Jerome Schneider

Zachary Jablons

BlueMesh RCOS Fall 2012 Proposal

## Brief Description of Project:

We are planning on writing a bluetooth mesh network library in Java for use on Android phones and bluetooth enabled computers. This library will be able to maintain a mesh network between a small number of devices (on the order of 10) and allow data packets to be sent among all of the devices. The motivation for this project is to add another level of functionality to bluetooth technology so that it can be used in mobile situations where the devices are less stationary than normal. It will also extend the range of bluetooth devices so that it is not required that all slave devices are connected to the same master.

## Key Design Points:

* As for routing, there are a few reasons I think this solution may be a good idea even though it is very limited. I was planning on having every node receive every packet and here are my reasons:

1. As bluetooth was not meant to support a very large number of devices, the overhead of routing packets may bog down the network more than sending every packet to every node.
2. The main uses of this library that I could think of are game related where each node would want to maintain a similar state as the others in which case it would make sense that each node should receive each packet.
3. It would be much easier to develop, and if this simple way of distributing packets proves the concept of non-conventional use of bluetooth, then it would be worth investigation larger networks with more complex routing protocols.

* I am assuming that I will have non-malicious hosts. Bluetooth is a very secure network in that it requires user authentication on both nodes that wish to communicate, so I am hoping that the user trusts the devices on the network.
* I was thinking of adding a new layer to the protocol stack. As it would be really cool to have TCP sit right on top of bluetooth, it may not be highly practical with small networks (8 to 10 devices). I was thinking that if I could prove that bluetooth could be useful for non-conventional uses then I would investigate using TCP which would definitely make sense on larger networks. Thus, if I use my own protocol (which I plan on being very simple and light weight), I would have to provide an API that users would have to be aware of.
* Voice over bluetooth does not seem as though it will be supported by the bluetooth protocol stack regarding the goals of my project. There are a few problems I encountered when reading up on bluetooth that makes this idea seem impractical. The first is that the voice over bluetooth functionality is different than data packets over bluetooth such that only 3 connections per device can be made when using voice whereas 8 connections per device can be made when using data packets. As this limitation almost definitely rules out using voice over large networks I chose to focus on simply using data packets. I would however like to explore the possibility of using bluetooth devices that do not have my project installed on them to act as a repeater. This however brings me back to the fact that both devices must authenticate the connection, this idea would require further research. I was planning on functionality being limited to devices that have my project installed, however it would be an interesting idea to explore when I get more of an idea how large bluetooth networks will behave.
* As for TCP on top of bluetooth (which I discussed briefly above), I definitely think that it would be highly useful and make it easy for others to develop using this project, however before I take on a larger task like that I would like to see that this kind of use of bluetooth can achieve speeds high enough and networks large enough to be useful. As for IP tunneling over bluetooth, this is a definite YES. I was also thinking of attempting this; however it would require a more complex routing protocol than the one I suggested above. If I can prove this type of use of bluetooth then I definitely want to attempt IP tunneling over bluetooth.
* The mesh networking protocol we will implement is Ad-hoc On-demand Distance Vector Routing (AODV). We chose this protocol since it was designed for mobile applications with temporary network architectures, allowing for robust routing in volatile mobile situations.

## Cycle Structure:

I plan on the following as my development schedule where a cycle is 3 weeks:

Cycle 1:

Finish development on connectivity. At this point the project should work in its most basic form.

Cycle 2:

Add routing functionality and create a well-defined API.

Cycle 3:

Develop a simple and a complex example app using BlueMesh with tutorials about how to use the project.

Cycle 4:

Add more network functionality and options for developers, begin experimenting with IP over BlueMesh.

Other potential applications could be a number of card games (such as poker and blackjack) or board games such as Chinese checkers or risk.