IMS Social Network Application with J2ME compatible Push-To-Talk Service

Christian Menkens, Nils Kjellin, Anders Davoust
Georgia Institute of Technology
College of Computing
Atlanta, Georgia 30332–0250, USA
christian@gatech.edu, {nkjellin3,adavoust3}@mail.gatech.edu

Abstract

People in any type of community use different kinds of systems, social network services and technologies to interact, communicate, inform themselves and others about news, events etc. within their community. So far it was necessary to use different systems and technologies like bulletin boards, cell phones, pagers, instant messaging, E-Mail, social network services etc. to do that. This paper describes an IMS social network application that can be used to do all these tasks using IMS enabled cell phones. A typical "community" for the application would be a college campus, metropolitan business or downtown area or large scale events with multiple locations such as the Olympic Games or World Cups. Typical users would be students, professors, businessmen, travelers, spectators etc. The main interface of the application is an interactive map of the community/area that enables users to interacting with their whole community.

1 Introduction

1.1 IP Multimedia Subsystem

In the move towards a converged network architecture the IP Multimedia Subsystem (IMS) [1] standard defines a generic Next Generation Networking (NGN) architecture. It is a set of specifications for offering mobile and fixed Voice over IP (VoIP) and multimedia services. The IMS standard was introduced by the Third Generation Partnership Project (3GPP) as a part of their standard in Release 5. The standard supports multiple access types, including GSM, GPRS, EDGE, WCDMA, UMTS, Wireline broadband access and WLAN. IMS truly merges the Internet with the cellular world; it uses cellular technologies to provide ubiquitous access and Internet technologies to provide appealing services. It enables the integration of different enabling services, such as presence [6], location [5], messaging [7], video, voice, picture and text, into one application and supports charging and Quality of Service (QoS). IMS shortens application development time and supports deployment and re-use of applications and services. IMS uses the Internet Protocol (IP) [2] as its underlying network protocol and the Session Initiation Protocol (SIP) [3] as signaling protocol.

1.2 Social Network Service

A social network service is social software specifically focused on the building and verifying of online social networks for whatever purpose.It provides a way for people, with all types of interests, to connect with each other online and to interact in numerous kinds of ways. Generally these services require users to create a profile with information such as education, interests, hobbies and relationship status. Users can upload pictures of themselves, be "friends" with other users and share interests. It becomes a way of telling people who you are; creating an image of yourself. Some major social networks (e.g. Myspace, Facebook) have additional features, such as the ability to create groups that share common interests or affiliations, upload videos, and hold discussions in forums.

2 Design and Implementation of IMS Social Network Application

This sections describes the design, architecture and implementation of our social network application. We present our application features, show how our application uses IMS as underlying network architecture and how we leverage many different IMS services. Our IMS research lab at Georgia Tech contains all components of a large scale commercial IMS deployment and so we were able to test our implemented application extensively as well as run it in a real world environment with many users.

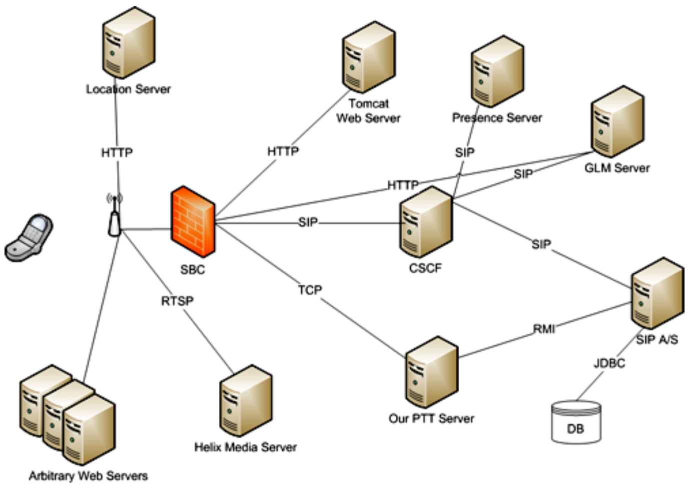

Figure 1. IMS Social Network Application Architecture

2.1 System Architecture

Figure 1 shows an overview of the main parts of the IMS system in our IMS research lab. All parts shown are used by our social network application and fulfill the following features or tasks,

- **Cell Phone Clients** - Our phone client is developed for Nokia Series60 3rd edition mobile phones and is based on J2ME. Nokia N80 phones were used for testing and the Nokia Series 60 3rd Edition emulator was used for development and debugging. Since the application complies with J2ME MIDP2 and the MMAPI 1.2 it can be run on every J2ME phone that fulfills these requirements. For development and testing we used a standard 802.11g WLAN access point to get access to the IMS system network and we tested all our features Georgia Tech campus wide with WLAN as access network.

- **Session Border Control (SBC) and Core IMS** - Our IMS research lab is equipped with an IMS compliant Session Border Controller (SBC) that protects the IMS components and servers. Since our system communicates session based between phone applications, the SIP application server and Push-To-Talk server, it needs to interact with the SBC at application start-up. Dynamic port assignments on both sides need to be requested and stored for later use.

  A Siemens @vantage CSCF server is the core server of our IMS research lab system, it comes with an integrated HSS and handles all registrations, session setups and other SIP messages. Triggers on the CSCF advise the server to forward certain messages to our social network application on the SIP application server.

  The Ubiquity SIP application server (SIP A/S) hosts the server (SIP Servlet) part of our social network application. This part processes all forwarded SIP messages, parses SDP [4] and XML payload data, communicates with our Push-To-Talk Server over RMI and sends responses or signaling information to the phone applications. In addition to that it manages all sessions of invited users, IP addresses, dynamic port assignments with the SBC and uses a XML parsing and generating component to work with our MySQL database. This XML component generates and parses all XML payload data used in our application and manages a JDBC connection to the database to read and write the processing data.

- **Location and Presence** - Location and presence information and group list management are essential for our social network application.

  For location we use a WLAN access point/HTTP based location service offered campus wide by Georgia Tech (Not yet compliant to [5]). We incorporated this service into the IMS system and our social network application by defining own signaling messages and payloads.

  For presence and group list management our application uses the fully standard compliant IMS research lab Siemens MPM Server [6].

- **Push-To-Talk (PTT) Server** - This is a proprietary and very simple PTT server developed as part of our social network application in order to have a push-to-talk feature in our application. The reasons why we could not use the Push-over-Cellular (PoC) server that is already part of the IMS system are mainly hardware and software limitations on the J2ME phone. (See Section 2.3 for more information).

- **Tomcat and Helix** - The Tomcat J2EE server hosts a JSP/HTML web based system and user profile administration web application. In addition to that it is used to host user album pictures and various other media items for the application and users.

  The Helix server hosts all streaming media items such as video clips and trailers published by community Hotspots, videos published by users, recorded lectures for campus communities etc.

- **Internet services** - Since IMS is an ”all-IP” network and system architecture our application can query media items, pictures and photos from every Internet service and web server that is accessible and available. This offers many possibilities and our application could be integrated with existing services and systems such as for example Facebook, Flickr etc. During testing and development we used several media resources from our Facebook, Google and Flickr accounts.

2.2 Application Features

This section presents an overview of all features our application has in it's current version. Since usability was a major factor in our application and system design we strove to make all features easily accessible even with the limited user interface possibilities that are available on a mobile phone.

In our development we focused on a campus community, so some of the features are specifically for that. All other features enable our application to be a full social network application with additional extended communication and information system features.

- **Map, Buddies and Hotspots** - One of our major goals at designing and developing this social network application was excellent usability and easy handling even though user input capabilities are very limited on mobile phones. Our main user interface is a map of the entire community. The left screen in Figure 2 shows our applications map interface. The map shows roads and landmarks of the community and visualizes two types of objects: Buddies and Hotspots. A user can browse through buddies and hotspots on the screen and select and interact with them. The right screen in Figure 2 shows a buddy profile after a user selected the buddy on the map.

  Buddies are other community members a user added to his/her own buddy list. All buddies are rendered on the map and the location at which a buddy icon is rendered and what type of icon is used to visualize the buddy is tightly tied to location and presence. (See next item for more information)

  Hotspots are points of interests within the community. Basically there are no limitations to what a hotspot can be and in our implemented campus social community network application types of hotspots range from restaurants, over landmarks and student centers to lecture halls and research labs.

- **Extended Presence and Location Information** - Location information, presence information and group list management are essential for our social network application and bring a lot of additional value to our solution that other social network applications don't have.

  The Georgia Tech location service offers an HTTP interface to query the device's location that sends the HTTP query (GPS coordinates) based on its IP address and WLAN access point assignment. The phone application requests and publishes this information to our application server which notifies the notifying users buddies about the location update.

  Since our application is about social networking, communicating and interacting with your buddies and people in your community, presence status and the location information can give you a lot of information by just looking at your buddy's icon and profile information on the phone screen. We defined eight different presence/mood states for users which are visualized with different icons on the map interface. All this information is published through the IMS presence service and is combined with location information within our application. SoCoNet enables a user to generate groups on the IMS group list server, add buddies to these groups and change the group to visualize at runtime.

  Location and presence information are both privacy sensitive data, so the user needs full control about publishing or hiding this information. In our application presence information querying is only first party, so the user has full control about location data and can always choose to not publish it at any point. The same applies to presence information. A user can choose to be "not available" at any time and hide all privacy sensitive data from other users.

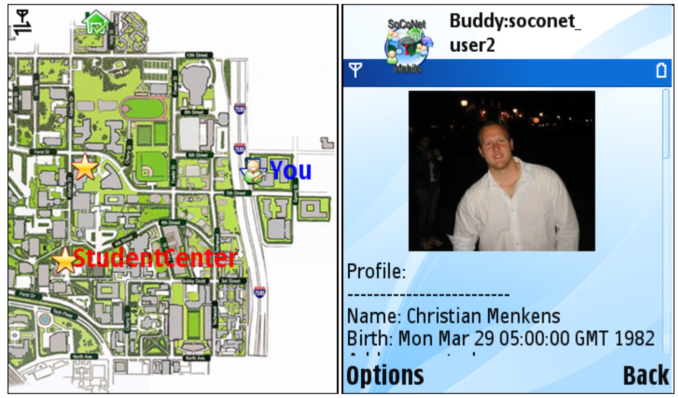

Figure 2. IMS Social Network Application Phone Screens

- **Profiles** - A Profile contains data about users and hotspots stored in our application database. After selecting a buddy or hotspot on the map interface the profile data is loaded and presented. Figure 2 on the right shows a buddy profile screen and figure 3 shows the call flow that is executed when loading profile data.

  This call flow is the standard call flow for loading

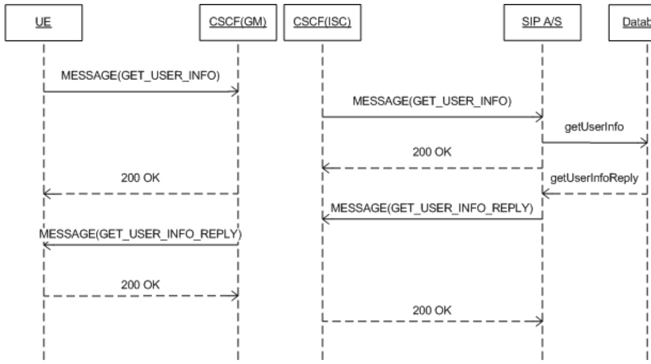

Figure 3. Standard Query for reading Data from or writing Data in the DB - Example: Reading User Profile Data

any data from our systems database and it is used with different request - response XML documents (GET_USER_INFO and GET_USER_INFO_REPLY XML documents in this case) to load and store things like media album information, profile information, lecture information and more.

In our current version a user profile contains basic information such as full name, date of birth, interests, field of study, contact information etc. In addition to that location and presence information are part of the profile and every user has a personal guest book called ”wall” that is also presented on the profile screen.

A hotspot profile contains basic information about the hotspot such as what type of hotspot, address, contact information etc. In addition to that location, offers, news and announcements are part of the profile.

- **Communication** - Since a mobile phone is still mainly a communication device our social network application focuses a lot on communication and interaction features.

  – Push-To-Talk - unidirectional audio messaging feature to send recorded audio to one or multiple buddies. All received audio messages are stored in a local inbox.

  – Text Message / Chat - Can be an unidirectional text message sent to one or multiple buddies or a bidirectional text communication including history [7].

  – Picture Message - Unidirectional picture transfer of captured picture sent to one or multiple buddies. The received picture messages are stored in a local inbox.

  Figure 4 shows the call flow that implements picture messaging. The sender informs the receiver about it’s attempt of sending picture data. The receiver informs the sender about it’s current IP address and the sender opens a TCP connection to send the picture data to the receivers IP address. If a picture is sent to multiple buddies, this sequence happens for every buddy individually.

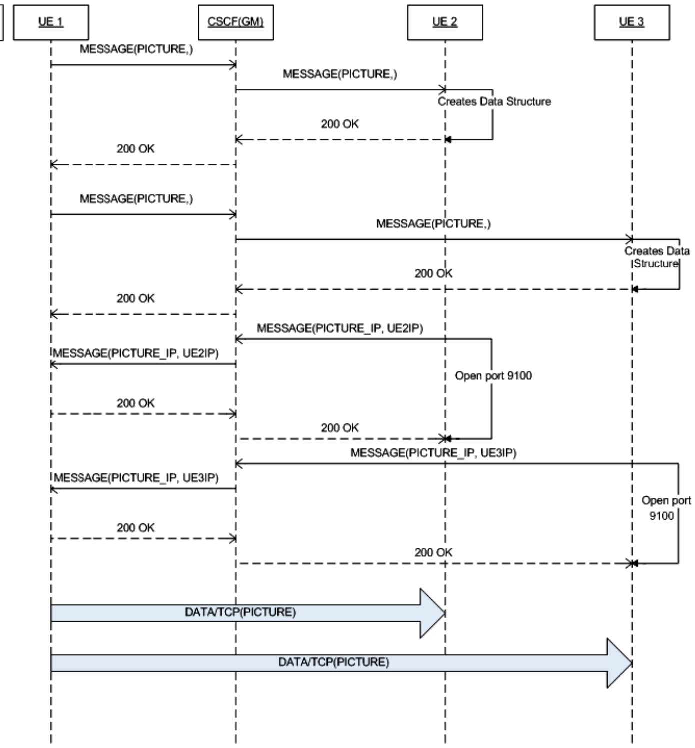

Figure 4. Picture Messaging Call Flow (Example: 2 Recipients)

  – Voice / Video Call - Voice leverages the GSM capabilities of the mobile phone and starts a circuit switched GSM phone call. Video is a bidirectional audio/video streaming between two phones. Currently not implemented due to J2ME limitations.

  – Wall - Public text message storage that is part of a user’s profile. Every user can leave messages on his buddies’ walls and all wall messages are sorted and presented in chronological order on the profile screen.

  – Blog - Every user has a public blog he/she can post messages on. All buddies can read and comment these posts.

- **Multimedia** - Every user, hotspot and lecture has a media album that can contain links to video clips, live video, pictures and audio clips. Our application contains a media player to render pictures, play audio and video clips. It uses the Real Player available on the Nokia N80 phones to play live video and large bit-rate video clips.

- **Lectures** - A user can access all lectures he/she is currently registered for and access publish lecture material such as PDF documents, text announcements, media items and recorded or live lectures.

- **Events, Ticketing and News** - Certain hotspots may offer events and shows and these hotspots can allow users to purchase tickets for these events right away when accessing the profile. In addition to that hotspots can publish news and announcements with their profile.

- **Subscriptions and Notifications** - Users can subscribe to hotspot news, event information, announcements as well as buddy blogs and walls to get automatic notifications if new information is available.

2.3 J2ME compatible Push-To-Talk

This section presents the Push-To-Talk (PTT) enabling service that we implemented for our social network application and a problem description about why we could not use the already existing Siemens PoC Server that is available in our IMS research lab. Since we designed our PTT service as a module within our application architecture it can be extracted and added to any other J2ME based IMS application easily. Our design defines a very simple PTT service and focuses only on audio data transmission. Features like floor handling and PTT groups are not implemented yet.

2.3.1 J2ME MMAPI Limitations

A major limitation for any real-time audio and/or video application implemented in J2ME with the MMAPI is that it does not allow to use recorded bytes recorded from the phone's microphone or camera while still recording. The application needs to stop the recorder in order to access the captured bytes from the capture device, after that the application could encode and send these bytes. This means it would not be possible to generate a audio/video stream from the phone to any other network device such as the PoC server. One would need to start/stop the recorder for example every 500ms or so to send small chunks of recorded data. This would then lead to breaks in the recorded audio due to the fact that it takes a few ms to stop and start the recorder in J2ME.

Another major limitation for any streaming application implemented in J2ME with the MMAPI is that it does not

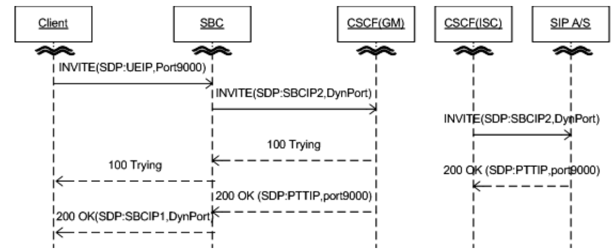

Figure 5. SBC Dynamic Port Assignments

support encoding/decoding of RTP or the RTCP protocol. Since the PoC server requires RTP and RTCP and even puts it's floor handling protocol on top of RTCP it is not possible to exchange real-time audio data with the PoC server, unless somebody would implement full duplex RTP and RTCP for J2ME. On the other hand, this would probably need to much processing power and the phones would not be able to handle that in Java.

2.3.2 PTT Service Architecture

Figure 1 already showed all parts of our IMS system architecture that our PTT service touches. The Ubiquity SIP A/S and our proprietary PTT server build the central part of our service. In addition to that our service needs to interact with the SBC at session setup in order to request dynamic port assignments for later TCP data transfers on both sides of the SBC.

Figure 5 shows the typical session setup at application start-up time. The INVITE message's SDP [4] payload includes the phones IP address and the standard TCP client port for our PTT service - 9000. The SBC changes the IP address and port to it's own internal IP address and a random port. Our SIP A/S application needs to store this IP and port assignment for the user that sent the INVITE and returns the IP address of the PTT server and the standard port as 200 OK SDP [4] payload to the phone application. Again, the SBC changes the IP address to it's external address and a random port and the phone application needs to store that for later use. These IPs and ports can later be used for a TCP audio data transfer between the phone and the PTT server. At that point the SBC will only act as TCP proxy.

For signaling within our PTT service we use SIP MESSAGE messages with specifically defined plain text payloads that our applications on the SIP A/S as well as on the phones generate and parse. Figure 6 shows a PTT service call flow and audio data transfer using the previously requested dynamic port assignments at the SBC. When a SIP MESSAGE message with a command to start a PTT session arrives the SIP A/S application contacts the PTT Server over RMI and informs the server from what user with

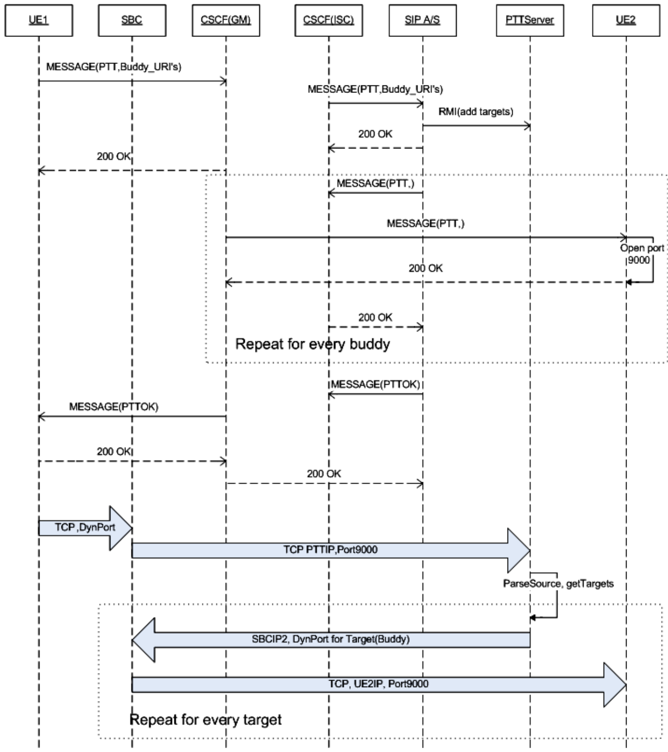

Figure 6. Push-To-Talk Call Flow

what IP and from what port audio data will be received and to whom the audio should be sent. The recipients are defined by there SIP URIs and the PTT server can handle any number of recipients which are defined as a list of buddy URIs, IPs and ports. The SIP A/S application informs all participating phone applications about the PTT setup and the sending phone application sends audio data to the PTT Server. This server uses the information previously received to send the audio data to all targets that were defined for the incoming audio data.

3 Conclusion and Future Work

This paper presents the design, architecture and some important implementation details of an IMS based social community network application. Typical standard features for social community network applications were determined, these were combined with additional features that the use of IMS as underlying technology enabled. This lead to a design and implementation of a social community network application that goes beyond every social community network available and that offers feature no other social community application offers.

In the process of designing and developing the application, software and hardware limitations were encountered and due to them a proprietary fully functional J2ME compatible audio messaging / Push-To-Talk enabling service prototype was developed and integrated into the application.

In our future work we will consider the integration of RTP and RTCP into our J2ME application in order to interact with the PoC server in our IMS lab. In addition to that, we will focus on more types of communities, implement specific features for these communities and extend the application to being able to support different configurations for different communities at the same time. Furthermore we will focus on current research work in the area of context-sensitive recommender systems and will add filters, rules and a recommender system that is tightly tied to location and presence information in order to support and inform users based on their location, profile, interests, presence status etc.

Acknowledgment

The Georgia Institute of Technology IMS research lab is sponsored by Georgia Tech, Nokia-Siemens Networks and AT&T. The authors would like to thank the sponsors and their colleagues Russ Clark, Matt Sanders, Frank Park, Nirmal Thakker, Guilherme Giacon and Alan Bryant very much for their support at configuring and administrating the lab servers and resources as well as for their help at solving various problems with the new IMS lab system.

References

[1] 3GPP Organizational Partners, 3rd Generation Partnership Project; IP Multimedia Subsystem (IMS) (Release 7), http://www.3gpp.org/ftp/Specs/html-info/23228.htm, 2006.

[2] University of Southern California - Information Sciences Institute, RFC791 - Internet Protocol, http://www.faqs.org/rfcs/rfc791.html, September 1981.

[3] J. Rosenberg et. al., RFC3261 - SIP: Session Initiation Protocol, http://www.ietf.org/rfc/rfc3261.txt, June 2002.

[4] V. Jacobson, RFC2327 - SDP: Session Description Protocol, http://www.ietf.org/rfc/rfc2327.txt, April 1998.

[5] 3GPP Organizational Partners, 3rd Generation Partnership Project; Location Services (LCS) (Release 7), 3GPP TS 22.071 V7.4.0, 2005.

[6] J. Rosenberg, RFC3856 - A Presence Event Package for the Session Initiation Protocol (SIP), http://www.ietf.org/rfc/rfc3856.txt, August 2004.

[7] B. Campbell et. al., RFC3428 - Session Initiation Protocol (SIP) Extension for Instant Messaging, http://www.ietf.org/rfc/rfc3428.txt, December 2002.