

# CORBA 기반 고신뢰성 네이밍 서비스를 이용한 SDR 시스템 환경에서 지역 투명성 제공

송상현<sup>0</sup> 김문기 이병호  
한양대학교 정보통신대학원  
{92facno2<sup>0</sup>, mkim, bhrhee}@scann.hanyang.ac.kr

## Local Transparency using High Reliability Naming Service based on CORBA in SDR Systems

Sang Hyun SONG<sup>0</sup>, Mun Gi KIM, Byung Ho RHEE  
The Graduate School of Information and Communication, Hanyang University

### 요 약

분산 시스템 상에서 SDR 소프트웨어 시스템을 구현하려면 재구성성과 유연성이 필요하다. 따라서 이를 충족시켜줄 수 있는 객체지향 컴퓨팅 기술이 필요하게 된다. 이러한 분산 객체 구조를 가능하게 하는 핵심은 CORBA 미들웨어이다. CORBA는 소프트웨어 버스로서 분산성과 이종성을 높이고 분산 객체들이 유연하게 통신할 수 있게 해주는 핵심 역할을 담당한다. 분산 네트워크 환경 하에서 CORBA의 네이밍 서비스는 네임 서버에 저장한 논리적 이름을 통해서 객체의 위치 정보를 얻는다. 이 방식은 서버의 물리적인 위치 변화에 상관없이 객체의 위치 투명성을 제공한다. 본 논문에서는 SDR 시스템에서 클라이언트가 재구성을 위한 소프트웨어를 다운로드를 받기 위해 서버의 구현객체에 논리적인 이름을 이용하여 접근을 가능하게 해주는 신뢰성 있는 CORBA 네이밍 서비스를 설계함으로써 SDR 시스템의 지역적 투명성을 지원하는 모델을 제안한다.

### 1. 서 론

SDR(Software Defined Radio)은 개방형 구조와 모듈화 된 단일 하드웨어 플랫폼 상에 객체지향구조의 응용소프트웨어를 다운로드하여 언제(anytime), 어디서나(Anywhere) 끊임 없이(seamless) 전역(global) 통신이 가능하게 하는 기술로 기존 시스템(legacy system)과의 역호환성(backward compatibility) 제공과 All-IP 기반 무선 멀티미디어를 추구하는 미래 시스템(4G) 이동통신의 무선망 통합을 위한 해결 방안으로 고려되고 있다.[1][2]

SDR은 소프트웨어의 다운로드와 재구성이 핵심이므로, SDR을 지원하기 위해 시스템 소프트웨어가 제공해야 할 기능은 SDR에서 사용되는 다양한 하드웨어를 추상화시켜 SDR 응용을 고급 언어로 개발할 수 있게 하는 것과 SDR 응용이 재구성될 수 있도록 컴포넌트 기반 컴퓨팅을 지원할 수 있어야 하는 것이다. 이 중 후자의 기능이 응용 소프트웨어와 운영체제 사이에 위치하는 중간 계층인 내장형 미들웨어에 의해서 제공된다. 또한 SDR 소프트웨어 참조 모델에서 기술하는 다양한 기능들을 구현하기 위해서 SDR 시스템은 여러 하드웨어 노드들로 구성된 분산 시스템으로 설계되어야 한다. 이러한 분산 시스템 상에서 SDR 소프트웨어 시스템을 구현하려면 재구성성과 유연성(flexibility)이 필요하며 따라서 이를 충족시켜줄 수 있는 객체지향 컴퓨팅 기술이 필요하게 된다.

SDR Forum[3]에서는 JTRS(Joint Tactical Radio System)[4]의 SCA(Software Communication Architecture)[5]를 SDR 소프트웨어 구조의 표준으로 인정하였으며, SCA는 SDR을 구성하는 분산 노드들의 유연한 통신을 위한 미들웨어의 표준으로 CORBA를 채택하였다. SDR에서 미들웨어의 기능은 다양한 이기종의 하드웨어, 운영체제, 네트워크의 존재를 추상화 시켜주고, 특정 컴포넌트를 찾기 위한 디렉토리와 네이밍 서비스 등의 서비스를 제공하고, 필요에 따라 특정 도메인에서 필요한 라이브러리 서비스를 제공한다. 특히 SCA는 소프트웨어 다운로드와 재구성을 위한 분산 환경의 소프트웨어 컴포넌트들을 검색하기 위하여 CORBA의 네이밍 서비스(Naming Service)를 사용할 것을 명세하고 있

다.

본 논문에서는 CORBA 네이밍 서비스에 대해서 알아보고, SDR 시스템에서 클라이언트가 재구성을 위한 소프트웨어를 다운로드 받기 위해 서버의 구현객체에 논리적인 이름을 이용하여 접근을 가능하게 해주는 신뢰성 있는 CORBA 네이밍 서비스를 설계함으로써 SDR 시스템의 지역적 투명성을 지원하는 모델을 제시하고자 한다.

### 2. CORBA Naming Service

CORBA(Common Object Request Broker Architecture)는 소프트웨어 객체가 분산 환경에서 작동하는 방식에 대한 일단의 명세서이다. 이 명세서는 많은 소프트웨어 업체들에 의해 결성된 OMG(Object Management Group)에서 관리하고 있다. OMG의 CORBA Naming Service는 CORBA 표준 서비스에서 가장 기본적인 서비스이다. 이 네이밍 서비스는 분산된 객체(Object)에서 객체 참조를 쉽게 얻어올 수 있도록 한곳에 모아서 이용 가능하게 해주는 것으로, CORBA application에서 클라이언트는 객체의 이름으로 분산된 객체를 쉽게 찾을 수 있도록 한다.[6]

OMG 네이밍 서비스는 표준 CORBA 명세의 가장 기본이 되는 서비스이다. 네이밍 서비스는 객체에 고유한 이름을 제공하는 것이다. 객체에 이름을 매핑하기 위해서는 먼저 application server에서 컨텍스트(context)에 이름을 바인드(bind)한다. 고유한 이름으로 바인딩 된 객체는 네이밍 서버에 트리(tree)구조의 그래프 형식으로 등록이 된다. 클라이언트는 네이밍 서버에 객체 참조를 얻기 위해 이름을 질의 하고 해석하여 이름에 바인딩 된 객체 참조를 얻을 수 있다. 이러한 방법으로 네이밍 서버에 저장한 논리적 이름을 통해서 객체의 위치 정보를 얻을 수 있고, 서버의 물리적인 위치 변화에 상관없이 객체의 지역 투명성(Local Transparency)을 제공한다.[7][8][9] 그러나 기존의 네이밍 서비스의 기능은 단순히 데이터베이스 역할만 할 뿐, 동적인 환경 하에서 서버와 클라이언트 간의 유동적인 실효 정보를 업데이트 하지 못하였다. 다음 절에서 이러한 제약

을 해결하기 위한 신뢰성 있는 네이밍 서비스 모델을 제안한다.

### 3. 고신뢰성 Naming Service의 설계

#### 3.1 시스템 구성

아래 그림 1은 SDR 터미널과 네이밍 서버, 그리고 각각의 구현 객체를 포함하고 있는 서버와의 관계를 보여주고 있는 전체 시스템 구성도이다.

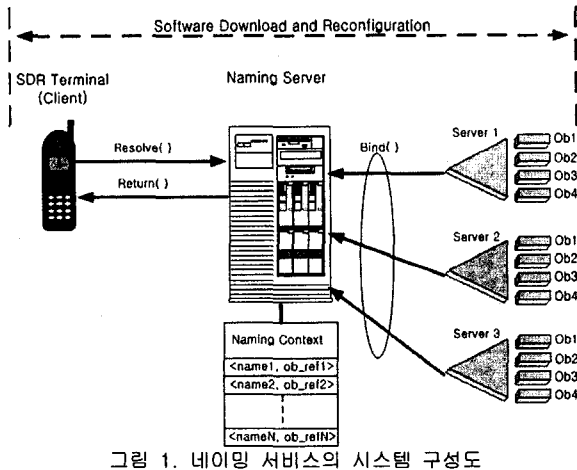


그림 1. 네이밍 서비스의 시스템 구성도

구현 객체를 포함하고 있는 각각의 서버는 지역 투명성을 제공하기 위해 활성화가 되면서 객체들을 논리적인 이름과 바인딩하고 네이밍 서버에 이름과 참조 정보를 등록시킨다. 서비스를 받아서 단말기를 재구성 하려는 SDR 터미널은 네이밍 서버의 등록 테이블을 검색하게 되는데, 그 과정은 Bind( ) 동작을 통해서 트리(tree)구조의 그래프 형태로 형성된 네이밍 그래프를 Resolve( ) 동작으로 객체의 이름과 참조를 찾는다. Resolve( ) 동작을 통해 네이밍 그래프를 탐색하는 형태는 그림 2와 같다. 각각의 이름은 특정 컨텍스트(context)에 바인딩되는 구조적인 형태를 갖게 된다. 이러한 구조적 형태를 네이밍 그래프라고 하며, 검색 노드는 네이밍 컨텍스트이고 흰색 노드는 이름과 객체 참조가 바인딩된 것을 말한다. Resolve( ) 동작은 가장 최상위 Root부터 시작하여 트리를 순회하며 객체를 찾게 된다. 네이밍 서버는 SDR 터미널이 Resolve( )를 통해서 검색한 이름과 객체 참조를 Return( )하게 되고 이 과정을 통해 이름과 객체 참조를 얻은 SDR 터미널은 객체가 어디에 존재하는지 상관없이 객체가 제공하는 서비스를 다운로드 받아서 터미널을 재구성하게 된다.

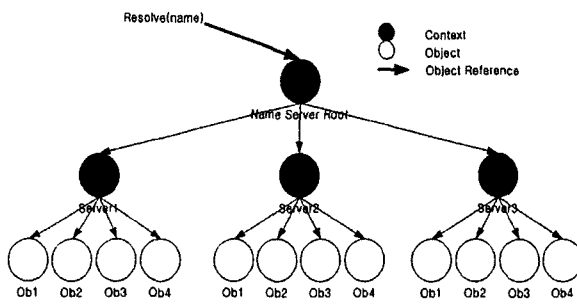


그림 2. 네이밍 그래프에서 객체 검색 과정

#### 3.2 제안된 네이밍 서비스의 수행모델

기존의 네이밍 서버는 단순히 데이터베이스 역할만 할 뿐 동적인 환경 하에서 서버 객체의 유동적인 실패 정보를 효과적으로 업데이트 하지 못하였다. 이에 대해 본 논문에서 제안하는 모델은 네이밍 서버에 접근한 SDR 터미널이 최초 객체 검색에 실패할 경우, 네이밍 서버가 구현 객체를 포함하고 있는 서버에게 재 바인딩을 할 것을 요구하고 서버로부터 새로운 바인딩 정보를 돌려받아 네이밍 서버의 등록 테이블을 새롭게 업데이트한 후 SDR 터미널의 서비스 요청에 대해서 그 동작을 재 수행 하는 모델을 추가하는 것이다. 이에 대한 네이밍 서비스의 과정을 그림 3과 그림 4에 나타내었다.

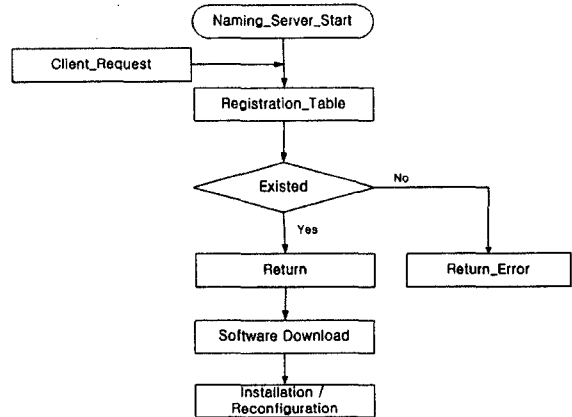


그림 3. 기존의 네이밍 서비스의 흐름도

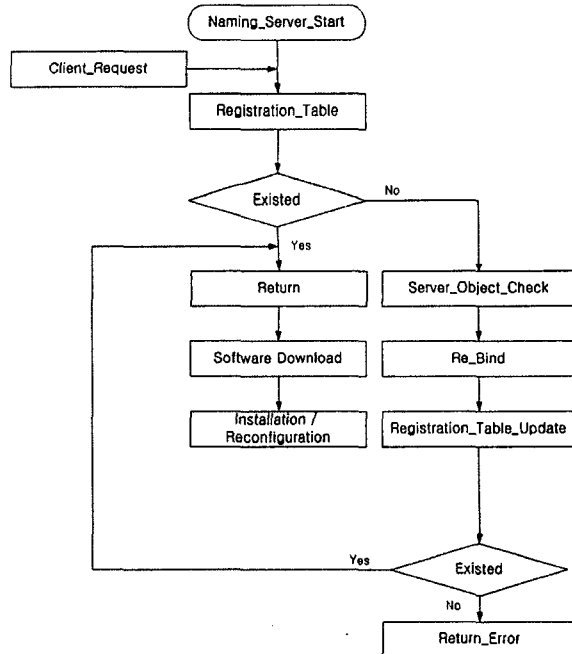


그림 4. 제안된 네이밍 서비스의 흐름도

바인딩 동작을 재 수행하는 과정을 통해, 분산 네트워크 환경에서 서버의 구현 객체의 변화, 즉 객체의 생성, 삭제, 변경

그리고 서버의 위치변경 등의 동적인 상태에 대해서 유연성 있게 등록 테이블을 업데이트 할 수 있다. 업데이트 후에 등록 테이블에 접근하고자 하는 객체의 이름이 존재한다면 객체 참조 반환 전의 과정으로 되돌아가서 클라이언트의 요구를 계속 수행 하고, 존재하지 않는다면 객체 참조 반환에 실패하게 된다.

3.3 시스템 호출(call) 처리 절차

이번 절에서는 제안된 신뢰성 있는 네이밍 서비스에 대해서 SDR 터미널과 네이밍 서버, 그리고 구현 객체를 포함하고 있는 서버 사이에서 발생하는 시스템 호출의 처리 절차에 대해서 살펴본다. 아래 그림 5의 경우는 SDR 터미널이 최초 검색을 통해 객체를 찾지 못하였을 때, 네이밍 서버로부터 부정응답(NAK) 신호를 받은 후 네이밍 서버와 구현 객체를 포함하는 서버 간에 바인딩의 재수행 요구와 등록 테이블의 업데이트 과정을 수행하는 시스템 호출 과정을 설명하고 있다.

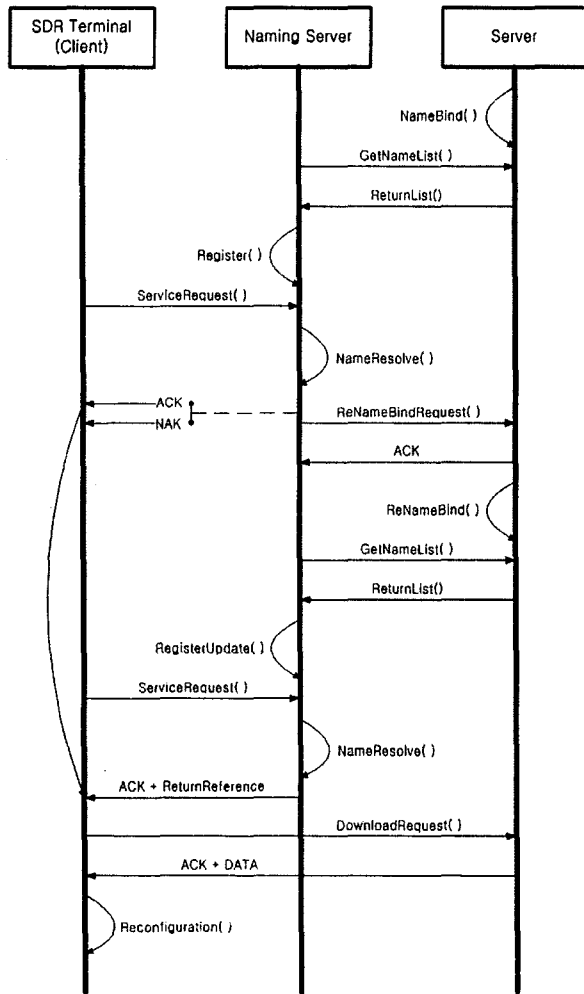


그림 5. 제안된 네이밍 서비스의 신호처리 절차

제안된 네이밍 서비스의 신호처리 절차는 네이밍 서버로부터 ACK와 NAK 신호에 따라 동작이 달라진다. ACK 신호를 받게 되면 정상적인 명시적 알고리즘을 수행하고, NAK 신호를 받게 되면 재수행 바인딩 동작과 등록 테이블의 업데이트 동작이 요구된다. 이러한 요구가 완료 되면 SDR 터미널은 Resolve() 동작

을 재수행 함으로써 구현 객체의 이름과 객체 참조를 얻어올 수 있다. 여기까지가 네이밍 서비스의 동작이며 SDR 터미널은 반환된 객체의 이름과 객체 참조를 통해서 소프트웨어 다운로드와 그 이후의 터미널의 재구성 동작을 수행하여 지역적으로 투명한 서비스를 받을 수 있게 된다.

4. 결론 및 향후 연구과제

본 논문에서는 전반부에서 SDR 시스템의 개념과 SDR이 유연한 통신 환경이 되기 위해서 필요한 소프트웨어 다운로드와 재구성성, 그리고 이와 같은 시스템을 지원하는 필수적인 미들웨어에 대해서 살펴보았다. 미들웨어의 분야에서는 SCA가 표준으로 채택한 CORBA 미들웨어가 있으며, 그 서비스 중에는 분산 네트워크 환경 상에서 논리적인 이름을 이용하여 CORBA 객체가 다른 객체를 효과적으로 찾을 수 있도록 지원하는 CORBA 네이밍 서비스에 대해서 살펴보았다. 후반부에서는 기존의 네이밍 서비스에 환경 변화에 동적으로 적응할 수 있는 바인딩의 재수행 동작과 그 결과로 생성되는 정보들을 네이밍 서버에 업데이트하는 알고리즘을 추가시켜서 신뢰성 있는 네이밍 서비스 모델을 제안하였다. 본 논문을 통해 제안된 고신뢰성 네이밍 서비스를 통해서 SDR 시스템이 다운로드와 재구성 시에 지역적인 투명성을 보장받을 수 있을 것으로 기대된다.

현재는 제안된 네이밍 서비스의 평가를 위해 테스트망의 구축을 진행하고 있으며, 향후에는 이 모델이 실제 시스템 상에서 효과적으로 동작하는지에 대한 검증과 이러한 네이밍 서비스의 절차들이 SDR 시스템에서 다운로드와 재구성 시에 과도한 오버헤드로 작용하지 않는지에 대해서 검증을 해보아야 할 과제이다.

5. 참고 문헌

- [1] 김지연, 김진업, SDR 기술의 현재와 발전방향, 한국통신학회지, 19권, 11호, 2002.
- [2] T.Shono, K.Uehara and S.Kubota, Proposal and evaluation of system diversity for software defined radio, IEEE International Conference, vol 1, April 2002.
- [3] Software Defined Radio(SDR) Forum, <http://www.sdrfourm.org>
- [4] Joint Tactical Radio System(JTRS), <http://www.jtrs.sault.army.mil/>
- [5] Software Communications Architecture(SCA) Specification MSFC-5000SCA V2.2, Joint Tactical Radio Systems, November 17, 2001.
- [6] OMG, The Common Object Borker: Architecture and Specification, OMG Document Formal/98-12-01, Revision 2.3, 1999.
- [7] Reaz Hoque, CORBA 3, 정보문화사, 2000.
- [8] 김영욱, 장연세, CORBA3 Programming Bible, 도서출판 그린, 2000.
- [9] Henning Vinoski, Advanced CORBA Programming with C++.