

프록시 기반의 저 통신 비용 모바일 신디케이션

정회원 황기태*, 김남윤**

Proxy-based Mobile Syndication with Low Communication Cost

Kitae Hwang*, Namyun Kim** *Regular Members*

요약

현재 웹 신디케이션 기술은 주로 데스크탑 환경에서 구현 활용되고 있기 때문에 이동성에 대한 고려가 되어 있지 않다. 본 논문에서는 데스크탑 기반의 웹 신디케이션 기술을 모바일 인터넷으로 확장하는데 초점을 맞추었다. 모바일 단말기에서 데스크탑 기반의 웹 신디케이션을 단순히 적용하게 되면 정기 구독으로 인해 높은 무선 통신 비용을 초래할 수 있다. 본 논문은 낮은 통신 비용으로 모바일 신디케이션을 구현할 수 있는 PMS(Proxy-based Mobile Syndication) 모델을 제안한다. PMS 모델은 모바일 단말기와 웹 사이트 사이에 프록시 서버인 MWProxy를 정의한다. 프록시 서버는 웹 사이트에서 수신한 채널에서 변경된 아이템을 선택하여 간소화한 채널로 재구성함으로써 통신 비용을 줄일 수 있는 장점이 있다. 본 논문에서는 PMS 모델을 실제 휴대폰 상에서 설계 구현하고 에뮬레이터를 이용하여 성능을 평가하였다.

Key Words : Web Syndication, Proxy-based Mobile Syndication, Low Communication Cost

ABSTRACT

Currently, web syndication is usually utilized in desktop environments, but not considering mobile environments. In this paper, we focus on the extension of the desktop based web syndication to the mobile internet. Simply adapting the desktop based web syndication to mobile devices can cause high wireless internet communication cost. We propose PMS(Proxy-based Mobile Syndication) model which achieves low cost mobile syndication. In the PMS model, MWProxy is defined as a proxy server between mobile devices and web sites. It can reduce the communication cost by peeking up modified items from the web sites and reducing the channel size. In this paper, we designed and implemented the PMS system in a mobile phone and evaluated the performance of the PMS system by conducting real experiments using mobile emulators.

I. 서론

웹의 등장과 함께 인터넷은 정보의 바다가 되었으며 웹 서버를 기반으로 정보를 저장 관리하고 인터넷을 이용하여 누구든지 정보를 얻을 수 있게 되었다. 그러나 웹은 정보 소비자의 입장에서 다분히 수동적인 면이 있다. 즉, 인터넷에 새로운 정보가 발생한다고 해도 검색하기 전까지는 알 수 없으며,

새로운 정보를 찾기 위해 사용자가 직접 여러 웹 사이트를 방문하여야 하는 부담을 지니고 있다.

이러한 정보 검색의 전통적인 방식의 문제점을 개선하기 위해 웹 신디케이션(Web Syndication) 방법이 제안되었다^{[1]~[3]}. 웹 신디케이션은 정보 소비자의 컴퓨터 상에 정기 구독기 소프트웨어를 설치하고 이 정기 구독기가 정보 소비자를 대신하여 주기적으로 정보 검색을 시행하여 새로운 정보를 정

※ 본 연구는 2009학년도 한성대학교 교내 연구비 지원 과제임.

* 한성대학교 컴퓨터공학과(calafk@hansung.ac.kr), ** 한성대학교 정보시스템공학과(nykim@hansung.ac.kr)

논문번호 : KICS2009-02-048, 접수일자 : 2009년 2월 5일, 최종논문접수일자 : 2009년 8월 10일

보 소비자에게 알려주는 방법이다. 이것은 마치 아침마다 정기 구독한 일간 신문이 배달되는 것과 같은 이치이다. 정기 구독기와 웹 사이트 사이에는 RSS(Really Simple Syndication) 프로토콜 등과 같은 신디케이션을 위한 프로토콜이 수행되며⁴⁾, 웹 사이트는 정기 구독기에게 자신이 가진 정보 콘텐츠의 프로파일을 묘사한 XML 형식의 채널(channel)이라고 불리는 메타 정보를 전송한다. 채널은 RSS나 ATOM⁵⁾ 등 다양한 형식으로 표현되고 있지만 본 논문에서는 RSS에 한정한다. 정보 소비자는 정기 구독기 소프트웨어에게 원하는 웹 사이트를 등록하면 이 사이트들을 직접 방문하지 않고도 새로운 정보에 대해 알 수 있다.

최근 들어 정기 구독기 소프트웨어들이 속속 개발되고 있다. 그러나 현재 개발 중인 웹 신디케이션 기술은 대부분 데스크탑 PC 환경에 사용될 수 있도록 개발되었기 때문에 이동성이 보장 되지 않는다. 본 논문은 모바일 인터넷 사용자들이 늘어가고 있으며 언제 어디서나 정보를 획득하고자 하는 정보 시대에, 모바일 인터넷에 신디케이션 기술을 구현하는데 초점을 맞추어 모바일 신디케이션 시스템을 제안한다.

데스크탑에 적용된 신디케이션 기술을 모바일 단말기에 바로 적용하는 것은 간단하지만은 않다. 모바일 단말기에 설치된 정기 구독기가 주기적으로 사이트를 방문하여 RSS 채널을 가져오게 되므로 여러 가지 부가적인 통신 비용이 발생한다. RSS 사이트로부터 채널 데이터를 직접 가져오게 되면 지난번에 이미 전송받은 동일한 아이템이 중복 전송 받게 되고, RSS 사이트로부터 관심 있는 아이템만을 선별적으로 전송할 수 없고 모든 아이템을 받을 수 밖에 없다. 이것은 통신 비용 뿐만 아니라, 단말기의 메모리를 효율적으로 사용하지 못하는 어려움을 동반하게 된다. 또한 모바일 단말기 LCD의 크기나 상황 때문에 데스크탑에서 적용하였던 사용자 인터페이스를 그대로 사용할 수는 없다.

본 논문에서는 정보 소비자의 만족도를 높이면서 무선 인터넷의 통신량을 줄이고 적은 통신 비용으로 모바일 단말기에서 신디케이션 기술을 적용하기 위한 방법으로 PMS(Proxy-based Mobile Syndication)를 제안한다. PMS는 모바일 웹 신디케이션 프록시 서버, MWProxy(Mobile Web Syndication Proxy Server)를 도입하였다. 프록시 서버는 RSS 채널을 가지고 오고 채널에서 중복되거나 불필요한 아이템을 줄인다. 또한 모바일 단말기의 요구 사항에 따라

아이템의 개수를 제한하는 지능적인 알고리즘을 구현하여 모바일 단말기와 프록시 서버 사이의 무선 통신 비용을 줄인다. 본 논문에서는 MWProxy와 정기 구독기를 설계 구현하여 실험을 통해 성능 평가를 수행하였다.

본 논문의 구성은 다음과 같다. 2절에서는 신디케이션 시스템의 연구 배경을, 3절에서는 모바일 신디케이션 시스템을 모델링하고 4절에서 설계한 내용과 성능 이슈에 대해 기술한다. 5절에서는 구현 사례를 6절에서는 성능 평가에 대해 기술하고 7절에서 결론을 맺는다.

II. 연구 배경

2.1 RSS

웹 신디케이션이란 웹 사이트가 자신이 제공하는 정보에 대해 요약된 메타 데이터인 채널(channel)을 생성하면 이를 수집하는 응용 프로그램을 통해 빠른 시간 내에 사용자들에게 최신 정보를 알려주는 기술이다. 현재 웹 신디케이션의 활용 분야는 블로그, 뉴스 사이트 뿐만 아니라, 날씨, 도서관, 교통정보, 홈쇼핑, 금융정보, 영화, 음악 등으로 점점 넓어가고 있다⁶⁾.

RSS는 현재 웹 신디케이션에서 가장 많이 사용하고 있는 채널 형식이다⁴⁾. 채널은 피드(feed)라고도 하며 XML 형식으로 구성된다. RSS 개념은 넷스케이프(Netscape)사에서 운영하던 인터넷 최대의 포털 사이트인 넷센터(NetCenter)에서 유명 신문사의 기사를 제공하기 위하여 등장하였으며, 그 후 메타 데이터의 기술 방법을 XML과 결합시켜 표준 데이터 정의 언어로서 RSS(RDF Site Summary) 버전 0.9를 제안하게 되었다. 현재는 RSS-DEV Working Group의 RSS(Rich Site Summary) 버전 0.92, 1.0과 Userland의 RSS(Really Simple Syndication) 버전 2.0 등 두 종류의 RSS가 이용되고 있다. 본 논문에서 제안하는 시스템은 Userland 사의 RSS 2.0 규격을 적용한다.

하나의 RSS 피드는 하나의 채널을 표현하며 RSS 피드는 그림 1과 같이 여러 개의 아이템(item)으로 구성된다. 뉴스의 예를 들자면, 하나의 채널이 경제면 전체의 기사이고, 각 아이템은 경제면에 게재된 각 단위 기사에 해당한다. 아이템은 제목, 간단한 설명, 아이템 생성 시간, 아이템에 해당하는 웹 페이지의 URL 링크 등을 포함한다. 그림 2는 실제 조선일보사의 RSS 채널 예를 보이고 있다.

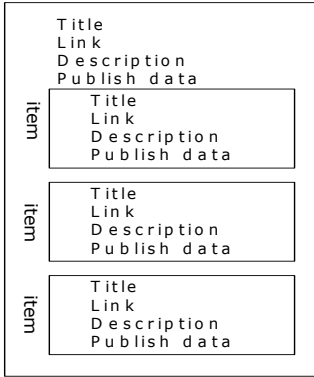


그림 1. RSS 피드(혹은 채널) 구성

```
<?xml version="1.0" encoding="euc-kr" ?>
<rss version="2.0" xmlns:dc="http://purl.org/dc/elements/1.1/">
- <channel>
  <title>english.chosun.com : Total</title>
  <link>http://english.chosun.com</link>
  <description>english.chosun.com RSS Service | Total</description>
  <lastBuildDate>2008.05.28 10:25:45.492</lastBuildDate>
  <copyright>Copyright (c) 2004 chosun.com All rights reserved</copyright>
  <webMaster>letters@chosun.com</webMaster>
  <language>ko</language>
- <image>
  <title>english.chosun.com</title>
  <url>http://image.chosun.com/img/200305/logo_rss.gif</url>
  <link>http://english.chosun.com</link>
  <description>Chosun.com - English Edition</description>
</image>
- <item>
  - <title>
    <![CDATA[Ties With China Require Delicate Handling ]]>
  </title>
  <link>http://english.chosun.com/w21data/html/news/200805/200805280026.html</link>
  <dc:date>2008-05-28 10:25:45</dc:date>
  <author />
</item>
- <item>
  - <title>
    <![CDATA[Discontent Growing as Gov't Marks 100 Days in Office ]]>
  </title>
  <link>http://english.chosun.com/w21data/html/news/200805/200805280025.html</link>
  <dc:date>2008-05-28 09:57:29</dc:date>
  <author>(englishnews@chosun.com)</author>
</item>
```

그림 2. 조선일보 사이트의 RSS 채널 예

2.2 관련 연구

웹 신디케이션 기술이 제안되고 데스크탑에서 사용가능한 RSS Reader 등의 정기 구독기가 여러 개 배포되고 있지만 아직 활성화가 이루어진 단계는 아니다. 모바일 단말기 등의 무선 인터넷 환경에서 활용 가능한 RSS 피드를 제공하는 사이트는 더 더욱 많지 않다. 그러나 신디케이션 기술이 점점 확산 될 것으로 보이며, 가까운 미래에 안정적인 유비쿼터스 네트워크가 확산됨에 따라 언제 어디서나 이동 중에도 휴대폰 등의 모바일 장치를 이용하여 최근 정보를 검색 시간의 낭비 없이 빠르게 얻을 수 있는 모바일 신디케이션 기술은 더욱 필요할 것으로 예측된다.

현재 모바일 장치에 웹 신디케이션을 적용하는

연구는 많지 않다. 국내에서는 열악한 자원을 가진 모바일 단말기에서도 사용이 가능하도록 모바일 RSS 구독기를 경량화한 연구가 수행되었으며^[7], 2007년 독일의 Mannheim 대학은 무선 인터넷 연결 비용을 줄이기 위해 모바일 단말기에 설치된 블루투스(bluetooth)를 이용하여 모바일 단말기들을 P2P 네트워크로 연결하고, 피어끼리 서로 데이터를 교환하게 함으로써 인터넷 비용을 줄이도록 하였지만, 블루투스의 연결 거리가 짧아 활용성에 한계가 있다^[8]. 2005년 벨기에 Ghent 대학은 모바일 단말기로 전송할 채널 크기를 줄이기 위해 웹 서버를 구축하고 압축된 채널 정보를 모바일 단말기로 보내는 방법을 제안하였다^[9]. 이 방법은 채널 크기는 줄이지만 압축/해제에 따른 시간 및 알고리즘의 복잡도가 높다. 2005년 숭실대학교는 사용자가 정의한 키워드와 관련 정보를 검색하여 새로 채널을 생성하고 사용자에게 보내는 시스템을 제안한 바 있다^[10]. 데스크탑 PC를 사용하여 신디케이션을 수행하는 경우, RSS Reader를 수행하지 않고 있거나 PC를 사용하지 않고 있는 경우에 발생하는 뉴스 정보를 잃어버리지 않기 위해 프록시 등의 분산 시스템을 구성하여 이를 항상 저장하는 시스템이 연구되기도 하였다^[11].

III. 모바일 신디케이션 시스템

3.1 시스템 개요

본 논문의 목적은 모바일 단말기에 적합한 신디케이션 시스템을 제안하고 구현하여 성능을 평가하여 제안된 시스템의 효율성을 보이는데 있다.

전체 시스템은 그림 3과 같이 묘사된다. 모바일 단말기와 본 논문에서 정의된 MWProxy 서버로 구성된다. RSS 정보를 제공하는 뉴스 사이트, 인터넷 서점, 연구 사이트, 블로그 등의 웹 사이트를 대상으로 최신 RSS 정보를 주기적으로 받을 수 있는 시스템이다. 모바일 단말기와 MWProxy 사이의 통신은 RSS 프로토콜을 이용한다. 모바일 단말기에는 사용자로부터 정기 구독 신청을 받아 웹 사이트의 RSS 정보를 주기적으로 얻어 주는 역할을 하는 MWReader 구독기를 개발하여 설치한다.

MWReader는 사용자가 등록한 정기 구독 정보를 MWProxy에게 다시 등록한다. MWProxy와 단말기는 독립적으로 작동된다. MWProxy는 등록된 사이트를 주기적으로 방문하면서 RSS 채널을 가지고 있으며 항상 최신의 정보를 유지한다. MWReader는



그림 3. 모바일 신디케이션 시스템 모델

등록한 채널의 구독 주기에 따라 MWProxy에 RSS 채널을 요청하고 MWProxy는 미리 준비된 RSS 채널을 단말기로 전송한다. MWProxy는 등록되 채널 주기에 따라 RSS 사이트에서 가져온 채널에서 아이템을 선별하여 새로운 simplified RSS 채널을 따로 만들어 전송한다. 선별하는 방법은 이 채널 내에 이전에 전송하였던 것과 중복된 아이템을 제거하는 등 여러 가지 방법이 있을 수 있다. 이렇게 함으로써 단말기와 MWProxy 서버 사이의 통신량을 줄일 수 있다.

3.2 모바일 신디케이션 모델

3.2.1 가입(Apply)

가입은 모바일 단말기의 사용자를 MWProxy에 등록하는 과정이다. 가입의 과정에서 사용자의 이름과 암호가 등록되며 MWProxy에 의해 MID(모바일 ID)가 부여된다. 사용자의 이름과 암호는 모바일 단말기에서 생성되며, 모바일 단말기와 MWProxy의 통신에서 MID는 단말기를 인식하는데 이용된다.

3.2.2 채널 등록(subscribe)

채널 등록이란 사용자가 언고자 하는 웹 사이트의 채널 주소와 주기를 MWReader에게 등록하는 것이다. 사용자는 MWReader가 출력한 UI 화면에 표 1의 정보들을 입력한다.

MWReader는 이 정보를 내부적으로 저장 관리하며 입력받는 즉시 그림 4의 과정을 거쳐 MWProxy에게 전달한다. MWProxy는 지체 없이 바로 RSSURL에 주어진 웹 사이트로부터 RSS 채널을 요청하여 전송받고 RSS 채널을 간소화한 뒤 이를 내부에 저장해 두고 한편 모바일 단말기로 전송함으로써 채널 등록의 과정을 마친다. 그림 4에서 추천 과정은 채널 아이템을 간소화하는 과정을 말한다.

표 1. 채널 등록 데이터

등록 필드	설명
MID (모바일 ID)	MWProxy로부터 초기에 부여받은 ID
CName (채널 등록 이름)	RSS 채널 이름
RSSURL (사이트와 채널 주소)	RSS 사이트와 채널 경로명에 대한 주소 URL
Interval (채널갱신 시간간격)	RSS 채널을 가져오는 주기로서 시간 간격
MaxItem (채널아이템 개수)	MWReader가 MWProxy로부터 RSS 채널을 전달 받을 때 채널에 포함될 최대 아이템 개수. 0 이면 RSSURL로부터 받은 아이템을 삭제없이 그대로 전송.
Keyword (키워드)	관심 있는 아이템을 선택하기 위한 키워드 문자열. 하나의 이상의 키워드가 문자열로 구성.

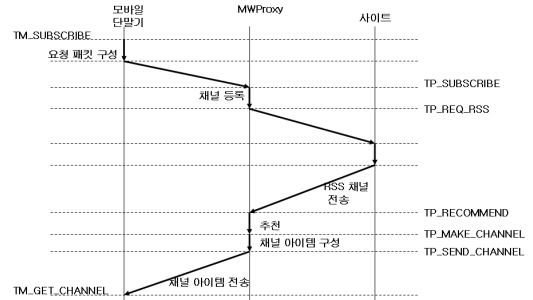


그림 4. 채널 등록 시간 모델

3.2.3 대행(agent)

MWProxy는 사이트 등록의 과정이 끝난 뒤부터 그림 5와 같이 바로 주기적으로 RSSURL에 주어진 RSS 사이트로부터 RSS 채널을 가지고 온다. 그리고 추천의 과정을 거쳐 RSS 채널을 간소화한다. 간소화된 RSS 채널은 MWProxy에 저장 관리된다. 이

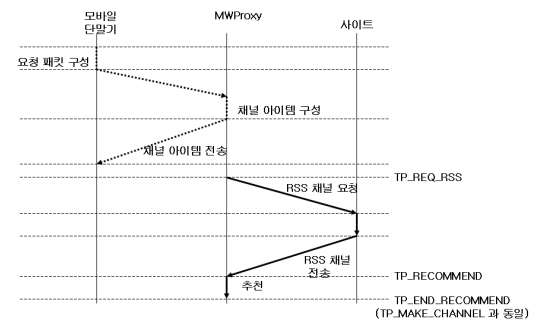


그림 5. 대행 시간 모델

과정은 MWProxy 내부에서 자율적으로 이루어진다.

3.2.4 채널 갱신(refresh)

MWReader는 사용자가 설정한 주기에 따라 정기적으로 MWProxy에 접속하여 채널 갱신을 요청한다. MWProxy는 이 요청에 대해 RSS 사이트를 방문하지 않고, 요청한 CName(채널 등록 이름)에 해당하는 채널을 MWProxy 내부에서 관리하는 간소화된 RSS 채널 리스트에서 찾아 곧바로 단말기로 전송한다. 이 간소화된 RSS 채널은 가장 최근에 MWProxy가 RSS 사이트로부터 받아와서 추천의 과정을 거쳐 간소화된 것이다. 이 과정은 그림 6과 같이 묘사된다.

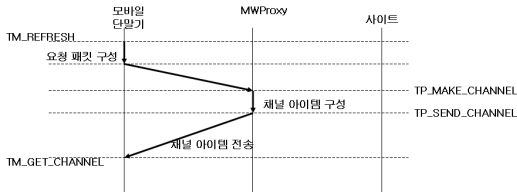


그림 6. 채널 갱신 시간 모델

3.2.5 아이템 콘텐츠 얻기

사용자는 MWReader로부터 단말기가 가지고 있는 채널 리스트를 볼 수 있으며, 원하는 채널을 선택하고 그 채널의 아이템 중 관심 있는 아이템을 선택하면 단말기의 플랫폼이 무선인터넷을 통해 해당 사이트에 접속하고 원하는 웹 리소스 혹은 웹 콘텐츠를 가지고 와서 화면에 출력한다.

IV. 신디케이션 시스템 설계

4.1 설계 원칙 및 고려사항

본 논문에서 제안한 MWReader는 채널 정기 구독기의 필수적인 기능은 MID 얻기, 채널 등록, 채널 변경, 채널 갱신, 아이템 콘텐츠 얻기만을 가지도록 하고, 분/시간/일 단위로 웹 신디케이션이 가능하도록 설계한다. 소프트웨어 모듈은 CPU 시간을 절약하도록 설계한다.

MWProxy의 설계의 중점 사항은 단말기의 채널 요청의 응답 시간을 가능하면 빠르게 하는데 있다. 이를 위해 첫 채널 크기를 간소화하고 채널 갱신 요청이 발생하면 곧바로 준비된 채널을 전송하도록 한다. 또한 다수의 단말기를 구분하고 인식하기 위해 ID를 등록하는 기능을 구현하며, 분/시간/일 단위로 웹 신디케이션이 가능하도록 설계한다. 표준화

된 프로토콜을 사용하기 위해 HTTP 기반의 웹 서비스로 구현하여 모바일 단말기와 통신 체계를 구축하며, 간소화된 채널 리스트 역시 XML 기반으로 형성한다.

4.2 MWReader 설계

MWReader의 구조는 그림 7과 같이 표현된다. MWReader에 등록된 채널은 채널 갱신 주기를 분(minute) 단위, 시간(hour) 단위, 일(day) 단위의 3가지 경우로 지원하므로 등록된 채널을 이 3가지 유형 별로 day list, hour list, minute list 등의 리스트를 이용하며 관리한다. 그리고 리스트마다 하나의 스레드를 두어 채널 주기별로 MWProxy 서버에 접속하여 채널 정보를 가지고 온다. 이 정보는 채널 저장소(subscription & channel storage)에 저장되며 사용자가 채널 정보를 보고자 하면 channelParser를 이용하여 채널 정보와 아이템 정보를 화면에 보여준다. 채널 내의 한 아이템을 직접 읽고자 하면 단말기 상의 브라우저를 이용하여 채널 아이템을 직접 가져오게 한다.

channelParser는 XML 형식으로 전송되는 채널 데이터를 파싱하는 것으로 XML 라이브러리를 사용하지 않는다. XM위피에서 작동되는 XML 라이브러리가 없을 뿐 아니라 새로 만들어 이용한다고 하더라도 XML 파싱으로 인한 CPU의 소모시간이 너무 길기 때문이다. 또한 서버에서 온전한 형식의 XML 채널을 보냈다고 가정하면 굳이 파싱할 이유도 없기 때문이다. 채널 데이터의 파싱은 전송된 채널 스트링에서 필요한 태그를 찾는 방식으로 간단히 작성하였다.

GUI는 위피의 컴포넌트를 이용하여 간단히 구성하도록 설계하였으며 subscription list에는 표 1에 설계된 등록 항목들이 저장된다. day list, hour list, minute list는 subscription list로부터 해당 주기에 따라 분배된 채널 등록 아이템을 가진다.

각 스레드는 가지고 온 채널 피드를 저장소에 저

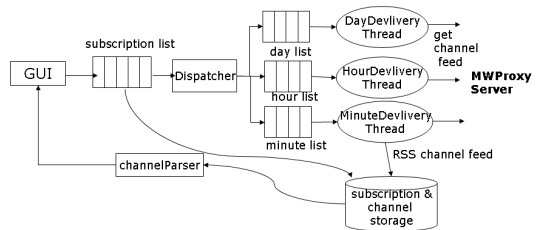


그림 7. MWReader 구조

장하고 GUI에서 사용자의 요청에 따라 파싱된 후 화면에 출력된다.

4.3 MWProxy 설계

MWProxy의 구조는 그림 8과 같이 표현된다.

- MWReader와의 통신

MWReader와 통신하기 위해 3 개의 자바 서브릿(java servlet)을 설계하였다. register servlet은 사용자 ID를 생성하여 사용자를 등록하는 작업을, subscribe servlet은 채널 등록 작업을, refresh servlet은 채널 갱신 요청을 받아 응답하는 작업을 각각 수행한다.

- 채널 피드 페치

모바일 단말기의 MWReader와 마찬가지로 등록된 채널을 갱신 주기에 따라 분, 시간, 일의 3 유형으로 나눈다. 각 스레드는 RSS 웹 사이트로부터 주기적으로 RSS 채널을 가지고 와서 채널 스토리지에 저장한다. 동시에 이 채널 피드를 channelCooker를 이용하여 파싱하여 지난번 전송한 아이템들을 제거한 후 새로 추가된 아이템만으로 구성된 간소화된 채널(simplified RSS channel)을 만들고 간소화된 채널 리스트(simplified channel list)에 저장한다.

- 짧은 채널 갱신 응답

refresh servlet이 MWReader로부터 채널 갱신 요청을 받으면 간소화된 채널 리스트에서 요청 채널을 찾고 단말기로 전송한다.

- 채널 간소화

채널 간소화를 위해 이 논문에서 설계 구현된 방식은 “recent-item-first” 이다. 이 방식은 RSS 채널의 생성 시간을 고려하여 최근에 발생한 아이템에게 높은 우선 순위를 주는 방식이며, 현재 구현된 것은 지난번에 단말기로 전송한 아이템을 모두 제

거하고 최신 아이템만을 선별하여 간소화된 채널 리스트를 만든다.

4.4 성능 이슈와 추천 알고리즘

4.4.1 BMS와 PMS 알고리즘

본 논문에서 제안한 모바일 신디케이션 시스템의 성능을 분석하기 위해 기본적인 모바일 신디케이션 모델인 BMS(Basic Mobile Syndication) 모델을 정의한다. BMS는 PC 상에서 현재 사용 중인 신디케이션을 모바일에 적용한 단순 모바일 신디케이션 시스템이다. 모바일 상에서 구동되는 정기 구독기는 프록시를 거치지 않고 사용자가 설정한 주기에 따라 직접 RSS 사이트에 접속하여 RSS 채널을 가져온다. RSS 채널에 포함된 아이템의 개수를 조절할 수 없기 때문에 관심 없는 아이템을 가져오기 전에 분리해 낼 수 없으며, 자주 갱신되지 않는 채널의 경우 이미 이전에 가져온 아이템이 중복해서 전송받게 되므로 높은 통신 비용을 초래하게 된다. 또한 단말기 상의 메모리를 효율적으로 사용하기 위한 방법을 따로 준비하여야 하는 부담이 있다.

이에 반해 본 논문에서 제안하는 PMS(Proxy-based Mobile Syndication)는 프록시 서버를 두고 추천 알고리즘을 이용하여 통신 비용을 줄이는 신디케이션 시스템이다. 현재 추천 알고리즘은 단순히 이전에 전송하였던 아이템을 제거하는 단순한 방식만 사용한다.

4.4.2 성능 이슈와 통신 비용

모바일 단말기의 응용 시스템을 개발함에 있어 주요한 고려 사항 중의 하나가 통신 비용이다. 모바일 신디케이션의 목적은 신속한 정보를 얻는 것이지만 이에 따른 통신 비용의 지불이라는 대가를 치러야 한다. 동일한 정도의 정보의 신속성이 주어진다면 통신 비용을 낮출 수 있는 방법들이 필요하다.

정보의 신속성은 RSS 사이트에 새로운 정보가 생겼을 때 얼마나 빠른 시간에 얻는가에 달려 있는데, 이것은 사용자 관점에서 설정하는 채널에 대한 주기로서, 사용자가 얼마나 자주 RSS 채널을 확보하고자 하는지에 달려 있다. RSS 사이트에서 RSS 채널을 요청하여 피드를 얻는데 걸리는 시간 역시 신디케이션 시스템에 의존하기 보다는 RSS 사이트의 응답 시간과 통신 매체에 의존하는 요소이기 때문에 정보의 신속성은 신디케이션 시스템 관점에서 관련성이 거의 없는 요소라고 분석된다.

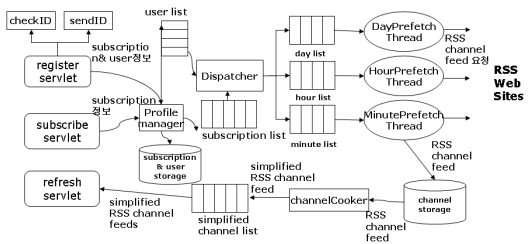


그림 8. MWProxy 구조

그러므로 본 논문에서는 모바일 신디케이션의 통신 비용을 줄이는데 성능의 초점을 맞추었다.

4.4.3 통신 비용 분석

모바일 단말기에서 채널 등록 혹은 채널 갱신을 수행하는 동안 소요되는 통신 시간은 BMS 모델과 PMS 모델의 대해 각각 다음 수식 (1), (2)와 같이 표현된다. 수식 (1), (2)에서 BMSReqSize와 PMSReqSize는 정기 구독기에 의한 채널 등록 혹은 채널 갱신 시의 요청 패킷 크기이고, BMSChannelSize와 PMSChannelSize는 정기 구독기가 받는 응답 패킷의 크기이며, bps(bits per second)는 모바일 단말기가 연결된 무선 통신 대역폭이다. proxyDelay는 MWProxy가 요청 받은 채널 사이트에 접속하여 RSS 채널을 받아오는데 걸리는 시간이다.

$$T_{BMS}(n) = \sum_{i=1}^n (BMSReqSize_i + BMSChannelSize_i) / bps, \tag{1}$$

여기서 n은 통신 횟수

$$T_{PMS}(n) = \sum_{i=1}^n (PMSReqSize + PMSChannelSize_i) / bps + proxyDelay_i, \tag{2}$$

여기서 n은 통신 횟수

BMS 모델에서 채널 등록 및 갱신에 관계없이 통신 시간은 수식 (1)과 같이 표현된다. 그러나 PMS에서는 수식 (2)에서 채널 등록 시 proxyDelay가 0이 아니지만 채널 갱신 시 proxyDelay가 0이 되어 수식 (2)가 수식 (3)과 같이 단순화된다.

$$T_{PMS}(n) = \sum_{i=1}^n (PMSReqSize_i + PMSChannelSize_i) / bps \tag{3}$$

여기서 n은 통신 횟수

두 모델의 통신 시간을 자세히 비교해보자. 정기 구독기를 사용하는 동안 발생하는 통신 시간은 채널 등록과 채널 갱신에 따른 시간의 합이다. 그러나 채널 등록은 하나의 채널에 대해서 일회적으로 발생하지만 채널 갱신은 등록된 채널에 대해서 주기적으로 이루어지므로 채널 갱신에 따른 통신 시간이 보다 중요하다.

채널 등록에 있어서 소요되는 통신 시간을 수식

(1)과 수식 (3)을 자세히 비교하여 분석해보자. RSS 채널을 요청하는 것은 동일하므로 수식 (1)과 수식 (2)에서 BMSReqSize와 PMSReqSize는 거의 같다. PMS 모델에서는 MID 정보와 Interval, Keyword 정보를 포함하기 때문에 PMSReqSize는 BMSReqSize보다 몇 바이트 많을 수 있지만 이들은 바이트 수가 매우 작기 때문에 통신 시간에 영향을 미치지 않고 예측되지 않는다. 그러므로 통신 시간을 결정하는 요소는 다음 두 가지로 압축된다. 응답받는 RSS 채널에 대한 크기(BMSChannelSize, PMSChannelSize)와 통신 횟수 n 값이다. BMS의 경우 BMSChannelSize의 크기를 사용자나 시스템에서 줄일 수 있는 방법이 전혀 제공되지 않으며 통신 횟수 n 역시 사용자가 임의로 조절하는 방법 외에 특별한 방법이 없다. PMS 모델에서는 n과 PMSChannelSize를 줄일 수 있는 다양한 방법이 있을 수 있다. 본 논문은 다음 절에서 이 두 값을 줄일 수 있는 방법들을 나열하였으며 이 중에서 간단한 방법 선택하여 PMS 알고리즘을 구현하였다.

4.4.4 PMS에서 통신 비용 절감 방법 고찰

그림 9는 PMS에서 n과 PMSChannelSize를 줄이기 위한 가능한 방법들을 제안하고 체계적으로 정리하였다.

우선 n을 줄이기 위한 알고리즘들은 다음과 같다.

- “on/off selection” : 모바일 단말기의 정기 구독기가 주기적으로 RSS 피드를 가져오는 활동을 제어하는 방법이다.
- “refresh cycle recommendation” : 임의의 채널에 대해 최적의 정기 구독 주기를 결정하여 n을 줄이는 방법이다.
- “update detection by proxy” : MWProxy에서 통계적으로 채널에 대해 최적의 정기 구독 주기를 계산하여 모바일 단말기에 추천함으로써 n을 줄이는 방법이다.

채널 크기를 줄이는 방법들은 다음과 같다.

- “compression” : MWProxy에서 RSS 채널을 압축하여 모바일 단말기로 전송하여 통신 비용을 줄이는 방법이다.
- “item recommendation” : RSS 채널을 구성하는 아이템의 개수를 조절하는 방법이다.
- “time-based” : RSS 채널에서 시간을 고려하여 아이템을 선별하는 방법이다.

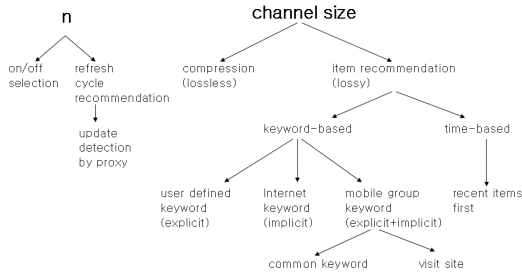


그림 9. 통신 비용을 줄이기 위한 방법들

- “recent items first”: MWProxy가 RSS 채널의 아이템들이 만들어진 시간을 기준으로 이전에 보낸 아이템을 제외하고 최신의 아이템 즉 새로운 아이템만을 보내는 방법이다.
- “keyword-based”: 사용자에게 가치 있는 정보를 구분하여 아이템의 개수를 줄이는 방법으로서, 키워드를 중심으로 아이템을 선별한다.
- “user defined keyword”: 사용자가 제시한 키워드를 중심으로 아이템을 선별하는 방법이다.
- “internet keyword”: 현재 인터넷에서 가장 관심 있는 정보로 추천되는 키워드를 검색하고 이 키워드를 포함하는 아이템을 선별하는 방법이다.
- “mobile group keyword”: 동일한 MWProxy에 등록된 단말기 들이 공통적으로 관심 있는 키워드를 판별하여 이 키워드를 포함하는 아이템을 선별하는 방법이다.
- “common keyword”: 모바일 단말기 들이 명시적으로 제시한 키워드로부터 공통 키워드를 찾아내는 방법이다.
- “visit”: 모바일 단말기의 사용자들이 채널의 여러 아이템 중 실제 정보를 획득한 아이템 (visit item)을 MWProxy에게 통보하여 이들로부터 정보의 관심도를 계산하고 추천하는 방법이다.

4.4.5 PMS 알고리즘

PMS 알고리즘의 목표는 이동 중에도 사용자가 원하는 정보를 얻을 수 있고 동시에 통신 비용을 줄이는데 있다. PMS에서 채널의 간소화를 위해 쉽게 구현 가능한 “recent items first”의 알고리즘을 사용한다. MWProxy는 사이트로부터 RSS 채널을 가지고 오면 그 사이트로부터 가져온 이전의 RSS 채널과 비교하여 새로 추가된 아이템을 우선적으로 선별하여 간소화된 채널을 구성한다. 이때 사용자는

통신 비용을 줄이기 위해 간소화된 채널에 포함될 아이템의 개수에 제한을 가할 수 있다.

V. 구현

본 논문에서는 MWReader를 위피(WIFI) 플랫폼 상에서 작동하는 jlet 응용 프로그램으로, MWProxy는 리눅스를 탑재한 PC 상에서 자바 서브릿(Servlet)으로 작성하였다. MWReader는 아로마 위피 에뮬레이터를 이용하여 PC에서 테스트를 수행한 후, 위피 플랫폼을 탑재한 SKT 휴대폰으로 실제 테스트를 수행하였다.

그림 10은 RSS 사이트를 등록하는 화면이며, 그림 11은 MWProxy로부터 받은 채널 Reuters_Sports의 아이템 리스트를 보는 화면이며, 그림 12는 “Stairs blast lifts Phillies past Dodgers” 아이템을 선택하였을 때 웹 정보를 받아보는 화면이다.

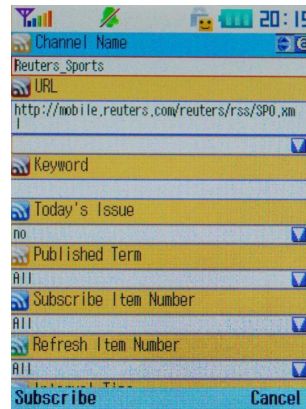


그림 10. RSS 채널 등록

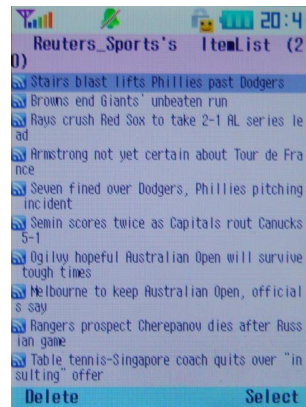


그림 11. 채널의 아이템 리스트 보기

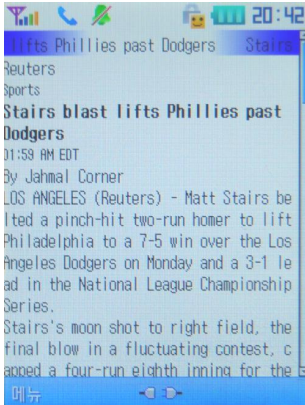


그림 12. 아이템 선택, 웹 정보 보기

VI. 성능 평가

6.1 성능 평가 방법

PMS와 BMS의 성능을 비교하기 위해 위피 자바를 이용하여 BMS 시스템을 추가적으로 구현하였으며, 통신 비용을 나타내는 수식 (1)과 수식 (2)의 응답 시간을 분할하여 구체적으로 표 2와 같은 성능 평가 지수를 정하였다. 또한 시간 측정을 위해 에뮬레이터에서 BMS와 PMS를 실행시키고 실측하는 방법을 이용한다. 이동 통신의 특성상 실험 장소, 시간, 환경에 따라 성능이 매우 가변적이어서 PMS와 BMS 시스템을 모두 휴대폰에 탑재하여 실측하였지만 응답 시간이 불규칙하게 나타났기 때문이다.

표 2. 성능 평가 지수

평가 지수	단위	의미	방법
채널 등록 시간	ms	각 채널에 대해 1회적 사건으로 성능에 미치는 영향은 미미	에뮬레이터에서 실측
채널 갱신 시간	ms	등록된 채널에 대해 주기적으로 반복되므로 통신 비용에 미치는 영향이 매우 큼	에뮬레이터에서 실측

6.2 성능 평가 결과

6.2.1 채널 등록 성능

RSS 채널을 제공하는 5 개의 사이트를 대상으로 채널을 등록하는데 소요되는 시간을 측정하였다.

채널 등록에 따른 응답 시간

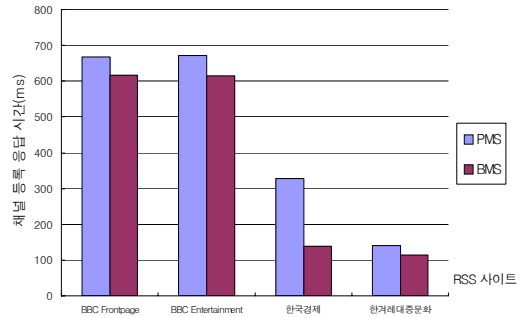


그림 13. 채널 등록에 따른 응답 시간

이 결과는 그림 13과 같으며, PMS의 채널 등록 시간이 BMS보다 높게 나타났다. 당연한 결과로서 BMS의 경우는 프록시 서버를 거치지 않고 바로 RSS 사이트에 접속하여 RSS 채널 피드를 가져오지만, PMS의 경우 MWProxy가 RSS 사이트로부터 RSS 채널 피드를 가지고 오는 과정을 거치기 때문이다.

6.2.2 채널 갱신 성능

채널 갱신에 따른 응답 성능을 측정하기 위해 우선 1시간 간격으로 PMS 모델로 MWReader가 채널을 갱신하도록 하였다. MWProxy는 등록된 사이트에서 주기적으로 RSS 채널을 가지고 와서 채널의 아이템을 분석하여 이전에 전송받은 동일 채널의 아이템과 비교하여 새로운 아이템을 골라내고 이들을 간소화된 채널(Simplified RSS Channel)로 재구성하여 모바일 단말기로 보낸다. 본 실험은 원본 채널에 비해 간소화된 채널에 포함된 아이템의 개수를 비율 S로 다음과 같이 정의하고 이를 측정하였다.

$$S(\%) = \frac{\text{간소화된 채널 아이템 개수}}{\text{원본 채널의 아이템 개수}}$$

실험 결과는 그림 14와 같다. 이 결과가 의미하는 것은, 한국 경제 사이트의 경우 약 7시간이 지나면 채널 아이템이 100% 다른 아이템으로 변하게 되며 BBC Entertainment 사이트의 경우 약 4시간 지나면 채널 내에 3% 만이 새로운 아이템으로 추가되었음을 의미한다. 이 결과는 만일 어떤 사용자가 단말기를 이용하여 BBC Entertainment 사이트의 뉴스 정보를 얻기 위해 BMS 알고리즘을 이용하는 경우 항상 채널 전체를 가져 와야 하지만, PMS

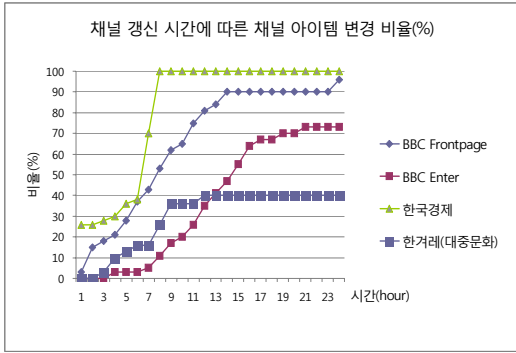


그림 14. PMS에서 채널 갱신 시간에 따른 채널 아이템 변경 비율

알고리즘을 이용하는 경우는 모바일 단말기가 그중 3%의 아이템만 MWProxy로부터 가져오게 됨을 의미한다. 이는 모바일 단말기의 무선 수신량을 BMS의 약 3%의 수준으로 낮추는 효과를 의미한다.

그림 15는 BBC Entertainment 사이트의 채널 갱신에 따른 채널 갱신 응답 시간을 측정된 값이다. PMS 알고리즘의 응답시간이 BMS에 비해 짧은 결과를 얻었다. 그러나 이 시간은 절대 시간으로서 MWProxy의 구현이나 네트워크의 상황 BBC Entertainment 사이트의 작동 등에 절대적인 영향을 받기 때문에 그 결과에 대한 정확한 해석은 어렵다. 다만 확실한 것은 PMS는 BMS에 비해 모바일 단말기가 채널 갱신 시에 전송받는 채널의 크기가 작다는 점과 MWProxy는 미리 준비된 채널 데이터를 바로 전송하기 때문에 응답 시간이 더욱 짧아질 수 있다는 점이다.

그림 14와 그림 15의 결과에서도 볼 수 있는 것처럼 채널 갱신에 따른 PMS 알고리즘의 성능이 우수하며, 채널 등록은 한 채널에 대해서 한 번 일어나지만 채널 갱신은 지속적으로 일어나기 때문에

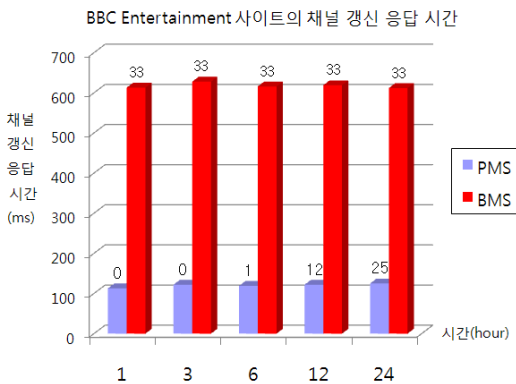


그림 15. 채널 갱신 응답 시간

PMS 시스템은 단말기의 무선 통신 시간과 양을 지속적으로 줄인다고 결론 내릴 수 있다.

VII. 결 론

본 논문은 모바일 단말기 상에서도 메모리 사용량과 무선 통신량을 줄이고 웹 신디케이션을 이용할 수 있는 모바일 웹 신디케이션 시스템을 제안하였다. 제안된 시스템은 모바일 단말기와 RSS 웹 사이트 사이에 프록시 서버를 두고 프록시 서버가 RSS 사이트로부터 주기적으로 가지고 오는 RSS 채널의 아이템 개수를 줄이는 방식으로 모바일 단말기의 무선 통신량을 줄인다.

본 논문에서는 프록시 서버를 위한 MWProxy 서버를 웹 기반으로 구현하고 RSS 채널을 정기 구독할 수 있는 MWReader 소프트웨어를 모바일 단말기에 위피 기반으로 실제 구현하였다. 그리고 에뮬레이터 상에서 MWReader를 직접 실행하여 성능을 측정하였다. 성능 측정 결과 MWProxy를 이용하여 채널 갱신을 대행하고 채널의 크기를 간소화함으로써 MWReader에 의해 주기적으로 수집되는 RSS 채널의 크기는 줄어드는 결과를 얻었다.

참 고 문 헌

- [1] Reusch P.J.A., Stoll, B., Schulwandt T., Serwatowski P., "New Communication Concepts Based upon Advanced RSS Feeds," *The Third Workshop on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications*, pp. 699-702, Sept. 2005.
- [2] Ying Zhou, Xin Chen, Chen Wang, "A Self-Organizing Search Engine for RSS Syndicated Web Contents," *Proceedings of the 22nd International Conference on Data Engineering Workshops*, 2006.
- [3] Luo Zhong, Shishi Huang, Peng Liu, Yi Yu, "A Design for an Online RSS Reader Based On AJAX," *ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing*, pp.794-798. 2007.
- [4] RSS 2.0 specification, <http://validator.w3.org/feed/docs/rss2.html>.

- [5] Mark Nottingham and Robert Sayre, "The Atom Syndication Format," IETF RFC 4287, Dec 2005.
- [6] 박지강, "당신은 웹 2.0 개발자입니까?," *한빛미디어*, 2007.
- [7] 설재민, 김성환, "효율적인 경량 모바일 RSS 리더의 구현," *한국정보처리학회 춘계학술발표대회 논문집 14권 1호*, pp.637-638, 2007.
- [8] Baumgart A.S., Knapp H., Suetterlin P., Schader M., "A Profile-Based Peer-to-Peer RSS Information Distribution," *Wireless Pervasive Computing*, 2007.
- [9] Robbie De Sutter, Sam Lerouge, Davy De Schrijver, Rick Van de Walle, "Enhancing RSS Feeds: Eliminating Overhead through Binary Encoding," *Proceedings of 3rd International Conference on Information Technology and Applications*, 2005.
- [10] 한승현, 류동엽, 임영환, "유비쿼터스 환경을 위한 RSS 뉴스 채널 콘텐츠의 개인화 모바일 서비스 기법," *정보처리학회 논문지 14-D권 4호*, pp.427-434, 2007.
- [11] Chmielewski, D., Gongzhu Hu, "A distributed platform for archiving and retrieving RSS feeds," *4th Annual ACIS International Conference on Computer and Information Science*, pp.215-220, 2005.

황기태 (Kitae Hwang)

정회원



1986년 2월 서울대학교 컴퓨터 공학과 졸업
 1988년 2월 서울대학교 컴퓨터 공학과 석사
 1994년 2월 서울대학교 컴퓨터 공학과 박사
 2000~2001년 University of California, Irvine 방문 교수

1994년~현재 한성대학교 컴퓨터공학과 교수
 <관심 분야> 분산 시스템, 모바일 시스템

김남윤 (Namyun Kim)

정회원



1992년 2월 서울대학교 컴퓨터 공학과 졸업
 1994년 2월 서울대학교 컴퓨터 공학과 석사
 2000년 2월 서울대학교 컴퓨터 공학과 박사
 1999년~2002년 2월 삼성전자

무선사업부 책임 연구원
 2002년~현재 한성대학교 정보시스템공학과 부교수
 <관심 분야> 멀티미디어 통신, 정보 보안