

이동 환경에서 웹서비스를 위한 WAP Gateway의 디자인과 구현 이슈

강성훈*, 한영태*, 민덕기*

한국대학교 컴퓨터·정보통신공학과

e-mail : {shkang,ythan,dkmin}@cse.konkuk.ac.kr

Design and Implementation Issues of Scalable WAP Gateway for Web Services at Mobile Environment

Seonghun Kang*, Youngtae Han*, Dugki Min*

Dept. of Computer Science and Engineering, Konkuk University

요약

PCS, Digital Cellular, CDPD, 그리고 Handheld PC와 같은 휴대단말기를 사용해 각종 정보망의 서비스를 받으려는 움직임이 점차 커지고 있다. 이에 발 맞추어 일반 이동단말기에서 인터넷을 접속 할 수 있는 여러 기술들이 개발되었으며, 현재 사용하는 것으로 Unwired Planet사가 개발한 HDML(Handheld Device Markup Language)기술이 있고, WAP Forum에서 제시한 WAP(Wireless Markup Language)가 있다. WAP의 구조에서 가장 핵심적인 요소로서 WAP Gateway가 있으며, 현재까지 WAP Gateway에 대한 특별한 표준화가 이루어지지 않은 단계에서 각 컨벤트 사업자가 WAP Gateway를 나름대로 구현하고 사용하는 실정이다. 본고는 보다 확장성을 가지고 무선이라는 환경에 적합하도록 서비스를 개선할 수 있는 WAP Gateway의 설계 및 구현 이슈에 대하여 다룬다.

1. 서론

1992년 말부터 등장한 Web은 급 가속적인 기술의 발전과 함께 상업화, 대중화, 멀티미디어화 되고 있으며 가히 인터넷 세상이라고 할 정도로 현재 정보흐름의 반 이상을 인터넷이 차지하고 있다 해도 과언이 아니다. 인터넷이 이만큼 발전하는 동안 이동 통신 기술 역시 발전에 발전을 거듭하여 현재 국내인구의 절반에 가까운 1천 8백만 인구가 휴대전화기를 사용하는 수준에까지 이르렀다. 이런 환경으로 말미암아 인터넷이라는 정보물결을 일반적인 데스크 탑(desktop)으로 뿐만 아니라 이동성을 가진 각각의 단말기를 즉, PCS, Digital Cellular, PDS, Handheld PC등을 통해 제공받으려는 사용자의 욕구 역시 날로 커지고 있다.

그러나 이동 단말기의 제한된 특성 즉, 중앙처리장치와 메모리의 지용량성과 통신상태의 고르지 못한 때문에 기존의 웹 기반 언어인 HTML(Hyper Text Markup Language)로서는 이런 제한된 제약을 가진 휴대용 이동장비에서 많은 정보를 효율적으로 나타낼 수 없다[1]. 이에 대한 해결책 중 하나가 Unwired Planet(현재 Phone.Com)에서 개발된 HDML(Handheld Device Markup Language)[2]로서 1997년 5월 W3C(World Wide Web Consortium)에 휴대용 무선장비의 웹 접근을 위한 표준언어로 제출되었다. 이에 후속으로 Ericsson, Motorola, Nokia, Unwired Planet등 움직인 세계 유수의 기업들이 WAP(Wireless Application Protocol) Forum[3]을 구성하여 이동장비에서의 웹 접근을 위한 표준제정을 위해 HDML을 기반으로 하는 WML(Wireless Markup Language)[4]를 포함한 무선응용프로토콜(WAP)의 표준 제정을 추진하고 있다. 이에 발 맞추어 전 세계적으로 WML/HDML을 이용한 여러 응용프로그램의 제작과 연구가 급속도로 증가하고 있다. 따라서 WAP에 관련한 연구는 향후 다가올 정보화 사회의 구축을 위한 매우 중요한 일이라 할 수 있을 것이다.

본 고에서는 휴대 이동단말기를 통해 웹 접근을 가능하게 하는 무선 응용 프로토콜 WAP(Wireless Application Protocol)에서 가장 중요한 요소로 다루어지고 있는 WAP Gateway를 효율적으로 설계 및 구현하기 위한 이슈를 세 가지를 들어 제시한다. 첫째, 접속환경이 불안

정한 이동 통신에서 서버와 클라이언트간 Queue된 Session Manager를 사용함으로써 네트워크, 할라언트 및 서버의 통신 실패에 대하여 유용적으로 대처할 수 있는 설계를 제시했으며, 둘째로 WAP의 Push 기능을 적극 활용한 Agent기능을 도입함으로써 사용자가 보다 활성화된 서비스를 제공받을 수 있는 구조를 제시하였다. 그리고 마지막으로 확장성을 가질 수 있도록 하기 위하여 여러 WAP Gateway를 하나처럼 이용 가능한 Load Balancing구조[5]를 포함한 새로운 WAP Gateway구조를 제시한다.

이 논문은 2장에서 WAP의 대략적인 기능을 설명하고, 3장에서 WAP구조 중 가장 중요시되는 WAP Gateway의 구조와 기능을 설명하고, 4장에서 Gateway의 설계와 구현을 위한 이슈들을 다룬 뒤 마지막으로 5장에서 결론을 내리도록 하겠다.

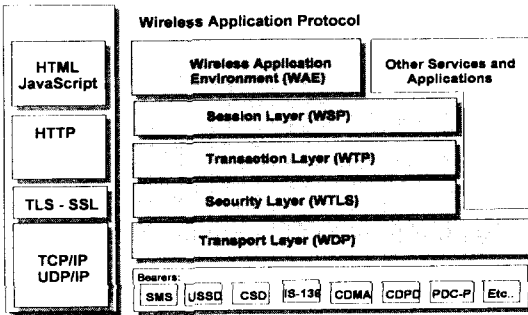
2. WAP(Wireless Application Protocol)

WAP은 이동전화와 그 밖의 다른 무선 터미널 상에서 무선정보와 전화서비스의 표현과 전달을 위한 사실상의 세계표준 이라고 할 수 있으며, 무선 통신 네트워크 상에서 동작하는 응용서비스를 개발하는데 있어서 유용한 기술을 위한 산업표준 스펙을 촉진하기 위한 WAP forum의 연구 결과이다[3]. WAP 모델에서는 이동 단말기(Client)와 인터넷 서버 사이에 WAP Proxy라 불리는 WAP Gateway를 두도록 하고 있다. WAP Gateway를 통하여 이동 단말기와 웹 서버간의 모든 통신이 이루어진다.

2.1 WAP Protocol Architecture

아래의 <그림 1>는 WAP Protocol의 구조와 기존 인터넷 프로토콜을 비교해 놓은 것으로 이 프로토콜 구조가 WAP을 지원하는 단말기와 WAP Proxy에 탑재되어야 한다[1,3]. 그림에서 보듯 WAP은 이동 통신 환경인 Bearer 계층[6], TCP/IP의 TCP에 해당되는 WTP(Wireless Transaction Protocol) 계층[7], 보안을 지원해 주는 WTLS(Wireless Transport Layer security) 계층[8], HTTP/1.1에 상응하는 기능 및 세션용 정의하고 관리하는 WSP(Wireless Session Protocol) 계층[9], 그

리고 WML로 이루어진 문서와 각종 응용서비스가 작동하게 되는 WAE(Wireless Application Environment) 계층[10]으로 이루어져 있다.



<그림 1> WAP Protocol Architecture

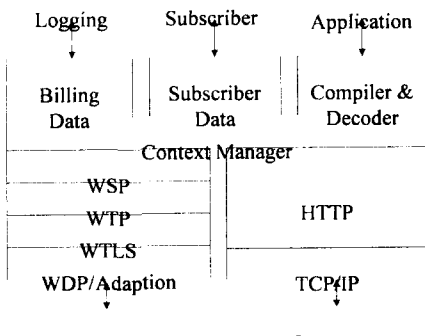
2.2 WML(Wireless Markup Language)

WML은 일반적인 이동단말기에서 HTML의 모든 페이지 구성요소를 볼 수 없음을 따라 HDML(Handheld Device Markup Language)의 후속으로 이동 단말기에 적합하도록 최적화된 Markup Language이다 [4]. WML의 특징은 그림과 텍스트 모두 지원할 수 있다는 것이며 HTML과 마찬가지로 강조기호(Bold, 이탤릭)를 지원하는 것이다. 그밖에 사용자의 입력 폼, 즉 text와 password entry를 지원하며, URL을 사용한 네비게이션 메카니즘을 제공하며 히스토리 관리기능을 지원한다.

3. WAP Gateway의 기능과 구조

WAP의 전반적인 구조에 있어서 가장 중요한 역할을 담당하고 있는 것이 바로 WAP Gateway이다. 이는 2.1에서 설명된 WAP Protocol Stack을 Web Protocol Stack 즉, HTTP와 TCP/IP로 전환하는 기능을 수행한다[1,3,11]. WAP Gateway 구조를 <그림 2>에 나타내었다.

WAP Gateway의 역할은 크게 다음과 같은 다섯 가지로 나타낼 수 있다. 첫째, Origin Server에 대한 Proxy Server로서의 역할이다. 둘째, 무선 Bearer에 대한 WAP 서비스 즉, connection-oriented와 connectionless의 두 가지 방법을 모두 제공한다. 셋째, WML과 WMLScript를 지원하며 간단한 HTML의 변환을 지원한다. 넷째로 WAP 서비스에 접근하는 여러 어플리케이션들에 대하여 선택적으로 WAP Security를 지원한다. 그리고 마지막으로 데이터의 Push기능을 제공한다. 그밖에 사용자가 WAP에 접근한 시점으로부터 과금 서비스를 개시하는 기능을 담당하고 수행할 수 있으며, 각 사용자의 데이터를 보관하고 각 사용자에 따른 차별적인 서비스를 지원할 수 있다[11].



<그림 2> WAP Gateway Architecture

4. WAP Gateway의 설계 및 구현 이슈

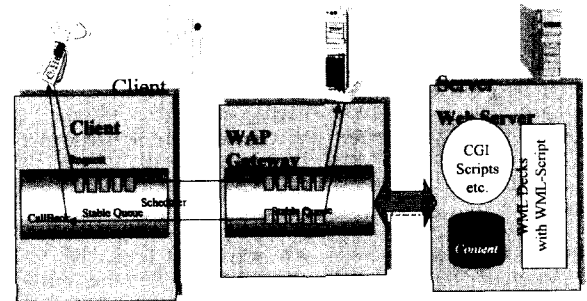
4.1 Queue를 이용한 이동 단말기(Client)와 WAP Gateway 간 통신

이동 통신망의 특성 중 하나는 고속이동 등 위치 가변에 따른 통신 연결이 불안정하다는 것이다. 즉, Session이 활성화된 상태를 지속적으로 유지하기가 힘들다. 이에 따라 WAP에서는 WSP 계층에서 Session의 이동에 따른 상태정보를 가지고 있으며 Session의 보류와 재시작을 담당하게 된다.

여기서 고려하고자 하는 사항은 이동 단말기(Client)와 WAP Gateway간 Connection을 수행할 때 Queue를 이용하게 되는 방법을 말한다. 즉, 사용자의 요구와 이에 따르는 응답을 Queue를 도입함으로써 응용프로그램의 동작상태가 접속이 끊기더라도 계속 유지할 수 있도록 하는 것이다. 이는 MIT에서 개발된 ROVER Toolkit[12]의 동작 개념을 응용한 것으로써, 네트워크의 연결이 끊긴 후 재접속이 시도될 때 상호 교환되는 클라이언트(이동 단말기)와 WAP Gateway간의 요구와 응답으로부터, 수행중인 응용프로그램이 non blocking될 수 있도록 하기 위하여 WAP Gateway측으로부터 Call back을 계속 받아 통신하도록 하는 메카니즘이다. 즉, 기존의 이동 통신에서의 Data 서버가 클라이언트와 서버간 응용프로그램이 실행중인 상태에서 접속이 끊길 때 재 접속하여 실행 중이던 응용프로그램을 실행하기 위해서는 처음부터 다시 시작해야하는 번거로움이 있었으나 Queue를 도입함으로써 재접속 후에도 지속적인 응용프로그램의 수행을 실현할 수 있다. Queue를 통한 이동 단말기와 WAP Gateway간의 동작을 <그림 3>에 나타내었다. 그림의 구조에 따른 동작을 살펴보면 다음과 같다.

- 1) 모바일단말기(Client)로부터의 요구를 Client측 Queue에 저장한다.
- 2) 이때, Queue는 FIFO로 Gateway에 요구를 전달한다.
- 3) Gateway는 Client로부터의 요구를 입력측 queue에 저장한다.
- 4) 요구에 대한 응답을 출력측 queue에 저장한다.

이들 통하여 네트워크, Client 및 서버의 실패에 대하여 보다 유연적으로 대처할 수 있다.



<그림 3> Queue를 이용한 Client - Gateway간 통신

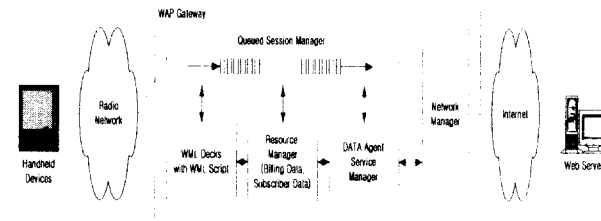
4.2 Agent 기능을 이용한 Data Push/Pull 서비스

이는 WAP의 Push와 Pull 기능을 적극 활용하여 사용자에게 활성화된 서비스를 제공하기 위하여 Agent기술[14]을 도입하는 측면으로서의 설계이다. 분산 시스템에 있어서 Agent란 상호협력력을 통하여 수행되는 컴퓨터 프로그램이라 할 수 있다. 이 정의의 바탕으로 생각하면 기존의 컴퓨터 프로그램은 자신의 고유기능만 처리한다고 볼 수 있다. 즉, 우리가 사용하는 통신프로그램에서 네일이 전달되었을 때 이동 단말기에 문자 메시지를 보내준다거나 하는 등의 기능을 Agent를 통해 수행할 수 있다. Agent를 도입함으로써 사용자가 필요로 하는 요구사항 즉, 웹 검색 등에 소요되는 접속시간을 줄일 수 있으며, 부가적으로 접속에 소요되는 비용문제를 줄일 수 있다.

여기서는 두 가지의 Agent를 도입하려 한다. 첫째는 Client측면에서 사용자 나름대로 요구사항을 구성할 수 있는 Agent를 들 수 있다. 예를 들어 사용자가 차표예매 등에 대한 정보를 원할 경우 이를 등록

해 놓음으로써 Gateway가 등록된 정보를 찾아줌과 동시에 찾았을 때는 SMS서비스와 함께 WML문서를 등록자에게 보내주어(Push) 예매를 보다 편리하게 할 수 있는 구성이다. 둘째는 이미 정해진 Agent를 서버 측에서 가지고 있어서 사용자가 그 Agent에 등록하면 등록된 사용자 모두에게 변경되는 내용을 전달해 주는 서비스이다. 이는 증권정보, 부동산 정보 등에 이용될 수 있으며, 서버에서 클라이언트로 데이터를 보내줄 때(Push) 역시 WML문서를 함께 보냄으로써 구입여부와 시세에 따른 추가정보도 제공할 수 있을 것이다.

위의 4.1과 4.2에서 제시한 두 가지의 고려사항을 통해 대략적인 WAP Gateway의 구조를 <그림 4>에 나타내었다. Session Manager에서는 4.1에서 언급한 Session을 관리하는 기능과 요구 응답을 관리하는 기능을 수행한다. 뿐만 아니라, History기능과 직접적으로 무선 네트워크에 연결되는 WAP 프로토콜이 실행되는 Block이다. Resource Manager에서는 과금데이터, 사용자 데이터 등 Resource를 관리하는 기능을 수행한다. DATA Agent Manager에서는 4.2에서 언급한 사용자에게 의한 Push Item의 설정기능과 데이터 Push기능을 수행한다. 마지막으로 Network Manager가 웹 서버와 연결되는 HTTP 프로토콜과 보안 기능(SSL/TLS)을 수행하며 Internet과의 connection을 관리한다.

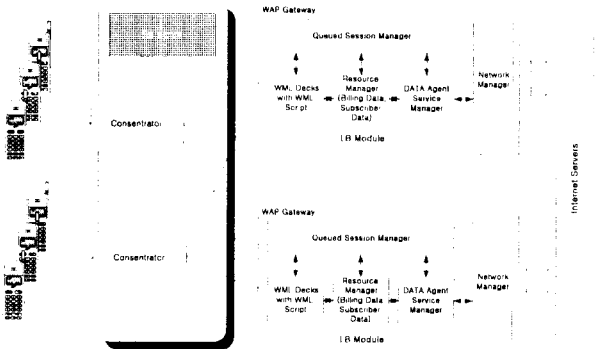


<그림 4> The Proposed WAP Gateway Architecture

4.3 Scalable WAP Gateway Architecture

위에서 제시한 WAP Architecture의 구조 변경을 최소화함과 동시에 Scalable한 WAP Gateway의 구조를 갖게 하기위해 현재 연구중인 Load Balancing 구조를 접목시킨 Gateway의 구조를 <그림 5>에 나타내었다.

이 구조는 이미 설계되어 있는 WAP Gateway의 변경 없이 WAP Gateway 소프트웨어와 Hardware만을 추가하여 확장시킬 수 있는 구조를 구현할 수 있도록 한다[5]. 즉, 이미 구현된 Gateway를 재사용한다는 측면에서 중요성을 들 수 있다. <그림 5>는 Load Balancing Service를 지원하는 위한 WAP Gateway의 구조를 보여주고 있다.



<그림 5> The Scalable WAP Gateway Architecture

1) Concentrator : 이동 단말기로부터의 연결을 관리한다. 즉, 부하가 적은 Gateway로의 연결을 담당한다.

2) Selector : Concentrator가 여러 개 존재할 경우 Gateway의 Port상태를 관리한다. Gateway의 정보가 변경될 경우 이 사항을 Concentrator들에게 알린다. 또, 이동 단말기로부터 요청이 들어 올 경우 적당한 Concentrator를 이동 단말기에 알린다.

3) LB Module : 기존의 게이트웨이를 위한 일종의 Agent로서 Gateway를 대신하여 Selector와 통신하는 Module이다. 기존의 Gateway가 Concentrator와 Selector에 대한 정보를 갖지 않고 구현되었을 것이므로 이러한 LB Module을 통하여 WAP Gateway의 존재를 Selector와 Concentrator에 알리고 시스템의 현재상태 즉 성능 및 부하 여부를 진단하여 Load Balancing 서비스를 실행하도록 한다.

5. 결론

본 논문에서는 이동 단말기를 통하여 웹에 접근을 가능하게 하는 무선 응용 프로토콜인 WAP에서 가장 중요하게 다루어지는 WAP Gateway를 사용자에게 보다 효율적인 서비스를 제공할 수 있도록 다음의 세가지 WAP Gateway의 설계에 대한 제안을 하였다. 첫째는 접속환경이 불안정한 이동 통신에서 서버와 클라이언트간 Queue된 Session Manager를 사용함으로써 네트워크, 클라이언트 및 서버의 통신 실패에 대하여 유동적으로 대처할 수 있는 설계를 제안하였으며, 둘째는 WAP의 Push기능을 활용한 Agent기능을 도입함으로써 사용자가 필요로 하는 각각의 웹 검색 등의 요구사항과, 이를 실행하는데 소요되는 접속시간 및 이에 따르는 비용문제를 줄일 수 있는 설계방안의 제안이다. 그리고 마지막으로 위에 제시된 WAP Gateway의 구조를 이용하여 확장성을 가질 수 있도록 하기 위하여 여러 WAP Gateway를 하나처럼 이용하면서 각 Gateway에 걸리는 부하에 따른 성능을 해결하기 위하여 부하를 이동시키는 즉, 실행되고 있는 세션을 부하가 적은 다른 Gateway로 옮기는 Load Balancing구조를 포함한 새로운 WAP Gateway구조를 제안하였다.

6. 참고문헌

[1] WAP forum, "WAP White Paper", June, 1999. URL: <http://www.wapforum.org/>
 [2] W3C, "Handheld Devices Markup Language Specification", URL : <http://www.w3c.org/TR/NOTE-Submission-HDML-spec.html>
 [3] WAP Forum, "Wireless Application Protocol Architecture Specification", April 30, 1998. URL: <http://www.wapforum.org/>
 [4] Phone.Com Incorporation, "WML Language Reference v1.0". URL: <http://www.phone.com>, January, 1999.
 [5] 이동훈,한영태,민덕기, "확장성있는 서버 시스템을 구현하기 위한 부하 균등화 서비스에 대한 연구", Technical Report ku-dms-2000 c 1, 건국대학교 컴퓨터정보통신공학과, 2,2000.
 [6] WAP Forum, "Wireless Datagram Protocol", April 30, 1998. URL: <http://www.wapforum.org/>
 [7] WAP Forum, "Wireless Transaction Protocol", April 30, 1998. URL: <http://www.wapforum.org/>
 [8] WAP Forum, "Wireless Transport Layer Security", April 30, 1998. URL: <http://www.wapforum.org/>
 [9] WAP Forum, "Wireless Session Protocol", April 30, 1998. URL: <http://www.wapforum.org/>
 [10] WAP Forum, "Wireless Application Environment Specification", April 30, 1998. URL: <http://www.wapforum.org/>
 [11] International Engineering Consortium, "Mobile-Originated Example of WAP Architecture", 1999, URL: <http://www.webproforum.com/wap/>
 [12] Anthony D. Joseph, Joshua A. Tauber, and M. Frans Kaashoek, "Mobile Computing with the Rover Toolkit", IEEE Transactions on Computers: Special issue on Mobile Computing, February, 1997.
 [13] Joseph P. Bigus and Jennifer Bigus, "Constructing Intelligent Agents with Java", Wiley, 1997, pp.181-191.