

모바일 환경에서 이동성을 고려한 Chord 시스템 알고리즘

홍 록 지* · 문 일 영**

Chord System Algorithm Based on the mobility in the mobile environment

Hong, Rok Ji · Moon, Il Young

〈Abstract〉

Due to the increasing use of smart phones, using Wifi, Wibro of mobile devices spread. Accordingly, Need of using peer-to-peer file sharing between mobile devices is growing. However, common Peer-to-Peer(P2P) system is too complex and is not suitable to be applied to mobile devices. Thus, the need of research has shown to improve way in a mobile environment that has a constraints as mobility and the scope of communication.

Among them, there is a Chord as contents look up algorithm. Chord is the issue. Chord as the DHT-based P2P protocol shares files, index-key and distributed key across the network. However, exist Chord doesn't consider the mobility. So, It has a lot of problems in the mobile environment.

Thus, in this paper, I will try to explore an appropriate way of Chord in a mobile environment.

Key Words : Chord, Mobile, Mobility

I. 서론

미래의 인터넷 기반 애플리케이션에는 확장성, 보안 및 신뢰성, 새로운 서비스에 대한 유연성 및 QoS 등의 요구사항이 제시된다. 그러나 기존 클라이언트-서버 방식에서는 비용 문제와 복잡성 문제가 야기된다. 이에 모든 형태의 분산 자원 접근이 가능한 Peer-to-Peer(P2P) 통신 방식으로 보다 간단한 해결방안을 제안함으로써 새로운 가능성을 열고 있다.

이러한 P2P방식은 최근 스마트폰의 사용 증가로 인

해 Wifi, Wibro의 이용이 확산되면서 모바일환경에서도 많은 각광을 받고 있다. 이는 사용자들이 Social Network Service (SNS)와 다양한 모바일 콘텐츠를 이용하게 되면서 모바일 기기 간의 파일 공유 필요성이 증가하고 있기 때문이다[1, 2]. 그러나 일반적인 P2P 컴퓨팅 네트워크를 모바일 기기에 적용하기에는 너무 복잡하다. 모바일 기기는 이동성을 가지는 객체이고, 각각의 기기는 통신범위라는 제약조건을 가지고 있기 때문이다. 그리하여 기존 유선환경의 P2P방식에서 모바일 환경에 적합한 방식으로 개선해야 하는 필요성이 대두되고 있다. 현재까지 파일 및 서비스 공유를 위한 기술 개발은 이루어져있으나 정식적인 모바일 P2P는 개발되지 않은

* 한국기술교육대학교 대학원 컴퓨터공학부

** 한국기술교육대학교 컴퓨터공학부(교신저자)

상태이다. 또한 현재의 연구는 이동성이 많은 모바일 환경에서의 문제점이 나타나고 있다. 그리하여 모바일 환경에 적합한 방식으로 개선하는 연구의 필요성을 인지하고 본 논문에서는 DHT의 Contents Look up 알고리즘 중 하나인 Chord를 모바일 환경에서 적합하도록 하는 방식을 모색해보려 한다.

2장에서 기존 비구조적(Unstructured) P2P 방식의 단점을 보완하기 위한 방식인 DHT(Distributed Hash Table)와 DHT 알고리즘 중 많은 곳에서 연구가 진행 중인 Chord 알고리즘에 대한 이론을 알아본다. 3장에서 모바일 환경에서 Chord의 문제점을 파악하고 본 논문에서 제안한 Chord 시스템 알고리즘을 소개한다. 4장에서는 본 논문의 결론을 맺는다.

II. 관련연구

오버레이 네트워크(Overlay network)란 기존의 네트워크 위에 논리적인 네트워크를 구성하여 이루어진 가상 네트워크를 말한다. 물리적인 네트워크를 사용하되 그 구성을 고려하지는 않는다. 이러한 오버레이 네트워크 안의 노드들은 가상 링크나 논리 링크로 연결되며 기존의 네트워크를 활용하여 효율적인 네트워크 서비스를 제공할 수 있게 된다. P2P 네트워크는 오버레이 네트워크의 대표적인 예라 할 수 있다.

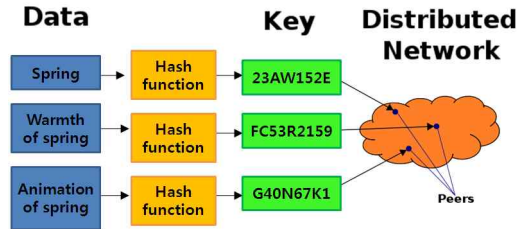
2.1 DHT (Distributed Hash Table)

구조적 P2P 오버레이 네트워크 구축 기술은 비구조적 방식에 비해 비교적 최근에 제안된 P2P 프레임워크 구축 기술이다.

구조적 P2P는 주로 DHT를 기반으로 구축되는 추세이다. DHT를 사용하여 만들어진 P2P 네트워크는 하나의 오버레이 네트워크를 만들고 데이터 검색은 오버레이 네트워크 위에서 이루어지게 된다. 이 DHT 기반 오버레이

네트워크는 검색의 효율성을 높이고자 P2P 오버레이 네트워크에 구조적인 특성을 부여한 것이다. 대표적인 프로토콜로 Chord[3], Pastry[4], Tapestry[5], CAN[6] 등이 있다.

DHT 기반 오버레이 네트워크는 초창기에는 분산 스토리지, 파일 공유, 웹 캐싱 등에 응용되었으며, 최근 들어서는 P2P 기반의 멀티캐스트 서비스를 제공하기 위한 Tree 생성 및 관리 기법으로도 많이 응용되고 있다.



<그림 1> DHT의 개념도

그림 1은 분산해시 테이블의 개념을 쉽게 보여주고 있다. 이름에서 알 수 있듯이 시스템 내의 각 노드(node)들이 키 셋을 나누어 가지고 있는 분산 시스템을 말한다. DHT 시스템 외부에서 보면 일반 해쉬 테이블과 다를 게 없지만 일반적인 해쉬 테이블과 다른 특성을 가지고 있다. DHT 시스템 전체를 중앙에서 관리하는 조직이 없다. 그리하여 DHT를 이루고 있는 노드 수에 영향을 받지 않고 확장할 수 있다. 또한 DHT 시스템 내부에 노드가 추가되거나 없어지거나 오동작을 하더라도 DHT 시스템 전체의 기능에 영향을 끼치지 않는다.

기존 P2P 솔루션은 Napster의 중앙집중관리 방식[7]과 Gnutella와 같은, 주변 노드들을 활용하는 완전히 분산된 네트워크를 구성하는 방식[8]이 있다. P2P 네트워크를 중앙에서 컨트롤할 경우에 네트워크 이용 효율은 좋아지겠지만 컨트롤이 중앙에 모여 있기 때문에 중앙 관리 시스템이 취약점이 될 수 있다. 그에 반하여 피어(Peer)에 의지하는 방식은 중앙 집중 조직이 없어서 뚜렷한 취약점은 없지만 네트워크를 비효율적으로 사용한다

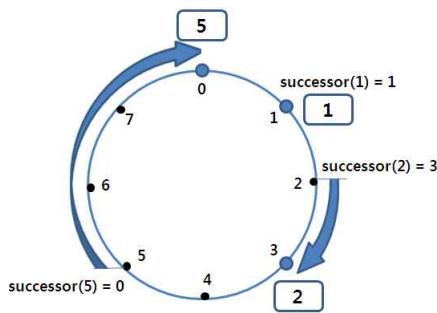
는 문제가 있다. DHT는 structured key based routing (이하 SKBR)이라는 방법으로 위와 같은 P2P시스템의 약점을 극복 했다.

DHT는 크게 Keyspace 파티션과 오버레이 네트워크의 두 부분으로 나뉘는데 해쉬의 키 집합들을 DHT시스템 내부의 각 노드에 분산시키고 DHT의 엔트리 포인트에 상관없이 키의 위치를 찾아갈 수 있도록 라우팅하는 알고리즘을 이용해서 DHT 시스템 내부의 네트워크를 떠돌며 목적지를 찾아가게 된다. 이게 바로 SKBR인데 쉽게 얘기하자면 해쉬 테이블 내부의 버킷(bucket)을 그룹으로 묶고 그룹 사이를 링크로 연결하여 자기가 원하는 정보가 있는 그룹을 찾아갈 수 있도록 해주는 것이다. 외부에서 보면 해쉬처럼 보이지만 내부에서 원하는 버킷을 찾아가는 방식은 기존의 해쉬와 다르다.

2.2 Chord

Chord는 여러 프로토콜 중 쉽고 간단한 방법으로 여러 곳에서 가장 많이 쓰이고 연구되고 있다. Chord는 하나의 circle을 가정하고 이 circle위에 $0 \sim 2^m - 1$ 만큼의 키(Key)/노드 식별자들이 가능하다는 것으로 시작한다.

우선 8개의 식별자를 가지는 Chord를 예로 들어 설명한다. 8개의 식별자를 가지다는 말은 $m = 3$ 이라는 말이며, 결국 modulo 2^m 으로 모든 키와 노드 id를 표현할 수 있다는 뜻이다.

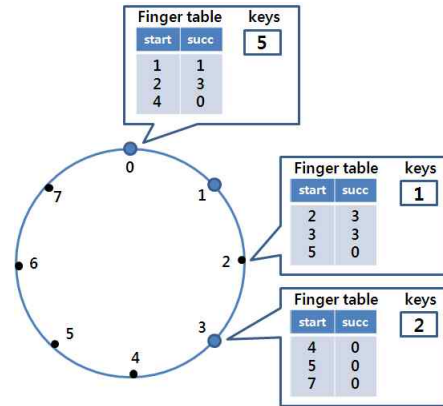


<그림 2> Chord의 기본적인 구조

그림 2에서 볼 수 있듯이 3개의 노드가 각각 0, 1, 3 위에 배치되어 있는 것을 확인할 수 있다. 여기에서 중요한 것은 각 노드가 과연 어떠한 키를 담당할 것이냐를 결정하는 문제다.

Chord는 Successor라는 개념을 사용하여 하나의 노드가 담당할 키의 range를 결정한다. 위에서 키 1은 노드 1이 담당하고, 키 2는 노드 3, 그리고 키 6는 노드 0이 담당한다는 것을 쉽게 알 수 있다.

Chord는 담당해야할 키 range에 대한 효율적인 관리를 위해서 또한 인접한 노드간의 라우팅을 위해서 다음과 같은 Finger Table을 각 노드에서 관리하고 있다.



<그림 3> Chord의 Finger Table

Finger Table은 주소공간의 크기에 따라 테이블의 전체 크기가 변하게 된다.

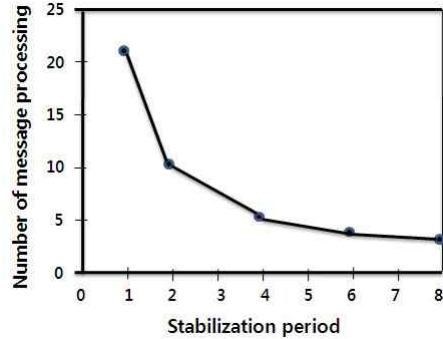
우리가 살펴본 Chord의 Finger Table 방식은 arbitrary 키에 대한 정보를 포함하고 있지 않다. 그리하여 Chord 알고리즘은 임의의 키를 담당하고 있는 노드를 찾는 과정을 거치게 된다. 이 과정은 $O(\log N)$ 효율로 키를 찾을 수 있게 된다.

그림 3과 같이 Chord에서는 데이터 요청 메시지를 보낼 때 Finger Table을 보고 메시지를 보내게 된다. 하나의 노드가 메시지를 보내면 가장 먼저 요청 메시지를 해쉬하여 키 값을 얻어 온다. 받은 키 값과 자신의 Finger

Table을 비교한다. 자신의 Finger Table에서 그 키 값보다 작은 값 중에 최댓값인 Successor로 메시지를 보내도록 한다. 받은 노드는 앞의 과정을 반복하여 메시지를 지정된 장소에 보낸다.

Chord는 노드의 실패에 대해서 파급효과를 최소화하기 위해 Stabilization 함수를 주기적으로 호출한다. Stabilization 함수는 자신의 Successor가 살아있는지 확인하기 위해 ping메시지를 주기적으로 보낸다. ping메시지에 대한 응답이 오지 않으면 그 Successor를 자신의 Finger Table에서 지운다. 그리고 Stabilization 함수는 Find_successor 메시지를 보내 자신의 Successor를 주기적으로 찾고 등록한다. Find_successor 메시지는 Ping 메시지 보다 많은 트래픽을 발생시킨다.

이는 Successor를 찾기 위해서 자신은 하나의 메시지만 발생시키지만 다음 노드로 가게 되면 전체 네트워크에서 여러 메시지가 발생되기 때문이다.



<그림 4> Stabilization 주기에 따른 메시지 수

다. 그림을 보면 주기가 느리면 메시지 양이 적고 빨라지게 되면 메시지의 양이 급격히 증가하는 것을 알 수 있다.

이러한 과정으로 발생하는 메시지의 양은 각각의 노드와 전체 네트워크의 트래픽을 발생시키는 문제점을 발생시키게 된다.

III. 모바일 환경에서의 Chord와 기존의 해결방안 및 제안하는 시스템

3.1 모바일 환경에서 Chord의 문제점

기존의 Chord는 모바일 환경과 같은 Join/Leave가 자주 일어나는 환경에서 요청 메시지의 성공률이 떨어진 다.

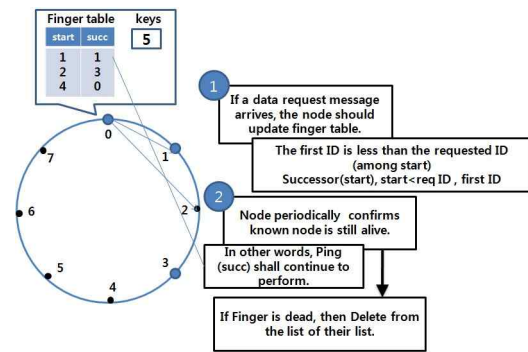
이러한 환경에서 Chord는 데이터 요청 메시지의 실패가 일어나지 않기 위해 Finger Table을 최신으로 유지해야 한다.

그러기 위해서는 Stabilization 함수의 주기를 빠르게 해야 하지만 주기가 빨라지게 되면 하나의 노드가 처리해야 하는 메시지의 양은 지수적으로 증가하게 된다.

그림 4는 Stabilization 주기에 따른 메시지 양의 변화를 보여주는 그래프이다. x축은 주기를 나타내고 y축은 하나의 노드가 초당 처리하는 메시지의 개수를 나타낸

3.2 Reactive Chord

Reactive Chord에서는 stabilization에서 발생하는 Find_successor 메시지를 통해 주기적으로 업데이트 하는 방식에서 데이터 요청 시에 업데이트 하는 방식을 제안하였다[9]. 이 때, ping 메시지는 그대로 사용하였다.



<그림 5> Reactive Chord 구조에서의 Finger Table 요청

모든 요청에 대해서 Find_successor 메시지를 보내게 되면 요청이 많을 경우 문제가 생길 수 있기 때문에 이전에 요청했던 것과 같은 요청이 올 경우에는 n개까지 버퍼에 저장한다. 메시지의 개수가 n개를 초과하면 다시 Find_successor 메시지를 보낸다. 이러한 방법으로 Find_successor 메시지의 수를 줄이게 된다.

이 방법은 네트워크 트래픽을 해결하였지만 요청 메시지가 많아지면 오히려 기존의 Chord보다 트래픽이 커지는 경우도 생기게 된다.

또한 모바일 환경의 빈번한 이동성으로 인한 메시지 트래픽에 대한 해결방안은 제시하였으나 이동성이라는 요소를 유동적으로 고려하여 해결하지 못한다는 한계를 가지고 있다.

3.3 모바일 환경에서 이동성을 고려한 기법

클러스터링 기법에서 모바일의 이동성을 고려한 경우를 살펴보고자 한다.

먼저, LIDAR(Lowest-ID with Adaptive ID Reassignment)기법은 초기 클러스터 헤드를 선정한 후 노드들의 이동성을 고려하여 클러스터 헤드를 재선정하게 된다. 이 때 선정된 클러스터 헤드는 각 클러스터 멤버 노드의 이동성을 산술 평균화하여 이동성을 판단하고 Chord에서의 ping 메시지와 같은 역할을 하는 Hello 메시지 주기를 조절하게 된다.

그러나 이동성을 산술 평균화하여 클러스터의 이동성을 판단하는 것은 잘못된 클러스터 이동성 정보를 제공할 수 있다. 해당 클러스터 멤버가 클러스터 헤드의 통신 영역 외곽에 존재할 경우 클러스터 헤드가 관리하는 통신 영역 밖에 노드의 이동성이 반영되게 되는 것이다. 따라서 클러스터의 이동성 측정에 있어 외부 노드의 영향을 최소화하여 클러스터의 이동성을 판단할 수 있는 방법이 필요하며, 더불어 측정된 이동성 정보를 토대로 적응적으로 Hello 메시지 전송 주기를 조절할 수 있는 방법이 필요하다.

이것을 해결하기 위한 기법으로 모바일 애드 혹 네트워크에서 이동성을 고려한 안정적인 클러스터링 기법이 있다[10]. 이 기법에서 노드의 이동성의 측정은 주변 노드의 통신 영역 가입 노드와 탈퇴 노드의 수로 산출된다. 즉, 가입 노드는 임의의 이동 노드 자신의 전송범위 안으로 새로운 노드가 들어옴으로써 자신의 주변 노드로 판단되는 노드 수를 의미하고, 탈퇴 노드의 경우 기존에 자신의 주변 노드였으나 통신영역 밖으로 이동함으로써 자신의 링크 상태가 끊기는 노드를 말한다. 산출 결과 가장 낮은 이동성을 갖는 노드가 클러스터 헤드로 선정되게 되는 것이다.

3.4 제안하는 Chord 시스템 알고리즘

기존의 Chord 시스템에는 이동성으로 인한 메시지 트래픽의 연구는 이루어져 왔으나 이동성이라는 요소에 대한 유동적인 해결방안이 명확하지 않았다. 그리하여 본 논문에서는 많은 트래픽을 발생시키는 요청메시지 수를 측정하고 이동성을 고려한 클러스터링 기법을 활용하여 모바일 환경에서 적합한 Chord 시스템 알고리즘을 제안하여 한다.

1) 노드의 이동성 측정

각각의 노드는 자신의 통신 범위 안에 속한 주변 노드들과의 거리를 측정한 후 일정 시간 동안의 거리의 변화에 따라 노드의 이동성을 측정하게 된다. 또한 노드가 Leave/Join되는 경우의 노드의 거리변화는 측정대상에서 제외하게 된다.

2) Stabilization 주기 재설정

측정된 이동성에 따라 각각의 노드는 자신의 Stabilization 주기를 재설정하게 된다. 이동성이 적은 경우에는 Stabilization 주기를 느리게 하여 Chord에서 처

리하는 불필요한 메시지 개수를 줄이고 이동성이 큰 경우에는 Stabilization 주기를 빠르게 하여 Finger Table을 최신으로 유지하게 된다.

3) 요청 메시지 수 측정

데이터 요청 메시지 수에 따라 Stabilization 주기를 변경한다. 평균적인 데이터 요청 메시지 수가 많은 경우 Stabilization 주기를 빠르게 하고 적은 경우 Stabilization 주기를 느리게 설정한다.

이러한 방법으로 데이터 요청 메시지가 매우 많은 경우 서로 다른 요청 메시지에 대해 Find_successor 메시지를 보내는 것에 비해 Stabilization 주기 빠르게 재설정하는 방법을 통하여 네트워크 트래픽을 줄일 수 있게 된다.

이러한 과정으로 기존의 Chord에서 일괄적으로 적용되었던 Stabilization 주기를 이동성 측정을 통하여 유기적으로 변화시키게 된다. 기존의 Chord에서는 Chord 내의 전체 노드의 Stabilization 주기를 높임으로써 이동성이 적은 노드에서도 불필요하게 발생한 메시지 양을 줄이고 이동성이 높은 노드는 Stabilization 주기를 빠르게 하여 데이터 요청 메시지의 실패가 일어나지 않게 할 수 있다. 그에 따라 전체 네트워크의 신뢰도를 높이고 트래픽을 줄일 수 있게 된다.

또한 데이터 요청 메시지의 수에 따라 Stabilization 주기를 변경함으로써 요청 메시지가 많은 경우에도 네트워크가 유기적으로 적응할 수 있게 될 것이다.

3.5 향 후 연구방향

본 시스템은 모바일 환경과 같은 이동이 빈번한 환경에서의 전체 네트워크 트래픽은 개선하였지만 요청 메시지가 많은 노드에서의 메시지 양을 줄이는 데에는 아직 한계점을 가지고 있다. 또한 본 논문에서 제시한 이동성을 측정하는 방법과 더불어 가장 효율적이고 정확한 측도를 제시할 수 있는 연구가 필요할 것이라 본다. 더하여

Reactive Chord와 같이 데이터 요청 시에 업데이트 하는 시스템과 보완하여 각각의 시나리오에 맞는 시스템의 연구가 더욱 필요할 것이다.

IV. 결론

현재 스마트폰 가입자가 늘어나면서 이제 스마트폰은 일상 속으로 깊숙이 자리하고 있다. 와이파이(Wi-Fi) 무선인터넷과 3G 이동통신망을 모두 사용가능하다. 이러한 멀티인터페이스가 가능한 모바일이 증가하면서 모바일 시장 또한 확대되고 있으며 다양한 서비스들이 등장하고 있다.

본 논문에서는 이러한 시대를 맞이하면서 모바일 P2P의 사용도 증가될 것으로 예상하여 기존에 사용되고 있는 P2P 알고리즘 중 가장 쉽고 간단한 방법이며 많은 분야에서 사용되고 있는 Chord에 대하여 알아보고 변화가 많은 모바일 환경에서는 부적절한 점을 파악하였다. 그리고 그것을 개선한 Reactive Chord에 대해 알아보고 문제점을 파악하였다.

또한 다양한 상황에 맞는 Chord 알고리즘 적용과 연구의 필요성을 시사했다.

우리는 이동성이 있고 변화가 많은 모바일환경에서 제공하는 서비스의 성격과 상황에 맞도록 Chord와 Reactive Chord를 효과적으로 사용하도록 해야 할 것이다. 또한 앞으로 Chord와 Reactive Chord 알고리즘의 단점을 개선할 수 있는 방법을 고안해야 할 것이다.

참고문헌

- [1] 김대업, 이일금, 오재신, "이용수준에 따른 모바일 엔터테인먼트 서비스 만족도에 관한 연구 :게임, 음악, 동영상 서비스를 중심으로," 디지털산업정보학회 논문지, 제1권, 제2호, 2005. 9, pp.79-89.

- [2] 권준희, 김성립, “유비쿼터스 환경에서 상황 데이터 기반 모바일 콘텐츠 서비스를 위한 추천 기법,” 디지털산업정보학회논문지, 제6권 제2호, 2010. 6, pp. 1-9.
- [3] I. Stoica, R. Morris, D. Liben-Nowell, D. Karger, “Chord : A Scalable P2P Lookup Service for Internet Applications,” In IEEE/ACM Transaction on Networking, Vol. 12, No. 2, Apr 2004, pp. 205-218.
- [4] A. Rowstron and P. Druschel, “Pastry: Scalable, Distributed Object Location and Routing for LargeScale Peer-to-peer Systems,” In IFIP/ACM Int’l Conf. on Distributed Systems Platforms (Middleware), Nov. 2001, pp.329-350.
- [5] B. Zhao, L. Huang, J. Stribling, S. Rhea, A. Joseph, and J. Kubiatowicz, “Tapestry: A Resilient Globalscale Overlay for Service Deployment,” IEEE Journal on Selected Areas in Communications, Vol. 22, No. 1, 2004, pp.41-43.
- [6] S. Ratnasamy et al., “A Scalable Content Addressable Network,” Proc. ACM SIGCOMM, 2001, pp.161-72.
- [7] Napster, “http://www.napster.com/”.
- [8] Gnutella, “http://www.gnutella.com/”.
- [9] 윤영효, 박후근, 김정길, 정규식 “모바일 P2P 환경에서 효율적인 네트워크 자원 활용을 위한 반응적인 코드,” 정보과학회논문지, 36(2), 2009, pp.480-489.
- [10] 김혁수, 황준호, 유명식, “모바일 애드 혹 네트워크에서 이동성을 고려한 안정적인 클러스터링 기법,” 한국통신학회논문지, Vol. 34 No. 5, 2009. 5, pp.480-488.

■ 저자소개 ■



홍 록 지
Hong, Rok Ji

2010년 9월~현재
한국기술교육대학교 컴퓨터공학부 대학원 재학
2007년 3월~2010년 8월
한국기술교육대학교 컴퓨터공학부 졸업

관심분야 : 모바일 프로그래밍, 무선 네트워크, Future 네트워킹
E-mail : shj1062@kut.ac.kr



문 일 영
Moon, Il Young

2005년 3월~현재
한국기술교육대학교 컴퓨터공학부 조교수
2004년~2005년
한국정보문화진흥원 선임연구원
2005년 2월 한국항공대학교 정보통신공학과 졸업(공학박사)
2002년 2월 한국항공대학교 항공통신정보공학과 졸업(공학석사)
2000년 2월 한국항공대학교 항공통신정보공학과 졸업(공학사)

관심분야 : 무선 인터넷 응용, 모바일 인터넷, 모바일 IP
E-mail : iymoon@kut.ac.kr

논문접수일 : 2010년 11월 4일
수 정 일 : 2010년 11월 11일(1차), 11월 18일(2차)
게재확정일 : 2010년 11월 24일