

WAP 게이트웨이 설계 및 구현

김수진, 정동열, 이용학*, 전서현, 김승구**
동국대학교 컴퓨터공학과
*김포대학 컴퓨터 계열 인터넷 정보 전공
**(주)아이넥션

Design and Implementation of WAP Gateway

Soo-Jin Kim, Dong-Youl Jeong, Yong-Hack Lee*, Seo-Hyun Cheon, Soong-koo Kim**
Dept. of Computer Engineering, Dongguk University

*Dept. of Internet Computer, internet information major, Kimpo College

**Injection co., Ltd

요 약

최근 이동 전화 단말기를 이용한 무선 통신 인구가 급증하고 있고, 이들은 무선 통신을 이용한 다양한 서비스를 요구하고 있다. 이런 서비스들 중 기존의 월드 와이드 웹과의 연결을 통한 서비스들도 요구되고 있다. 이러한 서비스들을 제공하기 위해 여러 업체들이 포럼을 구성하여 무선 통신 망과 기존의 인터넷망을 연결하는 프로토콜인 Wireless Application Protocol(WAP)을 만들었다. 이 프로토콜을 이용하여 서비스를 제공하는 WAP 서버와 무선 통신 단말기와의 연결하는데 있어 이를 담당하는 게이트웨이가 필요하다. 본 논문에서 WAP 프로토콜을 이용하여 이동 통신망과 인터넷망을 이용한 WAP 서버 사이에 브릿지 역할을 하는 게이트웨이를 설계하고 구현했다.

서론

사용자는 월드 와이드 웹(WWW : World Wide Web)의 기술의 발달로 인해서 언제 어디서 컴퓨터를 사용하여 수많은 정보를 검색하고 획득할 수 있게 되었다. 이로 인하여 월드 와이드 웹이라는 것은 일상 생활의 공간으로 자리 잡아가고 있다. 월드 와이드 웹을 통해서 실생활에서 필요한 물건을 구매하고, 문화 활동을 위한 티켓을 예약하고, 필요한 자료를 검색하고, 많은 사람들과 대화를 하기도 한다.

최근에 이동 전화 단말기를 이용한 무선 통신 인구가 급증하고 있고, 이들은 무선 통신을 이용한 다양한 서비스를 요구하고 있다. 이를 위해 무선 통신 업체들은 기존의 통화 서비스 외에 월드 와이드 웹과 연계한 다양한 서비스를 제공하는 추세이다.

휴대폰 단말기를 이용한 각종 데이터 송·수신 및 인터넷 검색은 무선 통신 기술을 기반으로 한다. 일반 휴대폰 단말기는 TCP/IP(Transmission Control Protocol / Internet Protocol) 기반의 인터넷을 바로

접속하여 검색할 수 없기 때문에 TCP/IP와 무선 통신망과의 브릿지 역할을 하는 게이트웨이나 서버가 필요하다. 현재 이러한 브릿지를 위한 프로토콜중 국제 표준에 가장 근접한 것이 WAP이다. WAP 게이트웨이와 WAP 서버는 내부적으로 5계층으로 구성된 WAP 프로토콜과 TCP/IP 기반의 HTML 문서와 WAP 프로토콜 기반의 WML(Wireless Markup Language) 문서간의 변환기 역할을 하는 필터(Filter)로 구성된다[1][2].

본 논문에서는 무선 통신과 인터넷을 연결하기 위한 WAP 프로토콜에 대한 연구와 WAP 프로토콜을 사용한 WAP 게이트웨이의 구현을 기술하였다.

본 논문의 구성은 다음과 같다. 먼저 2 장에서는 WAP에 대한 정의 및 구조를 간략히 기술하였고 3 장에서는 본 논문에서 설계한 WAP 게이트웨이의 내부 구성요소를 설명하며 4 장에서는 WAP 게이트웨이의 설계 및 구현에 관해 설명한다. 마지막으로 5 장의 결론에서는 향후의 연구과제를 제시한다.

2 WAP 정의 및 구조

WAP이란 디지털 이동 전화기, 페이저(Pager), 휴대형 개인정보단말기(PDA : Personal Digital Assistant) 그리고 그 외 무선 단말기들에게 인터넷 통신과 진보된 전화 서비스를 제공하는 프로토콜을 말한다. 이런 서비스를 제공하기 위해서는 기존 월드 와이드 웹 모델과는 다른 형태의 모델이 필요하다. WAP 모델은 그림 1 과 같다.

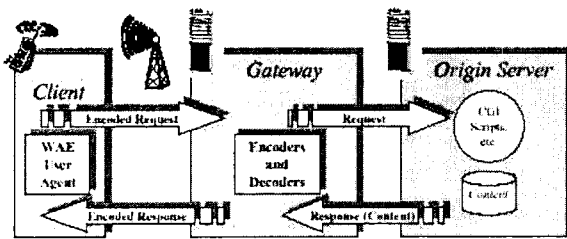


그림 1. WAP 모델

WAE 사용자 에이전트(User Agent)는 사용자에게 특별한 기능을 제공하는 클라이언트 쪽의 소프트웨어이며, 이것은 월드 와이드 웹 모델에서의 웹 브라우저와 같은 역할을 수행한다. WAP 게이트웨이는 프로토콜 변환 작업을 수행하며, 클라이언트와 웹 서버 중간에서 WML을 처리(encoder & decoder) 수행한다. 오리진 서버(Origin Server)는 월드 와이드 웹 모델에서의 웹 서버와 같은 역할을 수행한다[2][3].

전체 네트워크 형태로 살펴보면 2가지의 통신 방법이 가능하다.

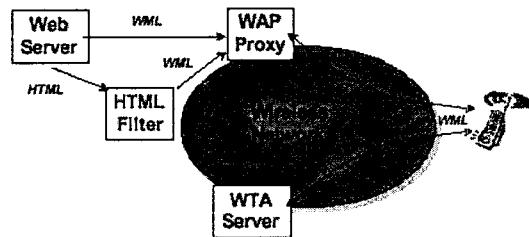


그림 2. WAP 네트워크

첫 번째 방법은 일단 이동 전화 단말기는 WAP 프락시에 요청(Request) 메시지를 전송하면, 웹 서버는 해당 응답(Response) 메시지를 전송한다. 이때 웹 서버가 WML을 지원하면 직접 WAP 프락시로 전송한

다. 만약에 웹 서버가 WML을 지원하지 않는 경우에는 HTML 필터링을 통해 WML으로 변환 후, WAP 프락시에게 전송한다.

두 번째 방법은 무선 전화기에서 WAP 프락시를 거치지 않고, 바로 WTA 서버를 통해서 서비스를 받을 수 있다.

이렇게 서로간의 요청과 응답을 주고받기 위해서는 WAP의 아키텍처에 맞춰서 행해져야 한다. WAP는 6개의 레이어 계층으로 구성되어 있으며, 각 계층은 각각 WAE, WSP, WTP, WTLS, 그리고 WDP이다. WAP의 아키텍처는 그림 3 과 같다.

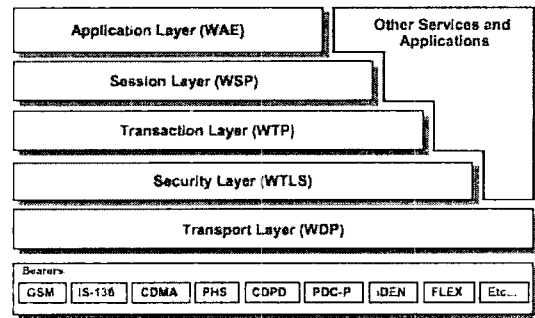


그림 3. WAP 계층도

이런 WAP은 다음과 같은 특징을 가지고 있다.

- 월드 와이드 웹 모델과 유사한 프로그래밍 모델
기존 월드 와이드 웹의 기본 지식이나 기반 기술들을 WAP에 쉽게 적용하여 사용할 수 있다. 그럼으로써 보다 쉽고 빠르게 WAP 서비스를 개발할 수 있다. 또한 인터넷 기술을 재사용 한다는 측면에서 매우 경제적이다.

- WML

웹에서 사용되는 HTML(HyperText Markup Language)의 목적과 동일하게 무선 환경에서 서비스의 저작을 위해 사용되는 언어이다. HTML과 다른점이라면, 작은 핸드헬드 장치(Handheld Device)에 적합하다는 것이다.

- WMLScript

WMLScript는HTML에서 자바스크립트(JavaScript)의 사용 목적과 동일하게 서비스의 기능을 강화하는데 사용한다.

· 무선 전화 어플리케이션(WTA)

WTA 프레임워크는 텔레포니 서비스를 생성하는데 필요한 수단을 제공한다. 예를 들면, 사용자 인터페이스, 네트워크 이벤트 처리기, 서비스의 실시간 처리를 위한 저장소(Repository), 그리고 서버 초기화 서비스 등이 있다[15].

· 최적화된 프로토콜 스택(Optimized protocol stack)

WAP은 HTTP(HyperText Transmission Protocol)와 TCP와 같은 인터넷 프로토콜에 기초를 한다. 이것은 무선 환경의 제약 사항(예를 들면, 좁은 대역폭과 높은 대기 시간)을 처리하는데 최적화 되어 있다.

3 게이트웨이 내부 구성

다음은 게이트웨이의 내부 주요 컴포넌트와 그들의 상호작용을 나타낸다.

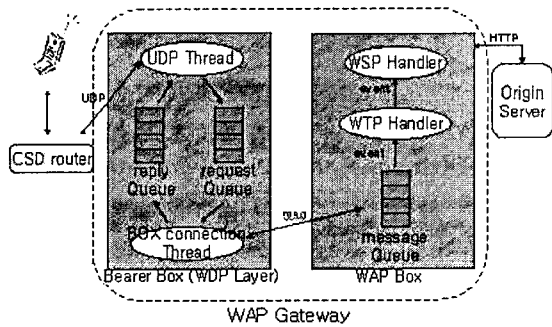


그림 4 WAP 게이트웨이 구성도

본 논문에서 제안하는 게이트웨이는 두 가지 다른 타입의 호스트로 프로세싱 로드가 나뉜다.

· 베어러 박스 (Bearer Box) : 이 호스트는 CSD 라우터와 연결되며 다른 호스트들에게 일관된 인터페이스를 제공한다. 베어러 박스는 WAP 스택의 WDP 레이어를 구현한 것이다.

· WAP 박스 (WAP Box) : WAP 박스는 WAP 스택의 상위 레이어를 실행한다. 해당 세션에 속한 각 세션과 트랜잭션은 같은 WAP 박스에서 처리된다. WAP 박스간 세션과 트랜잭션은 전이되지 않는다.

베어러 박스는 CSD 라우터에서 오는 UDP 패킷을 받고 패킷들을 해당 WAP 박스로 보낸다.

WAP 박스는 WTP와 WSP 레이어를 구현한 부분이다. 이들은 단말기에서 오는 HTTP와 같은 요청들

을 받아 콘텐츠 서버들에게로의 실질 HTTP 요청을 만들고 응답을 압축하고 해당 단말기로 되돌려 보낸다.

4 설계 및 구현

◆ 베어러박스(Bearer Box)

베어러박스는 CSD 라우터와 연결되어 있으며 적합한 포트에서 UDP 패킷을 받아들인다. 베어러박스는 받아들인 패킷을 적절한 WAP 박스로 보내며 보내지 못한 패킷들은 큐에 넣는다.

베어러박스는 몇가지 스레드와 리스트들로 구현되어진다. 구성부분들을 모듈화하여 캡슐화 함으로써 수정과 보수를 용이하게 한다.

· UDP sender/receiver 모듈 : UDP 패킷을 통해 전송된 모든 WDP 패킷들을 받아들이거나 전송한다.

· WAPbox connection 모듈 : WAPbox들에 대해 서버역할을 하며 WAP box들이 연결하고 메시지를 송,수신할 수 있도록 한다.

· HTTP administration 모듈 : 유저 인터페이스 역할로서 게이트웨이의 지연, 재개, 중지 등의 명령을 처리한다.

◆ WAP 박스 (WAP Box)

각각의 트랜잭션과 세션은 해당 레이어마다 상태머신(state machine)으로 구현된 자료구조에 의해 표현된다. 각 새로운 WDP 패킷은 WAP 박스에서 새로운 스레드에 의해 처리된다. 스레드는 알맞은 WTP 상태머신을 찾거나 새로운 상태머신을 만든다. 만일 상태머신이 세션을 다룰 필요가 없다면 WSP 상태머신을 찾거나 만들기 위해 WSP 엔진을 호출하고 해당 상태머신에게 적절한 이벤트를 전해준다. 이는 그 WDP 패킷을 다루기 위해 원래 만들어진 같은 스레드에 의해 모두 처리된다. 마찬가지로 만일 WSP 상태머신이 WTP 상태머신에게 이벤트를 줄 필요가 있다면 그것은 같은 스레드에 의해 처리된다. 일단 WDP 패킷에 의해 생성된 모든 이벤트들이 처리되면 해당 스레드는 소멸된다.

WAP 박스내의 WTP 와 WSP 스택의 프로세스 처리 알고리즘은 다음과 같다.

· WTP process algorithm

```

-read message from bearer box
-convert it into a WTPEvent
-find for create the appropriate WTPMachine
-feed the WTPEvent to the WTPMachine
-WTPMachine handles the event according to
the state tables in the WAP spec.
-WTPMachine sends events to WSP
    
```

알고리즘 1. WTP stack process

· WSP process algorithm

```

get WSPEvent from WTP
find or create the appropriate
WSPSessionMachine or WSPMethodMachine
feed the WSPEvent to the state machine
the state machine handles the event according to
state tables in WAP spec.
    
```

알고리즘 2. WSP stack process

5 결론

본 논문에서 언급한 WAP 게이트웨이는 WAP 포럼에서 제안한 WAP 프로토콜을 이용하여 Windows NT Server 4.0 상에서 C++로 구현하였다. 이로써 앞으로 늘어날 무선 통신 및 무선 인터넷의 변화 및 요구에 대응을 할 수 있게 되었다. 또한 기존의 웹 서버 뿐만 아니라 WAP 서버를 구축함으로써 보다 원활한 서비스를 제공할 수 있을 뿐만 아니라 보다 강화된 기능을 무선 인터넷에서 제공하기 위해 기존의 인터넷 기술이 대부분 무선 인터넷으로 이동해 가리라는 예측도 할 수 있다.

본 논문에서는 아직 WAP 프로토콜 상에서 WTLS 스택은 구현하지 않았다. 이는 앞으로 더 연구되어야 할 과제이다. 또한 배어러박스의 병목현상 방지를 위한 로드밸런스(Load Balance)도 연구되어야 할 것이다. 그럼으로써 보다 더 효율적인 WAP 게이트웨이를 기대할 수 있을 것이다.

[참고문헌]

- [1] [WAP] "Wireless Application Protocol Architecture Specification", WAP Forum, April 30, 1998. URL : <http://www.wapforum.org>
- [2] [WAEOver] "Wireless Application Environment Overview", WAP Forum, June 16, 1999. URL

- : <http://www.wapforum.org>
- [3] [WAE] "Wireless Application Environment Specification", WAP Forum, May 24, 1999. URL : <http://www.wapforum.org>
- [4] [WML] "Wireless Markup Language Specification", WAP Forum, June 16, 1999. URL : <http://www.wapforum.org>
- [5] [WMLScript] "WMLScript Language Specification", WAP Forum, June 17, 1999. URL : <http://www.wapforum.org>
- [6] [WMLScriptLib] "WMLScript Standard Libraries Specification", WAP Forum, June 17, 1999. URL : <http://www.wapforum.org>
- [7] [WMLIntent] "WMLScript Statement of Intent", WAP Forum, April 30, 1998. URL : <http://www.wapforum.org>
- [8] [WAPCachingMod] "WAP Caching Model Specification", WAP Forum, February 11, 1999. URL : <http://www.wapforum.org>
- [9] [WSP] "Wireless Session Protocol Specification", WAP Forum, May 28, 1999. URL : <http://www.wapforum.org>
- [10] [WTP] "Wireless Transaction Protocol Specification", WAP Forum, June 11, 1999. URL : <http://www.wapforum.org>
- [11] [WDP] "Wireless Datagram Protocol Specification", WAP Forum, May 14, 1999. URL : <http://www.wapforum.org>
- [12] [WCMP] "Wireless Control Message Protocol Specification", WAP Forum, May 14, 1999. URL : <http://www.wapforum.org>
- [13] [WTA] "Wireless Telephony Application Specification", WAP Forum, July 16, 1999. URL : <http://www.wapforum.org>
- [14] [WTAI] "Wireless Telephony Application Interface Specification", WAP Forum, May 31, 1999. URL : <http://www.wapforum.org>
- [15] AU-System Radio, WAP White Paper <http://www.ausystem.com>