BBS-ONE: Bulletin Board and Forum System for Mobile Opportunistic Networks

Kiwoon Sung, Suman Srinivasan, and Henning Schulzrinne

Abstract**—**Electronic bulletin boards and forum systems are commonly used to exchange opinions, news, event notifications, documents and other media on the Internet. However, such systems usually require a central server hosting the content. Such servers cannot be installed in ad-hoc opportunistic wireless networks, which are created when mobile devices congregate to form a localized and short-lived network without Internet connectivity.

We present BBS-ONE, a bulletin board system for opportunistic networks, and describe its service model and implementation. BBS-ONE works in highly mobile opportunistic networks, considers the mobility of nodes, and allows nodes to operate even when churn is high when nodes join and leave the network. It transparently disseminates public data and posts and persists desired data by operating in a peer-to-peer fashion and using a store-carry-forward model of communication. It maintains the data consistency needed for a BBS and forum system. We have implemented the application on generic desktop OS platforms (Windows, Linux, Mac) as well as a mobile platform (iPhone/iPod).

Keywords**—**BBS, forum, opportunistic network, mobile ad hoc network, opportunistic network, ipod, iphone, implementation

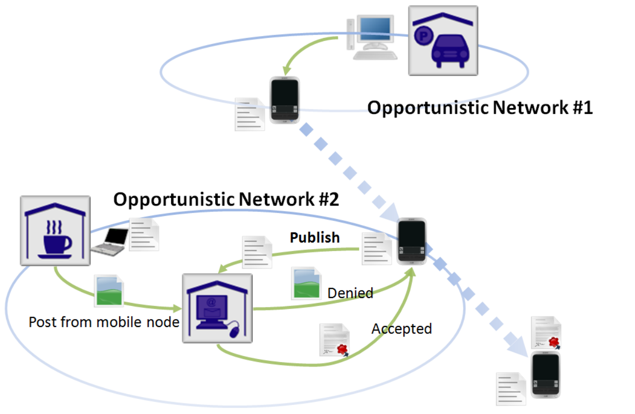


Fig. 1. **Usage Scenario for the BBS-ONE. A mobile node moves from one isolated opportunistic network to another one, carrying desired information to a location where no connection to the infrastructure. Dotted lines indicate the direction of the movement of the node in and out of an area with wireless ad-hoc connectivity.**

I. INTRODUCTION

Bulletin board systems are an important tool for collaboration and information exchange among peers. The paper-based bulletin board systems on college campuses, apartment complexes and other social areas provide a way for students, neighbors and peers to interact with each other and allow others to be mutually aware of events going on as well as facts that may interest others.

In recent years, with the rise of the Internet, the BBS has come to refer to a central, online repository or forum where users can post messages and files to exchange with other members of the board. The online forums or BBSes are often associated with a specific subject, topic or neighborhood, and provide similar online community features for its members.

However, with the growing use of opportunistic networks, where mobile nodes join together to form network islands with ad-hoc wireless connectivity, the traditional BBS model begins to fail. In the previously described scenarios, the BBS model only works because of the presence of a central server that handles the forum and community information. In opportunistic networks, there is no such central server, and to add to the problem, there is high mobility and churn rate, with nodes leaving and joining the network frequently.

Kiwoon Sung is with Columbia University, New York, USA, email: kiwoon.sung@gmail.com

Suman Srinivasan is with Columbia University, New York, USA, email: sumans@cs.columbia.edu

Henning Schulzrinne is with Columbia University, New York, USA, email: hgs@cs.columbia.edu

In this paper, we explore how we successfully built a real-world BBS application for opportunistic networks, called BBS for Opportunistic Networks (BBS-ONE). BBS-ONE allows users of mobile devices, such as iPhones, to share and distribute content with their local community connected by an opportunistic network, such as an 802.11 ad-hoc network.

By operating in a peer-to-peer manner, BBS-ONE provides BBS and forum functionality without requiring a central server. Our service model also enables us to keep data persistent, even when mobile nodes move across networks, thus fulfilling the basic requirement of a forum which allows for information to be spread in the local community.

We present our design of the system model of BBS-ONE to handle opportunistic networks and the state transitions that occur when nodes move in and out of the network.

In Section II, we introduce the concept of BBSes and forum software. In Section III, we describe the service model on which this system is based, in order to keep, disseminate and carry data for BBS. In Section IV, we show how the system was designed and implemented, with details of the specific technologies involved. Screenshots of the BBS-ONE application on these different platforms are also included.

In Section V, on related work, we compare this system to various services that provide information sharing, online forums, traditional BBSes, and synchronization applications for portable devices such as PDA and mobile phones. Section VI and VII provide our thoughts on future work and conclusion.

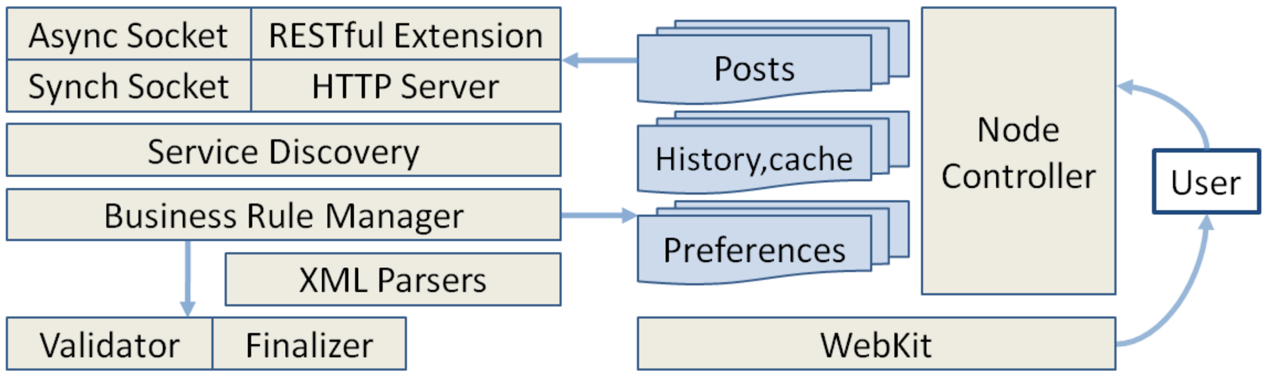


Fig. 2. **System Architecture of the BBS-ONE system. The diagram shows the networking, service discovery and rule validators, the post and node managers and how the components interact.**

II. FORUMS AND BBS SOFTWARE

Forums and bulletin boards are a central place on college campuses and other settings where students and peers can post information about events and general information that might be interest to their peers. These often form the central "water cooler" in campus environments where students can share information about events that happen throughout campus.

Newsgroups (list servers) and web-based forums became quickly popular on the Internet, providing a forum for online communities to form and discuss topics and events, based on either interests or locality.

Generic BBS systems utilize a centralized system hosted on one or more physical servers which is controlled by system administrators and forum administrators. However, often, users are allowed to generate, post and exchange content to their heart's content, unless such content violates the terms set by the forum administrators.

BBSes can be expanded one step further to become social networking sites used to build online communities. The most prominent use of these online communities recently has been seen during political events, such as DeanSpace [4] and Facebook Statuses for Obama [5].

A. BBS IN OPPORTUNISTIC NETWORKS

With the rising use of mobile devices, opportunistic networks are more commonplace. For example, in highly dense cosmopolitan areas and underground subways, it is hard to get persistent high-bandwidth connections, such as 3G [14] connectivity, but opportunistic networks using wireless ad-hoc technologies can be formed easily.

Take an example of two or more users on a subway or a remote train station with very poor 3G reception. The users want to share data with each other, but are not able to because of the lack of connectivity. However, the users are able to connect with each other through a wireless 802.11 [1] ad-hoc connection. Unfortunately, since most forum software relies on the use of the Internet or a central server architecture, without software written to work specifically in this situation, the users will not be able to share information with each other.

BBS-ONE enables us to share information in such opportunistic networks. Unlike the traditional BBS, our system does not require a central server machine to store all data from users.

Our implementation needs to work on devices running on opportunistic networks with high mobility and churn rates and in the absence of a central server. Hence, we need a unique way of solving the problem of running a BBS or forum. We need to solve the issue of sharing data in a disconnected network. The nodes need to be aware of such mobility, and be able to recover gracefully in case of any abrupt cessation of communication.

III. SERVICE MODEL

The fundamental requirements of bulletin board system are providing a virtual space where people can interact with others by sharing information of interests. BBS-ONE fulfills this requirement. Throughout the system, data management and networking schemes are used to provide bulletin board and forum features.

Figure 1 shows one possible scenario in which a mobile user whose device is the part of the system enters and leaves opportunistic networks enroute to a location where no wireless service is available. BBS-ONE works in both networked area (where the node is connected to an infrastructure such as the Internet) as well as a disconnected area (where the only connections are local connections). These scenarios show an example of the store-carry-forward scheme scenario, and the system can be utilized for the different use cases. User A stores information acquired based on a user's preferences consisting of keywords.

A. DATA MANAGEMENT

1) POST HANDLING: A post is a user submitted message in traditional BBSes, and we define it the same way for BBS-ONE. A post is the conceptual elementary entity of information that users want to share or to acquire through this system. With a post, user can describe their idea and information as text with attached resources such as images, music, and possibly other types of files. We regard a post as the elementary form of information in this system, in that all operations, such as creating and sharing a post, are applied to posts based on privileges which can be determined by

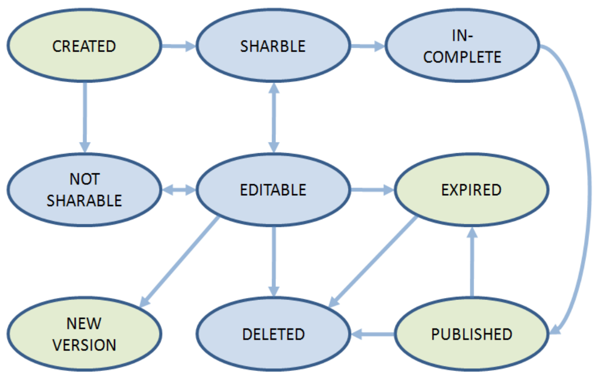


Fig. 3. **Life Cycle of a Post in the BBS-ONE system.**

examining the author of a certain post and other permissions set on that particular post.

It also has a life cycle that a post must follow. According to the current state of a post in the life cycle, BBS-ONE determines if a specific post can be shared, edited, or limited in terms of dissemination.

Figure 3 shows the life cycle of a post. The vertices show the state of a post and the directional edge of the vertices connect to the next possible state. A post is always in one of the states described here.

Each post has a field to keep information about the nodes which the post traversed and other trace information. It is possible for several nodes to acquire the same post containing the same content at a certain time. However, it is possible for several copies of an identical post to have different trace information in it, because it is quite common for a post to pass traverse different routes.

B. Deployment

Our BBS-ONE can operate in two modes: a client-server mode in the presence of an stationary node, and a true P2P mode in the absence of an stationary node.

1) Operation in presence of stationary node: The BBS-ONE stationary node refers to a node that is placed in a static location and does not move. Such a stationary node could be deployed at locations with high traffic, such as subway stations. The stationary node stores data from incoming nodes and forwards data to newcomers. When a node comes in a network area, it finds the information of a neighboring stationary node through service discovery. When it finds an stationary node, the node sends all posts to that stationary node, including metadata to avoid duplication.

2) P2P operation: In the absence of an stationary node, there is no facility for storing data, so all operations share data among the mobile nodes and they work in a totally peer-to-peer fashion. Every node needs to find connection information using multicast service discovery, and it directly connects to the peer node to exchange posts. When a node wants to acquire a post containing the desired content, it first sends search keywords to neighboring nodes found during the discovery phase, and then one of nodes that has the related posts sends an offer to it in order to get an acceptance response.

C. Networking

We try to relate the various networking operations to existing HTTP and REST [6] features. This way, not only does our application relate to existing standards, but it might be possible in the future for some sort of compatibility with existing networking and web forum applications, for example, using XML-RPC and other web service methods.

1) Pull-based: We have made the pull-based approach the default option for exchanging data on mobile devices. This has one major advantage, which is that clients can access information when they need it. However, there is also one major disadvantage, since the nodes will not be easily aware of any network connection below the application layer. Furthermore, in order to check if there are any updates on posts in nearby peers, clients will need to constantly check for updates by polling, which might reduce the battery life. Traffic will increase based on the number of existing nodes, no matter how many updates nodes have.

2) Push-based: The main advantage of push-based exchanging is that updates will arrive when available, not when needed and hence continuous polling with all other nodes is not needed. Being notified of updates is an efficient way for being made aware of when communication has to be established. However, in contrast to the previous pull-based approach, the traffic to inform a node of updates will increase as a function of the number of nodes holding updates.

Furthermore, one of characteristics that this system has, is that providing information (which is pushed to the reader) is determined according to the preference of the reader. The preferences are supposed to be multicast first, before the information is delivered.

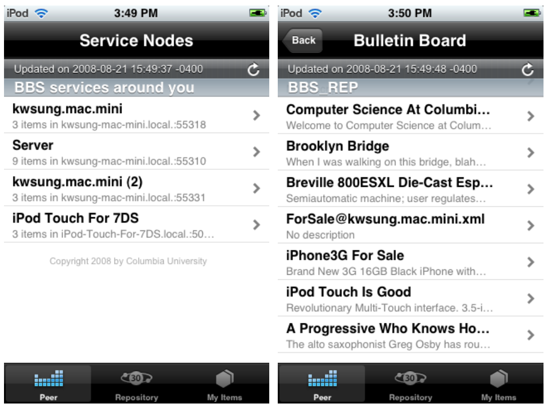
3) Publish/Subscribe: Our BBS-ONE system does not rely on the publish/subscribe messaging paradigm, in that data holder always needs to know where it should push data. Subscribers in the system also multicast their keywords of interest. Publishers get to know which nodes are interested in certain contents or data, and initiate exchange of posts by offering them data which are supposed to be pushed to the subscribers.

Also, a stationary node can act as a stationary or mobile repository. This node is able to perform a store-and forward function to deliver data from original authors to readers. Thus, even if nodes that make contact with this access point are disconnected with each other, they are still able to receive information from nodes that came earlier.

4) Epidemic Dissemination: In epidemic dissemination, a peer node finds information about how it can communicate with another node using connection information obtained during the discovery phase, and is able to initiate a connection with a specific node.

When a node has a post to be published but there no access point in the network that the node has just found, it picks up another node randomly to send posts for advertisements and public posts in order.

However, epidemic dissemination cannot be applied to every type of data in the system, since the right to choose which posts can be delivered is on the receiver, not on the data holder. Every node is restricted by how many posts it can disseminate,


iPod 3:49 PM

Service Nodes

Updated on 2008-08-21 15:49:37 -0400

BBS services around you

kwsung.mac.mini
3 items in kwsung-mac-mini.local.:55318

Server
9 items in kwsung-mac-mini.local.:55310

kwsung.mac.mini (2)
3 items in kwsung-mac-mini.local.:55331

iPod Touch For 7DS
3 items in iPod-Touch-For-7DS.local.:50...

Copyright 2008 by Columbia University

iPod 3:50 PM

Back Bulletin Board

Updated on 2008-08-21 15:49:48 -0400

BBS_REP

Computer Science At Columbi...
Welcome to Computer Science at Colum...

Brooklyn Bridge
When I was walking on this bridge, blah...

Breville 800ESXL Die-Cast Esp...
Semiautomatic machine; user regulates...

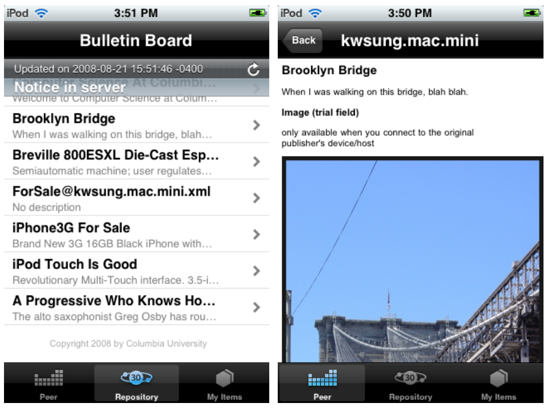
ForSale@kwsung.mac.mini.xml
No description

iPhone3G For Sale
Brand New 3G 16GB Black iPhone with...

iPod Touch Is Good
Revolutionary Multi-Touch interface. 3.5-i...

A Progressive Who Knows Ho...
The alto saxophonist Greg Osby has rou...

Fig. 4. **The first image shows the nodes that are discovered via service and node discovery. The second image shows the list of posts in the various nodes.**


iPod 3:51 PM

Bulletin Board

Updated on 2008-08-21 15:51:46 -0400

Notice in server

Welcome to Computer Science at Colum...

Brooklyn Bridge
When I was walking on this bridge, blah...

Breville 800ESXL Die-Cast Esp...
Semiautomatic machine; user regulates...

ForSale@kwsung.mac.mini.xml
No description

iPhone3G For Sale
Brand New 3G 16GB Black iPhone with...

iPod Touch Is Good
Revolutionary Multi-Touch interface. 3.5-i...

A Progressive Who Knows Ho...
The alto saxophonist Greg Osby has rou...

Copyright 2008 by Columbia University

iPod 3:50 PM

Back kwsung.mac.mini

Brooklyn Bridge

When I was walking on this bridge, blah blah.

Image (trial field)

only available when you connect to the original publisher's device/host

Fig. 5. **The first image shows the lists of posts in the repository. The second image shows the contents on a post that has been chosen by the user.**

and this is based on how much it contributes when it comes to publishing valid articles.

IV. Design and Implementation

In this section we explain our design and implementation of BBS-ONE for two platforms, Java virtual machine and iPhone, which allow the given service model described in the previous section, to work.

A. Architecture Overview

Service discovery and the 7DS [11] and BonAHA [13] frameworks are the architectural components that support the system's working in an opportunistic networking environment, while abstraction layers and the model-view-controller pattern describe how the system has been written from the perspective of actual implementation.

Figure 2 shows the overall architecture of BBS-ONE.

1) SERVICE DISCOVERY: In order to develop a framework for exchanging information about nodes entering and leaving the network, as well as the individual properties of the nodes, we looked to service discovery as an example of discovering and communicating with other nodes in the network.

Service discovery refers to protocols which enable automatic detection of devices and services on a computer network. Service discovery protocols range from lightweight protocols such as DHCP [10] to heavyweight protocols like JXTA [7].

However, service discovery protocols that can be applied to writing ad-hoc applications that run in highly mobile or disconnected networks are very few in number. We use Apple's Bonjour implementation of Zero Configuration Networking [2], which seems to be the most mature and stable protocol set. Bonjour uses multicast DNS (mDNS) messages [3] to send information on the local network and detect presence or absence of nodes. It also enables us to find out metadata or properties of the nodes using DNS TXT records.

However, Bonjour, in its native form, is not suitable for ad-hoc applications because it is primarily concerned with changes in the network, such as services entering and leaving the network. Hence, to overcome these shortcomings, we have developed our own stack of applications and frameworks on top of the Bonjour protocols to enable applications to run in opportunistic networks.

2) BonAHA framework: For developing the BBS-ONE application, we use a framework called BonAHA [13], which provides an easy and intuitive middleware for application developers to develop networking applications that run on opportunistic networks and can handle nodes entering and leaving the network.

We are using a modified version of the BonAHA framework in the BBS-ONE application. The original version of BonAHA runs on Windows, Linux and Mac OS systems, and in order to create the BBS-ONE system to run on iPod and iPhone devices, we rewrote portions of the framework in Objective-C to provide the functionality needed for writing the BBS-ONE application.

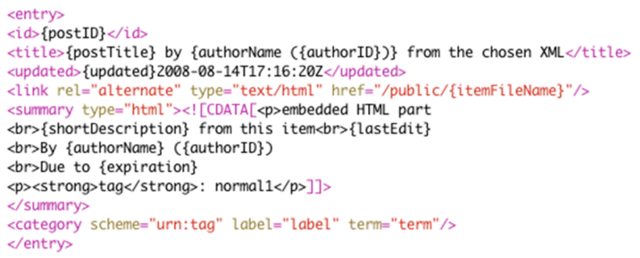
B. Implementation

To implement BBS-ONE, we developed a command-line version of the system in Java that runs on Windows, Linux and Mac OS platforms. This Java version is built using our BonAHA framework and allows a user to create, edit and share his or her posts with others.

The iPhone implementation of the BBS-ONE includes a GUI that is very similar to other iPhone applications. The implementation allows a user to see other posts on the network and be able to view them through a scroll screen based user interface. The iPhone implementation was written in Objective-C.

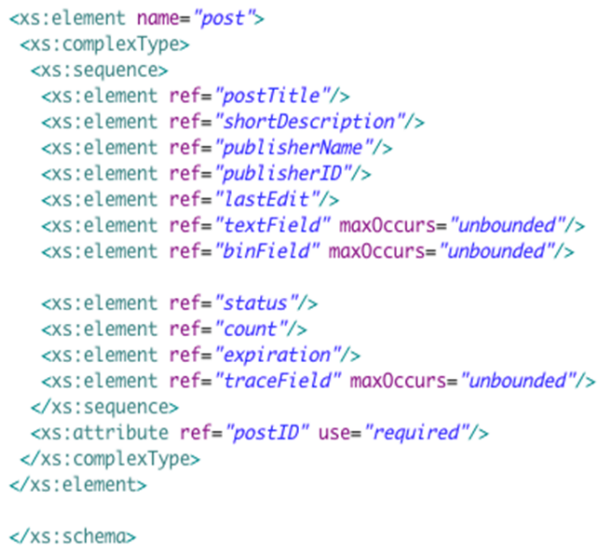
Because the iPhone platform does not currently support Java, we rewrote portions of the BonAHA library in Objective-C and integrated it into our application in order to develop the discovery mechanism of the BBS-ONE application.

The basic components of the implementation are Zero Configuration (Zeroconf) using Apple's Bonjour implementation, an internal HTTP proxy (a HTTP server) and its RESTful service extension, a cache manager as well as data service, and user interfaces, implemented both on Java virtual machine using JRE1.5 and iPhone OS 2.1 or later.



```
<entry>
<id>{postID}</id>
<title>{postTitle} by {authorName ({authorID})} from the chosen XML</title>
<updated>{updated}2008-08-14T17:16:20Z</updated>
<link rel="alternate" type="text/html" href="/public/{itemFileName}"/>
<summary type="html"><![CDATA[<p>embedded HTML part
<br>{shortDescription} from this item<br>{lastEdit}
<br>By {authorName} ({authorID})
<br>Due to {expiration}
<p><strong>tag</strong>: normal1</p>]]>
</summary>
<category scheme="urn:tag" label="label" term="term"/>
</entry>
```

Fig. 6. **The XML template for a post that can be fed to an Atom feed for syndication.**



```
<xs:element name="post">
 <xs:complexType>
  <xs:sequence>
   <xs:element ref="postTitle"/>
   <xs:element ref="shortDescription"/>
   <xs:element ref="publisherName"/>
   <xs:element ref="publisherID"/>
   <xs:element ref="lastEdit"/>
   <xs:element ref="textField" maxOccurs="unbounded"/>
   <xs:element ref="binField" maxOccurs="unbounded"/>

   <xs:element ref="status"/>
   <xs:element ref="count"/>
   <xs:element ref="expiration"/>
   <xs:element ref="traceField" maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:attribute ref="postID" use="required"/>
 </xs:complexType>
</xs:element>

</xs:schema>
```

Fig. 7. **The major fields for a post, and their types in an XML schema document.**

1) Data Format and Identification: A post is identified based on fields such as author information, the title, the description of the post and so forth. These fields contain hashed values. Some fields in a post such as trace information cannot be subjected to hashing in order to identify a post. We used MD5 as the hash function to identify a post.

We have attempted to make BBS-ONE meet the spirit of Web 2.0 and be as easy to reverse engineer so that alternative implementations compatible with BBS-ONE can be built. The default format of files presenting a post is based on XML. The schema is described by XML Schema definition language (XSD) [15] and the publication of the post information contained in each node is based on the Atom feed format [12].

Figure 6 describes the schema and XML template used to represent a post in the system. Figure 7 shows the structure of a post and the major fields, and their types.

2) Versioning and Expiration: Once a post leaves its original node, it is not easy for the author of the post to do any work on it, especially when the Post arrives on another network and it has been completely isolated from the connected environments. In many cases, disseminated data cannot be reached by any other nodes that work in physically separated networks.

Accordingly, it is more reasonable that, if a post has been pushed to another node, it is regarded as an independent data set that should identify itself. In such a case, any modification generates a new version of the original post.

3) Platforms: We wanted to implement the BBS-ONE application on a variety of platforms in order to prove its usefulness, while at the same time, meet reasonable goals and deadlines. Hence, we built the application using Java, which we have tested on Windows, Linux and Mac OS. It is also quite likely that this application runs on other operating systems and platforms that are Java compatible.

In choosing a mobile platform, we chose the iPhone/iPod platforms because it is widely deployed [16].

One major difference in the HTTP proxies on the iPhone and the command-line Java version of BBS-ONE is the socket layer underlying the application layer protocol. The Java version can use multiple threads and blocking sockets, but in the iPhone version, the implementation of the HTTP server relies on an asynchronous style or non-blocking sockets. However, this does not have any major impact on our implementation.

4) Mobile Nodes: For the iPhone OS, BBS-ONE implements only the mobile reader version, allowing a user to publish, advertise and gather post information from either stationary nodes or other mobile nodes. The GUI for this version is based on the native user interface on iPhone as well as a web browser page displaying posts.

Figure 4 shows the list of the neighbor nodes available in the network that the user has just arrived in. The names of the nodes are provided according to the auto-naming rules of the Bonjour discovery protocol stack. When a user selects one of the discovered nodes, the screen shows the list of posts that the selected node contains. Figure 5 shows the screen when a user selects a post from the list given in Figure 4. The final form of the post is the one that has been translated by XML to HTML, and the contents appear on a WebKit webpage component available on the iPhone.

V. Related Work

The original BBSes were virtual communities created when individuals or service provides allowed people in the local communities to dial-in and connect to their systems, download, and upload files and other data, as well as share information with others in the community.

USENET [9], which evolved from UUCP, allowed users to read and post messages to categories known as newsgroups. It supported threaded discussions and could be accessed by any compatible newsreader software. USENET is widely regarded as a precursor to today's web forums.

However, with the advent and popularity of the Internet, Internet-based BBS systems became more and more popular. A huge number of forums now exist around the world, spanning almost every conceivable topic. Due to the open nature of these forums, several have implemented some form of monitoring and checking.

Among forum use on disconnected, ad-hoc or opportunistic networks, there is very little work. The primary reason for this seems to be that forums are naturally oriented towards online, long-term communities.

The only available work similar to ours in this field seems to be the JXTA forum software [8]. This forum software, built on top of Sun Microsystems's JXTA protocol and framework, implements a forum software by converting an existing client-server system into use in a P2P system. However, even this system requires the use of a connected network. The authors state, "total decentralization is often neither necessary nor desirable." However, opportunistic networks are completely decentralized and mobile. Further, the system requires a naming service, which requires some long-term or super nodes, as well as setting up peer groups to avoid a flat search structure. This is not possible in opportunistic networks either.

VI. Discussion and Future Work

In our system, we have assumed that users are authenticated by providing their email address and password. In this case, it is necessary to develop a particular authentication system for BBS-ONE separately.

Exchanging personal contacts in places such as conference halls or public transportations might be one way to utilize the peer-to-peer exchanging information in an isolated opportunistic network, if permissions are considered.

We can also consider the functionality of supporting concurrent transmission of posts, and limiting the number of times a post moves from/into other nodes. Preventing spamming using identical posts and prohibited words also needs to be considered. Also, a method for handling partially transmitted posts needs to be devised.

VII. CONCLUSION

In this paper, we presented BBS-ONE, a system that enables forum and BBS functionality in opportunistic networks that are highly mobile and disconnected from the network. BBS-ONE allows users to exchange information and posts even in the absence of central servers or connection to wide-area networks.

We also presented our implementation of BBS-ONE on desktop platforms using our Java implementation, as well as on the iPhone and iPod platforms.

We believe that the BBS-ONE application provides a unique and novel implementation of BBS and forum functionality in a new networking scenario that is rapidly evolving yet has very few functional applications. BBS-ONE, as with our other applications for opportunistic networks, will provide users in disconnected and opportunistic networks to be able to share data with relative ease without needing some physical form of data transmission or file copying.

We have attempted to build BBS-ONE on as many standard technologies as possible, such as the Bonjour protocol stack, W3C standards (HTTP, XML and related technologies), and common concepts of architecture such as REST style and MVC pattern so that compatible implementations for the same or other platforms can be easily built.

VIII. ACKNOWLEDGEMENT

This work was supported by the National Science Foundation (NSF) under Grant 04-54288.

References

[1] IEEE 802.11, the working group setting the standards for wireless lans. http://www.ieee802.org/11/.

[2] Zero Configuration Networking (Zeroconf). http://www.zeroconf.org/.

[3] S. Cheshire and M. Krochmal. DNS-Based Service Discovery. *IETF draft, February*, 2004.

[4] Civicspacelabs. civicspacelabs.org. http://civicspacelabs.org/services_main.

[5] Facebook. Facebook. http://www.facebook.com/.

[6] R. T. Fielding. *Architectural Styles and the Design of Network-based Software Architectures*. PhD thesis, UNIVERSITY OF CALIFORNIA, 2000.

[7] Li Gong. JXTA: a network programming environment. *Internet Computing, IEEE*, 5(3):88–95, 2001.

[8] Emir Halepovic and Ralph Deters. Building a P2P forum system with JXTA. In *P2P '02: Proceedings of the Second International Conference on Peer-to-Peer Computing*, page 41, Washington, DC, USA, 2002. IEEE Computer Society.

[9] M. Horton and R. Adams. Standard for interchange of usenet messages. Technical report, RFC 1036, 1987.

[10] A.J. Mcauley and K. Manousakis. Self-configuring networks. In *MILCOM 2000. 21st Century Military Communications Conference Proceedings*, volume 1, pages 315–319 vol.1, 2000.

[11] A. Moghadam, S. Srinivasan, and H. Schulzrinne. 7ds-a modular platform to develop mobile disruption-tolerant applications. In *Second IEEE Conference and Exhibition on Next Generation Mobile Applications, Services, and Technologies (NGMAST 2008)*, September 2008.

[12] R. Sayre M. Nottingham. RFC 4287 - the atom syndication format. http://tools.ietf.org/html/rfc4287, December 2005.

[13] A. Moghadam S. Srinivasan and H. Schulzrinne. BonAHA: Service Discovery Framework for Mobile Ad-Hoc Applications. In *IEEE Consumer Communications & Networking Conference 2009 (CCNC'09)*. IEEE, 2009.

[14] C. Smith and D. Collins. *3G Wireless Networks*. McGraw-Hill Professional, 2001.

[15] H. S. Thompson, D. Beech, M. Maloney, and N. Mendelsohn. XML Schema Part 1: Structures. *W3C Recommendation*, 2, 2001.

[16] Los Angeles Times. iPhone becomes top handset in U.S., passing RAZR. November 2008.

Kiwoon Sung is a software engineer in Unified Communication Center at Samsung SDS. He is working on developing FMC (fixed and mobile convergence) solutions to realize mobile office environments using smartphone and mobile devices. He has a M.S. in Computer at Columbia University, and a B.S. from Korea University.

Suman Srinivasan is a fifth-year PhD candidate in the Computer Science department of Columbia University . He works under Dr. Henning Schulzrinne in the Internet Real Time Laboratory. Suman also has a M.S. from University of Florida (U.S.A.) and a B.E. from University of Madras (India).

Prof. Henning Schulzrinne , Levi Professor of Computer Science at Columbia University, received his Ph.D. from the University of Massachusetts in Amherst, Massachusetts. He was an MTS at AT&T Bell Laboratories and an associate department head at GMD-Fokus (Berlin), before joining the Computer Science and EE departments at Columbia University. He served as chair of Computer Science from 2004 to 2009.