Bluetooth Primer

Author: Aman Kansal

Contents

Contents **2**
Introduction - What is Bluetooth? **4**
Motivation for Bluetooth 4
Bluetooth History 5
System Challenges 5
System Requirements 5
Technical Challenges 6
The Basic Bluetooth System Architecture **7**
Overview of the Protocol Stack 7
Radio layer 7
Baseband 8
Link Manager Protocol 8
Logical Link Control and Adaptation Protocol 9
Host Controller Interface 9
Application layer 10
Establishing a connection in Bluetooth **11**
Inquiry and Paging 12
Inquiry 12
Inquiry Scan 13
Inquiry Response 13
Page 13
Page Scan 14
Page Response 14
Active mode 14
Sniff Mode 15
Hold Mode 15
Park Mode 15
Link Establishment 15
SDP 15
L2CAP link 16
Security 17
Application link 18

Bluetooth Security **19**
The basic structure 19
Detailed Description 19
Generation of the link key 20
Generation of the encryption key 21
Why is Blutooth the way it is? **23**
Ad hoc connectivity 23
The choice of the radio spectrum. 24
Medium access control 24
Connection Establishment. 24
Medium Access Control and channel allocation 24
Service prioritization 25
Interference 25
Modulation Scheme Used 26
Protection of Data 26
Power Consumption 26
Design of the higher layers 27
The Host Controller Interface. 27
Conclusions 27
FAQs **28**

Introduction - What is Bluetooth?

Bluetooth is the name given to a technology standard using short-range radio links, intended to replace the cable(s) connecting portable and/or fixed electronic devices. The standard defines a uniform structure for a wide range of devices to communicate with each other, with minimal user effort. Its key features are robustness, low complexity, low power and low cost, which make it especially suited to mobile handheld devices. The technology also offers wireless access to LANs, PSTN, the mobile phone network and the Internet for a host of home appliances and portable handheld interfaces.

The standard has achieved global acceptance such that any Bluetooth device, anywhere in the world, can connect to other Bluetooth devices in its proximity, regardless of brand. Bluetooth enabled electronic devices connect and communicate wirelessly via short-range, ad hoc networks called piconets. Each unit can simultaneously communicate with up to seven other units per piconet. Moreover, each unit can simultaneously belong to several piconets. These piconets are established dynamically and automatically as Bluetooth devices enter and leave the radio proximity.

Motivation for Bluetooth

The motivations for Bluetooth come from both technology push and market pull. The ability to pack ever more transistors on a small area of silicon has made small embedded devices capable of running complex protocols. Embedded controllers in devices are now capable of being programmed, controlled and used in various smart ways. Thus intelligent devices can now be embedded into the user's work and home areas. Various techniques are available to connect these embedded devices to the Internet, thus forming the so-called "embedded Internet". Significant progress has been made in developing small and cheap sensors that can pick up useful signals from the user environment without user interaction or explicit command. New kinds of electronic tags, JINI and Piano (which can be built on top of Bluetooth units, and specifies what sort of information they exchange and how they communicate) that enable interaction among a variety of devices have become available. This has opened up the possibility for creating a "ubiquitous computing" environment. In this environment, the devices are controlled and activated by a combination of intelligent systems and strategically located sensors that work without explicit user support. The facility to automate depends heavily on the ability of devices to communicate wirelessly with each other, with more intelligent central servers, information repositories, sensors and actuators. Bluetooth can provide a solution to this requirement.

The immediate need for Bluetooth as mentioned earlier came from the desire to connect peripherals and devices without cables. The available technology-IrDA OBEX is based in infrared links that are limited to line of sight connections. Bluetooth is further fuelled by the demand for mobile and wireless access to LANs, Internet over mobile and other existing networks, where the backbone is wired but the interface is free to move. This not only makes the network easier to use but also extends its reach. The advantages and rapid proliferation of LANs suggest that setting up personal area networks, that is, connections among devices in the proximity of the user, will have many beneficial uses. Bluetooth could also be used in home networking applications. With increasing numbers of homes having multiple PCs, the need for networks that are simple to install and maintain, is growing. There is also the commercial need to provide "information push" capabilities, which is important for handhelds and other such mobile devices and this has been partially incorporated in Bluetooth. Bluetooth's main strength is its ability to simultaneously handle both data and voice transmissions, allowing such innovative solutions as a mobile hands-free headset for voice calls, print to fax capability, and automatically synchronizing PDA, laptop, and cell phone address book applications.

These uses suggest that a technology like Bluetooth is extremely useful and will have a significant effect on the way information is accessed and used.

Bluetooth History

Bluetooth was invented in 1994 by L. M. Ericsson of Sweden. The standard is named after Harald Blaatand "Bluetooth" II, king of Denmark 940-981A.D. A runic stone has been erected in his capitol city Jelling(Jutland) that depicts the chivalry of Harald and the "runes" say:

Harald christenized the Danes.

Harald controlled Denmark and Norway.

Harald thinks notebooks and cellular phones should seamlessly communicate. (!)

The Bluetooth Special Interest Group (SIG) was founded by Ericsson, IBM, Intel, Nokia and Toshiba in February 1998, to develop an open specification for short-range wireless connectivity. The group is now also promoted by 3COM, Microsoft, Agere Systems (formerly Lucent) and Motorola. More than 2500 companies have joined the SIG.

The following section describes some of the requirements from the Bluetooth system and in essence, suggests the functionalities planned for it.

System Challenges

Although, originally conceived to enable the design of universal wireless connections for laptops, computers and cellular telephones, it quickly became apparent that there were many other applications for the Bluetooth standard. Thus, the Bluetooth standard not only tries to overcome the limitations of the wired networks but also offers a variety of other services and creates opportunities for new usage models.

System Requirements

The Bluetooth system is now recognized more than just a cable replacement technology. Various innovative usage models have opened up new areas where Bluetooth can be used. These also impose many requirements on the system, some of which are discussed below.

The most important requirement from the wireless link is that there should be a universal framework that offers means to access information across a diverse set of devices (for example, PDA's, laptops, PC's, mobile phones, home appliances etc.) in a seamless, user friendly and efficient manner.

In the practical scenario all devices are not expected to be capable of all functionalities and users too may expect their familiar devices to perform their basic functions in the usual way. So Bluetooth must offer the facility for collaboration between devices, in proximity of one another, where every device provides its inherent function based on its form, user interface, cost and power, but additional services emerge due to the synergy resulting out of the collaboration.

The standard must enable the devices to establish ad hoc connections. Also, introduced is the "unconscious connectivity" paradigm, where devices can connect to those in proximity almost without any user command or interaction. This shall allow utilization of various information recourses for the benefit of the user.

Support for both data and voice is expected, as these are the two most important kinds of information being transmitted over networks today. (The requirements of video and streaming multimedia are also being imposed on the future versions of Bluetooth).

The standard should be able to incorporate new usage models without requiring any registration of the new service with a central authority.

The communications should offer similar protection as in cables. There should not be any compromises on security in switching over to wireless.

The implementations of the standard should be simple, small and power efficient for easy mobile usage.

It is necessary for the rapid deployment of the system and for the Bluetooth benefits to actually reach the users that a large number of devices be enabled with the Bluetooth standard. The devices to be enabled comprise a highly non-uniform set and no single company can have the expertise to manufacture all these. For this and other reasons, the Bluetooth standard has been made royalty free and its worldwide acceptance should be facilitated.

Technical Challenges

The above requirements involve great technical complexity not only in terms of the functionalities to be provided but also in terms of the power and size requirements. The technology designed to meet the above requirements must face the following technical challenges:

The system has to use an unlicensed band for universal acceptance and usage. Thus the Industrial, Scientific and Medical (ISM) band has been selected for Bluetooth. The challenge here is to make the system robust to interference from other sources in this band, which include not only ISM band communication systems but also microwave ovens. Preferably, each transmitter itself should use the minimum power required so as not to increase the noise for other users.

The transceivers should be able to adapt to a rapidly changing environment, as the devices will usually be mobile. The well-known problems in wireless systems such as multi-path fading must be handled. Also, the connection establishment and routing protocols have to operate in an environment where the number, location and variety of Bluetooth devices will change dynamically with fair amount of rapidity.

The size of the implementation should be small for easy integration into handheld and mobile devices.

The power consumption should not be more than a small fraction of the host device into which the Bluetooth capability is to be introduced.

The technology should be adaptable to devices of varying computing power and memory recourses. This will ensure that more and more devices can inter operate.

Automatic and unconscious connection establishment must be provided. The number and identity of devices in proximity will change quite frequently and it will be very inconvenient to establish connections manually each time. Also, the number of devices will be too large for most users to be able to remember or search the device address of the device they need to connect to.

Synchronization of clocks among the communicating units will have to be achieved. As each unit will have its own free running clock with its own drift, carrying out successful communication, especially CDMA, is a challenge in itself.

Security considerations have to be satisfied. The Bluetooth devices will be part of people’s personal usage and will contain and communicate their personal information, sensitive business information or other data that must be protected from being spoofed or mutilated. Encryption facility must thus be provided among other security features.

There are some existing wireless applications in personal and office area networks that provide services similar to the ones provided by Bluetooth, like the IrDA OBEX and HomeRF. These services though similar, have certain differentiating features that make them more suitable to certain classes of applications. Thus, to achieve complete integration, Bluetooth must provide means to inter operate with these other technologies in at least those services for which they are better suited. It will be advantageous for the standard to be amenable to existing higher layer protocols such as TCP-IP and WAP for speeding up development.

The products should provide a good out of the box experience, that is, they should provide high value with existing applications.

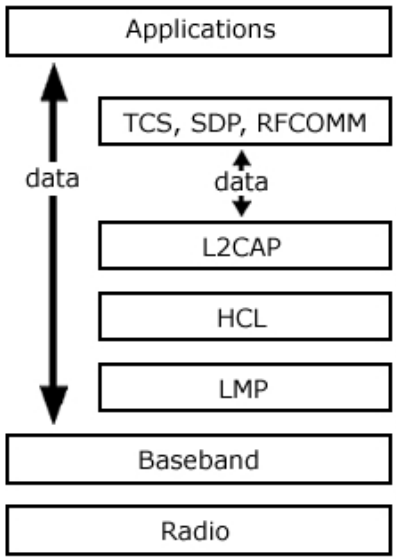
Bluetooth is a technology intended to meet all of the above demands. Thus, the standards body has tried to fix the specifications such that the above requirements can be met and provisions have been kept to facilitate the developers in meeting the technical challenges in their implementations. Some design choices are discussed in the section "Why is it the way it is?"

The Basic Bluetooth System Architecture

The system architecture for Bluetooth is briefly described here. The system design has been segmented into various almost independent layers for conceptual ease of description. These layers are described in detail in the core Bluetooth specifications. The design specifications also describe certain properties for certain common classes of applications to be implemented over Bluetooth to achieve uniformity across diverse manufacturers. These are described in profiles of the Bluetooth Specification. Both the core and profiles are available royalty free at the official Bluetooth website: www.bluetooth.com.

Overview of the Protocol Stack

The basic protocol stack consists is shown in Fig .1.



The figure shows that the protocol stack consists of a radio layer at the bottom that forms the physical connection interface. The baseband and Link Manager Protocol (LMP) that reside over it are basically meant to establish and control links between Bluetooth devices. These three bottom layers are typically implemented in hardware/firmware. The Host Controller layer is required to interface the Bluetooth hardware to the upper protocol-L2CAP (Logical Link Control and Adaptation Protocol). The host controller is required only when the L2CAP resides in software in the host. If the L2CAP is also on the Bluetooth module, this layer may not be required as then the L2CAP can directly communicate with the LMP and baseband. Applications reside above L2CAP. The following subsections give a brief description of each layer.

Figure 1: The Bluetooth Protocol Stack.

Radio layer

This link operates in the unlicensed ISM band around 2.4GHz and uses spread spectrum communication. The band extends from 2400 to 2483.5 MHz in a vast majority of countries and this whole range is utilized for optimizing the spectrum spreading. However, for some countries with a smaller ISM band a down-scaled version is also provided. For spread spectrum, the frequency hopping (FH) technique is used. As multiple uncoordinated networks may exist in this band and cause interference, fast FH and short data packets are used. This is because the error rate may be high, especially due to strong interference from microwave ovens that operate at this frequency. CVSD coding has been adopted for voice, which can withstand high bit error rates. Also the packet headers are protected by a highly redundant error correction scheme to make them robust to errors.

The frequency hops are fixed at 2402+k MHz, where k= 0,1,...,78. The nominal hop rate is 1600 hops per second. This gives a single hop slot of 625 microseconds. The modulation used is Gaussian pre-filtered Binary FSK. The Gaussian filter has BT=0.5. The transmitter power is fixed at 0dBm for a 10m range while it can be increased to 20dBm for a 100m range. Various restrictions are specified on clock accuracies and drift, spurious emissions and radio frequency tolerances.

Baseband

The baseband is the layer that controls the radio. The frequency hop sequences are provided by this layer. Baseband also takes care of lower level encryption for secure links. The packet handling over the wireless link is the responsibility of baseband. Two types of links can be established:

- **SCO:** Synchronous Connection Oriented. These links are meant for synchronous data- typically voice.
- **ACL:** Asynchronous Connection less. These links may be used for data transfer applications, which do not require a synchronous link.

The baseband provides the functionalities required for devices to synchronize their clocks and establish connections. Inquiry procedures for discovering the addresses of devices in proximity are also provided. Error correction for packets is provided depending on the type of packet. Various packet types are specified for some common applications, differing in their data capacity and error correction overheads. Five different channel types are provided for control information, link management information, user synchronous data, user asynchronous data and isosynchronous data. Data whitening is also carried out at this layer. The functions required for generating encryption keys and link keys are defined. A more detailed description of some of the baseband operations related to connection establishment is provided in the next section: *Establishing a connection in Bluetooth.*

Link Manager Protocol

The basic functions of LMP can be classified as:

- Piconet management
- Link configuration
- Security functions

A piconet is a group of devices connected to a common channel, which is identified with its unique hop sequence. One of the devices, usually the one which first initiated the connection is the master. Up to seven other devices can be actively connected to this master, and many more could be connected in a low power "parked" state. The devices on one piconet can communicate with each other over SCO or ACL links. The channel sharing is managed by the master, with the help of Link Managers on each device. Any two or more devices that need to communicate must establish a piconet among themselves. Devices can be part of many piconets at the same time (Fig 2).

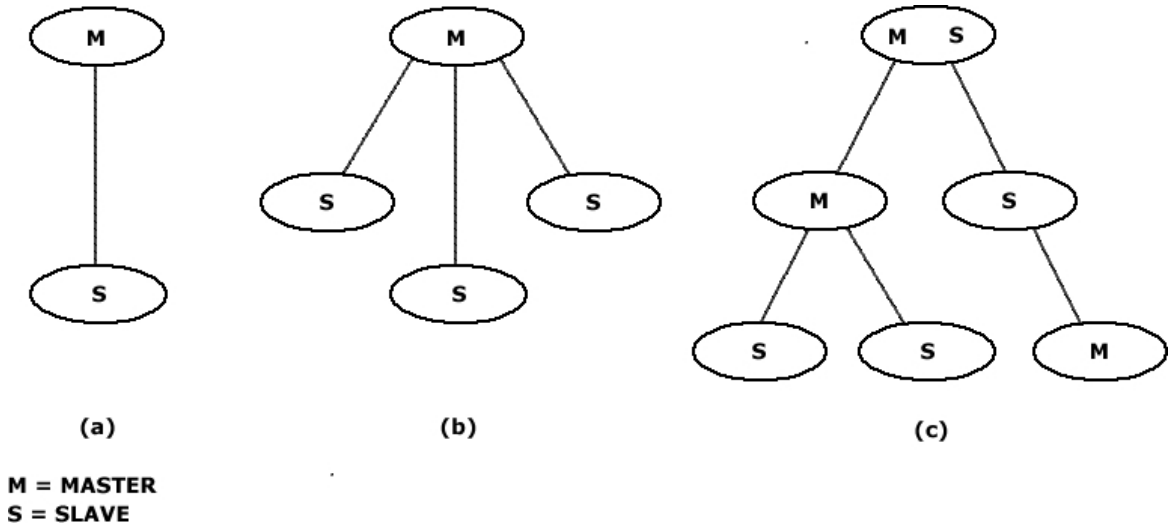


Figure 2: Piconets and a Scatternet.
a) A piconet between two devices,
b) A piconet between many devices
c) A scatternet, a combination of piconets with some devices common to the piconets.

The LMP provides the functionality to attach/detach slaves, switch roles between a master and a slave and to establish ACL/SCO links. LMP also handles the low power modes-hold, sniff and park, designed to save power when the device does not have data to send.

Link configuration tasks include setting link parameters, Quality of Service and power control if the device supports it. Authentication of devices to be linked and managing link keys is also taken care by LMP. The role of the LMP in link establishment is discussed in next section: Establishing a connection in Bluetooth.

Logical Link Control and Adaptation Protocol

This is the protocol with which most applications would interact unless a host controller is used. The basic functions of the L2CAP are:

Multiplexing	The protocol must allow multiple applications to use a link between two devices simultaneously.
Segmentation and Reassembly	The protocol must reduce the size of packets provided by applications to the size of packets accepted by baseband. L2CAP itself accepts packet sizes up to 64kb but the baseband packets can accept a payload of at most 2745 bits. The reverse procedure, that of combining the segmented packets in the proper order, has to be carried out for received packets.
Quality of Service	L2CAP allows applications to demand QoS on certain parameters like peak bandwidth, latency and delay variation. L2CAP checks if the link is capable of providing it and provides it if possible.

Basically, L2CAP provides the network layer functions to applications and higher protocols.

Host Controller Interface

The basic structure showing how the host controller layers are fitted into the protocol stack is shown in Fig.3.

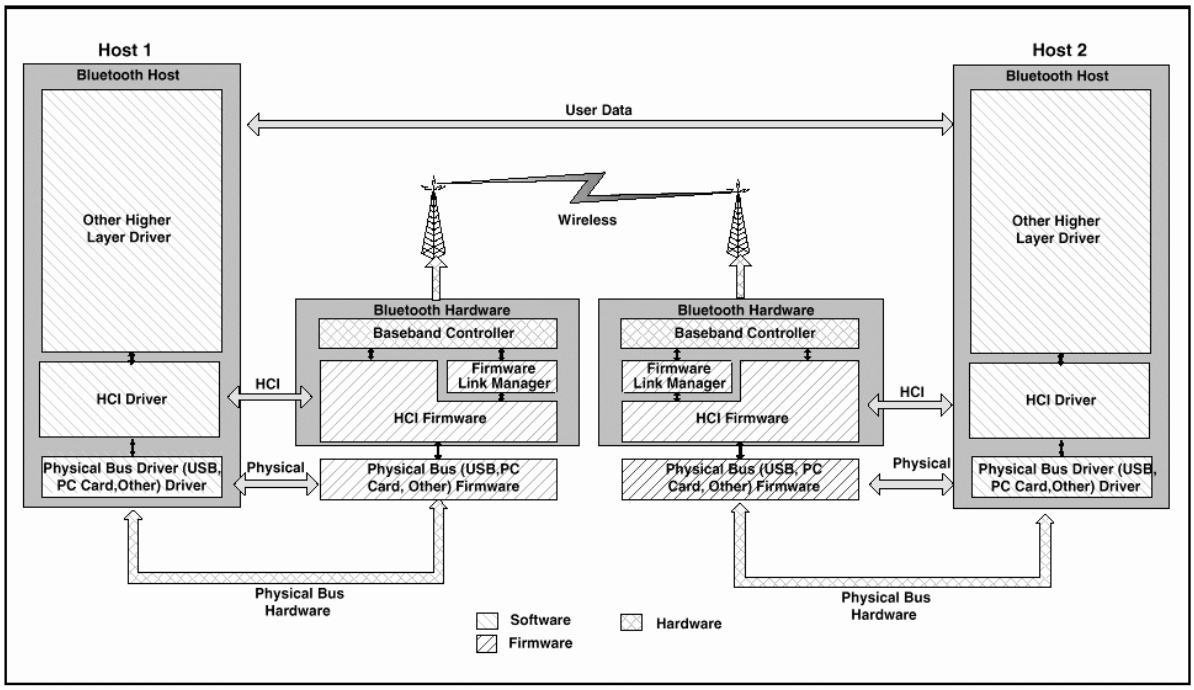


Figure 3: Host controller in the protocol stack.

For many devices, the Bluetooth enabling module may be added as a separate card, for instance, on a PC or a laptop, the Bluetooth hardware may be added as a PCI card or a USB adapter. Hardware modules usually implement the lower layers-radio, baseband and LMP. Then the data to be sent to LMP and baseband travels over the physical bus like USB. A driver for this bus is required on the "host", that is the PC, and a "host controller interface" is required on the Bluetooth hardware card to accept data over the physical bus. Thus, if the higher Bluetooth layers, L2CAP and above are in software and the lower ones in hardware, the following extra layers are at least required:

HCI driver	This is the driver for host controller interface. It resides in the host, above the physical bus, and formats the data to be accepted by the Host Controller on the Bluetooth hardware.
Host Controller Interface	This resides on the Bluetooth hardware and accepts communications over the physical bus.

Application layer

The L2CAP may be accessed directly by the applications or through certain support protocols like RFCOMM, TCS and SDP mentioned earlier. The applications may use other protocols like TCP-IP or WAP and Bluetooth allows these to inter operate. The applications may themselves run PPP(Point to Point protocol), FTP(file transfer protocol) or other specific protocols as required by the application. An application may use the SDP to discover whether the service it needs from a remote device is available. Many usage models have been proposed by several manufacturers. Some of these are:

***Three in one phone**:* A single handset works as an intercom in the office (no call charges), as a PSTN phone whenever an access point to the PSTN is available, and as a mobile otherwise.

The Briefcase Trick: The RF link does not need line of sight. So a mobile could connect to a laptop even while it is in the briefcase and allow access to its facilities like email.

The Automatic Synchronizer: Seamless connectivity between the user's PDA's, laptop and mobile will allow applications to automatically update and synchronize schedules and other data when modifications are made on one device.

***Wireless headsets*:** These will allow access to user's mobiles and audio services even while the devices are in the user's pocket. Thus hands free operation will be possible.

Car Kits: Hands free devices will allow users to access their phones without letting their hands off the steering wheel.

Apart from these, a large variety of other applications in home automation, data sharing during meetings without the use of extra equipment, testing factory devices over a wireless handheld while walking through, toddler alarms, security systems, network access at public places and hidden computing have been suggested, some of which have been successfully demonstrated.

Establishing a connection in Bluetooth

This section describes the basic procedures to be followed by two or more Bluetooth devices to start a connection between them. Consider the following scenario: A person walks in to a hotel lobby and wants to access her email over her Bluetooth enabled device, which could be a laptop or a Personal Digital Assistant. What would she have to do? Depending on the implementation., she would be clicking on a menu or an email application icon. The device would automatically carry out the following steps, (except perhaps for the authentication step if the device has come to the environment for the first time):

- **Inquiry:** The device on reaching a new environment would automatically initiated an inquiry to find out what access points are within its range. (If not, it'll do so when the email application asks for a link.) This will result in the following events:
  - all nearby access points respond with their addresses.
  - the device picks one out the responding devices.
- **Paging:** The device will invoke a baseband procedure called paging. This results in synchronization of the device with the access point, in terms of its clock offset and phase in the frequency hop, among other required initializations.
- **Link establishment:** The LMP will now establish a link with the access point. As the application in this case is email, an ACL link will be used. Various setup steps will be carried out as described below.
- **Service Discovery:** The LMP will use the SDP(Service Discovery Protocol) to discover what services are available from the access point, in particular whether email access or access to the relevant host is possible from this access point or not. Let us assume that the service is available, otherwise, the application cannot proceed further. The information regarding the other services offered at the access point may be presented to the user.
- **L2CAP channel:** With information obtained from SDP, the device will create an L2CAP channel to the access point. This may be directly used by the application or another protocol like RFCOMM may be run over it.
- **RFCOMM channel:** Depending on the need of the email application an RFCOMM or other channel(in case of other applications) will be created over the L2CAP channel. This feature allows existing applications developed for serial ports to run without modification over Bluetooth platforms.
- **Security:** If the access point restricts its access to a particular set of users or otherwise offers secure mode communications to people having some prior registration with it, then at this stage, the access point will send a security request for "pairing". This will be successful if the user knows the correct PIN code to access the service. Note that the PIN is not transmitted over the wireless channel but another key generated from it is used, so that the PIN is difficult to compromise. Encryption will be invoked if secure mode is used.
- **PPP:** Assuming that a PPP link is used over serial modem as in dial up networking, the same application will now be able to run PPP over RFCOMM(which emulates the serial port). This link will allow the user to login to his email account.
- **Network Protocols:** The network protocols like TCP/IP, IPX, Appletalk can now send and receive data over the link.

In the above procedure, user interaction is required only at the usual login for his email and additionally for the security to be implemented. The remaining steps are automatic. The above procedures now be described in detail to demonstrate the connection establishment process. The explanation of the above procedures requires a brief description the device clocks in Bluetooth.

Clock: Every Bluetooth unit has an internal system clock that determines the timing and hopping of the transceiver. The Bluetooth clock is derived from a free running native clock that is never adjusted and is never turned off. For synchronization with other units, only offsets are used that, added to the native clock, provide temporary Bluetooth clocks which are mutually synchronized. The Bluetooth clock has no relation to the time of day. The Bluetooth clock is very important for the Bluetooth transceiver as it is involved in timing a number of important events without which communication is not possible. Its resolution is at least half the TX or RX slot length, or 312.5 microseconds. The clock has a cycle of about a day. If the clock is implemented with a counter, a 28-bit counter is required that wraps around at 2^{28} -1. The LSB ticks in units of 312.5 microseconds, giving a clock rate of 3.2 kHz.

The timing and the frequency hopping on the channel of a piconet is determined by the Bluetooth clock of the master. When the piconet is established, the master clock is communicated to the slaves. The slaves store the required offset to be used while communicating with the particular master and use it to synchronize to the channel. As the local clock itself is not modified, different offsets can be used to participate in various piconets.

The minimum clock accuracy required is +/- 20ppm in active mode and +/-250ppm in low power states like Hold, Sniff, Standby and Park.

Inquiry and Paging

These are the initial steps in starting a connection. The device before, during and after these procedures can be viewed to be in different states shown in Fig. 4.

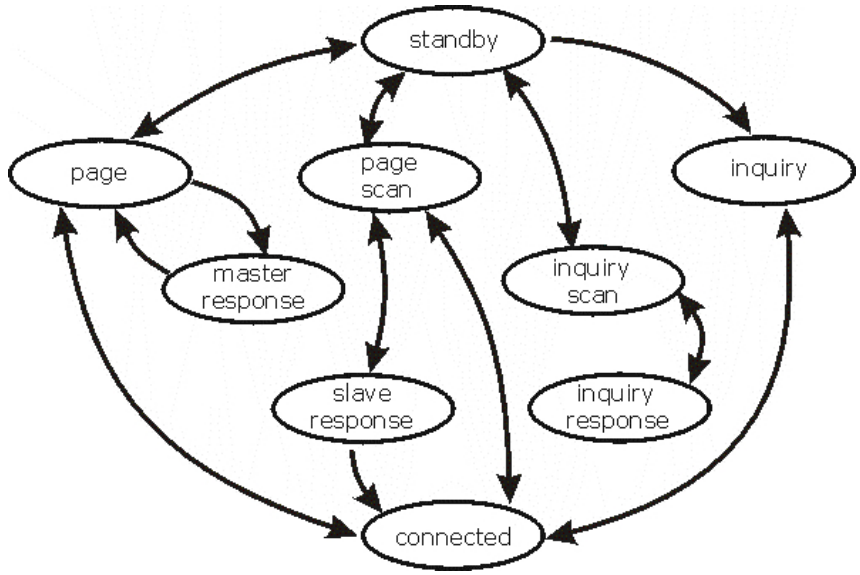


Figure 4: State diagram of the link controller.

The device is in **Standby** state by default. In this state only the native clock is running and power consumption is very low. It may leave this state to go to **Inquiry, Inquiry Scan, Page** or **Page Scan** states. These states are described below:

Inquiry

In this state, the device sends an Inquiry packet addressed to either the General Inquiry Access Code(GIAC) or Dedicated Inquiry Access code(DIAC) which refers to a particular class of devices, say printers. The transmission is repeated at 16 frequencies which form the inquiry hop sequence, called a train. A device which is allowing itself to be inquired will be listening at one of these frequencies. The transmission is carried out in every alternate slot and the intermediate slots are used for listening to responses if any. There are two trains of hop frequencies- A and B. Each train must be repeated 256 times to collect all responses in an error free environment. The total time required for doing this is 10.24 seconds. However, if enough responses are collected in a smaller interval, inquiry may be aborted in between.

If the inquiry procedure is automatically initiated, say once every minute, then the interval between successive inquiries must be random to avoid synchronization and hence a collision with another device involved in inquiry.

Inquiry Scan

A device which allows itself to be discovered will periodically enter the inquiry scan substate and listen for inquiry packets at a single frequency, which it will chose out of the 16 frequencies in the inquiry hop sequence depending on its device address. It will stay in that state long enough for an enquiring device to cover 16 different frequencies. A device may be entering inquiry scan state from standby or connected states. If it is entering from the connection state, the SCO links in operation will be maintained while the ACLlinks will be suspended. The presence of SCO inks may prolong the inquiry procedures.

Inquiry Response

When an inquiry message is received in the inquiry scan state, a response packet containing the responding device address must be sent. However it is not sent in the immediately following slot after the slot in which inquiry is received as that might cause many devices listening at a given frequency to respond simultaneously, resulting in a collision. So the responding device waits for a random number of slots and then sends its FHS packet to the inquirer. The FHS packet contains the device address; its clock and information about when the device enters its page scan states. After responding to an inquiry, the device continues its inquiry scan at another frequency, without waiting for an acknowledgement.

The inquiring device on receiving an inquiry does not acknowledge the response but continues its inquiry procedure as long as it wishes to. Only when the inquiring device wants to page the device that responded, say at a later time when a connection is required, it will use the response information to page.

After the inquiry has been successfully carried out, or the device address of the device to which a connection has to be made has been determined by some other means like information from previous connections, the device will start a paging procedure if a connection is desired. Paging requires only the address of the device to be paged but the clock information, from the FHS response packet, may be used to speed up the procedure. The device starting the paging procedure is called the master, and it will be the master of the piconet consisting of itself and the paged device if the paged device accepts the connection. Before starting data communications however, the devices may exchange their roles.

This procedure will usually occur whenever the Bluetooth device enters a new environment or some older links become unavailable. Now, when the application is invoked, the device will start paging procedures.

Page

This state requires the master to do the following:

The master uses the clock information, if any, about the slave to be paged, to determine where in the hop sequence, the slave might be listening in the page scan mode. This estimate may be totally wrong.

The master calculates the Device Access Code (DAC) of the slave from the device address of the slave using a well-defined procedure.

The master sends a page message. The master transmits this page message at a number of frequencies in the page hop sequence, starting with the frequency at which it had estimated the slave to be listening. The page hop sequence consists of 32 frequencies divided into two trains of 16 each. The train A includes the 16 frequencies surrounding the predicted frequency and train B the remaining ones. If the clock estimate is within □8x1.28 to +7x1.28 seconds then the slave will be able to respond within the train A itself. The master does not know when the slave enters the page scan mode, so the page train is repeated N_{page} times, unless a response is received earlier. N_{page} is determined such that slaves using any of the allowed scanning intervals may be covered. If train A fails, train B is tried, again N_{page} times. If train A is successful, the paging procedure will be over in 1.28 seconds, else it will take 2.56 seconds.

Page Scan

The page scan substate can be entered from the standby state or the connection state. In this state, the slave listens to page packets addressed to its DAC for an interval $T_{w\text{-}page\text{-}scan}$ at a frequency it chooses out of the page scan sequence. This window is long enough to cover 16 frequency hops of a paging device. These listening periods are separated by time interval of $T_{page\text{-}scan}$. This interval may be zero (continuous scan). Three different scan modes, that is values of $T_{page\text{-}scan}$ are fixed. Other values may be used by a slave after informing the master. Thus one of the standard values is used for the first link.

Page Response

On receiving the page message, the slave enters the slave page response substate. It sends back a page response consisting of its ID packet that contains its DAC, at the frequency for the next slot from the one in which page message was received. The master on receiving this packet enters the master page response substate. At this point, it knows which frequency the slave had been listening. The master sends its FHS packet to the slave informing the slave of the master clock, still using the slave DAC, at the slave's listening frequency. The FHS packet also assigns a three-bit active member address to the slave. The slave acknowledges this packet again sending its ID packet at its slave response frequency. The slave now uses the FHS packet received from the master to determine the channel access code for the piconet newly formed, or to which this slave has newly entered. It also calculates the clock offset to be used while communicating over this piconet. The next packet from the master to slave, which is the POLL packet addressed to the active member address of the slave, is at the master clock dependent frequency hop and uses the channel access code. The slave may respond to this packet with any packet, say a NULL packet (containing only channel header), but it must respond. If the response procedure is successful, the paging is over, and the slave is in connected state. Otherwise, paging is considered to have failed and the error procedures are followed.

Step	Message	Direction	Hopping Sequence	Access Code and Clock
1	slave ID	master to slave	page	slave
2	slave ID	slave to master	page response	slave
3	FHS	master to slave	page	slave
4	slave ID	slave to master	page response	slave
5	1st packet master	master to slave	channel	master
6	1st packet slave	slave to master	channel	master

Figure 5: Initial message exchanges during start-up.

After the page procedure that is in the connected state the devices are in a position to establish a link.

The Link Managers of the devices in connection state now exchange vital information for starting up the link, which will be described below. They may later detach the link, in which case the address and clock information will stay valid after detachment, or the link may get snapped due to other reasons in which case all information related to the link is reset.

The Bluetooth units can be in several modes of operation during the connection state: active mode, sniff mode, hold mode, and park mode. These modes are now described.

Active mode

In this mode, the Bluetooth unit actively participates on the channel. Master and slaves transmit in alternate slots. The master transmits in all even numbered slots and the addressed slave transmits in the subsequent slot. Regular transmissions are made by the master to keep the slaves synchronized to the channel. Various optimizations are provided to save power. For instance if the master informs the slave when it will be addressed, the slave may sleep until then. The active slaves are polled by the master for transmissions.

Sniff Mode

This is a low power mode in which the listening activity of the slave is reduced. The LMP in the master issues a command to the slave to enter the Sniff mode giving it a sniff interval T_{sniff}, an offset D_{sniff}, and number of attempts N_{sniff}. In the sniff mode, the slave listens for transmissions only at fixed intervals T_{sniff}, at the offset slot D_{sniff} for N_{sniff} times.

Hold Mode

In this mode the ACL link to a slave is put on hold. This means that the slave temporarily does not support ACL packets on the channel any more (possible SCO links will still be supported). With the hold mode, capacity can be made free to do other things like scanning, paging, inquiring, or attending another piconet. The unit in hold mode can also enter a low-power sleep mode. During the hold mode, the slave unit keeps its active member address (AM_ADDR). The master and slave agree upon a duration for the hold interval, after which the slave comes out of hold mode.

Park Mode

This is a very low power mode. The slave has very little activity in this mode. It gives up its active member address and is given an eight bit parked member address and an eight bit access request address. The parked member address can be used by the master to unpark a slave while the access request address is used by the slave to ask the master to unpark it. The slave however, stays synchronized to the channel. Any messages to be sent to a parked member are sent over the broadcast channel, that is the active member address of all zeros. The parked slave has to be informed about this transmission in a beacon channel that is supported by the master to keep parked saves in synchronization and send them any other information. The parked slaves regularly listen for beacon signals at intervals decided by the beacon structure communicated to the slave during the start of parking. Apart from saving power, the park mode helps the master to have more than seven slaves(limited by the three bit active member address space) in the piconet.

Link Establishment

Once the device is in connection state, the LMP can start with the link establishment. The LMP uses its fixed LMP packets for this, which are sent by the baseband in its payload, in place of L2CAP packets with higher priority. The LMP packet consists of an opcode, transaction ID and some content(depending on the opcode). The LMP packets received by a device can be recognized from a field L_CH in the baseband packet header. These packets are then sent to LMP rather than to L2CAP and higher layers. The basic steps in setting up the connection can be summarized as:

- The POLL packets and response are used to pass configuration information without host interaction.
- The packet LMP_host_connect_request is sent.
- The remote device responds with either a LMP_not_accepted, if the application requested by the first device does not want to or cannot respond. Otherwise, an LMP_accepted is sent as a response.
- The responding slave may ask for a role switch if required for some reason. The first device responds with the appropriate packet for accepting or not accepting the request.

The link has now been established at the Link Manager level.

The application may not be knowing what services are available on the slave it paged and will use the SDP to discover those.

SDP

The Bluetooth environment changes rapidly and thus the available services have to be discovered with this in view. The SDP provides a means for the application to discover which services are available and the characteristics of these as described in the core specifications. A Bluetooth device that is willing to allow its services to be discovered runs a SDP server. A device that wants to discover services on other devices runs a SDP client. One client may be run for each application but one device runs only one SDP server. The SDP server maintains a service record of each service that the device is allowing to get discovered. A client sends a request to the server. The request may be a search for a particular class of services or a browsing attempt to see all the classes of services available. The server responds with the appropriate response.

If the server device had a few services only, they may not be divided into classes and their service handles are sent to the slave. Otherwise class descriptors are sent and the client may further search for details within a class. The SDP only allows services to be discovered. The access has to be through other protocols, based on L2CAP.

L2CAP link

The information obtained over the LMP link and through SDP will be used by L2CAP to establish a channel for the application. L2CAP establishes only ACL links, for SCO links the application uses the baseband directly.

The L2CAP links are based on the concept of channels, which are identified by channel identifiers, analogous to sockets in TCP-IP. The channel, distinct from the piconet channel, is identified by the device address to which the link is made and a channel identifier allotted to the remote device for the particular connection for one instance of an application. Each channel is assumed to be full duplex, with a QoS specification in each direction. Further the channel can be point-to-point or multipoint. L2CAP establishes links when a demand for a link is expressed by an application and a link to the required device has not already been set up. The request from lower layers regarding connections demanded by applications on remote devices is also handled by L2CAP, in consultation with the application involved.

The link is datagram based, with no streaming. SCO links do not go through L2CAP but send their data directly to Baseband. L2CAP establishes a separate signalling channel for connection request, configuration, disconnection and echo (for testing). L2CAP packets have been designed with low overhead and do not provide CRC or other error checks. It relies on baseband for data security and ordered delivery.

The interaction of this protocol with upper and lower layers is viewed in terms of events and actions. Events are all messages received by L2CAP from lower or higher layers while the actions are the responses produced for them. The lower layer may be LMP or HCI, while higher layer could be any application. A typical sequence of events and actions for establishing a connection could be as follows:

Event and Action 0: The event is a connection request from a higher layer. The action is that the device L2CAP sends a connection request packet to remote L2CAP. This packet is carried by baseband to the remote device.

Event and Action 1: The remote L2CAP receives this packet and responds with a connection response packet. Before doing so, that L2CAP would have contacted the referred application to check if the demanded request would actually be handled by that application.

Event and Action 2: The reception of the response packet is an event for the local device L2CAP. The reciprocal action is to ask for configuration parameters like maximum payload unit and timeout limit. These may include QoS among other things.

Event and Action 3: The configuration request is an event for the remote L2CAP. Its action is the configuration response. Also, it may send its own configuration request for additional parameters.

Event and Action 4: The above packet is an event for the local device. It replies with the configuration response.

These steps are summarized in Fig.6. The "initiator" is the L2CAP in the local device and the "target" is the L2CAP in the access point, or other Bluetooth device being contacted. Note that the packet names on arrows pointing towards the initiator or target are events for L2CAP while the arrows pointing away are the actions. The names beginning with L2CAP are communications between two L2CAPs on different devices. The vertical lines named LP are the Link Managers in the two devices. The names beginning with WA represent communication with the higher layer application for which the channel is being set up.

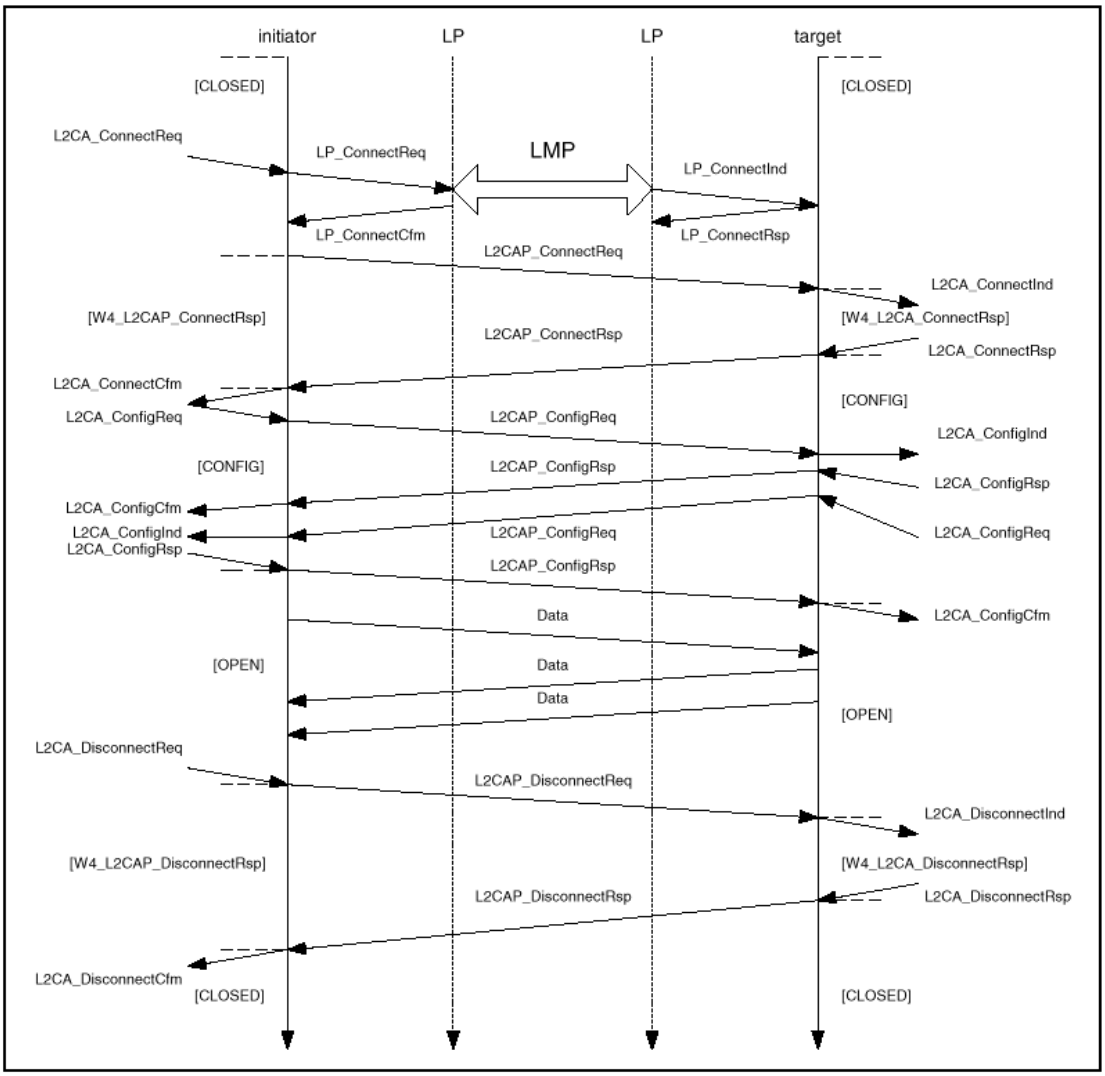


Figure 6. The basic message sequence in L2CAP channel establishment.

The OPEN states mark the interval when the applications communicate. The last steps in the above figure refer to connection disconnect.

Application data may now be transferred or security procedures may be carried out, which are briefly described in the next subsection.

Security

The communicated data may have to be encrypted or the access to the device may have to be restricted by providing an authentication point. Both these functions are provided by the Bluetooth baseband. The application may itself encrypt its data for added security. These procedures use four values: the device address (which is public), a private authentication key (128 bits), private encryption key (8-128 bits, configurable) and a random number. As the keys have to be secret, they cannot be obtained by inquiry. The security procedures require a secret PIN to be known to the user (or stored by his application) for accessing a particular device. The main steps in the procedure are:

An initialization key is generated using the PIN, the length of the PIN, a random number and the device address. The dependence on the device address makes it more difficult for a fraudulent device to try a large number of PINs, as each has now to be tried with different device addresses.

An authentication procedure is carried out using a challenge response scheme. The verifier unit sends a random number generated by a specific process for the authentication. This random number is such that a claimant device that has the correct initialization key (or a link key if the devices had exchanged that during an earlier communication) and the required device address will be able to produce a response number that is known to the verifier. This response number is sent back and checked by the verifier.

The claimant may also carry out a verification on the verifier using a similar procedure as above.

Each Bluetooth unit has a unit key, installed in its non-volatile memory. The device now uses the initialization key to encrypt this unit key and sends it to the other device that decrypts it using the initialization key exchanged earlier.

The second device may add its own unit key to the unit key of the first device and generate a combination link key if both the devices are capable of handling this. Otherwise the unit key of one of the devices is treated as the link key. The link key is communicated to the first device. The initialization key is discarded.

An encryption key is now generated from the link key, a random number and another number obtained from a fixed procedure. Both the devices can generate this encryption key as all the required information is known to both devices. This key is used to encrypt data payloads.

The link key is remembered. If another link is to be established between the two devices at a later time, this link key can be directly used. This eliminates the need to send keys over the channel again. Thus, data can be transmitted securely with minimum user interaction.

Application link

The application data will now be transmitted over the link as all the Bluetooth specific link establishment procedures have been carried out. The application may need to run a higher level protocol over L2CAP. Three useful protocols have been defined by Bluetooth to help port applications over the Bluetooth stack. These are:

- **RFCOMM** This is an emulation of the serial port over wireless links.
- **SDP** This is the Service Discovery Protocol, which helps devices discover which services are available in the proximity.
- **TCS** This is the Telephony Control Protocol Specification and describes the call control and signalling of voice channels over Bluetooth.

All user applications and other existing network access mechanisms like TCP-IP, PPP, IrDA OBEX, WAP and HomeRF can be implemented over the L2CAP layer or above the above three protocols if the application wants to use their services.

The application will finally indicate it no longer needs the link, if the link was not snapped during the application's running time. The LMP sends a packet LMP_detach packet to the remote device. There need not be any response to this. The link is then disconnected.

Bluetooth Security

The Bluetooth system is intended to be used as a uniform interface to all of a person's information sources and will thus be expected to transfer sensitive personal data. Security of the data is thus understandably an important issue. Further, Bluetooth devices are expected to be omnipresent and at some places the access to these devices by public users may have to be restricted. This calls for authentication procedures to be provided. As the channel used is wireless and the packets being transmitted are available to all members of a piconet, the security initialization communications should not send any information that can allow an unauthorized device to know the secret authentication keys. Further, the mechanisms should be appropriate for a peer-to-peer environment. The methods adopted by the Bluetooth standards take care of these issues. The scheme used is referred to as the challenge response scheme.

The application may itself encrypt its data for added security. This can add to the safety of the data, but most of the authentication is based on the link level security procedures, as it is difficult to achieve uniformity in this step at the application level.

The basic structure

The procedures for security use four values: the device address (which is public), a private authentication key (128 bits), private encryption key (8-128 bits, configurable) and a random number. As the keys have to be secret, they cannot be obtained by inquiry. The exchange procedures will be described below. The security procedure requires a secret PIN to be known to the user (or stored by his application) for accessing a particular device. The main steps in the procedure are:

An initialization key is generated using the PIN, the length of the PIN, a random number and the device address. The dependence on the device address makes it more difficult for a fraudulent device to try a large number of PINs as each has now to be tried with different device addresses.

An authentication procedure is carried out using the challenge response scheme. The verifier unit sends a random number generated by a specific process for the authentication. This random number is such that a claimant device that has the correct initialization key (or a link key if the devices had exchanged that during an earlier communication) and the required device address will be able to produce a response number that is known to the verifier. This response number is sent back and checked by the verifier.

The claimant may also carry out a verification on the verifier using a similar procedure as above.

Each Bluetooth unit has a unit key, installed in its non-volatile memory. The device now uses the initialization key to encrypt this unit key and sends it to the other device that decrypts it using the initialization key exchanged earlier.

The second device may add its own unit key to the unit key of the first device and generate a combination link key if both the devices are capable of handling this. Otherwise the unit key of one of the devices is treated as the link key. The link key is communicated to the first device. The initialization key is discarded.

An encryption key is now generated from the link key, a random number and another number obtained from a fixed procedure. Both the devices can generate this encryption key as all the required information is known to both devices. This key with some modification as described later, is used to encrypt data payloads.

The link key is remembered. If another link is to be established between the two devices at a later time, this link key can be directly used. This eliminates the need to send keys over the channel again. Thus, data can be transmitted securely with minimum user interaction.

Detailed Description

This section describes the above steps in some detail. Note that the key used for authentication will be different from the one used for encryption.. Further, the authentication key will not usually change for one link after being set once in a session, that is the duration of belonging to a piconet. It might even be stored for future links, thus making it convenient for the user for she will not have enter the PIN again for accessing devices for which she has been authenticated once. This key is thus referred to as the link key. The encryption key is generated each time for a new link and this helps improve the security.

Generation of the link key

Setting up an initialization key: A link key used temporarily during initialization is required before the actual link keys can be generated. This is called the initialization key. This key is derived by the E_{22} algorithm from the device address-BD_ADDR of the claimant unit, a PIN code, the length L' of the PIN (in octets), and a random number IN_RAND A issued (and created) by the verifier. See Fig .7.

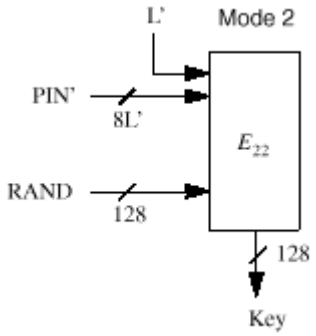


Figure 7: Generation of the Initialization Key by Mode 2 of the E_{22} algorithm.

The PIN is augmented with the device address-BD_ADDR of the claimant unit. Since the maximum length of the PIN used in the algorithm cannot exceed 16 octets, all octets of BD_ADDR might not be used. This procedure ensures that the initialization key depends on the identity of the unit trying to connect to it (at least when PIN codes shorter than 16 octets are used). The linkkey has not been transmitted from either device. The PIN has to be entered on each device (it may be prefed in devices that do not have an interactive input) and the initialization key is generated locally in each device. A fraudulent Bluetooth unit may try to test a large number of PINs by each time claiming a different BD_ADDR. It is the application's responsibility to take counter measures against this threat. If the device address is kept fixed, the waiting interval until next try is permitted is increased exponentially.

Authentication: The initialization key is now used to authenticate in both directions. The procedure used is the E_1 algorithm described in the core specifications.
The verifier challenges the claimant to authenticate a random input (the challenge), denoted by AU_RAND_A, with an authentication code, and return the result SRES to the verifier. The verifier can check whether the SRES is as expected. See Fig.8. The link key here is the initialization key.

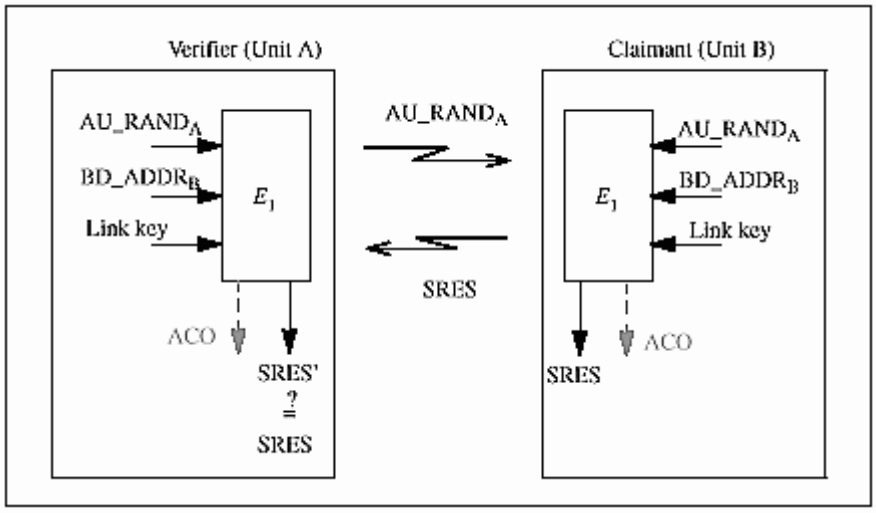


Figure 8: Authentication using the E_1 algorithm.

Each system has a unit key. This is generated when the device is first time in operation and is stored in a non-volatile memory for use for the lifetime of the device (In some rare instances it may be changed). A 128-bit RAND value and a 48-bit address is used by mode 1 of E_2 algorithm to generate this 128-bit unit key. See Fig.9.

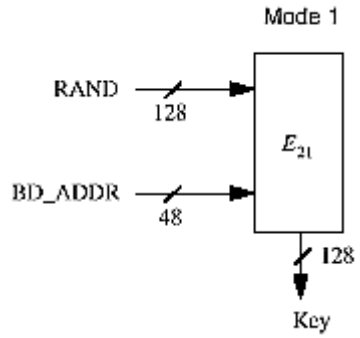


Figure 9: Generation of unit key.

The unit key is now used to generate the link key. There are two possible ways of doing this. If one of the units has a limited memory capacity, then the unit key of that unit is encrypted with the initialization key and used as the link key. See Fig.10.

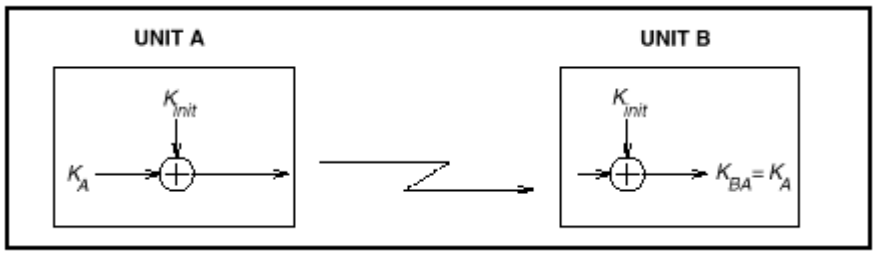


Figure 10: Setting up the link key derived from only one unit key.

In the other case when both devices can support it, a combination key depending on the unit keys of both the devices involved is used. In this procedure, both the devices generate a random number. Then using the E_{21} algorithm as in Fig.3, they generate a 128 bit key, independently. The two random numbers generated are exchanged securely by encrypting with the initialization key (or the current link key if any). From the random number received from the remote device and from the knowledge of the device address of the remote unit, the device can find out the random number used by the remote device in generating this key. This number is bitwise XORed with its own random number to generate the link key. Thus the link key is obtained for use in all subsequent authentications. Once the new link key has been generated, the initialization key is discarded.

Generation of the encryption key

The encryption key is derived by algorithm E_3 (details given in core specifications, Baseband, section 14) from the current link key, a 96-bit Ciphering OFfset number (COF), and a 128-bit random number. The COF is determined in one of two ways. If the current link key is a master key, then COF is derived from the master BD_ADDR. Otherwise the value of COF is set to the value of Authenticated Ciphering Offset (ACO) as computed during the authentication procedure (ACO was computed in authentication using E_1 as shown in Figure 2 above). This encryption key is generated each time the unit enters encryption mode. See Fig.11.

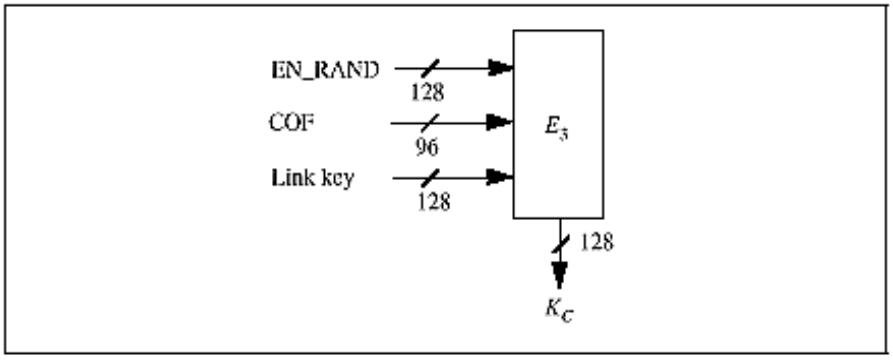


Figure 11: Generation of the encryption key.

This encryption key is not used directly to encrypt the data. Rather this is used to generate another encryption key using the algorithm E_0 for each packet. The encryption key thus changes for each packet transmitted. The E_0 algorithm uses the clock information of the sender in generating the new encryption key. See Fig.12.

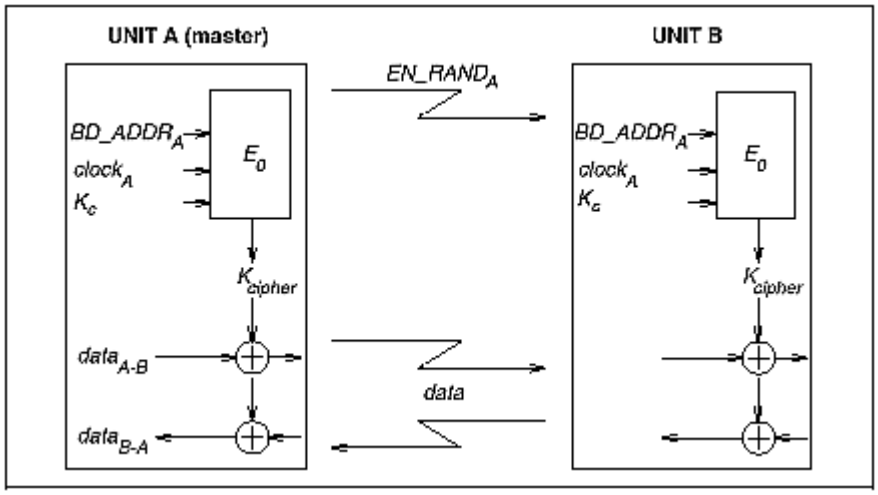


Figure 12: Encryption of data over a Bluetooth link.

Thus secure communication can be achieved.

There are many possible modes in encryption for instance to support point to multipoint configurations. As the encryption key length is variable, the facility to exchange the key lengths is provided. Further provisions are made to allow the change of link keys before link expiry. The unit key may also be changed in certain circumstances but that will require reinitialization of all link keys previously stored by devices authenticated at an earlier point in time. This may be desirable if the set of users allowed access has to be changed.

This article briefly described the security facilities provided by Bluetooth. The standard has made attempts to achieve a high degree of security.

Why is Blutooth the way it is?

This section attempts to explain some of the design and technology tradeoffs in Bluetooth and the main considerations in the wireless link design are explored.

Ad hoc connectivity

Most wireless communication systems like the public cellular phone networks- GSM, DAMPS, IS-95 or other private networks like Hiperlan-II, DECT or Personal Handyphony system, use a network architecture in which the radio units: base stations and mobile terminals are strictly distinct. This is advantageou in design as channel access, channel allocation, traffic control, interference minimization etc. can be taken care by the base stations, making the design of mobile terminals simpler. In ad hoc networks, there is no difference between radio units. Communication is peer to peer with no central controller. Conventionally in ad hoc wireless networks, all devices sharing a common space will share the same channel, and will mutually coordinate in its sharing. In Bluetooth usage models however, even this is not sufficient as the number of Bluetooth devices in a given region of space may be very large and only a few of them may need to communicate among themselves, making mutual coordination among them very difficult and unlikely. This has led to the concept of scatternets: a group of networks in the same space but communicating over different channels, with some overlapping devices. There need not be any coordination among devices belonging to different networks within the scatternet. See Fig.1

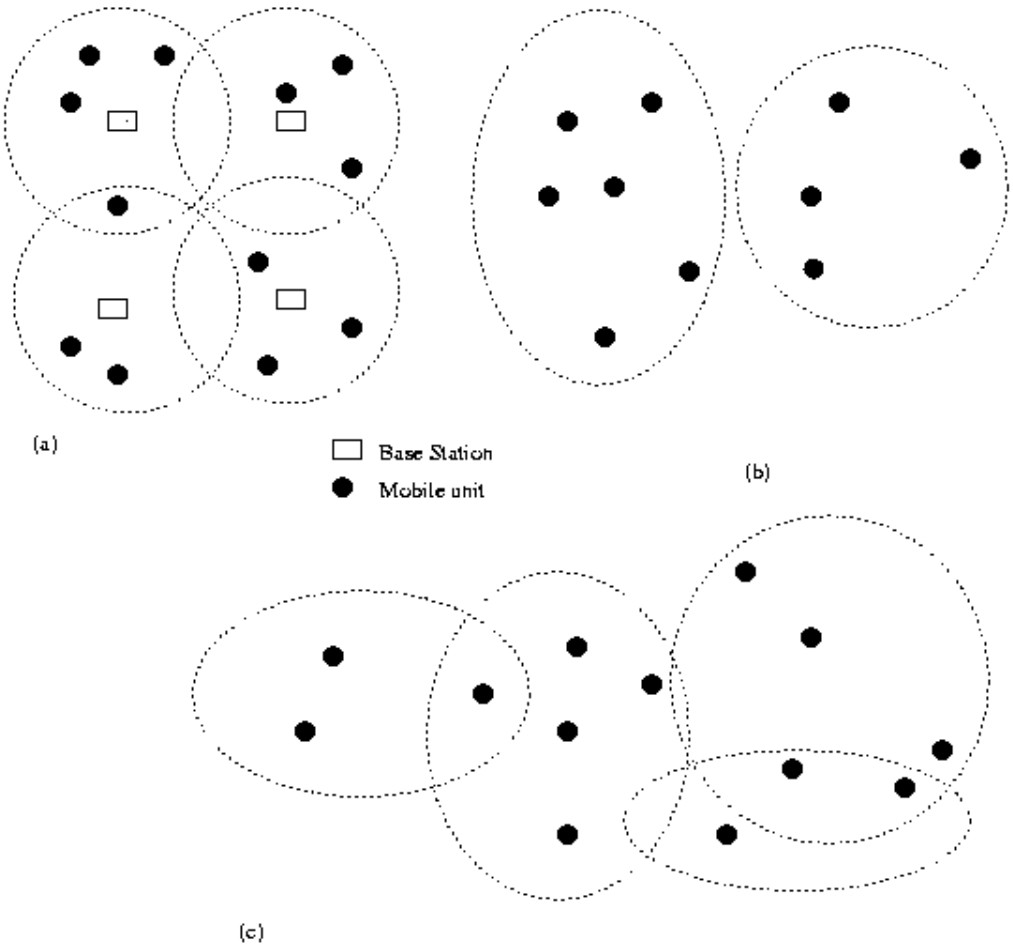


Fig.1 (a) A cellular network with squares representing stationary base stations
(b) A conventional ad hoc system
(c) scatter ad hoc network

Thus it is clear that the Bluetooth system due to its very nature of application will have to use scatternet kind of ad hoc connectivity. The major considerations in this design are:

- Choice of the radio spectrum.
- Connection Establishment, and determining units in range that can be connected.
- Choice of the multiple access scheme.
- Channel allocation.
- Medium Access control.
- Service prioritization. voice may have higher priority over data.
- Interference, mutual and from other sources.
- Power consumption.
- Protection of data over the channel.

The choice of the radio spectrum.

The issues to be considered here are:

- There will not be any coordination between operators as in a cellular network.
- The spectrum must be available worldwide without the need for licences. This will make the interoperability truly global. This is important, as mobility is one main advantage for Bluetooth devices.

These considerations encourage the adoption of the ISM band (around 2.45GHz.) that is globally available. It is governed by different regulations in different parts of the world and hence the system must be designed considering the minimum common availability.

Medium access control

The choice of the Medium access scheme must take into account the lack of coordination among devices in the ISM band. FDMA (Frequency Division Multiple Access) cannot be used, as it does not satisfy the frequency spreading characteristics of the ISM spectrum. TDMA (Time Division Multiple Access) requires strict timing synchronization that is rather cumbersome for ad hoc connections. CDMA (Code Division Multiple Access) is the clear choice as it fulfils the spreading requirements and can work with uncoordinated systems. The next question is DS (Direct Sequence) or FH (Frequency Hopping). DS suffers from certain disadvantages like the requirement of a common timing reference, which is not a good idea in the scatternet ad hoc scenario. Further, due to the near far problem, a coordinated power control is required. Finally, high data rates will require higher chip rates-which is not advisable due to the wide bandwidth and the resulting interference, and due to excessive current drain. FH, apart from taking care of these problems, offers other advantages. The average signal is spread over a large bandwidth but the instantaneous signal is only in a narrow band, making it easy to filter out a lot of potential interference. Bluetooth thus uses FH-CDMA. To keep interference effects minimal and to make the data robust, very short packets are used and the frequency hop rate is kept high-1600 hops per second.

Connection Establishment.

The environment of a Bluetooth device will change dynamically with fair amount of rapidity. Further the number of devices will be quite large and new devices will usually be have to accessed whenever the user takes his device outside his usual workplace. Then how do the devices find each other and establish links over the CDMA channel? It is important that the devices should not depend on manual commands to establish connections. So the standard must provide for procedures using which a device can discover the addresses of other Bluetooth devices in its proximity without any user support. Further there should be some mechanism to start a link. This will require some synchronisation to be achieved between the two devices. Procedures called inquiry, paging and scan have been defined to take care of these requirements and are discussed in the tutorial on connection establishment.

Medium Access Control and channel allocation

As noted earlier, a large number of independent channels need to exist in the same space, each serving its own participants. On the modulation scheme used, the data rate available is 1Mbps. So,

to conserve capacity, only the units that need to transfer data among themselves should be put on a particular channel. For this reason, the concept of piconets has been introduced. Each channel is identified with a unique hop sequence and the clock of one coordinating device on that channel, referred to as the master. This channel called a piconet and multiple piconets that overlap in terms of devices connected are referred to as a scatternet. To simplify implementation of duplex communication, TDD has been applied. Each device transmits in alternate slots and uses the intermediate slots to receive. See Fig.2.

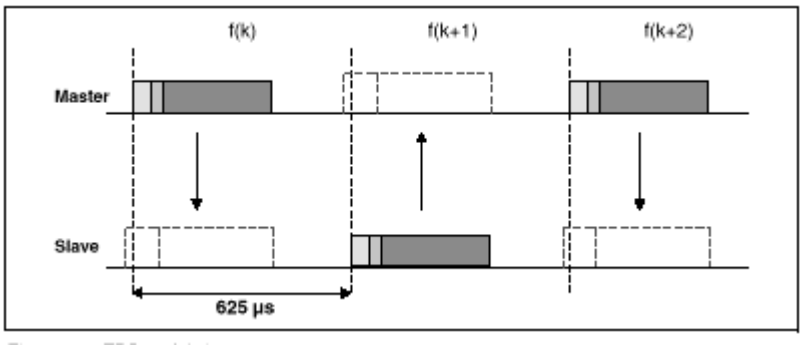


Fig.2. The TDD scheme in Bluetooth.

The master of the piconet manages which device transmits when. Thus, the channel can be effectively shared. One may note that there are 79 frequencies in the hop sequence opening the possibility for 79 orthogonal frequency hop sequences. That can provide 80Mbps of data transfer capacity within a local space. However, as the Bluetooth devices do not coordinate, the hop sequences will not be orthogonal and the theoretical capacity of 80Mbps will not be reached.

Service prioritization

Bluetooth devices intend to provide both voice communication as in mobile phones, headsets or voice browsers and also data communications to access networks in the traditional way. The voice channel must be synchronous and provide guarantees on delay and latency for acceptable voice quality. Thus a separate link to support voice is required which must be given priority over asynchronous data. Hence, Bluetooth uses two kinds of links:

- Asynchronous Connectionless-ACL.
- Synchronous Connection Oriented-SCO.

The SCO link is a point-to-point link in which resources are reserved-TDD slots at regular intervals are reserved to guarantee the continuity of the voice channel. The remaining slots can be used by ACL links. The SCO links is such that even during paging, inquiry and scan procedures, which may be required for some parallel application, the synchronous packets can be transferred at their fixed slots.

Interference

The ISM band being an unreserved band will have a variety of heterogeneous transmitters in its frequency range. Further, microwave ovens and lighting sources emit in this band-which was in fact the original reason for unlicensing this band. Another source that may be expected to be present in the same areas as the Bluetooth environment would be the 30dBm WLAN transmitters. Thus the Bluetooth devices must be immune to such interference. Two common approaches are suppression and avoidance. Suppression, which can be obtained by coding or DS spreading, makes less sense for Bluetooth because the near far ratios may be too high to handle with practical attainable coding gains. Interference avoidance is more attractive because the desired signal is transmitted at points in frequency and/or time where interference is low or absent. Avoidance in frequency is more practical because most radio systems are nband limited and it should be possible to find some band where the interference is fairly low. Further, robustness to errors is required as interference in short bursts due to collisions or other reasons may corrupt data packets in a wireless environment. This calls for error correction at higher layers above the physical link. For this purpose Bluetooth has chosen to provide an acknowledgement based scheme with automatic repeat request (ARQ). The header information in packets that is very critical to the link operation is protected first by a cyclic redundancy check and further a 1/3 rate Forward Error Check (FEC) is applied, which repeats each bit three times. For protection of data various levels of redundancy are supported differing in their overheads.

The required packet type may be used by an application as per its requirements. For the synchronous voice links it is difficult to retransmit a packet in case of errors, hence the CVSD voice coding is employed which is robust to high bit error rates.

Modulation Scheme Used

In the ISM band the bandwidth of signals is limited to 1MHz. For robustness, a Binary modulation scheme has been chosen. This is what places the data rate limit of 1Mbps. For FH systems and support for bursty data traffic, a non-coherent detection scheme is most appropriate. A Gaussian pre-filtered Frequency Shift Keying(FSK) is used with a nominal modulation index of 0.3. The Time Bandwidth product of the Gaussian pre-filter is 0.5. Logical ones are sent as positive frequency deviational and zeros as negative frequency deviations. In this technique, demodulation can be simply accomplished by a limiting FM discriminator, thus enabling the implementation of low cost radio units.
See Fig 3.

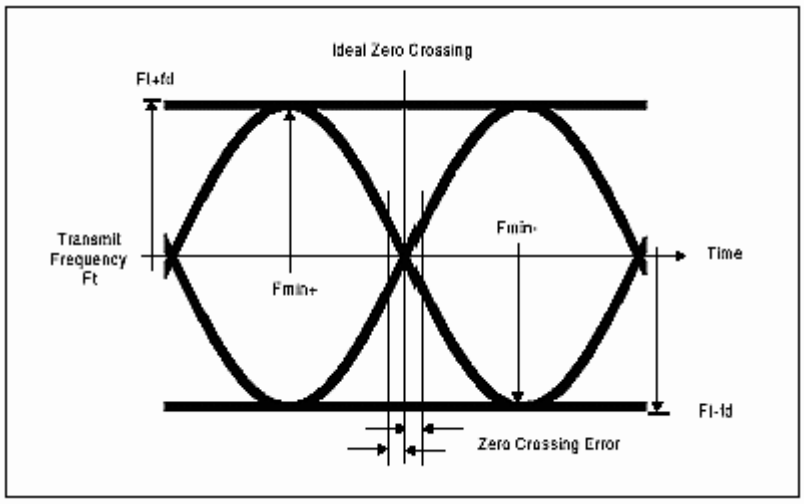


Fig.3. The Modulation scheme used in Bluetooth.

Protection of Data

Bluetooth devices will usually carry and transmit a person's very personal data as they will be the interface to most of her valuable information resources. Thus, it is essential that no compromise on security be made while shifting from the wired interfaces to wireless ones. For this reason it is necessary to provide data encryption facility. Further, services offered through Bluetooth access points may have to be limited to a certain set of authorized users. This means that the standard should provide some mechanism to allow restricted access and identification of registered devices. Keeping both these requirements in mind, Bluetooth provides both link level encryption and authentication. These are controlled by providing a PIN (Personal Identification Number) to users, and this number must be known to the user if he has to access a secured device. Thus, access control can be implemented. To ensure safety of encryption long encryption keys are used. Further these keys are not transmitted over the wireless channel, rather other parameters are intimated using which in combination with the information specifically known to authenticated devices, the keys can be generated. Different types of keys have been provided to allow for varying computing power on non-uniform devices. Apart from encryption at link level, which uses 128 bit keys, regarded as fairly safe currently, there is always the option to further encrypt at the application layer itself. Link level encryption however allows a universal standard. The details of how the security is implemented are provided in the tutorial on security.

Power Consumption

Wireless connectivity is rendered useless if a power cable has to be connected. Mobility can be achieved only if the device is battery operated. Thus power consumption becomes an important consideration. Attempts have been made to keep power consumption to the minimum, preferably not more than a small fraction of the power consumed by its host device. Various low power modes (Hold, Sniff, Park) have been provided to allow the unit to adjust its power consumption to the bare minimum required and use full power only when actively engaged in communication.

Design of the higher layers

Some issues to be considered in the design of the higher layers were that the Bluetooth module should be easily integratable into any existing device that may benefit from being connected. Further, for rapid acceptance into the market, it should be possible for existing applications developed over conventional protocols to be easily portable to the Bluetooth platform. In addition, other wireless initiatives like IrDa OBEX, WAP and HomeRF should also be able to interoperate with this standard. Keeping this in view, interface points where existing protocols may be used over the Bluetooth platform have been provided.

The Bluetooth devices have some peculiar needs not covered by existing protocols, like the need to discover what services are available on devices in proximity. Special protocols have been provided to meet these needs. For example, to discover services on nearby devices, Service Discovery Protocol (SDP) is provided. Similarly, for voice channel control, TCS (Telephony Control Specification) has been provided. To allow the applications using the serial port, an emulation of the same, the RFCOMM has been provided. Thus, an attempt has been made to provide for interoperability and universal acceptance. Existing protocols can be supported over Bluetooth for rapid development of applications.

The Host Controller Interface.

This is a special layer added to the Bluetooth protocol stack due to the special requirements that can be met using this interface. This is especially relevant for integrating Bluetooth into computers and other general-purpose devices. As the lower three layers of the protocol stack-up to the LMP, may be implemented in hardware and the rest may be in software, especially the L2CAP, which is the main interface to applications, a means of communication is required between the two levels. In a PC or a laptop, the Bluetooth module would be preferred as an add-on card, either o the USB or the PCI. Then this Host Controller Interface provides the required functionalities for allowing the transfer of L2CAP data in the USB or PCI formats.

Conclusions

The above sections described some of the important considerations in the design of the Bluetooth standard. There are however a lot of other considerations such as the avoidance of collisions and satisfying the radio spectrum regulations. The hop selection procedures in the CDMA-FH, in inquiry and paging schemes and in other procedures related to management of low power modes have thus been selected to ensure that minimum interference is generated and collisions are avoided. Timing intervals have been designed using random numbers such that no two devices accidentally get in sync and collide in their transmissions, for instance in replying to an inquiry message. Attempts have been made to meet the requirements in optimal ways. Certain requirements that seem to have emerged in the usage scenario, like video streaming and other high data rate multimedia applications are being considered in the design of Bluetooth version 2 and 3, along with the IEEE 802.15 standards.

FAQs

Q1. What is the significance of "PIN CODE" in Bluetooth?

The PIN code is a secret code provided for security applications. This number has to be manually entered by the user or stored by his application. Once the device knows the PIN of the device it wants to connect to, it can start the security procedures relevant to authentication and encryption. In some devices like access points or headsets, which do not have an interface good enough to allow the users to enter a PIN, the PIN is prefed and used by other devices to connect to it.

Q2. What forms the host and what is the host controller in a Bluetooth device?

The host is the device that wants to use Bluetooth for its communication requirements. It could be a laptop, a headphone, a PDA or any other device. The host controller is a special module on the Bluetooth hardware that formats the data packets being communicated from the host to the Bluetooth module into a format suitable for the link between the host and the module. For example, if the host is a laptop and the Bluetooth module a USB card, the host controller will format the packets from host's USB format to the Bluetooth LMP format.

Q3. What is the modulation index used in the radio layer in Bluetooth?.

The modulation index is between 0.28 and 0.35. This gives a frequency deviation of 140kHz to 175 kHz on the 1MHz carrier, with binary FSK.

Q4. How is the situation handed if two devices launch an inquiry simultaneously?

When two devices start an inquiry simultaneously, then they cannot discover each other in that inquiry sequence (they do not cause collisions for other devices as the inquiry hop sequence is different for the two devices). To prevent the devices never being able to discover each other, the specification requires that the interval between two inquiries be random. This ensures that the two devices that collided once will not collide again in the next inquiry.

Q5. How does the device schedule itself (figure out which piconet to service/listen)? For example, there may be slots in which the device has obligations on BOTH piconets.

The Bluetooth specification says that different piconets accessed are multiplexed in time. The exact details of how the device wants to do it depends on the implementation, as the specification does not give any fixed priority or scheduling structure. Depending on whether a slave is active/parked etc. it would spend different amounts of time with different masters.

Q6. For a computer having all its peripherals connected through Bluetooth, who will be master and who will be slave?

The Bluetooth specification does not require any particular device to be the master. However as all communications would probably be through the unit attached to the CPU, it seems most efficient for that unit to become the master.

Q7. Are there any specifications for the timing recovery in the radio layer?

The specifications do not provide any fixed procedures. The receiver designer may use any of the well-known methods for the purpose.

Q8.Where does a Bluetooth device store the list of other Bluetooth devices in the area?

This is implementation specific. The Profile’s specifications specify that the Bluetooth controller module should store addresses of all devices in proximity and the addresses of devices with which links are open.

Q9. What is the Bluetooth data capacity and throughput?

The raw data rate is 1Mbps. But the available data rate is 723 kbps. Bluetooth can support:

- an asynchronous data channel,
- up to three simultaneous synchronous voice channels, or
- a channel which simultaneously supports asynchronous data and synchronous voice.

Each voice channel supports 64 kb/s synchronous (voice) link in each direction. The asynchronous channel can support an asymmetric link of maximally 723.2 kb/s in either direction while permitting 57.6 kb/s in the return direction, or a 433.9 kb/s symmetric link. The actual data rates depend on the kind of error correction capability introduced into the data which determines the type of packet used. See the table below.

Packet Type	Max Symmetric rate (two way)	Max Forward (Assymetric) rate	Max Reverse (Assymetric) rate
DM1	108.8	108.8	108.8
DH1	179.2	179.2	179.2
DM3	258.1	387.2	54.4
DH3	390.4	585.6	86.4
DM5	286.7	477.8	36.3
DH5	433.9	723.2	57.6
AUX1	185.6	185.6	185.6

Note: The packet types containing an 'H' in their type field (e.g. DH1) refer to packet types with low error correction overhead and high data rates. 'M' refers to medium data rate .

Q10. If a master unit leaves a piconet, what happens?

If the master leaves, the link managers of all the slaves eventually time out the connections to the master and the piconet is lost. A new piconet must then be established.

About the author

Aman Kansal is a doctoral student at the University of California, Los Angeles, pursuing research in the area of wireless networks. He has a Bachelor's degree in electrical engineering and a masters degree in Communications and Signal Processing from Indian Institute of Technology, Bombay.

He was a recipient of the Microsoft Innovation Award at IEEE Computer Society's International Design Competition featuring Bluetooth related products. He has been a technical consultant to technology start-ups including Etrek Solutions and Network Programs Inc. He is the web directory editor for Bluetooth category at Open Directory Project (www.dmoz.org).

His research areas are in wireless networks, mobile computing, embedded Internet and multimedia over packet networks, and he has published a number of technical papers in these fields.

Copyright