

# Non-Java 디바이스를 지원하기 위한 Jini 서로게이트 시스템 설계

최현석<sup>o</sup>, 모상덕, 정광수, 이혁준<sup>\*</sup>

광운대학교 전자공학부, 광운대학교 컴퓨터공학과<sup>\*</sup>

{hschoi,sdmo}@explore.kwangwoon.ac.kr, {kchung,hlee}@daisy.kwangwoon.ac.kr

## Design of Jini Surrogate System for Supporting Non-Java Devices

Hyunseok Choi<sup>o</sup>, Sangdok Mo, Kwangsue Chung, Hyukjoon Lee<sup>\*</sup>

School of Electronics Engineering, Kwangwoon Univ.

Dept. of Computer Engineering, Kwangwoon Univ.<sup>\*</sup>

### 요 약

내장형 디바이스를 인터넷과 연결하고자 하는 요구가 늘어나고 있다. 이러한 정보기기들을 동적으로 상호 작용하게 하는 기술로서 지니(Jini)가 최근 주목을 받고 있다. 그러나 내장형 디바이스에서 지니 서비스를 제공하기 위해 요구되는 높은 하드웨어 사양이 문제점으로 지적되고 있다. 본 논문에서는 이러한 non-Java 디바이스를 지니 네트워크에 접속하여 지니 서비스를 제공할 수 있도록 서로게이트 시스템(Surrogate System)과 non-Java 디바이스간의 프로토콜을 설계하였다.

### 1. 서론

최근 인터넷이 발전하면서 정보기기 및 디바이스들을 인터넷에 연결하고자 하는 요구가 늘어나고 있다. 이러한 디바이스들과 소프트웨어를 분산 네트워크로 통합하는 기술이 썬 마이크로시스템즈에서 연구되어 자바를 기반으로 한 지니(Jini) 기술이 99년 1월 공식적으로 발표하였다. 지니는 사무실 및 가정 내의 정보기기를 네트워크로 연결하기 위한 분산 네트워크(Distributed Network) 분야의 미들웨어(Middleware)로서 주목받고 있다. 그러나, 온도계와 같은 소형의 내장형 디바이스에 지니 기술을 탑재하려면 기존 디바이스보다 더 빠른 연산 속도와 많은 메모리가 요구되기 때문에 지니를 탑재하기란 쉽지 않다. 이와 같은 내장형 디바이스에서 지니가 갖는 문제점을 해결하기 위해 지니 네트워크에서 지니 서비스를 검색 및 등록 기능을 대신해주는 서로게이트 호스트(Surrogate Host)가 필요로 한다.

본 논문에서는 non-Java 디바이스의 지니 서비스를 지니 네트워크에 등록하고 non-Java 디바이스와 지니 클라이언트간의 중계역할을 담당하는 서로게이트 시스템(Surrogate System)을 제시하고, 서로게이트 호스트와 디바이스간에 필요한 프로토콜을 설계하였다.

### 2. 관련 연구

지니는 썬 마이크로시스템즈에서 제안하고 있는 접속 기술로서 자바를 기반으로 LAN, 모뎀, 무선 등의 다양한 방식으로 망에 접속된 디바이스나 소프트웨어를 동적으로 상호 작용하게 하는 분산 미들웨어 기술이다. 실행 시에 네트워크상에 분산되어 있는 구성요

소들을 탐지하여 동적으로 시스템을 구성하는 구조이며, 각종 디바이스를 통해서 네트워크에 접속이 이루어지면 시간과 장소에 상관 없이 각종 서비스를 받을 수 있는 분산 네트워크 기술이다. 따라서 하드웨어나 소프트웨어에 상관없이 지니를 채택하고 있는 디바이스들에 대한 인위적인 설치나 조작과 같은 일련의 절차를 배제할 수 있는 Plug and Work가 가능하다.

지니 시스템에 참여하고 있는 모든 디바이스들이나 소프트웨어들은 서비스(Service)로 정의된다. 지니 시스템은 서비스 이용자(Jini Client), 서비스 제공자(Jini Service), 서비스 관리자(Lookup Service)로 구성된다. 그리고 각각의 구성요소들은 Discovery, Join, Lookup 프로토콜을 이용하며, 서비스 이용자와 서비스 제공자는 자바 RMI(Remote Method Invocation)를 사용하여 상호 통신한다.

그러나 내장형 디바이스에서 지니 서비스를 이용하기 위해 자바를 시스템에 탑재하는 데에는 많은 문제점들이 있다. 내장형 운영체제 위에 자바를 탑재하려면 상당량의 메모리가 필요하게 되며, 지니 네트워크의 기본 구성요소인 Discovery 및 Lookup 서비스에 사용되는 자바 RMI 클래스까지 포함되어야 하므로 상당한 오버헤드를 갖기 때문에 내장형 디바이스에 지니를 적용하기 힘들다. 본 논문에서는 이러한 문제를 해결하기 위해 지니 개발자 공동체에서 시작된 서로게이트 프로젝트를 기초로 하여 디바이스의 지니 역할을 대행해주는 서로게이트 시스템(Surrogate System)을 연구하고 설계하였다.

### 3. 서로게이트 시스템 설계

일반적으로 디바이스를 지니 네트워크에 연결하기 위해서는 디바이스에 자바가 탑재되어야 한다. 하지만 디지털 온도계와 같은 소형 디바이스에 자바를 탑재하려면 더 많은 메모리와 프로세싱 파워가 요구되어 진다. 그러나, 내장형 운영체제 위에 소켓 API (Socket Application Programming Interface) 정도만 제공하는 내장형 디바이스 제품이 자바까지 올린 디바이스 제품보다 더욱 범용성이 있고 구현하기가 쉽다. 내장형 디바이스가 지니 네트워크와 연결되어 서비스를 제공하려면 지니의 Discovery 및 Lookup 서비스 기능이 필요하므로 소켓 API만 제공하는 네트워크 디바이스는 Discovery와 Lookup 기능을 대행해주는 서로게이트 시스템이 요구된다.

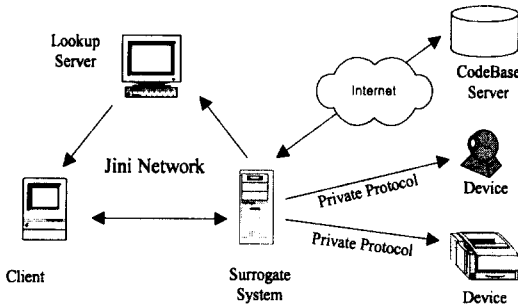


그림 1. 서로게이트 시스템

Non-Java 디바이스를 지니 네트워크에 접속하기 위한 서로게이트 시스템 모델을 그림 1에 나타내었다. 디바이스들은 서로게이트 시스템을 통해서 지니 네트워크에 접속하여 지니 서비스를 제공할 수 있다.

#### 3.1 서로게이트 시스템 구조

서로게이트 시스템을 구성하는 중요한 요소는 서로게이트 호스트와 서로게이트 프로토콜이다. 서로게이트 호스트는 지니 환경이 없는 디바이스의 서비스 프락시(Service Proxy)를 다운받아 동적으로 실행시켜주는 시스템이다. 서로게이트 프로토콜은 서로게이트 호스트와 디바이스간의 서로의 존재를 찾고 서비스 프락시를 다운받은 후, 서로게이트 호스트 내에 서비스 프락시와 디바이스의 상태를 유지 관리하기 위해 이용되는 프로토콜이다.

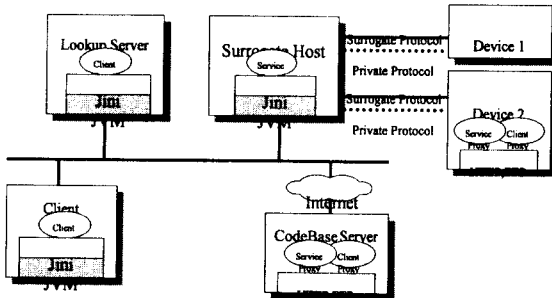


그림 2. 서로게이트 시스템 구조

그림 2에서와 같이 디바이스 1은 디바이스 2보다 리소스가 더욱 제한되어 서비스 프락시와 클라이언트 프락시도 갖지 못하는 환경이므로 외부 네트워크의 Code Base 서버에서 다운받도록 프락시 URL을 서로게이트 호스트에게 알려준다. 만약 CodeBase 서버가 디바이스를 제공한 회사에서 관리될 경우 향후 서비스의 기능 개선 및 유지관리의 측면에서 많은 장점을 가질 수 있다.

서로게이트 호스트는 Interconnect Adapter 모듈, Host Resource Manger 모듈, Surrogate Activator 모듈로 구성된다. 그림 3에서와 같이 Interconnect Adapter 모듈은 디바이스를 검색하고 서로게이트 프락시를 다운받는 기능을 담당하고, Surrogate Activator 모듈에게 서로게이트 프락시를 넘겨준다. Surrogate Activator 모듈은 서로게이트 프락시를 자바의 동적 클래스 로딩 기능을 이용하여 서로게이트 프락시의 활성화/비활성화시키는 역할을 담당한다.

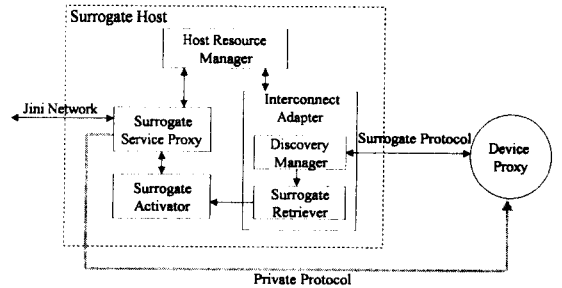


그림 3. 서로게이트 호스트 모듈 구성

#### 3.2 서로게이트 프로토콜

서로게이트 프로토콜이란 디바이스가 서로게이트 호스트를 찾아내거나, 또는 서로게이트 호스트가 디바이스에게 자신의 존재를 알리고, 디바이스를 서로게이트 호스트에 등록하기 위한 상호 규약이다. 또한 프락시로 사용될 서로게이트 코드를 디바이스에서 다운받는 방식과 서로게이트 호스트와 디바이스간의 상호 존재 유무를 검사하는 방식을 포함하고 있다.

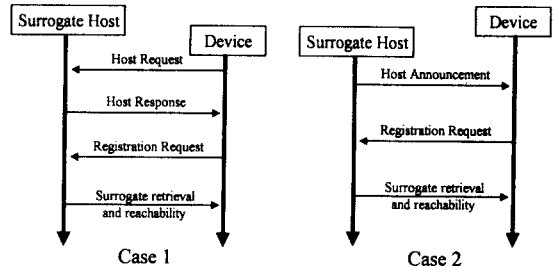


그림 4. 서로게이트 프로토콜

그림 4은 서로게이트 호스트와 디바이스간의 서로의 존재를 알리고 등록하는 과정을 나타내고 있다. 그림 4의 Case 1은 서로게이트 호스트가 이미 존재하는 네트워크에 디바이스가 나중에 접속한 경우이다. 이 후에 접속한 디바이스는 네트워크에 멀티캐스트로 자신

의 존재를 알린다. 서로게이트 호스트가 디바이스의 메시지를 확인한 후 해당 디바이스에게 호스트의 정보를 포함하는 응답 메시지를 보낸다. 이 후 디바이스는 서비스로 사용될 서로게이트의 정보를 등록 요구 메시지에 담아서 보낸다.

그림 4의 Case 2는 서로게이트 호스트가 재시작 되거나, 디바이스가 네트워크에 이미 존재한 후에 서로게이트 호스트가 접속된 경우이다. 이때 서로게이트 호스트는 자신의 존재를 알리기 위해 멀티캐스트 메시지를 내보낸다. 디바이스는 멀티캐스트 포트로 서로게이트 호스트의 메시지를 감시하다가 메시지가 발견되면 등록 요구 메시지에 서비스로 사용될 서로게이트의 정보를 담아서 보낸다.

등록요구 메시지를 받은 서로게이트 호스트는 디바이스가 보내온 등록요구 메시지에서 프락시의 위치를 파악하여 다운받은 후, 서로게이트의 서비스가 완료될 때까지 디바이스의 존재 유무를 계속 점검해야 한다.

다음은 그림 5의 서로게이트 프로토콜 메시지에 대한 설명이다..

• 호스트 요청 메시지(Host Request Message)

디바이스는 서로게이트 호스트를 찾기 위해 멀티캐스트 UDP 메시지를 사용한다. Device address 필드와 Port to connect to 필드는 디바이스 주소와 호스트 응답 메시지에서 사용될 포트 번호로 구성된다.

• 호스트 응답 메시지(Host Response Message)

호스트 요청 메시지를 보낸 디바이스에게 그에 대한 응답으로 서로게이트 호스트가 보내는 유니캐스트 UDP 메시지이다. Host address 필드는 디바이스가 등록요청 할 때 사용되는 호스트의 주소이다.

• 호스트 광고 메시지(Host Announcement Message)

서로게이트 호스트가 자신의 존재를 알리기 위한 멀티캐스트 UDP 메시지이다. 메시지 구조는 호스트 응답 메시지의 구조와 같다. 단지 유니캐스트가 아닌 멀티캐스트이라는 점만 다르다.

int	int	byte[]	int
Protocol version	Length of device address	Device address	Port to connect to

(a) Host Request Message

int	int	byte[]	int
Protocol version	Length of host address	Host address	Port to connect to

(b) Host Response Message / Host Announcement Message

int	byte[]	
Length of surrogate URL	Surrogate URL	
Length of initialization data	Initialization data	
Length of surrogate	Surrogate	

(c) Registration Request Message

그림 5. 서로게이트 프로토콜 메시지 구조

• 등록 요청 메시지(Registration Request Message)

디바이스가 서로게이트 호스트에게 서로게이트 등록을 위해 필요한 정보를 보내는 메시지이다. 메시지 구조는 그림 5의 (c)와 같으며, 서로게이트 URI 필드에는 서로게이트 jar 파일을

다운받을 수 있는 주소와 다운 시 사용될 프로토콜(HTTP, FTP)이 들어간다. Length of Surrogate URL 필드 값이 0이면, 메시지 중에 서로게이트 필드 내부에 서로게이트가 함께 포함되어 전달된다. 서로게이트의 크기가 작을 경우에는 이와 같은 방법을 사용하면 따로 서로게이트 jar 파일을 다운받을 필요가 없지만, 컴파일 시 서로게이트가 함께 포함되므로 서비스의 유지보수가 어려워지는 단점이 있다.

이상의 네가지 메시지는 서로게이트와 디바이스가 서로의 존재를 확인하고 서로게이트 서비스를 등록하기 위한 것들이다. 디바이스와 서로게이트 간의 존재 확인을 위한 메시지(Reachability Message) 방식은 구현 의존적이라고 할 수 있다. 이는 다양한 디바이스들이 존재하고 각 디바이스마다 각기 다른 특성을 가지고 있으므로 존재 확인을 위한 방법도 달라야 하기 때문이다. 예를 들어, 온도를 측정하는 디바이스는 존재확인 주기가 길어도 되지만, 침입감지를 위한 디바이스는 존재확인 주기도 짧아야 하며 오류가 발생했을 경우의 대처 방안도 각기 달라야 한다.

4. 결론 및 향후 과제

본 논문에서는 non-Java 디바이스가 지니 네트워크에 접속하여 서비스할 수 있는 서로게이트 시스템을 설계하였고, 디바이스와 서로게이트 시스템이 서로 동적으로 존재를 확인하고 등록된 후 서로의 연결을 확인하는 서로게이트 프로토콜을 설계하였다. 이를 통해 보다 다양한 디바이스가 지니 네트워크에 접속하여 서비스를 제공할 수 있다.

향후 서로게이트 시스템을 홈 게이트웨이(Home Gateway) 서버에 적용할 수 있도록 구현할 예정이다.

5. 참고 문헌

- [1]. Sun Microsystems Inc., Jini Connection Technology, <http://www.sun.com/jini/>
- [2]. W. K. Edward, "Core Jini", Prentice Hall, 1999.
- [3]. Scott Oaks, Henry Wong, "Jini in a Nutshell", O'reilly, March 2000.
- [4]. C. E. McDowell, K. Shankari, "Connecting non-Java devices to a Jini network", Proc. of 33rd International Conference on Technology of Object-Oriented Languages, June 2000.
- [5]. Paul J. Perrone, Venkata Chaganti, "Jini in the box", Embedded Systems Programming, November 1999.
- [6]. Jini Community, "Jini Surrogate Project", <http://developer.jini.org/exchange/projects/surrogate/>
- [7]. Sun Microsystems Inc., "Jini Device Architecture Specification", November 1999.
- [8]. Erik Guttman, James Kempf, "Automatic Discovery of Thin Servers: SLP, Jini and the SLP-Jini Bridge", Proc. of the 1999 IEEE 25th Annual Conference of the IEEE Industrial Electronics Society, 1999.