행은 모두 브로커를 통하여 이루어지며, 발행된 코인은 코인의 유효기간이 지나면 유효기간 갱신을 브로커에게 요청한다. 피어들은 시스템에서 빈번하게 출입을 한다. 피어들의 lifetime은 지수 분포(exponential distribution)를 따르고 있으며 평균 시간은 30분이다. 피어들의 downtime 평균 시간도 30분이며 역시 지수 분포를 따르고 있다. 각각에 피어들이 지불을 수행하는 빈도는 균일 분포(uniform distribution)를 따르며 평균값은 9.26×10^{-3}을 사용하였다. 각각의 원자적 동작(atomic action)들의 비용은 PPay에 기반을 두었으며 PPay에 사용되지 않는 분산 해시 테이블에 대한 비용이 추가되었다.

이 실험에서 다양한 환경에 따라 브로커와 피어들의 부하에 대해 관찰할 수 있었다. 시스템에 참여하는 피어의 수는 100개에서 1000개까지, 피어들의 lifetime의 평균값은 15분에서 30분, downtime은 30분에서 2시간의 평균값이 사용되었다. 표 2는 실험에 사용된 프로토콜 매개변수이며 다음은 실험에 대한 결과이다.

표 2 DPay 프로토콜 매개변수

이름	기본 값	설명
네트워크 크기	1000	네트워크에 있는 노드의 수
Lifetime	30분	피어들의 평균 온라인 시간
Downtime	30분	피어들의 평균 오프라인 시간
DHT	Chord	지불 정보 저장을 위한 분산 해시 테이블
지불 스키마	DPay	소액 지불 시스템 스키마
지불 빈도	9.26×10^{-3}	초당 사용자의 지불 횟수의 기대값
코인 갱신 주기	50,000초	주어진 코인의 갱신 주기
시뮬레이션 시간	100,000초	각각의 시뮬레이션 시간

5.2 실험 결과

DPay의 결과를 분석하기 위하여 PPay를 비교대상으로 선정하였다. 두 개의 시스템의 가장 큰 차이점은 다운타임 프로토콜에 있다. PPay는 코인의 소유자가 오프라인인 경우, 브로커가 대신 지불 메시지를 처리한다. 이것이 PPay의 다운타임 프로토콜이다. 그러나 DPay는 각각의 peer가 지불 메시지를 처리하고 분산 해시 테이블을 조회하여 이전의 지불 정보를 확인한 후 처리한다.

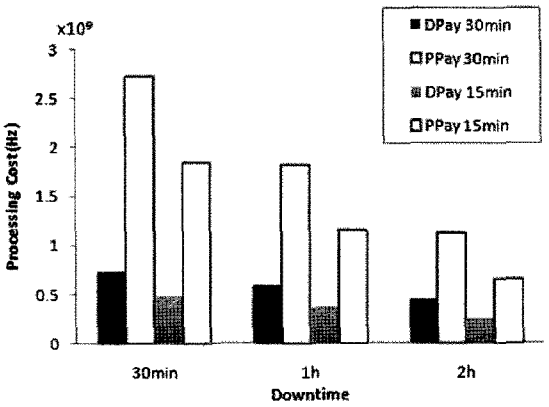

그림 2 다양한 downtime 값에 대한 DPay와 PPay에서의 브로커의 처리 부하 비교

그림 2는 downtime이 변화함에 따라 브로커의 부하의 변화를 보여준다. 여기에서는 lifetime이 30분, downtime이 30분인 경우를 기본 값으로 하여 살펴본다. 피어들의 온라인 시간이 증가함에 따라 지불 횟수도 따라서 증가한다. 이 결과를 통하여 우리는 지불이 일어나는 횟수는 lifetime의 평균값과 downtime의 평균값에 의해 결정되는 것을 확인하였다. PPay는 코인의 소유자가 오프라인인 경우 브로커가 소유자를 대신하여 메시지를 처리하기 때문에 브로커의 부하가 증가한다. 시뮬레이션 결과를 살펴보면 PPay에서의 브로커의 부하가 DPay에서의 브로커의 부하에 비하여 3배 이상 많은 것이 관찰되었고, 이를 통해 DPay가 보다 확장성이 좋은 것을 확인하였다. 앞에서 말한 것과 같이, DPay는 다운타임 프로토콜을 사용하지 않기 때문에 브로커의 부하가 PPay에 비하여 상대적으로 적다. 브로커에 의하여 코인의 발행, 현금화, 유효기간 갱신이 처리되고 피어들에 의해 지불 메시지들이 처리되기 때문이다. 그 결과로 피어들의 부하는 PPay에 비하여 DPay가 좀 더 높은 것을 그림 3을 통해 확인할 수 있다.

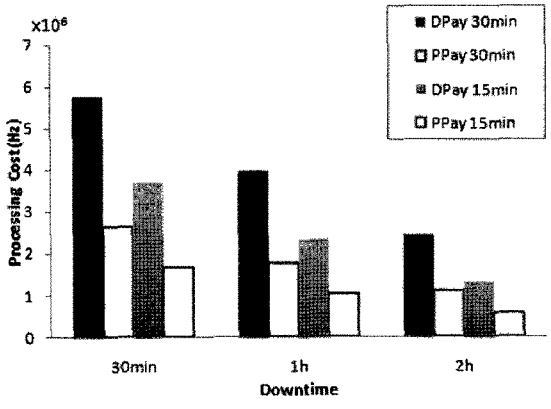

그림 3 다양한 downtime 값에 대한 DPay와 PPay에서의 피어의 처리 부하 비교

그림 4는 시스템에 참여하는 노드의 개수 변화에 따른 브로커의 처리 부하 변화를 보여준다. 총 지불 횟수는 시스템에 참여하고 있는 노드의 숫자와 lifetime의 평균값에 영향을 받는 것을 알 수 있다. 노드의 숫자가 증가하게 되면 DPay와 PPay간의 브로커의 부하의 차이는 점점 증가한다. 하지만 피어의 부하는 일정한 수준으로 유지되는 것을 그림 5에서 확인할 수 있다. 만약 lifetime의 평균값이 고정된다면, 각각의 피어들의 지불 횟수는 변화가 없다. 이러한 경우 오직 lifetime의 평균값에 의해서만 지불 횟수가 변경된다. 그러므로 피어들의 프로세싱 비용은 일정한 수준을 유지한다. 그림 6과 7은 DPay에서 각각의 피어가 지불 정보를 저장하기 위한 공간 비용에 대한 결과이다. 이러한 저장 공간의 비용 역시 전체 지불 횟수의 수에 따라 변화한다. 참여하는 노드가 증가하거나 lifetime의 평균값이 증가, downtime의 평균값이 감소하게 되면 지불 횟수가 증가하게 되어 더 많은 양의 저장 공간 비용이 발생되는 것을 살펴볼 수 있다.

이러한 결과를 통하여 DPay는 PPay에 비하여 브로

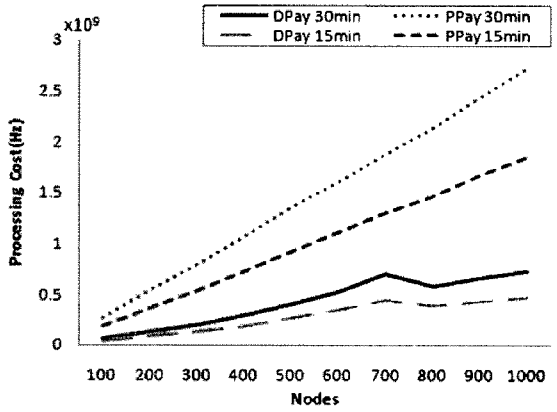

그림 4 시스템에 참여하는 노드의 개수 변화에 대한 DPay와 PPay에서의 브로커의 처리 부하

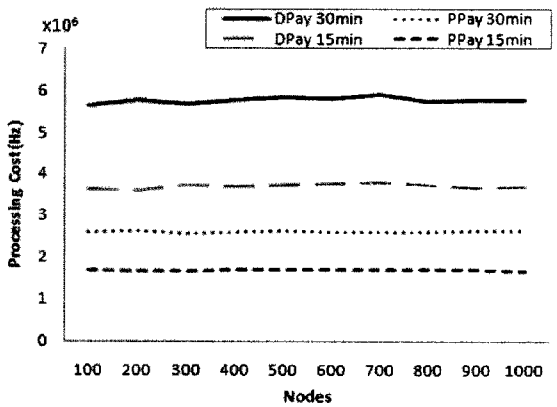

그림 5 시스템에 참여하는 노드의 개수 변화에 대한 DPay와 PPay에서의 피어의 처리 부하

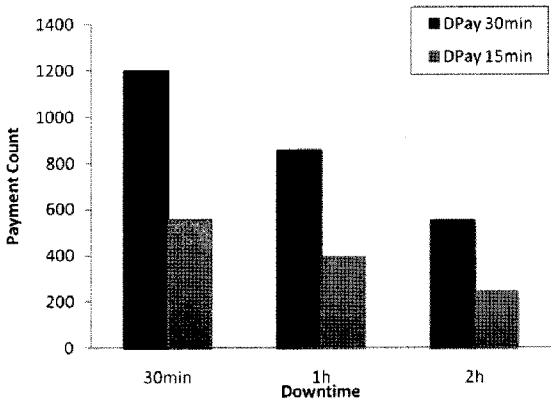

그림 6 다양한 downtime 값에 대한 DPay에서의 저장 공간 비용

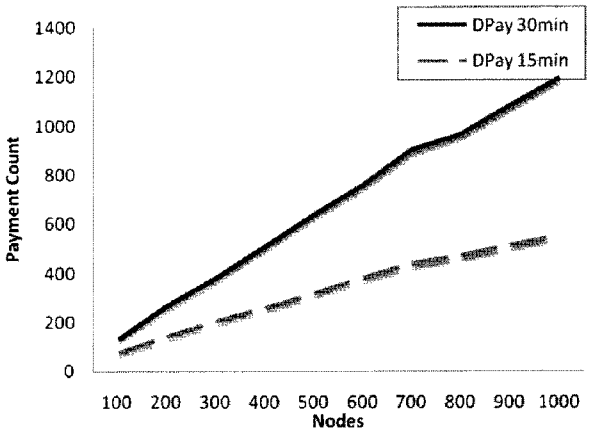

그림 7 시스템에 참여하는 노드의 개수 변화에 대한 DPay에서의 저장 공간 비용

커의 부하는 피어의 가용성에 의한 영향을 받지 않는 것을 확인할 수 있었다. 피어들은 시스템에 있는 다른 피어들의 도움을 통하여 지불 정보를 분산 해시 테이블에 저장하고 있기 때문에 몇몇의 피어들이 오프라인인 경우에도 지불 정보를 조회할 수 있고 브로커의 부하를 증가시키지 않으면서 지불 시스템을 사용할 수 있다. DPay는 다운타임 프로토콜을 사용하지 않기 때문에 브로커의 부하가 줄어드는 것을 살펴볼 수 있다. 대부분의 지불 메시지를 브로커가 처리하는 대신 피어들이 처리한다.

5.3 안전성 분석 및 평가

소액 지불 시스템은 작은 금액의 빈번하게 일어나는 거래를 효율적으로 처리하기 위하여 디자인된 시스템으로 대상 상품의 가격이 높지 않으므로 지불 처리에 사용되는 비용이 높아서는 곤란하다. 즉, 작은 금액에 대한 지불을 처리하기 위한 시스템으로 최대한의 보안이 요구되지 않는 경량화된 메커니즘이 사용된다. 또한 이러한 지불 과정은 오프라인으로 이루어지기 때문에, 코인이 잘못 사용된 경우에는 이후에 코인에 대한 사용 내역을 확인하기 전까지는 발견하기 어려운 구조를 지

니고 있다.

DPay는 기본적으로 PPay에서 사용되고 있는 프로토콜을 따르고 있다. 두 방식에서 관리하는 코인의 지불 정보는 거의 동일하고 가장 큰 차이점은 지불 정보 기록 장소이다. 즉, PPay의 경우 코인의 지불 정보를 기본적으로 코인의 소유자가 관리하는 반면, DPay의 경우 분산 해시 테이블에 저장하고 있다. 지불 정보는 분산 해시 테이블에 서명된 형태로 저장되기 때문에 분산 해시 테이블에 치명적인 보안 오류가 발생하지 않는다면 DPay는 PPay가 제공하는 안정성을 포함하고 있다고 말할 수 있다.

더 자세하게 안정성을 분석해 보면 DPay는 코인을 브로커를 제외한 다른 사용자는 발급할 수 없으며, 이러한 이유로 발급된 코인이 브로커의 키로 서명되지 않은 경우에는 올바른 코인이 아니다. 또한 사용자는 자신이 가지고 있는 코인에 대해서 원하는 때에 코인의 유효성을 검사할 수 있다. DPay에서는 사용자가 코인을 사용할 때 마다 지불 순차번호를 1씩 증가하여 사용하며, 이러한 지불 정보는 분산 해시 테이블에 기록되게 되고, 판매자가 구매자로부터 코인을 받으면, 분산 해시 테이블을 이용하여 이전의 지불 정보에 대해서 조회할 수 있다. 코인을 전달하기 위하여 구매자의 비밀키를 이용하여 코인과 코인의 지불 순차번호를 서명하여 판매자에게 전달하고, 코인을 받은 판매자는 이를 이용하여 전달된 코인정보를 이용하여 해시값을 구해 분산 해시 테이블에서 이전 지불 정보 기록을 조회할 수 있다. 이 정보는 구매자의 비밀키에 의해 서명되어 있으므로, 판매자는 구매자의 공개키를 이용하여 정보를 검사한다. 서명된 내용에 구매자의 식별자와 이전 구매자로부터 받은 코인 정보가 들어있으므로 이를 이용하여 코인의 유효성을 판단할 수 있다. 구매자의 이 정보를 이용하여 판매자는 지불 정보가 올바르게 기록되었는지 확인할 수 있다.

사용자들의 코인 지불 정보는 코인과 지불의 순차번호를 SHA-1과 같은 해시 함수를 이용하여 생성된 키 값에 의해 P2P 시스템 내에 있는 노드 중 임의로 선택된 노드에 기록된다. 이렇게 선택된 노드는 분산 해시 테이블에 코인의 지불 정보를 코인과 코인의 지불 순차번호를 이용하여 해시값을 생성한 후 이를 키로 하여 코인의 지불 정보를 기록하는데, 저장되는 데이터는 현재 코인을 가지고 있는 보유자와 연관성이 없는 임의의 노드에 기록 된다. 만약 판매자가 받은 코인이 구매자에 의하여 코인의 이전 지불 정보를 임의로 변경하여 코인의 정보 또는 코인의 지불 순차 번호를 변경하였다면 해시값 역시 변경되기 때문에 판매자는 구매자로부터 받은 코인 정보를 이용하여 분산 해시 테이블에서 이전 지불 정보를 조회했을 때 정확한 이전 지불 정보를 획득할 수 없다. 또한 구매자가 코인을 사용한 후 동일한 지불 순차번호를 이용하여 또 다른 판매자로부터 서비스를 구매하려 하는 경우에는 위의 4.3 절에서 제시한 실시간 중복 지불 결제 검출 방법에 의하여 잘못된 거래를 검출해 낼 수 있다. 위의 방법을 통한 추후 검증을 사용하여 DPay에서는 잘못된 코인이 사용된 경우 누가 잘못된 코인을 사용하였는지를 찾아낼 수 있다.

마지막으로 본 논문에서 제안된 DPay를 기존의 소액 지불 시스템[5,10,11]과 다음과 같은 항목에 대해 비교하였다.

• 안전성 : 안전성은 사용자간의 거래에서 코인의 중복 결제 또는 위조된 코인의 사용의 검출에 대한 항목으로 PPay는 지불된 코인의 유효성을 확인하기 전까지는 위조된 코인의 발견이 늦은 단점을 지니고 있다. Whopay는 그룹 서명 방식을 사용하여 보다 높은 안전성을 제공하고 있으며, Netpay는 touchstones를 사용한 중복 결제 검출 방법을 제공하고 있다. DPay는 실시간 중복 결제 검출 방법을 제공하여 지불에 사용되는 코인의 유효성을 즉시 확인하여 안전성이 높은 거래 방법을 제공한다.

• 코인 이전성 : 코인 이전성은 사용자가 거래를 통하여 다른 사용자로부터 받은 코인을 서비스를 구매할 때 사용 할 수 있는가에 대한 항목으로 PPay는 계층적 코인을 이용하여 코인이 거래된 정보를 담아 다른 사용자에게 지불할 수 있는 코인 이전성을 제공한다. Whopay는 다른 사용자에게 지불할 때 구매 당 공개키 연산을 통하여 높은 코인 이전성을 보여주며, Netpay는 판매자간에 e-coin을 이용하여 사용자가 사용한 코인의 거래 정보를 교환하고 있다. DPay는 코인의 지불 정보를 분산 해시 테이블에 저장하여 구매자와 판매자가 독립적으로 필요할 때 조회가 가능한 높은 코인 이전성을 지원한다.

• 확장성 : P2P 시스템에서의 소액 결제 시스템은 참여하는 사용자가 증가함에 따라 늘어나는 부하가 브로커에게 할당되지 않고 사용자에게 분산되어 많은 수의 사용자가 참여할 수 있도록 확장성을 제공하는 것이 중요하다. PPay와 Whopay는 온라인 다운타임 프로토콜을 사용하여 코인의 소유자가 오프라인인 경우에 브로커가 코인의 소유자를 대신하여 코인 재할당 메시지를 처리하는 구조를 가지고 있어 사용자가 증가함에 따라 코인의 소유자가 오프라인인 숫자가 증가하게 되어 브로커의 부하가 증가하는 문제가 발생한다. 또한 PPay의 경우 계층적 코인을 사용하고 있기 때문에 지불횟수가 증가함에 따라 코인의 길이가 늘어나는 단점을 지녀 확장성에 영향을 준다. Netpay는 오프라인 지불 방식이며 해시체인 기반의 코인을 사용하여 사용자가 증가함에 따라 늘어나는 부하가 사용자에게 부과되는 구조를 가져 높은 확장성을 지니고 있다. DPay는 코인의 거래

횟수에 관계없이 코인 길이 일정하며 코인의 지불 정보가 분산 해시 테이블에 저장되는 구조를 가져 시스템에 참여하는 사용자가 증가하더라도 브로커에 부하를 주지 않고 사용자들이 부하를 나눠 가지게 되어 높은 확장성을 제공한다.

6. 결 론

Peer-to-peer 시스템은 참여한 사용자들의 리소스를 공유함으로써 전체가 혜택을 받는 시스템이다. 그러나 악의적인 행동을 하거나 free-riding을 하는 사용자에 의한 문제를 겪고 있다. 이러한 문제점들을 해결하는 것은 P2P 시스템에서의 가장 중요한 과제 중 하나로, 소액 지불 시스템은 위의 문제를 해결할 수 있는 방법 중의 하나로써 사용자가 제공받은 서비스에 대한 지불 방법에 보안성, 확장성을 부여하여 해결할 수 있도록 하고 있다. 이 논문은 peer-to-peer 환경을 위한 분산 해시 테이블 기반의 소액 지불 시스템인 DPay를 제안하였다. 확장성을 보다 개선하기 위해서 본 논문은 분산 해시 테이블을 이용하여 참여하는 피어들에게 지불 과정에 발생하는 부하를 분산시켜 코인의 소유자가 오프라인인 경우에도 브로커가 지불 과정에 참여하지 않도록 한다. 실험결과 DPay는 브로커의 부하가 기존 시스템 대비 평균 30%로 줄어든 것으로 나타났으며, 실시간 중복 결제 검출과 보다 안전한 지불 정보 기록을 가능하게 하여 다양한 P2P 시스템에 적용할 수 있을 것으로 기대된다.

참 고 문 헌

[1] BitTorrent Homepage. http://www.bittorrent.com

[2] eMule Homepage. http://www.emule-project.net

[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," *ACM SIGCOMM*, pp.149-160, 2001.

[4] A. Rowstron and P. Druschel, "Pastry: Scalable, distributed object location and routing for large-scale peer-to-peer systems," *IFIP/ACM International Conference on Distributed Systems Platforms (Middleware)*, pp.329-350, November 2001.

[5] B. Yang and H. Garcia-Molina, "Ppay: Micropayments for peer-to-peer systems," ACM Conference on Computer and Communications Security (CCS 2003), October 2003.

[6] I. Baumgart, B. Heep, and S. Krause, "OverSim: A Flexible Overlay Network Simulation Framework," *10th IEEE Global Internet Symposium (GI '07)*, pp.79-84, May 2007.

[7] S. Glassman, M. Manasse, M. Abadi, P. Gauthier, and P. Sobalvarro, "The millicent protocol for inexpensive electronic commerce," *Proc. of WWW4*, 1995.

[8] M. Sirbu and J. D. Tygar, "Netbill: An internet commerce system optimized for network delivered services," *COMPCON '95*, pp.20-25, 1995.

[9] R. Rivest and A. Shamir, "Payword and micromint: Two simple micropayment schemes," *CryptoBytes*, pp.69-87, 1996.

[10] X. Dai and J. Grundy, "Netpay: An off-line, decentralized micro-payment system for thin-client applications," *Electron. Commer. Rec. Appl.*, vol.6, no.1, pp.91-101, 2007.

[11] K. Wei, A. J. Smith, Y.-F. R. Chen, and B. Vo, "Whopay: A scalable and anonymous payment system for peer-to-peer environments," *ICDCS '06*, pp.13-22, 2006.

서 대 일

2008년 아주대학교 정보 및 컴퓨터공학부(학사). 2008년~현재 과학기술연합대학원대학교 석사과정. 관심분야는 P2P, 분산 시스템, 소셜 네트워크

김 수 현

1996년 서울대학교 물리학과 학사. 1998년 서울대학교 전기공학부 석사. 2005년 서울대 전기컴퓨터공학부 박사 졸업. 2005년~2007년 IBM T.J. Watson 연구소 근무. 2007년~현재 한국과학기술연구원 재직중. 관심분야는 가상 머신, P2P, 내장형 시스템 등

송 규 원

2006년 아주대학교 정보 및 컴퓨터공학부(학사). 2007년~현재 과학기술연합대학원대학교 석박사통합과정. 관심분야는 분산 시스템, P2P