

웹서비스를 이용한 서비스 기반 디바이스 연동 기술

이원석*, 이강찬*, 전종홍*, 이승윤*

*한국전자통신연구원 표준연구센터

The Technology of Device-to-Device Integration using Web Services

Lee, Won-Suk, Lee, Kang-Chan, Jeon Jong-Hong, Lee, Seung-Yun

Electronics and Telecommunications Research Institute

E-mail : {wslee, chan, hollobit, syl}@etri.re.kr

요 약

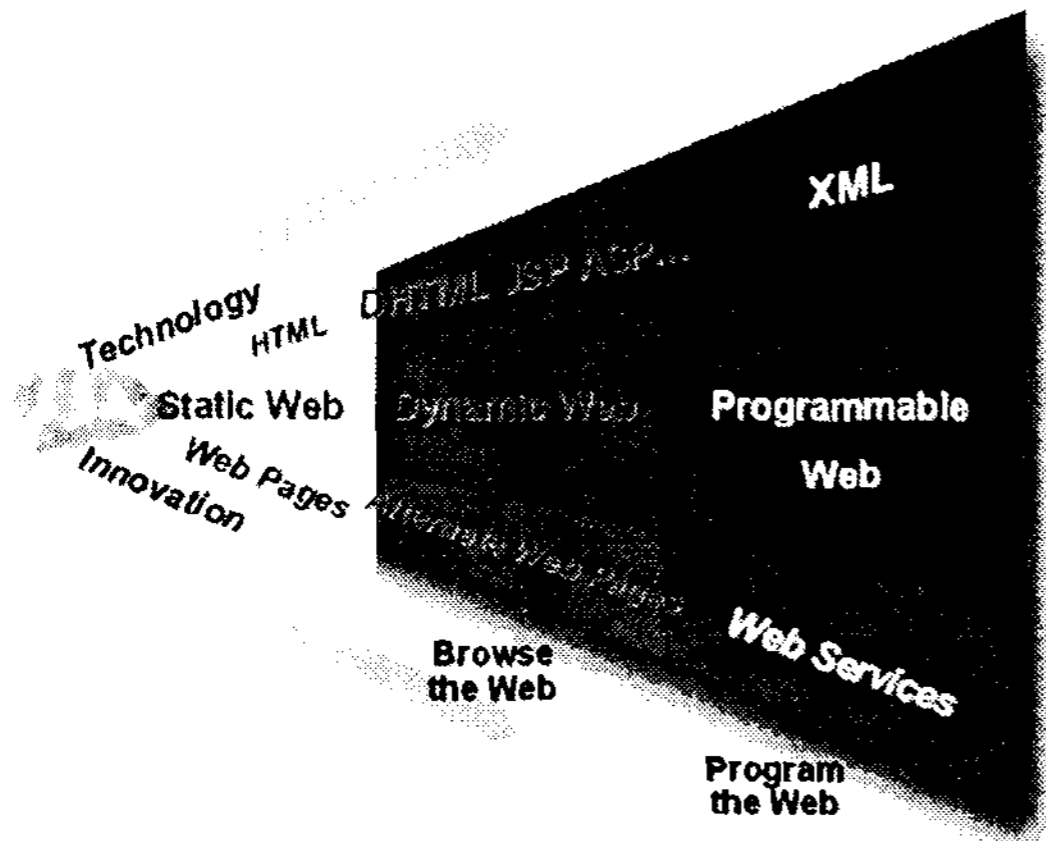
초기의 웹서비스는 인터넷에 존재하는 응용들을 보다 효율적으로 통합할 수 있는 기술로 인식되어, 지리적으로 분산되어 있는 기업 시스템을 통합하기 위한 목적이나 서로 다른 기업 간의 시스템 연동을 위한 목적으로 활용되기 시작하였다. 그러나, 최근에는 웹서비스의 적용이 크게 확산되면서 인터넷의 응용을 넘어 이동통신 망 응용까지 확산되고 있는 추세이다. 이는 웹서비스가 아스키 기반의 XML 표준을 기반으로 하고 있어 플랫폼, 프로그래밍 언어, 통신망에 독립적인 특징을 갖고, 또한 웹의 국제 표준화 기구인 W3C를 중심으로 개발된 표준 기술이기 때문이다. 현재 MS, IBM 등 주요 IT 기업들은 이와 같은 웹서비스의 장점을 이용하여 유비쿼터스 환경에 존재하는 다양한 종류의 디바이스를 연동하기 위한 기술에 집중하고 있다. 본 논문에서는 유비쿼터스 환경에서 웹서비스를 이용하여 디바이스를 연동하는 기술과 디바이스가 기존에 인터넷에 존재하는 웹서비스를 연동하기 위해 필요한 기술에 대하여 설명한다.

1. 서론

웹서비스는 웹상에서 모듈화 된 소프트웨어 컴포넌트로서, 개방형 표준 데이터 표현 기법인 XML과 인터넷 프로토콜을 결합시킨 새로운 패러다임에 의해서 탄생된 차세대 분산 컴퓨팅 기술이다. 특히, 웹서비스의 상호운용성은 다양한 종류의 웹서비스를 동적으로 발견하고 결합함으로써 부가가치를 가진 새로운 형태의 복합 웹서비스의 창출을 가능하게 한다[1]. 이러한 특징이 부각되면서 초기의 웹서비스는 궁극적으로 기업 내, 기업 간은 물론 공공 기관 간의 프로세스 통합 및 협력 자동화를 위한 핵심 기술로 적용되기 시작하였다. 그러

나, 다양한 분야에서 웹서비스의 적용이 점진적으로 이루어지면서, 웹서비스를 유비쿼터스 환경에 적용하기 위한 노력들이 시도되고 있다. 이는 인터넷의 다양한 시스템, 다양한 플랫폼, 다양한 프로그래밍 언어로 개발된 응용들이 산재되어 있는 환경이 유비쿼터스 환경과 유사하여 뛰어난 상호운용성을 지원하는 웹서비스 기술이 적합할 것으로 예상되기 때문이다. 실제로 주요 IT 기업인 마이크로소프트, IBM 등은 웹서비스를 이용하여 다양한 디바이스를 연동하는 기술에 대한 연구를 진행하고 있는 상황이다. 본 논문은 웹서비스를 이용한 디바이스 연동 기술에 대한 전반적인 기술에 대해 설명한다. 2장에서는 웹서비스 개요 및 핵심 표준에 대해서 설명하며, 3장에서는 Ad-hoc 환경에서

의 웹서비스 검색 프로토콜 기술, 디바이스 프리퍼런스 프로파일 교환 기술, IPv4와 IPv6 공존 환경에서의 웹서비스 프로토콜 기술, SOAP 전송 프로토콜 변환 기술 그리고 웹서비스 성능 향상 기술에 대해서 설명하며, 4장에서 결론 및 향후 연구 방향에 대해서 설명한다.



<그림 1> 웹의 진화 방향

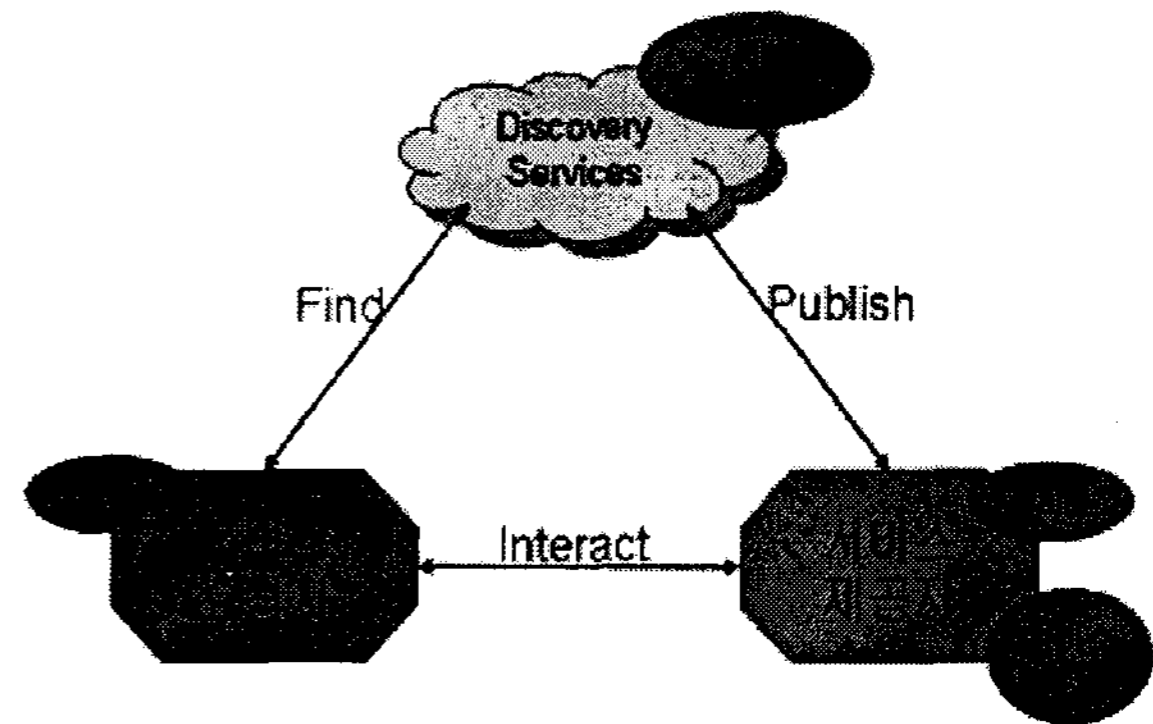
2. 웹서비스 개요 및 핵심 표준

웹서비스는 전통적인 웹과 달리 기계와 기계간의 상호작용을 위한 구조로 되어 있다. 이는 클라이언트 응용과 서버 응용간의 상호작용 형태를 갖으며, 클라이언트는 사용자의 요구사항을 받아서 요구사항을 처리하기 위해 필요한 서비스들을 다시 찾아 이를 처리한 후 결과를 사용자에게 전달할 수 있다. 이러한 형태는 보다 지능적인 웹을 구현할 수 있는 근간을 제공하여, 사용자가 편하게 자신이 원하는 일을 수행할 수 있는 환경을 제공한다.

웹서비스에서 클라이언트와 서버 간에 주고받는 메시지는 XML 형태를 사용하며, 전송프로토콜은 HTTP, SMTP, FTP 등과 같은 다양한 프로토콜 사용이 가능하다[7].

일반적인 웹서비스의 기본 구조는 역할의 관점에서 크게 세 부분으로 나뉘 질 수 있는데 서비스를 제공해 주는 역할을 하는 서비스 제공 엔티티, 서비스를 요청하는 서비스 요청 엔티티, 마지막으로 서비스 제공 엔티티와 서비스 요청 엔티티를 연결시켜 주는 서비스 관리 엔티티가 그것이다. 첫

번째 서비스 제공 엔티티는 웹서비스를 제공해 주는 플랫폼으로서 웹서비스가 실제적으로 운영된다. 다시 말하면, 서비스 요청자에 의해 요청된 서비스를 제공해 주는 역할을 수행한다. 두 번째 서비스 요청 엔티티는 서비스 받기를 원하는 사용자와 웹서비스를 연결하는 인터페이스를 제공하거나 또는 사용자 인터페이스를 제공하지 않는 프로그램 상에서 서비스를 직접 접근할 수 있는 클라이언트 어플리케이션 기능을 제공하는 역할을 하는 소프트웨어가 될 수 있다. 세 번째 서비스 관리 엔티티는 인터넷상에 분산되어 있는 여러 서비스 제공 엔티티들이 제공하는 웹서비스를 서비스 레지스트리(Service Registry)에 등록하고, 이를 공개하여 서비스 요청 엔티티가 서비스 제공 엔티티와 연결할 수 있게 정보를 검색할 수 있는 기능을 제공하는 역할을 한다[7]. <그림 2>는 웹서비스의 기본 구조를 보여준다.



<그림 2> 웹서비스 기본 구조

웹서비스는 인터넷에 존재하는 많은 서비스 제공자들이 있고 이를 서비스 클라이언트들이 자신들이 원하는 서비스를 찾아서 이용하는 구조이므로, 무엇보다도 웹서비스에 관련된 표준이 중요하다. 웹서비스의 핵심표준으로 W3C에서 표준화를 진행 중인 WSDL(Web Service Description Language)과 같은 W3C에서 표준화가 완료된 SOAP(Simple Object Access Protocol), 그리고 OASIS에서 표준화를 진행 중인 UDDI(Universal Description, Discovery, and Integration)가 있다. 이 밖에도 웹서비스의 보안에 대한 표준인 WS-Security, 웹서비스 메시지의 라우팅에 대한 표준인 WS-Routing, 웹서비스의 협업을 위한 표

준인 WS-Coordination 등 웹서비스와 관련된 많은 표준들이 비영리 표준화 기구 및 업체를 중심으로 연구 및 개발되고 있다.

2.1 WSDL

서비스 요청자가 서비스에 접근하기 위해서는 서비스 접근에 대한 정보가 필요한데, 이러한 서비스에 관련된 정보를 기술하기 위해 사용되는 표준이 WSDL이다[8,9]. 다시 말해, WSDL은 서비스가 어디에 존재하며, 무엇을 할 수 있고, 또 이를 실행하기 위해서는 어떻게 해야 하는가를 XML 형식으로 제공하는 메타언어(meta language)라고 할 수 있다.

본질적으로 WSDL 명세는 세 가지 웹 서비스 기본 속성을 기술하고 있다.

- 서비스가 하는 일: 서비스가 제공하는 오퍼레이션
- 서비스에 접근 방법: 서비스 오퍼레이션에 접근하기 위해 필요한 데이터 포맷과 프로토콜에 대한 구체적인 내용
- 서비스가 위치한 곳: URL과 같은 구체적인 네트워크 상의 주소 정보

2.2 SOAP

SOAP이란 분산 환경에서 소프트웨어 또는 서비스들 간에 정보를 교환하기 위한 XML 기반의 간단한 프로토콜이다. SOAP은 RPC 또는 IIOP와 같이 바이너리방식을 사용하는 프로토콜과 달리 텍스트방식의 프로토콜로서 다양한 어플리케이션 간 분산 객체의 사용을 쉽게 이룰 수 있게 해준다. 또한, SOAP은 XML을 사용함으로써 특정 형식의 제약이 없고, 유연성 높은 접근 기능을 제공한다.

2.3 UDDI

UDDI는 비즈니스 조직과 같은 엔티티들이 자신과 제공하는 서비스에 대한 상세 정보를 공개적으로 등록할 수 있도록 한다. 또한 비즈니스, 표준단체, 산업계에서 정의한 서비스 유형 표준과 추상화에 관한 정보들을 등록하고, 식별자를 할당하여 그

것들을 참조할 수 있게 해 준다. 결과적으로 UDDI는 비즈니스 레지스트리와 참조유형 레지스트리를 제공한다고 할 수 있다. UDDI는 두 가지 핵심 기본 데이터 구조로서 비즈니스 엔티티(Business Entity)와 티모델(tModel)을 갖는다[5].

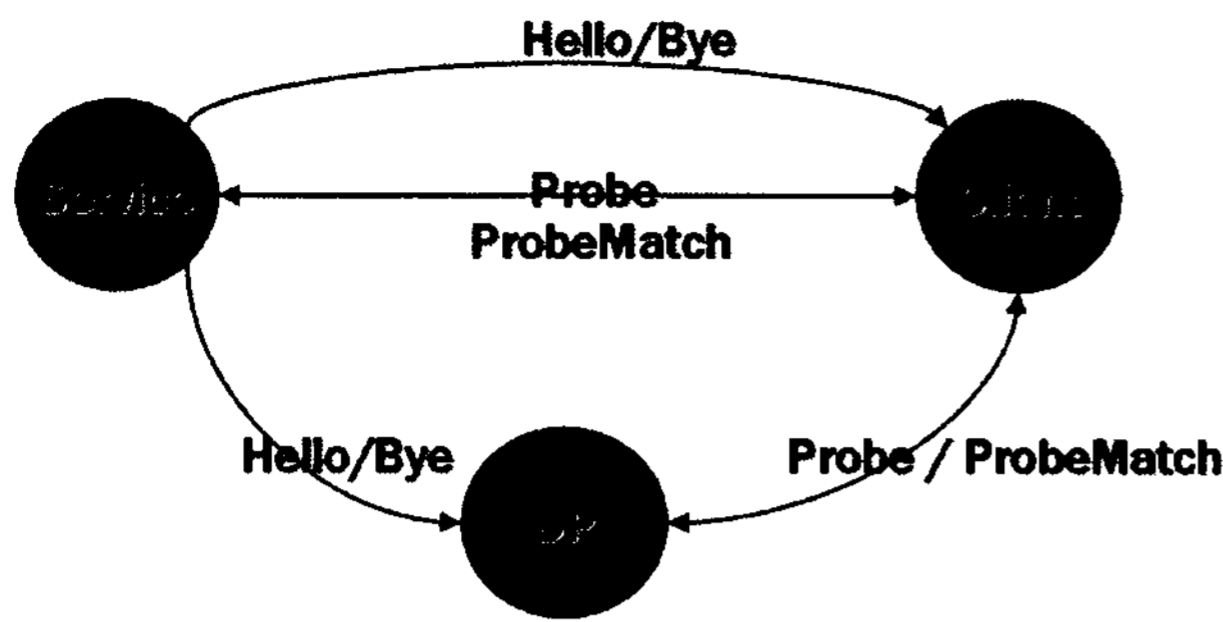
비즈니스 엔티티의 정보는 화이트, 옐로우, 그린 페이지로 나뉜다.

- 화이트 페이지(White Page) : 엔티티에 관한 일반적인 연락정보를 포함한다. 기업의 이름, 주소, 전화, 팩스, 이메일과 같은 정보가 포함될 수 있다.
- 옐로우 페이지(Yellow page) : 엔티티가 제공하는 서비스 유형과 위치에 관한 분류정보를 포함한다. 전화번호부와 같은 역할을 한다고 볼 수 있다.
- 그린 페이지(Green page) : 제공하는 서비스를 이용하는 방법에 대한 상세정보를 포함한다.

3. 웹서비스 기반 디바이스 연동 기술

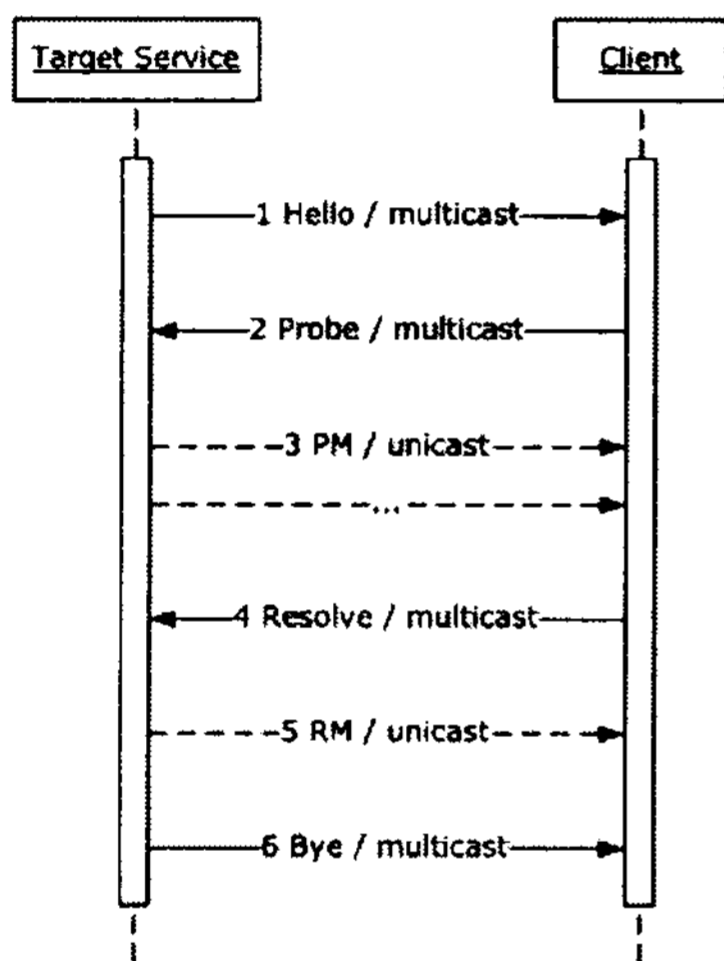
3.1 Ad-hoc 환경에서의 웹서비스 검색 프로토콜 기술

기존의 웹서비스의 검색은 UDDI를 이용하여 수행하는 방법이였으나, 이와 같은 방법은 망 자체가 동적으로 변하는 Ad-hoc 망에서는 적용하기 어려운 방법이다. 따라서, 마이크로소프트에서는 Ad-hoc 환경에서 웹서비스를 효과적으로 찾기 위한 새로운 검색 프로토콜을 설계하고 있다. 이는 WS-Discovery 라는 이름으로 두 번째 드래프트까지 작성되어 있다. WS-Discovery는 Hello, Bye, Probe, Probe Match(PM), Resolve, Resolve Match(RM) 총 6개의 메시지 형태를 갖는다. 서비스가 망에 조인하면 Hello 메시지를 보내서 자신의 존재를 알리고, 그 망을 떠날 때는 Bye 메시지를 보내서 자신이 망에서 사라질 것을 알린다. Probe는 type, scope 등의 조건을 표현하여 원하는 서비스를 찾기 위한 것이다. Resolve는 이름으로 Target 서비스를 검색할 때 사용한다. 아래의 <그림 3>은 WS-Discovery의 구성요소와 간단한 동작 방식을 보여준다. DP는 Discovery Proxy를 의미하며, 선택적인 구성요소이다.



<그림 3> WS-Discovery의 구성요소 및 동작 과정

아래의 <그림 4>는 WS-Discovery에서 클라이언트가 원하는 서비스를 찾는 전체적인 과정의 메시지 교환의 흐름을 보여준다.



<그림 4> WS-Discovery의 메시지 교환 흐름

<그림 6>에서 1은 Target 서비스는 망에 조인할 때, Hello 메시지를 멀티캐스트 형태로 보내고, 클라이언트는 이를 받는다는 것을 의미한다. 2는 또한, Target 서비스는 어느 때든 클라이언트로부터 Probe 멀티캐스트 메시지를 받을 수 있음을 뜻한다. 3은 만일 Target 서비스가 Probe에 매치되면, PM(Probe Match) 유니캐스트 메시지를 클라이언트에게 보낸다는 것을 의미한다. 4는 유사하게 Target 서비스는 어느 때든 클라이언트로부터 Resolve 멀티캐스팅 메시지를 받을 수 있음을 뜻한다. 5는 Target 서비스가 만일 Resolve의 대상이면, RM(Resolve Match) 유니캐스트 메시지를 클라이언트에게 보낸다는 것을 의미한다. 마지막으로 6은 Target 서비스가 망을 떠날 때 Bye 멀티캐스트 메시지를 보내야 하고, 클라이언트는 이를 받

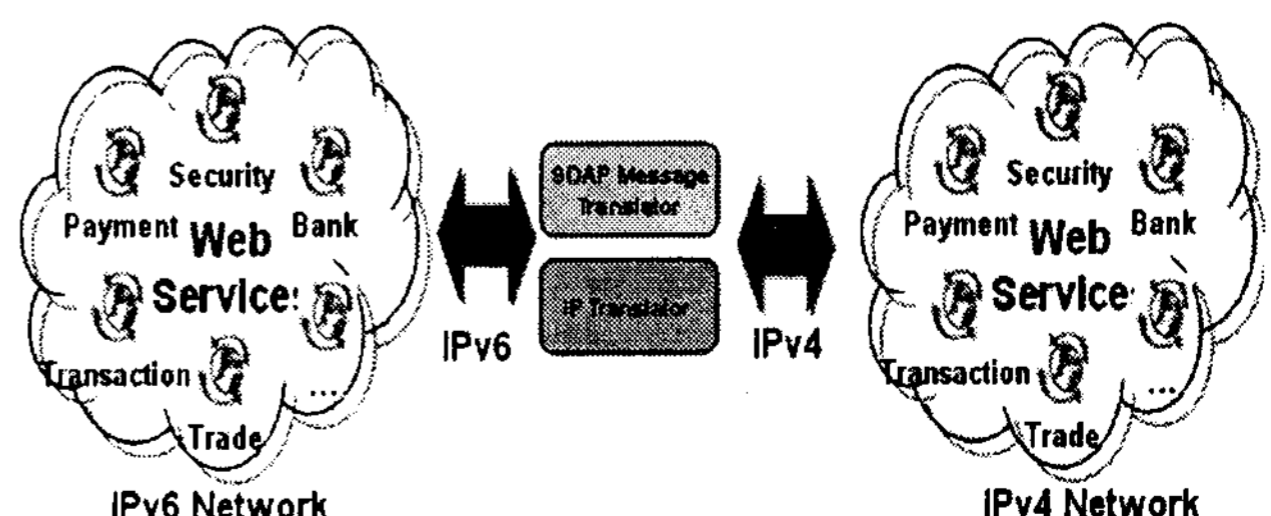
음을 의미한다.

3.2 디바이스 프리퍼런스 프로파일 교환 기술

유비쿼터스 환경에서 디바이스 간의 효율적인 연동을 위해서는 연동할 디바이스의 프리퍼런스 정보가 필요하다. 즉, 홈네트워킹 환경에서 웹서비스를 이용하여 셋탑박스에 저장된 동영상을 PDA에서 보고자 할 경우 PDA의 LCD 크기, 지원하는 동영상 타입 등의 프리퍼런 정보를 셋탑박스에 전달하는 것이 필요하며, 이와 같은 정보는 서비스를 받는 디바이스의 특성을 고려한 서비스를 가능하게 한다. 이 뿐만 아니라 디바이스의 환경에 따라 지속적으로 변경되는 컨텍스트 정보들 교환하기 위한 프리퍼런스 프로파일 교환 기술이 필요하다. 이를 위한 기술로 W3C의 WS-CDL(Web Service Choreography Description Language) 기반의 WS-CPP(Web Service Conversation Preference Profile) 기술을 개발하고 있다. WS-CPP는 XPath를 이용하여 WS-CDL의 위치를 표현하며, 필요한 시점에 프리퍼런스 프로파일 정보를 전달하는 메커니즘을 제공한다. 또한, WS-CDL을 기반으로 미리 전달 가능한 정보, 프로세스 상에서 선택, ACK 메시지의 생략 등에 대한 표현도 가능하여 디바이스 간의 메시지 교환을 최소화하는 것이 가능하다.

3.3 IPv4와 IPv6 공존 환경에서의 웹서비스 프로토콜 기술

유비쿼터스 환경은 기본적으로 IPv6를 기반으로 한다고 생각할 수 있으나, 현실적으로는 상당기간 IPv4와 IPv6가 공존하는 형태로 유지될 것으로 보고 있다. 아래의 <그림 5>은 웹서비스가 IPv6 망과 IPv4 망에 혼재되어 있는 상황을 보여준다.

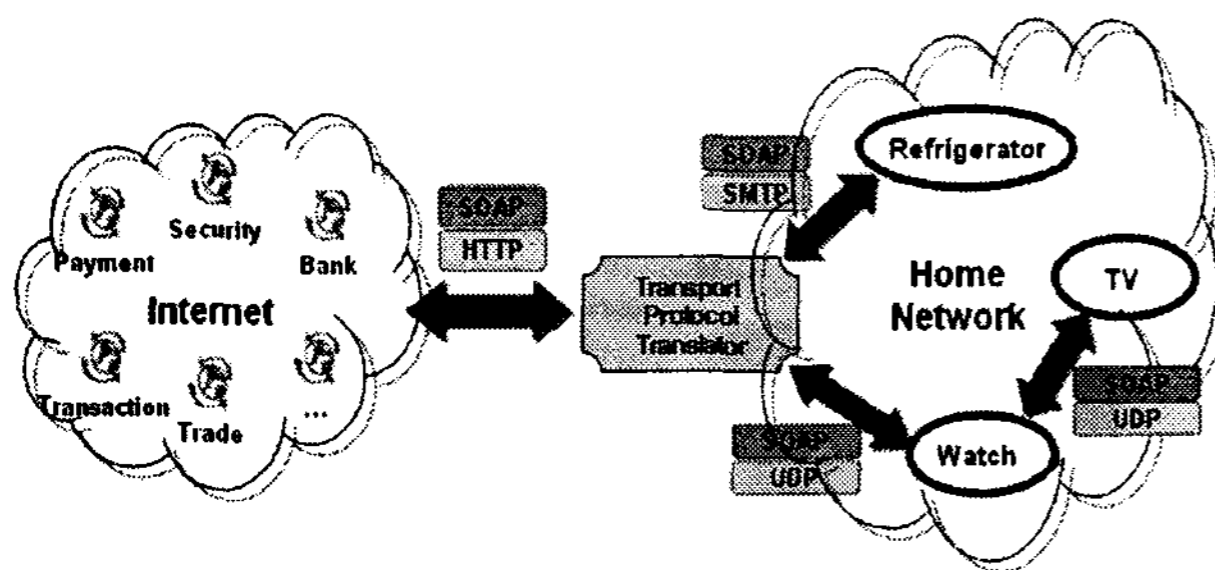


<그림 5> IPv4 와 IPv6 간 주소 변환

공존하는 환경을 만들기 위해서는 IPv6 망과 IPv4 망에 각각 존재하는 응용들이 서로 통신 가능하도록 지원하는 전환 메커니즘이 필요하며, 이는 크게 IPv6/IPv4 이중스택(Dual Stack), 터널(Tunnel), IPv6/IPv4 변환(Translation) 기술로 나눌 수 있다. 그러나, 이러한 방법들은 기존 IPv4 환경에서 동작하던 응용들이 IPv6/IPv4 공존환경에서는 완벽하게 동작할 수 있는 환경을 제공하지 못하고 있다. 따라서, IPv6/IPv4 공존 환경에서 웹 서비스를 활용하기 위해서는 전환 메커니즘에 대한 고려와 응용의 특성에 대한 고려를 통한 개발이 필요하다. 예를 들어, 웹서비스의 SOAP은 헤더 정보에 대한 표현이 옵션으로 되어 있지만, 보다 지능적인 서비스를 위해서는 헤더 정보를 활용해야 한다. IPv4와 IPv6 망이 혼재되어 있는 경우, SOAP 헤더에 표현되는 주소에 대한 매핑 처리가 필요하다.

3.4 SOAP 전송 프로토콜 변환 기술

SOAP은 HTTP 뿐 아니라 SMTP, FTP 등의 다양한 프로토콜 전송을 위해 사용할 수 있다. 유비쿼터스 환경은 다양한 디바이스, 플랫폼 등이 존재할 수 있고, 또한 디바이스가 지원하는 프로토콜의 종류가 디바이스 별로 각기 상이할 수 있다. 따라서, 이러한 경우에 이런 상이한 전송 프로토콜을 맞추어 주는 기능을 하는 개체가 필요하며, 이 개체는 가능한 모든 전송 프로토콜을 지원해야 한다. 또한, 이 개체는 SOAP 메시지를 주고받는 대상이 지원하는 프로토콜을 확인하여 필요에 따라 적절한 프로토콜로 바꾸어 전송해 주어야 한다. 아래의 <그림 6>는 SOAP의 전송 프로토콜 변환에 대한 설명이다.

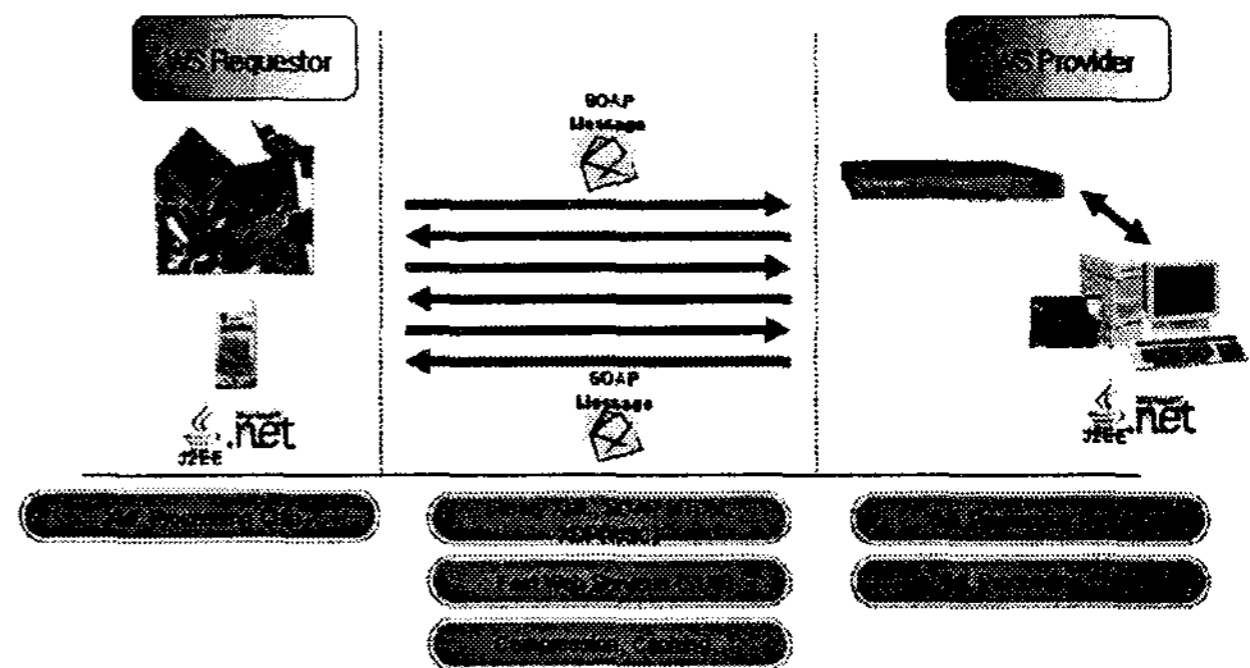


<그림 6> SOAP 전송 프로토콜 변환

현재 마이크로소프트는 SOAP Over UDP[4]에 대한 규격을 개발하고 있는데, 이는 Ad-hoc 환경에서 SOAP 메시지를 멀티캐스팅을 이용해서 전송할 필요성을 제기하고, 이를 구현하기 위해 UDP를 전송 프로토콜로 사용하는 방법에 대한 내용이다. UDP를 사용하는 경우 기본적으로 멀티캐스팅을 지원하므로, 전송에 대한 확인이 필요 없어 속도가 빠르다는 장점을 갖는다.

3.5 웹서비스 성능 향상 기술

웹서비스의 치명적인 약점으로 지적되는 문제 중의 하나가 성능에 대한 문제이며, 모바일 디바이스인 경우 컴퓨팅 파워가 PC에 비해 상대적으로 더욱 취약하다. 또한 무선 통신망 또한 유선망에 비하여 속도와 안정성이 크게 떨어진다는 점과 웹서비스가 XML 기반의 SOAP을 메시지 표준으로 사용하므로 기존의 다른 프로토콜에 비해 교환하는 메시지 크기가 크다는 문제가 있다.



<그림 7> 웹서비스에서 부분별 성능 개선 방법

웹서비스의 성능 개선 부분은 <그림 7>과 같이 크게 세 부분으로 볼 수 있다. 첫 번째는 웹서비스 요청자 부분에서 성능을 개선하는 것으로, 웹서비스 요청자가 모바일 디바이스인 경우에는 XML 처리 칩 등을 통해서 성능 개선이 가능하다. 두 번째는 메시지 전송에서 성능을 개선하는 것으로 테스트 기반의 SOAP 메시지를 기존의 상호운용성을 유지하면서 바이너리로 표현하여 메시지의 크기를 줄이는 방법이다. 이와 같은 방법으로 W3C의 Binary XML, SOAP MTOM(Message Transmission Optimization Mechanism), XOP(XML-binary Optimized Packaging), SUN의

Fast Web Services에서 사용하는 ASN.1 기반의 표현 방법이 있다. 또한 압축, 캐싱 등의 다양한 방법들이 이 부분에서 연구되고 있다. 세 번째는 웹서비스 제공자 부분에서 성능을 개선하는 것으로, 웹서비스 제공자가 컴퓨팅 파워가 약한 소형 디바이스인 경우에는 첫 번째 경우와 같이 XML 처리 칩을 활용하는 방법이 있고, 인터넷에서 웹서비스를 제공하는 전용 서버인 경우에는 XML 처리 전용 장비나 XML 전용 처리 카드를 이용하여 성능을 개선할 수 있다.

4. 결론 및 향후 연구 방향

본 논문은 기존의 웹서비스 기술이 유비쿼터스 환경에서 모든 디바이스들을 통합하는 핵심적인 기술로 발전하면서, 최근 연구되고 있는 웹서비스를 이용한 디바이스 연동 기술에 대해서 설명하였다.

핵심적인 내용으로 Ad-hoc 환경에서의 동적으로 서비스를 검색할 수 있는 웹서비스 검색 프로토콜 기술, 디바이스 간에 효율적인 서비스 제공을 위한 디바이스 프리퍼런스 프로파일 교환 기술, IPv4와 IPv6 공존 환경에서의 웹서비스 프로토콜 기술, 다양한 디바이스들이 서로 다른 전송 프로토콜을 지원하는 경우에 고려해야 하는 SOAP 전송 프로토콜 변환 기술 그리고 웹서비스 성능 향상 기술에 대해서 알아보았다.

웹서비스 기술은 앞으로 유비쿼터스 환경에서도 핵심적인 통합 기술로 자리 잡을 것으로 예상되므로, 앞으로 이를 위한 다양한 기술 개발과 표준 개발이 필요할 것으로 예상된다.

[참고문헌]

[1] 이강찬, 이승윤, "웹서비스 표준기술 동향과 전망", 정보과학회지, 2004.10.
 [2] Alessandro Forin, Johannes Helander, Paul Pham, Jagadeeswaran Rajendiran, "Component Based Invisible Computing", IEEE Real-time Embedded System Workshop, Dec 3, 2001.
 [3] Microsoft, "Web Services Dynamic Discovery (WS-Discovery)", October, 2003.

[4] Microsoft, "SOAP over UDP", September, 2004.
 [5] Tom Bellwood, "Understanding UDDI: Tracking the Evolving Specification", IBM developerWorks Web Services zone, July 2002.
 [6] Vinay Bansal and Angela Dalton, "A Performance Analysis of Web Services on wireless PDAs," Duke University Computer Science, 2003.
 [7] W3C, "Web Services Architecture, W3C Working Draft", May 14, 2003.
 [8] W3C, "Web Services Description Language (WSDL) 1.1, W3C Note", March 15, 2001.
 [9] W3C, "Web Services Description Language(WSDL) Version 1.2", W3C Working Draft, March 3, 2003.
 [10] Yong Xiong, Johannes Helander, Alessandro Forin, Gideon Yuval, "Secure Invisible Computing", Microsoft Technical Report, October 7, 2003.