
Bluetooth

Part 1: Overview

Kjell Jørgen Hole

UiB



Last updated 20.02.09

Mail: Kjell.Hole@ii.uib.no

URL: www.kjhole.com

- What is Bluetooth?
- The protocol stack
- Using Bluetooth
- Bluetooth profiles
- Java APIs (Application Programming Interfaces)

- Bluetooth is a low cost, low power, short-range radio technology
- Originally developed as cable replacement to connect mobile phones, headsets, portable computers, and Personal Digital Assistants (PDAs)
- Standardized wireless communication enables **Personal Area Networks** (PANs)
- Bluetooth specifications available at www.bluetooth.com

- Bluetooth stack defined by series of layers (see Figure 1-1)
- Usually implemented partly in hardware and partly in software
- Allows devices from different manufacturers to communicate with one another
- Enables applications to discover other Bluetooth devices, and determine what services they offer

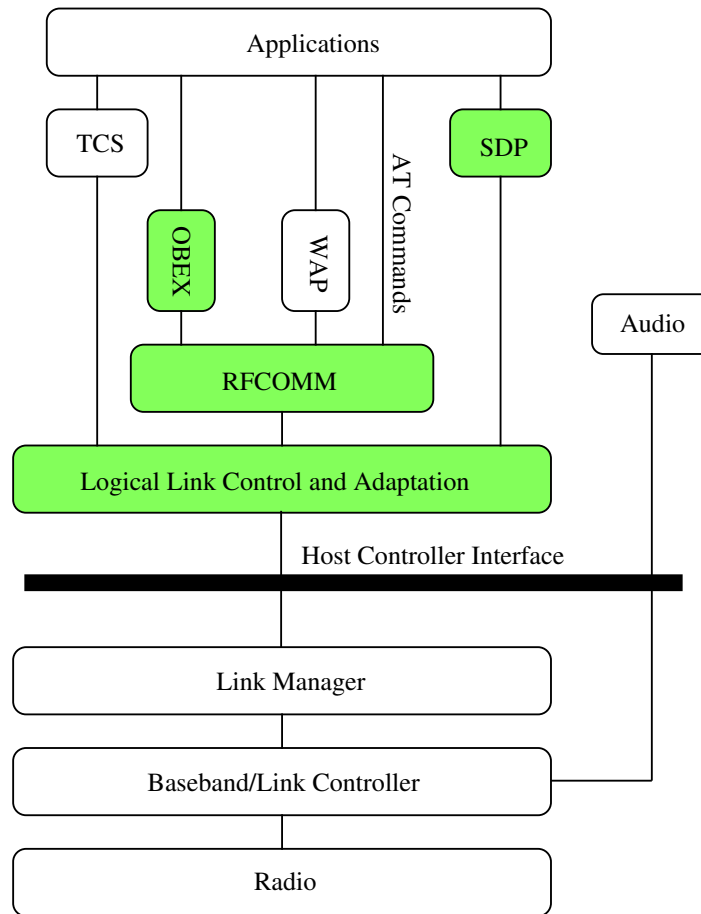


Figure 1-1 Protocol stack. The colored boxes represent the protocols addressed by Java APIs

Radio Modulates and demodulates data for transmission and reception on air

Baseband/Link Controller Controls the physical links via the radio, assembling packets and controlling frequency hopping

Link Manager Responsible for security and link set-up between Bluetooth devices

Host Controller Interface (HCI) Handles communication between a separate host and a Bluetooth module (also referred to as the *Bluetooth controller*)

Logical Link Control and Adaptation (L2CAP) Multiplexes data from higher layers, converts between different packet sizes

RFCOMM Emulates an RS-232 like serial interface

WAP and OBEX Adopted protocols

SDP (Service Discovery Protocol) Lets Bluetooth devices discover what services other Bluetooth devices support

TCS (Telephony Control Protocol Specification) Provides telephony services

- Operates at 2.4 GHz in the globally available, unlicensed (i.e., free) *Industrial, Scientific, and Medical* (ISM) band
- Handheld Bluetooth devices require antennas which radiate in a pattern close to a sphere, i.e., the performance of the devices should appear to be independent of operating angle
- Bluetooth signaling must be robust since there are many other systems using the same spectrum, thus creating interference

- Operating band (2.400–2.4835 GHz) divided into 79 channels with carrier frequencies $f = 2402 + k$ MHz, $k = 0, \dots, 78$
- Channel spacing is 1 MHz. To comply with out-of-band regulations, 2 MHz and 3.5 MHz lower and upper guard bands are used
- Gaussian Frequency Shift Keying (GFSK) modulation with one bit per symbol
- The symbol rate is 1Mps, resulting in a gross data rate of 1 Mbps

Enhanced Data Rate (EDR)

KJhole.com

- In June 2004, EDR was introduced to increase the gross data rate from 1 Mbps to 2 Mbps or 3 Mbps
- EDR uses PSK modulation and has two variants: $\pi/4$ -DQPSK which is mandatory if EDR is supported, and 8DPSK which is optional
- EDR is backwards compatible with the previous specification

- *Frequency Hopping Spread Spectrum* (FHSS) for robust communication
- 625 microseconds* time slots (1600 hops per second)
- One hop per packet (every slot, every 3 slots, or every 5 slots)
- Re-transmission of lost data packets

*micro = 10^{-6}

Transmit Power Classes

KJhole.com

Class	Max. output power	Range	Power control
1	100mW (20 dBm)	100m+	mandatory
2	2.5mW (4 dBm)	10m	optional
3	1mW (0 dBm)	1m	optional

- Power control reduces interference and power consumption

- The link controller and baseband are responsible for:
 - determine which data bit stream to transmit
 - connection establishment (inquiry & paging)
 - frequency hop selection
 - logical transports (ACL, SCO, eSCO)
 - medium access control
 - power modes
 - security algorithms

Masters and Slaves

KJhole.com

- Each Bluetooth device is a **Master** or **Slave**. A Master initiates an exchange of data and the Slave responds to the Master
- Communicating Bluetooth devices must use same sequence of frequency hops
- Slaves synchronize to frequency hop sequence used by Master

Frequency Hop Sequence

KJhole.com

- Every Bluetooth device has unique device (48-bit IEEE MAC) address and clock
- Each Slave receives Master's address and clock. Slave uses this information to calculate frequency hop sequence
- Usually, all 79 channels are used during the frequency hopping, but an *adapted* hopping sequence utilizing a minimum of 20 channels may also be employed

- Time Division Multiplexing (TDM) is used to divide the total bandwidth between Bluetooth devices
- Master assigns time slots to Slaves
- Packets are joined together in *transmit* and *receive* pairs; a packet pair can be 2, 4, 6, 8, or 10 slots long

Piconet Group of Bluetooth devices joined together into a short-range network by Bluetooth links. The group is synchronized to the timing and hopping sequence of the Master (see Figure 1-2)

Scatternet Group of Bluetooth piconets joined together by devices that are in more than one piconet. (Routing of packets between piconets is not defined in version 2.1 of the Bluetooth standard)

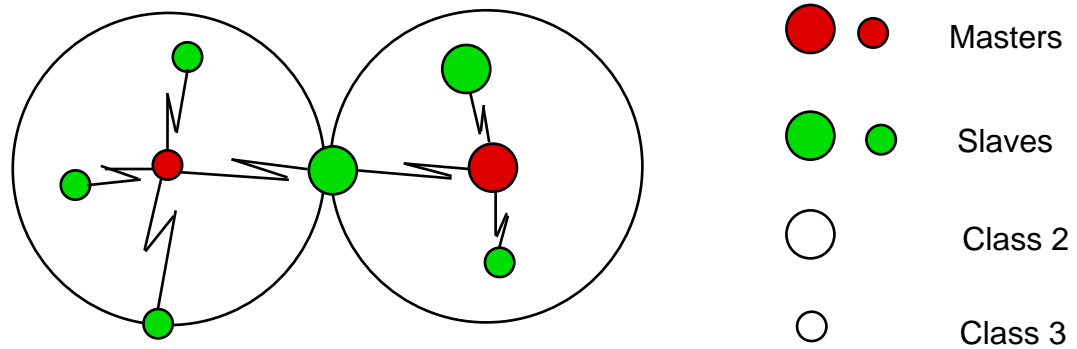


Figure 1-2 Scatternet consisting of two piconets with different power class devices

- The Slaves in a piconet only have links to the Master; there are no direct links between Slaves in a piconet
 - Master may become bottleneck for data transfer
 - One piconet can be split into two piconets by one Slave becoming a Master. This may increase the aggregated throughput
- There are no more than seven active Slaves in a piconet. Many more slaves can remain connected in a parked state

- A device present in more than one piconet must time-share, spending a few slots on one piconet and a few slots on the other
- A device may not be Master of two different piconets since all Slaves in a piconet are synchronized to the Master's hop sequence
- *Piconets making up a scatternet do not coordinate their frequency hopping*
- Unsynchronized piconets in an area will randomly collide on the same frequency

- **SCO** (Synchronous Connection-Oriented) transports for 64 kbps voice communication. Use reserved time slots at regular intervals
- **eSCO** (Extended SCO) constant rate transports locked to the piconet clock. Different data rates available. Use reserved time slots. Limited retransmission in case of error
- **ACL** (Asynchronous Connection-Oriented)* transports for data communication. No reserved time slots

*Originally, the abbreviation ACL stood for Asynchronous ConnectionLess

ACL Data Packets

KJhole.com

- ACL data packets contain a 72-bit access code, 54-bit header, 16-bit Cyclic Redundancy Checksum (CRC), and varying amount of data
- The largest packet, i.e. the DH5 packet, stretches over five slots
- Maximum data rate at application level is about 650 kbps

- SCO links operate at 64 kbps
- Can have up to three voice transports at once
- SCO transports are not suitable for delivering CD-quality sound (!)

- Bluetooth provides *authentication* and *confidentiality*. There is no support for data *integrity*
- A Personal Identification Number (PIN) is entered into both client and server device for mutual authentication
- Public domain cipher algorithm SAFER+ generates 128-bit cipher keys from 128-bit plaintext
- (Symmetric-key) stream cipher used for link encryption

- The Link Manager Protocol (LMP) has a large set of control messages denoted LMP Protocol Data Units (PDUs)
- Special security related PDUs have been defined to carry out
 - pairing
 - authentication
 - encryption
 - changing of link key

Using Bluetooth, Step 1

Three steps are carried out when a laptop (*device A*) wants to utilize a mobile phone (*device B*) as a modem using a Dial Up Networking (DUN) connection.

Step 1 Discovering Bluetooth device (see Figure 1-3):

- device A transmits inquiry packets
- device B replies with **Frequency Hop Synchronization** (FHS) packet which contains device class information
- device A may scan for other devices using the same procedure

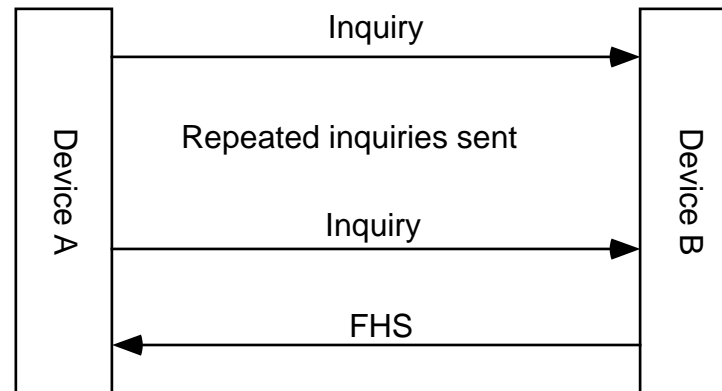


Figure 1-3 Discovering a Bluetooth device

Step 2 Connecting to service discovery database (see Figure 1-4):

- ACL baseband connection is established
- **Logical Link Control and Adaption Protocol (L2CAP)** connection is set up over ACL channel
- L2CAP adds Protocol and Service Multiplexor (PSM) to L2CAP packets to distinguish between different higher-layer protocols and services (PSM=0x0001 for service discovery)
 - PSM is similar to port number in IP network

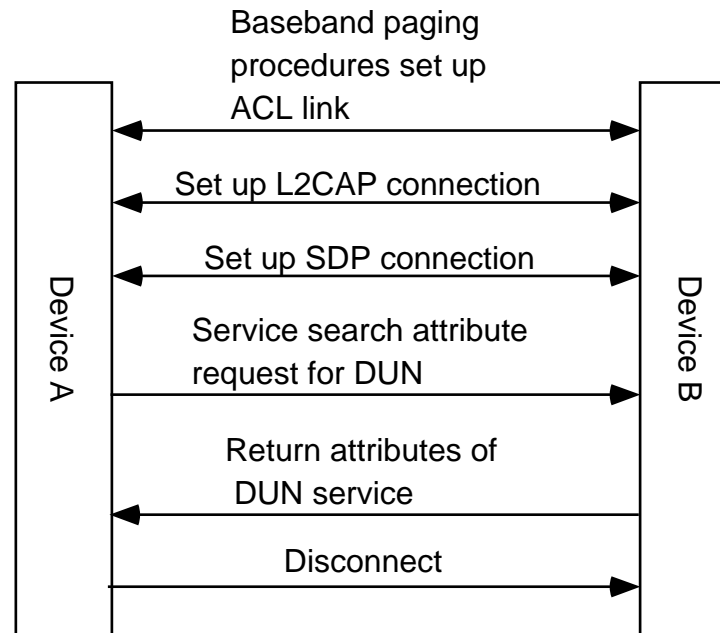


Figure 1-4 Retrieving information on services

- **Service Discovery Protocol** (SDP) connection over L2CAP channel
- device A asks device B for DUN information
- device A receives DUN information from B's service discovery database
- device A disconnects because it wants to collect service discovery information from other devices in the area

Step 3 Connecting to Bluetooth service (see Figure 1-5):

- ACL link is set up
- device A utilizes **Link Management Protocol** (LMP) to configure link
- L2CAP connection using the RFCOMM protocol for RS-232 serial cable emulation is set up (PSM=0x003)
- DUN connection is set up using RFCOMM connection

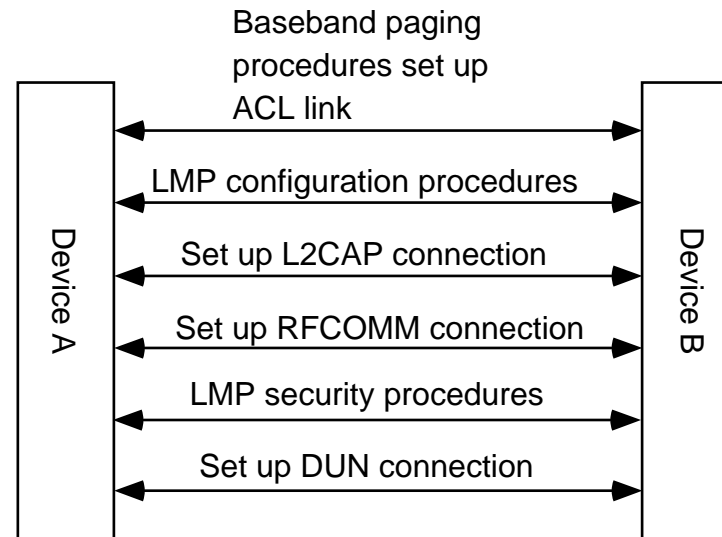


Figure 1-5 Connecting to a Dial Up Networking service

- Device manager (also called Bluetooth control center) needed to manage links at RFCOMM and SDP level. Not defined by Bluetooth specification
- Implementation and complexity of device manager depend on requirements of Bluetooth device
- Device manager can provide fault, accounting, configuration, performance, and *security management*

Profile A set of rules for how to use the Bluetooth protocol stack in a device

- different profiles are defined for different types of devices
- profiles ensure that different device types are able to interoperate

As an example, a Bluetooth headset purchased from one manufacturer is able to interwork with a Bluetooth enabled mobile phone purchased from another manufacturer

Generic Access Profile (GAP) Most basic Bluetooth profile. All other profiles are built upon it and uses its facilities. GAP defines the generic procedures related to establishing connections between two devices, including

- discovery of Bluetooth devices
- link management and configuration
- procedures related to use of different security levels

Java ME

- The document *Java APIs for Bluetooth Wireless Technology* (**JSR-82**) defines the Bluetooth APIs for the Java Platform, Micro Edition (Java ME), <http://www.jcp.org/en/jsr/detail?id=82>
- JSR-82 APIs enable open, third party application development
- APIs designed to operate on top of the *Connected, Limited Device Configuration* (**CLDC**), <http://www.jcp.org/en/jsr/detail?id=139>
- APIs can be used together with *Mobile Information Device Profile* (**MIDP**), <http://www.jcp.org/en/jsr/detail?id=118>

- JSR-82 APIs are based on the Bluetooth specification version 1.1. The APIs contain a total of 21 classes
- APIs are designed to operate on Bluetooth devices characterized as follows:
 - 512K minimum memory available for Java platform. Application memory requirements are additional
 - Compliant implementation of CLDC
- APIs only support data transmissions (no voice transmissions)

Supported Bluetooth Protocols

KJhole.com

- JSR-82 describes APIs for the following Bluetooth protocols:
 - L2CAP
 - RFCOMM
 - SDP
 - OBEX (OBject EXchange protocol)

Supported Bluetooth Profiles

KJhole.com

- JSR-82 supports the following Bluetooth profiles:
 - Generic Access Profile (GAP)
 - Service Discovery Application Profile (SDAP)
 - Serial Port Profile (SPP)
 - Generic Object Exchange Profile (GOEP)

- The functionality of the JSR-82 APIs falls into three categories:

Discovery Discover devices, services and register services

Communication Establish RFCOMM, L2CAP and OBEX connections

Device Management Managing and controlling connections

- The JSR-82 document describes APIs for each category

- Bluetooth is a low power, short-range radio technology for wireless communications
- Large effort made to ensure
 - high usability
 - low cost
 - interoperability between devices from different manufacturers
- There exist standard Java APIs for Bluetooth devices

