KUNGL TEKNISKA HÖGSKOLAN
Institutionen för mikroelektronik och informationsteknik

2G1330 Mobile and Wireless Network Architectures

Bluetooth

Lecture notes of G. Q. Maguire Jr.

Last modified: 2002.03.14:11:57

Lectures 7 & 8

- Bluetooth: Piconets, Scatternets

Bluetooth name comes from Danish king Harald Blåtand (Bluetooth), credited with uniting the Scandinavian people during the 10th century.

The idea was that Bluetooth wireless technology would unite personal computing devices.

Bluetooth™

Bluetooth is a trademark owned by the Bluetooth SIG, Inc., USA.

Bluetooth Special Industry Group (SIG) formed in winter of 1998 by Ericsson, IBM, Intel, Nokia, and Toshiba.

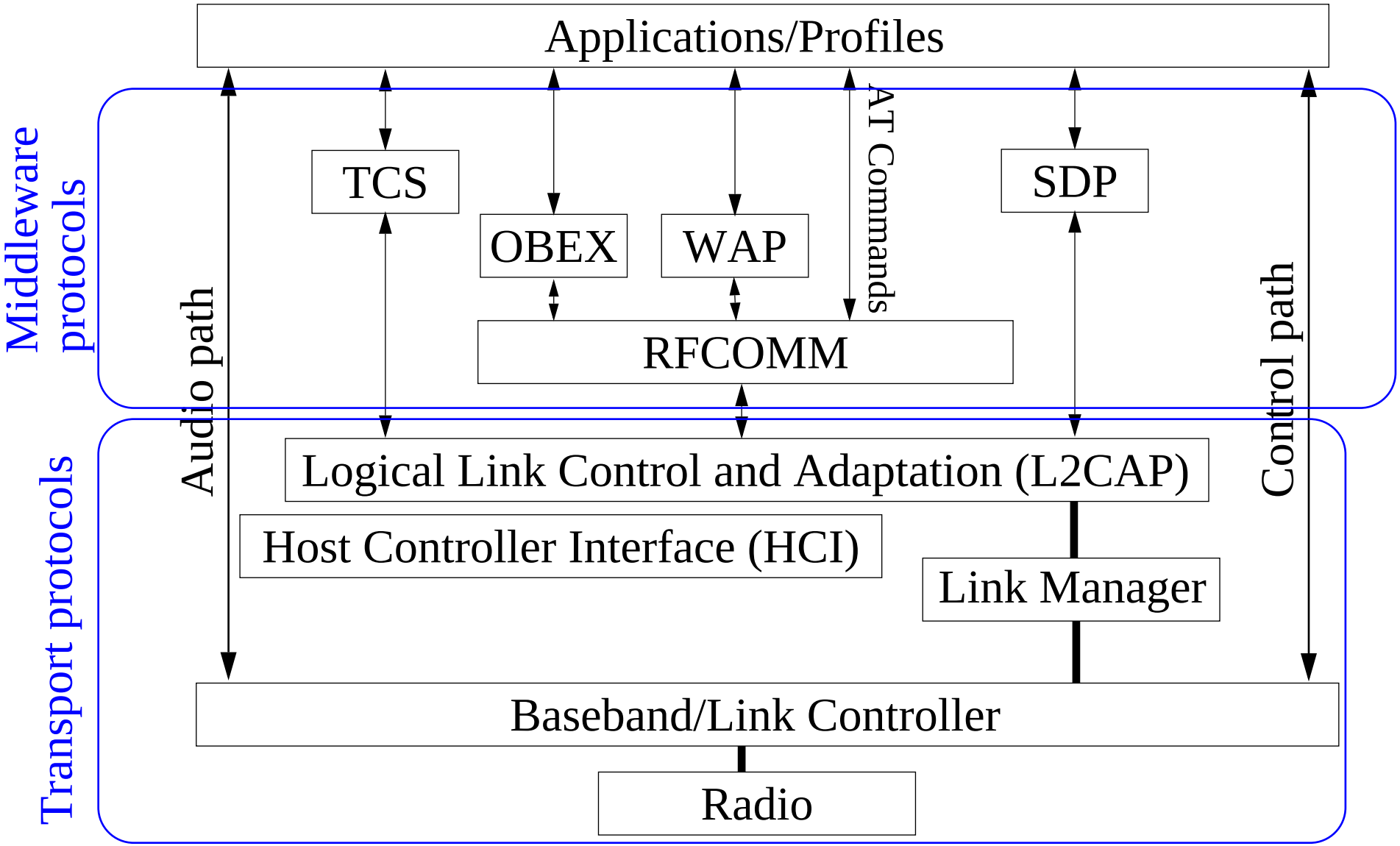
Goals

- low cost
- low power
- primarily a cable replacement (to connect mobile phones to headsets)
  - There are those who believe it can be used as a Wireless Personal Area Network (WPAN), hence it was the basis for IEEE 802.15.

Using:

- short-range radio technology
- ad hoc networking
- dynamic discovery of other Bluetooth devices & the services they offer

Bluetooth protocol stack



Physical Layer

- Uses 2.4 GHz unlicensed Industrial, Scientific, and Medical (ISM) band (globally portions of this band are available)
  - many other systems using the same spectrum
    - interference to other systems
    - interference from other systems
  - 2.400-2.4835 GHz, i.e., 83.5 MHz divided into 79 channels with carrier frequencies f = 2402 +k MHz, k = 0, … , 78; Channel spacing is 1 MHz
  - Gaussian Frequency Shift Keying (GFSK) modulation with one bit per symbol
  -
- uses fast (1600 hops/s) frequency hopping spread spectrum (FHSS)
  - 625 microsecond long time slots
  - one hop per packet, but a packet can be 1 slot, 3 slots, or 5 slots long

Tranmit Power

- Low transmit power
- original goal was a 10m radius of operation, but some thought about using Bluetooth for longer ranges ⇒ Transmit Power Classes

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

- most manufacturers producing Class 3 radios
- power control is to reduce both interference and power consumption

Masters vs. Slaves

Each Bluetooth device is a Master or Slave:

- master initiates exchange of data and the slave responds to the master
- in order to communicate devices must use same sequence of frequency hops, hence slaves synchronize to hop sequence of master
- master assigns an **A**ctive **M**ember **add**ress (AM_ADDR) to the slaves participating in active communications within the piconet

Additional devices may be registered with the master and be invited to become active as necessary -- their state is called "**parked**"

Devices not currently associated with any piconet are in **stand-by mode**.

Frequency Hop Sequence

Each device has a 48 bit IEEE MAC address (called a Bluetooth device address (BD_ADDR)) and a local free-running 28-bit clock that ticks once every 312.5 µs (which corresponds to half the residence time in a frequency when the radio hops at the nominal rate of 1,600 hops/sec)

Each slave receives master's address and clock, then uses this to calculate frequency hop sequence

Time Division Multiplexing (TDM)

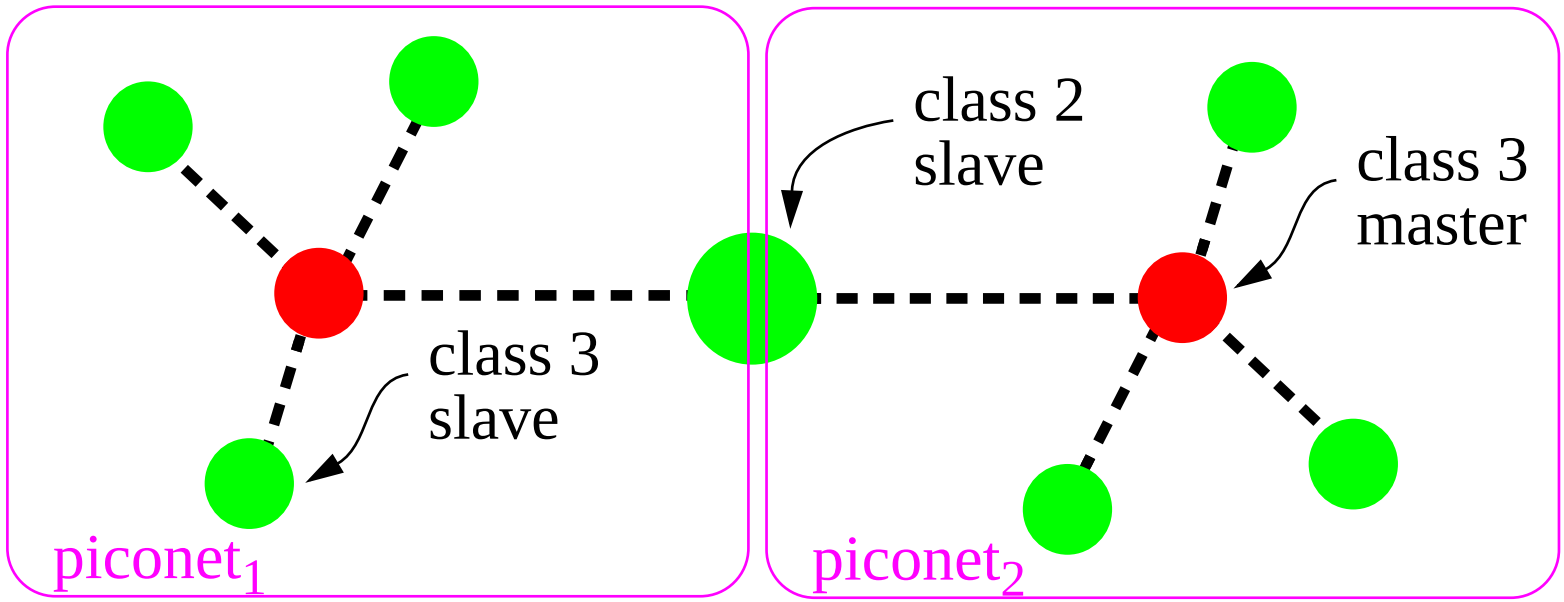
Divide the total bandwidth between Bluetooth devices using a given hop sequence

- Master assigns time slots to slaves
- packets are joined together in transmit and receive pairs; master and slaves alternate in time-division duplex (TDD)

Network Topology

Piconet	subnet of Bluetooth devices, synchronized to the timing and hopping sequence of a master • slaves only communicate with the master • maximum of 7 slaves in a piconet (as there are only 3 address bits!)
Scatternet	multiple Bluetooth piconets joined together by devices that are in more than one piconet • Routing of packets between piconets is not defined)

Scatternet



Scatternets

If a device is present in more than one piconet, it must time-share, spending a few slots in one piconet and a few slots in 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, thus if the slaves were all synchronized with a single master -- they would be part of the **same** piconet!

This means that piconets making up a scatternet do **not** coordinate their frequency hopping ⇒ unsynchronized piconets in an area will randomly collide on the same frequency.

Voice + Data support

As an important application of Bluetooth was a cable replacement between handset and headset and this was developed in a telecom company's development lab ⇒ synchronous voice support was the focus of the link protocol design

- Synchronous Connection Oriented (SCO) links for voice
  - circuit-switched connections - 64 kb/s in each direction per voice channel (using their own voice coding or) using reserved slots
  - up to three voice channels active at one time (may be to 1, 2, or 3 slaves)
  - ~78% overhead for data! (this is without FEC)
- Asynchronous Connectionless (ACL) links for data
  - ACL Data Packets: 72-bit access code, 54-bit header, 16-bit Cyclic Redundancy Checksum (CRC), and varying amount of data
  - with largest packet (Data High rate, DH5, packet stretching over five slots) ⇒ maximum data rate of ~650 kb/s
  - a best effort delivery service - maintains integrity by using retransmissions and sequence members, as well as forward error correction (FEC) **if** necessary
  - a master can have an ACL link to each of several slaves, but only one per slave
  - Broadcast packets: packets that are not addressed to a specific Slave

Baseband

Baseband controls the radio and is responsible for low level timing, error control, and management of link during a single data packet transfer

Packet types:

- SCO, ACL - carrying payload
- ID packet consists of access code, used during re-connection
- NULL packet consists of access code and header, used for flow control or to pass ARQ
- POLL packet same structure as NULL packet, must be acknowledged
- FHS (Frequency Hop Synchronization)

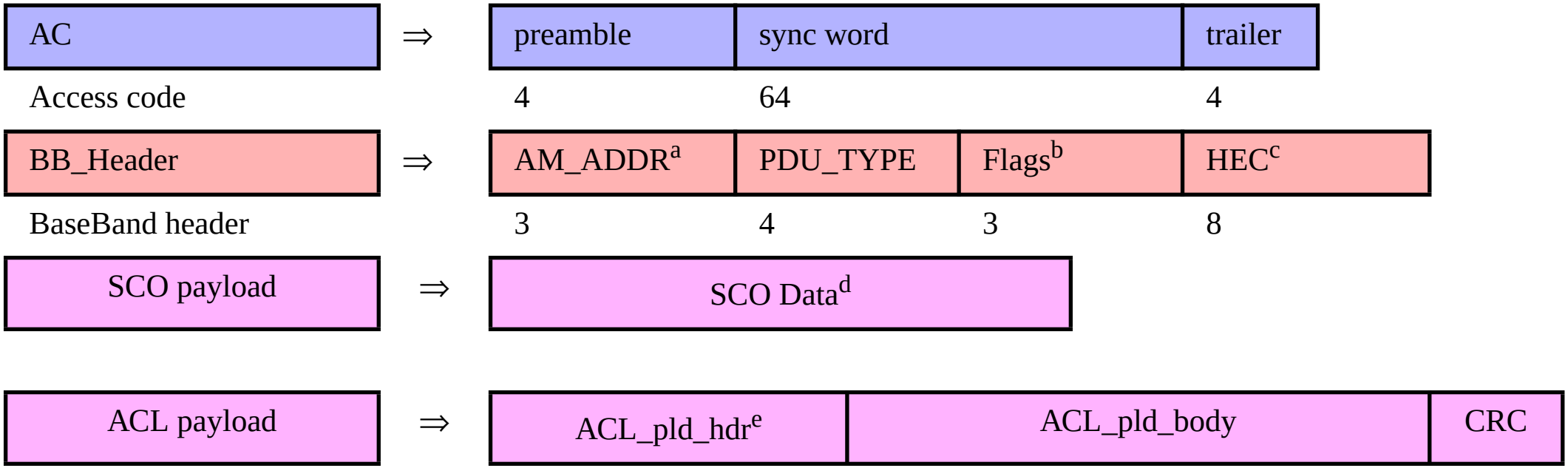
Baseband Packet formats

LSB				MSB
ID	AC			
(bit count)	68 or 72			
POLL/NULL	AC	BB_Header		
	68 or 72	54 (1/3 FEC)[a]		
FHS	AC	BB_Heade	FHS payload	
	68 or 72	54 (1/3 FEC)	240 (2/3 FEC)	
ACL/SCO	AC	BB_Heade	ACL or SCO payload	
	68 or 72	54 (1/3 FEC)	0-2744 ({1,2,3[b]}/3 FEC)	
DV	AC	BB_Heade	SCO payload	ACL payload
	68 or 72	54 (1/3 FEC)	80	32-150 (2/3 FEC)

a. 54 bits includes the FEC bits (there are 18 bits of information with each bit repeated 3 times)

b. 3/3 FEC implies no FEC

Baseband Packet formats



a. Broadcast packet has address zero

b. Flow (=1 means receive buffer is full), ARQN (ACK represented by ARQN=1 and NAK by ARQN=0), SEQN (alternating bit)

c. Header error check (HEC)

d. 30 bytes (240 bits), error control code with rate 1/3, 2/3, or 1 (no FEC) used for source data size of 10, 20, or 30 bytes; note BB_Header flags for ARQN and SEQN are not used - since there is no flow control or retransmission, similarily the HEC is not used

e. L_CH (Logical CHannel) Field (3 bits) indicates whether payload is start or continuation of message, Flow field (1 bit) controls for data transfer at L2CAP level, Length field (8 bits) indicates the number of data bytes in the payload' header ends with 4 undefined bits

Synchronization Word Algorithm

1. Get 24-bit Lower Address Part (LAP) of Bluetooth device address (48 bit IEEE MAC address)
2. Append 6-bit Barker sequence to improve auto-correlation properties
3. XOR with bits 34 to 63 of full length, 64-bit Pseudorandom Noise (PN) sequence
4. Encode resulting 30-bit sequence with (64,30) BCH (Bose-Chaudhuri-Hocquenghem) block code to obtain 34 parity bits
5. 34-bit parity word XOR'd with the remaining bits, 0 to 33 of PN sequence to remove cyclic properties of block code

Note: 34 bits BCH parity word exhibits **very high auto-correlation** and **very low co-correlation** properties, therefore a correlator can be used to obtain a match between the received and expected (reference) synch word

Security

Some think that the high speed, pseudo-random frequency hopping algorithm makes it difficult to listen in on a connection - but of course this is false, because once you know the master's MAC address and clock you can calculate the next hop too!

Authentication and negotiation for link encrypting are both part of the Link Manager Protocol (LMP) specification.

- authentication is based on a challenge/response mechanism based on a common shared secret, a link key is generated through a user-provided PIN
- link level encryption using a public domain cipher algorithm SAFER+[1] generates 128-bit cipher keys from 128-bit plain text

1. J. L. Massey, On the Optimality of SAFER+ Diffusion, available at *http://csrc.nist.gov/encryption/aes/round1/conf2/papers/massey.pdf*

Link Control Protocol (LCP)

- configures and controls baseband
- packet level access control - determines what packet is going to be sent next
- high level operations: inquiry and paging
- configures and controls multiple links between devices and piconets
- does **not** require its own packets, but uses the (ARQN and SEQN) bits in baseband packets for SCO and ACL links to signal between link controllers - thus forming a logical LC (Link Control) channel

Link Control states

State	Description
Standby	inactive, radio not switched on
Inquiry	device tries to discover all Bluetooth enabled devices in the close vicinity; uses a special fast hopping sequence; FHS packets with device information, such as clock, frequency hop sequence, and BD ADDR, received from available devices; ⇒ a list of all available devices
Inquiry Scan	devices periodically enter the inquiry scan state to make themselves available to inquiring devices; a special slow hopping sequence used
Page	master enters page state and transmits paging messages to slave using access code and timing information which it learned earlier
Page Scan	device periodically enters page state to allow paging devices to establish connections
Connection-Active	Slave synchronizes to master's frequency hop and timing sequence. Master transmits a POLL packet to verify link, Slave sends NULL packet in reply
Connection-Hold	device ceases to support ACL traffic for a period of time, keeps Active Member address (AM_ADDR)
Connection-Sniff	device listens in pre-defined time slots only
Connection-Park	device listens for traffic only occasionally, gives up its AM address

Link Manager

Translates commands from Host Controller Interface (HCI) into operations at baseband level to implement the following operations:

- attaching Slaves to a piconet, and allocating active member addresses (AM addr)
- tearing down connections when slaves leave piconet
- configuring links, e.g., controlling Master/Slave switches
- establishing ACL and SCO links
- putting connections one of the low-power modes
- communicates with other LMs using the Link Management Protocol (LMP) which is a set of messages, or Protocol Data Units (PDUs), whose payloads contain the following fields:
  - single bit Transaction Identifier equal to 0 (1) for PDU sent from Master (Slave)
  - Operation Code (OpCode) defining type of message being sent
  - message parameters
  - PDUs sent as single slot packets on link management logical channel (L_CH =3)

Host Controller Interface (HCI)

- interface between a host and a Bluetooth module
- having a standard interface enables Baseband and Link Manager to run on a processor in the Bluetooth module while higher layers and applications running on host
- Bluetooth module can wake the host via a message across this interface

HCI Transport Layer

Three different transport interfaces are defined to transfer HCI packets from the host to the Bluetooth module:

USB	Universal Serial Bus
RS-232	serial interface with error correction
UART	Universal Asynchronous Receiver Transmitter, a serial interface without error correction

Logical Link Control and Adaptation Protocol (L2CAP)

L2CAP only transfers data and all applications **must** use L2CAP to send data.

provides:

- multiplexing to allow several higher layer links to pass across a single ACL connection
- segmentation and reassembly to allow transfer of packets larger than lower layers support
- Quality of Service (QoS) management for higher layer protocols

L2CAP Signalling

labels packets with channel numbers

L2CAP entities communicate with each other using control channels with a special channel number (used for connecting, configuring, and disconnecting L2CAP connections)

packet contains a length field (2 bytes), a channel identifier (2 bytes), and a data field (0 .. 65535 bytes)

L2CAP Command

OpCode	identifying contents of command
Identifier	used to pair up requests and responses
Length	of data field

More than one command can be sent within a L2CAP packet

Configuring a Connection

Parameters which can be configured are:

- Maximum Transmission Unit (MTU) < 65,535 bytes
- Flush timeout -- time (in milliseconds) a device will spend trying to transmit an L2CAP packet before it gives up
- QoS option can select best effort, or a guaranteed QoS

Disconnecting and Timeouts

Two ways for an L2CAP channel to be closed down:

- disconnection request from higher layer protocol or service
- time out: every time L2CAP sends a packet, a Response Timeout Expired (RTX) time is started; if the RTX timer expires before a response is received, the channel may be disconnected

For A to talk to B

Step 1: Discovering a Bluetooth device:

- device A transmits one or more inquiry packets[1]
- device B replies with Frequency Hop Synchronization (FHS) packet which contains device class information (including its BD_ADDR)

Step 2: Connecting to service discovery database:

- 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)
- Service Discovery Protocol (SDP) connection over L2CAP channel
- device A receives Dial-Up Networking (DUN) info from B's service discovery database
- device A disconnects

Step 3: Connecting to Bluetooth service:

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

---

1. A piconet master may explicitly page devices to join its piconet; if it knows their BD_ADDR it can skip the inquiry process and directly paging the device

Service Discovery Protocol (SDP)

- only provides information about services, does not provide access to these services
- “optimized” for usage by devices with limited capabilities over wireless links
  - uses binary encoding of information
  - unique identifiers (UUIDs) describe services and attributes of these services such that you don’t need a central registration authority for registering services
  - generally UUIDs are 128 bits long; however, for known services 16-bit and 32-bit UUIDs may also be used.

RFCOMM Protocol

- provides a serial interface over the packet-based transport layers
- emulates the signals on the nine wires of an RS-232 cable
- based on the ETSI 07.10 standard (also used by GSM terminals), allows multiplexing (via L2CAP) several serial ports over a single transport
  - supports flow control on individual channels
  - has a reserved Protocol and Service Multiplexer (PSM) value used by L2CAP to identify RFCOMM traffic
- no error control
- enables legacy applications -- written to operate over serial cables -- to run without modification

RFCOMM Frame Types

Five frame types (the first 4 are control frames):

SABM	Start Asynchronous Balanced Mode (startup command)
UA	Unnumbered Acknowledgement (response when connected)
DISC	Disconnect (disconnect command)
DM	Disconnected Mode (response to a command when disconnected)
UIH	Unnumbered Information with Header check • each RFCOMM channel has a Data Link Connection Identifier (DLCI) • UIH frames with DLCI = 0 are used for control messages, while DLCI ≠ 0 are used data

Telephony Control Signaling (TCS) Protocol

TCS-AT	Telephony control can be performed using the AT command set use the RFCOMM to send and receive control signaling based on the AT command set (for example to implement a dialer application)
TCS-BIN	(BIN stands for the binary encoding of information), that runs directly on top of L2CAP; supports normal telephony control functions such as placing and terminating a call, sensing ringing tones, accepting incoming calls, etc. TCS-BIN supports point-to-multipoint communications as well, for example, a cordless base station can pass the ringing signal of an incoming call to several cordless headsets associated with the base station.

Bluetooth Profiles

- specifications for building interoperable applications
- All profiles depend on the Generic Access Profile (GAP) -- defines the basic rules and conditions for connecting devices with each other and establishing Bluetooth links and L2CAP channels.

Profile	Description
serial port profile	defines how RFCOMM runs on top of the Bluetooth transport protocols
generic object exchange profile	defines how objects can be exchanged using the OBEX protocol running on top of RFCOMM

add more profiles - such as LAN access

Management

- needed to manage links, but not defined by Bluetooth spec!
- could provide fault, accounting, configuration, performance, and security management
- link level encryption using a public domain cipher algorithm SAFER+ generates 128-bit cipher keys from 128-bit plain text

Low Power Modes

sniff mode	a slave agrees with its master to periodically listen for the master's transmissions; the period is configured through LMP transactions
hold mode	a device (in a piconet) agrees to remain silent (in that particular piconet) for a given amount of time; note: **keeps** its temporary address, AM_ADDR
park mode	a slave device agrees with its master to park until further notice; **relinquishes** its active member address, AM_ADDR, periodically listens to beacon transmissions from the master • device can either be invited back (by the master) to active communications using a broadcast transmission during a beacon or • if the slave wants to be unparked, it sends a message to the master in the slots following the beacon

Although the radio is often the biggest power drain on a Bluetooth device, the voltage controlled oscillator (for the Bluetooth clock) also consumer power and can be shut off -- instead you can use a less accurate lower power oscillator when the accuracy of the normal oscillator is not needed (for example when sleeping)

IEEE 802.15 standard

Bluetooth proposal chosen to serve as the baseline

- IEEE 802.15.1draft standard is in its final stages
- IEEE 802.15.2 task group studies coexistence issues between 802 wireless technologies
- IEEE 802.15.3 task group developing standards for high-rate radios (>20 Mb/s)
- IEEE 802.15.4 task group developing standards for low-rate radios (<200 kb/s)

Further reading

The lecture notes are based on material from:

[1] “Bluetooth: Part 1: Overview”, Kjell Jørgen Hole <Kjell.Hole@ii.uib.no>, NTNU, UiB, http://kjhole.com/Bluetooth/Downloads.html

which is in turn based on Ch. 1, 2, and 3 of:

[2] Bluetooth 1.1: Connect Without Cables by Jennifer Bray and Charles F. Sturman

[3] C. Bisdikian, “An overview of the Bluetooth Wireless Technology”, IEEE Communications Magazine, pp. 86-94, Dec. 2001.

[4] Bluetooth specification, http://www.bluetooth.com

[5]