

# WAP 게이트웨이의 유/무선 프로토콜 변환 기능에 대한 설계 및 구현

이정은<sup>0</sup>, 이종렬, 노강래, 신동일, 신동규  
세종대학교 컴퓨터 공학과

e-mail:{jelee, leemaster, krnoh, dshin, shindk}@gce.sejong.ac.kr

## Desing and Implementation of wire/wireless protocol Converting Function for WAP Gateway

Jung-Eun Lee, Jong-Youl Lee, Kang-Rae Noh, Dongil Shin, Dongkyoo Shin

\*\*Dept of Computer Engineering, Sejong University

### 요 약

무선통신 단말기의 발전과 함께 이를 활용할 수 있는 방법론에 대한 논의도 활발해지고 있으며, 이 중 가장 주목받고 있는 것이 이동 통신 단말기를 이용한 무선인터넷 서비스이다. 이러한 서비스의 기반이 되는 것이 무선 인터넷 프로토콜인 WAP(Wireless Application Protocol)이다. 현재의 셀룰러망을 이용한 통신은 비교적 낮은 전송률을 가지고 있어, 대용량의 데이터들을 포함하는 WWW 문서 및 대량의 무선 마크업 언어(WML: Wireless Markup Language) 문서를 전송하기에는 너무 느린 단점을 가지고 있다. 이에 대한 대안으로 본 논문에서는 WAP 포럼 내에서 제정한 프로토콜에 변환기능을 추가한 Gateway를 설계 및 구현 하였다.

### 1. 서 론

지난 몇 년 동안 무선통신 환경은 급변하고 있다. 호출기를 시작으로 하여 아날로그 핸드폰과 디지털 핸드폰, 개인 휴대전화기(PCS)를 거쳐 2002년 월드컵에 발맞춰 상용화될 화상이동통신(IMT2000)까지 10여 년 사이에 무선통신 환경은 예측하기 힘들 정도로 발전하고 있다.

무선통신 단말기의 발전과 함께 이를 활용할 수 있는 방법론에 대한 논의도 활발해지고 있으며, 이 중 가장 주목받고 있는 것이 이동 통신 단말기를 이용한 무선인터넷 서비스이다. 이 서비스의 기반이 되는 것이 WAP(Wireless Application Protocol)[1]이다.

WAP은 무선 환경에서 원활한 인터넷 정보 검색을 지원해주는 프로토콜의 하나로 짧은 역사에도 불구하고 정보통신 업계의 전폭적인 지원에 힘입어 무선 통신 환경의 표준으로 자리잡아 가고 있다.

현재 무선통신 단말기는 일반 데스크탑과 달리 저

용량의 프로세서와 메모리를 사용하고 있고, 사용하는 배터리의 용량에도 한계가 있기 때문에 무선통신 단말기에 적합한 다른 프로토콜의 지원이 필요하게 된다. WAP은 이 역할을 담당하기 위해 탄생한 프로토콜이다.

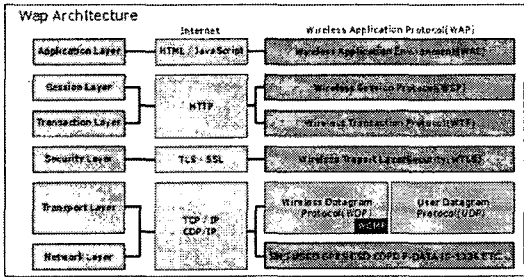
WAP 포럼 내에서는 기본적으로 모빌컴퓨터에 최적화한 마이크로브라우저의 규격을 명시함은 물론 프로토콜 전환이나 모빌컴퓨터에 맞는 데이터전송을 최적화시킴으로써 무선 네트워크와 유선 인터넷간의 게이트웨이 서버 규격도 조건짓는다.

본 논문의 구성은 다음과 같다. 2장 관련연구에서는 모바일을 통한 웹에 대한 접근을 가능하게 하는 무선응용 프로토콜인 WAP(Wireless Application Protocol)에 대한 개념과 구조에 대해 기술하고, 3장에서는 WAP Gateway내의 유/무선 프로토콜 변환 기능에 대한 구조적인 개념과 전체 시스템에 대한 설계와 구현을 설명하며, 4장에서는 구현환경과 5장에서는 결론 및 향후과제를 기술한다.

## 2. 관련 연구

### 2.1 WAP 프로토콜

WAP 구조는 이동 통신 장치를 위한 응용프로그램 개발을 위해 아래 [그림 1]에서 보는 바와 같이 확장 가능한 프로토콜 환경과 전체 네트워크의 계층화된 구조를 제공한다.



[그림 1] WAP 기본 구조

WAP 구조에서 각각의 계층은 상위 계층에 의해서 접근 될 수 있으며 다른 서비스나 응용프로그램에 의해서도 이용 될 수 있다. 외부 응용프로그램은 세션, 트랜잭션, 보안, 그리고 트랜스포트 계층에 직접 접근 할 수 있다. 다음은 각 레이어에 관한 개략적인 역할에 대한 설명이다.

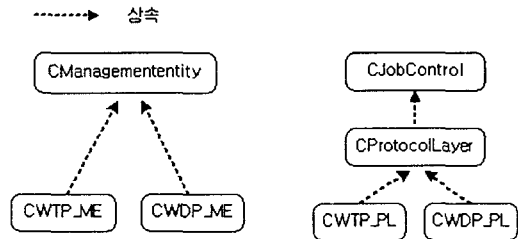
- WAE (Wireless Application Environment)[2] 서비스 제공자가 효과적이고 유용한 방법으로 방대한 다른 무선 플랫폼(platform)에 접근 할 수 있는 응용과 서비스를 만들 수 있는 상호작용 환경을 구축.
- WSP (Wireless Session Protocol)[3] WSP는 두 개의 세션 서비스를 위한 일관된 인터페이스를 가진 WAP의 응용 계층 프로토콜을 제공.
- WTP (Wireless Transaction Protocol)[4] 데이터그램 서비스의 상위에서 실행되고 인터넷 접속 전용 컴퓨터에서 실행하기에 적당한 간단한 트랜잭션 지향형 프로토콜.
- WTLS (Wireless Transport Layer Security)[5] 산업표준 트랜스포트 계층 보안 (TLS: Transport Layer Security) 프로토콜에 기반을 둔 보안 프로토콜.
- WDP (Wireless Datagram Protocol)[6] 다양한 네트워크 형태에 의해 지원되는 데이터를 운

반하는 능력을 가진 전송 서비스 위에서 동작.

### 3. 프로토콜 변환 기능을 가진 Gateway

본 논문에서는 무선 애플리케이션 프로토콜(WAP)용 프로토콜 처리 기능 중에서 WTP, WDP의 기술 개발 결과에 대하여 기술한다.

#### 3.1 데이터 송수신부와 트랜잭션 처리부의 기본 구조와 기본 클래스별 용도



[그림 2] 주요 클래스의 상속도

각 계층의 구현에서 [그림 2]와 같이 모든 계층에서 사용할 수 있는 CManagementEntity 클래스와 CProtocolLayer 클래스를 각각 XXX\_ME 클래스와 XXX\_PL 클래스가 상속하여 사용하고 CProtocolLayer 클래스는 CJobControl클래스를 상속하도록 하였다.

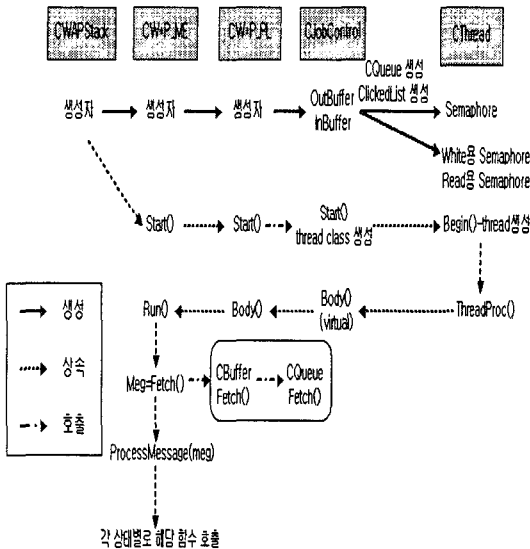
CManagementEntity클래스는 각 계층에서 상위 계층과 하위 계층의 유무에 따라 상위 계층 또는 하위 계층 또는 상위와 하위 계층을 지정해줄 수 있는 함수를 가지고 있는 클래스이고, CProtocolLayer 클래스는 위 CManagement클래스에서 계층을 지정할 때 지정함수 내부에서 CProtocol클래스 내부의 계층 지정함수를 호출하고 지정함수에서 CProtocol 클래스 내부의 멤버변수에 상위 또는 하위 또는 상위,하위 계층을 지정하게 한다.

CJobControl 클래스는 상위 또는 하위 또는 상위,하위 계층으로의 정보 전송 및 수신을 위한 OutBuffer와 InBuffer를 생성하고, CLinkedList 클래스를 사용하는 CQueue 클래스를 생성한다. 또, OutBuffer와 InBuffer용 semaphore를 CThread 클래스에서 생성한다.

Thread는 위 [그림 2]의 CJobControl 클래스에서 start()함수를 이용하여 thread객체를 생성하고 CThread 클래스 내에서 Begin()함수를 이용하여

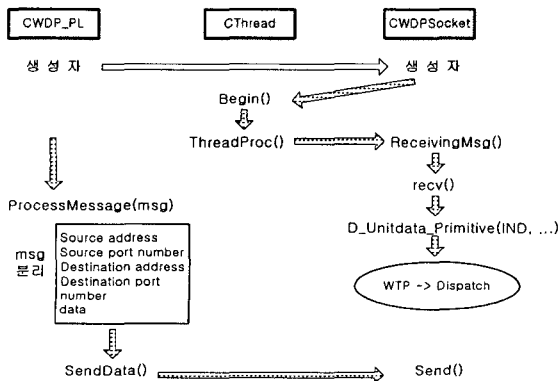
thread를 생성한다. thread를 생성함으로써 CW\*P\_PL객체에서 CJobControl클래스에서 생성한 InBuffer로부터 데이터를 읽어들이며 메시지를 처리(ProcessMessage(msg))한다. 그리고, 다른 계층으로 데이터를 넘겨줄 때는 상위 또는 하위 계층의 Dispatch()함수를 이용하여 상위 또는 하위 계층의 InBuffer로 데이터를 저장한다. 메시지를 Buffer에 저장할 때 동기화 시키기 위해 semaphore를 이용하였다.

### 3.2 데이터 송수신부의 추가 모듈



[그림 3] 주요 클래스 상속도

WDP 계층은 위의 [그림 2]와 같은 구조 외에 [그림 3]과 같은 모듈이 추가되어 있다.

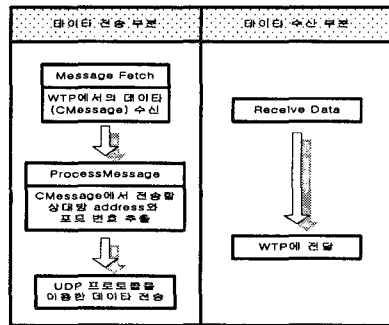


[그림4] WDP 구조도

이 추가된 모듈은 CWDP\*Socket 클래스를 CWDP\_PL객체에서 생성하여 CThread객체의 Begin()함수를 호출하여 새로운 thread를 생성해서 서버에서 오는 데이터를 수신하는 일을 하는 또 다른 thread를 가지고 있다.

위의 [그림 4]는 WDP의 데이터 수신 부분과 전송 부분을 도식화한 그림으로 서버에서 데이터를 수신하는 부분과 WTP로 데이터를 넘겨주는 부분이다.

### 3.3 트랜잭션 처리부의 추가 모듈

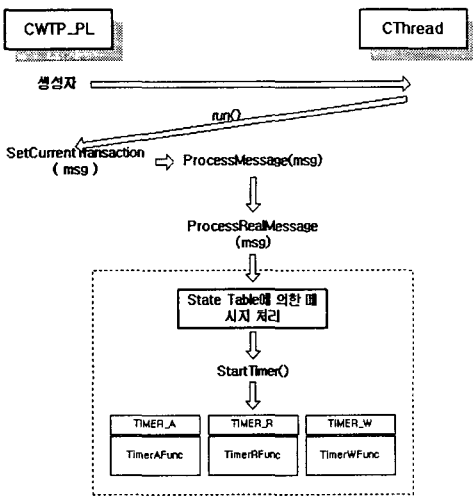


[그림5] WTP 추가 구성 모듈

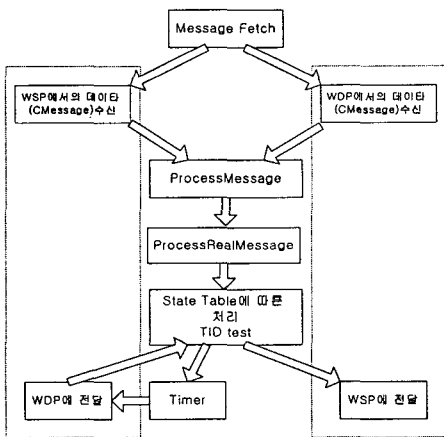
WTP 계층은 위의 [그림 2]와 같은 구조 외에 [그림 5]와 같은 모듈이 추가되어 있다.

CWTP\_PL의 Run() 함수는 CThread 개체에 의해 쓰레드로 생성되어 실행되고 이 함수는 상위레이어(WSP)나 하위레이어(WDP)의 데이터를 수신하여 ProcessMessage() 함수를 호출하여 mutex개체를 만들고 실제 데이터에 대한 이벤트를 처리하기 위한 ProcessRealMessage() 함수를 호출한다.

이 함수는 State Table에 의한 메시지를 처리하며 WDP를 통한 데이터 전송시 StartTimer()함수를 호출하여 새로운 쓰레드를 생성하여 Time\_A, Timer\_R, Timer\_W 일 경우로 나누어서 각각 TimerAFunc(), TimerRFunc(), TimerWFunc() 함수를 호출하여 메시지를 처리한다. 만약 세 번의 재전송이 실패하게 되면 트랜잭션의 연결은 종료된다 (Abort PDU 생성 및 전달). 아래의 [그림 6]과 아래의 [그림 7]에서는 각각 WTP의 구조도와 전송된 메시지를 처리하는 과정으로 본 논문에서 구현한 프로토콜 변환기능의 과정에 대한 흐름도를 보여주고 있다.



[그림 6] WTP 구조도



[그림 7] 전체 구성 흐름도

이러한 WAP 게이트웨이는 WAP 서비스를 지원하기를 희망하는 CP(Content Provider)들이나 무선 인터넷 개발자들에게 도움이 될 것이다.

향후에는 WAP 프로토콜 변환에 따르는 게이트웨이의 버전관리와 효율적인 전송을 위해 WAP 게이트웨이 내부에 캐쉬 서버를 내장하는 작업등이 요구된다.

참고문헌

- [1] "Wireless Application Protocol Architecture Specification", WAP Forum, April 30, 1998, <http://www.wapforum.org/>
- [2] "Wireless Application Environment Specification", WAP Forum, April 30, 1998, <http://www.wapforum.org/>
- [3] "Wireless Session Protocol", WAP Forum, April 30, 1998, <http://www.wapforum.org/>
- [4] "Wireless Transaction Protocol Specification", WAP Forum, April 30, 1998, <http://www.wapforum.org/>
- [5] "Wireless Transport Layer Security", WAP Forum, April 30, 1998, <http://www.wapforum.org/>
- [6] "Wireless Datagram Protocol", WAP Forum, April 30, 1998, <http://www.wapforum.org/>
- [7] webmin, <http://www.webmin.com>

4. 구현환경

본 논문에서 구현한 WAP 게이트웨이의 유/무선 프로토콜 변환 기능은 게이트웨이 서버를 원격으로 관리할 수 있는 리눅스 기반으로 하여 C로 구현 하였다. 구현한 WAP 게이트웨이 관리 프로그램으로는 webmin[7]의 모듈을 perl로 구현하여 시험하였다.

5. 결론

본 연구에서는 휴대용 무선장비에서 웹 접근을 가능하게 하는 무선 응용 프로토콜인 WAP에 대한 개념과 구조에 대해 살펴보았고, WAP 게이트 웨이의 유/무선 프로토콜 변환 기능에 대하여 설명하였다.