

## 무선 인터넷 환경에서의 Push 서비스를 위한 Push Content Generator 설계

이정은, 박병하, 신동일, 신동규  
세종대학교 컴퓨터공학과  
e-mail:{jelee, bhpark, dshin, shindk}@gce.sejong.ac.kr

### A Push Content Generator for Push Service in Wireless Internet Environment

Jungeun Lee, Byoungaha Park, Dongil Shin, Dongkyoo Shin  
Dept. of Computer Engineering University

#### 요약

최근 무선 인터넷의 보급과 더불어 다양한 기능이 요구되고 있으며, 특히 WAP(Wireless Application Protocol)상에서 지원되는 Push 서비스의 요구가 늘어나고 있다. 본 논문에서는 무선 프로토콜인 WAP상에서 사용하는 푸시(push) 메시지의 보다 쉬운 구현을 위해 Push 서비스의 구조와 동작 원리를 소개하고 푸시 콘텐츠 생성기(Push Content Generator)의 구조와 설계에 대해 기술한다.

#### 1. 서론

우리는 지금 메인 프레임의 시대를 기점으로 PC의 시대를 지나, 어디에나 컴퓨터가 존재하는 컴퓨팅의 시대, 포스트 PC의 시대로 가고 있다. 그리고 그 중심에는 '모바일'이라는 패러다임이 존재한다. 선으로부터 자유로운 통신의 시대가 도래한 것이다. 그중 모바일 인터넷 서비스란 기본적으로 휴대형 단말기(정보기기, 통신기기 포함)를 통해 무선으로 데이터통신이나 인터넷서비스를 제공하는 것으로 다양한 형태의 무선접속 방식(다양한 단말기와 상이한 주파수 환경)과 서비스 형태를 보이고 있다. 현재 인터넷 서비스는 기본적으로 클라이언트가 브라우저에서 직접 URL을 입력하거나 하이퍼링크를 선택해 정보를 찾는 풀(pull)방식의 서비스와 클라이언트에 대한 정보를 가지고 클라이언트의 어떠한 조작 없이 데이터를 전송하는 방법 푸시(push) 방법 두 가지가 있다. 즉 풀(pull)방식의 트랜잭션은 항상 사용자의 측면에서 시작되고, 푸시(push) 트랜잭션은 서버측에서 시작되는 것이다. 특히 푸시(push) 서비스 기능은 무선 인터넷에서 그 중요성이 더욱 부각된다. 그 이유는 고객을 중심으로 고객이 원하는 시점에, 고객이 원하는 장소에서, 고객이 원하는 정보를 푸시(push)하기 때문이다. 이러한 요구가 증

가되는 시점에서 본 논문에서 설계하는 푸시 콘텐츠 생성기(Push Content Generator)는 무선 인터넷에 기반한 무선인터넷 환경의 Push 프레임웍에서 효율적으로 Push 메시지를 생성하고 관리하는데 목적을 두고 있다.

논문의 구성은 다음과 같다. 2장 관련연구에서는 모바일을 통한 웹에 대한 접근을 가능하게 하는 무선응용 프로토콜인 WAP(Wireless Application Protocol)상에서의 푸시(push)에 대한 개념과 구조에 대해 기술하고, 3장에서는 사용자에게 메시지를 푸시(push)하기 위해 메시지를 변환해주는 푸시 콘텐츠 생성기(Push Content Generator)의 구조적인 개념과 전체 시스템에 대한 설계를 제안하며, 4장에서 결론 및 향후과제를 기술한다.

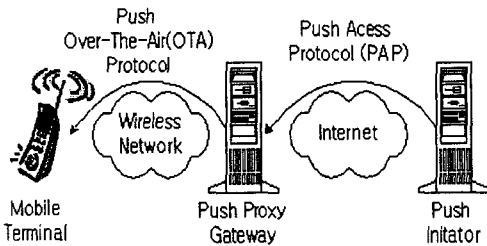
#### 2. 관련 연구

##### 2.1 WAP Push 서비스의 동작 구조

WAP Push 서비스의 기본 동작구조는 아래 [그림 1]과 같다. WAP Spec 1.2에 정의된 푸시 오퍼레이션은 PI(Push Initiator)가 Push OTA(Over-The-Air protocol)이나 PAP(Push Access Protocol)을 이용하여 푸시 콘텐츠를 클라이언트에게 전송하면서

시작된다. 그러나 PI는 인터넷상에 존재하고 WAP 클라이언트는 WAP 도메인(domain)안에 존재하기 때문에 PI는 WAP 클라이언트에게 직접 접근할 수 없으므로 게이트웨이(Gateway)를 필요로 한다. PI는 인터넷상에서 PPG(Push Proxy Gateway)에 접촉을 하게되고, WAP 프로토콜을 사용하여 목적지 이용자를 위해 콘텐츠를 전달한다. 그러한 후 브라우저에 내장된 푸시 에이전트는 푸시 메시지를 전달받아 해당 오퍼레이션을 처리한다.

또한 PPG는 푸시 오퍼레이션의 최종적인 결과를 PI에게 통지하며, 단말기의 성능 탐색 서비스를 제공하여 PI가 푸시 메시지를 받는 특정 클라이언트를 위해 디바이스에서 보여지기 적합한 콘텐츠를 구성하여 전송할 수 있게 한다.



[그림 1] WAP기반 상에서의 Push 구조

## 2.2 WAP Push 서비스의 구성 요소

WAP Push 서비스는 다음 4가지로 구성되며, 각각의 역할과 기능에 대해 기술한다.

### • PPG(Push Proxy Gateway)

PPG의 역할은 무선 네트워크와 인터넷 사이에서 푸시 메시지에 대한 인증, 보안, 클라이언트 제어에 관한 모든 것을 관장하는 것이다. 그러므로 PPG는 WAP 네트워크의 접근에 대한 통제, 푸시 콘텐츠 전송에 관한 통제를 어떻게 할 것인가에 대한 정책을 결정할 수 있는 곳에서 운영해야 한다. WAP포럼은 PPG가 WAP 게이트웨이에 포함되는 것을 권장하고 있다. 이것은 리소스의 공유와 무선에서의 세션 공유라는 이점을 주기도 한다.

### • PI(Push Initiator)

PI는 PAP 프로토콜을 통해 인터넷에서 PPG에 푸시 콘텐츠를 전송한다. PPG가 PI로부터 푸시 콘텐

츠를 수신하면, 응답을 보내준다. 이 응답에는 푸시 메시지가 성공적으로 전달되었음을 알리거나, 푸시 메시지 파싱 결과 발생한 에러에 대한 결과를 요청했을 경우와 PPG가 푸시 메시지를 단말기에 푸시 오퍼레이션을 수행하고 난 후에 그 결과를 PI로 보낼 수도 있다.

### • PAP(Push Access Protocol)

PAP는 PI와 PPG간을 서로 연결해주는 프로토콜이다. PPG와 PI는 TCP/IP 기반의 인터넷으로 연결돼 있으므로 기존 인터넷 프로토콜에 대한 터널링을 명시하고 있으며, SMTP 터널링을 포함할 것도 고려하고 있다. PAP 메시지는 HTTP1.1 메시지 위에 덧씌워져서 전달된다. PAP는 푸시 메시지 제출, 결과 통지, 푸시 메시지 삭제, 상태조회와 Client Capability 조회 등의 임무를 수행한다.

### • Push OTA(Over-The-Air)

Push OTA 프로토콜은 PPG가 PI로부터 받은 메시지를 무선 네트워크에서 단말기에 전송하기 위한 프로토콜이다. Push OTA는 단말기에 콘텐츠를 전달하기 위해 WSP 세션을 사용한다. 단말기와의 연결을 맺기 위해 PPG는 클라이언트의 SIA(Session Initiation Application)에게 연결 요청을 의뢰하고, OTA서버가 SIA의 연결 요청을 기다렸다가 WSP 세션을 맺어 연결 지향의 푸시 메시지를 전송하게 된다. 단방향 네트워크의 경우에는 비연결지향(connectionless) 푸시를 할 수도 있다.

다음에 설명하는 SI(Service Indication)과 SL(Service Loading)은 WAP에서 제공하는 푸시 서비스의 일종으로 그 기능은 다음과 같다.

### • SI(Service Indication)과 SL(Service Loading)

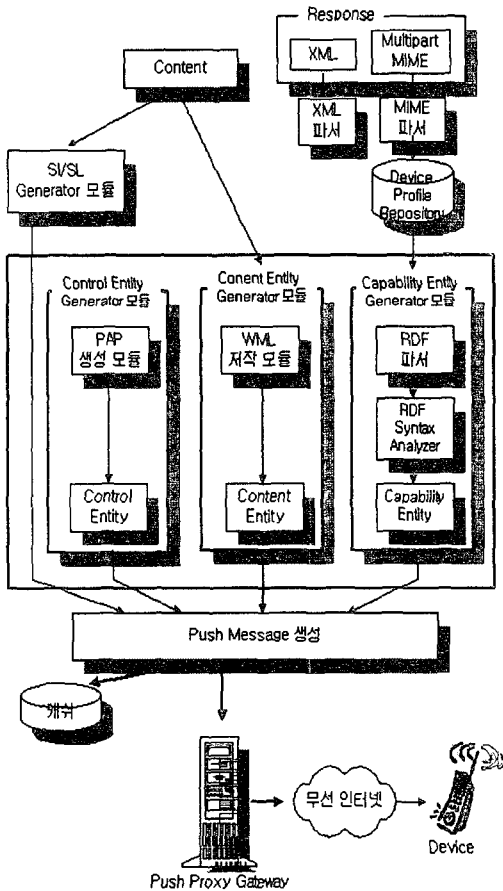
SI이나 SL은 정의된 XML기반의 메시지로 새로운 전자 메일이 도착했음을 알리거나 간단한 메시지와 함께 실제 데이터를 보관하고 있는 URI(Universal Resource Identifier)만을 포함하고 있다.

## 3. Push Content Generator의 설계

### 3.1 Push Content Generator의 동작 구조

본 논문에서 설계한 푸시 콘텐츠 생성기는 아래 [그림 2]에서 보는 바와 같이 크게 Control Entity Generator 모듈과 Content Entity Generator 모듈 그리고 Capability Entity Generator 모듈로 구성된다.

컨텐츠가 PCG에 입력되면 PCG는 사용자에게 컨텐츠를 SI(Service Indication)나 SL(Service Loading) 혹은 푸시 메시지로 작성할 것인지에 대해 선택하도록 하고, 만일 사용자가 SI나 SL을 선택한다면 PCG는 헤더값을 입력받아 각각의 컨텐츠와 헤



[그림 2] Push Message Generator 구조

더 정보를 구성하여 PPG에 전달하고 캐쉬에 저장한다. 그러나 푸시 메시지의 경우에는 그 과정이 복잡하다.

우선 컨텐츠가 생성기의 Main 모듈 안으로 들어오면 컨텐츠는 Content Entity Generator 모듈 안으

로 들어간다. 그리고 WML 저작모듈을 이용하여 컨텐츠를 WML문서로 바꾼다.

Control Entity 모듈은 PAP 생성 모듈을 이용하여 Control Entity를 작성한다.

또한 사용자가 미리 입력하거나 혹은 PAP에게서 받은 Client Capability Profile을 RDF 파서에 넣어 인식하고 RDF Syntax Analyzer를 사용하여 RDF 문서를 점검하거나 변경하여 Capability Entity를 작성한다. 이때 만일 PCG가 PPG에게 Client Capability Profile을 요구하여 응답을 받은 경우에 응답 message는 MIME Type으로 이루어져 있기 때문에 MIME 파서를 이용하여 메시지를 파싱하고 Device Profile Repository에 저장한다. 이러한 일련의 과정이 끝나면 Control Entity와 Content Entity 그리고 Capability Entity를 합쳐 Push Message를 생성하게 된다. 생성된 푸시 메시지는 PPG에 전달되고 캐쉬에 저장된다.

PCG가 보낸 컨텐츠에 대한 PPG의 응답은 XML과 Multipart MIME 형태를 갖는데 XML로 오는 응답은 PCG가 보낸 푸시 메시지에 대한 상태를 전하는 것으로 메시지가 현재 어느 선까지 수행됐는지 혹은 수행이 중단되었는지 등의 상태를 전달하고, Multipart MIME으로 오는 응답은 Client Capability Profile이므로 MIME 파서로 파싱을 한 후에 Device Profile Repository에 저장한다. 그리고 Device Profile Repository에 저장된 Client Capability Profile은 추후 변경 혹은 삭제 가능하다.

### 3.2 푸시 콘텐츠 생성기의 주요 모듈 설명

메시지 생성기는 다음 3가지 모듈로 구성되며, 모듈별 세부 설명은 다음과 같다.

#### ▪ Control Entity Generator 모듈

이 부분에서는 수신자 정보, 타임아웃 콜백 요청 등의 메시지 제어 관련 정보를 포함하고 있어, 메시지의 형식을 규정하고 메시지가 전달될 클라이언트의 주소를 사전에 입력 받게 된다. 그 주소를 가지고 컨텐츠가 도달해야 할 클라이언트 디바이스의 어플리케이션의 주소를 정하고 컨텐츠에게 고유의 아이디를 발급하고, PI의 주소를 첨가한다. 그리고 최종적으로 PAP DTD에 기반을 둔 Control Entity를 생성하게 된다.

• **Capability Entity Generator** 모듈

이 모듈은 클라이언트의 정보를 담을 수 있는데, 옵션이므로 생략할 수도 있다. 사용자의 모바일 폰에 대한 성능이나 종류를 저장한 Client device capability profile을 사전에 Device Profile Repository에 저장한 후에 Capability Entity 모듈을 사용해야 한다. Client Capability Profile은 RDF 형식으로 생성되어 있어 RDF 파서를 이용하여 파싱한다. 이 RDF 파서를 사용하여 device의 성능을 인식하고 그 Syntax를 분석한 후 모바일 폰에 최적화된 푸시 메시지를 작성하도록 한다.

• **Content Entity Generator** 모듈

이 모듈에서는 클라이언트 디바이스에 실질적으로 디스플레이되는 콘텐츠를 무선 인터넷 상에서 전송 가능한 형태로 바꿔주는 기능을 한다. 사용자에게 전달할 푸시 정보(통치, 뉴스, 경고등)와 같은 콘텐츠를 WML 문서로 변환해 주는 모듈로 사용자 인터페이스 상에서 입력받은 text나 wbmp 형태의 이미지를 전송 가능한 WML 문서로 변환하여 Multipart 메시지의 Content Entity를 구성하게 된다.

4. 결론 및 향후 과제

본 논문에서 설계한 푸시 콘텐츠 생성기는 Window NT/2000환경에서 구현할 것이며, 플랫폼의 이식성과 보안등을 위하여 Java(Java2 SDK)를 사용할 것이다. 그리고 사용될 WML 파서는 IBM의 XML4J이고 RDF 파서로는 SiRPAC을 사용할 것이며, MIME 파서는 자체 구현 할 것이다.

Push Content Generator는 앞으로 증가할 사용자나 콘텐츠 제공자의 Push 서비스에 대한 요구를 좀더 쉽게 구현할 수 있을 것이다. Pull 서비스에서와 같은 검색 과정을 줄일 수 있고, Device Profile을 이용해 사용자 개개인의 Device 성능에 맞고 사용자가 원하는 정보를 사용자가 원하는 시간에 전달하는 서비스를 구현할 수도 있다. 이러한 맞춤형 서비스는 앞으로 더욱 요구되어지며 보다 향상된 무선 인터넷 서비스를 위한 기술로 자리 매김 할 것이다.

참고 문헌

[1]Wireless Application Protocol Push Architecture

Overview, WAP Forum, November 10, 1999, <http://www.wapforum.org/>

[2]Wireless Application Protocol Push Message Specification, WAP Forum, November 10, 1999, <http://www.wapforum.org/>

[3]Wireless Application Protocol Push Proxy Gateway Service Specification, WAP Forum, November 10, 1999, <http://www.wapforum.org/>

[4]Wireless Application Protocol Service Indication Specification, WAP Forum, November 10, 1999, <http://www.wapforum.org/>

[5]Wireless Application Protocol Service Loading Specification, WAP Form, November 10, 1999, <http://www.wapforum.org/>

[6]Wireless Application Protocol Push Access Protocol Specification, WAP Forum, November 10, 1999, <http://www.wapforum.org/>

[7]Wireless Application Protocol Push OTA Protocol Specification, WAP Forum, November 10, 1999, <http://www.wapforum.org/>

[8]Wireless Application Group User Agent Profile Specification, WAP Forum, November 10, 1999, <http://www.wapforum.org/>

[9]Nokia WAP Toolkit 2.0 Developer's Guide <http://www.nokia.com/>