

모바일 분산 인터미디어리 기반 웹 서비스를 위한 클라이언트/서버 메시지 모델의 설계

김인화* 황준* 이응재**

◆ 목 차 ◆

- | | |
|--------------------|------------------------|
| 1. 서론 | 4. 모바일 웹 서비스의 메시지 처리모델 |
| 2. 인터미디어리 기반 웹 서비스 | 5. 시뮬레이션 |
| 3. 컨버전스 기반 웹 서비스 | 6. 결론 |

1. 서론

웹 서비스(Web Service)는 기존의 WWW를 기반으로 하는 분산처리 기술로, 다수의 다른 기종 플랫폼간의 상호 운용성을 향상 시켜주며, 낮은 진입장벽, Just-in-time(JIT) 통합, 웹 서비스 사이의 협력이 실행 시에 동적으로 바인딩 되는 등의 우수한 특징을 가지고 있다. 웹 서비스는 기존의 시스템의 복잡성은 감소시키면서도 유연성과 확장성을 가지고 있는 제 2세대 웹이라고 볼 수 있다. 즉, 기존의 1세대 웹의 브라우저 기반, 사용자 중심, 어플리케이션 지향의 성격을 지닌 URL을 통한 수동적 의미의 접근에서 어플리케이션 기반, 사용자와 어플리케이션 중심의, 서비스 지향의 자동적인 웹의 이용이 가능한 제 2세대의 웹 서비스 시대가 도래 하고 있는 것이다[1].

웹 서비스는 혁신적인 XML 기술을 기반으로 기존 1세대의 웹 환경을 이용한 분산 컴퓨팅을 가능케 함으로써 웹을 통한 시스템 통합을 가능하게 해주며, 이를 통하여 DCOM, CORBA, JAVA RMI 등의 기존 분산 컴퓨팅 모델들을 대체하는 새로운 기술로 인정받고 있다. 현재 이러한 웹 서비스는 XML을 기본 기술로 하여 SOAP, WSDL,

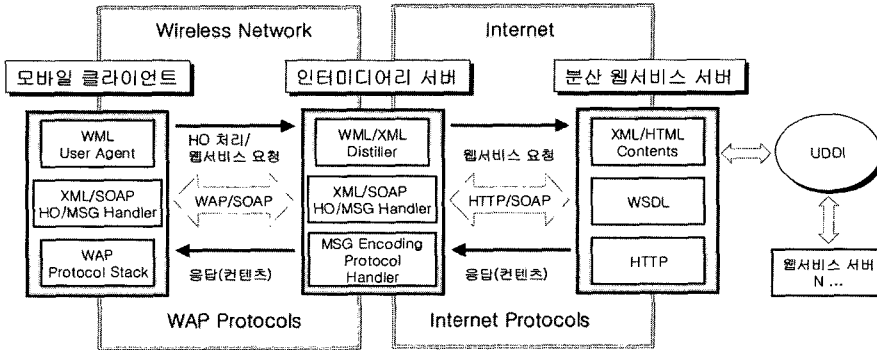
UDDI 등의 상호 연동된 기술들로 발전을 거듭해가고 있다. 이를 통하여 사용자는 클라이언트 기종에 관계없이, 접근 데이터 서버의 플랫폼 등의 관계없이 더욱 편리하게 자동화된 웹을 이용할 수 있게 된다. 이에 현재 국내에서는 혁신적인 웹 서비스 기술을 잘 구축된 기존의 우수한 모바일 환경에 접목시키고자 하는 움직임 일어나고 있다. 기존의 유선 서비스에 한정된 웹 서비스는 인터넷을 통하여 서버와 클라이언트가 연결된 구조, 혹은 다수의 중개자 서버들이 추가된 분산처리 네트워크 형태로 구성되어 있었다. 여기에 이동식 장치, 가령 휴대폰, 스마트 폰, 차량용 네비게이터, DMB 장치 등의 모바일 클라이언트들이 추가되면서 모바일 웹 서비스가 이루어진다. 따라서 본 논문의 목적은 무선 환경에서의 웹 서비스를 가능하게 하는 메시지 프로토콜을 설계하고 모바일 기기 특성상 이동시 발생하는 핸드오프 상황과 프로토콜 변환을 해결하기 위해 XML/SOAP 메시지 처리 모델을 제시하였다.

2. 인터미디어리 기반 웹 서비스

본 논문에서 제안하는 인터미디어리 기반 웹 서비스를 위한 클라이언트/서버(C/S) 모델은 그림 1에서처럼 기본적으로 모바일 웹 서비스 인터미디어리라는 서버와 무선 통신 상의 모바일 클라이언트로 구성된다.

* 서울여자대학교 대학원 석사과정

** 서울여자대학교 정보미디어대학 교수



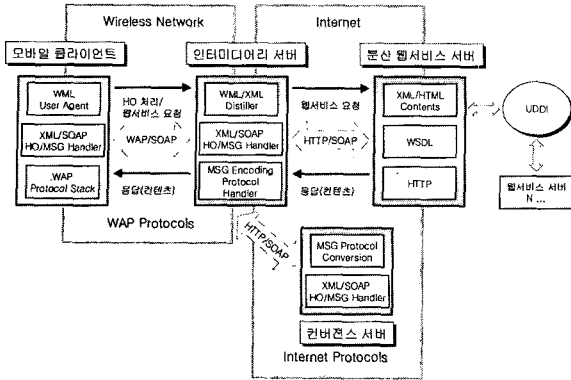
(그림 1) 인터미디어리 기반 무선 웹 서비스

모바일 웹 서비스를 위한 무선 웹 서비스 인터미디어리는 사용자 단말기(핸드폰, 스마트폰, 네비게이터 등)와 WAP/Web-Service 분산 서버들 사이에 위치하여 유선 웹 서비스 환경에서 제공되는 각종 콘텐츠들과 동적인 웹 서비스를 모바일 웹 서비스 사용자에게 제공하기 위해 HTTP/SOAP 프로토콜과 해당 WAP 프로토콜 사이의 변환 기능 및 XML과 WML 문서의 변환/인코딩 기능 등을 제공하는 분산 미들웨어 시스템으로 정의할 수 있다 [2,3]. 여기에서 모바일 기기의 특성상 이동시 핸드오프가 발생한다. 핸드오프(Handoff HO-채널 이동 전환) 처리란 이동 클라이언트 장치 가입자가 한 무선 구역에서 다른 무선 구역으로 이동해 갈 때, 현 통신 채널을 다른 무선 구역의 통신 채널로 자동적으로 전환해 줌으로써 통신이 끊어지거나 QoS (Quality of Service)가 낮아지지 않고 양질의 통신 서비스가 사용자에게 계속 제공되게 하도록 하는 기능이다. 모바일 웹 서비스에서는 이러한 서비스가 인터넷상의 웹 서비스 서버와 프락시로 사용되는 인터미디어리 서버를 통한 모바일 클라이언트 장치에서의 웹 서비스를 말한다[4]. 모바일 클라이언트가 무선 웹 서비스를 받던 중에 해당 인터미디어리 서버와의 통신 유효 Cell 영역을 벗어나 새로운 Cell 영역으로 진입하면서 핸드오프 처리가 시작된다. 모바일 클라이언트는 새 통신 유효 Cell 영역에 연결된 새 인터미디어리 서버와 새로운 통신 연결(Session)을 설정하고, 모바일 웹 서비스 핸드오프 처리 메시지를 새 인터미디어리에 전송한

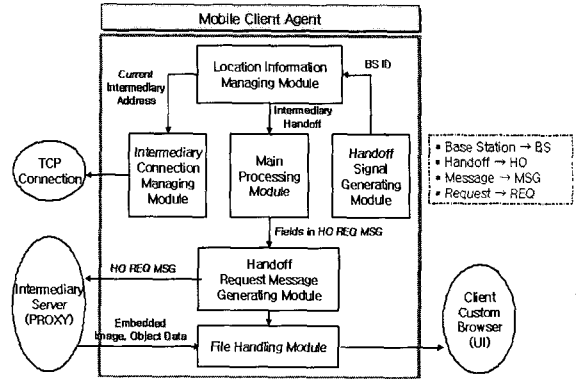
다. 이 메시지를 기반으로 새 인터미디어리 서버는 해당 웹 서비스 서버에 HTTP 요청 메시지를 보내고, 계속해서 해당 모바일 클라이언트가 모바일 웹 서비스를 새로운 인터미디어리 서버의 통신 연결을 통해서 받을 수 있도록 허가 응답을 보내준다.

3. 컨버전스 기반 웹 서비스

앞에서 설명한 일반적인 핸드오프 처리 과정을 통해서 모바일 웹 서비스의 QoS를 핸드오프 발생했을 때에도 지속적으로 유지하는 것이 가능해진다. 그러나 모바일 클라이언트가 하나의 통신 유효 Cell 영역에서 다른 Cell 영역으로 이동했을 때, 새로운 영역의 연결 세션을 담당하는 인터미디어리 서버와 이전 영역의 인터미디어리 서버와의 메시지 인코딩 프로토콜이 다른 경우가 발생할 수 있다. 왜냐하면 이기종의 서버 시스템에 이기종 운영체제를 사용하기 때문인데, 이는 사업자의 통신 기기 설치 여건과 통신업체와 서버업체의 이익 배분 구조와 연관이 있기 때문이다. 이러한 경우 앞에서 설명한 기본적인 핸드오프 처리 과정으로는 핸드오프 처리가 원활하게 이루어지기 어렵다. 따라서 본 논문에서는 이러한 모바일 웹 서비스 핸드오프 처리 방법에 인터미디어리 서버간의 프로토콜 차이를 조정해주는 컨버전스 서버를 추가한 핸드오프 처리 방법을 도입하였다. 컨버전스 서버가 추가된 모바일 웹 서비스의 핸드오프 처리 모델의 개념을 그림 2에 나타내었다.



(그림 2) 컨버전스 기반 무선 웹 서비스



(그림 3) 모바일 이동 클라이언트의 모듈 세부 구성

그림 2에서 볼 수 있듯이 컨버전스 서버는 몇 개의 기지국(Base Station; BS)들과 연결된 인터미디어리 서버와 같은 통신 연결세션을 담당한 인터미디어리 서버들과 연결되어 있는 상위 서버이다. 컨버전스 서버에 연결된 모든 인터미디어리 서버는 자신의 메시지 인코딩 프로토콜 방식과 데이터 타입을 사전에 컨버전스 서버의 레지스트리에 등록시켜야 한다. 즉, 컨버전스 서버는 이러한 등록 정보를 기반으로 프로토콜을 중재해준다.

4. 모바일 웹 서비스에서 메시지 처리 모델

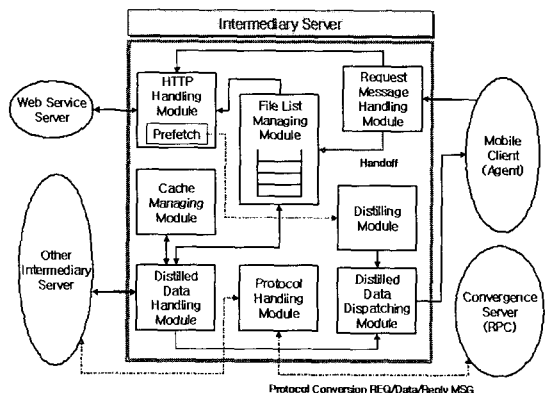
4.1 모듈 구성도

모바일 웹 서비스에서의 핸드오프 처리를 위한 C/S 메시지 처리 모델은 크게 모바일 클라이언트 모듈(MC), 인터미디어리 서버 모듈(IS), 컨버전스 서버 모듈(CS)의 세 가지 모듈들로 구성되어 있다. 그림 3에서는 모바일 이동 클라이언트의 모듈 세부 구성을 볼 수 있다.

하위 모듈들의 연계 동작 흐름을 보면, 먼저 Handoff Signal Generating 모듈은 Location Information Managing 모듈을 통하여서 모바일 클라이언트가 기존 무선통신 셀 영역을 벗어나서 새로운 셀 영역으로 진입하는 것, 즉 핸드오프가 발생하게 되면 기지국 BS(Base Station) ID를 이용하여 이전 인터미디어리 서버를 통하여 이루어졌던 웹 서

비스에 핸드오프가 발생되었음을 Main Processing 모듈에 알려 처리하도록 한다. 또한 현재 인터미디어리 서버의 주소를 Intermediary Connection Managing 모듈을 통해 확보한다. Handoff Request Message Generating 모듈에서는 Main Processing 모듈로부터 핸드오프메시지 생성에 대한 정보를 받아 HO REQ MSG를 만들어 새로 연결 세션이 성립된 인터미디어리 서버에 핸드오프처리 요청 메시지를 전송한다. 그리고 FileHandling 모듈에서 요청에 대한 응답 메시지들이 인터미디어리 서버로부터 전송되면 해당 데이터를 디코딩 하여 모바일 클라이언트의 고유 브라우저에 데이터를 넘긴다.

그림 4에서는 모바일 분산 인터미디어리 서버의 일반적인 모듈 구성을 보여주고 있다. Request Message Handling 모듈은 모바일 클라이언트 에이



(그림 4) 인터미디어리 서버의 일반적인 모듈 구성

전트로부터 일반 웹서비스 관련 메시지나 혹은 핸드오프 처리 요청 메시지를 수신하여 분류 처리한다. 일반 웹서비스 관련 메시지는 HTTP Handling 모듈에서 인터넷으로 연동된 해당 웹서비스 서버와의 HTTP 통신 기반 XML/SOAP 메시지 송수신으로 연결된다. 그리고 해당 웹서비스 서버로부터 수신 받은 웹서비스의 데이터는 Distilling 모듈에서 모바일 형식(WML)으로 정제되어 Distilled Data Dispatching 모듈을 통하여 해당 요청을 한 모바일 클라이언트로 전송된다. 만약 핸드오프 처리 요청 메시지라면 File List Managing 모듈로 분기되어 모바일 클라이언트(MC)가 이전 인터미디어리 서버(IS)에서 재전송 받아야 할 해당 웹서비스 파일 리스트 등의 데이터를 리스트에 저장시킨다. 그리고 이전 Distilled Data Handling 모듈에서 이전 인터미디어리 서버와의 핸드오프 처리를 시도하며, 그 결과로 받은 데이터를 Distilled Data Dispatching 모듈을 통해 해당 모바일 클라이언트에 WAP/WML 기반으로 전송한다. 여기서 만약 이전 인터미디어리 서버의 메시지 인코딩 프로토콜이 다르다면, Protocol Handling 모듈에서는 SOAP 메시지를 통하여 컨버전스 서버(CS)의 프로토콜 중재 모듈을 원격 호출함으로써 이전 인터미디어리 서버와의 프로토콜 중재를 요청한다. 컨버전스 서버에서 응답된 프로토콜 중재 메시지를 이용하여 핸드오프 처리 메시지를 만들어 통신하게 되면 이전 인터미디어리 서버로부터 손실 없는 모바일 웹서비스 데이터를 전송 받을 수 있게 된다. 그림 5에서는 모바

일 웹서비스의 프로토콜 중재를 통한 효율적인 핸드오프 처리를 위한 컨버전스 서버의 세부 모듈 구성을 보여준다.

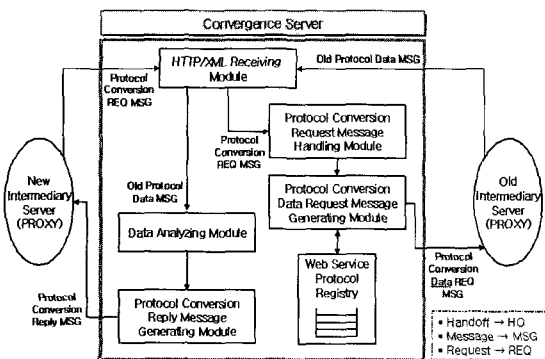
새 인터미디어리 서버로부터 프로토콜 변환 메시지(SOAP/XML)를 HTTP/XML Receiving 모듈을 통하여 전송 받아 Protocol Conversion Request Message Handling 모듈로 분기함으로써 해당 프로토콜 중재 메소드가 실행된다.

Web Service Protocol Registry에서 이전 인터미디어리 서버의 메시지 인코딩 프로토콜을 검색한다. 그리고 현재 인터미디어리 서버의 프로토콜 중재 요청 데이터를 이용하여 Protocol Conversion Data Request Message Generating 모듈에서는 프로토콜 변환 데이터 요청 메시지를 만들어 이전 인터미디어리 서버에 전송한다. 이전 인터미디어리 서버로부터 HTTP/XML Receiving 모듈로 이전 프로토콜 데이터 메시지가 수신되면 Data Analyzing 모듈에서는 이전 인터미디어리 서버에서 전송된 메시지를 분석하여 핸드오프 처리를 위한 프로토콜을 중재한다. 그리고 이전 인터미디어리 서버의 인코딩 프로토콜 등의 정보를 가진 프로토콜 변환 응답 메시지를 생성하여 프로토콜 중재를 요청했던 새로운 인터미디어리 서버에 전송한다.

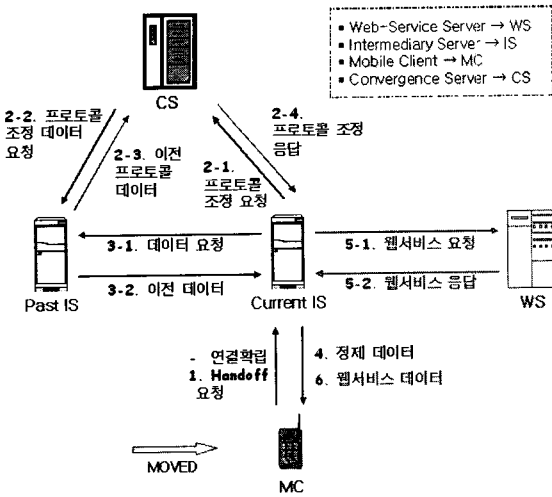
4.2 메시지 처리 모델

프로토콜 조정을 고려한 모바일 웹 서비스 핸드오프 처리를 위한 클라이언트/서버(Client/Server; CS) 메시지 처리 모델은 그림 6과 같은 XML/SOAP 메시지들의 요청/응답 흐름으로 구성되어 있다.

그림 6에서 모바일 클라이언트(Mobile Client; MC)가 모바일 웹 서비스를 받던 중에 해당 인터미디어리 서버(Intermediary; IS)와의 통신 유효 Cell 영역을 벗어나 새로운 Cell 영역으로 진입하면서 핸드오프 처리가 시작된다. MC는 새 통신 유효 Cell 영역에 연결된 Current IS와 새로운 통신 연결을 시도하고, 모바일 웹 서비스 핸드오프 처리 메시지를 Current IS로 전송한다(1). 이 메시지를 기반으로 Current IS는 컨버전스 서버(Convergence Server; CS)에 프로토콜 조정 요청 메시지를 보내



(그림 5) 컨버전스 서버의 모듈 구성



(그림 6) XML/SOAP 메시지 처리 모델

고(2-1) CS에서는 Past IS에 데이터를 요청하는 메시지(2-2)를 보내 Past IS와 MC 사이에서 전송된 콘텐츠 데이터를 요구하게 된다. 이에 대한 응답으로 파일 리스트가 담긴 메시지(2-3)를 전송하게 된다. CS에서는 Past IS의 인코딩 스타일을 알려주고 (2-4) 이를 통해 현재 MC와 연결된 Current IS는 Past IS에 데이터를 요청(3-1)하게 된다.

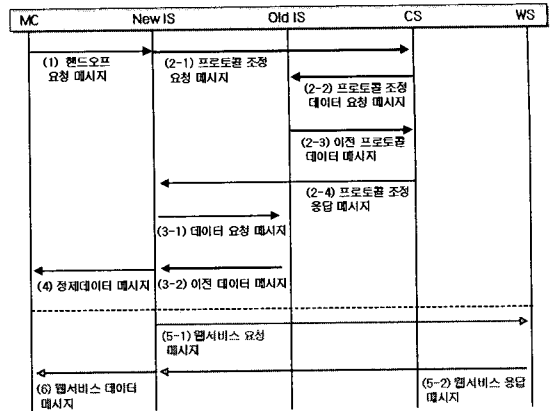
그림 7은 그림 6에서 나타난 메시지 처리 모델에 대한 시퀀스 다이어그램이다.

4.3 설계된 메시지 프로토콜

인터미디어리 서버간 프로토콜 조정을 포함한 모바일 웹서비스에서의 핸드오프 처리를 위한 XML/SOAP 클라이언트/서버 메시지 처리 모델에서 사용되는 XML/SOAP 메시지들의 내용을 다음과 같은 표 1에 나타내었다.

5. 시뮬레이션

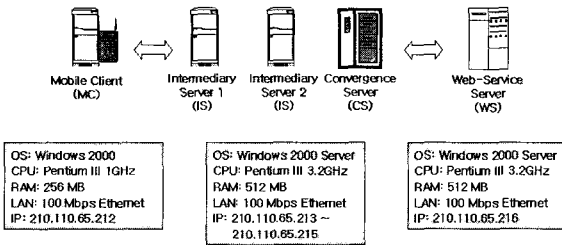
모바일 웹서비스 클라이언트/서버 메시지 처리 모델의 시뮬레이션은 모바일 클라이언트 에이전트로써의 휴대용 단말기, 인터미디어리로서의 프락시 서버들, 프로토콜 조정 컨버전스 서버, 그리고 인터



(그림 7) 프로토콜 조정을 포함한 모바일 웹 서비스 핸드오프 처리 시퀀스

(표 1)

메시지	메시지 내용	전송 방향	구분
1. 핸드오프 요청 메시지	Total Size (2 Bytes) Past Proxy IP Address (4 Bytes) Convergence Server Address (4 Bytes) Past Proxy Protocol Type (2 Bytes) User Request URL (n Bytes) List of Files (n Bytes) Received Bytes (4 Bytes)	MC → new IS	
2-1. 프로토콜 조정 요청 메시지	Total Size (2 Bytes) Past Proxy IP Address (4 Bytes) Past Proxy Sub Address (n Bytes) Past Proxy Protocol Type (2 Bytes) Current Proxy Protocol Type (2 Bytes)	new IS → CS	(프로토콜이 다르다면 2-2번, 같으면 3-1번으로 분기함)
2-2. 프로토콜 조정 데이터 요청 메시지	Total Size (2 Bytes) Number of Files (1 Byte) File name × (32 Bytes) Expired Time × (8 Bytes) 씩	CS → old IS	
2-3. 이전 프로토콜 데이터 메시지	Total Size (2 Bytes) List of Files (n Bytes) Size of Previous Data (4 Bytes) Previous Format Type (32 Bytes) Dispatch Method URI (n Bytes)	old IS → CS	
2-4. 프로토콜 조정 응답 메시지	Total Size (2 Bytes) List of Files (n Bytes) Size of Previous Data (4 Bytes) Previous Format Type (32 Bytes) Dispatch Method URI (n Bytes) Previous Encoding Style URI (n Bytes)	CS → new IS	
3-1. 데이터 요청 메시지	Total Size (2 Bytes) Number of Files (1 Byte) File name × (32 Bytes) Expired Time × (8 Bytes) 씩 Current Encoding Style URI (n Bytes)	new IS → old IS	
3-2. 이전 데이터 메시지	Total Size (2 Bytes) File name (32 Bytes) Size of Data (2 Bytes) Data (n Bytes)	old IS → new IS	(반복)
4. 정제 데이터 메시지	Total Size (2 Bytes) File name (32 Bytes) Size of Distilled Data (2 Bytes) Distilled Data (n Bytes)	new IS → MC	(반복)
7. 오류 응답 메시지	Fault Code (n Bytes) Fault String (n Bytes) Fault Factor (n Bytes) Detail (n Bytes)	IS, CS, WS → IS, CS, WS	(옵션)



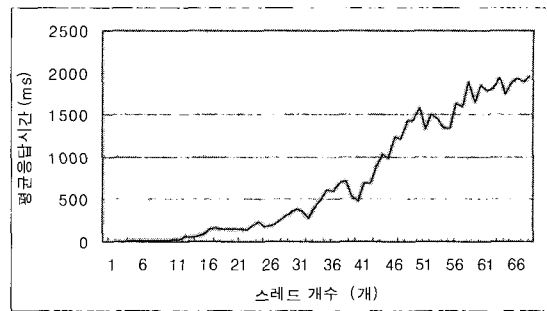
(그림 8) 시뮬레이션 환경 사양

넷으로 연결된 웹 서비스 서버들이 상호 통신하는 것이라고 볼 수 있다. 그러나 휴대용 단말기를 WAP 모바일 클라이언트로 사용하여 실험하는 것은 무선 통신망 라우터인 CSD(Circuit Switch Datagram) 라우터를 직접 사용해야 하는 어려운 점이 있다. 그러므로 본 논문의 시뮬레이션에서는 노키아 WAP 툴킷, WinWAP 32, Phone.com UP32 등의 PC용 모바일 에이전트 시뮬레이터를 사용하는 것이 좋을 것이다.

인터미디어리 서버(프락시, WAP 게이트웨이)로는 이전 Cell 영역 용과 새 Cell 영역 용의 두 대의 PC가 필요하고 컨버전스 서버와 시험용 웹 서비스 서버에도 각각 한 대씩의 PC가 필요할 것이므로, 모바일 에이전트 시뮬레이터용 PC와 함께 고정 IP로 인터넷에 연결된 PC가 최소한 5개 이상은 필요하다.

일반적으로 서버 측 프로그램들은 적은 수의 요청에 대해서는 모두 안정적인 작동을 할 것이다. 하지만 동시에 많은 서비스 요청이 서버 측에 발생하는 경우에는 그 처리 과정에서의 오류 증가로 인해 프로그램의 기능이 중단되거나 혹은 프로그램 자체가 종료되는 경우도 많이 발생하게 될 것이다. 하지만 각 서버 측의 프로그램들은 자신의 처리 한도 이상의 요청이 들어와서 그 요청을 일시적으로 처리하지 못하는 경우가 발생하는 것은 수용이 가능하나 그로 인해 서비스가 궁극적으로 중단되어서는 안 될 것이다.

본 논문에서 제안된 인터미디어리 서버도 역시 서버 측 프로그램이므로 해당 프로그램이 중단되지 않고 클라이언트의 동시 요청을 안정되게 서비스 하는지를 알아보기 위해서 쓰레드의 개수를 변



(그림 9) 인터미디어리 서버와 모바일 클라이언트의 쓰레드 연결 쌍개수 증가에 따른 메시지 처리 응답 시간

경해 가면서 평균 응답 시간의 변화와 응답 안정성 등을 조사해보면 해당 인터미디어리와 모바일 클라이언트와의 메시지 처리 성능을 테스트해 보았다.

이러한 기본 시뮬레이션 테스트 결과는 시스템의 메시지 처리 쓰레드의 생성 개수가 많아질수록 응답 시간이 증가하고 메시지의 길이가 커질수록 응답 시간이 증가할 것이다. 즉, 여기서 평균 응답 시간이 급격히 증가하는 때의 인터미디어리 서버의 쓰레드 수를 통하여 성능을 알 수가 있다.

6. 결론

본 논문에서는 무선 환경에서의 웹 서비스를 위한 메시지를 설계하는 모델을 제시하였다. 우선 모바일 클라이언트가 인터넷의 웹 서비스 서버들로부터 모바일 웹 서비스를 제공 받을 때, 그 성능 향상을 위한 서비스 중재 역할을 담당하는 인터미디어리 서버를 제안하였다. 또한 이러한 모바일 분산 인터미디어리 서버 기반 웹 서비스에서 모바일 웹 서비스의 핸드오프 문제와 프로토콜 변환 문제를 해결하기 위해 핸드오프 처리와 함께 프로토콜을 중재해주는 컨버전스 서버 기반의 XML/SOAP 메시지 처리 모델을 설계하여 제안하였다.

제안된 분산 인터미디어리와 컨버전스 기반 모바일 웹 서비스 메시지 처리 모델은 모바일 클라이언트가 이동하는 셀 영역 계층, 이와 직접적으로 연결된 인터미디어리 서버 계층, 프로토콜 중재 등

을 담당하는 상위 컨버전스 서버 계층, 서비스의 콘텐츠를 제공하는 웹 서비스 계층이 상호 유기적으로 연동되도록 설계 되었다. 이를 통해 모바일 웹 서비스 사용자는 이동 중에도 지속적으로 유지되는 QoS가 보장되는 서비스를 제공 받을 수 있을 것이다.

참 고 문 헌

- [1] 이경하, 이규철, "웹 서비스의 향후 발전 방향", 정보처리학회지, 제 9 권, 제 4 호, pp. 24-30, 2002년 2월.
- [2] 이양선, 임철수, 이용학, "무선 인터넷 응용 서비스를 위한 WAP 게이트 웨이의 설계 및 구현", 한국콘텐츠학회논문지, Vol. 1, No. 1, pp.109-117, 2001
- [3] 이원석, 전중홍, 이강찬, 이규철, "인터미디어 리 기반의 모바일 웹서비스 프레임워크", 한국해양정보통신학회논문지, 제 7 권, 제 8 호, pp.1661-1673, 2004
- [4] 김두용, 박상현, "낮은 지연을 갖는 사전등록 핸드오프의 성능분석", 한국산학기술학회논문지, Vol. 4, No. 4, pp. 320-333, 2003
- [5] Martin Gudgin, et al., "SOAP Version 1.2 Part 1: Messaging Framework", <http://www.w3c.org/TR/2002/WD-soap12-part1>, Working Draft, W3C, June 2002.
- [6] Roberto Chinnici, et al., "Web Services Description Language(WSDL) Version 1.2", <http://www.w3c.org/TR/wsdl12>, Candidate Recommendation, W3C, December 2002.
- [7] Ariba Inc., Microsoft Corp., "UDDI Technical White Paper", http://www.uddi.org/pubs/Iru_UDDI_Technical_White_Paper.pdf, Uddi.org, September, 2000.
- [8] WAP Forum Ltd., "Wireless Application Protocol Architecture Specification", <http://www.wapforum.org>, 30-Apr-19
- [9] Frank P. Coyle, "Mobile Computing, Web Services and the Semantic Web: Opportunities for M-Commerce", Computer Science Department, School of Engineering, Southern Methodist University, Dallas Texas 75275 USA.
- [10] W3C, "Web Services Architecture, W3C Working Draft", May 14, 2003
- [11] WAP Forum Ltd., "Wireless Application Protocol Binary XML Content Format Specification Version 1.2", <http://www.wapforum.org>, 4-Nov-1999

◎ 저자 소개 ◎



김인희

2002년 서울여자대학교 컴퓨터학과 졸업(학사)
2002년 서울여자대학교 대학원 컴퓨터학과 입학(석사)
2005년 서울여자대학교 대학원 컴퓨터학과 재학중(석사)
관심분야 : 분산처리, 웹 서비스, 디지털 방송



황준

1985년 중앙대학교 컴퓨터공학과 졸업(학사)
1987년 중앙대학교 대학원 컴퓨터공학과 졸업(석사)
1991년 중앙대학교 대학원 컴퓨터공학과 졸업(박사)
1992년~현재 서울여자대학 정보통신공학부 교수
관심분야 : 유비쿼터스, 분산 처리



이웅재

1981년 연세대학교 전기공학과 졸업(학사)
1985년 일리노이주립대학교(UIC) 대학원 전산학과 졸업(석사)
1991년 일리노이공대(IIT) 대학원 전산학과 졸업(박사)
1993~현재 서울여자대학교 미디어학부 교수
관심분야 : 인공지능, 자연어처리, 멀티미디어, 게임, etc.
E-mail : wjlee@swu.ac.kr