□특집□

무선 응용 프로토콜 기술

김 기 조[†], 최 윤 석[†], 최 은 정[†], 임 경 식[††]

◈ 목 차 ◈

1. 서 론

세계적으로 이동통신 가입자가 급속히 증가하고 인터넷이 일반화되면서 휴대성과 이동성으로 대표되는 이동통신을 인터넷과 결합한 새로운 형태의 서비스가 보편화되고 있다. 우리 나라의 경우 2000년 3월말을 기준으로 이동통신 가입자가 수가 2,750만을 넘어 이동전화 보급률이 58.2%에 이르고 있으며 이에 발맞추어 이동통신 사업자들은 이동전화를 이용한 무선 인터넷 서비스를 경쟁적으로 제공하고 있다.

현재 이동통신 시장에서는 무선 인터넷 서비스를 지원하기 위한 표준화가 Wireless Application Protocol(WAP) 포럼, World Wide Web Consortium (W3C)과 마이크로소프트사 등을 중심으로 진행되고 있다. 그 중에서 WAP은 에릭슨, 모토롤러, 노키아, 폰닷컴등 유력한 이동통신 업체들을 중심으로 AT&T, 벨사우스, IBM을 포함하여 200여 개의 통신 및 컴퓨터업체들이 참여하여 표준 및 관련 응용을 활발히 개발하고 있어 산업 표준이 될 가능성이 높다. WAP은 기존의 인터넷 표준을 기반으로 무선 통신망 및 이동단말 환경에서 최적의 기능을 수행하도록 설계된 무선 인터넷 응용 프로토콜이다. 본 논문에서는 WAP의 전체적인 구조와 주요 구성요소 및 게이트웨이 구축 기술에 대하여 간단히 소개하고자 한다.

† 준 회 원 : 경북대학교 컴퓨터과학과 석사과정
†† 정 회 원 : 경북대학교 컴퓨터과학과 조교수

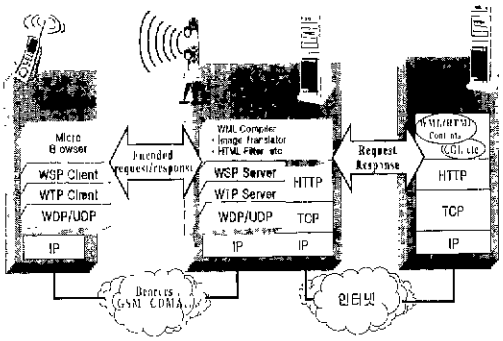
2. WAP 구조

무선통신 환경은 휴대 단말의 크기, 컴퓨팅 능력, 제한된 입·출력 장치로 인한 제약만이 아니라 무선망의 낮은 대역폭, 데이터 전송 지연과 불안정한 접속 등 다양한 문제점들이 있다. WAP은 이러한 문제점들을 완화하면서 안정된 무선 인터넷 서비스를 제공하기 위하여 다섯 가지 요구사항 즉, 상호운용성(interoperability), 확장성(scalability), 효율성(efficiency), 신뢰성(reliability) 그리고 보안성(security)을 고려하여 설계되었다[1].

2.1 WAP 모델

(그림 1)은 WAP에서 제안한 무선 인터넷 서비스 모델이다. WAP 모델은 무선통신 환경의 문제점들을 극복하기 위하여 유선망에서는 기존의 유선 프로토콜을 그대로 사용하고 무선망에서는 무선환경에 적합하도록 설계된 프로토콜을 사용하는 연결 분리(split connection) 기법을 채택하고 있다. 그러므로, 이러한 분리된 연결을 동적으로 연

동하기 위하여 WAP 게이트웨이(gateway)가 필요하며, 클라이언트는 항상 게이트웨이를 통하여 인터넷으로 연결된다.



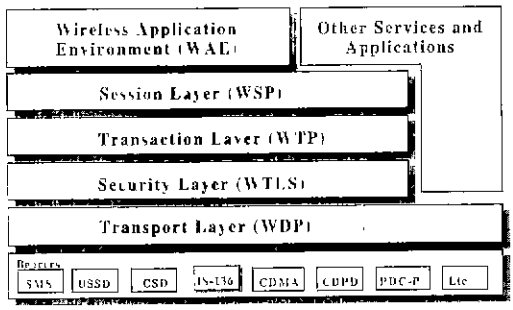
(그림 1) WAP 모델

WAP 게이트웨이는 이동 클라이언트로부터 전달되는 인코딩된 요구 메시지를 HTTP[2] 요구 메시지로 전환하고 웹 서버로부터 수신된 응답 메시지를 인코딩된 응답 메시지로 변환하여 클라이언트로 전달한다. 또한 게이트웨이에서는 텍스트 Wireless Markup Language(WML)[3] 문서를 바이너리 WML 문서로 인코딩 하는 기능과 WML Script[4]를 바이너리 형식으로 변환하는 기능도 제공한다. WAP 모델은 기존의 인터넷과 연동 모델로 클라이언트에서 인터넷 컨텐트를 검색하기 위해서는 HTML 컨텐트를 WML 컨텐트로 변환하는 기능을 추가적으로 필요로 한다.

2.2 WAP 프로토콜 스택

(그림 2)의 WAP 프로토콜 스택[1]은 Wireless Application Environment(WAE)[5], Wireless Session Protocol(WSP)[6], Wireless Transaction Protocol(WTP)[7], Wireless Transport Layer Security(WTLS)[8], Wireless Datagram Protocol(WDP)[9] 및 Bearers로 구성되어 있다.

WAE는 무선 인터넷 서비스와 이동전화 서비스를 지원할 수 있는 개발환경으로서 어떠한 무



(그림 2) WAP 프로토콜 스택

선 플랫폼에서도 운용될 수 있는 환경을 만드는 데 그 목적이 있으며, WWW와 이동전화 기술에 기반한 일반적인 목적의 응용 개발환경을 제공한다. 현재 WAE 계층에서는 이동단말 장치에 적합한 컨텐트와 사용자 인터페이스를 제공하는 WML, 이동단말에서 수행되는 프로그램을 작성하기 위하여 자바 스크립트의 부분집합을 확장한 WMLScript, 전화 서비스와 관련된 응용 개발 인터페이스를 제공하기 위한 WTA(Wireless Telephony Application)[10] 및 WTAI(Wireless Telephony Application Interface)[11]를 비롯한 다양한 서비스와 형식을 정의하고 있다. 이러한 서비스를 이용한 대표적인 응용으로 마이크로브라우저가 있다.

WSP는 현재 브라우징 서비스를 위한 제한된 기능만을 포함하므로 WSP/B라고도 한다. WSP/B는 연결형(connection oriented) 세션 서비스와 비연결형(connectionless) 세션 서비스를 정의하고 있다. 연결형 세션 서비스는 WTP 상위 계층에서 동작 하며 비연결형 세션 서비스는 WDP나 User Datagram Protocol(UDP) 위에서 동작한다. WSP/B는 인터넷과의 연동을 위하여 HTTP/1.1에 상응하는 기능들을 제공하고, 무선 구간에서 컨텐트 전달의 최적화를 위하여 클라이언트/서버 기능(capability)을 협상하며 헤더 및 기능을 인코딩하는 작업을 수행한다. WSP/B는 클라이언트의 요구가 없이도 서버가 컨텐트를 전달할 수 있는 푸시(push) 기능과 세션이 유휴 상태에 있을 경우 네트워크

리소스를 해제하거나 배터리를 절약하기 위하여 세션을 일지 중지(suspend) 및 재개(resume)하는 기능을 추가적으로 제공하고 있다.

WTP는 데이터그램 서비스 상에서 동작하는 트랜잭션 지향 프로토콜이다. WTP는 제한된 성능을 갖는 클라이언트 환경에서도 쉽게 구현될 수 있고 무선 데이터그램 네트워크 상에서 효과적으로 동작하도록 설계되었다. 트랜잭션 서비스를 세 종류의 클래스 즉, 비신뢰적 단방향 요구(class 0), 신뢰적 단방향 요구(class 1) 및 신뢰적 양방향 요구/응답(class 2)으로 나누어서 차별화된 서비스를 제공한다. 이러한 WTP의 대표적인 장점은 다음과 같다.

- 유일한 트랜잭션 식별자, 응답(acknowledgment), 재전송을 통하여 WTP 상위 계층에게 신뢰성을 제공한다.
- 명시적인 연결 설정/종료를 하지 않으므로 링크 오버헤드를 줄인다.
- 메시지 지향 서비스이며 브라우징을 위한 요구 및 응답 형식을 지원하는 트랜잭션 서비스를 제공한다.

WTLS는 산업 표준인 Transport Layer Security (TLS) 프로토콜에 기반을 둔 보안 프로토콜로서 무결성(data integrity), 기밀성(privacy), 인증(authentication) 및 부인봉쇄(denial of service protection) 기능을 가진다[1].

WDP는 종단간 전송을 위한 어드레싱을 제공하며 인터넷의 UDP와 같은 멀티플렉싱(multiplexing) 기능을 담당한다.

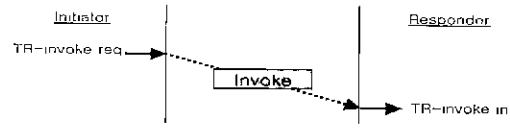
3. WAP 구성 요소

3.1 WTP

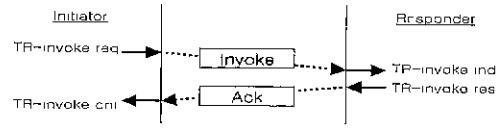
본 단락에서는 상호적인 "브라우징"(요구/응답) 응용 소프트웨어가 필요로 하는 서비스를 제공하기 위한 WTP의 프리미티브, 트랜잭션 클래스 형식 및 동작, 그리고 주요기능에 대해서 기술한다.

3.1.1 WTP 클래스 종류와 동작

모든 트랜잭션 클래스는 요구자가 설정하여 응답자에게 전달한다.



(그림 3) 클래스 0 트랜잭션

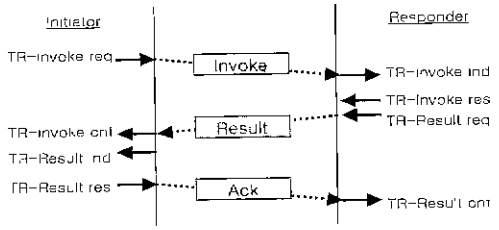


(그림 4) 클래스 1 트랜잭션

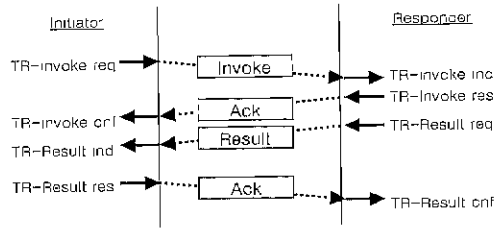
(그림 3)은 신뢰성 없는 단방향 데이터 전송을 위한 클래스 0 트랜잭션을 나타내고, (그림 4)는 신뢰성이 있는 단방향 데이터 전송을 위한 클래스 1 트랜잭션을 나타낸다. 클래스 0 트랜잭션은 요구자가 Invoke 메시지를 응답자에게 전달하면서 시작되며, 응답 또는 메시지 재전송 기능이 없다. 이에 비하여, 클래스 1 트랜잭션은 메시지 재전송 기능을 가진 요구자가 Invoke 메시지를 전달함으로써 트랜잭션을 시작하고, 응답자는 Ack 메시지를 전달함으로써 신뢰성 있는 데이터 전송을 가능하게 한다. 클래스 0 트랜잭션은 응용 계층의 비신뢰적 푸시 서비스를 지원하기 위한 데이터그램 서비스를 제공하며, 클래스 1 트랜잭션은 응용 계층이 신뢰적 푸시 서비스를 지원하기 위한 신뢰성 있는 데이터그램 서비스를 제공한다.

(그림 5)와 (그림 6)은 신뢰적 양방향 요구/응답을 수행하는 클래스 2 트랜잭션이다. 요구자는 Invoke 메시지를 응답자로 전달하고, (그림 5)에서와 같이 응답자는 Result 메시지를 전송함으로써

묵시적으로 응답한다. 이렇게 함으로써 전송되는 메시지의 수를 줄여 전송 성능을 향상시킬 수 있다. 이에 비해 (그림 6)은 Result 메시지를 전송하기 전에 응답자의 응답 타이머가 타임아웃 되면 Hold on ack 메시지를 전송함으로써 요구자가 불필요하게 Invoke 메시지를 재전송 하는 것을 방지한다.



(그림 5) 클래스 2 트랜잭션



(그림 6) 클래스 2 트랜잭션 (hold on 응답)

3.1.2 WTP의 주요 기능

응답확인을 위한 재전송(re-transmission until acknowledgment)은 WTP 제공자 사이에 신뢰적인 데이터 전송을 제공한다. 재전송 타이머는 패킷이 전송될 때 카운터가 0으로 초기화되어 시작된다. 재전송 타이머가 타임아웃 될 때까지 전송된 패킷에 대한 응답이 없으면, 재전송 카운터가 증가되고 패킷이 재전송 되어 재전송 타이머가 다시 시작된다. 이러한 과정이 되풀이되어 재전송 카운터가 최대 재전송 값을 초과하게되면, 제공자는 트랜잭션을 취소하고 사용자에게 알려준다.

사용자 응답(user acknowledgment)은 사용자가 수신된 메시지에 대하여 응답하기 위하여 사용된다. 응답자는 U/P 필드가 설정된 Invoke 메시지를 받으면 사용자에게 알리고 응답 타이머를 시작한 후 사용자로부터 응답을 기다린다. 만약 사용자가 응답하기 전에 요구자로부터 Invoke 메시지가 재전송 되면, 응답자는 메시지를 버리고 응답 타이머를 재시작 한다. 정해진 시간이 지난 후에도 사용자로부터 응답이 없으면 트랜잭션은 취소된다.

수신자는 다중의 패킷 손실에 대하여 재전송을 요구할 때 제공자에 의해 전송되는 패킷의 수를 최소화하기 위하여 Invoke나 Result 메시지를 분리하고 재결합(segmentation and re-assembly)한다.

메시지가 네트워크 최대 전송 크기를 초과할 경우에는 메시지를 분할된 몇 개의 그룹으로 나누어서 전송하며 각각의 그룹은 정지 및 대기(stop-and-wait) 방식으로 동작한다.

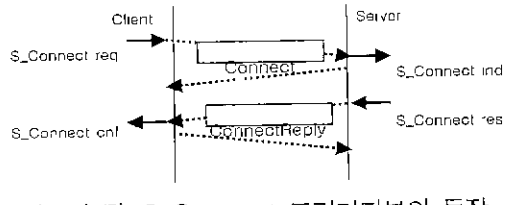
3.2 WSP

WSP의 연결형 서비스와 비연결형 서비스의 기능과 동작 방식을 간단히 기술하고, 헤더 인코딩 및 기능 협상에 대하여 알아본다.

3.2.1 WSP 프로토콜 동작

(1) 세션 관리 기능

세션은 클라이언트와 서버 사이에서 메소드나 푸시 트랜잭션 등을 수행하기 위하여 논리적으로 유지되는 연결을 의미한다.

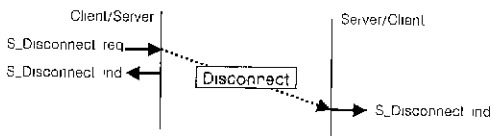


(그림 7) S-Connect 프리미티브의 동작

(그림 7)의 S-Connect 프리미티브는 클라이언트가 서버에 세션을 설정하기 위하여 사용되며,

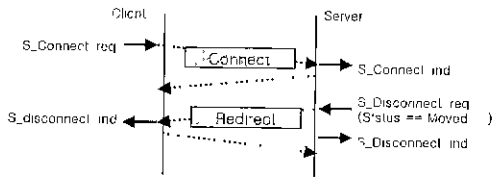
WTP 클래스 2 트랜잭션을 사용한다. 클라이언트는 서비스 사용자로부터 전달된 정보들을 Connect 메시지로 만들어 서버로 전달하며, 서버는 이에 대하여 ConnectReply 메시지로 응답한다. 세션 설정 시에는 요구자가 기능을 제안하고 응답자는 협상된 기능을 알려주는 단방향 기능 협상(one-way capability negotiation)과 헤더 정보 교환이 이루어진다. 헤더 정보는 세션이 종료될 때까지 일정하게 유지되면서 사용자가 필요에 따라 사용할 수 있다. 프리미티브 이벤트 발생 시에는 수행해야 할 다양한 작업들이 있으며 그 세부적인 내용은 스팩의 상태 테이블(state table)에 명시되어 있으나 지면 관계상 본 논문에서는 생략한다.

(그림 8)의 S-Disconnect 프리미티브는 클라이언트 또는 서버에서 세션을 종료할 때 사용되며 WTP 클래스 0 트랜잭션을 사용한다. 이 경우 서비스 제공자는 세션 식별자에 대한 정보가 있는 Disconnect 메시지를 전달하며 세션의 종료와 관련된 정보들을 서비스 사용자에게 알려준다.



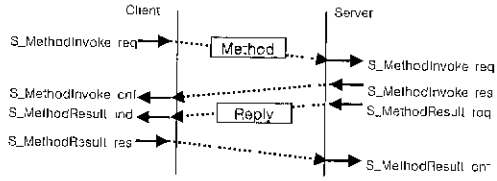
(그림 8) S-Disconnect 프리미티브의 동작

또한, S-Disconnect 프리미티브는 서버가 세션을 설정하는 과정에서 클라이언트가 세션을 재 설정할 수 있도록 메시지를 전달하는 역할도 한다. 이 때 전달되는 메시지는 Redirect 메시지이며 세션 연결이 가능한 서버의 주소들에 대한 정보가 있다.



(그림 9) 세션 연결 중 S-Disconnect 프리미티브의 동작

(2) 메소드 호출 기능



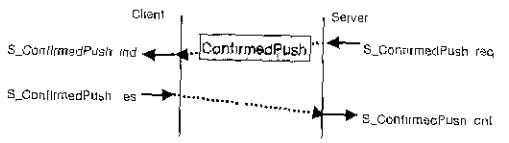
(그림 10) S-MethodInvoke 및 S-MethodResult 프리미티브의 동작

(그림 10)은 S-MethodInvoke와 S-MethodResult가 메소드 호출 기능을 수행하기 위하여 WTP 클래스 2 트랜잭션을 사용하여 수행되는 과정을 나타낸다. 클라이언트는 Method 메시지를 전달하여 서버에서 수행될 메소드와 그 동작을 지정하고, 서버는 수행된 결과를 Reply 메시지로 응답한다. Method 메시지에는 Get 메시지와 Post 메시지 두 가지가 정의되어 있으며, 사용자가 요구한 메소드 종류에 따라서 결정된다. 메소드의 종류는 HTTP/1.1 [2]에 정의된 메소드와 동일하며, GET이나 HEAD 메소드의 경우에는 Get 메시지가 사용되고, POST나 PUT 메소드의 경우에는 Post 메시지가 사용된다. 메소드 요청 및 응답 시에 첨부되는 헤더 정보는 인코딩 되어서 전달되므로 Method 메시지의 크기를 줄여준다.

(3) 푸시 기능

푸시는 클라이언트가 정보를 요청하지 않은 상태에서 서버가 임의로 정보를 전달할 수 있는 기능을 제공하며 비신뢰적 푸시와 신뢰적 푸시 두 가지가 있다. 그러나, 사용하는 메시지의 형식은 동일하다. 비신뢰적 푸시는 WTP 클래스 0 트랜잭션을, 신뢰적 푸시는 WTP 클래스 1 트랜잭션을 사용하여 구현된다. (그림 11)은 신뢰적 푸시 기능을 구현하기 위한 S-ConfirmedPush 프리미티브의 동작

과정을 나타낸 것이며, 푸시 기능을 구현한 S-Push 프리미티브의 그림은 지면 관계로 생략한다.



(그림 11) S-ConfirmePush 프리미티브의 동작

3.2.2 헤더 인코딩

WSP 헤더는 HTTP/1.1[2]의 헤더을 모두 수용하며 WAP에서 사용하기 위한 헤더가 추가되어 있다. WSP 헤더는 텍스트 형식이므로 전송되는 데이터의 양을 줄이기 위하여 헤더 정보를 축약된 바이너리 형태로 인코딩 한다. WSP 헤더는 <표 1>과 같이 필드 이름과 필드 값으로 구성되어 있으며 인코딩 기법에 의하여 인코딩 된 헤더는 (그림 12)과 같다.

〈표 1〉 WSP 헤더 형식

Field Name	Field Value
Accept-Language	en

HTTP/1.1 헤더 [19 byte] : Accept-Language: en
인코딩 된 헤더 [2 byte] : 0x83 0x99
[0x83] = Accept-Language [0x99] = en

(그림 12) WSP 헤덩 인코딩 예

3.2.3 기능 협상 및 인코딩

기능은 클라이언트와 서버 양자간 서비스 수준을 조절하고 서비스 제공자의 수행을 최적화하기 위해 사용된다. 기능 협상 시에는 클라이언트와 서버 사이에서 전송되는 최대 메시지 크기, 세션 수행 중에 동시에 수행될 수 있는 최대 메소드 및 신뢰적 푸시 트랜잭션의 수가 협상된다. 그리고, 세션 일시중지나 재개 또는 푸시 기능의 사용 여부 등이 결정된다. 기능 협상 요구는 인코딩 되어서 전달되며 인코딩 형식은 <표 2>와 같고 그 예는 (그림 13)과 같다.

〈표 2〉 기능 협상 요구의 인코딩 형식

Length	Capability Identifier	Parameter

Capability field : Server SDU Size 1400
인코딩 된 field : 0x03 0x81 0x85 0x78
0x03 :메시지 길이 0x81 : Server SDU Size
0x85 0x78 : 1400의 바이너리 값

(그림 13) 기능 협상 인코딩 예

3.3 WSP/WTP 구현

본 단락에서는 WSP/WTP 구현 시 고려되어야 할 여러가지 사항들과 응용 프로그래밍 인터페이스 설계에 대하여 기술한다.

3.3.1 구현 시 고려사항

프로토콜은 라이브러리 모델과 프로세스 모델 두 가지 방법으로 구현 할 수 있다. 프로세스 모델은 각각의 프로토콜 계층을 프로세스로 구현하며 각 계층간 InterProcess Communication(IPC) 방식으로 메시지를 교환한다. 이 모델은 IPC로 인한 부하가 많고, 메시지 버퍼 관리가 복잡하지만, 많은 응용을 처리하는데 유리하다. 이에 비하여, 라이브러리 모델은 각각의 프로토콜 계층을 라이브러리로 구현하며 계층간 메시지 교환이 함수 호출 방법을 통하여 수행되므로 IPC로 인한 부하가 없으며 메시지 버퍼 관리도 간단하다. 그리고, 프로토콜은 각각의 응용 프로세스 내에서 독립적으로 수행되므로, 수행되는 응용이 적은 모델에 적당하다. WSP/WTP 구현의 경우, 클라이언트는 컴퓨팅 능력이 제한되어있으므로 응용 프로세스 내에서 프로토콜이 수행되는 라이브러리 모델로 구현하고, 서버는 다수의 클라이언트 요구를 처리

하기 적당한 프로세스 모델로 구현될 수 있다.

WTP 계층에서 재전송을 통한 데이터의 신뢰성 있는 전달을 위하여 사용되는 타이머는 효율적인 관리를 위해 독립된 쓰레드(thread)로 구성할 수 있으며, 메시지 버퍼를 효율적으로 관리하여 서비스 데이터 단위(service data unit)로부터 프로토콜 데이터 단위(protocol data unit)를 생성하는 과정에서 메모리 복사를 최소한으로 줄이는 것이 바람직하다. 또한, 수신된 데이터를 일괄적으로 관리하기 위하여 일반적으로 독립된 형태의 수신 쓰레드를 구현한다. 그러나, 다수의 쓰레드를 지원하지 않는 이동단말에서 WSP/WTP를 구현 할 경우에는 하나의 프로세스 또는 쓰레드 내에서 모든 기능을 수행할 수 있도록 프로토콜이 구현되어야 한다. 이러한 것들이 결정되면, WSP/WTP 프리미티브를 이용하여 함수를 설계하고 구현한다.

3.3.2 응용 프로그래밍 인터페이스 함수

프리미티브는 함수와 유사한 형식을 갖고 있지만, 계층사이에서 서비스를 제공하기 위한 추상적 개념으로 함수와는 다르다. 함수는 프리미티브의 이벤트와 상관관계를 가지는데, 하나의 프리미티브 이벤트가 하나의 함수로 구현되기도 하고, 다수의 프리미티브 이벤트가 하나의 함수로 구현되기도 한다. 함수의 이름은 임의로 작성 가능하지만, 함수의 인자는 프리미티브의 매개변수 정보를 고려하여 설계되어야된다. <표 3>은 TR-Result 프리미티브를 함수로 나타낸다. tr_result_req 함수는 TR-Result.req와 TR-Result.cnf 이벤트를 함수로 구조화한 것이다. tr_result_req 함수 호출은 TR-Result.req 이벤트를 tr_result_req 함수 리턴은 TR-Result.cnf 이벤트를 의미한다 tr_result_req 함수의 인자는 TR-Result.req 및 TR-Result.cnf시 사용되는 매개변수로부터 구성한다. 실제로 구현할 수 있는 tr_result_req 함수의 예는 (그림 14)과 같다.

〈표 3〉 Tr-Result 프리미티브와 tr_result_req 함수

Primitive	TR-Result				tr_result_req api
Parameter	*req*	*ind*	*res*	*cnf*	
User Data	O	C(=)	-	-	TR-Result req
Exit Info	-	-	O	C(=)	TR-Result.cnf
Handle	M	M	M	M	

```
tr_result_req(msgb *sendpdu, u_short handle);
```

(그림 14) tr_result_req 함수

3.3 WAE

본 단락에서는 무선 인터넷 서비스를 제공하기 위한 WML 응용과 기존의 이동전화 서비스를 WAP에서 수용하기 위한 WTA 응용에 대해 알아본다.

3.3.1 WAE 컨텐트 형식

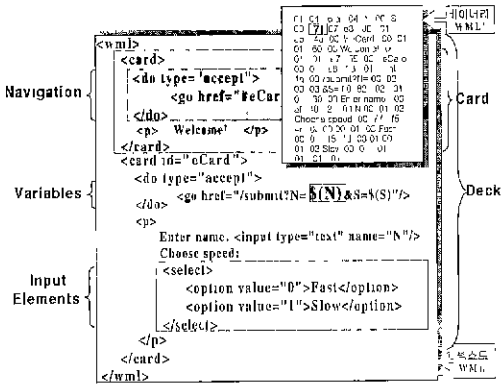
(1) WML과 바이너리 WML

WML은 협대역 무선 망과 제한된 자원을 갖는 이동단말에 적합하게 설계된 XML 기반의 무선 마크업 언어이다. 즉, WML은 무선 인터넷 서비스를 제공하기 위해 설계된 언어로 기존 HTML의 일부 기능에 무선 환경을 고려하여 새로운 기능을 추가하여 구성되었다.

WML은 다음과 같은 기능을 지원할 수 있는 요소들을 정의한다. 첫째, 텍스트와 이미지를 화면에 다양하게 보여줄 수 있는 방법을 제공한다. 둘째, 화면의 크기를 고려하여 정보를 여러 개의 단위로 나누게 되는데 이 단위를 카드(card)라 하며, 여러 개의 카드로 구성된 하나의 전송단위를 덱(deck)이라 한다. 따라서 카드는 선택 메뉴나 텍스트 화면과 같은 사용자와의 상호작용을 위한 단위로 네비게이션 단위가 되며, 덱은 URL로 구별되는 컨텐트의 전송 단위라는 점에서 HTML 페이지와 유사하다. 셋째, 덱 내에 있는 카드를 링크시켜 카드와 덱 사이의 이동을 명시적으로 관리할 수 있다. 또한 장치의 이벤트를 처리할 수

있으며 이는 네비게이션 목적으로 사용되기도 하고 스크립트를 실행하기도 한다. HTML에서와 같은 하이퍼텍스트 기능도 지원한다. 넷째, 하나의 변수를 여러 개의 카드에서 계속 사용할 수 있다. 예를 들면, 하나의 카드를 통해 받아들인 변수 값을 저장하고 있다가 다른 카드가 실행 될 때 이를 치환하여 사용하는 등 네트워크 자원을 효율적으로 사용할 수 있게 한다.

WAP 포럼에서는 이동단말과 WAP 게이트웨이 사이의 무선 선로에서는 바이너리 WML을 사용해 전송함으로써 전송되는 데이터의 양을 줄여준다. 바이너리 WML은 WBXML[12] 기반으로 WML 문서를 압축시킨 형태이다. WBXML은 WAP 포럼에서 무선 환경을 고려하여 정의한 XML과 같은 메타 언어로 모든 XML 문서에 적용 가능하다. WML의 모든 태그와 속성들은 16진수로 정의되어 있으며, 한 바이트로 표현된다. 변수의 경우는 덱 전체에서 같은 값을 가지게 되므로 바이너리 WML로 인코딩될 때는 실제 값은 한 번만 저장하고 변수가 나타나는 위치만을 명시해 준다.



(그림 15) WML 인코딩 예제

(그림 15)에서는 문서 선언이 생략된 WML 덱과 이를 바이너리로 압축한 예를 보여준다. 태그 <wml>의 바이너리 형식은 0x7F이고 바이너리 WML의 0x7F 이전에 표시된 값들은 WBXML 버전과 WML DTD의 공개 식별자, 문자 집합(디폴트로 UTF-8), 문자열 테이블들의 바이너리 값이다. 변수 N과 S는 문자열 테이블에 저장되고 실제 덱 내에서는 문자열 테이블을 참조한다.

(2) WMLScript

WMLScript는 무선 단말장치에 사용되는 스크립트 언어로 자바 스크립트의 부분집합에 일부 기능을 확장하여 구성되었다. 이는 무선 마크업 문서에 내장된 텍스트로서 WAP 게이트웨이가 바이트코드로 컴파일하여 이동단말에 전송하면, 이동단말의 마이크로브라우저가 바이트코드로 된 스크립트를 인터프리트하여 실행한다. 스크립트의 기능으로는 사용자 입력의 유효성을 검사하거나 장치의 기능을 사용할 수 있게 하고, 메시지나 다이얼로그를 이용하여 경고, 에러, 확인 메시지 등을 보여주는 기능을 제공한다. 또한 장치가 작동중일 경우에도 장치 소프트웨어에 대한 확장과 장치의 설정을 가능하게 한다. 이러한 기능들은 서버와의 통신 횟수를 줄임으로써 네트워크 자원의 불필요한 사용을 억제하는 역할을 한다.

(3) WBMP

WBMP(Wireless BitMaP) 이미지는 이동단말의 제한된 메모리를 사용하여 그래픽 정보를 표현함으로써 무선 응용 환경에 적합하도록 설계되었으며, 타입을 정의하는 필드에 제한을 두지 않음으로써 높은 확장성을 제공한다. WBMP는 헤더와 바디로 구성되어 있으며 헤더는 타입, 고정 헤더, 너비, 높이로 구성되어 있다[5].

JPEG, GIF, BMP 등의 이미지를 WBMP 이미지로 변환하는 WBMP 변환기의 전체 동작은 다음과 같다. 이미지의 비트 깊이, 팔레트 사용 여부, 가비지 값 유무 등을 고려하여 픽셀의 RGB 값을 추출한다. 이렇게 추출한 값을 이용하여 픽셀의 명암 단계 값을 계산하고, 이미지 배경과 물체의

명암 단계 값 평균이 같아지도록 임계값을 결정한다. 임계값이 결정되면 픽셀의 명암 단계 값과 다시 비교하여, 명암 단계 값이 작으면 검은색으로 표현하고 크면 흰색으로 표현한다.

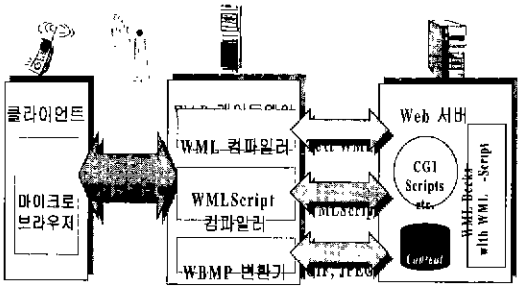
(4) WTA 서비스

WTA는 WAE 계층에 속한 서비스 중 하나로서 전화서비스를 WAP으로 수용하기 위한 프레임워크를 제공한다. WTA는 WML 사용자 에이전트의 기능과 이동 전화 장치에서 가능한 모든 서비스들을 수용할 수 있는 구조를 제안하고 있다.

3.3.2 WAE 응용

(1) WML 사용자 에이전트

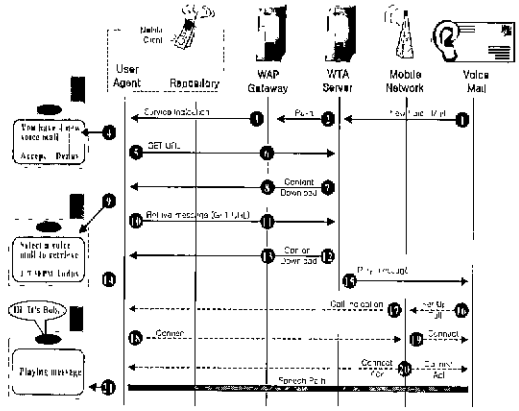
WML 사용자 에이전트는 WAP 프로토콜을 기반으로 무선 인터넷 서비스를 제공하기 위한 응용 프로그램을 가리킨다. 이러한 응용으로는 이동단말에서 바이너리 WML 문서 및 WBMP 이미지를 브라우징하기 위한 마이크로브라우저, WWW 컨텐트를 WAP 컨텐트로 변환하기 위한 WML 컴파일러 및 WBMP 변환기 등을 들 수 있다. (그림 16)은 이러한 에이전트들 사이의 동작과정을 보여준다.



(그림 16) WAP 프로그래밍 모델

WAP은 무선 인터넷 서비스를 지원하기 위한 개방 통신 규약으로 인터넷망과 이동통신망의 접속점에 게이트웨이를 두어 두 프로토콜 체계 사이의 중계역할을 하도록 한다. 이때, 무선 도메인과 WWW 사이를 연결하기 위해 프락시 기술을 이용한다. 보통 WAP 프락시는 프로토콜 게이트웨이와 컨텐트 인코더 두 가지 기능으로 이루어져 있다. 프로토콜 게이트웨이는 WAP 프로토콜 스택에서의 요청을 인터넷 프로토콜 스택에 적합하도록 변환한다. 컨텐트 인코더는 WAP 컨텐트를 네트워크상의 데이터 크기를 줄이기 위해 압축된 바이너리 형식으로 변환한다. 만일, 이동단말의 마이크로브라우저에서 인터넷상에 위치한 서버에 있는 WML 문서를 요청하면, WAP 게이트웨이는 기존의 인터넷 프로토콜을 사용하여 요청한 텍스트 WML 문서를 가져와서 바이너리 형식으로 변환시켜서 마이크로브라우저로 보낸다. 이러한 동작구조는 이동단말 사용자로 하여금 다양한 WAP 컨텐트와 응용을 이용할 수 있게 하며, 응용 개발자에게는 WWW 기술을 사용하여 무선 컨텐트 서비스와 응용을 개발할 수 있도록 해준다. WAP 컨텐트와 응용은 WWW의 형식과 유사하며 이동단말기에 위치한 마이크로브라우저 역시 이러한 컨텐트를 브라우징하는데 있어 표준 웹 브라우저와 유사하게 동작한다.

(2) WTA 사용자 에이전트



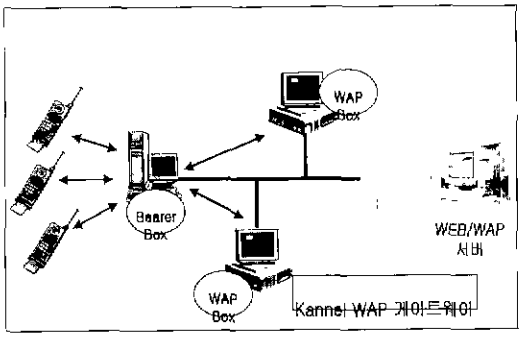
(그림 17) WTA를 이용한 음성 사서함 관리 예

WTA 사용자 에이전트는 기존의 무선 인터넷 서비스를 위한 WML 사용자 에이전트의 기능에 이동전화 장치에서 가능한 모든 서비스들을 수용할 수 있는 기능을 제공한다. (그림 17)은 음성 사서함을 관리하는 예를 통하여 WTA의 동작원리를 나타낸다. 사용자는 음성 사서함에 새로운 음성 메시지들이 있음을 알게되고, 서버로부터 메시지들의 리스트를 받아 온다. 이 리스트는 클라이언트의 화면에 나타나게 되고, 사용자가 그중 하나를 선택하면 서버는 클라이언트와 전화연결을 하여 선택한 음성 메시지를 듣는다. 이때, WTA 서버는 컨텐트를 제공하게 되는데, 사용자에게 제공되어질 서비스를 WML 문서로 생성하게 되고 사용자는 URL을 사용해 문서를 가져온다. WAP 게이트웨이는 WSP/HTTP 프로토콜을 변환하고, WTA 서버와 클라이언트를 중계하는 역할을 한다.

4. WAP 게이트웨이 구축

4.1 Kannel WAP 게이트웨이

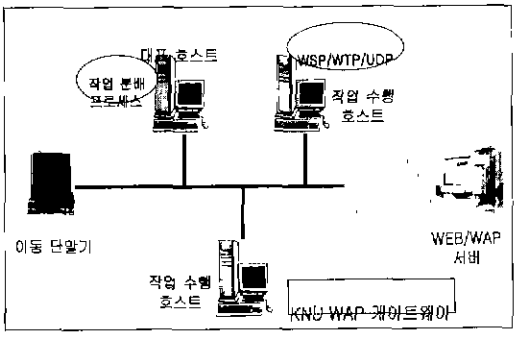
Kannel WAP 게이트웨이는 소스가 공개되어 있으며 SMS 서비스 위주의 게이트웨이에서 WSP/WTP 프로토콜을 포함한 WAP 박스(Box)의 기능을 추가하여 2000년 5월 현재 버전 0.9가 발표되었다.



(그림 18) Kannel WAP 게이트웨이 네트워크 구성도

(그림 18)은 Kannel WAP 게이트웨이 구성도이며 SMS 기능은 생략되었다. 베어러 박스(bearer box)는 이동단말의 UDP 데이터그램이 수신되면 WAP 박스로 적절히 부하를 분산시킨다. WAP 박스는 WSP/WTP 프로토콜 스택을 가지며 이동단말로부터 WML 문서 요구를 처리하기 위하여 웹 서버로 HTTP 요구 및 응답을 수행한다. 웹 서버로부터 전달된 WML 문서는 WAP 박스에서 바이너리 WML 문서로 변환된 후 베어러 박스를 통하여 전달된다. 그러나 Kannel WAP 게이트웨이는 모든 데이터가 하나의 베어러 박스를 경유하여 송·수신되는 구조이므로 동시에 수행하는 클라이언트의 수가 일정수준 이상으로 증가하면 베어러 박스에서 병목현상이 발생하여 전체 WAP 게이트웨이 센터의 처리능력에 문제가 생길 수 있다[13].

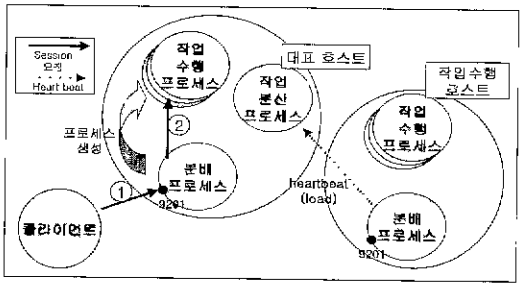
4.2 KNU WAP 게이트웨이



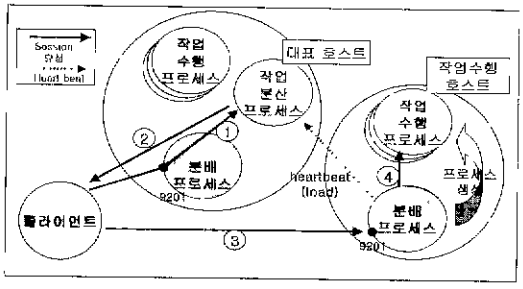
(그림 19) KNU WAP 게이트웨이 네트워크 구성도

(그림 19)는 경북대학교 컴퓨터통신·이동컴퓨팅연구실에서 개발한 KNU WAP 게이트웨이의 구성요소들과 부하 분산 기능을 수행하는 네트워크 구성도를 보여준다. 이동단말기에는 UDP 상에서 구현된 WSP/WTP 프로토콜과 그 위에서 동작하는 마이크로브라우저가 탑재되어 있다. KNU

WAP 게이트웨이는 WSP/WTP 프로토콜, WML 컴파일러, WBMP 변환기, 그리고 HTTP 연동 모듈로 구성되어 있으며, 하나의 대표 호스트와 다수의 작업 수행 호스트가 있다. 대표 호스트는 WSP의 연결 재 설정 기능을 사용하여 집중된 부하를 분산시키는 작업을 수행하고, 작업 수행 호스트는 분배된 작업을 수행한다.



(a) 클라이언트 요구가 대표 호스트에서 수행되는 경우



(b) 클라이언트 요구가 작업 수행 호스트에서 수행되는 경우

(그림 20) KNU WAP 게이트웨이의 내부 프로세스 구성도

(그림 20)은 KNU WAP 게이트웨이의 프로세스 구성도를 나타낸 것이다. 클라이언트는 대표 호스트의 9201포트로 접속을 시도한다. 대표 호스트의 분배 프로세스는 자신의 부하 정도를 감지하여 일정한 수준 이하일 경우 대표 호스트에서 작업을 수행하며, 그 이상일 경우에는 Connect 메시지를 작업 분산 프로세스로 유도하여 다른 호스트에서 작업이 수행되도록 한다. 작업 분산 프로세스는 WSP의 연결 재 설정 기능을 사용하여 클라이언트가 부하가 낮은 호스트로 접속을 하도록 유도한다. 이러한 부하 분산 기능은 게이트웨이의 확장성을 높이고 동시에 수행되는 클라이언트의 수가 많을 경우 빠른 응답 속도를 보장한다. 그리고, 작업 수행 프로세스에는 WSP/WTP 프로토콜 스택이 있으며 이동단말로부터 WML 문서 요구를 처리하는 기능을 한다.

KNU WAP 게이트웨이의 WSP/WTP 프로토콜 스택은 WAP 스펙 1.2의 모든 기능을 제공하며, WML 문서를 바이너리 WML 문서로 변환시켜주는 WML 컴파일러와 JPG, GIF 및 BMP 이미지 파일을 WBMP 이미지 형식으로 변환시켜주는 WBMP 변환기가 구현되어 있다. 또한, HTTP와 WSP를 연동하기 위한 모듈도 구현되어 있다.

5. 결 론

본 논문에서는 WAP의 전체적인 구조에 대하여 소개하였고, WAE/WSP/WTP의 주요 기능에 대하여 간략히 기술하였다. 또한, WSP/WTP 프로토콜 구현 시에 고려해야 할 사항들과 응용 프로그래밍 인터페이스 설계에 대하여 언급하였고, 마지막으로 Kannel과 KNU 게이트웨이를 비교 분석하며 게이트웨이 구축 기술에 대하여 기술하였다.

현재 무선 인터넷 서비스를 지원하기 위하여 WAP 포럼, W3C 및 마이크로소프트사 등에서 독자적인 표준을 제안하고 있다. 그 중에서 WAP 표준을 제안하고 있는 WAP 포럼에는 전 세계적으로 영향력 있는 통신 및 컴퓨터업체들이 참여하고 있으며, 활동도 활발하여 무선 인터넷 서비스를 위한 산업 표준이 될 가능성이 높다. 따라서, 향후 무선 인터넷 서비스를 위한 범용 프로토콜로써 WAP이 일반화 될 것에 대비하여야 할 것이다.

참고문헌

[1] "Wireless Application Protocol Architecture Specification," WAP Forum, November 8, 1999.
URL: http://www.wapforum.org/

[2] RFC2068, "Hypertext Transfer Protocol-HTTP/1.1," R.Fielding, et al., January 1997.
URL: ftp://ftp.isi.edu/in-notes/rfc2068.txt

[3] "Wireless Markup Language," WAP Forum, November 8, 1999. URL: http://www.wapforum.org/

[4] "Wireless Markup Language Script," WAP Forum, November 8, 1999. URL: http://www.wapforum.org/

[5] "Wireless Application Environment Specification," WAP Forum, November 8, 1999.
URL: http://www.wapforum.org/

[6] "Wireless Session Protocol Specification," WAP Forum, November 8, 1999.
URL: http://www.wapforum.org/

[7] "Wireless Transaction Protocol Specification," WAP Forum, November 8, 1999.
URL: http://www.wapforum.org/

[8] "Wireless Transport Layer Security Protocol Specification," WAP Forum, November 8, 1999.
URL: http://www.wapforum.org/

[9] "Wireless Datagram Protocol Specification," WAP Forum, November 8, 1999.
URL: http://www.wapforum.org/

[10] "Wireless Telephony Application Specification," WAP Forum, November 8, 1999.
URL: http://www.wapforum.org/

[11] "Wireless Telephony Application Interface Specification," WAP Forum, November 8, 1999.
URL: http://www.wapforum.org/

[12] "Binary XML Content Format Specification," WAP Forum, November 8, 1999.
URL: http://www.wapforum.org/

[13] Kannel 홈페이지, http://www.kannel.org/