유비쿼터스 환경을 위한 분산 해시 테이블 기반의 Peer-to-Peer 소액 지불 시스템

DHT Based Peer-to-Peer Micropayment System for Ubiquitous Environment

서대일 Daeil Seo*, 송규원 Gyuwon Song**, 김수현 Suhyun Kim***

요약 유비쿼터스 환경에서는 다양한 서비스를 구매하기 위한 지불 시스템이 필요하며, 빈번한 거래에 지불 처리 비용을 최소화하는 소액 지불 시스템이 요구되고 있다. 또한 장치가 계속적으로 온라인임을 보장할 수 없기 때문에 그 중 일부가 오프라인인 경우에도 사용할 수 있는 지불 시스템이 요구된다. 이러한 상황을 해결하기 위해 다른 시스템은 중앙의 브로커를 사용하는데, 대부분의 소액 지불 시스템의 문제는 브로커에 부하가 집중된다는 것이다. 본 논문에서는 유비쿼터스 환경을 위한 분산 해시 테이블 기반의 Peer-to-Peer 소액 지불 시스템을 이용해 온라인이 아니어도 지불할 수 있고, 브로커의 부하를 최소화하는 지불 시스템을 제안한다.

Abstract A payment system is required to buy various services and micropayment system is demanded for minimizing a frequent payment transaction cost in ubiquitous environment. It is not guarantee that devices are always online and payment system is required a robustness when some of them is offline. To solve this situation other systems use a centralized broker but the problem with most existing micropayment system is the heavy load on the broker. In this paper present a payment system a DHT based Peer-to-Peer micropayment system for ubiquitous environment can use even if payments need not be online and reduce broker's load.

핵심어: Ubiquitous, Peer-to-Peer, Micropayment, DHT

*주저자 : 과학기술연합대학원대학교 석사과정; 한국과학기술연구원 영상미디어연구센터; e-mail: xdesktop@imrc.kist.re.kr
**공동저자 : 과학기술연합대학원대학교 석사과정; 한국과학기술연구원 영상미디어연구센터; e-mail: sharp81@imrc.kist.re.kr
***교신저자 : 한국과학기술연구원 영상미디어연구센터 선임연구원; e-mail: suhyun.kim@imrc.kist.re.kr

1. 서론

유비쿼터스 컴퓨팅 환경이 도래함에 따라 사용자는 언제 어디서나 주위에 있는 컴퓨팅 자원을 사용할 수 있게 되었다. 이러한 변화에 따라 사용자가 사용할 수 있는 자원들이 다양하게 늘어나고 자신이 필요로 하는 서비스를 찾아 거래를 할 수 있는데, 사용자가 가지고 있는 저장 공간이나 문서들 또는 서비스를 제공하거나 자신이 필요한 서비스를 다른 사용자로부터 구매하는 등의 개인 간의 빈번한 거래가 일어나는 환경에서는 지불 처리비용을 최소화 하여 효율적으로 사용할 수 있는 소액 지불(Micropayment) 방법이 필요하게 된다.

이러한 소액 지불 시스템은 브로커가 코인의 사용에 관여하는데, Peer-to-Peer 시스템을 사용하면 각각의 피어들이 역할을 분담하므로 브로커의 부담을 줄일 수 있다. 분산 해시 테이블(DHT) 기반의 Peer-to-Peer 시스템은 키와 값을 쌍으로 하여 데이터를 저장할 수 있다. 이를 소액 지불 시스템의 코인 지불을 기록하면 브로커의 부담을 줄일 수 있으며 사용자가 오프라인인 경우에도 분산 해시 테이블을 통하여 거래 정보를 확인할 수 있다.

본 논문에서는 사용자간의 거래에서 분산 해시 테이블을 사용하여 사용자간의 거래에 사용되는 메시지의 수를 줄이고 다른 사용자에게 코인의 재발행을 맡겨 브로커의 부담을 줄일 수 있는 사용자 중심의 소액 지불 시스템을 제안한다.

본 논문의 구성은 다음과 같다. 2장에서는 관련 연구에 대해서 살펴보고 3장에서는 본 논문에서 제시하는 지불 알고리즘에 대해 살펴본다. 4장에서는 결론을 제시한다.

2. 관련 연구

2.1 소액 지불 시스템

소액 지불 시스템은 작은 금액의 지불을 처리하기 위한 시스템으로 처리비용을 최소화하여 지불 처리에 사용되는 비용이 상품의 가격보다 높지 않게 하는 시스템으로 지불 금액은 10$ 이하의 금액이며, 뉴스, 논문, 음악 등의 디지털 콘텐츠를 구매하거나 Peer-to-Peer 시스템에서 서비스의 구매에 사용된다. 이러한 소액 지불 시스템에서는 지불 처리에 사용되는 비용이 가장 중요한 논점이다. PPay[1]에서는 이러한 소액 지불 시스템의 특징을 다음과 같이 정의하고 있다. 소액 지불은 작은 금액에 대한 지불이므로 1)최대한의 보안이 요구되지 않으며, 2)지불 메커니즘은 반드시 경량이어야 하고 지불하고자 하는 가치를 넘어서면 안 된다고 이야기 하고 있다. 소액 지불 시스템에 관련된 연구로는 PPay[1], PeerMint[4], WhoPay[3] 등이 진행되었다.

PPay[1]는 대표적인 소액 지불 시스템으로 사용자가 브로커를 통하여 코인을 발급받아 이를 이용하여 다른 사용자와의 거래에 코인을 사용하는 시스템이다. 브로커로부터 처음 코인을 생성한 사용자는 코인의 소유자(owner)가 되며 거래에 의하여 현재 코인을 가지고 있는 사용자는 코인의 보유자(holder)가 된다. 즉, 코인의 소유자는 변하지 않지만 거래에 따라 보유자가 변경된다. 사용자 U가 첫 거래로 코인을 사용자 V에게 지불하게 되면 사용자 V가 코인의 보유자로 변경되고, 사용자 V가 사용자 U로부터 받은 코인을 사용하기 위해 사용자 U에게 코인의 보유자 변경 요청을 하고, 사용자 U가 보유자가 변경되었다는 내용을 알려주는 구조를 가지고 있다.

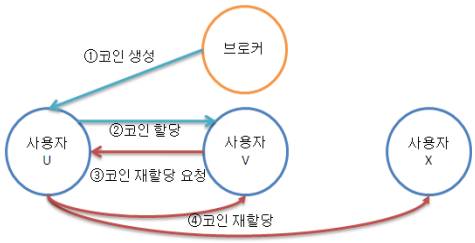

그림 1. PPay 기본 프로토콜

Peer-to-Peer 시스템에서는 사용자가 빈번하게 시스템에 들어오고 나가고를 반복하며 네트워크의 문제로 일시적으로 통신장애가 발생하여 기본 프로토콜을 적용할 수 없는 경우가 발생하게 되는데 이를 위해서는 다운타임 프로토콜이 사용된다. PPay도 이러한 경우를 위하여 코인의 소유자가 온라인이 아닌 경우 브로커에게 코인의 재할당을 요청하고 코인의 소유자가 다시 온라인이 될 때 브로커가 코인의 소유자가 오프라인인 동안 발생한 지불 정보를 전달하는 방법을 사용하고 있다.

2.2 분산 해시 테이블

분산 해시 테이블(DHT)을 이용한 Peer-to-Peer 시스템은 비구조적인 Peer-to-Peer 시스템이 브로드캐스팅을 통해 피어들의 데이터를 검색하여 효율성이 떨어지는 문제를 개선하기 위하여 등장하였다. 검색의 효율성을 높이기 위해 분산 해시 테이블을 이용하여 각각의 피어가 라우팅 테이블을 가지게 되고 이를 통하여 정형화 된 검색방법을 제시한다. 이를 통해 분산 해시 테이블 기반의 Peer-to-Peer 시스템은 O(logN)의 라우팅 알고리즘을 제공하고 있다.

DHT 기반 Peer-to-Peer 시스템은 SHA-1과 같은 해시 함수를 사용하여 각각의 노드와 키에 m-bit의 식별자를 할당한다. 할당된 식별자는 해시 성질에 의해 오버레이 네트워크에 pseudo random한 위치에 분포하게 되며 노드들은 노드의 식별자를 이용하여 자신이 관리해야 할 오브젝트의 키를 할당받게 되며 이에 따라 노드들 간에 로드 밸런싱이 이루어진다. DHT는 〈k, v〉 의 쌍으로 이루어진 key, value를

이용하여 오브젝트에 대한 정보를 저장한다. 이와 관련된 연구로는 Pastry[2], Chord[5], CAN[6], Tapestry[7] 등이 있다.

3. 지불 알고리즘

이 논문에서 제안하는 지불 알고리즘은 분산 해시 테이블을 사용하여 기본 프로토콜만을 이용하여 사용자간의 소액 지불을 할 수 있도록 하는 프로토콜이다. 즉, 일부의 사용자가 오프라인인 경우를 위한 다운다임 프로토콜이 존재하지 않는다. 제안된 알고리즘은 PPay와 유사하게 유효기간 내의 코인인 경우 코인을 현금으로 바꾸지 않고 재사용을 하고 있으며, DHT를 사용하여 고유한 NodeId를 부여하고 있다. 거래 후에는 지불 정보를 DHT에 기록하여 지불 정보를 검색한다.

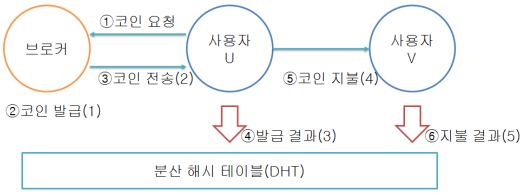

그림 2. 코인의 생성 및 지불

지불 정보를 일부 사용자가 오프라인인 경우에도 계속 유지하기 위해서는 PAST[8]와 같이 DHT에 들어있는 지불 정보의 사본을 자동적으로 k개 이상으로 유지시켜주는 DHT 응용 계층이 필요하다. 이를 통하여 사용자간에 거래를 통해 발생하는 지불 정보들이 임의의 피어에 저장되게 되며 사본을 k개 이상으로 유지되도록 함으로써 일부 피어가 온라인이 아닌 경우에도 지불 정보를 확인할 수 있다.

표 1. 프로토콜에 사용되는 기호

기호	설명
B, U, V, W	브로커(B) 및 사용자(U,V,W) 의 Node ID
SKx, PKx	X의 비밀키와 공개키
$\{msg\}_{Kx}$	X의 키 K를 사용하여 서명된 메시지
C	거래에 사용되는 코인
E	코인의 유효기간
⟨k, v⟩	분산 해시 테이블에 저장되는 키(key)와 값(value)

사용자 U가 거래를 하기 위해서는 브로커에게 코인을 요청하는데, 요청을 받은 브로커는 코인을 발급하기 위해 이전에 사용한 코인의 순차번호보다 큰 값을 이용하여 새로운 코인의 순차번호(sn)를 결정하고, 코인의 유효기간(E)을 결정한 뒤 브로커의 비밀 키를 이용해 서명하여 코인을 발행한다. 이렇게 발행된 코인은 유효기간이 만료되기 전까지 사용 가능하다.

$$C = \{U,\ sn,\ E\}_{SK_B} \quad (1)$$

발급된 코인을 거래에 사용하여 다른 사용자에게 전송하기 위해 코인의 거래에 사용된 순차번호(seq)를 할당한다. 이 순차번호는 브로커가 발행할 때에 0으로 지정되며 이후 거래에서 1씩 증가하며 이를 이용해 코인이 거래에 사용된 순서를 추적할 수 있다.

$$T_B = \{C,\ seq_0\}_{SK_B} \quad (2)$$

브로커로부터 T_B를 받은 사용자는 코인 정보와 브로커의 NodeId를 자신의 비밀 키로 서명한 후 코인과 거래의 순차번호를 키 값(C+seq)으로 하여 코인 정보를 DHT에 삽입한다. 이렇게 저장된 정보는 코인을 사용한 다음 거래 시 이전 거래정보를 검증하기 위해 사용된다.

$$\langle C+seq_0,\ T_{BU} = \{T_B,\ B\}_{SK_U}\rangle \quad (3)$$

사용자 U는 발급된 코인을 사용하여 다른 사용자 V에게 지불하기 위해서 거래에 사용된 순차번호를 1만큼 증가하여 사용할 코인과 함께 자신의 비밀 키로 서명을 한 후 상대방에게 지불한다.

$$T_U = \{C,\ seq_{i+1}\}_{SK_U} \quad (4)$$

지불을 받은 사용자 V는 사용자 U의 공개키를 이용하여 코인 정보를 검증하고 받은 코인의 거래에 사용된 순차번호를 이용하여 DHT에 받은 코인의 유효성을 검사하게 된다. 코인과 거래에 사용된 순차번호를 키 값으로 하여 DHT에 조회를 하여 값이 발견되지 않은 경우 정상적으로 지불된 것이며, 값이 발견된 경우에는 이미 사용된 코인이 재사용된 것을 확인할 수 있다. 정상적으로 거래가 진행된 경우 이전 사용자로부터 받은 코인 정보와 이전 사용자의 NodeId를 자신의 비밀 키로 서명한 후 코인과 거래에 사용된 순차번호를 키 값으로 하여 코인 정보를 DHT에 삽입한다.

$$\langle C+seq_{i+1},\ T_{UV} = \{T_U,\ U\}_{SK_V}\rangle \quad (5)$$

이렇게 발행된 코인은 유효기간 내에서는 계속적으로 다

른 거래에 사용할 수 있으며 유효기간이 만료되면 코인의 재발급을 시작한다. 지불 시스템에서 사용되는 코인은 발행자에 관계없이 동일한 가치를 가지고 있기 때문에 동일한 가치를 가지는 코인으로 재발급이 가능하다.

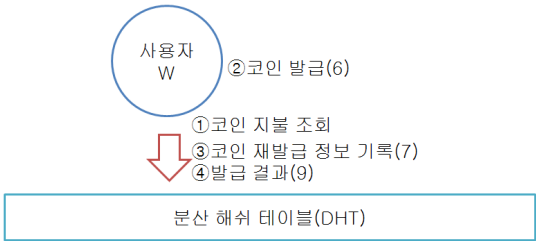

그림 3. 코인의 재발급

코인을 재발급하기 위해서는 사용자 W는 DHT에서 거래에 사용된 순차번호가 0이 될 때까지 1씩 줄여나가면서 코인 정보를 조회하여 지불 내역을 확인한다. 코인의 지불 내역이 이상이 없는 경우 새로운 유효기간 생성하여 자신의 비밀 키로 새로운 코인을 생성한다.

$$C' = \{U,\ sn,\ E'\}_{SKw} \quad (6)$$

새로 발급한 코인의 정보를 DHT에 기록하는데 이 정보는 사용자가 브로커에게 코인을 상환을 요청할 때 이전 코인의 변경 정보를 확인하는데 이용된다.

$$\langle C,\ \{C,\ C'\}_{SKw}\rangle \quad (7)$$

재발급 내용을 기록한 뒤 새로 발급된 코인과 거래에 사용된 순차번호를 0으로 할당한 후 자신의 비밀 키로 서명하여 코인을 초기화한다.

$$T_W = \{C',\ seq_0\}_{SKw} \quad (8)$$

재발급 된 코인을 사용하기 위해 코인 정보와 사용자의 NodeId를 자신의 비밀 키로 서명한 후 코인과 거래에 사용된 순차번호를 키 값(C'+seq)으로 하여 코인 정보를 DHT에 삽입한다.

$$\langle C'+seq_0,\ \{T_W,\ W\}_{SKw}\rangle \quad (9)$$

이를 통해 재발행 된 코인은 브로커가 발행한 코인과 동일하게 사용자들 간에 사용된다.

4. 결론

시간과 장소에 구애받지 않고 언제든 네트워크에 접속할 수 있는 유비쿼터스 환경이 다가옴에 따라 제안된 소액 지불 시스템과 같은 처리 비용을 최소화 하는 지불 시스템의 필요성이 점점 확대대고 있다. 본 논문에서는 사용자 간 거래에서 구조화 된 Peer-to-Peer 시스템인 DHT를 사용하여 코인의 지불 정보를 저장하는 소액 지불 시스템을 제안하였다. 이를 통해 브로커가 코인의 거래에 참여하는 빈도를 줄여 브로커의 부하를 적게 하고, 사용자가 오프라인인 경우에도 코인의 거래에 영향을 주지 않는 알고리즘을 제안하였다.

참고문헌

[1] B. Yang and H. Garcia-Molina. "Ppay: Micropayments for peer-to-peer systems," In 10th ACM Conference on Computer and Communications Security (CCS), 2003.

[2] Peter Druschel and Anthony Rowstron. "Pastry: Scalable, distributed object location and routing for large-scale peer-to-peer systems," ACM SIGCOMM, 2001.

[3] K. Wei, A. J. Smith, Y.-F. R. Chen, and B. Vo. "WhoPay: A Scalable and Anonymous Payment System for Peer-to-Peer Environments," In ICDCS, 2006.

[4] D. Hausheer and B. Stiller, "PeerMint: Decentralized and Secure Accounting for Peer-to-Peer Applications," 2005 IFIP Networking Conference, May 2005.

[5] I. Stoica, R. Morris, D. Karger, M. F. Kaashoek, and H. Balakrishnan. "Chord: A scalable peer-to-peer lookup service for Internet applications," In Proc. ACM SIGCOMM, Aug. 2001.

[6] S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Shenker. "A scalable content-addressable network," In Proc. ACM SIGCOMM, Aug. 2001.

[7] B. Y. Zhao, L. Huang, J. Stribling, S. C. Rhea, A. D. Joseph, and J. D. Kubiatowicz. "Tapestry: A resilient global-scale overlay for service deployment," IEEE JSAC, 22(1):41-53, Jan. 2004.

[8] P. Druschel and A. Rowstron. "PAST: A large-scale,persistent peer-to-peer storage utility," In Proc. HotOS VIII, Schloss Elmau, Germany, May 2001.