Advances in Engineering Science
Sect. C (2), April-June 2008, PP 19-24

# A JAVA PLATFORM BASED APPROACH FOR IMPLEMENTING CUSTOMIZED BLUETOOTH PROFILES

**Prof. P. B. Mane[1]*, Prof. M. S. Karyakarte[2], Prof. Dr. R. K. Prasad[3], Lalit Kulkarni[4]**

1, 2 - Bharati Vidyapeeth's College of Engineering for Women, Katraj-Dhankawadi, Pune-43, India
3 - KJ Somaiya IEIT, Sion Mumbai, India
4 - University of Abredeen, Abredeen, Scotland, UK

## ABSTRACT

*Bluetooth is a low cost, short-range radio technology, originally developed as a cable replacement to connect devices such as mobile handsets, headsets and portable computers. This technology has standardized wireless communications between electronic devices. It has created the notion of a Personal Area Network (PAN) approved by IEEE as 802.15.1 standard, a kind of close range wireless network that looks set to revolutionize the way people interact with the information technology landscape around them. .*

*Sun Microsystems developed Java2 Micro Edition (J2ME) for compatibility with mobile devices, which includes class definitions to make it possible for the programmers to design software according to the necessity, which is compatible with Bluetooth device. Connected Limited Device Configuration (CLDC) devices have slow processors and limited memory, operate on batteries, and have only intermittent network connections. To support CLDC devices J2ME also includes Mobile Information Device Profile (MIDP) provides the java runtime environment. The paper presents implementation of Bluetooth file transfer profile using java. The implementation is done using J2SE and use Object Exchange (OBEX) protocol as well as Radio Frequency Communication (RFCOMM) protocol.*

***Keywords:*** *Bluetooth, Profiles, JABWT, FTP, OBEX*

## INTRODUCTION

The layered Bluetooth protocol stack, like ISO OSI model, has each layer designed to perform a specific task in the overall Bluetooth device The minimal Bluetooth protocol stack embodied in the Bluetooth device is illustrated below (Figure 1). The stack is the software layer with direct access to the Bluetooth radio to control device settings, communication parameters and power levels. [1][3]

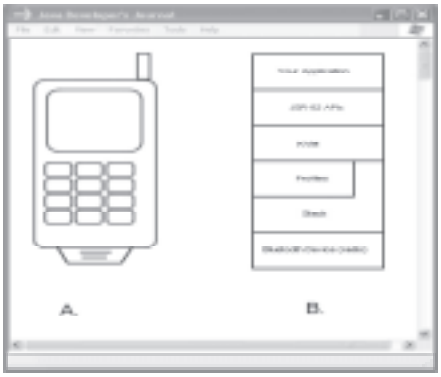

**Figure 1 : Bluetooth Profiles**

## BLUETOOTH PROFILES[2][7]

Bluetooth profiles (Figure 2) are designed set of functionalities to allow different Bluetooth devices to interoperate. A Bluetooth profile defines standard ways of using hardware and communication protocols that enable a particular usage model. In other words, it defines how different parts of the Bluetooth specification can be used for implementing customized applications. Two profiles may use a different set of protocol layers and a different set of features within the same protocol layer. A Bluetooth device can support one or more profiles adopted by the Bluetooth SIG.

*pbmane6829@rediffmail.com

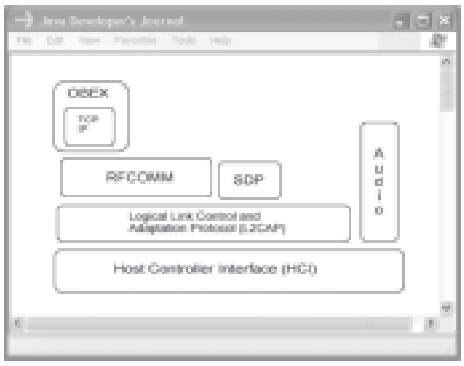


**Figure 2 : Different layers of Bluetooth Profiles-stack**

- **Generic Access Profile (GAP)**

  The GAP is the basis of all other profiles.GAP defines the generic procedures related to establishing connections between two devices, including the discovery of Bluetooth devices, link management and configuration and procedures related to use of different security levels.

- **Service Discovery Application Profile(SDAP)**

  The SDAP describes the fundamental operations necessary for service discovery. This profile defines the protocols to be used by applications to locate services in other Bluetooth-enabled devices.

- **Serial Port Profile(SPP)**

  The SPP defines the requirements for Bluetooth devices necessary for setting up emulated serial cable connections using RFCOMM between two peer devices.SPP directly maps to RFCOMM protocol and enables legacy applications using Bluetooth wireless technology as a cable replacement.

- **Generic Object Exchange Profile (GOEP)**

  The GOEP is an abstract profile on which concrete usage case profiles can be built. These are profiles using OBEX. The profile defines all elements necessary for support of the OBEX usage models.

- **File Transfer Profile (FTP)**

  Provides access to the file system on another device. This includes support for getting folder listings, changing to different folders, getting files, putting files and deleting files. It uses OBEX as a transport and is based on GOEP.

J2ME is targeted specifically at consumer devices and embedded devices. It consists of a set of configurations and profiles (not to be confused with Bluetooth profiles)[8]. A J2ME configuration defines the minimum set of Java class libraries and virtual machine features supported on a particular category of devices. The Connected Limited Device Configuration (CLDC) is the most relevant one for Bluetooth applications as it targets mobile devices with small memory budgets (160KB to 512KB) and wireless, potentially intermittent connectivity. It comprises a fast, small footprint virtual machine (the KVM) and a stripped down but fully functional Java API subset. A J2ME profile is targeted at application developers. It is layered on top of a configuration and is the set of APIs available on a particular family of devices. Configurations target horizontal market segments whereas profiles target vertical segments. One important profile is the Mobile Information Device Profile (MIDP), which provides a set of User Interface components, a persistence mechanism and a HTTP connection capability for use in mobile phones, PDAs and other handheld mobile devices [9].

How an end user uses Bluetooth wireless technology varies from person to person. Two people with the same model of a Bluetooth-enabled phone might want to use it for different purposes. To make downloading applications a reality, one needs a standard API that lets programmers write Bluetooth applications that work across many hardware platforms. To define this standard API, the java language is the ideal choice. A Java API enables applications to run on different types of hardware, operating systems, and classes of device. In addition to portability, the JAVA language provides several benefits such as:Rapid development of applications, Ability to dynamically expand a program's functionality during execution , Class files verification and security features.

Bluetooth Technology has been the leading field of research in past few years. Software's have been developed in various High Level languages (HLL) such as C++, dot net, C #, C sharp. Sun Microsystems developed a new version of Java (J2ME) to make it compatible with mobile devices. Class definitions have been added in both the versions of Java i.e. J2SE and J2ME using the **Java APIs for Bluetooth Wireless Technology (JABWT) [9]** to make it possible for the programmers to design software with customized profiles (applications) for Bluetooth-enabled desktops as well as mobile devices (Cell phone, PDA, laptop etc.).

The paper presents the case study for implementing ad-hoc[6] wireless local area network in small offices where the number of PC's range between 2 to 8 wherein the office data in the form of text, voice and image can be communicated among the host PC's and Bluetooth enabled devices like printer and mobile phones [5]. The chat option in the developed software is especially useful during meetings where confidential information is to be exchanged. The information collected by the Bluetooth enabled mobile phone can be immediately downloaded on the host PC in the office and may be printed [4].

**IMPLEMENTATION**

```
//PICONET Formation
main()
{     if login successful          then
case 1: discover_device(master x);break;
case 2: file_transfer(master x, slave y);break;
case 3: chat(node x, node y) break;
}//main ends
discover_device(master u)
{
//u is in INQUIRY mode while other devices are in
//the INQUIRY_SCAN mode
      i=0;
```

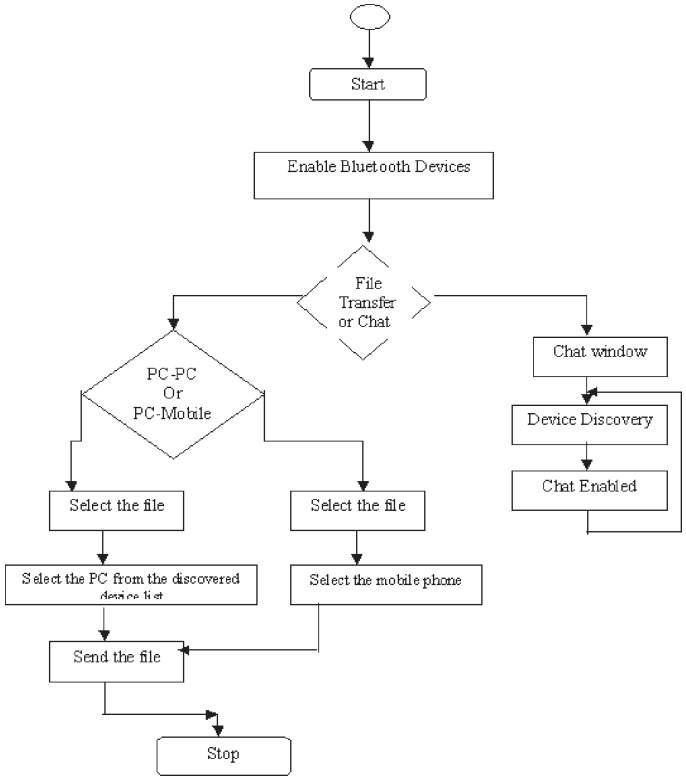


**Figure 3 : Flow diagram for communication system Pseudocode for the piconet formation [8]**

```
        while(i<3)
         for(j=0;j<256;j++)
         {if (INQUIRY(u))
  //A slave is in scan mode gives response and moves to page scan
            if (PAGE(u)) {
                send (u,v,pFHSS_packet);
                receive(u,v,FHSS_packet);
                // connection established
                add_to_slave_list(u,v);
            }}}}
  file_transfer(master u, slave v)
  {     //select a file at master u
        select (u, file);
        sendfilename(u,v);
        send(u,v,file);
        while(true){
         wait(u);
         if(file_received);
          break;}}
  file_receive(slave v, master u)
  {     while(!DAC)
         standby(v);
        fname=receivefilename(v,u);
        open(fname,"w");
        while(!eof){
         read_data(buff);
         write(fname, buff);
         flush(buff);       }
        close(fname);
        send_ack(v,u);
  }//file receiving
  chat(node u, node v)
  {     if(SEND){
         accept_kbd(chattext);
         send(u,v,chattext);       }
        if(RECEIVE){
         chattext=receive(v,u);
         display(chattext);        }}
```

**RESULTS**

The experimental setup was used to carry out the following tests.

1. Device discovery to test the time taken for establishing the piconet.
2. Text, voice and image file transfer for varying file size and fixed distance to test the date rate.

Text, voice and image tile transfer for fixed file size and varying distance to test the data rate.

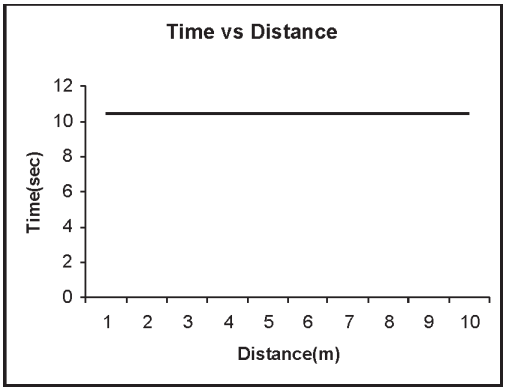


**Figure 4 : Device discovery**

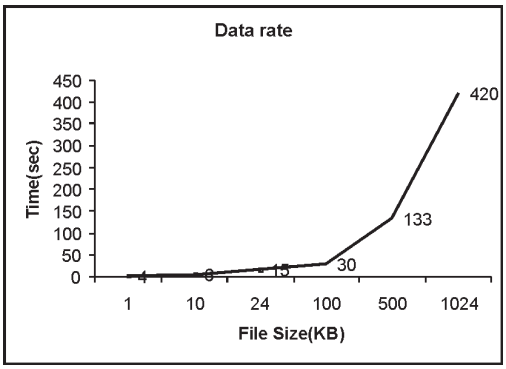


**Figure 5. file size vs. Time for 2m distance (mobile-PC)**

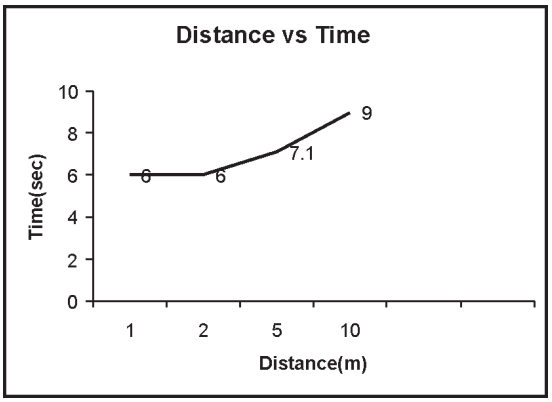


**Figure 6 : Distance Vs. time for 2 KB file size**

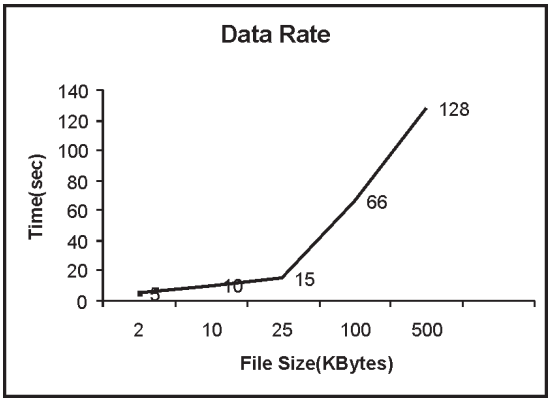


**Figure 7 : Data transmission file size vs. time (PC –PC)**

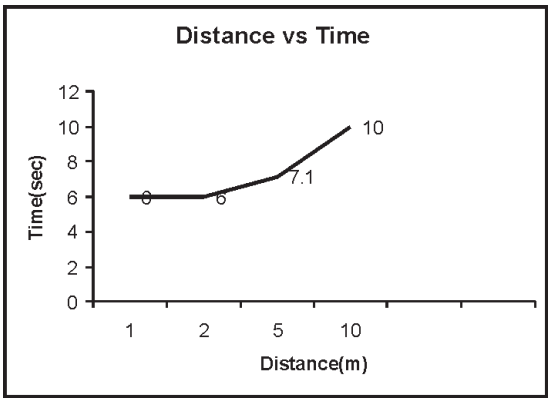


**Figure 8: Distance vs. time for 9 KB file size**

## CONCLUSION

The piconet system was designed and implemented using JAVA. The time for Bluetooh enabled PC, laptop or mobile discovery is independent of the distance between PC's within the piconet. It was observed that device discovery time is 9.5 seconds since its execution time is predefined during the software design and development. The file transfer time is directly proportional to the file size. It was observed that for transfer of 1 KB file between PC and Mobile the time taken was of 4 seconds and for 1 MB file it was 420 seconds and for transfer of 2 KB file between PC to PC the time taken was of 5 seconds and for 500 KB file it was 128 seconds The test was taken for fixed file for varying distances and it was observed that the file transfer time is directly proportional to distance. For the file size of 2 KB between PC to Mobile for distance of 1m the transfer time was 6 seconds and for 10 m it was observed to be 9 seconds and for the file size of 9 KB between PC to PC for distance of 1m the transfer time was 6 seconds and for 10 m it was observed to be 10 seconds

## REFERENCES

1. Jaap Haartsen et al., ***"Bluetooth: Vision, Goals, and Architecture,"*** Mobile Computing and Communications Review, Volume 1, Number 2,Oct, 1998 PP 1-8.

2. Riku Mettala, ***"Bluetooth Protocol Architecture Version 1.0,"*** Bluetooth white Paper, Nokia Mobile Phones, 29 August, 1999, PP 1-20.

3. Jaap Haartsen ***"The Bluetooth Radio System,"*** IEEE Personal Communication, February 2000, PP 28-36.

4. Noki Sugiyama, ***"Core Technologies of Bluetooth Systems,"*** Ok I Technical Review 184, Vol 167, Dec. 2000, PP 28-31.

5. Pravin Bhagwat, ***"Bluetooth: Technology for Short Range Wireless Apps,"*** IEEE Internet Computing, May-June 2001, PP. 96-103.

6. Mario Gerla et al, ***"Ad hoc Networking with Bluetooth,"*** WMI at Mobicon 2001

7. Bray & Struman, ***"Bluetooth: Connect without Cables"***, Printice Hall Publication, 1st Edition, 2001.

8. C. Bala Kumar, Paul J. Kline, **"*Bluetooth Application Programming with the Java APIs"***, Morgan Kaufmann Series n Networking, 2004

9. Java API for Bluetooth, www.developer.sun.com

10. Bluetooth USB hardware modules www.microsimcom.com

■