

패킷헤더축소 방법을 이용한 WAP 무선통신 효율 개선

강동우*, 하병훈**, 박기현**
*계명대학교 컴퓨터공학과
e-mail : {hooni76,khp}@kmu.ac.kr

WAP Wireless Communication Efficiency Improvement Using a Packet Header Reduction Method

Dong-Woo Kang*, Byoung-Hoon Ha**, Ki-Hyun Park**
*Dept of Computer Science, Kei-Myung University

요 약

최근 무선인터넷 서비스를 위한 다양한 기반기술들이 연구되고 있으며, WAP(Wireless Application Protocol)도 그런 기술들 중의 하나이다. 그러나, WAP의 제안에서는 데이터 인코딩 및 디코딩을 제외하고는, 무선통신의 효율성 문제를 심도있게 다루지는 못했다. 본 논문에서는 제한적인 환경을 가지는 WAP 이동무선통신망에서의 효율향상을 위한 방법의 일환으로써, 패킷헤드축소 방법을 이용하였다. 이 방법은 WAP 에이전트들에서 실행되도록 설계하고 구현하였다. 실험 결과, WAP 게이트웨이만을 이용한 데이터 전송량에 비하여 본 논문에서 구현한 패킷헤드축소 방법을 이용하면, 데이터 전송량을 17% 내지 25% 가량 줄일 수 있음을 알 수 있었다.

1. 서론

WAP 게이트웨이는 유선 인터넷과 이동무선통신 단말기의 일종인 WAP 단말기 사이의 다리 역할을 하며 WAP 무선통신 네트워크와 인터넷 사이에 위치하는 소프트웨어이다. 즉 기존의 HTTP/TCP/IP 기반의 패킷과 WSP/WTP/WDP 기반의 패킷간의 상호변환을 담당함으로써, 기존의 유선 네트워크도 이동무선 통신 환경의 인프라로 사용할 수 있게 한다[2-4,9]. 그러나 WAP 게이트웨이는 프로토콜 변환에 중점을 두고 있으므로 무선통신의 낮은 대역폭과 통신 효율성 문제에 대한 고려가 매우 부족하다. 따라서 본 논문에서는 이동무선통신의 데이터 전송량을 줄여 효율적인 통신이 되도록 WAP 에이전트들을 설계 구현하여, 패킷헤드축소 방법을 수행할 수 있게 한다.

본 논문에서 제안한 에이전트들은 WAP 단말기에

서 동작하는 CSA(Client Side Agent)와 WAP 게이트웨이에서 동작하는 SSA(Server Side Agent)로 구성된다. 각 에이전트는 패킷헤더축소(packet header reduction)를 이용한 데이터 전송량 축소를 시도함으로써, 이동무선통신 환경에서의 통신효율 향상에 기여한다.

본 논문의 구성은 모두 5절로 되어 있으며 2절에서는 WAP 게이트웨이의 구성에 대해서, 3절에서는 전송 데이터 축소 및 압축을 이용한 기존의 통신효율 향상 연구에 대해서 살펴본다. 4절에서는 CSA 및 SSA의 설계 및 패킷헤드축소방법에 대하여 설명하고, 5절에서는 실험을 통한 실제 동작과 결과를 알아본다. 마지막 6절에서는 본 논문의 결론과 향후 연구과제를 제시하면서 마무리한다.

2. WAP 게이트웨이의 구성

본 연구팀이 이미 제작한 WAP 게이트웨이는 WAP 포럼의 제안을 충실히 따르고 있으며, 그림 1과 같이 구성되어 있다[2].

** 본 과제(결과물)는 산업자원부의 출연금 등으로 수행한 지역전략산업 석박사 연구인력 양성사업의 연구결과입니다.

- WSP(Wireless Session Protocol) 서버 : WAP 게이트웨이 시스템이 이동 WAP 단말기에 대해서 WAP 서버 역할을 할 수 있도록 WSP 형식에 따라 단말기와 응답과 요청을 수행한다. WSP 서버는 WAP 브라우저로부터의 요청을 받아 메시지의 헤더 분석과 캐퍼빌리티(capability) 협상 등을 수행하며, 웹서버에 접근할 수 있도록 하기 위하여 프로토콜 변환기로 메시지를 전송한다.

- WTP(Wireless Transaction Protocol) 응답기 (responder) : WDP의 데이터그램 서비스 위에서 실행되며, 비교적 간단한 트랜잭션을 위한 프로토콜이다. 3개의 전송 클래스를 지원하며, Invoke PDU, Result PDU, Acknowledgement PDU, Abort PDU 등의 PDU들이 있다.

- HTML 필터 : 유선 인터넷에 있는 기존의 웹문서들도 이동 WAP 단말기를 이용한 검색이 가능하도록 하기 위하여, HTML 문서를 WML(Wireless Markup Language) 문서로 바꾼다. 그런데, 본 연구에서 테스트 환경으로 사용하는 Phone.com 에뮬레이터[8]의 한계 때문에, 본 연구팀이 개발한 HTML 필터는 일정한 크기(약 200bytes)의 카드(card)들로 분할된 WML 문서로 변환시킨다[2].

- 프로토콜 변환기 : 프로토콜 변환기 부분은 WSP와 HTTP 프로토콜의 상호변환을 수행한다. 이동 WAP 단말기에서 요청한 WSP 요청 메시지를 HTTP 요청 메시지로 변환을 하며 반대로 웹서버에서 오는 HTTP 응답 메시지를 WSP 응답 메시지로 변환한다.

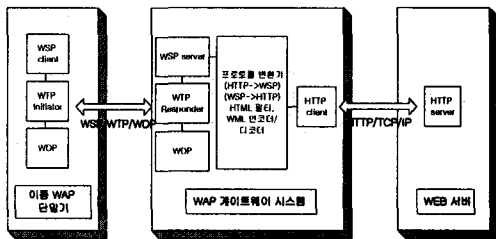


그림 1 WAP 게이트웨이의 구성

는 크게 데이터 압축방법[1,6,7,9]과 데이터 축소방법[4-6]으로 나눌 수 있다. 데이터 압축방법은 패킷의 데이터를 gzip, compress, zip 등과 같은 기존의 압축방법을 이용하여 데이터 전송량을 줄이는 방법이다. 이에 비해서 데이터 축소방법은 패킷의 자주 변경되지 않는 부분을 보관한 캐쉬를 이용함으로써, 이전에 보내어진 데이터와 현재 보내려는 데이터간의 차이만을 전송할 수 있도록 하여 데이터 전송량을 줄이는 방법이다. 전자의 방법이 데이터의 표현방법을 통한 축소라면, 후자의 방법은 일부 데이터의 전송 불필요에 의한 축소라고 할 수 있다.

본 연구는 데이터 축소방법을 사용하여 WAP 환경에 적용하되, 통신 프로토콜 의존 방법으로 패킷의 헤더부분 중에서 자주 변화되지 않는 부분을 캐싱(caching)하여 이전 전송과 동일한 부분을 전송하지 않는 방법을 사용하며, 본 논문에서는 패킷헤더 축소 방법이라고 표현한다.

4. CSA 및 SSA의 설계

본 절에서는 이동 WAP 단말기와 WAP 게이트웨이간의 데이터 전송량을 줄이는 방법의 일환으로서 구현한 CSA(Client Side Agent)와 SSA(Server Side Agent) 모듈에 대하여 설명한다. CSA와 SSA는 위에서 이미 설명한 구축된 WAP 게이트웨이에 구현되며, 캐쉬를 이용한 패킷헤더축소를 사용한다. 그림 2는 CSA 및 SSA와 기존 구성요소들과의 관계를 나타낸다. CSA와 SSA는 이동 WAP 단말기와 WAP 게이트웨이 시스템의 WDP에 각각 연결되어서 WSP/WTP/WDP를 이용한 통신을 수행한다.

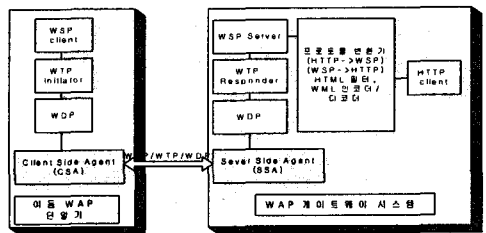


그림 2 CSA 및 SSA의 위치

3. 연구 동향

데이터 전송량 축소를 통한 통신효율 향상의 연구

4.1 헤더 데이터 축소 방법

CSA와 SSA의 패킷헤더축소 방법은 아래의 알고리즘과 같이 동작한다. 축소 대상이 되는 헤더부분은 문서작성 언어 혹은 문자표현 코드 등을 전달하는 부분으로서, 일반적으로 자주 바뀌지 않고 고정되게 전달되는 부분이다.

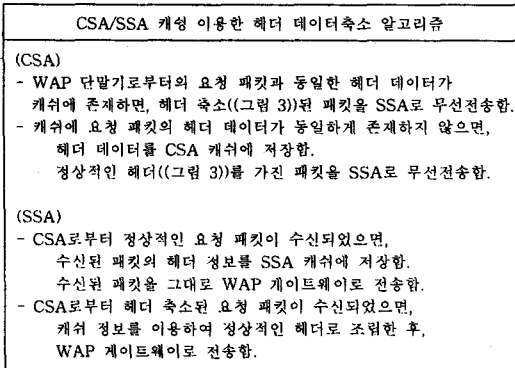


그림 3의 왼쪽은 CSA에서 SSA로 전송되는 정상적인 패킷의 헤더 포맷이며, 그림 3의 오른쪽은 축소된 헤더 포맷을 보여준다. 축소된 부분은 헤더 데이터의 시작에서 41byte짜부터 47bytes 만큼이며, 정상적인 헤더 데이터 크기(87bytes)의 54% 가량 축소된다. SSA에서 CSA로의 응답 패킷의 경우, 서버 관련정보가 들어가는 필드(16bytes) 만큼 축소가 이루어지며, 정상적인 헤더 데이터 크기(76bytes)의 21% 가량 축소된다.

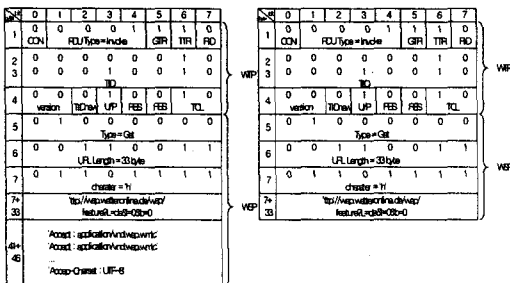


그림 3 CSA에서 SSA로 전송되는 정상 패킷헤더(왼쪽)와 축소된 패킷헤더(오른쪽)

5. 구현 및 실험

5.1 CSA와 SSA의 구현

그림 4는 본 논문에서 구현한 WAP 게이트웨이 시스템 및 CSA/SSA를 위한 개발 및 실험환경을 보여준다. Windows 2000 서버에는 Phone.com의 인터넷폰 에뮬레이터(T250)가 설치되어 WAP 단말기 역할을 하며, CSA가 구현되어 있다. WAP 게이트웨이 시스템과 SSA는 레드햇 리눅스 7.0의 서버에 설치되었는데, WAP 게이트웨이는 이전에 본 연구팀이 개발한 것을 사용하여 SSA와의 원활한 연결을 꾀하였다. 비록, Windows 2000 서버와 리눅스 서버가 유선으로 연결되어 있으나, Windows 2000 서버의 인터넷폰 에뮬레이터가 WSP/WTP/WDP 프로토콜만을 인식하므로, 이동무선통신 환경이 조성되었다고 할 수 있다.

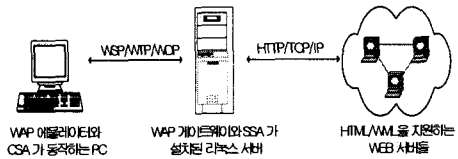


그림 4 구현 및 실험 환경

5.2 실험 및 실험 결과

구현된 CSA 및 SSA의 성능 테스트를 하기 위해 무선 인터넷 사이트들인 www.bbc.co.uk, wap.wetteronline.de, www.buzzed.uk 및 wap.pfft.net 등을 동일한 시나리오로 검색하였다. wap.wetteronline.de는 날씨정보를 제공하는 WAP 사이트이며, wap.pfft.net는 잡담 등을 다루는 WAP 사이트이다. 또한, www.buzzed.uk는 WAP 커뮤니티 사이트이며, www.bbc.co.uk는 영국 BBC 방송의 WAP 사이트이다.

WAP 게이트웨이가 패킷헤더축소 방법을 이용하여 위에서 언급한 4개의 무선인터넷 사이트들을 검색했을 때, 데이터 전송량의 축소 효과를 그림 5에 나타내었다. 데이터 전송량의 결과는 SSA쪽의 UDP 포트를 통하여 주고받는 패킷의 양을 의미한다. 그림에서 보듯이, www.bbc.co.uk 검색에서의 전송량은 25.5%, wap.wetteronline.de 검색에서의 전송량은 25.4% 만큼 상대적으로 줄어든 것을 볼 수 있다. 또한, www.buzzed.uk 검색에서는 25.4%, wap.pfft.net 검색에서는 17.1%의 축소 효과를 얻을 수 있다.

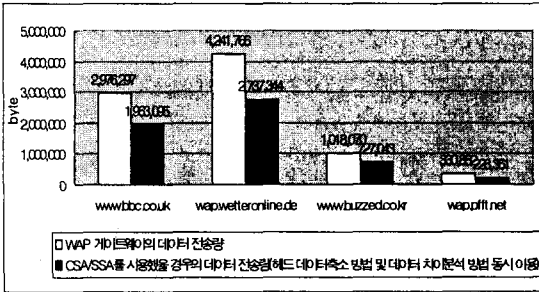


그림 5 패킷헤드축소 방법에 의한 데이터 전송량 축소 효과

6. 결론 및 향후 연구과제

본 논문에서는 WAP 게이트웨이와 에이전트를 이용하여 이동무선통신망에서의 통신효율을 증가시키는 방법을 제시하고, 시스템을 구현하여 실험결과를 비교 분석하였다. 에이전트들은 패킷헤드축소를 이용한 데이터 전송량 축소를 시도함으로써, 이동무선 통신 환경에서의 통신효율 향상에 기여하도록 구현하였다. 실험 결과에서 WAP 게이트웨이만을 이용한 방법에 비하여 CSA/SSA를 같이 이용한 방법에서 데이터 전송량이 17% 내지 25% 가량 축소됨을 확인할 수 있었다.

향후 연구과제로는 구축한 WAP 게이트웨이를 이용한 실험이 보다 광범위하게 이루어져야 하며, 특히, CGI 이외의 다른 전송방법들에 대해서도 데이터 차이분석에 의한 축소방법을 개발할 필요가 있다고 생각한다. 이외에도, 에이전트 사용이 보안문제에 미치는 영향에 대해서도 아울러 검토해야 할 것이다.

참 고 문 헌

[1] 김기조, 이동근, 임경식, "압축 기법을 이용한 WSP의 기능 확장과 성능 평가," 정보과학회논문지 컴퓨팅의 실제, 제29권, 제5호, pp.543-552, 2002년 10월.
 [2] 박기현, 강동우, 권정선, "HTML 필터 기능을 갖춘 WAP 게이트웨이 시스템 구축," 정보과학회논문지 컴퓨팅의 실제, 제7권, 제4호, pp.350-358, 2001년 8월.
 [3] C. Arehart, Professional WAP, WBOX Press, July 2000.
 [4] Hari Balakrishnan, Srinivasan Seshan, Elan Amir and Randy H. Katz, "Improving TCP/IP Performance over Wireless Network," Proc. 1st

ACM Intl' Conf. on Mobile Computing and Networking(Mobicom), November 1995.

[5] Barron C. Housel, George Samaras, David B. Lindquist, "WebExpress: A client/intercept based system for optimizing Web browsing in a wireless environment," Mobile Networks and Applications volume3 Issue 4, December 1998.

[6] Jeffrey C. Mogul, Fred Douglass, Anja Feldmann, and Balachander Krishnamurthy, "Potential Benefits of Delta Encoding and Data Compression for HTTP," DEC Western Research Lab., Dec. 1997.

[7] Eetu Ojanen, Jari Veijalainen, "Compressibility of WML and WML Script byte code: Initial Results," Proc. of the 10th Intl' Workshop on Research Issues in Data Engineering, Feb. 2000.

[8] Openwave System Inc., UP.SDK Developer's Guide-UP.SDK Release 4.1, <http://developer.phone.com/html/doc/41/devguide/index.html>.

[9] Perkins, Stephen J. and Mutka, Matt W., "Low-Bandwidth Access: An Evaluation of Application Level Protocol Compressibility," Proc. of the 4th Intl' Conf. on Telecommunication Systems, pp.97-108, 1996.

[10] WAP Forum, "Wireless Application Protocol Architectures Specification," SPEC-WAPArch, April 1998.