Ad-hoc Social Networking for the Google Android Platform
Project Status Report

CSC 714 Real Time Systems
David Gurecki
dwgureck@ncsu.edu

http://www4.ncsu.edu/~dwgureck/csc714/

NOTE: Since this project team consists of only 1 member, no peer evaluation is provided.

Completed Milestones

1. Investigate Wi-Fi tethering software to configure G1 phone in ad-hoc wireless mode

The *android-wifi-tether* project provides an open-source Android application for operating the G1 WiFi radio in ad-hoc mode. This open-source code was evaluated and found to be highly useful for this project. This tether application provides a GUI front-end to configure and execute a Linux script which performs a series of steps to tether the 3G data connection over the WiFi network.

- Shuts down any running WiFi services
- Reloads the WiFi system driver with a new "tiwlan.ini" config file specifying ad-hoc mode.
- Starts a DHCP server to serve clients on the adhoc network with addresses in 192.168.2.0/24 subnet
- Sets up iptables to perform a NAT function. The G1 is configured to act as the router on the 192.168.2.0 subnet. The script can optionally restrict access based on a white-list of MAC addresses using additional iptables rules.
- Registered WiFi clients append entries into a tether.log file, which can be viewed through the GUI.

2. Establish an ad-hoc network with a laptop

This step was completed successfully with the open-source tethering application. Initially, the application would not execute the WiFi ad-hoc script until the 3G data network became available. Since this project assumes no 3G data network, this code was disabled by commenting out some lines in the startTether() function. The application was then rebuilt and reinstalled on the G1 phone. The G1 phone's ad-hoc network could then be seen by 2 wireless clients (PCs) within wireless range. Each PC obtained an IP address from the G1 phone via DHCP, and was able to ping the G1 "router" IP address. The 2 PCs could be viewed in the "Connected Clients" list in the application.

3. Develop a scheme for discovering other ad-hoc devices

While the Android platform's built-in WiFi software provides a way to discover other 802.11 Access Points (via Settings → Wireless controls →WiFi settings menu), this software is not capable of discovering ad-hoc wireless endpoints. Furthermore, this WiFi mode must be disabled when configuring ad-hoc mode directly through the wireless network driver, as implemented by the tethering application. Without using the built-in Android functionality, it is unclear whether any other APIs exist for discovering 802.11 ad-hoc clients in a similar fashion. Such interfaces would need to be exposed at the driver level, which is not publicly documented and thus beyond the scope of this project. Instead, we seek to develop other mechanisms for discovery of ad-hoc devices. This project intends for each G1 device to activate ad-hoc mode using the same pre-configured service set identified (SSID), thus avoiding the need to discover SSID's in order to communicate. This leaves IP address assignment as the remaining problem to solve.

The tethering application currently uses DHCP to assign IP addresses to other ad-hoc devices on the network, since it is assumed that a particular G1 phone will always act as the router on the subnet and other ad-hoc devices will act as clients. This project assumes a slightly different operating mode, i.e. a G1 phone should be able to discover and communicate with other G1 phones without designating either to be the master a priori. The most obvious alternative is to use static IP addresses. These must be assigned uniquely for each phone by the user, possibly using a randomly-generated suggestion to aid in collision avoidance. Then, a UDP broadcast message can be sent by the application to announce a phone's presence on the subnet. A G1 can then learn of the presence of a peer on the ad-hoc network and begin communicating with it.

The UDP broadcast message has been successfully tested between the G1 phone and a PC configured in ad-hoc mode with the same SSID. The broadcast address 192.168.2.255 worked successfully (although 255.255.255.255 did not). Both devices can receive the other's announcement and gain information about IP address and identity.

Future milestones and challenges

- Test the discovery protocol between 2 G1 phones in ad-hoc mode
  - This will involve collaborating with another group and installing the same software on both phones.
- Develop software for integrating with GPS and text input functions of the G1 phone using Android SDK.
  - There are example open-source apps and the API is well documented for obtaining GPS coordinates.
- Develop software for displaying GPS location and text messages received from other ad-hoc peers on G1 phone.
  - This will likely be a simple text-based displaying location and text messages from other parties.
- Develop software for transmitting & receiving photos (time permitting).
  - This would be an interesting application for social networking, however it will only be attempted if there is sufficient time available.
- Adapt the discovery protocol to also discover other G1 phones on a WiFi network managed by an access point (time permitting).
  - The usefulness would only apply to phones in close proximity because the broadcast message only reaches to within the current subnet.