

트위터 사용자와 팔로워들 간의 실시간 메시지 교류 시스템 개발

박종은*, 권오진*, 이홍창*, 이명준**

A Real-Time Messaging System for Twitter Users and Their Followers

Jong-Eun Park *, O-Jin Kwon *, Hong-Chang Lee *, Myung-Joon Lee **

요약

최근 급속히 보급된 스마트폰과 소셜 네트워킹 서비스의 발전은 가상 세계와 실세계를 보다 밀접하게 연결하여 사람들 간의 다양한 상호작용을 가능하게 하고 있다. 일반적인 소셜 네트워킹 서비스는 사용자들 간의 네트워크를 쉽게 구성하는 방법에 치중하고 있으며 네트워크에 참여한 다른 사용자들과 단순한 정보 교환 기능을 지원한다. 세계적으로 널리 사용되는 서비스인 트위터는 정보 교환 기능으로 트윗이라는 단문 메시지만을 사용하고 있으며 2억 명이 넘는 사용자를 확보하고 있다. 논문에서는 대표적인 SNS인 트위터의 소셜 네트워크를 기반으로 실시간 그룹 채팅을 지원하는 기법을 제안하고 이를 활용하여 스마트폰 그룹 메시징 시스템을 개발하였다. 트위터의 소셜 네트워크를 반영하여 트위터 사용자와 그 사용자의 팔로워들이 참여하는 그룹을 자동적으로 형성하고 그룹 구성원이 모두 참여할 수 있는 실시간 그룹 메시징을 지원하는 기법을 제안하였다. 그리고 이를 바탕으로 개발된 스마트폰 그룹 메시징 시스템은 XMPP 프로토콜 기반의 메시징 서버와 트위터의 소셜 네트워크를 기반으로 실시간 메시징을 수행하는 스마트폰 클라이언트로 구성된다. 사용자는 트위터 메시지를 이용하여 손쉽게 메시징 시스템을 사용할 수 있으며, XMPP 서버에 자동적으로 형성되는 그룹을 통하여 자신의 팔로워들과 메시지를 실시간으로 교환할 수 있다.

▶ Keyword : 트위터, 소셜 네트워크, 소셜 네트워킹 서비스, 스마트폰, 메시징 시스템, 실시간 그룹 채팅

Abstract

Recently, as smartphones have rapidly come into wide use and various social networking services have grown, people perform a variety of interactions through the virtual and real world based on them. Usually, such services focus on easy formation of social links among users, supporting the exchange of simple messages among users on the networks. Twitter, one of such

• 제1저자 : 박종은 • 교신저자 : 이명준

• 투고일 : 2011. 04. 29, 심사일 : 2011. 05. 18, 게재확정일 : 2011. 06. 01.

* 울산대학교 컴퓨터정보통신공학부(School of Computer Engineering & Information Technology, University of Ulsan)

** 울산대학교 전기공학부(School of Electrical Engineering, University of Ulsan) 교수

※ 본 논문은 2011년도 정부(교육과학기술부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구 사업임 (No.2011-0006111)

services used worldwide, supports short message service named Tweet and has over 200 million members signed up. In this paper, we propose techniques for supporting real-time group messaging based on the social network of Twitter and describe a smartphone group messaging system developed with the techniques. The system automatically forms a group that include a Twitter user and the followers of the user, supporting real-time group messaging among the group members. The developed system is composed of an XMPP protocol-based messaging server and smartphone client applications which perform real-time messaging based on the social network of Twitter. Twitter users can easily use the system utilizing Tweet messages, exchanging real-time messages with their followers within the groups instantly established on the XMPP server.

▶ Keyword : Twitter, Social Network, Social Networking Service, Smartphone, Messaging System, Real-time Group Messaging

I. 서론

소셜 네트워킹 서비스(Social Networking Service, SNS)[1]는 사용자들 간 공통의 관심사나 활동 등을 바탕으로 서로 간의 사회적 관계를 형성하고 이를 반영한 온라인 기반의 서비스를 제공한다. 이러한 소셜 네트워킹 서비스는 인터넷 기반의 가상 세계에서 사용자들을 각각 고유하게 표현하며, 각 사용자들 간의 사회적 관계를 구성하여 커다란 소셜 네트워킹을 정의한다. 그리고 이를 바탕으로 다양한 서비스를 제공하면서 많은 사용자를 확보하고 있으며, 다양한 분야에서 활용되고 있다. 특히, 최근 급속히 보급된 스마트폰과 소셜 네트워킹 서비스의 발전은 가상 세계와 실세계를 보다 밀접하게 연결하여 사람들 간의 다양한 상호작용을 가능하게 한다.

트위터(Twitter)[2,3]는 세계적으로 널리 사용되는 소셜 네트워킹 서비스로서 간편한 설정으로 다양한 사용자들과의 소셜 네트워킹을 손쉽게 구성하며, 단문 메시지 전송하는 기능을 제공하여 간단한 정보를 편리하게 공유할 수 있다. 트위터는 사용자와 사용자의 정보를 수신하는 팔로워(Follower)를 정의하며, 이러한 요소들을 체계적으로 결합하여 소셜 네트워킹을 구성한다. 사용자가 작성한 단문 메시지는 트위터 서버를 통하여 자동적으로 해당 사용자의 팔로워들에게 전달되며, 해당 팔로워들은 자신의 팔로워들에게 메시지를 전달하면서 정보가 신속하게 확산될 수 있다. 최근, 스마트폰의 급속한 보급과 함께 트위터를 이용하는 사용자들이 늘어나고 있으며, 이동 환경 중에 트위터를 사용하여 정보를 공유하는 경우도 빈번해지고 있다.

트위터는 단순히 메시지를 전달하는 기능에 그치지 않고

자신의 기능을 OpenAPI 형태로 제공하는 하나의 거대한 웹 플랫폼으로 발전하고 있다. 트위터의 계정 정보를 이용하여 사진 공유 서비스를 제공하는 트윗픽(TwitPic)[4], 캠을 이용한 실시간 웹 방송을 지원하는 트윗캠(TwitCam)[5], 설문조사나 특정 주제에 관한 랭킹 서비스를 제공하는 트윗폴(TwitPoll)[6], 트위터 계정을 이용한 채팅 서비스를 제공하는 타이니챗(tinychat)[7] 등의 서비스가 사용되고 있다.

본 논문에서는 대표적인 SNS인 트위터의 소셜 네트워킹을 기반으로 실시간 그룹 채팅을 지원하는 기법을 제안하고 이를 활용하여 스마트폰 그룹 메시징 시스템을 개발하였다. 트위터의 소셜 네트워킹을 반영하여 트위터 사용자와 그 사용자의 팔로워들이 참여하는 그룹을 자동적으로 형성하고 그룹 구성원이 모두 참여할 수 있는 실시간 그룹 메시징[8]을 지원하는 기법을 제안하였다. 그리고 이를 바탕으로 개발된 스마트폰 그룹 메시징 시스템은 XMPP(eXtensible Message Presence Protocol) 프로토콜[9] 기반의 메시징 서버와 트위터의 소셜 네트워킹을 기반으로 실시간 메시징을 수행하는 스마트폰 클라이언트로 구성된다. XMPP 기반의 Openfire[10] 메시징 서버는 클라이언트와 비동기적으로 연결되어 트위터의 소셜 네트워킹 정보를 바탕으로 사용자와 그룹 정보를 표현한다. 그리고 스마트폰 그룹 메시징 클라이언트는 aSmack[11] 라이브러리를 통하여 Openfire 서버와 XMPP 통신을 수행하며, 트위터의 사용자와 팔로워들이 손쉽게 그룹 채팅을 수행할 수 있는 기능을 제공한다. 사용자는 트위터 메시지를 이용하여 손쉽게 메시징 시스템을 사용할 수 있으며, XMPP 서버에 자동적으로 형성되는 그룹을 통하여 자신의 팔로워들과 메시지를 실시간으로 교환할 수 있다.

본 논문의 구성은 다음과 같다. 서론에 이어 2장에서는 SNS와 XMPP에 대해 기술하며 3장에서는 소셜 네트워킹 기

반의 메시징 시스템과 이를 지원하는 기법을 제시한다. 그리고 4장에서는 제시된 기법을 바탕으로 시스템의 설계와 구성에 대해 다루며, 5장에서는 구현결과와 기존의 SNS 기반의 메시징 시스템의 비교를 보여준다. 마지막으로 6장에서는 결론을 다룬다.

II. 관련연구

1. 소셜 네트워킹 서비스

소셜 네트워킹 서비스는 사람들이 다양한 상호 관계를 구성할 수 있도록 지원하는 인터넷 기반의 서비스로서 트위터, 페이스북[12], 싸이월드 등이 대표적으로 사용되고 있다.

트위터는 다양한 개발 라이브러리를 통하여 개발자가 손쉽게 트위터 기반의 서비스를 개발하고 연동할 수 있도록 지원한다. 개발을 위한 대표적인 트위터 라이브러리는 Twitter4J[13], Java-twitter[14], Jtwitter[15] 등이 있다. 이러한 라이브러리를 활용하여 다양한 서비스를 제공하는 많은 시스템이 개발되었고, 트위터 사용자를 위한 메시징 서비스를 제공하는 타이니챗, 세이버챗(SavorChat)[16] 등의 서비스가 등장하였다.

페이스북(Facebook)은 6억 명 이상의 사용자를 확보한 세계 최대의 소셜 네트워킹 서비스이다. 페이스북은 사용자들이 친구(Friends) 목록을 공유함으로써 손쉽게 소셜 네트워크를 구성하며, 담벼락(WALL) 기능을 통하여 실시간으로 정보를 공유한다. 또한 OpenAPI를 지원하여 다양한 서비스와 연동하여 사용할 수 있다.

싸이월드(CyWORLD)는 대표적인 국내 서비스로 사용자들의 개인 공간을 표현하는 미니홈피를 제공하며, 인맥 관계를 정의하는 일촌 기능을 통하여 사용자의 다양한 정보를 공유할 수 있다. 그리고 네이트온 메신저와 연동하여 메시징 기능을 수행할 수 있다. 그러나 시스템의 폐쇄적인 정책 및 구조로 인하여 다양한 서비스와 연동되기는 어렵다.

2. XMPP

XMPP는 인스턴스 메시징을 위한 표준 프로토콜로서 인터넷에서 사용자 간 실시간 메시지나, 사용자 상태 정보(Presence) 등을 효율적으로 교환할 수 있게 한다. 이러한 XMPP는 1999년에 Jabber 공개 소스 커뮤니티에서 개발되어 2004년에 메시징을 위한 표준 프로토콜로 채택되면서 이를 이용한 많은 XMPP 기반의 메시징 시스템이 개발되었다. XMPP 기반의 대표적인 메시징 시스템은 구글 톡(Google Talk)[17]과 Facebook, Apple의 MobileMe[18] 서비스 등이 있다.

XEP(XMPP Extension Protocol)[19]는 기존 XMPP를 확

장하여 다양한 기능을 지원하는 프로토콜을 제공하도록 XMPP 표준 재단(XSF : XMPP Standards Foundation)에서 발표되었다. 그리고 기능별 프로토콜을 정의한 문서를 통해 XMPP 기반의 시스템의 개발을 지원한다.

표 1. XEP에서 추가된 주요 기능
Table 1. Key Features Added to XEP

기능(문서 이름)	
·서비스 발견(XEP-0030)	·다중 채팅(XEP-0045)
·클라이언트 구분(XEP-0115)	·사용자 상태 표현(XEP-0126)
·HTTP 바인딩(XEP-0124)	·파일 전송 (XEP0096)

Openfire는 XMPP Core[20]와 사용자 다중 그룹을 지원하는 XMPP IM[21] 프로토콜을 비롯하여 XEP의 주요 기능을 지원하는 대표적인 XMPP 서버로서 데이터베이스와 연동하여 웹을 통해 서버의 자원을 손쉽게 관리 할 수 있다. 그리고 서버의 기능을 확장하여 다양한 기능을 제공할 수 있도록 오픈소스로 제공된다. aSmack은 데스크톱 기반의 XMPP 지원 라이브러리인 Smack의 안드로이드 버전으로, 스마트폰에서 XMPP 통신을 수행하기 위한 기능들을 제공한다.

III. 트위터 기반의

그룹 메시징 시스템 기법

많은 사용자들의 다양한 상호작용을 표현하기 위하여 새로운 소셜 네트워크를 구성하는 것은 대단히 어려운 작업이며 많은 비용과 시간을 필요로 한다. 그리고 사용자들의 지속적인 참가와 상호작용에 따라 소셜 네트워크는 그 형태가 매우 빈번히 재구성된다. 트위터는 이미 많은 사용자들을 확보하고 있으며 이를 바탕으로 풍부한 소셜 네트워크를 구성하고 있다. 그리고 OpenAPI를 통하여 기본적인 트위터 기능은 물론 트위터의 소셜 네트워크 정보 또한 공개적으로 제공하기 때문에 이를 기반으로 한 다양한 서비스를 개발할 수 있는 환경을 마련하고 있다.

트위터는 간단한 단문 메시지 전송 기능과 풍부한 소셜 네트워크로 인하여 많은 사용자들이 활용하고 있으며, 특히 스마트폰의 급속한 보급과 함께 이동 환경에서도 그 활용도가 점점 높아지고 있다. 하지만 트위터는 공개된 형태의 메시지만으로 정보를 공유하기 때문에 비공개적인 형태로 수행되는 작업이나 정보 공유는 지원하기 어렵고 비동기적인 메시지 전달 방식으로 인하여 정보 공유가 실시간으로 이루어지기 어렵다. 따라서 트위터에 구축된 소셜 네트워크를 이용하여 많은 사용자들의 상호작용을 반영하고 이를 기반으로 정보 교환의

기밀성을 보장할 수 있는 실시간 메시징 시스템이 필요하다. 소셜 네트워크는 사용자들의 다양한 상호작용으로 인하여 그 형태가 매우 빈번히 재구성된다. 재구성된 소셜 네트워크는 해당 사용자가 시스템에 접근할 때 확인하는 것이 일반적이다. 따라서 트위터의 소셜 네트워크가 재구성될 때마다 메시징 시스템에 네트워크 정보를 반영하는 것은 매우 비효율적이다. 본 연구에서는 일반적으로 사용자가 시스템에 접근할 때만 재구성된 네트워크 정보를 확인하는 것에 착안하여 메시징 시스템에 효율적으로 소셜 네트워크를 반영하는 기법을 고안하였다. 다음은 이 알고리즘의 진행 단계를 보여준다.

- (1) 트위터의 사용자 계정별로 메시징 시스템에 사용자 계정 등록
- (2) 사용자가 메시징 시스템에 접속할 때, 자신과 팔로워들 간의 연결정도를 나타내는 소셜 네트워크 정보를 트위터에 요청
- (3) 트위터로부터 전달받은 정보를 바탕으로 메시징 시스템에서 사용자와 연관된 팔로워들을 하나의 그룹으로 변환하여 정의
- (4) 사용자별로 자신과 팔로워의 연결 정보가 메시징 시스템에 그룹으로 표현
- (5) (2)~(4)의 과정이 반복되면서 트위터 전체 사용자의 소셜 네트워크 정보가 메시징 시스템의 사용자와 그룹 정보로 저장
- (6) 메시징 시스템에 접속한 사용자가 자신의 팔로워들이 참여 가능한 그룹 채팅방 개설
- (7) 채팅방이 개설되면 자동으로 채팅방 참가 의사를 질의하는 트위터의 단문 메시지를 팔로워들에게 전송
- (8) 팔로워들은 전달 받은 메시지를 통하여 메시징 시스템을 실행하고 그룹 채팅에 참가

이 기법에서는 사용자가 트위터에 접근했을 때 본인의 소셜 네트워크를 확인하는 것에 착안하여, 트위터에 접속 시 메시징 시스템에 해당 사용자의 소셜 네트워크 정보를 업데이트하는 방법으로 정의하였다. 그리고 트위터에 접속 중일 때 변경되는 소셜 네트워크 반영 시간을 주기적으로 확인하여 메시징 시스템에 저장한다. 그림 1은 이러한 기법의 동작 과정을 보여준다.

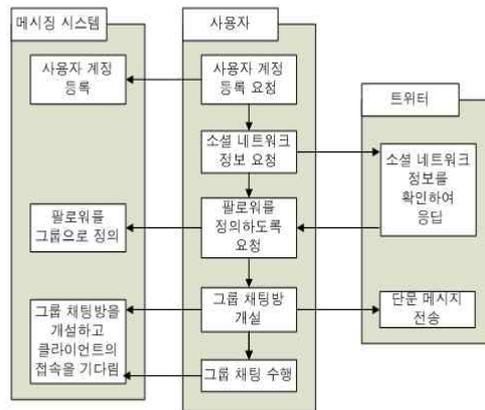


그림 1. 제안된 기법의 동작과정
Fig. 1. Operation Process of the Proposed Technique

본 연구에서는 정의된 알고리즘을 바탕으로 트위터 기반의 실시간 메시징을 수행할 수 있는 스마트폰 그룹 채팅 시스템을 개발하였다.

IV. 스마트폰 그룹 메시징 시스템

1. 시스템 구성 및 설계

스마트폰 그룹 채팅 시스템은 XMPP 프로토콜을 이용하는 메시징 서버와 이동 환경에서 사용자 간 실시간 메시징을 수행할 수 있는 스마트폰 클라이언트로 구성된다. 그림 2는 스마트폰 그룹 채팅 시스템의 전체 구조를 보여준다.

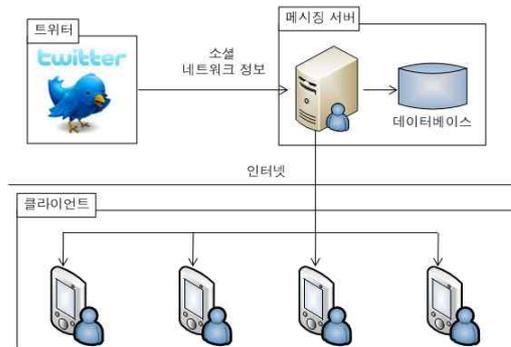


그림 2. 트위터 기반 그룹 메시징 시스템 구조
Fig. 2. The Structure Of Twitter-Based Group Messaging System

Openfire 메시징 서버는 자신의 데이터베이스에 트위터의 소셜 네트워크를 바탕으로 정의한 사용자와 그룹정보를 저장

한다. 이러한 정보를 바탕으로 스마트폰 클라이언트에서는 각 사용자와 그룹 정보를 확인하고 실시간 메시지를 수행할 수 있다.

메시징 서버는 이러한 정보를 저장하기 위하여 클라이언트와 함께 동작한다. 클라이언트를 통하여 사용자가 시스템에 접속할 때 해당 계정으로 트위터에 접근하여 추출한 팔로워 정보를 서버의 저장소에 저장한다.

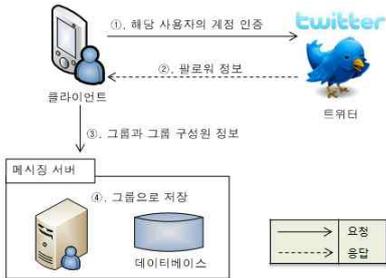


그림 3. 메시징 서버의 소셜 네트워크 정보 저장
Fig. 3. Social Network Information Stored in the System

그림 3은 메시징 서버와 클라이언트가 연동하여 트위터의 소셜 네트워크 정보를 그룹으로 표현하여 메시징 시스템에 저장하는 과정을 보여준다.

메시징 시스템에 등록된 트위터 사용자들이 편리하게 채팅 기능을 수행할 수 있도록 트위터 메시지에서 메시징 서비스를 바로 실행할 수 있도록 지정하는 프로토콜을 정의하였다. 그리고 채팅 개설과 동시에 참여 정보를 포함한 메시지를 트위터 팔로워에게 자동으로 전송하여 손쉽게 그룹채팅이 이뤄지도록 지원한다. 채팅방 개설과 함께 팔로워는 이와 관련된 트위터 메시지를 전달받아 채팅방을 확인하고 메시지내 실행 기능을 이용하여 즉시 채팅에 참가할 수 있다. 표 2는 정의한 프로토콜의 기능을 보여준다.

표 2. 편리한 메시지를 지원하기 위해 정의된 프로토콜
Table 2. Protocols to Support Easy Messaging

프로토콜 요소	기능
tchat://	클라이언트의 실행 조건
id	그룹 채팅방의 개설자의 트위터 아이디
service.host	서비스 종류와 Openfire 서버의 호스트 명
id@service.host	개설된 채팅방의 이름
현재시간	중복된 단문 메시지를 허용하지 않는 트위터의 정책을 피하기 위함

그림 4는 채팅방 생성시 전달되는 트위터 메시지의 프로토콜 내용과 채팅방에 관련된 각 요소 간 상관관계를 보여준다.

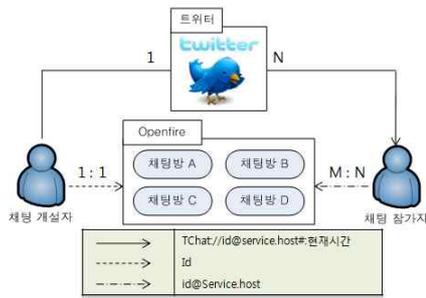


그림 4. 메시징 시스템을 실행하기 위한 요소들의 관계
Fig. 4. Relationship among Elements for Execution of the Messaging System

2 스마트폰 그룹 메시징 클라이언트의 구현

스마트폰 그룹 메시징 클라이언트는 트위터의 인증 기능과 단문 메시지 등록 및 확인 기능을 지원하고 Openfire 서버와 통신하며 메시지를 송·수신할 수 있다. 이러한 클라이언트를 이용하여 사용자는 그림 5와 같은 기능을 수행할 수 있다.

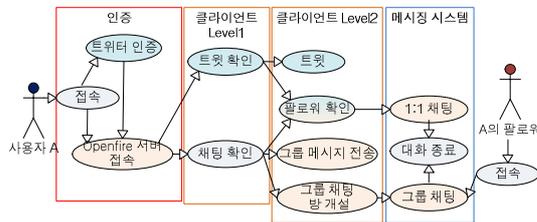


그림 5. 트위터 기반의 채팅 과정
Fig. 5. The Process of Twitter-Based Chatting

개발된 클라이언트는 트위터와의 연동을 위한 트위터 모듈과 메시징 서버와 연동하여 메시징 서비스를 실행하기 위한 메시징 모듈로 구성된다. 그림 6은 스마트폰 클라이언트의 내부 구조를 보여준다.



그림 6. 스마트폰 클라이언트의 내부구조
Fig. 6. The Internal Structure Of The Smartphone Client

트위터 모듈은 Twitter4j 라이브러리를 이용하여 트위터 서버와 통신하며, 트위터의 소셜 네트워크 정보를 불러오거나 트위터 단문 메시지 전송을 위한 내부 요청을 처리한다.

메시징 모듈은 aSmack 라이브러리를 이용하여 메시징 서버와 통신하며, 서버상의 그룹 및 그룹 구성원 목록 확인, 채팅방 개설 및 실시간 메시징 등의 기능을 위한 내부 요청을 처리한다. 클라이언트의 트위터 모듈과 메시징 모듈을 이용하여 트위터의 소셜 네트워크 정보를 메시징 서버에 적용하는 과정은 다음과 같다.

- (1) 트위터 계정을 OAuth 서비스를 통해 인증
- (2) 인증이 성공하면 AccessToken을 트위터로부터 부여받음
- (3) AccessToken을 클라이언트에 파일로 저장
- (4) 저장된 AccessToken을 이용하여 트위터에 접속
- (5) 트위터 계정 정보를 바탕으로 AccountManager 객체를 이용하여 Openfire에 계정을 생성하고 접속
- (6) 트위터의 팔로워를 리스트 형식으로 받아옴
- (7) 팔로워 정보를 바탕으로 Openfire에 Roster 객체를 통하여 사용자와 그룹으로 정의하도록 요청

그림 7은 트위터의 소셜 네트워크 정보를 메시징 서버에 반영하는 기능을 구현한 소스코드의 일부를 보여준다.

```

/*트위터의 팔로워 리스트를 반환*/
public PagableResponseList<User> getFollowers(){
    PagableResponseList<User> myFollowersList = null
    try{
        String myName = twitter.getScreenName();
        long cursor = -1;
        do {
            //팔로워 목록을 받아옴
            myFollowersList =
twitter.getFollowersStatuses(myName,
cursor);
        } while ((cursor = myFollowersList.getNextCursor()) !=
0);
    } catch (TwitterException te) {
        te.printStackTrace();
    }
    return myFollowersList;
}

/*Openfire 서버에 등록*/
Roster roster = xc.getRoster();
//팔로워 목록을 순회하며 필요정보를 추출
for(User follower:tc.getMyFollowersList()){
    String name = follower.getScreenName();
    String nickName = follower.getName();
    name = name.toLowerCase();
    try {
        if(!roster.contains(name)){
            //Openfire에 사용자로 등록
            roster.createEntry(name, nickName, null);
            ...
        }
    }
}

```

그림 7. 메시징 시스템에서의 소셜 네트워크 정보 처리 구현
Fig. 7. Implementation for Handling Social Network Information on the Messaging System

그림 7의 getFollowers() 메소드는 Twitter4j의 인증된 사용자의 정보를 얻을 수 있는 twitter 객체를 이용하여 팔로워 목록을 PagableResponseList<User> 로 불러오고 반환한다. 팔로워 목록은 반복을 통해 aSmack 라이브러리의 Roster 객체를 이용하여 Openfire에 그룹과 그룹 구성원으로 저장된다. 그림 8은 메시징 시스템에 정의된 트위터 사용자 간의 그룹 메시징 수행 기능을 구현하는 소스코드의 일부를 보여준다.

```

/*그룹 채팅을 지원하는 클래스의 생성자*/
public XMPPGroupChatService(Connection connection, String user,
    ArrayList<MultiUserChat> multiUserList, final FileIo fi){
    //참가해야 하는 채팅방을 위한 MultiUserChat 인스턴스를 생성
    muc = new MultiUserChat(connection,
user+SERVICENAME);
    String roomName = muc.getRoom();
    String userName = connection.getUser();
    if(!searchingRoom(roomName, multiUserList)){
        try {
            int end = userName.indexOf('@');
            //그룹 채팅방에 참가
            muc.join(userName.substring(0,end));
        } catch (XMPPException e) { e.printStackTrace(); }
        //메시지를 응답받을 리스너 등록
        muc.addMessageListener(new PacketListener() {
            public void processPacket(Packet packet) {
                String from = packet.getFrom();
                String body = packet.toXML();
                int start = from.indexOf("/")+1;
                //메시지를 보내는 사용자 아이디 추출
                String sender = packet.getFrom().substring(
start,from.length());
                start = body.indexOf("body")+5;
                int end = body.lastIndexOf("</body");
                //메시지 내용 추출
                String msg = body.substring(start,end);
                //메시지를 파일에 기록
                fi.writeFile(sender+" : "+msg+"\n");
            }
        });
        multiUserList.add(muc);
        ...
    }
}

```

그림 8. 그룹 메시징 처리 구현
Fig. 8. Implementation for Handling Group messages on the System

그룹 메시징을 위한 구현은 채팅방의 개설, 참가, 초대, 떠나기 등의 기능을 수행할 수 있도록 aSmack의 MultiUserChat 객체를 이용한다. 클라이언트에서 그룹 메시징을 수행하면 MultiUserChat 객체가 생성되고 각각의 MultiUserChat 객체는 그룹채팅에서 주고받는 응답 메시지의 처리를 위한 PacketListener 인터페이스를 리스너로 가진다. 이러한 리스너

는 응답받은 내용을 클라이언트에서 해당 채팅방과 관련된 파일에 기록하며 클라이언트 화면에 출력한다. 또한 팔로워 수에 맞게 자동으로 채팅방 크기를 결정하며, 팔로워 수가 너무 많을 경우에는 채팅방 참여 최대 인원 수를 100명으로 한정한다.

V. 개발 결과 및 기능 평가

1. 시스템 개발 결과

개발된 메시징 시스템은 트위터와 연동하여 트위터의 기본적인 서비스를 이용할 수 있으며, 동시에 스마트폰 기반으로 이동환경을 비롯한 다양한 환경에서 트위터 사용자들과 편리하게 실시간 메시지를 수행할 수 있다. 그림 9는 트위터 팔로워를 웹에서 확인한 모습과 이를 Openfire에서 반영한 사용자를 보여준다.



그림 9. 메시징 시스템에 반영된 트위터의 소셜 네트워크
Fig. 9. Social Network in Twitter Reflected on the Messaging System

특정 사용자와 그룹으로 반영된 팔로워들과 그룹 채팅을 수행하기 위해 트위터 단문 메시지 형태로 제공되는 프로토콜은 그림 10과 같은 모습으로 트위터 기반의 다양한 어플리케이션에서 확인할 수 있다. 팔로워는 자신이 받은 트위터 단문 메시지를 스마트폰에서 터치함으로써 메시징 클라이언트를 실행시키고 자동적으로 그룹채팅에 참가하여 실시간 그룹 메시지를 수행한다.



그림 10. 그룹 채팅 참가를 지원하는 트위터 메시지
Fig. 10. Twitter Message to Support Automatic Participation in Group Chatting

그림 11은 개발된 클라이언트에서 확인한 트위터 팔로워와 이들과 함께 그룹 채팅을 수행하는 모습을 보여준다.

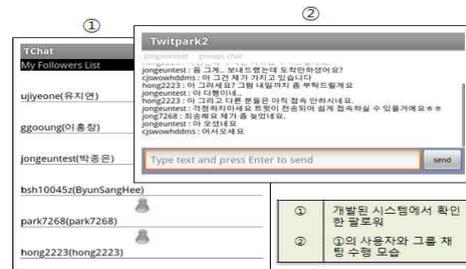


그림 11. 트위터 사용자들의 그룹채팅
Fig. 11. Group Chatting among Twitter Users

2. 타 시스템과의 기능 비교

트위터는 OpenAPI를 지원하며, 많은 사용자와 소셜 네트워크 정보를 정의하고 있기 때문에 이를 기반으로 한 여러 메시징 시스템이 등장하였다. 표 3은 본 연구에서 개발한 스마트폰 그룹 메시징 시스템과 함께 트위터를 이용하는 타 메시징 시스템과의 기능 비교를 보여준다.

표 3. 여러 SNS 기반 메시징 시스템의 특징
Table 3. Features Of each SNS-based Messaging System

	NateOn	TinyChat	개발된 시스템
SNS 기반 사용자 인증	X	O	O
실시간 그룹 채팅	O	O	O
SNS 기반 자동 그룹 정의	X	X	O
그룹 채팅 자동 초대 및 참가	O	X	O
클라이언트 플랫폼	윈도우/리눅스 /안드로이드	웹 브라우저	안드로이드

네이트온은 싸이월드와 연동하는 대표적인 국내 메신저 서비스로 다양한 플랫폼을 지원하고 효과적인 채팅을 위한 기능

들을 제공한다. 하지만 싸이월드의 일촌 관계를 그룹으로 직접적으로 표현하지 않으며 메신저의 사용자로 따로 싸이월드의 사용자를 등록해야 하며 그룹도 별도로 구성해야 하는 번거로움이 있다.

타이니챗은 트위터의 OpenAPI를 이용하는 대표적인 메시징 시스템으로 트위터의 계정을 그대로 이용하며, 자신의 팔로워를 초대하여 그룹 채팅을 수행할 수 있다. 하지만 웹 브라우저에서만 동작하기 때문에 많은 사용자가 참여할 경우 메시지의 입출력에 지연이 발생하며, 팔로워와의 그룹을 자동적으로 표현하지 못하기 때문에 다양한 상황에서 그룹 채팅을 수행하기에는 많은 어려움이 따른다.

본 연구에서 개발된 시스템은 소셜 네트워크를 메시징 시스템에 완전히 반영하고 이를 바탕으로 클라이언트에서 실시간 메시징을 수행할 수 있다. 따라서 실시간 메시징을 위해 새로운 소셜 네트워크를 구성해야 하는 불편함을 해결하고 스마트폰을 이용한 이동 환경에서 트위터의 팔로워와 그룹 채팅을 수행할 수 있다.

VI. 결론

본 논문에서는 대표적인 소셜 네트워킹 서비스인 트위터의 소셜 네트워크를 기반으로 실시간 그룹 채팅을 지원하는 기법을 제안하고 이를 활용하여 스마트폰 그룹 메시징 시스템을 개발하였다.

트위터의 소셜 네트워크를 기반으로 사용자와 팔로워들이 참여하는 그룹을 자동적으로 형성하고 그룹 구성원이 모두 참여할 수 있는 실시간 그룹 메시징을 지원하는 기법을 제안하였다. 그리고 이를 바탕으로 개발된 스마트폰 그룹 메시징 시스템은 XMPP 프로토콜 기반의 Openfire 메시징 서버와 트위터 사용자들 간 실시간 메시징을 수행할 수 있는 스마트폰 클라이언트로 구성된다.

Openfire 메시징 서버는 클라이언트와 비동기적으로 연결되며, 클라이언트의 요청에 따라 사용자들과 연결된 팔로워들로 정의된 트위터의 소셜 네트워크 정보를 사용자와 그룹 정보로 변환하여 연결된 데이터베이스에 저장한다. 스마트폰 그룹 메시징 클라이언트는 트위터와 연결하기 위한 트위터 모듈과 메시징 서버와 연결하기 위한 메시징 모듈로 구성되며, 안드로이드 사용자 인터페이스를 통하여 실시간 그룹 채팅을 위한 다양한 기능을 제공한다. 트위터 모듈은 twitter4j 라이브러리를 이용하여 트위터의 소셜 네트워크 정보를 요청하거나 트위터 메시지를 전송하는 등의 기능을 처리한다. 메시징 모듈은 XMPP 통신을 지원하는 aSmack 라이브러리를 이용하여 메시

징 서버와 통신하며 다른 클라이언트와 실시간으로 메시지 정보를 주고받는다.

트위터 사용자들은 개발된 메시징 시스템을 통하여 트위터의 기본적인 서비스를 이용하면서, 동시에 스마트폰 기반으로 이동 환경 중에도 자신의 여러 팔로워들과 편리하게 실시간으로 메시지를 주고 받을 수 있다.

향후 메시징 서버와 파일 서버를 연동하여 트위터 팔로워 구성원들 간의 파일 전송 및 공유가 가능하도록 시스템을 확장할 예정이다.

참고 문헌

- [1] danah m. boyd and Nicole B. Ellison, "Social Network Sites: Definition, History, and Scholarship," *Journal of Computer-Mediated Communication*, 13(1), article 11, 2007.
- [2] Twitter, "twitter.com"
- [3] Bernard J. Jansen¹, Mimi Zhang¹, Kate Sobel², Abdur Chowdury³, "Twitter power: Tweets as electronic word of mouth," *Journal of the merican Society for Information Science and Technology* Volume 60, Issue 11, pp.2169 - 2188, November 2009.
- [4] Twitpic, "www.twitpic.com"
- [5] Twitcam, "twitcam.livestream.com"
- [6] Twitpoll, "twitpoll.net"
- [7] Tynychat, "tynychat.com"
- [8] Lee Hong-Chang Park, Jin-Ho Kim, Seong-Hune Lee, Myung-Joon, "Jabber Messenger for a WebDAV-based Collaborative System," *The KIPS transactions. Part C*, v.14C no.6, pp.509~518, October 2007.
- [9] Peter Saint-Andre, "Streaming XML with Jabber/XMPP," *IEEE Internet Computing*, vol. 9, no. 5, pp.82-89, September 2005.
- [10] Openfire, "www.igniterealtime.org/projects/openfire/index.jsp"
- [11] aSmack, "code.google.com/p/asmack"
- [12] facebook, "pdfserve.informaworld.com/521666_76651179.pdf"
- [13] Yamamoto, Y., "Twitter4J - An open-sourced, mavenized and Google App Engine safe Java library for the Twitter API, released under the BSD license," "yusuke.homeip.net/twitter4j/en/index.html", 2009.
- [14] java-twitter, "code.google.com/p/java-twitter"
- [15] Jtwitter, "www.winterwell.com/software/jtwitter.php"
- [16] SavorChat, "www.savorchat.com"

[17] Google Talk, "en.wikipedia.org/wiki/Google_talk"
 [18] MobileMe, "en.wikipedia.org/wiki/MobileMe"
 [19] XMPP Extensions, "xmpp.org/xmpp-protocols/xmpp-extensions".
 [20] P. Saint-Andre, ed., "Extensible Messaging and Presence Protocol (XMPP): Core," RFC 3920, October 2004.
 [21] P. Saint-Andre, ed., "Extensible Messaging and Presence Protocol (XMPP): Instant Messaging and Presence," RFC 3921, October 2004.



이 명 준
 1980 : 서울대학교수학과 공학사.
 1982 : 한국과학기술원 전산학과 이학 석사.
 1991 : 한국과학기술원 전산학과 공학 박사
 1993~1994 : 미국 버지니아대학 교환 교수
 2005~2006 : 미국 캘리포니아주립대학 교환 교수
 1982~현재 : 울산대학교 전기공학부 교수
 관심분야 : 웹기반 정보시스템, 프로그래밍언어, 분산 프로그래밍 시스템, 생물정보학,
 Email : mjlee@ulsan.ac.kr

저 자 소개



박 종 은
 2011 : 울산대학교 컴퓨터정보통신공학부 공학사.
 현 재 : 울산대학교 컴퓨터정보통신공학부 석사과정
 관심분야 : 클라우드 컴퓨터, 모바일 프로그래밍, 분산 프로그래밍
 Email : cjsowwhddms@nate.com



권 오 진
 2011 : 울산대학교 컴퓨터정보통신공학부 공학사.
 현 재 : 울산대학교 컴퓨터정보통신공학부 석사과정
 관심분야 : 클라우드 컴퓨터, 모바일 프로그래밍, 웹기반 정보 시스템
 Email : ojini5@nate.com



이 흥 창
 2006 : 울산대학교 컴퓨터정보통신공학부 공학사.
 2008 : 울산대학교 컴퓨터정보통신공학부 공학석사.
 현 재 : 울산대학교 박사 수료
 관심분야 : 웹기반 정보시스템, 클라우드 컴퓨터, 모바일 프로그래밍
 Email : myhyunii@ulsan.ac.kr

