

KANNEL WAP게이트웨이와 연동하는 WAP브라우저의 설계 및 구현

진민식*, 황숙철*, 임동기*, 최원호*, 정민수*

*경남대학교 컴퓨터공학과

Design and Implementation of WAP Browser

Interworking Kannel WAP Gateway

Min-Sik Jin*, Ook-Chul Hwang*, Dong-Ki Lim*, Won-Ho Choi*, Min-Soo Jung*

*Dept. of Computer Engineering, Kyungnam University.

E-mail : comsta6@hawk.com.kyungnam.ac.kr

요약

요즘 작고 빠른 장치에 관한 새로운 열풍이 불고 있다. 이들은 이동 중에도 정보에 접근할 수 있고, 웹과 이동 전화기의 개념을 합쳐 놓은 장치라고 할 수 있다. 무선 애플리케이션 프로토콜(Wireless Application Protocol:WAP)은 산업 전반적인 표준으로서 인터넷 컨텐츠와 서비스를 이동 전화기에서 접속할 수 있도록 해주는 통신 프로토콜과 애플리케이션 환경을 정의한다.

본 논문에서는 오픈소스로 진행되고 있는 Kannel Wap Gateway를 이용하는 WAP브라우저를 설계 및 구현 하였다. 임베디드 리눅스가 탑재된 모든 단말기에서 동작할 수 있으며 무선 인터넷을 통해 WAP사이트에 접속하고 컨텐츠를 디스플레이 할 수 있도록 해준다.

1. 서론

월드와이드웹(WWW)으로 상징되는 인터넷은 이미 사회, 경제, 정치적인 측면에서 인류의 삶에 새로운 패러다임의 변화를 촉진하고 있다. 이동 통신 기술의 발달과 더불어 시공간을 초월한 인터넷 접근이라는 모토아래 수많은 기업들이 무선 인터넷이라는 새로운 기술과 기회에 대한 도전에 직면해 있다.

이러한 기술의 한 예로 모바일 인터넷 환경과 모바일 단말기는 현재의 데스크톱 기준으로 접근하기에는 아직 전력 소모량, 메모리 크기, 디스플레이 크

기, 전송 속도, 안정성 등에서 많은 어려움이 있다. 따라서 유선 인터넷의 표준을 그대로 사용하는데 한계가 있으므로, 기존 표준을 가능하면 따르면서 무선 환경에 적합한 프로토콜을 만들려는 움직임이 일어나게 되었다. 이러한 무선 인터넷의 제약사항과 시대적 조류에 편승하여 많은 솔루션들이 시장에 출현하였으며 대표적인 것으로 마이크로소프트사의 mHTML, Phone.com사의 HDML, 일본의 NTT DoCoMo사의 cHTML, WAP포럼의 WML등이 있다. 이 중 WAP 포럼의 WAP(Wireless Application Protocol)표준은 전 세계의 폭넓은 지지를 등에 염고 무선 인터넷 분야의

강력한 후보로 대두되고 있고 WAP을 사용하기 위해서는 WAP프로토콜 스택을 포함한 단말기와 WAP프로토콜 스택을 포함하고 일반 유선 인터넷을 액세스할 수 있는 WAP게이트웨이 그리고 WAP프로토콜이 사용하는 마크업언어인 WML로 제작된 인터넷 사이트가 있어야 한다.

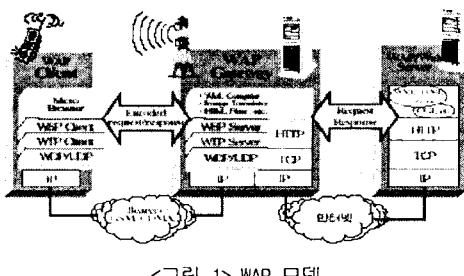
본 논문에서는 WAP클라인언트라고 할 수 있는 WAP게이트웨이와 통신하는 단말기 내에 탑재되는 WAP브라우저를 설계하고 구현한다.

2. 관련연구

2.1 WAP(Wireless Application Protocol)

무선단말 환경은 휴대 단말의 크기, 컴퓨팅 능력, 제한된 입출력 장치로 인한 제약만이 아니라 무선망의 낮은 대역폭, 데이터 전송 지연과 불안정한 접속 등 다양한 문제점들이 있다. WAP은 이러한 문제점을 완화하면서 안정된 무선 인터넷 서비스를 제공하기 위하여 다섯 가지 요구사항 즉, 상호운용성(interoperability), 확장성(scalability), 효율성(efficiency), 신뢰성(reliability), 보안(security)을 고려하여 설계되었다.

<그림 1>은 WAP에서 제안한 무선 인터넷 서비스 모델이다.



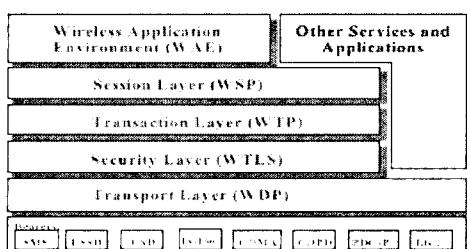
<그림 1> WAP 모델

WAP모델은 무선통신 환경의 문제점들을 극복하기 위하여 유선망에서는 기존의 무선 프로토콜을 그대로 사용하고 무선망에서는 무선환경에 적합하도록 설계된 프로토콜을 사용하는 연결분리(split conn-

ection)기법을 채택하고 있다. 그러므로, 이러한 분리된 연결을 동적으로 연동하기 위하여 WAP게이트웨이(gateway)가 필요하며, 클라이언트는 항상 게이트웨이를 통하여 인터넷으로 연결된다.

2.2 WAP스택과 WAP게이트웨이

<그림 2>는 WAP 프로토콜 스택을 계층별로 나타내고 있다.



<그림 2> WAP프로토콜 스택

WAE레이어는 응용 환경에 대한 규격을 정의하고 소규모 브라우저를 개발 할 수 있도록 하는 레이어이고 WSP레이어는 HTTP1.1에 상응하는 기능과 세션을 정의하고 관리하는 레이어이고 WTP레이어는 트랜잭션형태의 데이터 전송을 담당하는 레이어이다. 또, WTLS는 SSL과 상응하고 WAP스택의 제일 하단에 있는 WDP레이어가 인터넷의 UDP와 같이 실질적인 데이터를 전송하는 역할을 수행한다.

사용자가 단말기를 통해 WAP서비스를 요청하면 사용자 요구는 무선망을 통하여 WAP게이트웨이에 전달되며 WAP게이트웨이는 두개의 다른 망(WAP과 유선망)을 적절히 변형(Convert)하여 사용자의 요구를 전송한다. 즉. 사용자가 WAP서비스를 요청할 때마다 발생하는 일련의 액션은 다음과 같다.

Step1: 사용자의 무선 단말기와 무선망에 위치한 WAP게이트웨이 사이에 접속이 이루어 진다. 이 두 사이의 접속은 WAP프로토콜중의 WSP에 의해 이루어진다.

Step2: WSP로 접속이 이루어진 상태에서 사용자는 특정 WAP사이트로의 접속을 WAP게이트웨이에 요청한다.

Step3: WAP게이트웨이는 무선 단말기로부터 전달된 사용자의 WAP사이트 접속 요청(Request)을

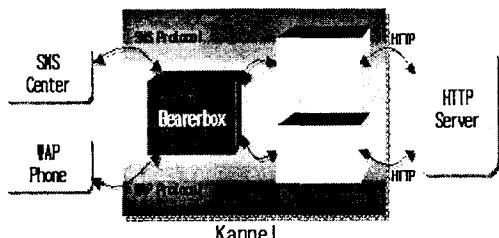
HTTP요청(Request)으로 바꾸어 해당 WAP사이트로 전송한다.

Step4: 해당 WAP사이트는 WAP게이트웨이로부터 받은 HTTP요청에 대한 응답(Response)을 HTTP프로토콜을 이용하여 WAP게이트웨이로 보낸다.

Step5: WAP게이트웨이는 WAP사이트로부터 HTTP응답(Response)을 WAP응답을 바꾸고 데이터를 인코딩(Encoding)하여 사용자의 무선 단말기로 전송한다.

Step6: WAP클라이언트를 가지고 있는 무선 단말기는 WAP게이트웨이로부터 받은 WAP응답을 해석하고 수신한 데이터를 디코딩(Decoding)하여 사용자의 무선 단말기 화면에 출력한다.

본 연구에서는 무선 인터넷(WAP)과 유선 인터넷을 연결시켜 주는 역할을 하는 게이트웨이로서 Kannel WAP게이트웨이를 사용하였다.



<그림 3> Kannel WAP게이트웨이의 구성

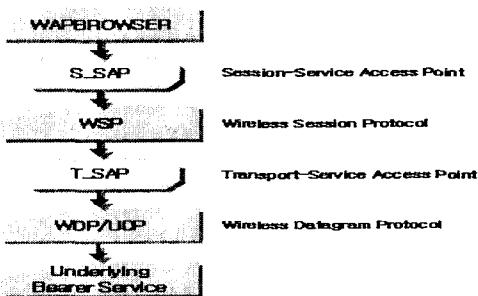
3. 이동 단말기용 WAP브라우저 설계

3.1 WAP스택 클라이언트의 설계

WAP클라이언트를 구성함에 있어 WAPClient 가지고 있는 모든 프로토콜을 전부 구현해야만 WAP클라이언트로서 사용이 가능한 것은 아니며 사용할 무선 네트워크 베어러의 특징 및 WAP애플리케이션의 사용목적에 따라 다양한 구성이 가능하다.

본 연구에서는 Connectionless 모드라고 할 수 있는 비접속 지향 방식의 클라이언트를 설계하고 구현한다. 즉, WDP레이어가 WSP레이어의 아래에 있고 데이터 송수신은 단지 클라이언트의 요청에 의해서만 이루어진다. 접속을 유지할 필요가 없으므로 경량의 데이터 트랜잭션을 관리하는 프로토콜인 WTP레이어가 존재하지 않는다.

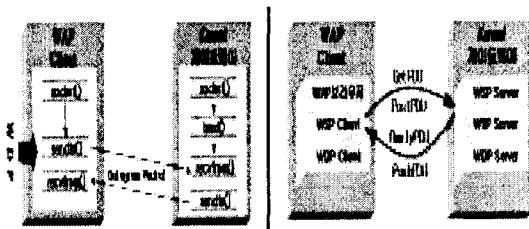
구현할 WAP스택은 단지 IPv4 Bearer위에서 Connectionless Mode를 지원한다. 중요한 문제는 다양한 다른 Bearer를 지원하기 위해 결국엔 ConnectedOriented Mode를 지원하도록 발전시켜야 한다. WSP의 주요한 역할은 클라이언트와 서버사이의 요청과 응답의 encoding/decoding이다. WSP레이어는 HTTP의 모든 메소드와 Content-Type, Date와 같은 중요한다고 생각되는 HTTP/1.1 헤더의 인코딩과 디코딩을 지원하도록 설계하였다.



<그림 4>비접속 지향 방식 클라이언트의 WAP스택

WAP클라이언트와 게이트웨이의 WSP계층에서는 요청의 형태에 따라 4종류의 PDU(Protocol Data Unit)가 존재한다. 클라이언트의 요청은 Get, Post에 따라서 GetPDU, PostPDU가 있으며 게이트웨이에서는 요청에 대한 응답인 ReplyPDU, 서버로 부터의 일방적인 데이터 전달인 PushPDU가 있다.

WSP아래에 있는 WDP는 Bearer와 통신을 담당하며 현재 IPv4만을 지원하며 WDP는 UDP소켓을 연결하고 사용자의 요청을 전달 게이트웨이에 전한다.

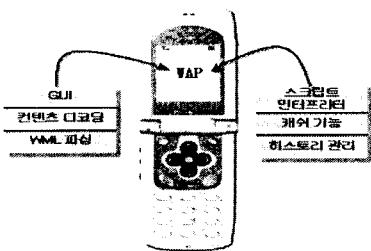


<그림 5> 설계된 WDP와 WSP 레이어

3.2 WAP브라우저의 설계

WAP브라우저는 게이트웨이로부터 전송 받은 WML인

코딩 데이터(Encoded Data)를 디코딩(Decoding)하여 무선 단말기 화면에 디스플레이하고 사용자 Input을 받아 들이는 역할을 한다. 많은 부분이 HTML브라우저와 유사한다. 하지만 WAP은 무선 이동통신망의 낮은 대역폭, 작은 디스플레이 등을 고려하여 컨텐츠를 인코딩하여 전송함으로 네트워크를 통해 전송되는 바이트의 수를 줄이는 방법을 사용한다.



<그림 6> WAP브라우저의 내부 구성

4. 이동 단말기용 WAP브라우저 구현

본 논문에서는 임베디드리눅스의 개발 경쟁에 빌 맞추어 OS로 리눅스를 사용하는 모든 Wireless장치에서 WAP의 기능을 가질 수 있는 WAP브라우저를 구현하였다.

운영체제로 레드햇 리눅스 7.0, 컴파일러는 GCC버전 2.96, XML라이브러리로 libxml2_2.2.0을 사용하였다.

WAP클라이언트인 WAP브라우저는 WAP게이트웨이에 있는 WAP스택의 동일한 레이어와 UDP패킷을 이용하여 통신하며 각각의 WAP스택은 패킷이 위 계층과 아래 계층으로 전달될 수 있도록 하는 큐(버퍼)의 역할을 수행하는 SAP이라는 스레드를 두어 패킷전송을 담당한다.

<그림 7>은 WDP레이어로 전달되는 DatagramPacket을 나타낸다. 모두 무선 환경이라는 특성 때문에 응답헤더와 문서정보가 인코딩되어 있다. 사용자 요청정보도 응답과 마찬가지로 WAP클라이언트에서 인코딩되어야 한다. 브라우저에서 사용자가 입력한 URL은 Accept와 User-Agent와 같은 요청헤더와 함께져서 SessionPacket을 생성한다.

SessionPacket은 WSP레이어로 전달되어 URL과

Header의 인코딩값과 패킷번호와 요청형태가 합쳐(Merge)진다. 인코딩되고 결합된 패킷은 WDP레이어로 전달되는 DatagramPacket이 된다.



<그림 7> WDP계층에 전달되는 패킷의 구조

5. 결론 및 향후연구방향

본 논문에서 구현한 무선단말기용 WAP브라우저는 WAP프로토콜을 사용하여 무선 인터넷에 접속하여 WAP컨텐츠를 브라우저화면에 표시할 수 있다.

본 논문을 작성하고 있는 시점에서 uCLinux리눅스를 탑재한 ARM보드환경으로의 WAP브라우저의 포팅에 대하여 연구하고 있으며 무선 단말에서 동작하는 프로그램이므로 WAP스택과 WAP브라우저의 크기를 좀 더 소형화하고 메모리의 사용량을 줄이도록 하는 연구가 기대된다. 또, IPv4방식이 아닌 차세대 인터넷통신규약인 IPv6를 사용하고 무선 접속 방식의 하나인 CDMA방식의 WDP레이어의 구현이 필수적이다.

[참고문헌]

- [1] WAPArch Spec, <http://www.wapforum.org/what/technical.htm>
- [2] 홍준호, 송건천, 김정석, about WAP, 영진출판사, 2000
- [3] Kannel WAP Gateway, <http://www.kannel.3glib.org/news.shtml>
- [4] 무선 인터넷 & WAP 개발 포럼, <http://mobile.daesang.co.kr>
- [5] Charles Arehart, Professional WAP, WroxPress 2000