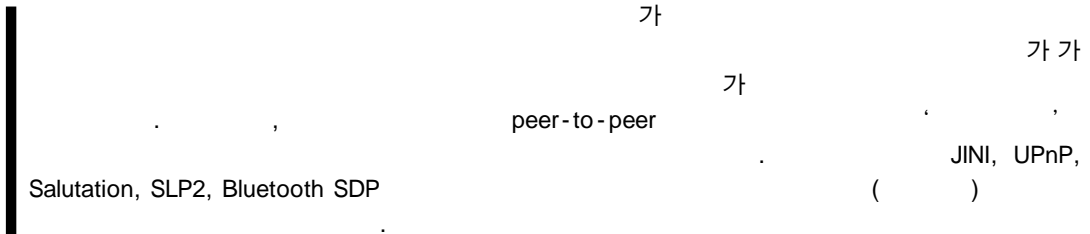


(S.I. Yoon)  
(G.C. Sohn)



I.

향후 유무선이 통합되고 위치인식 기반의 서비스가 제공되는 유비쿼터스 컴퓨팅 환경에서는 다양한 장치들이 상호 연결되어 네트워크 상의 서비스를 자동으로 발견하고 서비스는 자신의 능력을 광고할 수 있는 서비스 발견(service discovery)의 특성이 중요해질 것이다. 서비스 발견 프로토콜(service discovery protocol)은 네트워크 상에서 가용한 서비스를 동적으로 발견하고 특정 장치가 원하는 서비스를 정확하게 매핑하는 메커니즘을 제공하므로 네트워크 관리, 서비스 발견이나 이용 측면에서 사용자 및 서비스 제공자 모두에게 편의성을 제공해준다. 지금까지 서비스 발견과 관련된 연구가 진행되면서 정보가전에 SDP를 적용하는 다양한 시나리오들이 제안되어 왔으며 가까운 미래에는 이를 실제 생활에 적용할 수 있을 것으로 예측된다.

본 논문은 다음과 같은 내용으로 구성된다. 첫째, JINI, UPnP, Salutation, SLP2, Bluetooth SDP에서 제공하는 서비스 발견 프로토콜을 분석 기술하고

둘째, 각각 이들의 특성을 비교하고자 한다.

II.

1. JINI Connection Technology

JINI는 네트워크 상의 모든 종류의 디바이스와 소프트웨어 자원의 통합체를 구성하여 서비스와 자원을 공유하고 사용자의 위치에 관계없이 용이한 자원의 접근 및 네트워크의 개설, 갱신, 변경 작업의 단순화를 목표로 하는 JAVA 기반의 미들웨어 기술로서 1998년 SUN에서 개발한 분산 환경의 홈 네트워크 자원 공유 플랫폼이다[1]. JINI는 네트워크 미디어에 관계없이 JVM(Java Virtual Machine)만 있으면 동작이 가능하며 플러그 앤 플레이를 지원한다. JINI는 현재 정의되고 있는 대부분의 미들웨어 기술의 근간을 이루고 있으며, 특히 OSGi의 기술적 기반을 제공한다. 그러나, 기본 개념과 기술적인 우수성에도 불구하고 JVM과 RMI(Remote Method Invoke)를 기본적으로 포함하는 고사양의 컴퓨팅이

요구되어 실제 가전기기에 적용되는 데는 한계가 있어 많은 호응을 얻지 못하고 있다[1].

JINI 시스템은 클라이언트(client), 서비스 제공자(service provider), 룩업 서비스(lookup service)로 구성되며 이들 삼자간의 상호 통신으로 이루어진다.

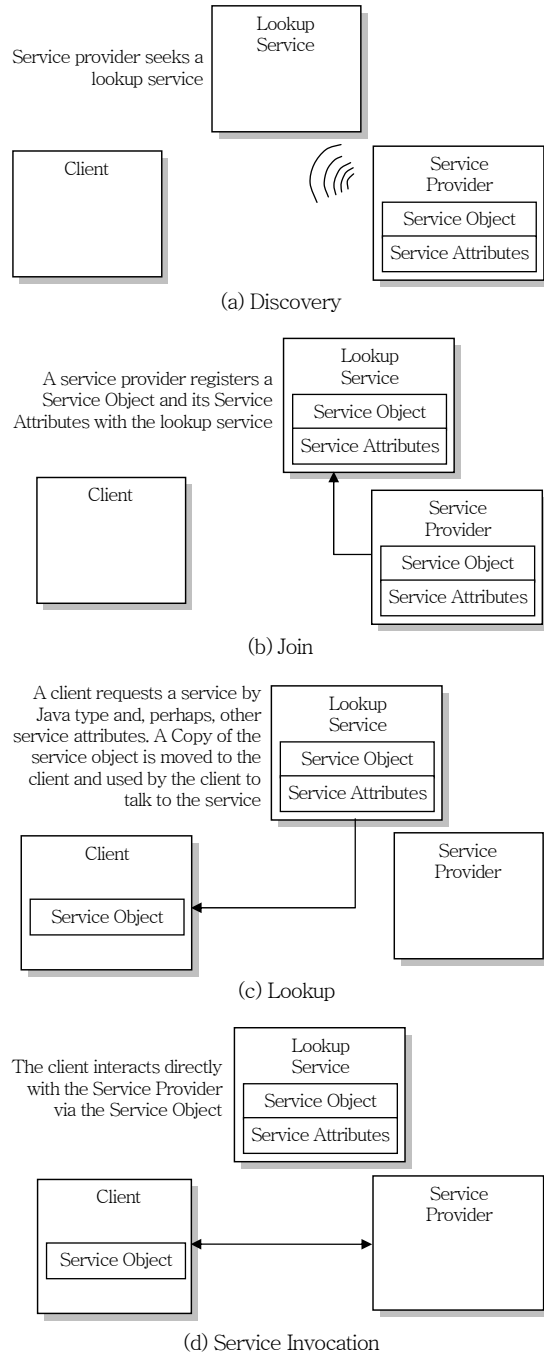
JINI 서비스 구조의 핵심은 discovery, join, lookup 프로토콜이다.

- Discovery: 서비스나 클라이언트가 네트워크 상에서 룩업 서비스를 찾는 과정으로 서비스는 자신을 등록하기 위해, 클라이언트는 등록된 서비스를 이용하기 위해 각각 룩업 서비스를 찾는다.
- Join: 서비스 객체가 룩업 서비스에 위치한 후 JINI 공동체에 참여하는 과정이다.
- Lookup: 클라이언트나 사용자가 룩업 서비스를 통해 자신이 원하는 서비스를 가져오는 과정이다.

JINI 공동체에 포함된 클라이언트들이 서비스를 이용하기 위해서는 (그림 1)과 같은 과정을 거친다 [2]. 서비스 제공자는 로컬 네트워크상에서 멀티캐스트 요청을 통해 룩업 서비스를 발견하여(그림 1(a)) 서비스 객체(service object) 및 서비스 속성(service attributes)을 룩업 서비스에 등록한다(그림 1(b)). 서비스 객체는 서비스 실행을 invocation 할 수 있는 방법과 함께 서비스를 위한 JAVA 프로그래밍 인터페이스를 포함하고 있다. 클라이언트가 JAVA 유형이나 서비스의 속성 등에 따라 서비스를 요청하면 서비스 객체의 사본이 클라이언트로 이동한다(그림 1(c)). 이후, 클라이언트와 서비스 제공자간의 통신이 이루어진다(그림 1(d)).

룩업 서비스는 일종의 '디렉토리 서비스'이며 서비스들은 룩업 서비스를 통해 발견되고 이용된다. JINI 공동체에서 서비스들은 discovery와 join의 과정을 통해 자신의 proxy object를 룩업 서비스에 등록하고 클라이언트들은 자신들이 원하는 서비스를 찾기 위해 룩업 서비스를 검색한다. JINI는 서비스 발견을 위해 다음의 3가지 프로토콜을 이용한다.

- Multicast request protocol: 서비스나 클라이언트가 UDP 패킷을 통해 멀티캐스트 메시지를 보내면 근접한 룩업 서비스가 이에 반응하여 통



( 1) JINI Service Discovery

신 객체를 내려 보낸다.

- Multicast announcement protocol: 룩업 서비스가 자신의 존재를 UDP 패킷에 담아 멀티캐스트로 알리면 관심있는 서비스나 클라이언트가

이를 인지하여 특업 서비스에게 통신할 수 있는 서비스 프록시를 요청한다.

- Unicast discovery protocol: 서비스나 클라이언트가 이미 특정 특업 서비스의 정보를 가지고 있어서 특정 특업 서비스와 직접 통신한다.

특업 서비스는 단순한 네임 서버 이상의 역할을 수행하며 각 인터페이스를 서비스 프록시 객체들의 묶음으로 매핑하여 클라이언트들이 서비스를 인터페이스로 인식하도록 한다. 클라이언트가 RMI를 통해 서비스 프록시를 다운로드하면 프록시 객체는 클라이언트로 하여금 사전 정보를 가지지 않은 서비스를 이용할 수 있도록 하며 서비스는 모든 통신 프로토콜과 대화할 수 있다.

JINI 시스템은 네트워크 자동 구성 기능을 가지므로 융통성 및 자가복구 능력을 필요로 한다. JINI의 서비스 접근은 leasing 개념을 기반으로 정해진 일정 시간 동안에만 특업 서비스에 저장된 서비스의 유효성을 보장하므로 서비스는 특업 서비스로의 주기적인 갱신을 통해 자신의 존재가 유효함을 알려야 한다. 만약, 특정 시간을 재할당 받으려면 서비스 사용자와 서비스 제공자간의 협상을 통해 반드시 만료 이전에 다시 갱신되어야 한다. 그렇지 않으면, JINI는 서비스와 연계된 모든 리소스를 해제한다.

JINI의 자가복구 능력을 위한 events 개념은 클라이언트가 서비스에 자신을 등록해 둠으로써 새로운 서비스의 추가나 서비스의 상태 변경과 같은 특정 이벤트가 발생하면 클라이언트의 서비스 객체는 특업 서비스로부터 변경사항에 관련된 이벤트 메시지를 수신한다.

## 2. Universal Plug and Play Connection Technology

UPnP는 다양한 미디어 및 복수의 벤더가 존재하는 맥내망 환경에서 운영체제, 언어 및 하드웨어에 독립적인 서비스 환경을 제공하는 미들웨어로 단순하고 유연하며 표준에 기반한 단대단(peer-to-peer) 방식의 연결성을 제공하므로, 사용자는 단지 장치를 네트워크에 연결시켜 주면 네트워크 상에 연결된 기

존의 장치들이 자동으로 새로 추가된 장치를 발견하여 제어하거나 다른 장치가 가진 서비스를 찾을 수 있다. 1999년 6월 마이크로소프트, 인텔, 컴팩, 미쓰비시, 필립스, 소니 등 150여 개 업체의 참여로 PC 중심의 가전기기 제어 소프트웨어 표준을 정의하였다[1].

UPnP는 TCP/IP, HTTP, GENA, SOAP, SSDP 등 IETF 표준 인터넷 프로토콜을 기반으로 하고 있으며 DA(Device Architecture), AV(Audio Video Architecture), SA(Security Architecture)의 3가지 구조를 가진다. UPnP는 IP 네트워킹을 채택하여 확장성 및 상호 운용성을 제공하며 브리징을 통해 non-IP 프로토콜의 미디어를 수용한다. 2004년 7월 UPnP AV 1.0 규격이 공개되었고 UPnP AV 1.0 규격이 곧 발표될 것이다. DLNA 1.0 스펙에서 UPnP DA와 AV를 기본 프로토콜로 활용하면서 UPnP의 입지는 향후 홈네트워크 미들웨어의 de-facto 표준으로 자리잡을 전망이다. 그러나, 취약한 보안과 더불어 상기한 표준 인터넷 프로토콜의 부담, non-IP로 구성될 것으로 보이는 가전 및 센서/액추에이터 네트워크와의 연결 방법 등이 향후 과제로 남아 있으며, UPnP 1.0 규격이 DLNA의 요구사항을 만족시켜야 하는 부담이 있다[1].

UPnP 네트워크 구성요소는 서비스(service), 장치(device), 컨트롤 포인트(control point)이다. 장치는 서비스와 내포(nested) 장치로 구성되고 컨트롤 포인트는 서비스나 다른 장치를 찾거나 제어하는 등 네트워크 내의 컨트롤러 역할을 수행한다.

UPnP는 서비스 발견을 위해 SSDP(Simple Service Discovery Protocol)를 이용한다. SSDP는 JINI의 discovery-join-lookup과 유사한 기능으로 다른 장치에게 자신의 존재를 알리거나 다른 장치 및 서비스를 발견하는 데 이용되는 프로토콜로서 HTTPU(HTTP over unicast)와 HTTPMU(HTTP over multicast) 상에서 구현된다.

네트워크 상에 새로 추가된 장치가 자신의 서비스를 광고하기 위해 컨트롤 포인트에게 멀티캐스트 광고(ssdp: alive) 메시지를 보내면 이후 장치는 서비스들의 잠재적인 클라이언트가 된다. 반면에, 새

로운 컨트롤 포인트가 네트워크 상에 추가되면 SSDP는 검색(ssdp: discover)을 위한 멀티캐스트 메시지를 보내고 메시지를 수신한 네트워크 상의 장치들은 유니캐스트 응답을 보낸다.

JINI와는 달리, UPnP에서는 중앙의 서비스 레지스트리가 존재하지 않는다. UPnP 장치들간에 교환하는 데이터 즉, 장치의 특성이나 능력은 XML로 표현되며 HTTP를 통해 통신한다.

UPnP Device Architecture 규격에 따른 UPnP 장치간 동작 절차는 다음과 같다[3],[4].

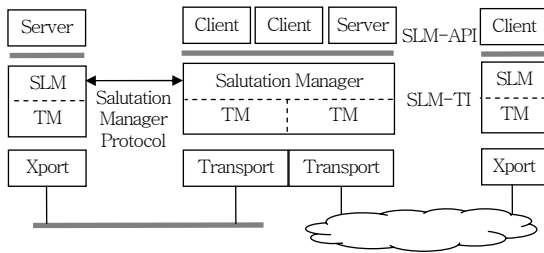
- Addressing: TCP/IP 프로토콜의 어드레싱은 UPnP 네트워킹의 핵심으로, 장치는 AutoIP를 통해 자동으로 네트워크에 참여할 수 있으며 이후 장치는 DHCP(Dynamic Host Configuration Protocol) 서버를 통해 IP 어드레스를 획득한다. DHCP 서버가 존재하면 DHCP가 할당할 어드레스를 사용하고 그렇지 않으면, 미리 예약된(reserved) 로컬 네트워크용 IP를 자동으로 할당받는다.
- Discovery: 장치가 네트워크에 접속하고 주소를 부여받으면, SSDP를 기반으로 하는 디스커버리 과정이 진행된다. 장치가 네트워크에 접속하면 SSDP는 장치가 가진 서비스를 네트워크 상의 컨트롤 포인트에게 통보하고 반면, 컨트롤 포인트가 네트워크에 접속하면 SSDP는 네트워크 내에 관심있는 장치들을 검색한다. 이때 타입, 구분자, XML 장치 기술 문서의 지시 포인터 등과 같은 장치 및 서비스의 필수 요소들이 교환된다.
- Description: 디스커버리 과정 이후 컨트롤 포인트는 장치에 대한 극히 일부의 정보만 가진다. 컨트롤 포인트는 더 많은 장치 정보를 습득하고 장치와의 상호작용을 위하여 디스커버리 메시지에 포함된 URL로부터 장치 디스크립션 정보를 수집한다. XML로 표현된 UPnP 디스크립션은 어떤 장치에 내포된 장치 정보, 서비스 정보, 제조사의 특정 정보를 포함할 수 있다.
- Control: 장치 디스크립션 정보를 수집한 후 장치를 제어할 수 있는 기본 정보를 가지게 된 컨

트롤 포인트는 이제 서비스 정보를 수집하기 위해 각 서비스의 디스크립션을 수집한다. SOAP을 통해 장치 디스크립션에 존재하는 해당 서비스의 제어 URL로 적합한 제어 메시지를 송신하여 장치를 제어하고 서비스는 이에 대한 응답으로 기능 수행에 따른 특정값이나 오류코드를 반환한다.

- Eventing: 서비스 디스크립션에 포함된 서비스 기능 수행 리스트나 서비스 상태 모델링 변수 리스트에 변화가 발생되면, 서비스는 컨트롤 포인트에게 이벤트 메시지를 보낸다. GENA 형식의 이벤트 메시지는 상태 변수에 대한 이름과 현재 값을 포함한다. 컨트롤 포인트가 최초로 정보를 입수할 때 특수한 초기 이벤트 메시지를 보내면 정보 수신자는 모든 서비스의 상태를 초기화한다(다중 컨트롤 포인트 기능도 지원).
- Presentation: 장치가 프리젠테이션을 위한 URL을 가지면 컨트롤 포인트는 이 URL로부터 페이지를 가져오거나 브라우저를 통해 볼 수 있으며, 페이지의 기능에 따라 사용자가 장치를 제어하거나 장치의 상태를 볼 수 있다.

### 3. Salutation

Salutation은 다양한 개발 플랫폼의 가전제품 및 장치간의 통신을 가능하게 하고 분산 환경에서의 연결성 및 이동성을 제공하기 위하여 Salutation Consortium이 개발한 cooperation architecture이다. Salutation Consortium은 2000년 설립되어 현재 아메리카 온라인, 액세스 커뮤니케이션즈, 캐논, 후지 제록스, 후지쓰, IBM, 코니카 등의 선두 IT 업체 및 여러 대학의 연구기관이 참여하는 비영리 단체로 개방형 소스, Salutation architecture, service discovery, session management protocol을 개발하고 있다. Salutation은 애플리케이션, 서비스 및 장치들이 자신의 능력(capability)을 기술하고 각각 다른 애플리케이션, 서비스 및 장치에게 자신을 광고하기 위한 표준을 제공하며 특정 능력을 가진 애플리케이션이나 서비스 및 디바이스들을 발견하거



( 2) Model of the Salutation Manager

나 그들을 이용할 수 있도록 상호통신을 위한 세션 요청 및 설정 기능을 제공한다.

Salutation 구조는 (그림 2)와 같다. 서비스 중개자 역할의 Salutation manager는 Salutation의 핵심으로 JINI의 록업 서비스와 동일한 기능을 수행한다. 서비스 제공자는 자신의 능력을 Salutation manager에 등록한다. 클라이언트가 Salutation manager에게 서비스 검색을 요청하면 Salutation manager들간의 협력을 통해 해당 서비스를 찾아 클라이언트에게 전달해준다. (그림 2)에서 Transport manager는 하위계층의 전송 메커니즘에 관계없이 신뢰성있는 통신채널을 제공한다. Salutation manager는 전송 계층에 독립적인 인터페이스 (SLM-API)를 서버 및 클라이언트의 애플리케이션에게 제공한다. SLM-API는 서비스의 등록, 서비스의 발견, 서비스의 접속 기능을 포함하며 Salutation manager와 Transport manager 사이에 존재하는 SLM-TI는 통신 프로토콜에 독립적인 Salutation 구조를 가능하도록 한다[5],[6].

- Service Registry: Salutation manager는 서비스 정보를 유지하기 위해 레지스트리를 가진다. 클라이언트는 로컬 혹은 연결된 Salutation manager를 통해 자신을 레지스트리에 등록/해지할 수 있으며 이는 JINI의 록업 서비스와 동일하다.
- Service Discovery: Salutation manager는 Salutation manager 프로토콜(SUN사의 ONC RPC 이용)을 이용하여 다른 Salutation manager나 미리 등록된 서비스를 찾거나 또는 협력하며 Salutation manager를 통해 미리 정의된

타입이나 속성과 비교하여 원격의 서비스를 찾아낸다. 서비스는 동일 디바이스 내의 로컬 Salutation manager에 등록되므로 ‘능력교환 (capability exchange)’ 기능을 가진다.

- Service Session Management: 서비스 활성화 (invocation)를 담당한다. 서비스 세션은 클라이언트가 service discovery를 통해 발견된 서비스를 이용하고자 할 때 설정되며 다음의 3가지 모드로 동작된다.

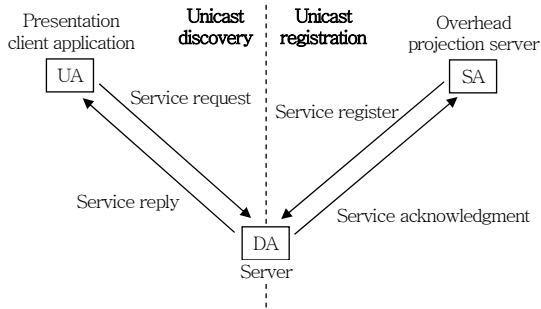
- Native mode: Native 프로토콜을 통해 메시지를 교환하며 Salutation manager는 메시지 교환에 관여하지 않는다.
- Emulated mode: 클라이언트와 서비스간의 메시지 교환을 위해 Salutation manager protocol을 이용하지만 Salutation manager가 콘텐츠에 영향을 주지 않는다.
- Salutation mode: Salutation manager가 메시지를 교환하거나 세션의 메시지 포맷을 정의하는 데 모두 관여한다.

클라이언트나 서비스는 기능적인 유닛 즉, 최소의 기능 블록들로 구성되며 기능적인 유닛의 묶음은 서비스 레코드로 정의한다. 예를 들어, 팩스 서비스는 프린트, 스캔, 팩스 데이터 송신과 같은 기능 유닛으로 정의할 수 있고 각 기능 유닛은 속성 레코드로 구성된다(ISO 8824 ASN.1. 규격).

Salutation-Lite는 Salutation 구조를 hand-held 및 palm-sized 컴퓨터, 이동단말, 홈 가전 등의 소형 정보가전에 이용할 수 있도록 축소한 형태로 1999년 발표되었다. Salutation-Lite 역시 서비스 발견 기능을 가지며 IR 또는 블루투스와의 협대역에서 활용 가능하다.

#### 4. SLP2(Service Location Protocol v.2)

IETF의 Service Location Protocol WG(SVRLOC)은 1997년에 RFC 버전 1을 발표한 이후 요구사항, 확장성, 보안, IESG 프로토콜 등을 보완하여 1999년에 RFC 버전 2를 발표하였다. SLP는 네트워크 기반의 애플리케이션이 서비스의 위치를 자동으로



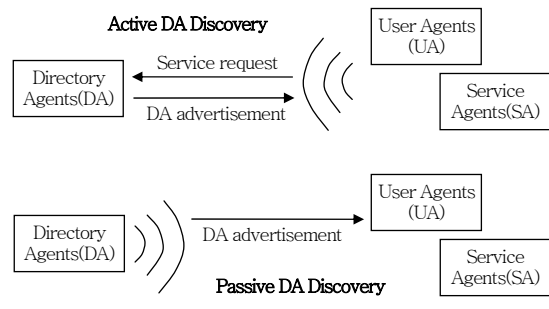
( 3) SLP

발견하는 IETF의 표준으로서 클라이언트와 서비스 간의 연결성을 제공한다. SLP는 서비스 발견을 위해 UA(User Agent), SA(Service Agent), DA(Directory Agent)를 가지며 SLP의 동작은 (그림 3)과 같이 이루어진다[7],[8].

UA는 클라이언트(사용자) 애플리케이션을 대신하여 DA에게 서비스 요청(service discovery request)을 보내고 DA로부터 응답(service reply message)을 수신함으로써 SA의 위치를 획득한다. SA는 서비스를 대신하여 서비스 등록 요청(service registration)을 보내고 DA로부터 확인 응답(service acknowledgment)을 받는다. 이제 SA는 DA에 등록함으로써 자신을 광고할 수 있게 되며 등록 메시지에는 광고할 서비스의 URL, 라이프타임, 속성 등을 포함한다. DA는 중앙의 서비스 정보 저장소 역할을 수행하며 UA와 SA간의 서비스 정보를 중재한다. 즉, DA는 SA로부터의 광고를 들은 후 UA의 요청에 응답한다.

UPnP나 JINI와 마찬가지로, 네트워크 리소스의 효율성을 높이기 위해 SA는 서비스 만료 이전에 주기적으로 DA에의 등록을 갱신하여야 한다.

SLP의 동작 초기에 UA와 SA는 네트워크 상에 DA가 존재하는지를 판단한 후 구성 파일(configuration file)을 이용하여 DA의 주소를 생성한다. DHCP를 통해 DA의 위치가 획득되는 경우에는 DA의 discovery 기능이 필요치 않으나 이외의 경우, UA와 SA는 SLP의 ‘멀티캐스트 컨버전스 알고리즘(multicast convergence algorithm)’을 이용하여 DA를 발견하고, 서비스 요청 메시지를 멀티캐스팅하여 DA의 광고 메시지를 획득한다.



( 4) SLP Service Discovery

DA의 서비스 발견 방법(그림 4)와 같이, SA와 UA가 네트워크 상의 DA로 멀티캐스트 서비스 요청을 보내어 DA의 서비스 광고를 수신하는 active discovery와 주기적으로 멀티캐스팅되는 DA의 서비스 광고를 통해 UA와 SA가 DA의 위치를 학습하도록 하는 passive discovery가 존재한다.

SLP의 동작은 DA의 존재여부에 따라 달라진다. 즉, DA가 존재하면 DA는 SA에 의해 광고된 모든 서비스 정보들을 수집하고 UA의 유니캐스트 요청에 따라 빠른 응답을 보낸다. 반면에, DA가 존재하지 않으면 UA는 반복적으로 멀티캐스트 서비스 요청을 보내고 SA는 요청을 수신하고 있다가 요청된 서비스의 광고가 발생되면 UA에게 유니캐스트 응답을 보낸다.

‘Equality’ 방식에 따른 단순한 서비스들간의 속성 매칭이 이루어지는 JINI와는 달리, SLP에서는 UA가 네트워크 상의 가장 적절한 서비스를 선택할 수 있도록 서비스 속성에 따른 서비스 브라우징 및 문자 기반의 검색(string-based query: AND, OR, comparators, substring) 기능을 제공한다. SLP는 인트라넷(SOHO)에서 최적의 서비스 발견 기술이라는 견해가 있으나 멀티캐스팅 메시지, 영역(scope) 및 다중 DAs 개념 등을 지원함으로써 네트워크 확장을 제공하며 SLP discovery는 DNS, DHCP, SLP DAs, 라우팅이 존재하지 않는 환경에서도 제공될 수 있다는 장점을 가진다.

### 5. Bluetooth SDP

10m 이내의 단거리에 위치한 컴퓨터와 주변기기,

이동단말, 가전제품 등을 무선 네트워크로 상호 연결하는 기술로서 저가 및 저전력으로 구현 가능한 WPAN의 표준은 Bluetooth SIG와 IEEE 802.15 그룹을 통해 진행중이며, 2001년 Bluetooth SIG에서 1Mbps급 버전 1.1 규격을 완료했고 현재는 2~11Mbps급 버전 2.0에 대한 규격이 작성되고 있다.

Bluetooth 기술은 둘 또는 그 이상의 Bluetooth 장치들이 상대 장치를 찾아 상호 연결하며 서비스를 탐색하거나 서비스 애플리케이션의 상태를 자동으로 확인할 수 있다.

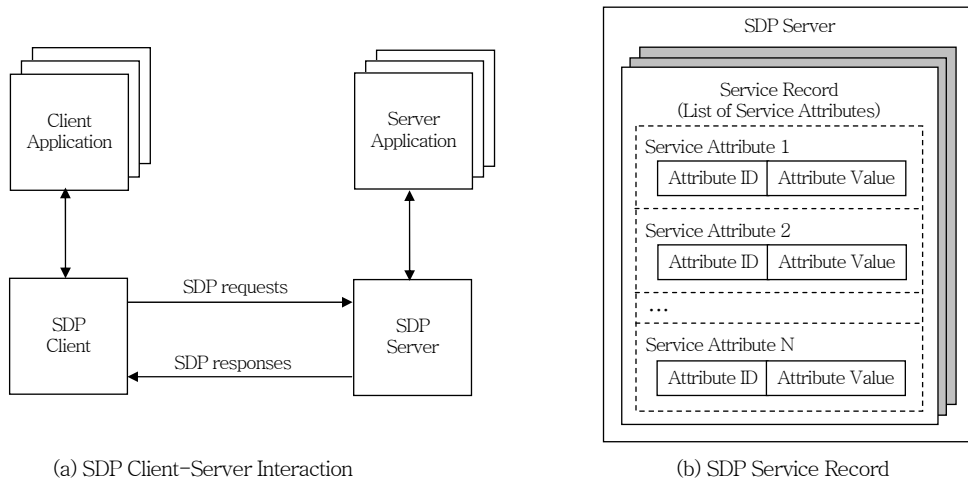
Bluetooth SDP(Service Discovery Protocol) [9],[10]는 Bluetooth 환경에서의 서비스 발견을 위한 프로토콜로 JINI, UPnP, Salutation 등과는 달리 Bluetooth 장치에만 적용되며 제한된 기능을 가진다. Bluetooth SDP는 서비스 액세스, 서비스 중재, 서비스 광고, 서비스 등록 및 이벤트 공지를 지원하지 않으며 서비스 클래스/속성에 의한 검색, 서비스 브라우징을 제공한다.

동적인 환경의 Bluetooth 장치들은 서로 상대장치를 서비스로서 인식하므로 서비스 발견 메커니즘이 요구되며 Bluetooth 개체들은 서비스의 존재나 서비스의 정보가 있어야만 발견된 서비스를 이용할 수 있다. (그림 5(a))는 SDP 서버와 SDP 클라이언트간의 통신을 위한 request-respond-protocol을 나타낸다. Bluetooth 장치 내의 클라이언트 애플리

케이션은 이를 통해 서버 애플리케이션에 의해 제공되는 가용한 서비스를 발견하고 서비스의 특성을 판단한다. SDP는 임의의 중앙 유닛이 필요로 하지 않으며 client-server 개념에 따라 서비스를 제공하는 모든 개체는 서버, 서비스를 이용하는 모든 개체는 클라이언트로 정의된다. (그림 5(a))와 같이 Bluetooth 장치 당 최대 하나의 SDP 서버를 가지며 복수의 클라이언트 애플리케이션은 서비스 발견을 위한 SDP 클라이언트를 오직 하나만 가진다.

Bluetooth 장치가 서비스를 제공하려면 서비스가 먼저 인스톨되어야 한다. 인스톨 시 새로운 서비스 레코드(service record)는 모든 SDP 서버 내에 존재하는 서비스 레코드 리스트에 추가되며 하나의 서비스 레코드는 하나의 서비스와 정확히 매핑된다. SDP 서버 내의 서비스 레코드는 서비스 레코드 핸들(service record handle)에 의해 유일하게 식별되나, 서비스 레코드 핸들은 각 SDP 서버에서 독립적으로 할당되므로 전체 네트워크 상에서 유일하지는 않다. (그림 5(b))에서와 같이 서비스 레코드는 서비스의 모든 정보 즉, 속성 ID와 속성 값을 쌍으로 갖는 서비스 속성(service attribute)을 기술하며 서비스의 한 가지 특성은 하나의 서비스 속성으로 표현된다.

4가지 일반적인 서비스 속성 즉, ServiceRecordHandle, ServiceClassIDList, ServiceID, Bro-



( 5) Bluetooth SDP

wseGroupList는 모든 서비스에 명시되어 있으며 몇몇 공통 특성을 가진 유사한 서비스들은 공통 수퍼클래스의 서브클래스들을 정의하여 세부적인 속성을 추가로 기술함으로써 유사한 서비스들을 구별할 수 있다. 모든 서비스는 적어도 하나의 서비스 클래스의 인스턴스이다.

UUID(Universally Unique ID)는 표준 규격에 따라 128bit 값으로 구성되며 분산환경에서 독립적으로 할당된 UUID를 기반으로 클라이언트는 서비스의 타입을 탐색하고 발견된 서비스가 장치 내의 클라이언트 애플리케이션에 의해 이용되는지 결정한다. 서비스 레코드의 중요한 서비스 속성값들은 UUID 형태로 저장되어 있으며 각 서비스 클래스는 유일한 식별자로 할당된다.

Bluetooth 서비스 발견 메커니즘을 예로 설명하면 다음과 같다.

프린팅 서비스를 제공하는 프린터(서버)와 프린팅 서비스를 이용하는 디지털 카메라(클라이언트)는 모두 Bluetooth 장치로서 동일한 PAN 영역 내에 위치한다. 디지털 카메라가 프린터를 이용하기 위해서는 먼저 프린팅 서비스가 인스톨되어야 하며 SDP는 서비스 접근 및 적용에 관한 메커니즘을 제공하지 않으므로 서비스 이용 전에 서비스 발견 및 적용 방법을 알아야 한다. 디지털 카메라는 프린팅 서비스를 기술하는 서비스 레코드를 접근하여 서비스 특성을 확인하고 특정 서비스 레코드를 식별하는 서비스 레코드 핸들이 획득되면 서비스 레코드의 값을 얻게 된다. 클라이언트는 원하는 유형의 모든 서비스들을 발견하여 서비스 레코드 핸들을 가져오으로써 서비스의 능력(capability)을 확인한다.

SDP는 서비스 발견 방법으로 '서비스 검색(searching for services)' 및 '서비스 브라우징(browsing for services)' 기능을 제공한다.

'서비스 검색' 메커니즘은 클라이언트들은 서비스 레코드에 포함된 특정 속성값을 기반으로 특정 서비스 레코드의 서비스 레코드 핸들을 검색한다. 예로써, 디지털 카메라가 프린팅 서비스를 찾기 위해 특정 속성값을 가진 모든 서비스들에게 SDP 클라이언트 request를 보내면 request 메시지를 수신한 모

든 SDP 서버는 특정 속성값과 일치하는 모든 서비스 레코드들을 확인하여 속성값이 일치하거나 그의 서브셋이면 SDP response를 통해 서비스 레코드 핸들을 디지털 카메라로 보낸다((그림 5(a)) 참조). 이때, 특정 서비스 레코드 검색에 적용되는 서비스 검색 패턴(service discovery pattern)의 UUIDs가 서비스 레코드 속성값 내의 UUID의 서브셋인 경우 서비스 검색 패턴은 서비스 레코드와 일치한다고 본다. 적절한 서비스 레코드 핸들을 수신한 디지털 카메라는 상대 서비스 레코드에 접근하여 가용한 서비스인지 확인하고 특정 요구사항에 맞는 사진을 인쇄할 수 있는 프린팅 서비스에 접근할 수 있게 된다. 참고로, 이러한 2단계의 발견 메커니즘을 가지는 이유는 SDP가 UUIDs의 발견 능력만을 제공하기 때문이다.

'서비스 브라우징' 메커니즘은 클라이언트들이 서비스 특성에 대한 사전 지식없이 서비스를 발견하도록 한다. '서비스 브라우징'에서는 모든 서비스 클래스가 공유하는 BrowseGroupList의 속성을 이용하고 계층구조 형태를 가지는 browse group은 서비스의 기능이나 사용자 그룹에 따라 서비스를 분류한다. 최상위 계층에 위치하는 root browse group은 모든 SDP 서버에 존재하며 가장 먼저 검색되어야 하는 각 서비스들은 BrowseGroupList 속성값에 UUID를 포함시켜 root browse group의 멤버로 지정한다. 새로 정의된 browse group은 차상위 browse group의 UUID를 가진다.

Bluetooth SDP에서는 클라이언트가 SDP 서버의 서비스를 검색하면 오직 하나의 UUID를 포함하는 서비스 검색 패턴이 생성되며 요청을 받은 SDP 서버는 root browse group과 관련된 모든 서비스 레코드 핸들에게 응답한다. 서비스 레코드의 특성에 대한 확인이 끝나면, 이제 클라이언트는 하위 browse group을 통해 서비스를 브라우징할 수 있다. 위에서 설명한 메커니즘을 통해 원하는 서비스의 서비스 레코드를 발견하여 모든 속성값들이 클라이언트의 요구와 일치하면 SDP의 임무는 끝난다.

클라이언트가 서비스를 더 이상 이용할 수 없다면



적절한 서비스 레코드는 SDP 서버로부터 제거된다. Bluetooth SDP는 '변경 공지(change notification)' 기능을 지원하지 않으므로 클라이언트는 프린팅 서비스 삭제에 대한 공지 메시지를 보내지 않는다. 만약, 클라이언트가 서비스 레코드 핸들을 이용하여 이미 삭제된 서비스를 적용하려 하면 오류 응답을 수신하게 되고 이로써 클라이언트는 서비스가 더 이상 유효하지 않다고 인식한다.

## 6.

앞서 기술된 JINI, UPnP, Salutation, SLP2, Bluetooth SDP에서 제공되는 서비스 발견 프로토콜의 특성을 비교하면 <표 1>과 같다[6],[11].

## III.

앞서 살펴본 여러 기술에서 제공되는 '서비스 발견(service discovery)'은 네트워크 자동 구성 및 자동 복구의 기능을 가지는 미래의 네트워크 및 위치인식 기반 서비스가 제공되는 유비쿼터스 컴퓨팅 환경의 주요 특성이 될 것이다. '서비스 발견'을 통해 장치들은 네트워크 상의 서비스를 자동으로 발견하고 서비스들은 자신의 능력을 광고할 수 있다. 현재 JINI, UPnP, SLP, Salutation에서 제공하는 다양한 서비스 발견 프로토콜들이 존재하며 Bluetooth SDP는 이들보다 제한된 서비스 발견 기능을 제공한다. 지금까지 이들은 서로 다른 시각으로 서비스 발견 프로토콜에 접근해 왔으므로, 향후에는 다양한 장치간의 서비스 발견을 위해 다른 서비스 발견 프

< 표 1 >

Features		JINI	UPnP	Salutation	SLP	Bluetooth
Home Page		www.sun.com/jini	www.upnp.org	www.salutation.org	www.svrloc.org	www.bluetooth.com
Service	Invocation	Java code	XML data	Remote procedure call	URL	N/A
	Status inquiry	Notification, event agent	Polling, notification	Notification	N/A	N/A
Directory	Centralized vs. distributed	Distributed	N/A	Either	Centralized	N/A
	Flat vs. hierarchical	Flat or hierarchical	N/A	Flat	N/A	N/A
	Service state in directories	Soft state	N/A	Hard state with periodically check	Soft state	N/A
Announcement and Lookup	Query vs. announcement	Both	Both	Both	Both	Query
	Directory-based vs Non-Directory-based	Directory-based	Non-Directory-based	Directory-based	Either	Non-Directory-based
	Communication	Unicast and multicast	Unicast and multicast	Unicast and broadcast	Multicast	Unicast and broadcast
Service Selection	User vs. protocol selection	User selection	User selection	User selection	User selection	User selection
	Service matching	Match all	Match all	Match one or match all	Match all	Match all
	Context-aware	N/A	N/A	N/A	N/A	N/A
	Scope-aware	Location, administrative domain	N/A	N/A	Administrative domain	Location/vicinity
	QoS-aware	N/A	N/A	N/A	N/A	N/A

로토콜간의 브리지 역할을 수행하는 기술에 대한 연구가 필요할 것으로 예측된다. 본 논문에서는 주요 서비스 발견 기술을 살펴보고 이들을 비교함으로써 각 서비스 발견 프로토콜의 특성을 분석하였다.

- [1] 박준희, 손영성 외, "홈네트워크 미들웨어 및 표준화 동향," ETRI, 전자통신동향분석 19권 5호, 2004. 10., pp.53-58.
- [2] "JINI Architectural Overview: Technical White Paper," <http://www.sun.com/software/jini/whitepapers/architecture.html>
- [3] "Understanding UPnP: A White Paper," [http://www.upnp.org/download/UPNP\\_UnderstandingUPNP.doc](http://www.upnp.org/download/UPNP_UnderstandingUPNP.doc)
- [4] UPnP Device Architecture v1.0.1 Draft, 2003.
- [5] Salutation Consortium, "Salutation Architecture Specification Version 2.0c Part 1," The Salutation Consortium, June 1, 1999, <http://www.Salutation.org>
- [6] C. Lee and S. Helal, "Protocols for Service Discovery in Dynamic and Mobile Networks," *International Journal of Computer Research*, Vol.11, No.1, 2002, pp.1-12.
- [7] Erik Guttman, "Service Location Protocol: Automatic Discovery of IP Network Services," *IEEE Internet Computing*, 1999.
- [8] E. Guttman, C. Perkins, J. Veizades, and M. Day, "Service Location Protocol, Version 2(RFC 2608)," 1999, <http://www.faqs.org/rfcs/rfc2608.html>
- [9] Maurice Uhlmann and Stephan Rupp, "Service Discovery in Bluetooth and JINI," INFOTECH seminar advanced Communication Services(ACS), 2004.
- [10] "Specification of the Bluetooth System, Version 1.2, Vol.3, Part B: Service Discovery Protocol(SDP)," 2003, <http://www.Salutation.org>
- [11] F. Zhu, M. Mutka, and L. Ni, "Classification of Service Discovery in Pervasive Computing Environments," Institution Michigan State University, MSU-CSE-02-24, 2002.