Bluetooth Scatternet Application

Sun Code for Freedom

Submitted for **Code For Freedom** Contest 2009

By

Ravi D Suvarna
Ananth V
Sandeep Jain
# Index

<table>
<thead>
<tr>
<th>Topic</th>
<th>Page No.</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Introduction</td>
<td>3</td>
</tr>
<tr>
<td>1.1 Purpose</td>
<td>3</td>
</tr>
<tr>
<td>1.2 Scope</td>
<td>3</td>
</tr>
<tr>
<td>1.3 What this application is</td>
<td>3</td>
</tr>
<tr>
<td>1.4 Terminologies used</td>
<td>3</td>
</tr>
<tr>
<td>1.5 Tools and technologies used</td>
<td>3</td>
</tr>
<tr>
<td>2. Overall Description</td>
<td>4</td>
</tr>
<tr>
<td>2.1 PC application</td>
<td>4</td>
</tr>
<tr>
<td>2.2 Mobile Scatternet Application</td>
<td>7</td>
</tr>
<tr>
<td>3. Usage Scenario</td>
<td>10</td>
</tr>
<tr>
<td>4. Testing Scatternet Application (Simulation)</td>
<td>11</td>
</tr>
<tr>
<td>5. Runtime Requirements</td>
<td>15</td>
</tr>
<tr>
<td>6. References</td>
<td>16</td>
</tr>
</tbody>
</table>
1. **Introduction**:

A Scatternet is a interconnected group of devices forming an ad-hoc network. It is more appropriate to say it consists of interconnected piconets. A piconet is a network of devices in range with each other. Two or more piconets connected to each other through member devices can be called as a Scatternet.

1.1 **Purpose:** The purpose of this application lies in establishing a scatternet between in-range and out of range Bluetooth enabled devices. Thus enabling different applications to exchange data over an extended range.

1.2 **Scope:** The Bluetooth scatternet has its advantages as any other ad-hoc network. Scatternets have the potential to bring the interconnectivity of the Internet to the physical world through wireless devices.

1.3 **What this application is:** This Scatternet application has been developed primarily, not as a texting or a product information service, but as an elegant demonstration of how a Scatternet can be formed at the application level.

1.4 **Terminologies used**:

- **Midlet**: A mobile application framework that can be implemented in java enabled mobiles.
- **MIDP**: Specification published for the use of java on embedded devices such as mobiles and PDAs.
- **CLDC**: Connected Limited Device Configuration (CLDC) defines the base set of application programming interfaces and a virtual machine for resource-constrained devices like mobile phones, pagers, and mainstream personal digital assistants.

1.5 **Tools and Technologies used**:

- Java Development Kit
- Java Mobile Edition
- MySQL
- Netbeans IDE
- Sun Wireless Toolkit
- BlueZ Utilities
- Bluecove library
- Nokia series 40 SDK
2. **Overall Description** :

To establish a scatternet, we require the constant communication between devices in order to obtain the information (Address, Friendly name) of devices in-range and devices out of their range. Once this information is available, the route can be established. Here, certain devices take the role of a server and some as clients. However, role reversal is certainly possible and required. To aid this role reversal we have a server thread running in the background always but a client thread is created only when required. For routing to occur, intermediate mobiles do act as routers.

Let us now have a brief look at our application. The scatternet application consists of two components.

- **PC Application**: Runs the scatternet application on a PC. It has a database integration used also as a product information server.
- **Mobile Application**: Runs the scatternet application on a mobile device. Acts as a text client and server.

2.1 **PC Application** :

The PC application was developed using Bluecove library. Bluecove is a Java library for Bluetooth (JSR-82 implementation) that currently interfaces with the Mac OS X, WIDCOMM, BlueSoleil and Microsoft Bluetooth stack found in Windows XP SP2 or Windows Vista and WIDCOMM and Microsoft Bluetooth stack on Windows Mobile. It was originally developed by Intel Research and currently maintained by volunteers. Since version 2.1 BlueCove distributed under the Apache Software License, Version 2.0.

Thus PC application mainly resides on native call called by bluecove and Bluetooth stack being used. It is best recommended to run on Fedora with bluecove and bluez library installed.

Below is an introduction to the PC version of the Scatternet application. The snapshot of the application is shown below in figure 1.1. It has three tabs in the main window.
1. The first tab is the Scatternet server tab. It has a start, stop, an update Product Information button and an exit button. This Scatternet application contacts a web server and retrieves information for a product and displays it by sending the information over Bluetooth to devices available through the Scatternet.

![Figure 1.1](image1.jpg) ![Figure 1.2](image2.jpg)

Figure 1.2 also shows the window that is opened when the update button is clicked. This may take several seconds as it has to contact the web server and then retrieve the data.

The window opened gives you the option of adding, deleting and editing the entries in the database. You can then save your changes. Although you might note that the role of mysql is minimal here. It is however just a demo of how it can be used to store and then retrieve data. This may also be video and audio files which are much larger. We are using a web server here because of the flexibility associated with it. This application retrieves the same data wherever it is run.

2. The second tab is the send message tab. This is incase you have to send any other message other than product information.

![Figure 1.3](image3.jpg)
Figure 1.3 shows the send message tab. It has a pop down menu for selecting the device to which the message has to be sent. It also has a text box where you can enter the message that is to be sent.

3. The third tab is the receive message tab which displays the message that has been received.

**Note**: If the message received is prefixed with ‘Search’, the product database will be searched keeping the message as an index and information of the product title matching it will be sent to the device. Thus acting as a product information application.

Figure 1.4

Figure 1.4 shows the received message tab. Any message that it receives will be displayed here.
2.2 Mobile Scatternet Application:

The mobile Scatternet application can be installed onto CLDC devices such as mobile phones. The application has been adequately tested on the sun wireless toolkit emulator. We will again before indulging in the details of its working, look at the software tools that were used to develop the application.

1. **J2ME**: J2ME(Java 2 Mobile Edition) is just another set of Java API’s written to aid in the development of mobile applications. J2ME combines a resource constrained JVM and a set of Java APIs for this. Due to the limited resources available in a mobile platform, J2ME strips down the java virtual machine so that it is compatible in a mobile.

2. **Sun Java Wireless Toolkit**: The Sun Java Wireless Toolkit (formerly known as Java 2 Platform, Micro Edition (J2ME) Wireless Toolkit) is a state-of-the-art toolbox for developing wireless applications that are based on J2ME’s Connected Limited Device Configuration (CLDC) and Mobile Information Device Profile (MIDP), and designed to run on cell phones, mainstream personal digital assistants, and other small mobile devices. This was ideal in developing our application as it was the closest in aping an actual mobile device.

Now we shall delve into the details of the mobile Scatternet application to understand its working and also on how to use the application.

To see the Scatternet in action, we’ll need at least three Bluetooth enabled devices, making sure that one device is out-of-range with another but both these devices must be in-range with one common device. A snapshot of the application running on the wireless toolkit is shown in figure 2.1.
As shown, as soon as the application is launched, it asks the user whether a Bluetooth connection is permissible with the device. Figure 2.1c shows the initial screen of the application when launched. It has four options. A “send new message”, “received message”, “rescan devices” and an “exit” option.

- **Send New Message**: This option when chosen opens up a text box in which the message can be written (Figure 2.2a). This can be sent with the send option present on the bottom right of the screen. Then the devices list is displayed to which the message can be sent. This device list contains both the in-range devices and the devices not in range. If the device selected is in-range, the message is delivered directly and if not the message is sent through a pre-determined route of the Scatternet. However it should be noted that the devices should be given sufficient time to exchange routing information and determine the ‘reachable’ devices.
• **Received Message**: The received message option works as an inbox of messages. On choosing this option, we get the inbox which displays the various devices from which messages have been received (Figure 2.2b). The corresponding text message can be viewed by choosing the device (Figure 2.2c).

• **Re-Scan devices**: This option is to refresh the device list so that any new device that may have entered the range or devices which may have stepped out of range can be detected and duly updated.

• **Exit**: Terminates the application.
3. Usage Scenario:

Let us now describe a use case scenario where this Scatternet Application might play a role. Consider a business establishment of various products as shown in the figure below. Every stall has a Scatternet enabled Bluetooth connection.

A customer walks in and has a difficult time assessing the stores present and the one which would provide him with the best favorable price for his item, a pendrive. So all he has to do is to key in the words “search pendrive” and send it to the nearest available Bluetooth device. The search is done throughout the Scatternet and the products matching his specification are returned with the price attached and the store which provides it. Thus saving him a lot of effort, time and maybe money.
4. Mobile Scatternet Application (Simulation):

The snapshot shown here shows the initial screen of the mobile simulation Scatternet application. Why we need to modify the original application to simulate is that there was no option in the wireless toolkit to keep devices out of range.

The only difference you might notice here from the other application is the extra option of a virtual Scatternet. This option is to simulate a Scatternet without using actual devices. This can be used to demonstrate the Scatternet over a large number of hops. Figure 3.1 below shows the snapshot of three emulators running the application and on which the virtual Scatternet option has been selected. On each screen you get a list of devices visible to it. Here you may notice that the first device has the address of the second and the second device has the address of the first. The third device is visible to neither but it detects the other two and then sends it information to it. Now to simulate the Scatternet we select a device in one of the screens and remove it by selecting the remove button.

Note: Even though we might not remove any device in the other two emulators, its necessary that we still select the ‘hide’ command on each screen. This is to get the Scatternet simulation going. Figure 3.2 shows the renewed routing table after this is done.
You can notice the hop count as being two in the case of the third device when it wants to send it to the first device. The details of what happens when the message has to be sent from the third device to the first can be explained, again, with the help of snapshots (Figure 3.3a to 3.3f).
Figure 3.3a: Third device gets ready to send a message to the first device.

Figure 3.3b: Device 3 chooses device 1 as the recipient of the message.
Figure 3.3c: After selecting device 1, above snapshot depicts the message in transmission. Device 3 sends it to device 2 (Indicated by both devices lighting up).

Figure 3.3d: Device 2 sends it to device 1 and device 1 flashes an instruction to check the inbox for the message.
Figure 3.3e: Device 1 checks the inbox and sees the message sent by device 3.

Figure 3.3f: Message displayed on screen of device 1
5.0 **Runtime Requirements.**

5.1 **PC Application Requirements.**

- Bluecove 2.1 API.
- MYSQL JDK JDBC driver.
- WIDCOMM (Broadcom) BTW Stack software version 1.4.2.10 SP5 or above
- Microsoft Bluetooth stack
- Linux with BlueZ Bluetooth stack (Best Recommended)
- J2SE Runtime Environment.

5.2 **Mobile Application Requirements**

- J2ME enabled device.
- MIDP 2.0
- CDLC 1.1
- Series 40 type

6.0 **Reference**

- en.wikipedia.org/wiki/Scatternet
- http://code.google.com/p/bluecove
- today.java.net/pub/a/today/2005/02/09/j2me1.html