

프로토콜 변환 기능을 포함한 WAP 전용 Gateway에 대한 설계 및 구현

이정은⁰ 노강래 이종렬 신동일 신동규
 세종대학교 컴퓨터 공학과
 (jelee, kmoh, leemaster, dshin, shindk)@gce.sejong.ac.kr

Desing and Impelementation of WAP Gateway for protocol Converting Fuction

Jung-Eun Lee⁰, kang-Rea Noh, Jong-Youl Lee, Dong-il Shin, Dong-kyoo Shin
 Dept. of Computer Engineering, Sejong University

요 약

현재의 CDMA(code-division multiple access) 셀룰러망을 이용한 통신은 비교적 낮은 전송률을 가지고 있어, 대용량의 데이터들을 포함하는 웹 문서 및 대량의 무선 마크업 언어(WML: Wireless Markup Language) 문서의 전송 속도면에서 미흡한 실정이다. 이에 대한 대안으로 해결을 위하여 WAP(Wireless Application Protocol) 포럼 내에서는 기본적으로 휴대용 기기나 모바일 기기에 최적화한 마이크로브라우저의 규격을 명시함은 물론 프로토콜 전환이나 모빌컴퓨터에 맞는 데이터전송을 최적화시킴으로써 무선 네트워크와 유선 인터넷간의 게이트웨이 기능을 하는 프록시서버 규격도 확립하였다. 본 논문에서는 이러한 점을 고려하여 Gateway 내에서 프로토콜을 변환하는 시스템을 설계 및 구현한다.

1. 서 론

최근의 WWW(World Wide Web)과 관련된 기술의 발달로 개인용 컴퓨터를 이용해서 언제, 어디서나 인터넷에 있는 정보를 취득할 수 있게 되었다. 국내외적으로 무선 애플리케이션 프로토콜(WAP: Wireless Application Protocol[1])이 업체들의 잇단 지원을 받으면서 휴대단말기의 표준데이터 전송규격으로 급속히 자리잡아가고 있다. 현재 W3C(World Wide Web Consortium)에서는 휴대용 무선장비의 Web 접근을 위한 표준제정을 위하여 이 기술 분야를 주도하고 있는 UP(Unwired Planet)사를 중심으로 에릭슨, 모토롤라, 노키아 등이 1997년에 결성한 WAP 포럼을 구성하여 표준화 작업을 진행하고 있으며, 2001년 8월 현재 WAP Specification 2.0까지 제정되었다.

현재의 CDMA 셀룰러망을 이용한 통신은 비교적 낮은 전송률을 가지고 있어, 대용량의 데이터들을 포함하는 웹 문서 및 대량의 무선 마크업 언어(WML: Wireless Markup Language) 문서를 전송하기에는 너무 느리다. 전송시간의 지연은 통화 요금을 상승시킬 뿐만 아니라 사용자의 불만을 가져오게 되어 무선 데이터 서비스 이용의 활성화에 가장 큰 장애로 작용할 수 있다. 이의 해결을 위하여 WAP 포럼 내에서는 기본적으로 모빌컴퓨터에 맞는 데이터전송을 최적화시킴으로써 무선 네트

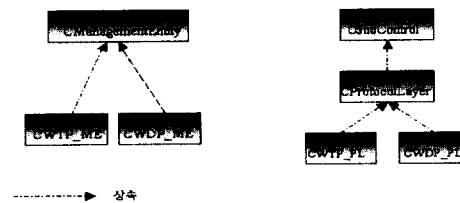
워크와 유선 인터넷간의 게이트웨이 기능을 하는 프록시서버 규격도 확립하였다. 본 논문에서는 이러한 점을 고려하여 Gateway 내에서 프로토콜을 변환하는 시스템을 설계 및 구현한다

본 논문에서는 WAP 명세서에 정의되어 있는 프로토콜을 기반으로 하여 프로토콜 변환 기능을 포함한 Gateway를 설계 및 구현한다.

2. 프로토콜 변환 기능을 가진 Gateway

본 논문에서는 무선 애플리케이션 프로토콜(WAP)용 프로토콜 처리 기능 중에서 WTP(Wireless Transaction Protocol Specification)[2], WDP(Wireless Datagram Protocol)[3]의 기술 개발 결과에 대하여 기술한다.

2.1 WTP, WDP의 기본 구조와 기본 클래스별 용도

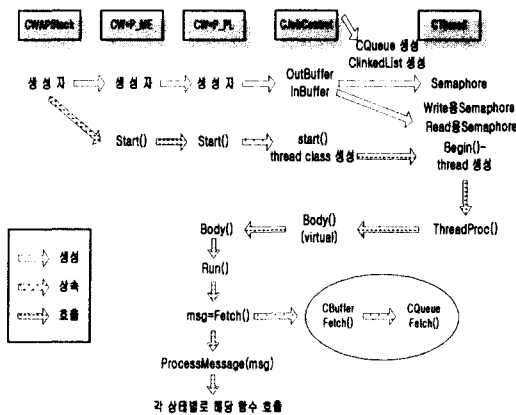


[그림1] 주요 클래스의 상속도

각 계층의 구현에서 [그림 1]과 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 계층과 혹은 하위 계층의 선택 또는 상위와 하위 계층을 지정해줄 수 있는 함수를 가지고 있는 클래스이고, CProtocolLayer 클래스는 위 CManagement클래스에서 계층을 지정할 때 지정 함수 내부에서 CProtocol클래스 내부의 계층 지정 함수를 호출하고 지정 함수에서 CProtocol 클래스 내부의 멤버 변수에 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 아래 [그림 1]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.

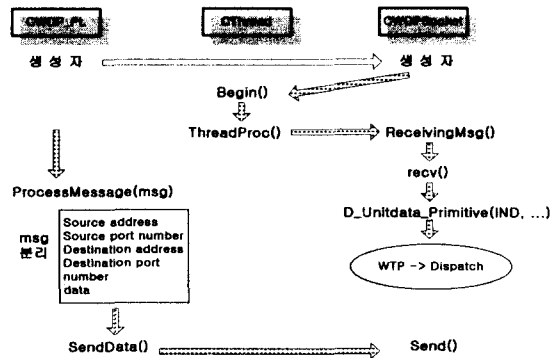
2.2 WDP의 추가 모듈



[그림 2] 주요 클래스의 상속도

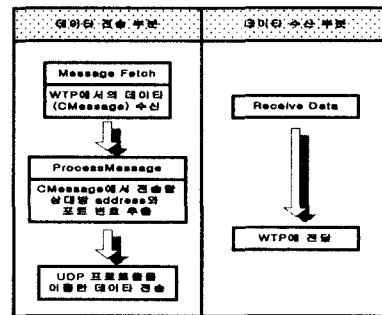
WDP 계층은 위의 [그림 1]와 같은 구조 외에 [그림 2]과 같은 모듈이 추가되어 있다. 이 추가된 모듈은 CWDP_Socket 클래스를 CWDP_PL객체에서 생성하여 CThread객체의 Begin()함수를 호출하여 새로운 thread를 생성해서 서버에서 오는 데이터를 수신하는 일을 하는 또 다른 thread를 가지고 있다.

아래의 [그림 3]는 WDP의 데이터 수신 부분과 전송 부분을 도식화한 그림으로 서버에서 데이터를 수신하는 부분과 WTP로 데이터를 넘겨주는 부분이다.



[그림3] WDP 구조도

2.3 WTP의 추가 모듈



[그림4] WTP 추가 모듈 구성도

WTP 계층은 위의 [그림 1]과 같은 구조 외에 [그림 4]와 같은 모듈이 추가되어 있다.

CWTP_PL의 Run() 함수는 CThread 개체에 의해서 스레드로 생성되어 실행되고, 이 함수는 상위레이어 (WSP[4])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 WDP

를 통한 데이터 전송시 StartTimer()함수를 호출하여 새로운 스레드를 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).

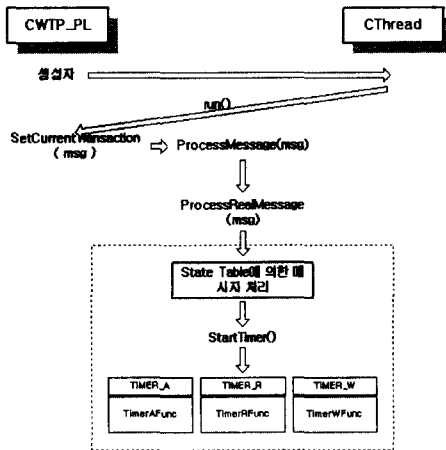
이트웨이 관리 프로그램으로는 webmin의 모듈을 perl로 구현하여 시험하였다.

5. 결론 및 향후 연구

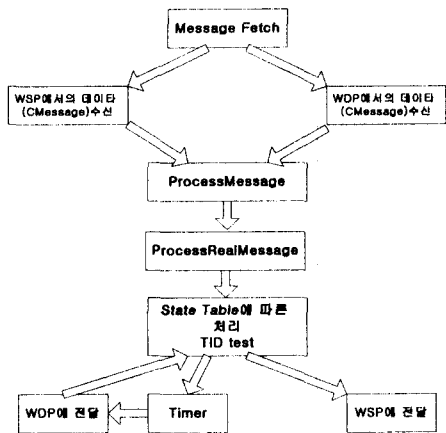
본 과제에서 구현한 WAP Gateway는 트랜잭션 처리를 담당하는 WTP와 데이터 수신 및 송신을 담당하는 WDP은 보다 효율적인 전송을 보장하지만, 이동통신망의 버전 관리를 위한 응용 프로토콜에 관한 연구와 전송 프로토콜에 따른 버전관리에 대한 연구가 요구된다.

참고문헌

- [1] [WAP] "Wireless Application Protocol Architecture Specification", WAP Forum, April 30, 1998. URL : <http://www.wapforum.org/>
- [2] [WTP] "Wireless Transaction Protocol Specification", WAP Forum, April 30, 1998. URL : <http://www.wapforum.org/>
- [3] [WDP] "Wireless Datagram Protocol", WAP Forum, April 30, 1998. URL : <http://www.wapforum.org/>
- [4] [WSP] "Wireless Session Protocol", WAP Forum, April 30, 1998. URL : <http://www.wapforum.org/>



[그림 5] WTP 구조도



[그림 6] WTP 구성도

4. 구현환경

본 논문에서 구현한 WAP Gateway는 Gateway 서버를 원격으로 관리할 수 있는 리눅스 기반으로 하였으며 프로그래밍 언어는 C를 사용하여 구현 하였다. 구현한 계