Designing ActiveSync for Pocket PC using Wireless Technology

Shalini Bhaskar, S.K.Saxena, *Member, KIMICS*

***Abstract*—This paper presents proposal for an application (only for Windows-based Mobile) that can synchronize the Third Generation Windows-based mobile with the prescribed server. This application can be used by mobile users to send information directly to the WAS server wirelessly. In this paper a comparison between Third Generation technology and WLAN (both are parallel technologies first one embodied by IMT-2000 standards and the second one by ITU Standards) has been made and also a comparison between Wi-Fi and Bluetooth has been made.

***Index Terms*— Wi-Fi, Bluetooth, WLAN, 3G, Pocket PC, Wireless USB, WAS Client, Next Generation Wireless Systems, CDMA2000, HSDPA, SACK, HRPD, AP, 3GPP.

I. INTRODUCTION

Over the past decade there is an explosive and parallel growth of the Internet and the mobile telephone services. Internet gives data communication benefits to the masses with email, E-Commerce, the Web whereas mobile services provide "follow-me-anywhere/always on" telephony. Now the two worlds (WLAN and 3G) are converging offering benefits of interactive multimedia services coupled to the flexibility and mobility of the wireless. The integration of different technologies with different capabilities and functionalities is an extremely complex task and involves issues at all the layers of the protocol stack. TCP is a major protocol among wireless access terminals supporting Web browsing and e-mail checking. Current packet data cellular systems, such as the CDMA2000 High Rate Packet Data(HRPD), 1xEV-DO, or TIA/EIA IS-856 [1], [2], and the WCDMA Release 5 High-Speed Downlink Packet Access (HSDPA) [3], cannot provide differentiated TCP services over shared-channel air links. TCP's end-to-end flow control is basically a closed-loop, additive-increasing-multiplicative-decreasing(AIMD) window control mechanism [4] regardless of TCP versions such as Reno [5], NewReno [6], and SACK [7]. Although such a mechanism underlines TCP's excellent scalability and robustness to unpredictable dynamics inside a large scale wired network, it is infamously heuristic and hard to manage in wireless environments.

Manuscript received March 12, 2008; revised May 18, 2008. Shalini Bhaskar, S.K.Saxena, Institute of Technology and Management, Gurgaon-122001, India, Delhi College of Engineering, University of Delhi, Delhi,

Next Generation Wireless Systems(NGWS) integrate existing wireless networks such as wireless local area networks (WLANs), third generation (3G) cellular networks and satellite networks to realize a unified wireless communication system that has the best features of the individual networks to provide ubiquitous "always best connection"[8] to mobile users [9]. The combination of cellular core network infrastructure with its service centers and accounting capabilities with the achievable data rate of WLANs can provide the possibility of offering new services to the customer [10]. Recent advances in wireless communications have expanded possible applications from simple voice services in early cellular networks(1G, 2G) to new integrated data applications. Wireless LANs (WLANs) based on IEEE 802.11 family have recently become popular for allowing high data rates at relatively low cost. WLAN access points(APs) may provide hotspot connectivity in the most common places, such as airports, hotels, shopping malls, schools, university campuses, and homes. The Third Generation Partnership Project (3GPP) has been developing an internetworking architecture for 3G cellular systems and WLANs with the aim of enhancing the services provided to subscribers by 3G operators. An overview of the proposed architecture is given in [11]. The 3GPP2 group is also working on 3G/WLAN inter-working, but it mostly addresses WLAN/CDMA2000 interoperation [12].

From the above discussion, we can outline the basic similarities and dissimilarities between 3G and WLAN. 3G and WLAN are similar in the sense that both : (1).are wireless. (2). are access technologies. (3). offer broadband data services. 3G and WLAN are different from the point of view of : (1). Spectrum policy and management. (2). Deployment status. (3). Embedded support for services. (4). Standardization. (5). Service/business model. A comparison of the two technologies reveals that Third generation technology has the capacity to send packets of data at high speeds across digital wireless network. We pay for the volume of data we send and receive. Downloading 15MB of data will take 6-8 minutes at 264 kbps or 2MB a minute. Wi-Fi is based on high frequency Wireless Local Area Network(LAN), is the way of the future. Its good value because we pay for the time we are connected, not the data we send and receive. Coverage is short range but available in many of the places you need it most (e.g. hotels, airports, etc.). Downloading 15MB of data will take 3-10 minutes at 500 kbps.

Table 1 Comparison between 3G and Wi-Fi

	3G	Wi-Fi
Speed	upto 384 kbps	upto 512 kbps
Availability	Most locations	Hotspots world-wide
Messages	Text-based	E-mail
Browsing	Limited	Full capability
Images	Limited	Full capability

It must be emphasized that Wi-Fi is efficient and fast, especially if we need to transfer large data files such as music or video downloads and data heavy or image rich documents. Wi-Fi saves our time. We can send and receive email, use the web and log in to a company network-when and where we want, and all at broadband speeds. These intelligent 3G mobiles are capable of storing large chunks of data. But there is a limit to storing information on the 3G mobile. So, the information should be sent to the server, which may be considered to act like a sink from time to time . For this an application already exists that synchronizes 3G mobile with the server. The existing technology used for connecting 3G mobile with the server is either Bluetooth in which the 3G Mobile needs to be in a specified range (i.e. 10 to 100 mts.) from the server or by using data cable to physically connect 3G mobile with the server.

The problem here is that the 3G mobile either needs to be in a specified range to the server (if not connected physically) or it is to be connected to the server by data cable for synchronization. To overcome this limitation of distance an application has to be devised that can synchronize the 3G mobile(Windows-based) with the server from anywhere in the world. This is done by using Wi-Fi for connecting the 3G mobile to the server.

Rest of the paper contains comparative study of Bluetooth and Wi-Fi technology, technology used for the proposed application, Hardware and Software requirements for the proposed application, implementation issues as well as working of the proposed application, few program implementation modules and finally the conclusion and future scope of work.

II. Bluetooth vs Wi-Fi

Bluetooth and Wi-Fi have slightly different applications in today's offices, homes, and on the move: setting up networks, printing, or transferring presentations and files from 3G Mobile to computers. Both are versions of unlicensed spread spectrum technology. Bluetooth differs from Wi-Fi in that : (1). The latter provides higher throughput and covers greater distances, but requires more expensive hardware and higher power consumption. (2).They use the same frequency range, but employ different multiplexing schemes. (3).While Bluetooth is a cable replacement for a variety of applications, Wi-Fi is a cable replacement only for local area network access. That's why Bluetooth is often thought of as wireless USB, whereas Wi-Fi is wireless Ethernet, both operating at much lower bandwidth than the cable systems they are trying to replace. However, this analogy is not entirely accurate since any Bluetooth device can, in theory, host any other Bluetooth device—something that is not universal to USB devices, therefore it would resemble more a wireless Firewire.

III. TECHNOLOGY USED

The technology used in our proposed application is Wi-Fi. As mentioned earlier, in section I, Wi-Fi provides connectivity of 3G Mobile to the desired computer acting as server(Sync) without using cables. Using a Wi-Fi setup with our 3G Mobile or laptop means that we can share a broadband or dialup Internet connection, swap files, or share a printer or CD ROM between computer users. Wi-Fi comes in two speeds: 802.11b (data transfer rates up to 11 megabits per second) or the newer to 54 mbps, 802.11g (data transfer rates up to 54 megabits per second). This compares with Bluetooth's much slower speed of 0.57 megabits per second. There's a newer standard, 802.11n due in 2007 that offers even faster connectivity. Devices based on the 802.11g standard are backwards compatible with 802.11b equipment, may be a better investment. Wi-Fi 802.11b/g operates in the 2.4GHz frequency band (also used by Bluetooth and microwave ovens), and has a typical range of around 500 feet (with clear line of sight). Indoors, you can expect around 150 feet with 802.11 - this will increase with the 802.11n protocol. There are a number of Mobile Phones and PDAs that have built-in Wi-Fi. You get cheaper voice calls by using a Wi-Fi phone to route your phone calls over the Internet.

IV. HARDWARE AND SOFTWARE REQUIREMENTS

In this section we would like to outline the basic hardware and software requirements for designing the application. These are:

A. ***Hardware:***

Pocket PC
Intel Celeron M Processor 420 (1.6GHz, 533 MHz FSB, 1MB L2 Cache)
256 MB DDR2
802.11b/g wireless LAN
Card Reader

B. ***Software:***

Windows XP
Microsoft Visual C++ 6.0
eMbedded VC++4.0 with SP2
Microsoft eMbedded Visual Tools

Windows Mobile 5.0 MSFP Emulator Images
Microsoft Windows SDK for Pocket PC 2002
Standard SDK for Windows CE .NET 4.2

V. IMPLEMENTATION ISSUES

In this section we will discuss various implementation issues. Implementation issues for the proposed application are as follows: The synchronization protocol should be designed in such a way that it requires minimum processing and bandwidth. It may require compression and decompression of heavy office and multimedia content. For folder synchronization, the entire directory structure should be available on the WAS server. For phone book synchronization, an input interrupt with new phone book details should be generated. The entire phone book should be updated on the WAS server. For SMS synchronization, there should be a complete duplicated data contents of all the SMS with header details on the WAS server which should be synchronized with the mobile client. The update time could be once in four hours or once in a day. There parameters could be configurable from end user point of view. It should generate input interrupt with new SMS details. The data should be arranged in native protocol format and delivered to the server instantly. The entire SMS should be updated on the server. Similarly, for synchronization of MMS with the WAS server is another issue. There should be a complete duplicated data contents of all the MMS with header details on the WAS server which should be synchronized with mobile client. The updated time could be once in 4 hours or even once in a day. Again, these parameters could be configurable from the end user point of view. It should generate an input interrupt with new MMS. Both SMS and MMS are not performance critical problems.

Being an embedded application, the client can communicate with other peripherals of the target device like TCP/IP module, GPRS/3G Modem as well as Application Processor. Client support a user friendly Graphical User Interface (GUI). End user should be able to configure various performance critical parameters as well as communication details through the GUI. Application has been developed on TCP/IP layer using Winsock2 framework and will interact with hardware through MMC driver, display driver. Performance Requirements: (1). Foot-print Requirements: As per Virtual Memory layout provided in WinCE 5.0, no application could go beyond 32MB as a run time memory requirement. Each process is allocated only 32MB slot for execution. WAS Client should be a tiny FTP server kind of thing with very light user defined communication protocols for the data synchronization. The client would not accept the application if it goes beyond 2MB in its run time memory requirement. (2). Speed Requirement: WinCE does not support Kernel mode driver that is why performance becomes a major constraint specially for networking applications. All communication drivers available in WinCE 5.0 are user mode drivers and when these drivers are loaded by the Kernel, it requires large number of process swapping while processing any interrupt generated by any third party application. Client commits not more than 115kbps application level bandwidth for the WAS Client. The WAS Client has to perform all synchronization operations within this available bandwidth even if it synchronizes high rich contents like multimedia and office data. (3).Real Time Requirement: The application has to be developed on a hard real time, pre-emptive kernel WinCE 5.0. to understand the complete O/S architecture and development environment.

VI. IMPLEMENTATION

Socket Programming is a very interesting activity in most of the programming languages. Its a nice activity to write Servers and Clients that communicate over a network. In Windows Platform, socket communications are based on MS Winsock architecture. Windows supports both stream based (TCP) and Datagram based (UDP) socket communication.

For Socket programming, MFC provides two built in classes named CAsyncSocket and CSocket. CSocket is inheriting its functionality from CAsyncSocket. CAsyncSocket class provides several notification functions, that will be called automatically upon occurrence of the socket events. More over, it acts as the base class for the complete event driven socket communication. We can create our on customized socket classes by inheriting from CAsyncSocket class, which will serve our application specific needs. At first socket environment needs to be initialized by calling AfxSocketInit() function.

A. *Initializing Socket*

To initialize sockets ,we need to call the function AfxSocketInit(). It is usually called from the InitInstance() function of the MFC application. If we are using wizard to generate the application, checking the option "use Windows Sockets" will automatically do this job for us. It returns a value to indicate success or failure of the call.

```
BOOL CEchoServerApp::InitInstance()
{....
if( AfxSocketInit() == FALSE)
{
AfxMessageBox("Sockets Could Not Be Initialized");
return FALSE;
}
```

B. *Creating Server Socket*

To Create a Server socket, we need to declare a variable of type CAsyncSocket or our own class derived from CAsyncSocket or CSocket, then we must call the create() function with the port to be listened as argument. It returns a value to indicate success or failure of the call.

```
UpdateData(TRUE);
```

```
m_sListener.Create(m_port);
if(m_sListener.Listen()==FALSE)
{
AfxMessageBox("Unable to Listen on that port,
please try another port");
m_sListener.Close();
return;
}
```

C. *Creating Client Socket*

To Create a Client socket, first we need to declare a variable of type CAsyncSocket or our own class derived from CAsyncSocket or CSocket. Then we must call the create() function without any arguments. It returns a value to indicate success or failure of the call.

```
m_sConnected.Create();
m_sConnected.Connect("server ip",port);
```

D. *Listening for incoming connection*

For making the server socket listen to the Specific port, we must call the function Listen(), it returns a value to indicate success or failure of the call .

```
if( m_sListener.Listen()== FALSE)
{
AfxMessageBox("Unable to Listen on that port, please
try another port");
m_sListener.Close();
return;
}
```

E. *Accepting Connection*

The incoming connection must be accepted to another socket (not the listening socket).its done by calling the Accept function with the second socket as argument.

```
void CEchoServerDlg::OnAccept()
{
CString strIP;
UINT port;
if(m_sListener.Accept(m_sConnected))
{
m_sConnected.GetSockName(strIP,port);
m_status="Client Connected,IP :"+ strIP;
m_sConnected.Send("Connected To Server",
strlen("Connected To    Server"));
UpdateData(FALSE);
}
else
{
AfxMessageBox("Cannot Accept Connection");
}
}
```

F. *Sending Data*

The data to be sent is kept in a buffer and a pointer to it and its length is passed to the send function

```
m_sConnected.Send(pBuf,iLen);
```

G. *Receiving Data*

Data is received by calling the function receive(buffer,maxlength)

```
void CEchoServerDlg::OnReceive()
{
char *pBuf =new char [1025];
CString strData;
int iLen;
iLen=m_sConnected.Receive(pBuf,1024);
if(iLen == SOCKET_ERROR)
{
AfxMessageBox("Could not Recieve");
}
else
{
pBuf[iLen]=NULL;
strData=pBuf;
m_recieveddata.Insert(m_recieveddata.GetLength(),str
Data);
 //display in server
 UpdateData(FALSE);
 m_sConnected.Send(pBuf,iLen);    //send the data
back to the Client
delete pBuf;
}
}
```

H. *Shutting Down Connection*

```
m_sConnected.ShutDown(0);Stops Sending
Of Data

m_sConnected.ShutDown(1);Stops Receiving of Data

m_sConnected.ShutDown(2);        Stops Both
Sending and Receiving of    Data
```

(i). Closing Connecton

```
m_sConnected.Close();
```

VII. CONCLUSION AND FUTURE SCOPE OF WORK

The application that has been proposed will work for the Pocket PC which is a windows-based 3G mobile. The application can not work for the 3G Mobile in which operating system is not Windows. To make it work for any of the 3G mobile an abstraction layer need to be developed that will make the application platform independent, so that, it can be installed on any of the 3G mobile.

REFERENCES

[1] 3rd Generation Partnership Project 2 (3GPP2) C.S0024, v. 4.0, CDMA2000 High Rate Packet Data Air Interface Specification Dec. 2001.

[2] 3rd Generation Partnership Project 2 (3GPP2) C.S0024-A, v. 1.0, CDMA2000 High Rate Packet Data Air Interface Specification Mar. 2004.

[3] High Speed Downlink Packet Access (HSDPA): Overall Description, v. 5.2.0, 3gpp Technical Specification 25.308, Mar. 2002.

[4] D. Chiu and R. Jain , " Analysis of the Increase and Decrease Algorithms for Congestion Avoidance in Computer Networks", Computer Networks and ISDN Systems, vol. 17 no.1, pp. 1-14, June 1989.

[5] W. Stevens, TCP Slow Start, Congestion Avoidance, Fast Retransmit, and Fast Recovery Algorithms, IETF RFC 2001, Nov. 1990.

[6] S. Floyd and T. Handerson, The NewReno Modification to TCP's Fast Recovery Algorithm, IETF RFC 2582, Apr. 1999.

[7] M. Methis et al., TCP Selective Acknowledgement Options, IETF RFC 2018, Oct. 1996.

[8] E. Gustafsson and A. Jonsson, "Always Best Connected", IEEE Wireless Comm., pp. 49-55, Feb. 2003.

[9] I.F. Akyildiz and W. Wang, "A Predictive User Mobility Profile for Wieless Multimedia Networks", IEEE/ACM Trans. Networking, vol 12, no. 6, pp. 1021-1035, Dec. 2004.

[10] Frank H. P. Fitzek, Petar Popovski, Michele Zorzi, " A Symbiotic Perspective on Low-Cost Cellular And Multihop WLAN Internetworking Solutions", IEEE Wireless Communication , Vol 12, no. 6, pp. 4-10, Dec. 2005.

[11] K. Ahmavaara, H. Haverinen and R. Pichna, "Inter-working Architecture Between 3GPP and WLAN Systems", IEEE Comm. Magzine, vol. 41, no. 11, Nov. 2003.

[12] M. Buddihikot et al., " Design and Implementation of a WLAN/CDMA2000 Inter-working Architecture", IEEE Comm. Magzine, vol. 41, no. 11, Nov. 2003.

Shalini Bhaskar

She did her B.E.(Computer Science and Engineering) from Deenbandhu Chhotu Ram University Of Science and Technology, Murthal(Sonepat), India(formerly, C.R.S.C.E., Murthal (Sonepat)) in 1999. She is M.E. (Computer Technology and Applications) from Delhi College of Engineering, University of Delhi, Delhi India. Presently, working as Senior Lecturer with Institute of Technology and Management, Gurgaon, India. Her research interests include Wireless Communications, Computer Networks, Database Management Systems, Emdedded Systems..

S.K.Saxena

Received M.E. degree in Computer Technology and Applications from Delhi College of Engineering, University of Delhi, Delhi, India. Got Ph.D. degree from Lucknow University, Lucknow, India in 1991. Joined Delhi College of Engineering, University of Delhi, Delhi, India in 1994. Presently, working as Head, Computer Centre, Dehi College of Engineering, University of Delhi, Delhi, India . His research interests include Object Oriented Programming, Computer Networks, Computer Graphics.