Safe In-flight Mobile Telephony

Indian Institute of Technology, Bombay

Powai, Mumbai – 400076,

INDIA.

SkyMobile

IEEE Computer Society International Design Competition 2001

Aditya Dua (dua@ee.iitb.ac.in)

Aman Kansal (aman@ee.iitb.ac.in)

Arjunan R (arjun@ee.iitb.ac.in)

Sumitra Ganesh (sumi@ee.iitb.ac.in)

Vivek Raghunathan (vivek@ee.iitb.ac.in)

Mentor: Professor U.B Desai (ubdesai@ee.iitb.ac.in)

Abstract

Air passengers are required by the law to switch off their mobile phones on board any flight [1]. This requirement has been imposed due to two reasons. First, signals emitted by the mobile phone interfere with Air Traffic Control (ATC) signals, undermining the safety of the flight. Second, a mobile at such an altitude connects to multiple base stations simultaneously, clogging the resources of the ground network. We have developed a novel solution based on the integration of diverse communication links: **Bluetooth, Cellular Network (GSM/IS-95)[1], PSTN** and **Air-to-ground connection**. Our solution enables the user to remain connected in-flight, while solving the above two critical issues. The switch over from the cellular network to our in-flight Bluetooth network does not require any user initiation or change of the mobile handset. Bluetooth, due to its low power, short range and fast frequency hopping presents negligible interference to ATC signals. When the passenger enters the plane, call forwarding is set up from the cellular network to our Ground Switching Center (GSC) and the hazardous GSM emissions of the mobile phone are automatically switched off. All voice (or data) is received at the GSC and transferred through an air-to-ground link to a Bluetooth Airplane Gateway (BTAG) in the plane. Data received at the BTAG is finally transmitted over an in-flight Bluetooth network to the passenger. We have implemented a Bluetooth enabled GSM phone, (on a laptop using a GSM modem and a Bluetooth kit), the Bluetooth Airplane Gateway and the Ground Switching Center (using a phone modem for connecting to the PSTN). The automatic setup up of various communication hops, call routing and transmission of voice over these links has been demonstrated. The system provides a unique and useful service and is perceived to be highly marketable.

[1] The abbreviations used throughout the document are summarized in the Glossary on page 30.

1. System Overview

Currently, air passengers are not allowed to direct access to the cellular network from their mobiles while in flight. The law prohibits the use of mobile phones on aircrafts for two reasons. First, the signals emitted by the mobile phone interfere with Air Traffic Control (ATC) signals, undermining the safety of the flight. Second, a mobile at such an altitude connects to multiple base stations simultaneously, clogging the resources of the ground network.

We develop a Bluetooth based solution for providing mobile phone users with *seamless connectivity* to the cellular network while inside an airplane (Figure 1). We define seamless connectivity to mean the following:

1) The switch over from the cellular network to the Bluetooth network is automatic, not requiring any user initiation.
2) The user's phone number stays the same, and she may receive calls on her usual mobile.
3) No change of handset is required while boarding a flight.

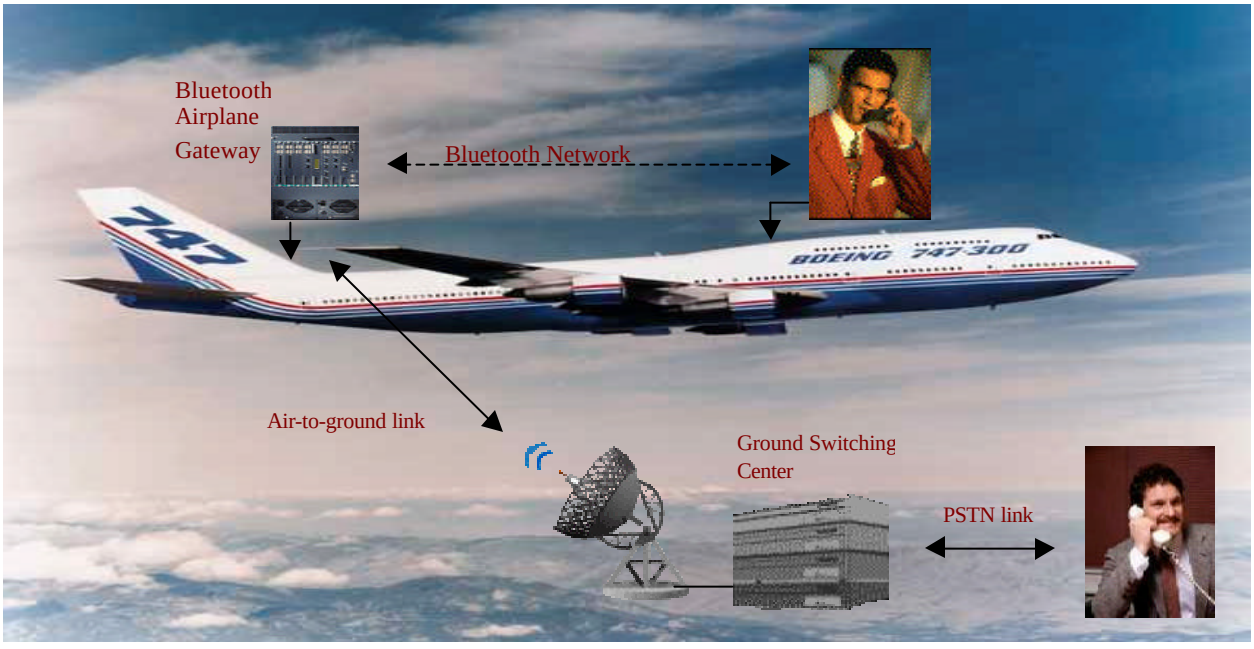

Figure 1: The SkyMobile System

Our Bluetooth Airplane Gateway (BTAG) detects a mobile phone as soon as it enters the airplane. The BTAG instructs the mobile phone to send a message to the cellular network (GSM), asking it to forward all incoming calls for the mobile to an assigned number at our Ground Switching Center (GSC). This is done just before take-off. The cellular network connection is switched off, resulting in all hazardous emissions from the handset being turned off. The handset is now connected through a Bluetooth link to the BTAG, which is in turn connected to the GSC over an approved air-to-ground link [2]-[3]. All incoming and outgoing calls are connected through the GSC to the BTAG, which forwards them to the mobile phone, thus allowing the user to make or receive calls on the usual handset. To execute the above steps, our system needs to perform the following tasks:

1) Automatic detection of mobile phones entering the airplane and exchange of specific instructions for call forwarding and GSM switch off
2) Establishment of a reliable communication link across diverse networks: the cellular network (GSM), the Public Switched Telephone Network (PSTN) and the in-flight Bluetooth network
3) Transfer of Voice Data across this composite communication channel
4) Authentication to provide security and prevent misuse

1.1 Performance Requirements

The main requirement from the system is that the change in connection, while boarding or alighting from a plane, should be seamless. Further:

1) The system should be able to establish connection with negligible failure rate.
2) The voice quality should be comparable to that of cellular networks.
3) Sufficient security measures should be provided to prevent unauthorized usage.

1.2 System Design

Our system consists of three hardware units:

1) Mobile Unit (a Bluetooth enabled mobile handset)
2) Bluetooth Airplane Gateway (BTAG)
3) Ground Switching Center (GSC)

There are three software modules:

1) GSM module (interfaces to the GSM network)
2) PSTN module (interfaces to the PSTN)
3) Bluetooth module (to carry out voice communication over Bluetooth)

In our implementation the GSM module resides on the Mobile Unit, the PSTN module on the GSC, and the Bluetooth module on both the BTAG and the Mobile Unit. These units together provide the functionality required by our solution.

In a full-scale implementation, the GSM module would be implemented on a mobile handset, the Bluetooth module on the mobile handset and on an in-flight BTAG, while the PSTN module would reside at the GSC. The air-to-ground link is not a part of the prototype because such links are proprietary and inaccessible [2]-[3].

We have chosen Bluetooth technology since it offers very low interference to the ATC signals due to fast frequency hopping and low power of transmission (0dBm) which makes the signal strength negligible beyond a short range. Moreover, the mobile phone does not directly connect to the cellular network. Thus, our solution solves the twin problems of ATC interference and ground network clogging. Our novel design integrates disparate communication networks to enable a highly desirable service which provides connectivity, safety and convenience.

2. Implementation and Engineering Considerations

2.1 Operation

An overview of the system operation is shown in Figure 2. The Bluetooth Module of the BTAG detects the Mobile Unit, automatically establishes a connection and performs an authentication procedure. The GSM module then sets up call forwarding and shuts down the GSM stack by sending appropriate commands to the GSM modem. All telephony activity at the Mobile Unit now takes place through Bluetooth. When someone on the ground calls the mobile, the PSTN module receives the call and transfers voice data to the Bluetooth module of the BTAG,

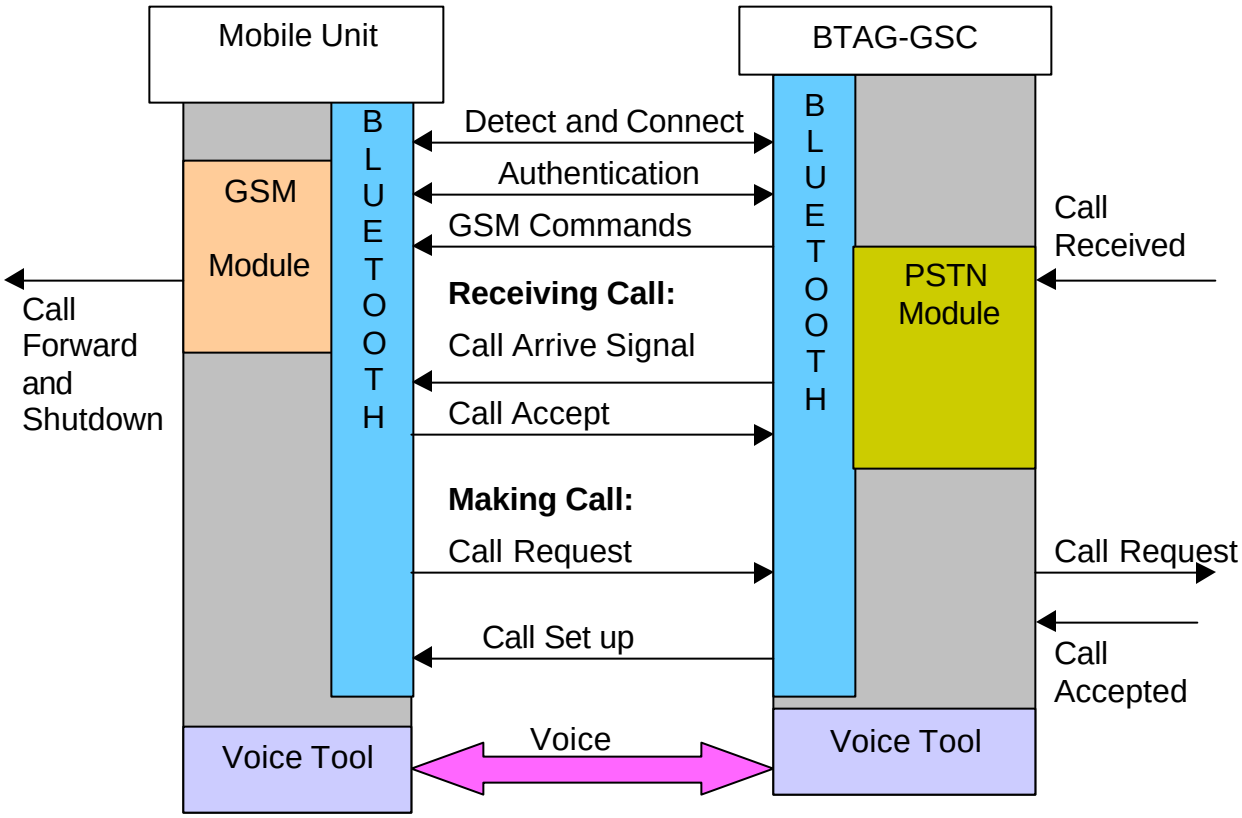

Figure 5: **Overview of the Interaction between the Mobile Unit and the BTAG/GSC unit**

which then transfers the voice across the Bluetooth interface to the Mobile Unit. The Bluetooth module on the Mobile Unit receives this voice data and the voice is streamed out on the speaker. A similar sequence is followed for voice transfer in the other direction.

2.2 System Specification

The main modules of the SkyMobile system (Figure 3) are described in the following sections.

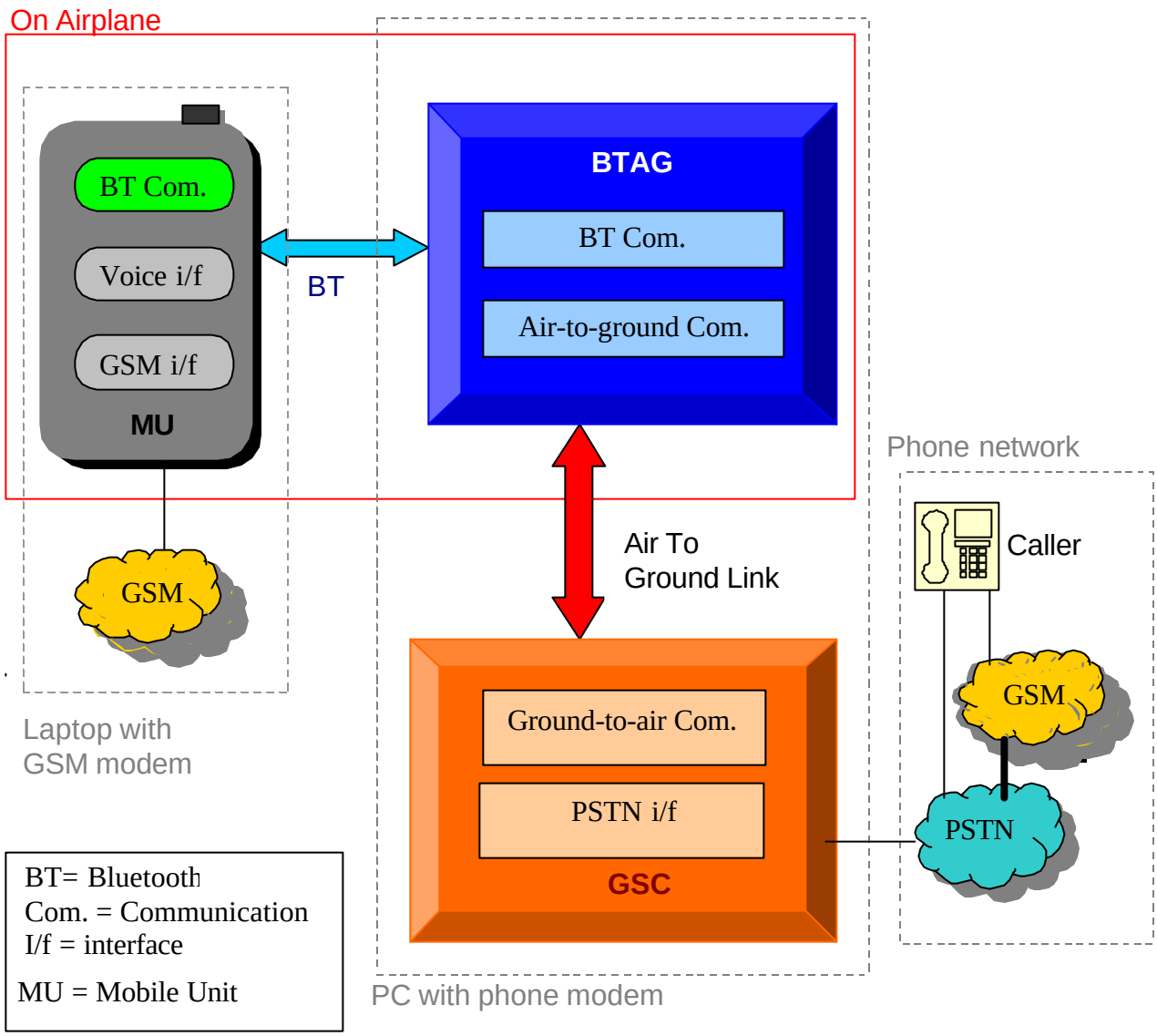

Figure 3: System Block Diagram. The different boxes show the various modules which are part of the system. The dotted boxes show the physical units while the colored ones show logical entities.

2.3 Hardware

The hardware setup consists of three entities:

2.3.1 Mobile unit

This unit emulates a Bluetooth enabled mobile handset. The unit consists of a laptop connected to mobile telephony hardware through a serial port, and to the provided Bluetooth kit through USB (Figure 4). The mobile telephony hardware consists of a GSM modem with its antenna and a Subscriber Identity Module (SIM) card. A headset interface is also available on the GSM modem for voice to be directly sent over GSM.

Figure 4: Mobile Unit

We have used the Wavecom WM0D2 GSM modem, capable of data transfer as well as voice transfer (input of voice being from the modem headset) [4]. The detailed specifications of the Wavecom WM0D2 modem are presented in Table 1. The permission for using this extra hardware was obtained as per CSIDC guidelines.

Frequency Band	900Mhz / 1800Mhz / 1900 Mhz
Audio Interface	Headset, Car Kit
Antenna	SMA Connector
Software Interface	AT Command Set based on V.25ter and GSM 7.07 / 7.05
Interface to Host	RS232 V.24/V.28 Auto Bauding
Services	SMS, Voice, Data, Fax - Class 1

Table 1: Wavecom WM0D2 specifications

Design Tradeoffs: We chose to work on the GSM standard since Bombay (India) is covered only by the GSM cellular network. Our application needs a Mobile Unit that is programmable. Therefore, we have chosen a modem rather than a mobile phone. Some mobile phones do have an interface for programming, but these interfaces are proprietary and do not support the complete GSM instruction set. We have chosen a modem with voice support and auto-bauding to simplify the initial testing. The Wavecom WM0D2 is a dual band 900/1800 MHz modem and will operate in any area with GSM coverage. This modem does not allow a keypad interface or access to its headset through serial port interface. As a result, the headset cannot be used for audio playout and an outgoing call number needs to be provided through the laptop rather than on a keypad.

2.3.2 In Flight BTAG

This unit consists of a PC connected to the provided Bluetooth kit through USB. We have utilized the USB interface rather than the serial interface since the provided API for the USB could be directly used. The BTAG handles the network of Bluetooth ports installed within the flight. It also takes care of routing voice data to the appropriate mobile phone through the corresponding Bluetooth port. The BTAG is connected to the GSC over the air-to-ground link.

2.3.3 Ground Switching Center

The Ground Switching Center consists of a PC connected to a phone modem (Figure 5). The modem is a standard GVC 56K speakerphone modem connected to the PC through its serial port.

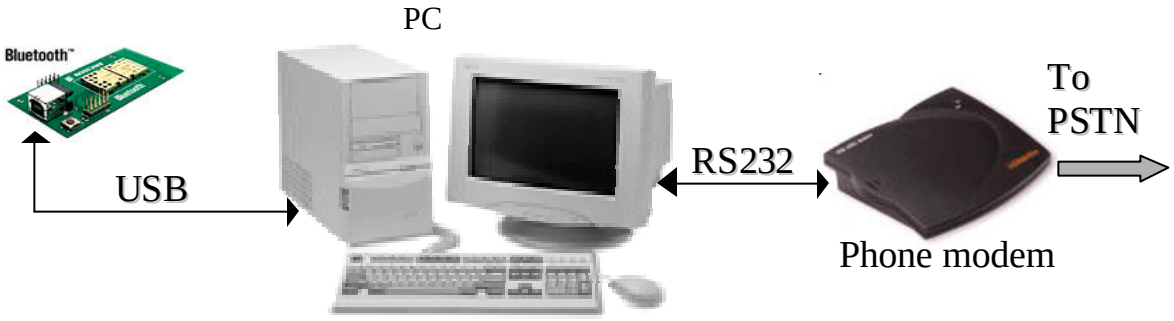

Figure 5: Combined GSC and BTAG

Design Tradeoffs: In our prototype, we allow one active voice call. Handling multiple calls would require a PABX with multiple connections instead of the phone line, and that is outside the scope of the project. If we use a PABX, the Ground Switching Center will be extended in the following manner to achieve the required capabilities:

1) The telephone line will be replaced by a PABX.

2) Every mobile phone number will be mapped to a PABX number in a one-one fashion, reflecting the flight and seat number of the passenger. The Ground Switching Center software will obtain the mapping from the airplane gateway and route phone-calls arriving at a particular PSTN number accordingly.

3) The PSTN module will be extended to support all connected phone lines. This can be done easily as the PSTN module has been implemented using the Microsoft Telephony API, which provides built-in functionality to handle multiple lines.

Implementation notes:

1) In our current implementation, the Ground Switching Center (GSC) and the In Flight BTAG reside on the same PC. The air-to-ground links are proprietary and hence inaccessible. Therefore, this link has been collapsed and the BTAG and the GSC have been implemented as two applications on the same PC as shown in Figure 5.
2) We have implemented an in-flight Bluetooth network consisting of one BTAG and one Bluetooth enabled mobile phone.

2.4 Software

The software for our system has been divided into the following modules:

2.4.1 Bluetooth Module

The Bluetooth module is the main software program from which other modules are invoked. The functions of this module are:

1) Automatic connection establishment and maintenance
2) Sequential invoking of various modules required for the integrated system operation
3) Voice transfer over ACL Bluetooth link

The Bluetooth module on the BTAG continuously scans the environment for Bluetooth enabled devices. All Bluetooth devices that come in range of the BTAG will capture one or more of the INQUIRY messages being broadcast by the BTAG and may reply to it. This module handles the replies sequentially and learns the Bluetooth device address of every device that replies. An Asynchronous Connectionless (ACL) link is then established with the devices that reply. Service Discovery (SDP) is used to determine whether the device is a mobile handset and if so whether it

wishes to avail the SkyMobile service. A two-way authentication procedure is then started to accomplish the following:

1) Enable the mobile to check that the BTAG is a genuine device authorized to instruct it
2) Enable the BTAG to determine that the mobile handset belongs to a passenger on flight
3) Allot a call forwarding number to the mobile handset

The Authentication Tool described later, is called by the Bluetooth module to perform this procedure. Once authentication is over, the Bluetooth Module invokes the GSM module. The GSM module implements call forwarding and then switches off the GSM stack . Thereafter, all communication is routed via the in-flight Bluetooth network, eliminating hazardous interference with ATC signals. The Bluetooth module thus establishes the various communication links as outlined in the system overview.

The Bluetooth module also handles the transmission of voice data across the Bluetooth network. At the BTAG-GSC end the Bluetooth module interacts with the PSTN module to playout received voice data on the phone line and acquire data to be transmitted from the PSTN line. At the Mobile Unit, the Bluetooth module interacts with the microphone and the speaker through the Voice Tool, explained in section 2.5, to record and playout voice data. The Bluetooth modules at both ends spawn the independent recorder application of the Voice Tool which provides the voice data to be transmitted. The Bluetooth module accesses this voice data through the Recorder process of the Voice Tool.

The module also transmits and receives voice packets over an asynchronous Bluetooth link. Even though the Synchronous Connection Oriented (SCO) link is prescribed for transmission of

audio data over Bluetooth, the ACL link was used due to lack of support for the SCO in the provided Ericsson's API. We feel that transmission over an asynchronous link serve as ample demonstration of our concept. The SCO link cab be incorporated given adequate support for SCO in the Bluetooth Application Programming Interface (API), using software modifications.

At the application layer, we use the RFCOMM protocol for data exchange. The RFCOMM link is set up using the Stack Connection Manager (SCM) interface. Both these protocols are available as a part of the software stack provided by Ericsson [7]. Packets of size 80 bytes are presented to the RFCOMM layer. The DH5 packet format has been employed for baseband transmission . The DH5 packet is a multislot packet and provides a data rate of 433Kbit/s and carries a payload of 341bytes [5]. A 16 bit CRC error correcting code is available in this packet type. We use packet retransmission to improve the reliability of data transfer. Each packet is uniquely identified by its sequence number. Retransmission has been implemented by keeping track of untransmitted packets, which the kit identifies by a message. A maximum of five retransmission attempts are made for each packet. It has been experimentally found that this is sufficient to ensure negligible packet loss. Also, the maximum number of retransmissions is constrained by the time required to deliver each packet, which is limited for acceptable voice conversation, and the associated overheads in terms of signaling between the part of the stack on the kit and the part of the stack in software. No extra error protection has been added at the application layer and the error correction capabilities provided by the Bluetooth baseband are relied upon. The voice packet received by the receiver is appended to a suitable header and presented to the voice playout tool.

The Bluetooth module thus provides a framework in which the various tasks are carried out in sequence with appropriate interfacing between the components.

2.4.2 The GSM Module

The GSM module is invoked by the Bluetooth module when the GSM modem has to be instructed as described in the system operation. The GSM module first initializes communication with the GSM modem through the serial port. This initialization is performed by the GSM-Connect TOOL explained later. The module instructs the GSM modem by sending GSM 7.07/7.05 AT commands [8] in ASCII format across the RS-232 serial interface. The various tasks performed by this module and the corresponding GSM-AT commands are described in Table 2.

Design Tradeoffs: The GSM module is driven by events occurring in the Bluetooth module. For every event, it passes on a series of commands to the GSM modem. This can be done in two ways:

1) *Transparent mode*: The software unit simply passes GSM-AT commands received from the BTAG.
2) *Local mode*: The software upon receipt of a request such as *Switch Off* generates the corresponding GSM-AT command string locally and then instructs the modem.

The transparent mode requires very low processing at the Mobile Unit, which is in keeping with practical constraints of low processing power on a mobile device. Therefore, we have chosen the transparent mode for our implementation. However, the local mode would have had the advantage of the BTAG being insulated from variations in mobile telephony standards.

TASK	AT COMMAND USED
GSM Stack operations	
GSM stack: Off : On	AT+CFUN=0 AT+CFUN=1
Call Forwarding	
Setup- Unconditional, for all classes (voice, data, SMS, fax), to the number specified	AT+CCFC=0,3, "forwarding number", 129/145, 7
Disable	AT+CCFC=0,4
Authentication Operations	
Select phonebook Search for entry with tag GENKEY Response of GSM modem – returns the generic key and the index number at which it is stored Response of GSM modem – Entry not found	AT+CPBS="SM" AT+CPBF="GENKEY" +CPBF=index number,"generic key", 129/145, "GENKEY" +CME ERROR: 22
Erase the authentication key	AT+CPBW=index number
Obtain phone number of mobile phone	AT+CNUM

Table 2: GSM-AT Command Set

2.4.3 PSTN module

The Bluetooth module executes in synchronization with the PSTN module at the BTAG-GSC end, as mentioned in the system operation. The PSTN module has been developed to enable the handling of calls arriving at the landline forwarding number assigned to each user's mobile at the GSC. The module provides the following features:

1) Accepting call from a landline caller and establishing a connection on the PSTN
2) Signaling to the BTAG to indicate call arrival
3) Streaming voice obtained from the BTAG over the PSTN connection and vice versa.
4) Directing a call from the Mobile Unit to a phone number on the ground

We chose the Microsoft Telephony API (TAPI) to implement this functionality. Traditionally, applications that want to use a modem for data communication access its features by issuing a series of standardized AT commands. However, the command sets for voice communication are

yet to be standardized and modems use one of the AT+V or AT#V voice command sets. The Microsoft TAPI provides a higher level abstraction for telephone lines and thus, insulates applications from the given modem's voice command set.

The PSTN module begins by initializing the phone line and setting it up for operation in an *automated answering mode* - in which the computer answers calls arriving on the phone line. The initialization procedure for a line device initializes every phone line attached to the system and associates wave device identifiers (one identifier for wave input and one identifier for wave output) with each line. These line device identifiers can be used as wave audio device identifiers to play or record sound in the Windows Wave API.

After initialization, the PSTN module operates through interrupts corresponding to status changes on the telephone line. We achieve the required functionality in the following manner:

1) *Initializing the line:* Initialization consists of four steps:
    i) Opening a logical line device
    ii) Negotiating the TAPI version to use
    iii) Getting the line device capabilities
    iv) Selecting the first line device that provides automated answering capability
    v) The line device is opened in owner mode (defined by TAPI [6]), the input and output wave device identifiers for the line are obtained and the module configures the line to receive all possible status messages

2) *Accepting a call*: When the device moves from IDLE state to RINGING state, the module waits for a fixed number of rings before taking the line off the hook and then answers the call. This action places the line in the ACCEPTED state, after which it goes into a CONNECTED state. As soon as the line goes into a CONNECTED state, we play out a message on the line to indicate to the calling party that we are in the process of establishing connection with the BTAG.

3) *Signaling to Bluetooth Airplane Gateway (BTAG):* As soon as the call is answered (line is placed off the hook), we send a signal to the BTAG indicating call arrival.

4) *Voice Streaming:* Voice streaming uses the wave device identifiers (for the line) and the Voice Tool functions. We have chosen CCITT-mu-law, 8 kHz, single channel encoding for voice data since phone lines support this format. The voice data captured from the PSTN line is transferred to the BTAG and vice versa. Voice is played out and recorded using the Voice Tool on the output and input wave device identifiers respectively, of the telephone line (obtained during line initialization). The Voice Tool provides non blocking playback and recording to support duplex voice. Thus, the PSTN module makes the PSTN line appear as a set of wave audio devices to the Bluetooth module.

5) *Calling a number:* Given a phone number to be called, the PSTN module handles dialing and call setup using TAPI functions. A list of core TAPI functions used in the PSTN module are detailed in Table 3.

TAPI Function	Description
LineInitialize	initializes the application's use of TAPI (tapi.dll) and the TAPI's line abstraction
LineGetDevCaps	queries a specified line device to determine its telephony capabilities
lineOpen	opens the line device specified by its device identifier and returns a handle to the line
LineGetID	gets a device identifier for the selected line or call
LineMakeCall	places a call to the given number
LineSetStatusMessages	enables an application to select which messages to receive for events related to status changes on the line
LineAnswer	answers the specified offering call
LineShutdown	shuts down the application's use of the line abstraction of the TAPI
LineClose	closes the specified open line device
LineGetCallStatus	returns the current status of the specified call

Table 3: TAPI Functions used by the PSTN module

Development Platform: The software has been developed in Microsoft Visual C++ programming environment, running on Windows 2000 operating system. Visual C++ has been chosen because it provides an easy interface for accessing various hardware components on a PC - the sound card and universal serial bus (USB) in our case. The API to the Bluetooth kit was also provided in VC++ and hence accurate testing as well as comparison with example applications was possible. Bluetooth functionalities have been developed using the stack and the API provided by Ericsson [7].

2.5 Tools

Three key tools have been developed to enable the handling of various software and hardware components. These are described below.

2.5.1 Authentication Tool

The tool has been developed to provide an authentication mechanism for a passenger boarding the plane. It enables the passenger to ascertain the validity of the Bluetooth port contacting his

mobile for data exchange and issuing GSM commands. This objective has to be accomplished without active user interaction. A generic format for packet exchange between the BTAG and the passenger's Bluetooth enabled mobile phone has been developed.

Scheme: Once the passenger is seated and a link has been established to the BTAG, the authentication procedure ensues. This involves a series of packet exchanges between the two Bluetooth ports. Each packet has a 5-byte header containing information about the packet. Every time a packet is exchanged, the header is stripped off and the contents of the packet are processed based on the header type.

Packet Contents	Packet Header Type
Generic Key; common to all passengers	GENKY
GSM Phone Number of passenger	PHNUM
Authentication Key; unique to passenger, Could also contain a GSM Command	AUKEY

Table 4: Packets used in authentication.

The generic key and authentication key are given to each passenger at the time of purchase of the ticket. The procedures for authentication, call redirection and GSM shut down are depicted in Figure 6.

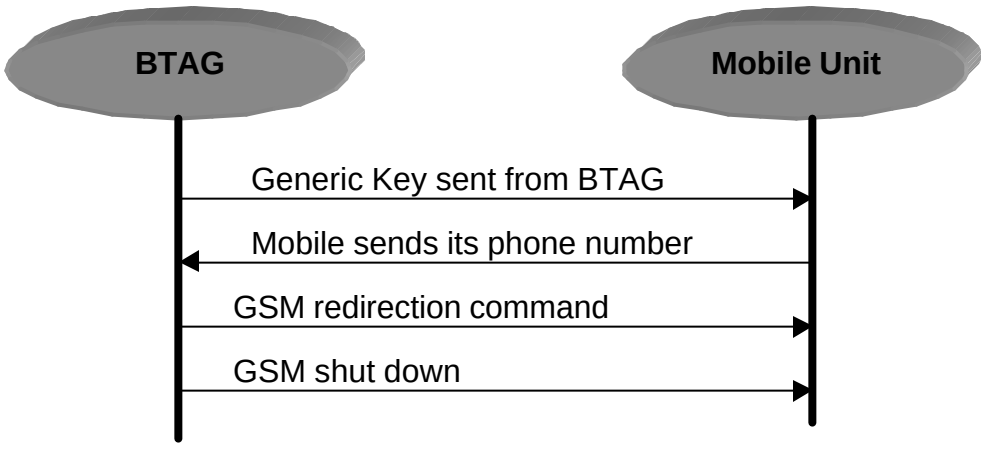

Figure 6: Authentication message flow

At the time of alighting, a similar procedure is followed for restoring the GSM network connection to the passenger's mobile phone. This is done by issuing commands to restart the GSM stack and disable call forwarding. The generic key and authentication key are then declared as invalid and erased from the memory of the Mobile Unit.

Security Issues: The scheme developed is secure against hostile attacks. Data exchange between the two Bluetooth ports is through a proprietary packet format, which acts as the first level of security. The generic key and authentication key, which only the BTAG and passengers know, act as a second tier of security. The contents of the packet are processed only when a matching key is received. This prevents the passenger's phone from revealing its phone number to any arbitrary Bluetooth port or executing GSM commands issued by an unauthenticated Bluetooth port. These keys are generated using algorithms known only to the concerned airline authorities. Finally, the keys expire as soon as the passenger alights, thereby ruling out misuse of the key after the flight.

2.5.2 GSM-Connect Tool

The GSM-Connect tool provides a simple interface, through which a generic application will be able to execute commands on the GSM modem. The GSM modem communicates with the PC through an RS-232 serial interface. The GSM-Connect Tool configures the serial interface and establishes a connection over it, using the MFC Comm Port utilities. The tool opens a connection through a *handle* to the relevant serial port. The settings of the port are retrieved into a *Data Control Block* structure. The *Data Control Block* is

Baud Rate	9600 bits/s
Byte Size	8 bits
Stop Bits	1 bit
Parity	None
Flow Control	None

Table 5: Connection parameters

then modified to configure the interface in accordance with the parameters specified in Table 5. This modified *Data Control Block* is applied to the port, thus creating a channel for communication with the GSM modem. This link can be used to issue GSM-AT commands to the modem. The GSM stack on the modem executes the command and provides an appropriate response. This response may be trapped as a string of characters and utilized for deciding further actions [4]. The operation of this tool is summarized in Figure 7. Typically, the GSM-Connect tool is provided a message string (containing a command) by the Bluetooth module after key matching. The command is streamed to the GSM modem by the tool and then executed on the phone by the stack. A typical example is that of call forwarding after authentication. In this case, the message *"AT+CCFC = phone number"* is passed from the BTAG to the mobile unit. The GSM tool forwards this message to the GSM network, which sets up call forwarding to the specified number.

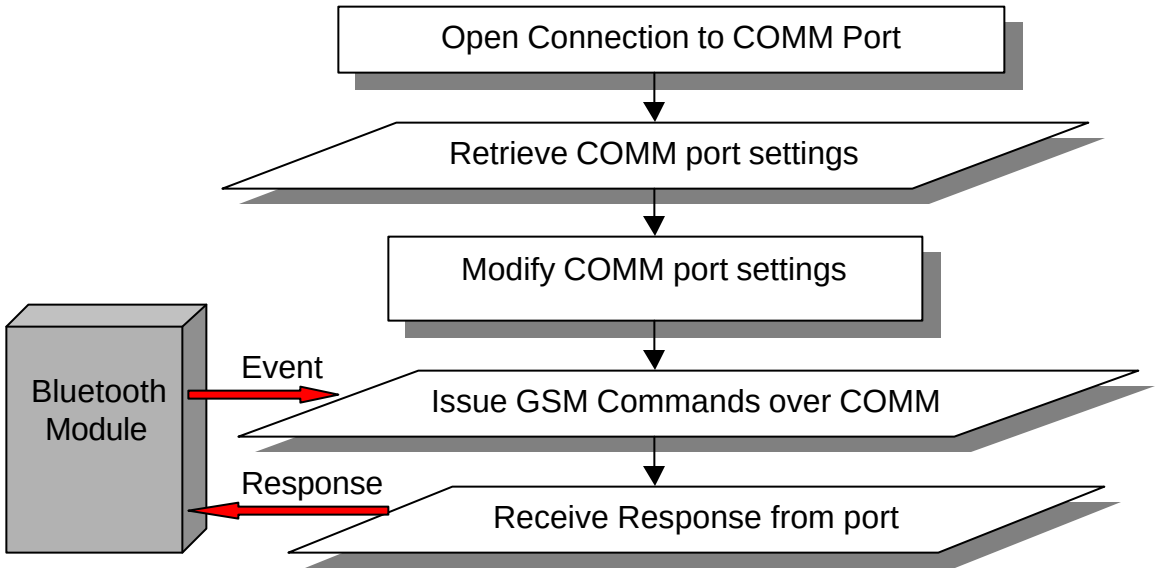

Figure 7: GSM Tool flow chart.

2.5.3 Voice Tool

This tool provides facilities for the recording and playout of voice. This tool is used by the Bluetooth module at both the BTAG-GSC and the Mobile Unit. The Voice Tool consists of a **Recorder** and a **Player**. The Player is executed from within the Bluetooth module, providing voice output on the speaker at the Mobile unit and presenting voice data to the PSTN module at the BTAG-GSC. The Recorder executes as a separate application. It captures voice from the microphone and passes it to the Bluetooth module at the Mobile Unit. At the BTAG-GSC end the Voice Tool receives voice from the PSTN module and presents it to the Bluetooth module for transmission.

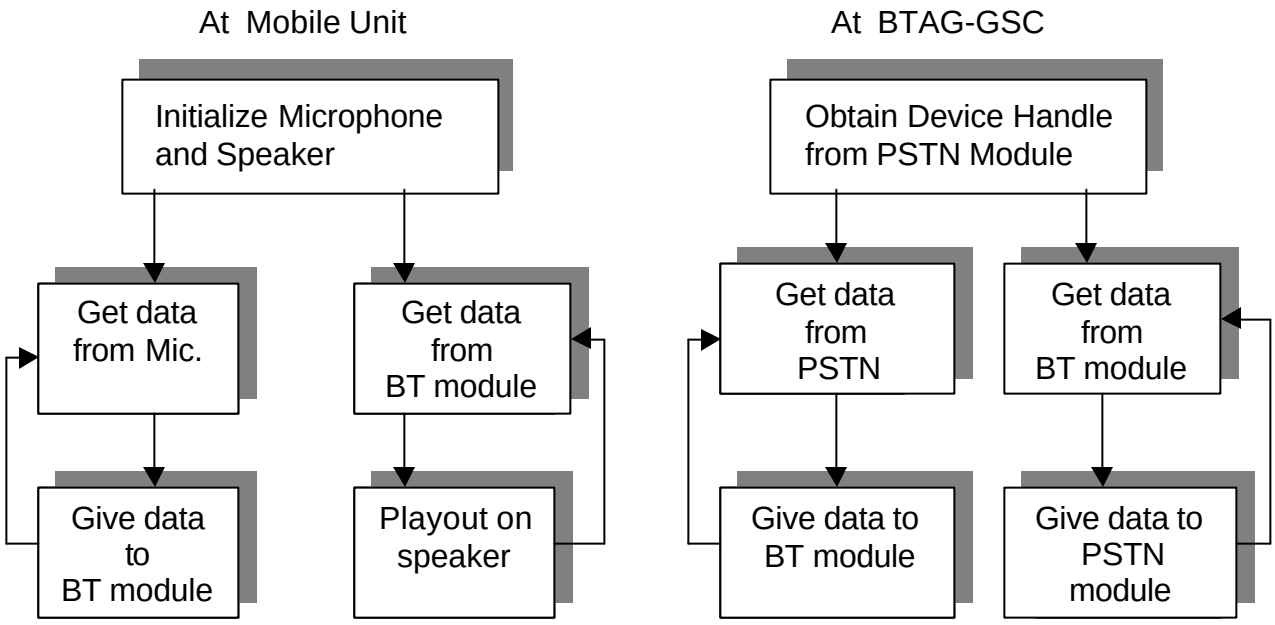

Figure 8: Voice Tool Flow Diagram

The voice recording and playout were built using the Visual C++ library *mmsystem.h*. The voice toolbox operation is presented in Figure 8. Voice is recorded at a sampling rate of 8000 Hz, mono channel, and stored in CCCIT-mu law format using 8 bits/sample. The mu-law format was chosen with regards to the capabilities of the PSTN modem, an intended playout device for the transmitted voice. A sampling rate of 8 kHz is standard for toll quality voice. The Player appends a 44-byte wave file header to the mu-law data received, before presenting it to the modem or to the speaker for playout. A list of VC++ functions used in the Voice Tool are detailed in Table 6.

Visual C++ Function	Function
WaveInOpen	Opens waveform audio input device for recording
WaveInPrepareHeader	Prepares a buffer for waveform audio input
WaveInAddBuffer	Sends an input buffer to waveform audio input device
WaveInStart	Starts input on waveform audio device
WaveInReset	Stops input on waveform audio device, resets current position to 0 and returns pending buffers to application
WaveInClose	Closes waveform audio input device
WaveOutOpen	Opens waveform audio device for playback
WaveOutPrepareHeader	Prepares a waveform audio data block for playback
WaveOutReset	Stops playback on waveform audio device , resets current position to 0 and returns pending buffers to the application
WaveOutWrite	Sends data block to waveform audio output device
WaveOutClose	Closes waveform audio output device

Table 6. Visual C++ functions used by the Voice Tool

3. Testing

3.1 Bluetooth Module

The Bluetooth module was tested in a phased manner:

1) *Data transfer:* Initially, we developed a file transfer application to familiarize ourselves with the API and to test the data transfer capability of the Bluetooth link. The data transfer proved extremely reliable with zero bit error rate. At this stage, we used Ericsson's sample application BTChatSecurity [7] for connection establishment.
2) *Connection establishment*: A utility was developed to automatically establish connection between two Bluetooth devices. The connection starting time from device discovery to the establishment of an RFCOMM channel was 8 to 11 seconds.
3) *Authentication*: The Authentication tool was tested in conjunction with the Bluetooth module. The scheme outlined in Figure 7 was verified by transmitting both correct as well as incorrect keys.
4) *Voice Tool*: We tested the recorder by storing voice to a file in the PCM 8 bit 8 kHz, mono channel format. The file was played out through MATLAB to check the output quality. The recorded voice quality was acceptable, as compared to the PSTN quality in India. We also tested the player and validated its performance in a similar fashion.
5) *Retransmissions:* We recorded the number of packets retransmitted while transferring voice data over ACL link. Average number of retransmissions per application layer packet was 0.12 packets, measured in groups of 1000 packets.
6) We then tested voice transfer across Bluetooth by combining tests 1,2 and 4. The voice quality was comparable to toll quality.

Problems Encountered:

1) The device discovery procedure is unreliable if short inquiry time is used. Devices are discovered reliably for an inquiry time of 10 seconds, while for short inquiry times (4-5 seconds), more than one attempt is sometimes required. As a result, our device discovery time is as high as 10 seconds depending on the number of discovery attempts.
2) Service discovery is erratic in nature and multiple attempts are usually required to obtain the service handles from the other device.
3) **The SCO is not provided in the provided API**. The API provides a function to request establishment of the SCO connection in the SCM part of the stack, but there is no function to transfer data over this connection. We worked around this problem by developing our voice streaming application over the ACL link.

3.2 GSM Module

1) The GSM modem hardware was tested utilizing *Hilgreave's Hyperterminal* software to establish a connection to the modem. The commands tested included making a call, retrieving network information and instructing the network to change various parameters of the subscriber's account. The commands to be implemented in the prototype, specifically call forwarding and stack shutdown, were tested a number of times to confirm their accurate execution.
2) After the development of the GSM-Connect tool, the same commands as in 1 above were tested through the tool. A number of other commands, including making a call, were tested.

3) The GSM module was then merged with the Authentication tool and the Bluetooth module to test authenticated call forwarding and GSM switch-off.

3.3 PSTN Module

The PSTN module was tested over the IIT-Bombay internal PABX. The following functions were checked:

1) *Call reception*: A call was placed to the phone modem on which connected to our GSC. The PSTN module took the phone off the hook and played a welcome message on the line. This pre-recorded message was reproduced with acceptable voice quality.
2) *Voice recording*: The PSTN module was extended to record the incoming voice data. After the PSTN module took the phone off the hook, the caller spoke over the line for 60 seconds. The message was recorded by the PSTN module as a .wav file.
3) *Making a call:* The PSTN module made a call to a phone number given to it. After the called party picked up the handset, we spoke into the microphone of the PC. This voice data was recorded (using the Voice Tool) in real time and transmitted over the phone line. The voice was reproduced without perceptible degradation at the receiver end.

Problems encountered:

On certain occasions, when the party at the other end of the phone line disconnected the call, the PSTN module refused to hang up and left the line in a connected state. In this case, further calls to the Ground Switching Center got an engaged tone from the phone line. This was because the TAPI was not getting a hangup signal from the phone line. This problem was not encountered

when we used a regular phone line provided by the local PSTN service provider (MTNL). This leads us to believe that the problem is due to the IIT-Bombay PABX.

After the modules were successfully tested, they were integrated into the complete system and the overall operation was tested. This is available for demonstration.

4. Implementability and Marketability

The SkyMobile system can be easily deployed in commercial airplane fleets with very little infrastructural modification. The core requirements of the system are an air-to-ground link and Bluetooth enabled mobile phones. The Bluetooth access points required to extend the BTAG will be small in size, light weight and low cost. This would facilitate their easy installation in aircrafts. The BTAG itself can be implemented on an inexpensive PC. The different modules developed use only off the shelf hardware components. Some other technologies have been proposed for providing phone connectivity in airplanes. Most of them allow the passengers to only make calls but our solution also allows the passenger to receive calls on her mobile, without having to change her phone number. Other solutions require the user to use a different mobile handset rather than the one that would be usually used on land. Our solution, allows the same handset to be used in air. The solution developed leads to a marketable product, as both the technology for its implementation and the demand for the service exist. Boeing for instance, have set up extensive air-to-ground communication links to support data transfer, which can be used in our solution. Some companies also provide reception of calls on fixed handsets while others permit the making of calls with specific hardware. SkyMobile, on the other hand, provides a user seamless connectivity on her own mobile, and is thus a thus a unique technology.

5. Summary

The SkyMobile system integrates disparate communication networks to provide the user with seamless connectivity on her usual mobile while traveling by air. The prototype developed by us has been able to successfully integrate the GSM, PSTN and Bluetooth networks to achieve the specified design objectives. Seamless switchover from GSM to Bluetooth, GSM call reroute to a preassigned PSTN number, and voice communication over Bluetooth have been demonstrated. The system could be further modified to make it more robust and eliminate some of our design compromises. A few such areas for improvement are:

1) The ACL link used for voice communication should be replaced by an SCO link, which is prescribed in the Bluetooth Specifications [5] for transfer of synchronous data.
2) The airplane BTAG should be extended to a scatternet with multiple users and Bluetooth access points in the airplane, to allow several active calls at any given time.
3) The Bluetooth application currently demonstrated on the laptop needs to be ported to a Bluetooth enabled GSM phone.

Further, value added services offered by the GSM network, such as Fax, SMS may be emulated and telephone SS7 signaling over Bluetooth may be incorporated.

Our solution benefits the users by enhancing the safety of airways. The solution also provides automatic switch-over of mobile phones from the cellular network to Bluetooth, while boarding a plane. It also enables convenient connectivity while air-borne. The user does not perceive any change in the services provided by the mobile regardless of whether she is on land or in air. The solution is scalable and easily adaptable to varied usage models. Thus, the system with its unique features promises to be very useful to both passengers and airlines.

A. References

1) CFR Title 47, Part 22, Subpart H, Section 22.925, “Cellular Radio Service – Prohibition on airborne operation of cellular telephones”; FAA Advisory Circular 91.21-1, “Use of Portable Electronic Devices Aboard Aircraft.”: www.fcc.gov
2) Connexion: www.mobilecomms-technology.com/projects/connexion/
3) Globalstar, Qualcomm alliance: in-flight broadband access: www.qualcomm.com/globalstar/bp/news/
4) Wavecom WM0D2, GSM modem specification: www.wavecom.com/showroom/specification/wm0d2.html
5) Bluetooth Core and Profiles specifications, v1.0b http://www.bluetooth.com/developer/specification/specification.asp
6) Microsoft Developer Network, www.msdn.microsoft.com
7) Bluetooth PC Reference Stack by Ericcson: User’s Manual.
8) WM2A GSM Module Specifications driven by AT commands: WISMO documentation.

B. Glossary

Terms and abbreviations specific to our prototype:

BTAG	Bluetooth Airplane Gateway
GSC	Ground Switching Center
MU	Mobile Unit, acts as Bluetooth enabled cellular phone in our prototype
seamless connectivity	connectivity to the user's cellular phone in-flight without user initiation or change of user handset hardware

Other Abbreviations used:

ACL	Asynchronous Connectionless
ATC	Air Traffic Control
BT	Bluetooth
CDMA/IS-95	Cellular system used in the United States.
GSM	Global system for Mobile Communication: used in India, Europe parts of USA
SMS	Short Messaging Service
MFC	Microsoft Foundation Classes
MTNL	Mahanagar Telephone Nigam Limited, Mumbai's local PSTN service.
PABX	Private Automatic Branch Exchange
PC	Personal Computer, acts as combined BTAG and GSC in our prototype
PSTN	Public Switched Telephone Network
SCM	Stack Connection Manager
SCO	Synchronous Connection-Oriented
SDP	Service Discovery Protocol
TAPI	Telephony Application Programmers Interface
USB	Universal Serial Bus