

BonSwing: A GUI Framework for Ad-Hoc Applications Using Service Discovery

Suman Srinivasan and Henning Schulzrinne, Columbia University, New York, NY
{sumans,hgs}@cs.columbia.edu

ABSTRACT

In typical single-link 802.11 ad-hoc networks, devices often move from one network to the next, forming transitory associations without a fixed support infrastructure. The state in this network frequently changes due to node transitions. In such a scenario, traditional models for writing networking software, such as the client-server model or even the peer-to-peer model, turn out to be unsuitable for writing "ad-hoc applications".

A new framework needs to be developed for this class of applications to communicate with and be aware of transitory nodes. This framework needs to be easy and intuitive, abstract the ad-hoc networking details, and at the same time provide flexibility to the developer to develop powerful and rich ad-hoc applications.

The BonAHA (Bonjour Ad-Hoc Applications) libraries provide such a framework for writing mobile ad-hoc applications. As the first phase of the BonAHA implementation, we have implemented the GUI framework for BonAHA, titled BonSwing, which we present in this paper.

1. INTRODUCTION

For software applications to work properly in highly transitory networks, it is necessary to maintain some sort of awareness of the network connectivity and discover nodes entering and leaving the network. This requires multicast queries with multicast/unicast responses in order to keep track of nodes in the network.

Service discovery protocols in fact provide a simple framework for our requirements. Several service discovery protocols have been standardized over many years, and some of them have been robustly implemented on multiple platforms. However, using raw service discov-

ery APIs for writing ad-hoc applications is unsuitable, and that is the issue we address with BonAHA and BonSwing.

2. APPROACH

Among the service discovery platforms that we looked at, it appears that Zero Configuration Networking (zeroconf) [1], in the form of Apple's Bonjour [2] implementation, is the most suitable for our work. The Bonjour implementation of Zeroconf enables the following:

1. Choosing IP addresses: Through IPv4 Link-Local Addressing (IPv4LL)
2. Name Resolution: Through Multicast DNS (mDNS)
3. Service Discovery: mDNS/DNS-SD

3. PROBLEM

While building applications on Apple's Bonjour technology, we found that we need to build an additional framework on top of Bonjour in order to enable mobile ad-hoc applications to be built easily.

Based on our use of Bonjour for building a mobile ad-hoc framework such as 7DS [3], we find that for a truly mobile application to be completely implemented and functional, it has to implement all callback functions in the Bonjour API in order to truly work in an ad-hoc network. This is difficult unless the developer completely understands Bonjour.

There is an additional problem. Even after implementing all these listeners and functions, a mobile ad-hoc application still needs to do the following:

- Maintain an internal state table of existing services, nodes and related metadata.
- The only time a service can be resolved is when it is either found or lost. However, applications need to be able to resolve services and nodes during regular operation.

4. BONSWING GUI FRAMEWORK

In order to enable developers to easily develop mobile ad-hoc applications, a framework that encapsulates

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

CoNEXT'07, December 10-13, 2007, New York, NY, U.S.A.
Copyright 2007 ACM 978-1-59593-770-4/07/0012 ...\$5.00.



Figure 1: A multimedia sharing application. The BonSwing tree component on the left lists the nodes in the network as well the files that they are sharing, and updates automatically as nodes arrive and leave. It is instantiated with one line of code. The player on the right streams the audio or video that the user has selected from the list.

Bonjour and presents an easy API for developers is needed.

We are working on building a comprehensive framework called BonAHA to address this. As a first phase of this implementation, we have implemented the BonSwing library, which provides developers a GUI library to build applications for mobile ad-hoc networks.

For instance, the multimedia sharing application shown in Fig. 1 lists all the multimedia files shared by each node. Selecting a file enables the user of the application to play the multimedia stream from the remote node. The application's BonSwing tree component automatically updates the list of nodes and files that they offer.

By using the BonSwing components, the single line of code shown below creates the tree component that is shown in Fig. 1.

```
BTree bt = new BTree(
    "_7ds_multimediashare._udp");
```

Without using the BonSwing library, an application developer would need to understand Bonjour thoroughly and implement at least several dozen lines of code to listen to network events and node arrival and departure.

Using these components, we have developed ad-hoc software applications such as a multimedia sharing program (like iTunes), an instant messaging program (like iChat) and a drag-and-drop file sharing program. The BonSwing library and sample applications are now available for download from the SourceForge website [4].

5. RELATED WORK

JXTA [5] is a P2P framework implementation made by Sun Microsystems for the Java platform. JXTA appears to be a very powerful framework and has been im-

plemented in J2ME for mobile platforms. However, it appears that JXTA does not have the necessary framework to handle network events, such as nodes joining or leaving the network, which is very necessary for single link ad-hoc networks.

Peer2Me [6] is a implementation of a mobile ad-hoc framework using JXTA. It appears to overcome the problem of network discovery using a native network library for detecting Bluetooth network changes. However, it appears to be currently limited to J2ME devices and devices that use Bluetooth.

6. FUTURE WORK

Currently, we are working on finalizing the API for the BonAHA framework. The BonAHA libraries have been posted to SourceForge as an open-source GPL project. We will soon be soliciting feedback from other Bonjour and mobile ad-hoc application developers, and we hope to be able to speed up the finalization of the API and release a full, working implementation in a few months.

7. CONCLUSION

We believe that the BonSwing framework provides a promising start for easing development of rich GUI interfaces for mobile ad-hoc applications. We also hope that our continuing work on the BonAHA framework will enable complete ad-hoc applications to be built easily with minimal overhead for the application developer.

8. ACKNOWLEDGMENT

This work was supported by the National Science Foundation (NSF) under Grant 04-12025. This funding was for the 7DS project, which aims to develop ad-hoc applications for transitory networks.

9. REFERENCES

- [1] Zeroconf working group. [Online]. Available: <http://www.zeroconf.org/>
- [2] "Apple Computer's Bonjour." [Online]. Available: <http://developer.apple.com/networking/bonjour/>
- [3] S. Srinivasan, A. Moghadam, S. Hong, and H. G. Schulzrinne, "7DS - Node cooperation and information exchange in mostly disconnected networks," in *IEEE International Conference on Communications (ICC)*, 2007.
- [4] Bonswing download page. [Online]. Available: <http://bonswing.sf.net/>
- [5] "Project jxta." [Online]. Available: <http://jxta.org/>
- [6] A. I. Wang, T. Bjrnsgard, and K. Saxlund, "Peer2me - rapid application framework for mobile peer-to-peer applications," in *2007 International Symposium on Collaborative Technologies and Systems (CTS 2007)*, 2007.