

논문 2011-06-47

퍼스널 클라우드 서비스를 위한 임의의 단말간 컨텐츠 동기화 방법

(The Method of Data Synchronization Among Devices for
Personal Cloud Services)

최은정*, 이전우
(Eunjeong Choi, Jeunwoo Lee)

Abstract : This paper describes the method of data synchronization among devices for personal cloud services. Existing data synchronization for mobile devices is based on a central server to mobile devices or a PC to a mobile device. However, the purpose of this paper is to share user data in heterogeneous environments, without depending on central server. This technology can be applied to synchronize personal data between a device and a personal cloud storage for personal cloud services. The ad hoc synchronization needs a sync agent service discovery module, a user authentication module, a network adapter, and an application data synchronization module. The method described in this paper is better than existing synchronization technology based on client-server in availability, performance, scalability quality attributes.

Keywords : ata synchronization, ad hoc, personalization, cloud computing

1. 서론

스마트폰이 일반화 되고, IPTV, PC등 사용자가 사용하는 단말의 종류도 장소 및 목적에 따라 다양해져 왔다. 이에, 사용자는 하나의 단말에서 사용되던 데이터를 다른 단말에서도 똑같이 사용하고 자하는 욕구가 증대되었다. 이는 기존의 USB등을 이용해 휴대단말과 사용자 PC간의 일부 데이터를 주고받던 것에서 훨씬 발전된 것이다. 다양한 단말간에 응용 프로그램의 데이터를 공유하기 위해 기존 기술을 확장시켜 서버를 통해 데이터를 업로드 및 다운로드할 수 있게 된 것이다.

본 논문에서는 사용자가 임의의 단말 간에 컨텐츠를 공유하기 위한 ad hoc 동기화 방법에 관하여 기술한다. 또한, 이는 퍼스널 클라우드 컴퓨팅에서 단말과 퍼스널 클라우드 스토리지 간에 사용자의 데이터를 동기화하는데 응용할 수 있다.

제2장에서는 이러한 기존의 클라이언트-서버 기반의 데이터 동기화의 문제점에 대해 논할 것이다. 제3장에서는 본 논문에서 제안하는 임의의 단말간 데이터 공유를 위한 ad hoc 동기화 기법에 관하여 기술하고 제 4장에서 결론을 맺는다.

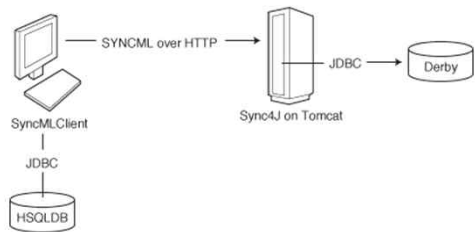


그림 1. Sync4J 네트워크 구성도
Fig. 1. Sync4J Network Topology

II. 관련 연구

1. 어플리케이션 데이터 동기화 기술
현재 서비스 되고 있는 대표적인 동기화 서비스

* 교신저자(Corresponding Author)
논문접수 : 2011. 02. 01., 수정일: 2011. 02. 22.,
채택확정 : 2011. 07. 29.
최은정, 이전우 : 한국전자통신연구원

는 SyncML[7, 22]을 기반으로 개발된 후남볼(Funambol)[19]과 마이크로소프트사에서 개발한 액티브싱크(ActiveSync)[20]와 동기화 프레임워크[23], 그리고 애플의 자체 프로토콜을 기반으로 한 모바일미(MobileMe)[21] 서비스에 사용된 동기화 기술 등이다.

그림1은 후남볼의 전신이었던 Sync4J라는 데이터 동기화를 위한 오픈소스의 네트워크 구조도를 보여준다. 그림에서 보는 것처럼, 기존의 어플리케이션 데이터 동기화 동작 구조는 동기화 클라이언트와 동기화 서버간에 SyncML을 이용해 데이터를 주고받는 클라이언트-서버 구조이다.

후남볼은 대표적으로 SyncML기반 동기화서비스를 개발하여 상용화시킨 기업이다. Sync4J에서는 모바일 기기와 컴퓨터 간의 데이터 동기화를 동기화 서버를 매개로 하여 수행한다.

애플(Apple)의 모바일미 서비스는 현재 애플의 아이폰에서 사용되고 있으며, 역시 동기화 서버를 매개로 하여 이메일, 연락처, 일정, 사진 등의 데이터를 동기화하는 서비스이다. 모바일미 서비스는 현재 프로토콜을 공개하지 않고 있으며, 애플에서 자체 개발한 프로토콜을 사용하는 것으로 알려져 있다.

마이크로소프트사의 액티브싱크 기술은 장치와 컴퓨터 사이에 이메일, 약속, 연락처, 일정 및 사진, 음악, 비디오 등의 미디어 파일을 교환하기 위해 게이트웨이를 제공한다. 장치는 마이크로소프트 익스체인지 서버(Microsoft Exchange Server)와 직접 연결함으로써 컴퓨터에서 떨어져있을 때, 무선으로도 장치에 데이터를 업데이트할 수 있다. 액티브싱크는 XML 기반의 자체 프로토콜을 사용하며 실제 데이터는 WBXML(WAP Binary XML)형식으로 변환하여 전송한다. 블루투스 및 IrDA 등에서도 동작한다. 액티브싱크는 휴대단말과 PC간 일대일 데이터 동기화 기술인 반면, 최근 마이크로소프트에서는 인터넷상에 서버를 통해 단말간의 데이터를 동기화하는 동기화 프레임워크(Microsoft Sync Framework)를 소개하고, Microsoft Sync Framework 2.1 SDK도 내놓았다.[23]

이들 동기화 서비스는 모두 단말과 동기화 서버간에 데이터 동기화를 기반으로 동작한다. 사용자의 단말은 서버와 연결이 되어야만 자신의 데이터를 동기화할 수 있다.

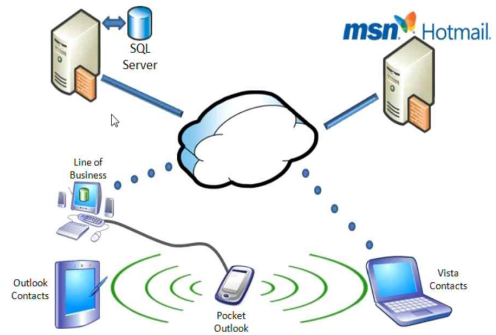


그림 2 마이크로소프트 동기화 프레임워크 예 [23]

Fig 2. Microsoft Sync Framework [23]

2. 클라이언트-서버 기반 동기화의 문제점

기존 어플리케이션 동기화 프레임워크는 클라이언트 서버 구조를 기반으로 동작한다. 클라이언트에서 수정된 데이터를 서버의 스토리지에 저장한다. 사용자가 여러 개의 단말을 사용하게 되어 모든 단말에 동일한 데이터를 저장하기 위해서는 사용자의 각 단말에 있는 동기화 클라이언트는 동기화 서버에 접속해서 데이터를 받아온다. 동기화 서버는 인터넷 상에 위치하게 되고, 사용자의 단말들은 서로 인접해 있음에도 불구하고 기존 동기화 구조에 따르면, 물리적으로 가까이 위치한 단말보다 원격지에 위치한 서버에서 데이터를 받아오게 되는 것이다. 이때, 많은 네트워크 트래픽을 유발하게 된다.

또한 기존 동기화 기술에 사용된 동작구조는 중앙 집중적인 구조를 갖게 되어 발생하는 몇 가지 문제점이 있다. 먼저, 서버와의 연결이 끊어지면 단말의 데이터를 서버로 동기화할 수 없는 단점이 있다. 두 번째, 서버와의 동기화가 이루어지지 못한 상태에서 단말의 분실 혹은 데이터 손상 시 데이터를 보존할 수 없는 문제점이 있다. 세 번째, 하나의 단말에서 생성한 데이터를 다른 단말을 통해 이용하고자 할 때, 서버와의 동기화가 이루어지지 않은 상태에선 이용할 수 없는 문제점이 있다. 이러한 문제점들은 모두 단말간 동기화는 서버를 매개로 여러 단말에서 변경된 데이터를 각각 서버로 동기화하기 때문에 발생하는 것들이다. 즉, 단말간 데이터 공유를 위해 반드시 서버를 거쳐야 하기 때문에, 서버와의 연결이 끊기게 되면 다른 단말과의 콘텐츠 공유는 어려워지는 것이다.

III. 임의의 단말간 Ad Hoc 동기화

본 논문의 목적은 종래 기술의 문제점을 해결하고자, 단말의 데이터 동기화 시에 서버와의 연결에만 의존하지 않고, 사용자의 임의의 단말들 간에 콘텐츠를 공유하기 위한 것이다. 즉, 하나의 단말에서 데이터가 수정되어 서버와 동기화를 시도할 때 서버와의 네트워크가 끊긴 경우에, 이용 가능한 네트워크(예: 블루투스, Zigbee 등)를 이용해 주변 사용자 기기로 데이터를 동기화하는 기술이다.

1. 네트워크 구성도

본 논문에서는 사용자가 다수의 장치를 사용하며 동일한 데이터로 동기화되기를 원한다는 가정을 두고 있다. 따라서, 그림2의 예에서처럼 사용자가 이용 가능한 다수의 장치가 존재하는 네트워크 구성도를 가정할 수 있다. 그림 3에서 보여지는 것처럼, 각 장치가 가진 네트워크 특성상 장치(A)와 장치(C)는 연결되어 있지 않을 경우, 장치(A)에서 생성 혹은 수정한 콘텐츠를 네트워크를 통해 장치(C)로 자동으로 옮겨갈 수 없다. 그러나, 하나의 장치가 주변에 연결 가능한 장치들에게 콘텐츠를 옮겨놓을 수 있다면, 장치(A)는 장치(B)를 통해 장치(C)로 콘텐츠를 옮겨놓을 수 있는 것이다. 이때, 기존의 동기화 서버는 사용자가 이용 가능한 하나의 장치가 될 수 있다. 장치(C)와 장치(D)가 연결되지 않았더라도 장치(A), 장치(B), 장치(C) 및 장치(D)는 데이터를 공유할 수 있다.

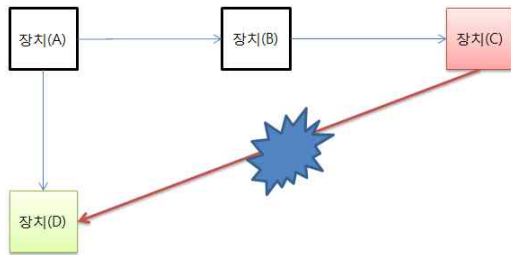


그림 3 단말간 동기화시 네트워크 구성도
Fig 3. Network Topoloby for Ad hoc Synchronization

2. Ad Hoc 동기화 프로시저

상기와 같이 직접 연결되지 않은 단말간에 데이터를 공유하기 위해서는, 중앙 집중적인 서버에 의존하지 않고, 주변의 연결 가능한 사용자의 임의의 단말과 데이터를 주고받을 수 있어야 한다. 이를 위

한 프로시저는 다음과 같다.

먼저, 동기화할 모든 데이터는 최초 생성된 시점과 수정된 시점을 기록한다. 이는 여러 단말들이 동시 다발적으로 동기화가 진행되었을 때, 데이터의 충돌을 방지하기 위한 방법 중 하나이다. 즉, 데이터를 동기화할 때, 데이터의 수정된 시점을 기준으로 다른 단말과 동기화를 수행한다.

다음으로, 어플리케이션 데이터의 수정이 발생하면, 이를 인지하여 동기화 에이전트가 수행된다.

다음 단계로, 동기화 에이전트는 수정되거나 생성된 데이터를 임의의 다른 단말들과 공유하기 위해 연결 가능한 단말을 찾는다. 블루투스, ZigBee, WiFi 등 연결 가능한 네트워크를 기반으로 연결된 단말들에게 메시지를 보내어 ad hoc 동기화 에이전트 서비스를 찾는다.

다음 단계는 인증단계이다. 상기 메시지에 응답하는 단말이 있으면, 사용자 아이디, 패스워드 등의 사용자 인증정보를 이용해 단말의 사용자를 인증한다.

단말의 인증 프로세스가 완료되면, 데이터가 수정되었음을 연결된 단말에 알려주고, 연결된 단말상의 데이터와 비교하여 동기화를 수행한다. 이 단계는 기존의 동기화 방법과 유사하다.

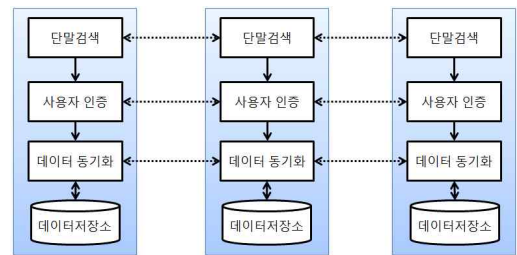


그림 4 Ad hoc 동기화 프로시저
Fig 4. Ad hoc Synchronization Procedure

3. 단말인증절차

단말 상에는 사용자를 인증하기 위한 아이디, 패스워드 등의 정보를 담고 있어야 한다. 단말검색 후, 임의단말에 동기화 에이전트가 응답을 하면 암호화된 아이디와 패스워드 정보를 임의 단말로 보낸다. 임의단말상의 동기화 에이전트는 갖고 있는 자신의 해당 아이디, 패스워드를 상대단말의 아이디, 패스워드와 비교하여 단말의 사용자임을 인증하는 메시지를 보내어 인증절차를 마무리한다. 동기화 에이전트는 해당 단말 사용자의 아이디, 패스워드 정보만 관리한다.

사용자 패스워드의 변경이 있을 수 있는데, 이것 역시 동기화의 대상이다. 서버는 사용자의 단말의 일부로 보면 된다. 단말의 인증을 위해서 각 단말을 인증할 수 있는 정보를 관리하는 중앙 집중적인 서버를 접속해야 할 필요는 없다. 만일, 인증 서버로의 접속이 끊어져 있으면 단말 인증이 이루어지지 않아, 단말간 동기화가 이루어질 수 없기 때문에, 단말 인증을 하는데 있어 서버에 의존하지 않는다.

하나의 단말이 사용자의 모든 단말 정보를 갖고 있을 필요는 없다. 동기화를 하게될 단말을 인증하는 것은 단말의 사용자를 인증하는 절차로 보면 된다. 따라서, 단말은 사용자의 아이디, 패스워드 정보만을 가지고 있으면 된다.

4. 동기화 에이전트 구조도

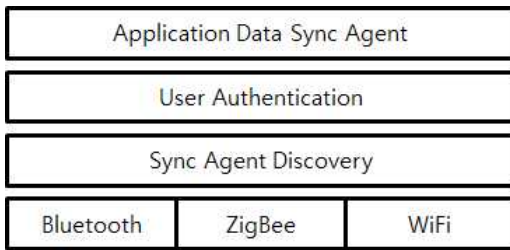


그림 5 동기화 에이전트 구조도
Fig 5. Sync Agent Architecture

본 논문에서, 사용자가 이용하는 장치마다 동기화 에이전트가 존재한다. 각 동기화 에이전트는 장치검색모듈(Sync Agent Discovery), 네트워크 어답터, 인증모듈(User Authentication), 동기화모듈(Application Data Sync Agent), 콘텐츠 저장소 등으로 구성된다. 장치검색모듈은 콘텐츠를 동기화할 장치를 찾는 모듈이다. 이때, 서버와 같이 사용자의 장치를 미리 등록하여 사용자의 장치들을 찾아 연결한다. 등록된 장치들이 연결이 안 될 경우, 블루투스에서 사용되는 SDP(Service Discovery Protocol)을 이용하여 네트워크 특성에 따라 적절히 주변 장치를 검색하여 ad hoc 동기화 에이전트를 찾아낸다. 이때, 동기화 에이전트를 찾기 위한 메시지를 보내고, 응답이 있으면 인증과정으로 넘어간다. 네트워크 어답터는 블루투스, ZigBee, WiFi 등 장치에서 이용 가능한 네트워크를 찾아 연결해주는 모듈이다. 인증모듈은 사용자가 동기화할 장치 및 동기화 에이전트에 대한 인증과정을 담당한다.

사용자 단말의 인증은 사용자 아이디와 패스워드를 이용하거나 장치의 고유번호를 이용한다. 동기화 모듈은 인증된 장치와 콘텐츠를 상호간에 동기화한다. 콘텐츠 저장소는 사용자가 설정한 동기화할 콘텐츠를 저장한다.

5. 클라우드 서비스를 위한 Ad hoc 동기화 분석

앞서 임의의 단말간 데이터 동기화에 대하여 논하였다. 본 논문에서의 동기화 방법을 이용하면, 퍼스널 클라우드 서비스에서의 퍼스널 스토리지와의 데이터 공유도 가능하다. 단말의 데이터를 퍼스널 클라우드 서버와 동기화할 때, 동기화 에이전트(A)는 동기화 서버와 연결을 할 수 없을 때, 연결 가능한 동기화 에이전트(B)를 찾아 데이터를 동기화한다. 동기화 에이전트(B)는 동기화 서버와 연결하여 퍼스널 클라우드 스토리지에 데이터를 저장할 수 있다. 어느 단말이든 동기화 서버와의 네트워크가 연결이 되면 최종적으로 동기화를 완료한다. 본 논문의 구성에 따르면, 단말과 서버간의 연결이 끊긴 상태에서도 주변의 연결 가능한 다른 단말에 데이터를 동기화할 수 있다. 따라서, 보다 안정적으로 단말 간에 혹은 단말과 퍼스널 클라우드 스토리지와 데이터 동기화를 할 수 있는 이점이 있다.

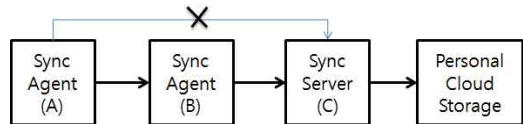


그림 6 클라우드 서비스를 위한 Ad hoc 동기화 예
Fig 6. Ad hoc Synchronization for Cloud Service

본 논문에서 제안한 임의의 단말간 ad hoc 동기화 방법은 가용성(availability), 성능(performance), 확장성(scalability)등의 품질 면에서 다음과 같은 장점을 갖는다.

가용성은 얼마나 오랫동안 정확히 동작 하는지를 평가한다. 가용성 측면에서, 기존 기술은 단말간 데이터 동기화가 서버에 의존하므로 서버와의 연결이 끊어지거나 서버가 작동하지 않으면 단말들간에 데이터 동기화는 할 수 없게 된다. 반면, 본 논문에서 제시한 방식은 서버와의 연결이 끊어지거나 서버가 작동하지 않더라도 가까이 있는 연결 가능한 단말과 데이터를 동기화할 수 있어, 단말간 네트워크 연결만 가능하다면 언제든 단말간 데이터 동기화가 가능하다는 이점이 있다.

확장성은 단말의 수가 늘어나도 동일한 기능과 성능을 보장하는 품질 요소이다. 기존 데이터 동기화 방식은 서버의 처리능력 및 서버와의 연결에 의존하기 때문에 동기화를 하기 위한 사용자 및 단말의 수가 제한적이다. 그러나, 본 논문에서 제시한 방식은 단말간 동기화가 가능하므로 사용자 및 단말의 수가 늘어나더라도 성능 및 기능에 제한이 덜하다.

이중 단말 간 데이터 동기화 속도에 있어서, 중앙 집중적인 데이터 동기화 방식에서의 성능은 네트워크 대역폭이나 서버의 처리성능에 따라 단말에서의 동기화 성능이 좌우된다. 즉, 두 단말이 인접하여 있고, 네트워크로 연결되었다 하더라도 하나의 단말에서 서버로 데이터를 업로드 한 후 다른 단말이 다시 서버와 연결하여 데이터를 다운로드받아야하므로 시간이 더 많이 소요된다. 그러나, 본 논문에서 제시한 방식은 서버를 거치지 않고도 데이터를 동기화할 수 있어 단말간 데이터 동기화 처리속도가 빨라질 수 있다.

IV. 결 론

본 논문에서의 동기화 방법을 요약하면 다음과 같다. 먼저, 사용자는 사용 중인 단말 상에서 동기화할 콘텐츠를 설정한다. 다음으로, 연결된 네트워크를 통해 동기화할 사용자의 단말을 찾는다. 주변의 모든 장치에 동기화를 하는 것이 아니라, 단말의 사용자 인증을 하여야만 장치에 콘텐츠를 동기화할 수 있다. 단말에 대한 인증 방법은 사용자 아이디와 암호를 통한 방법과 단말의 아이디를 미리 등록하는 방법을 사용할 수 있다. 마지막으로, 단말간 변경된 콘텐츠를 상호 동기화한다.

결과적으로 본 논문의 구성에 따르면, 다음의 시나리오가 가능하다. 핸드폰에서 사진을 찍어 놓으면 연결 가능한 사용자의 장치를 통해 콘텐츠가 동기화 된다. 사용자는 사진을 집안에서 TV처럼 큰 화면에서 사진을 볼 수도 있고, 서재에 있는 PC에서 사진을 편집할 수도 있다. 원본 데이터는 서버에 자동으로 보관되어 편리하다. 본 논문에서 제시한 동기화 방법은 중앙 집중적인 동기화 서버 없이도 임의의 단말들 간에 콘텐츠 공유가 가능한 이점이 있다.

본 논문에서의 단말간 ad hoc 동기화 방법은 가용성, 성능, 확장성 등의 품질 면에서 기존 방법보다 더 나은 장점을 갖는다.

참고문헌

- [1] 류수희, 최훈, 류시원, SyncML 프로토콜을 이용한 데이터 동기화 서버 Agent 설계 및 구현, 한국정보처리학회 추계 학술 발표논문집, 제8권 제2호, 2001년.
- [2] 박성진, 튜플 단위 메시지 다이제스트를 이용한 이기종 모바일 데이터 동기화 기법, 한국인터넷정보학회논문지, v.7 no.5 p.1-12, 2006.
- [3] 장대진, ActiveSync 자료 변환을 위한 SyncML 자료 동기화 게이트웨이, 한국인터넷정보학회논문지, v.7 no.3 p.61-69, 2006
- [4] 김은희, 무선 통신 환경에서 이동 서버간의 데이터 동기화 기법, 정보처리학회논문지 D, v.13 no.7 p.901-908, 2006
- [5] 심명선, 내장형 XML 데이터베이스를 기반으로 한 XML 문서 데이터 동기화 시스템,멀티미디어학회논문지, v.8 no.9 p.1153-1162, 2005
- [6] 라황균, 다중 디바이스에서 데이터 동기화를 위한 충돌 해결, 멀티미디어학회논문지, v.8 no.2 p.279-286, 2005
- [7] 장대진, SyncML 기반의 자료 동기화 클라이언트 개발, 정보과학회논문지 : 컴퓨팅의 실제, v.11 no.4 p.357-367, 2005
- [8] Sreeram, J., Pande, S., Exploiting approximate value locality for data synchronization on multi-core processors, (2010) IEEE International Symposium on Workload Characterization, IISWC'10, art. no. 5650333, .
- [9] Su, Z., Hou, X., Application of data synchronization based on ESB, (2010) 2010 2nd IITA International Conference on Geoscience and Remote Sensing, IITA-GRS 2010, 1, art. no. 5603013, pp. 295-297.
- [10] Zhu, Q., Cao, J., Adaptive synchronization under almost every initial data for stochastic neural networks with time-varying delays and distributed delays, (2011) Communications in Nonlinear Science and Numerical Simulation, 16 (4), pp. 2139-2159.
- [11] Li, N., Zhang, Y., Hu, J., Nie, Z., Synchronization for general complex dynamical networks with sampled-data, (2011) Neurocomputing, 74 (5), pp. 805-811.
- [12] Tennyson, M.F., A study of the data

- synchronization concern in the observer design pattern, (2010) ICSTE 2010 - 2010 2nd International Conference on Software Technology and Engineering, Proceedings, 1, art. no. 5608911, pp. V171-V175.
- [13] Wu, X., Data synchronization for integration systems based on message digest, (2010) ICEIT 2010 - 2010 International Conference on Educational and Information Technology, Proceedings, 1, art. no. 5607757, pp. V1197-V1199.
- [14] Cheng, C.-L., Du, J.-Q., Design and implementation of data synchronization in embedded TCM system based on SyncML, (2010) ICCASM 2010 - 2010 International Conference on Computer Application and System Modeling, Proceedings, 10, art. no. 5622636, pp. v10619-v10622.
- [15] Jiang, C., Song, M., Liu, K., Xu, K., An novel encryption protocol for mobile data synchronization based on SyncML, (2010) Communications in Computer and Information Science, 105 CCIS (PART 1), pp. 1-8.
- [16] Zhang, C.-K., He, Y., Wu, M., Exponential synchronization of neural networks with time-varying mixed delays and sampled-data, (2010) Neurocomputing, 74 (1-3), pp. 265-273.
- [17] Lee, S.-H., Kapila, V., Porfiri, M., Panda, A., Master-slave synchronization of continuously and intermittently coupled sampled-data chaotic oscillators, (2010) Communications in Nonlinear Science and Numerical Simulation, 15 (12), pp. 4100-4113.
- [18] Baccelli, E., Cordero, J.A., Jacquet, P., Optimization of critical data synchronization via link overlay RNG in mobile ad hoc networks, (2010) 2010 IEEE 7th International Conference on Mobile Adhoc and Sensor Systems, MASS 2010, art. no. 5663945, pp. 402-411.
- [19] Funambol, <http://www.funambol.com>
- [20] Microsoft ActiveSync, <http://www.microsoft.com/windowsmobile/en-us/help/synchronize/device-synch.mspx>
- [21] Apple MobileMe, <http://www.apple.com/mobileme/>
- [22] SyncML Initiative, <http://www.syncml.org>
- [23] Moe Khosravy and Liam Cavanagh, Introduction to Microsoft Sync Framework - Synchronization Framework for Enabling Roaming, Offline, and Collaboration Across Devices, Services and Apps, Microsoft Sync Framework Developer Center, <http://msdn.microsoft.com/ko-kr/sync/bb980926.aspx>

저 자 소 개

최 은 정



1999년 경북대 컴퓨터과학과 학사.
2001년 경북대 컴퓨터과학과 석사.
2007년 CMU 소프트웨어공학과 석사.

관심분야: 클라우드 컴퓨팅, 모바일 웹, 임베디드 소프트웨어, 소프트웨어 공학.
Email: ejchoi@etri.re.kr

이 전 우



1983년 경북대 전자공학과 학사.
1985년 경북대 전자공학과 석사.
1998년 경북대 전자공학과 박사.

현재, 한국전자통신연구원 소프트웨어연구부문 차세대컴퓨팅연구부 부장.
관심분야: 홈서버 구조, Post-PC 플랫폼, 가상화 기술, 클라우드 컴퓨팅.
Email: ljwoo@etri.re.kr