Bluetooth - Technology Overview

Manoj KR Shukla*

Bluetooth is the name given to a new technology using short-range radio links, intended to replace the cable(s) connecting portable and/or fixed electronic devices. It is envisaged that it will allow for the replacement of the many propriety cables that connect one device to another with one universal radio link. Its key features are robustness, low complexity, low power and low cost. Designed to operate in noisy frequency environments, the Bluetooth radio uses a fast acknowledgement and frequency hopping scheme to make the link robust. Bluetooth radio modules operate in the unlicensed ISM band at 2.4GHz, and avoid interference from other signals by hopping to a new frequency after transmitting or receiving a packet. Compared with other systems in the same frequency band, the Bluetooth radio hops faster and uses shorter packets. The following sections give the reader reference knowledge of the operation of Bluetooth and its layers, for use in the explanation of a possible handover/routing scheme.

1. BLUETOOTH PHYSICAL SYSTEM

The Bluetooth system consists of a radio unit, a link control unit and a support unit for link management and host terminal interface functions.

(a) Radio

The Bluetooth air interface is based on a nominal antenna power of 0dBm. Spectrum spreading is accomplished by frequency hopping in 79 hops displaced by 1 MHz, starting at 2.402GHz and finishing at 2.480GHz. The nominal link rang is 10cm to 10 m, but can be extended to 100m by increasing the transmit power.

(b) Link Controller

The LC carries out the **Baseband** protocols and other low-level link routines of the Bluetooth

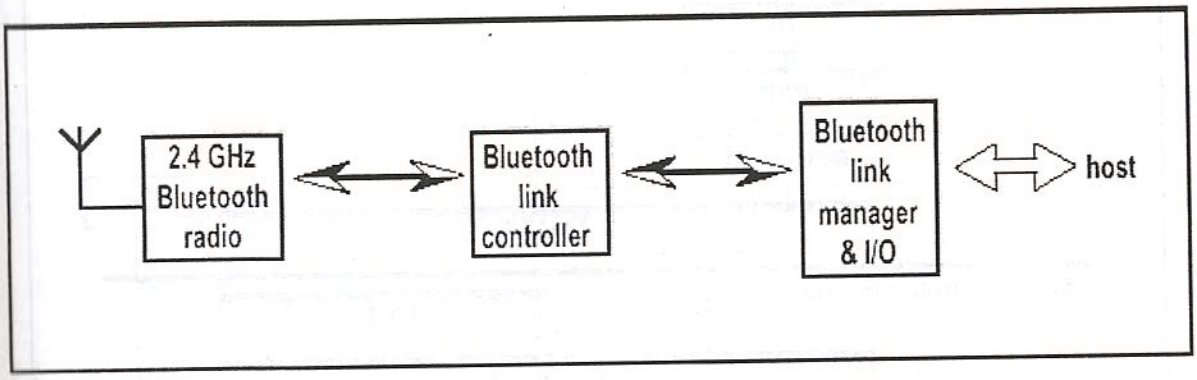

Figure 1: Different functional blocks in the Bluetooth system[1]

*Assistant Professor, Department of Electronics & Communication Engineering, Dehradun Institute of Technology, Dehradun

system. It uses the baseband to establish network connections, define link + packet types and provide error correction.

(c) Link Manager

This software entity carries out link setup, authentication, link configuration and other protocols. It discovers other remote LM's and communicates with them via the **Link Manager Protocol (LMP)**. To perform its service provider role, the LM uses the services of the underlying Link Controller (LC).

(d) Software Functions/Framework

Different Bluetooth devices have different requirements, and the function of the software functions is to meet these requirements. These are defined in the Bluetooth protocol stack, and range from radio module compliance and air protocols, up to application-level protocols and object exchange protocols. Here is where such functions as cable emulation, audio communication etc. are implemented.

2. PROTOCOL STACK

The Bluetooth protocol stack can be broken into 4 types:

1.	Bluetooth Core Protocols	:	Baseband, LMP, L2CAP and SDP
2.	Cable Replacement Protocol	:	RFCOMM
3.	Telephony Control Protocols	:	TCS Binary, AT-commands
4.	Adopted Protocols	:	Everything else (excluding Audio)

As can be seen in Figure 2 the complete protocol stack comprises of both Bluetooth-specific protocols like LMP and L2CAP, and existing protocols like OBEX and UDP.

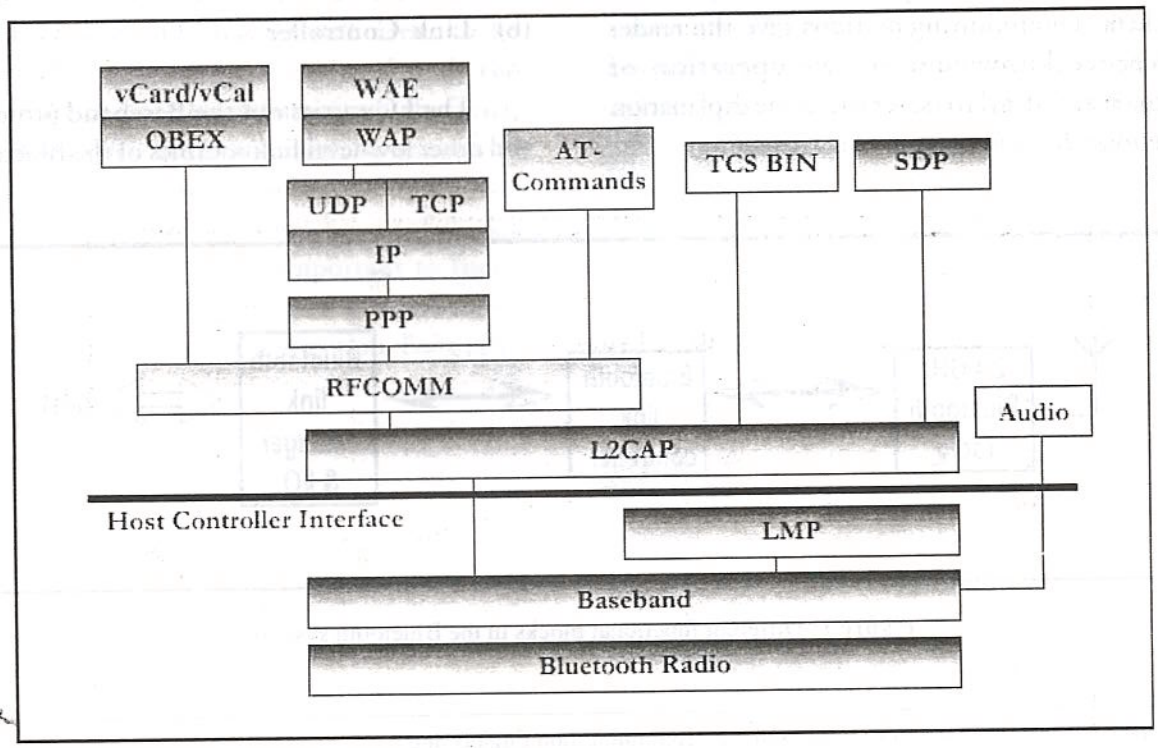

Figure 2: Bluetooth protocol stack

3. NETWORK TOPOLOGY

The Bluetooth system provides a point-to-point connection (only 2 Bluetooth units involved), or a point-to-multipoint connection (see Figure 3). In the point-to multipoint connection, the channel is shared among several Bluetooth units. Two or more units forming the same channel form a **piconet**. One Bluetooth unit acts as the master of the piconet, whereas the other unit(s) act as slave(s). Up to seven slaves can be active in a piconet, in addition many more slaves can remain locked to the master in a **parked** state. These parked slaves cannot be active on the channel, but remain synchronised to the master. Both for active and parked slaves, the channel access is controlled by the master.

Multiple piconets with overlapping coverage areas form a **scatternet**. Each piconet can only have a single master. However, slaves can participate in different piconets on a time-division multiplex basis. In addition, a master in one piconet can be a slave in another. Each piconet is identified by a different hopping frequency sequence. All users participating on the same piconet are synchronised to this hopping sequence.

4. TIME SLOTS

The Bluetooth channel is divided into time slots, each 625ms in length. The time slots are numbered according to the Bluetooth clock of the piconet master. A TDD scheme is used where master and slave alternatively transmits, (see Figure 4). The master starts it's transmission in even-numbered slots only, and the slave starts its transmission in odd-numbered time slots only. The packet start is aligned with the slot start.

Each packet is transmitted on a different hop frequency. A packet nominally covers a single slot, but can be extended to cover up to five slots.

5. FREQUENCY HOPPING SEQUENCE

The hopping sequence used within a piconet is determined by the Bluetooth device address of the master (the BD_ADDR; a unique 48-bit Bluetooth device address), and the phase in the hopping sequence is determined by the Bluetooth clock of the master. The channel is divided into time slots where each slot corresponds to a RF hop frequency. Consecutive hops correspond to different hop frequencies, with the nominal hop rate being 1600 hop/s.

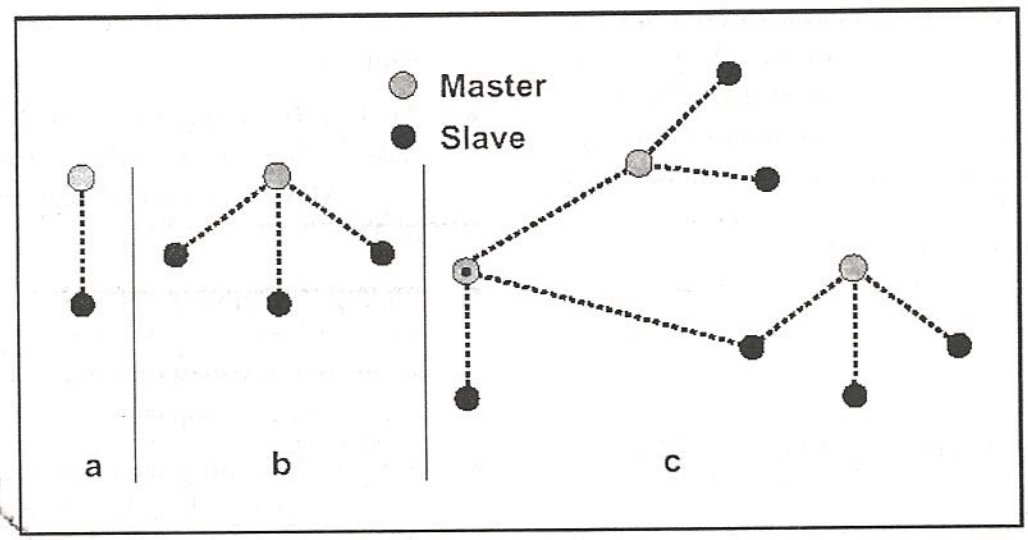

Figure 3: Piconets (a) single slave, (b) multi-slave, (c) scatternet operation

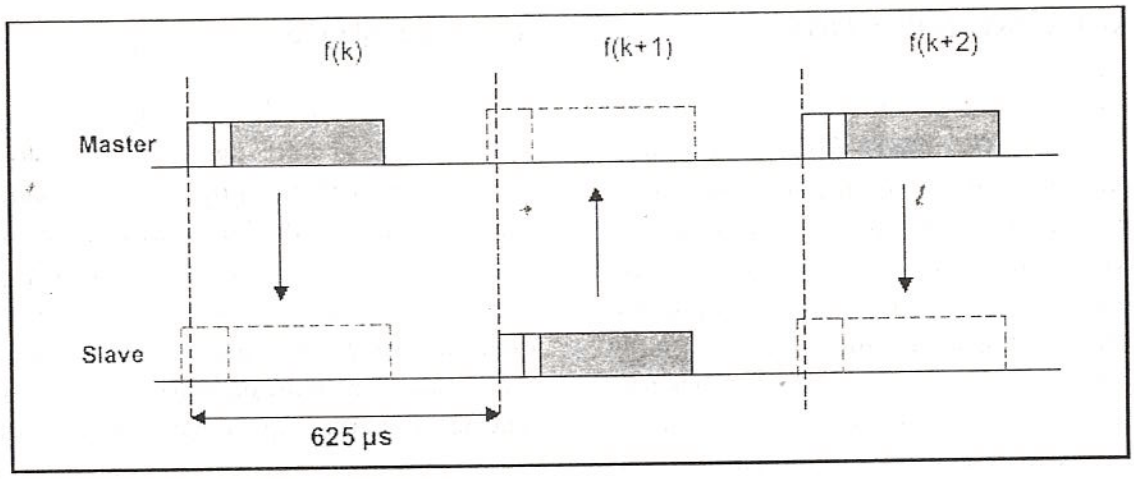

Figure 4: TDD scheme and timing

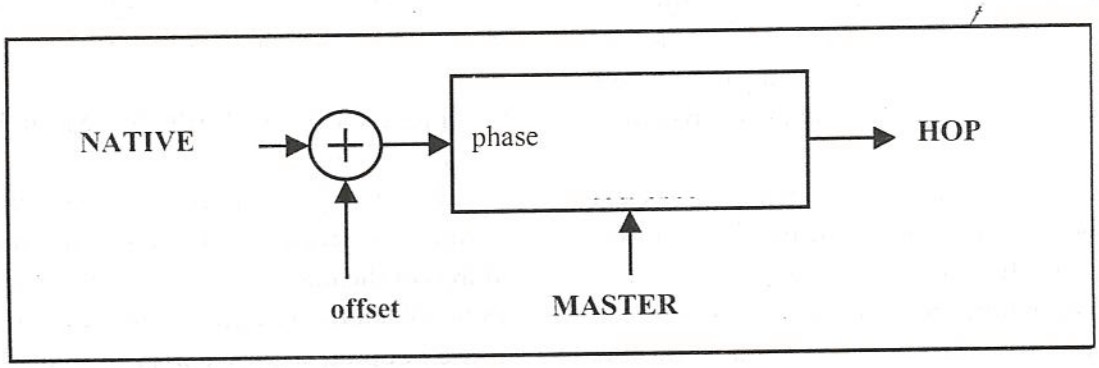

Figure 5: Hop Selection

Each Bluetooth device has its own free running clock. A slave can transmit on the correct piconet frequency sequence by inputting the master's BD_ADDR. The phase is obtained by the slave adapting their native clocks with a timing offset in order to match the master clock. This offset is adapted each time a packet is received from the master. The master does not need to insert any such offset, as it's native clock is the master clock.

6. HOP SELECTION

There are 5 types of hopping sequences defined. These are:

- A **page hopping sequence** with 32 unique wake-up frequencies distributed equally over the 79 MHz, with a period (segment) length of 32.
- A **page response sequence** covering 32 unique response frequencies that all are in a one-to-one correspondence to the current page hopping sequence. The master and slave use different rules to obtain the same sequence.
- An **inquiry sequence** with 32 unique wake-up frequencies distributed equally over the 79 MHz, with a period (segment) length of 32.
- An **inquiry response sequence** covering 32 unique response frequencies that all are in an one-to-one correspondence to the current inquiry hopping sequence.
- A **channel hopping sequence** which has a very long period length, which does not show repetitive patterns over a short time

interval, but which distributes the hop frequencies equally over the 79 MHz during a short time interval.

The selection scheme chooses a segment of 32 hop frequencies spanning about 64 MHz and visits these hops once in a random order. Next, a different 32-hop segment is chosen, etc. In case of the **page** and **inquiry modes**, the same 32-hop segment is used all the time.

7. LINK TYPES

Within Bluetooth, two types of links have been defined:

- **Synchronous Connection-Oriented** (SCO) link
- **Asynchronous Connection-Less** (ACL) link

The SCO link is a point-to-point link between a master and a single slave in the piconet. The master maintains the SCO link by using reserved slots at regular intervals, and thus can be considered as a circuit switched connection between the master and slave. The SCO link typically supports time-bounded information like voice. The ACL link is a point-to-multipoint link between the master and all the slaves participating on the piconet. In the slots not reserved for the SCO link(s), the master can establish an ACL link on a per-slot basis to any slave. It therefore functions as a packet-switched connection between the master and all active slaves participating in the piconet. 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.

8. PACKET TYPE/FORMAT

The data on the piconet channel is conveyed in packets, with each packet consisting of three entities, the access code, the header and the payload. The number of bits per entity is shown below:

Several different packet types have been defined. Packets may consist of the (shortened) access code only, of the access code + header, or of the access code + header + payload. Each packet must start with an access code. If a packet header follows, the access code is 72 bits long; otherwise the access code is 68 bits long. This access code is used for synchronisation, DC offset compensation and identification. Three different access codes are defined.

- device access code (DAC), which is used during paging, and is derived from the unit's BD_ADDR

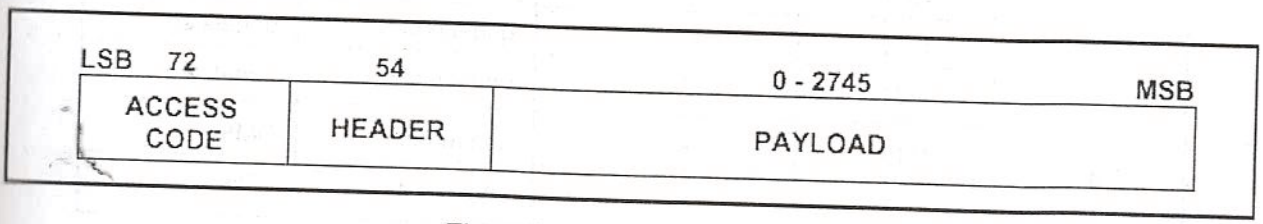

Figure 6: Standard packet format

- channel access code (CAC), which precedes all packets sent in the same piconet, and is derived from the master BD_ADDR
- inquiry access code (IAC), which is used in inquiry operations, and is a Bluetooth-wide predefined code. A general IAC (GIAC) or a dedicated IAC (DIAC) can be used.

There are 12 different packet types for both the ACL and SCO link types, with four control packets common to both link types (see Table I) In addition there is the identity or ID packet, which is not listed. This consists of the device access code (DAC) or inquiry access code (IAC) only.

9. BLUETOOTH STATES/MODES

Before any connections in a piconet are created, all devices are in STANDBY mode. In this mode, an unconnected unit periodically "listens" for messages every 1.28 seconds. Each time a device wakes up, it listens on a set of 32 hop frequencies defined for that unit. The connection procedure is initiated by any of the devices which then becomes master. A connection is made by a **PAGE** message if the address is already known, or by an **INQUIRY** message followed by a subsequent PAGE message if the address is unknown. In the initial PAGE state, the master unit will send a train of 16 identical page messages on 16 different hop frequencies

Table I: Packets defined for ACL and SCO link types

Segment	TYPE code $b_3b_2b_1b_0$	Slot occupancy	SCO link	ACL link
1	0000	1	NULL	NULL
	0001	1	POLL	POLL
	0010	1	FHS	FHS
	0011	1	DM1	DM1
2	0100	1	undefined	DH1
	0101	1	HV1	undefined
	0110	1	HV2	undefined
	0111	1	HV3	undefined
	1000	1	DV	undefined
	1001	1	undefined	AUX1
3	1010	3	undefined	DM3
	1011	3	undefined	DH3
	1100	3	undefined	undefined
	1101	3	undefined	undefined
4	1110	5	undefined	DM5
	1111	5	undefined	DH5

defined for the device to be paged (slave unit). If no response, the master transmits a train on the remaining 16 hop frequencies in the wake-up sequence. The INQUIRY message is typically used for finding Bluetooth devices with an unknown address, it is very similar to the page message, but may require one additional train period to collect all the responses.

such as a temperature sensor. Two more low power modes are available, the SNIFF mode and the PARK mode. In the SNIFF mode, a slave device listens to the piconet at reduced rate, thus reducing its duty cycle. The SNIFF interval is programmable and depends on the application. In the PARK mode, a device is still synchronized to the piconet but does not participate in the

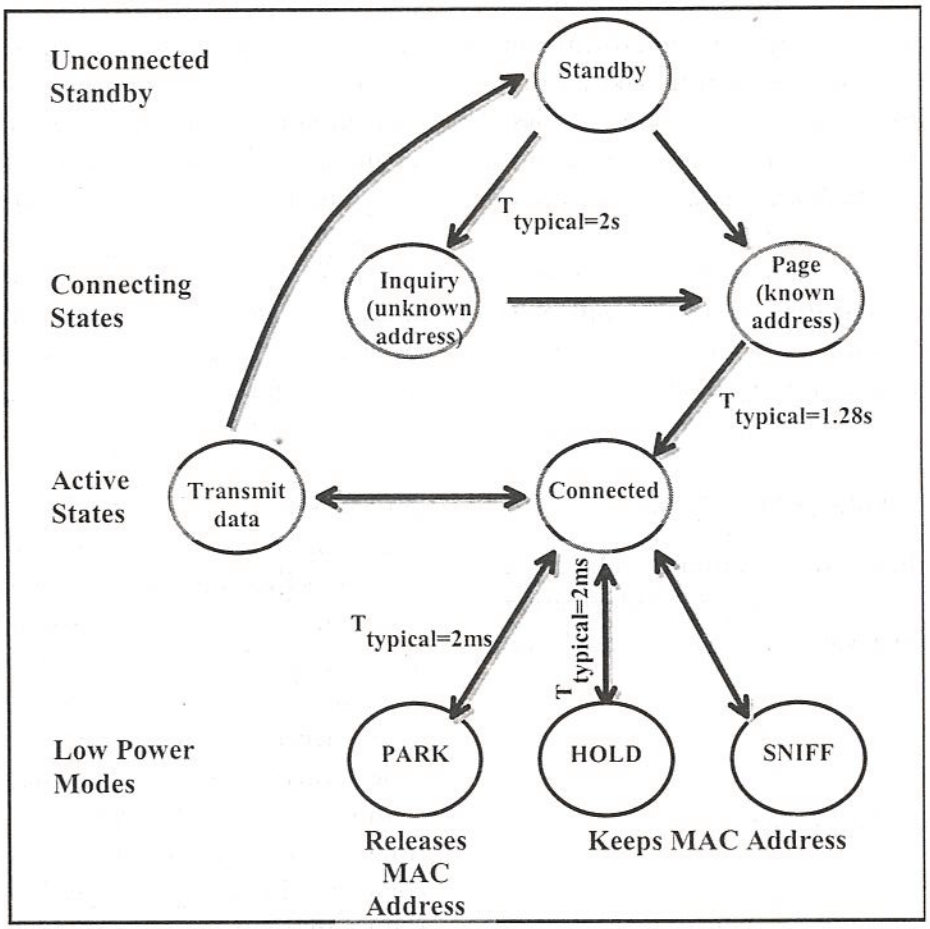

Figure 7: State diagram showing the possible Bluetooth states

A power saving mode can be used for connected units in a piconet if no data needs to be transmitted. The master unit can put slave units into HOLD mode, where only an internal timer is running. Slave units can also demand to be put into HOLD mode. Data transfer restarts instantly when units transition out of HOLD mode. The HOLD is used when connecting several piconets or managing a low power device

traffic. Parked devices have given up their MAC (AM_ADDR) address and occasional listen to the traffic of the master to re-synchronize and check on broadcast messages. If we list the modes in increasing order of power efficiency, then the SNIFF mode has the higher duty cycle, followed by the HOLD mode with a lower duty cycle, and finishing with the PARK mode with the lowest duty cycle.