What is Bluetooth?

- A short range radio link intended to replace the cables connecting portable and/or fixed electronic devices
  - Robust, low complexity, lower power, low cost
- Operates in unlicensed ISM band at 2.4GHz
  - Frequency hopping to combat interference and fading
  - Symbol rate: 1Ms/s, slot duration: 625µs, 1600 hops/s
  - TDD (time division duplex): master starts transmission in even-numbered slots only; slave starts transmission in odd-numbered slots only
- A combination of packet and circuit switching
  - Slots can be reserved for synchronous packets
What is Bluetooth? (cont’d)

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

- Point-to-point connection to point-to-multipoint connection provided
  - Piconet is formed to support point-to-multipoint connection
  - Several piconets with overlapping coverage areas form a scatternet.

Network topology in Bluetooth

- Each piconet has only one master node
- Slaves can participate in different piconets
- A master in one piconet can be a slave in another piconet.
Protocol stack (cont’d)

- RF layer specifies the radio modem used for transmission and reception
- The baseband layer controls the radio, and specifies the link control at the bit and packet level
- The link management protocol (LMP) handles configuration and control of the Bluetooth baseband links
- The logical link control and adaptation protocol (L2CAP) provides connection-oriented and connectionless data services to the upper layer protocols
- Service discovery protocol (SDP) finds the characteristics of the services and connects two devices to support a service
- Telephony control protocol specification (TCS) defines the call control signaling and mobility management for the establishment of speech
- RFCOMM is a cable replacement protocol that emulates the standard RS-232 control and data signals over Bluetooth baseband
Radio specification of Bluetooth

- Frequency bands and channel arrangement
  - The Bluetooth system operates in the 2.4GHz ISM (Industrial Scientific Medicine) band: 2400 – 2483.5 MHz.
  - 79 channels allocated, each with spacing of 1 MHz.
    - RF channels: \( f = 2402 + k \text{ MHz}, k = 0, 1, \ldots, 78 \)
  - Guard band is used at the lower and upper band edge: lower guard band: 2 MHz; upper guard band: 3.5 MHz

- Transmission power:
  - Three power classes defined: class 1, 100 mW (20dBm); class 2, 2.5mW (4dBm); class 3, 1mW (0dBm)

- Modulation: GFSK (Gaussian Frequency Shift Keying)

Physical channel

- Frequency band and RF channels

<table>
<thead>
<tr>
<th>Country</th>
<th>Frequency Range</th>
<th>RF Channels</th>
</tr>
</thead>
<tbody>
<tr>
<td>Europe” &amp; USA</td>
<td>2400 - 2483.5 MHz</td>
<td>( f = 2402 + k \text{ MHz} )</td>
</tr>
<tr>
<td>Japan</td>
<td>2471 - 2497 MHz</td>
<td>( f = 2473 + k \text{ MHz} )</td>
</tr>
<tr>
<td>Spain</td>
<td>2445 - 2475 MHz</td>
<td>( f = 2449 + k \text{ MHz} )</td>
</tr>
<tr>
<td>France</td>
<td>2446.5 - 2483.5 MHz</td>
<td>( f = 2454 + k \text{ MHz} )</td>
</tr>
</tbody>
</table>

- Channel definition: represented by a pseudo-random hopping sequence through the RF channels. The hopping sequence is determined by the master
  - Channel is divided into time slots where each slot corresponds to an RF hop frequency
  - Hopping rate: 1600 hops/s
Physical channel (cont’d)

- Each time slot is 625 µs in length.
- TDD (time division duplex) used. Master starts transmission in even-numbered slots. Transmission may be extended over up to 5 time slots.
- RF hopping frequency should remain fixed for the duration of the packet.

Multi-slot packet transmission
Physical links

- Two types of links defined in Bluetooth:
  - Synchronous connection-oriented (SCO) link: a point-to-point link between a master and a slave using reserved slots.
  - Asynchronous connection-less (ACL) link: a point-to-multipoint link between the master and all slaves.
- SCO link: symmetric, point-to-point link
  - Up to three SCO links to the same slave or different slaves
  - Master sends SCO packets at regular intervals, TSCO
  - The SCO slave is allowed to respond with an SCO packet in the following slave-to-master slot, even if it fails to decode the slave address
  - The next master-to-slave slot $CLK(k+1) = CLK(k) + TSCO$
    (CLK = master clock that determines the timing on a Bluetooth piconet)

Physical links (cont’d)

- ACL link: packet-switched connection between the master and all slaves
  - Both asynchronous and isochronous* services supported
  - Between a master and a slave, only one ACL link exists
  - Retransmission applied
  - A slave can return an ACL packet if and only if it has been addressed in the preceding master-to-slave slot
  - If the slave fails to decode the slave address, it is not allowed to transmit

*isochronous – information which must be transmitted within a fixed time, such as voice and streaming video.
Packets

- The bit ordering follows the Little Endian format: the LSB (least significant bit) sent first and shown on the left side.
- Each packet consists of 3 entities: access code, header, and payload.
  - Access code: 72 bits; Header: 54 bits; Payload: 0 – 2746 bits.
  - Packets may consist of (i) access code only, (ii) access code and header, or (iii) access code, header, and payload.

```
<table>
<thead>
<tr>
<th>LSB</th>
<th>72</th>
<th>54</th>
<th>0 - 2745</th>
<th>MSB</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACCESS CODE</td>
<td>HEADER</td>
<td>PAYLOAD</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
```

Access code

- Access code format:
  - The access code is used for synchronization, and identification.
    - All packets sent in the same piconet are preceded by the same channel access code.
    - A correlator at the receiver correlates against the access code and triggers when a threshold is exceeded.
  - Access code also used for paging and inquiring procedures: the access code is used as a signaling message, neither a header nor a payload needed.
Access code types

- Three types:
  - Channel access code (CAC)
  - Device access code (DAC)
  - Inquiry access code (IAC)

- CAC identifies a piconet: all packets exchanged on the piconet channel include this CAC

- DAC is used for special signaling procedures, e.g., paging and response to paging

- There are two variations for IAC: GIAC (general IAC) used for discovering which Bluetooth units are in range; DIAC (dedicated IAC) is common for a dedicated group of Bluetooth units

Access code (cont’d)

- Preamble: to facilitate DC compensation

<table>
<thead>
<tr>
<th>LSB</th>
<th>MSB</th>
<th>LSB</th>
<th>MSB</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 0 1 0</td>
<td>1---</td>
<td>0 1 0 1</td>
<td>0---</td>
</tr>
<tr>
<td>preamble</td>
<td>sync word</td>
<td>preamble</td>
<td>sync word</td>
</tr>
</tbody>
</table>

- Sync word: 64-bit code word derived from the 24-bit address (LAP: lower address parts)
  - CAC: master’s LAP used
  - IAC: dedicated LAP used
  - DAC: the slave unit LAP

- Trailer: appended when header follows the access code

<table>
<thead>
<tr>
<th>LSB</th>
<th>MSB</th>
<th>LSB</th>
<th>MSB</th>
</tr>
</thead>
<tbody>
<tr>
<td>...</td>
<td>0 1 0 1</td>
<td>...</td>
<td>1 0 1 0</td>
</tr>
<tr>
<td>sync word</td>
<td>trailer</td>
<td>sync word</td>
<td>trailer</td>
</tr>
</tbody>
</table>

a) b)
Packet header

- Header contains link control (LC) information:

<table>
<thead>
<tr>
<th>LSB 3</th>
<th>4</th>
<th>1</th>
<th>1</th>
<th>1</th>
<th>8</th>
<th>MSB</th>
</tr>
</thead>
<tbody>
<tr>
<td>AM_ADDR</td>
<td>TYPE</td>
<td>FLOW</td>
<td>ARQN</td>
<td>SEQN</td>
<td>HEC</td>
<td></td>
</tr>
</tbody>
</table>

- AM_ADDR: 3 bits, a member address, to distinguish between active members
  - All-0 address for broadcasting
- Type: 16 types of packets can be distinguished. The interpretation of the TYPE code depends on the physical link type
  - It also reveals how many slots the current packet occupies

Packet header (cont’d)

- Flow: for flow control of packets over the ACL link
  - When the buffer at the receiver side of the ACL link is full, a STOP indication (FLOW=0) is returned
  - When the buffer is empty, a GO indication (FLOW=1) returned
- ARQN: acknowledgement indication, to inform the source of a successful transfer
  - Successful: ARQN=1; otherwise: ARQN=0
- SEQN: sequential numbering to order the data packet stream
  - For each newly transmitted packet that contains data with CRC, the SEQN is inverted
- HEC: for header error check
Packet types

- **ID packet**: consists of the DAC or IAC. Used for paging, inquiring, and response routines.
- **NULL packet**: consists of AC and header only, 126 bits. Used to return link information, need not be acknowledged.
- **POLL packet**: no payload. Response needed.
- **FHS packet**: reveal the Bluetooth device address and the clock of the sender. Payload contains 144 information bits plus 16-bit CRC code. Used for frequency hop synchronization.
- **DM1 packet**: to support control messages in any link type.
SCO packets

- Used on the synchronous SCO link. No CRC included and no retransmission
- HV1 packet
  - 10 information bytes carried. 1/3 FEC coding
  - 1.25ms speech at 64kb/s rate. $T_{SCO}=2$
- HV2 packet
  - 20 information bytes, 2/3 FEC coding
  - 2.5ms speech at 64kb/s, $T_{SCO}=4$
- HV3 packet
  - 30 information bytes, no FEC coding
  - 3.75ms speech at 64kb/s, $T_{SCO}=6$

SCO packets (cont’d)

- DV packet
  - A combined data-voice packet, 80 bits voice and 150 bits data
  - Voice part not protected by FEC
  - Data part contains up to 10 information bytes (1 byte header) and 16-bit CRC. Encoded with rate 2/3 FEC
  - Data part can be retransmitted if necessary

<table>
<thead>
<tr>
<th>LSB</th>
<th>72</th>
<th>54</th>
<th>80</th>
<th>32 - 150</th>
<th>MSB</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACCESS CODE</td>
<td>HEADER</td>
<td>VOICE FIELD</td>
<td>DATA FIELD</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
ACL packets (1)

- DM1 packet: carries data information only.
  - Up to 18 information bytes (1 byte header) plus 16-bit CRC bit
  - 1 time slot
  - 2/3 FEC coding rate: 5 parity bits added to 10-bit segment
  - The length indicator specifies the number of user bytes (excluding payload header and CRC code)
- DH1 packet: payload is not FEC encoded
  - Up to 28 information bytes plus 16-bit CRC bit
  - A time slot

ACL packets (2)

- DM3 packet
  - Covers 3 time slots
  - Payload is up to 123 information bytes (2-bytes payload header included) plus 16-bit CRC code
  - The RF frequency remains the same when sending a DM3 packet
- DH3 packet
  - Payload is not FEC encoded
  - Carry up to 185 information bytes (including 2-byte payload header) plus 16-bit CRC code
  - 3 time slots covered
  - RF frequency not changed during transmission
ACL packets (3)

DM5 packet
- 5 time slots covered
- Payload contains up to 226 information bytes (including 2-byte payload header) plus 16-bit CRC code

DH5 packet
- No FEC encoding
- Payload is up to 341 information bytes (including 2-byte payload header) plus 16-bit CRC code

AUX1 packet
- Resembles a DH1 packet without CRC code
- Payload is up to 30 information bytes
- One time slot

Payload format

Payload header format:
- 1-byte header:
- 2-byte header:

**L_CH: logical channel**

<table>
<thead>
<tr>
<th>L_CH code (b1/b0)</th>
<th>Logical Channel</th>
<th>Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>NA</td>
<td>undefined</td>
</tr>
<tr>
<td>01</td>
<td>UA/UI</td>
<td>Continuation fragment of an L2CAP message</td>
</tr>
<tr>
<td>10</td>
<td>UA/UI</td>
<td>Start of an L2CAP message or no fragmentation</td>
</tr>
<tr>
<td>11</td>
<td>LM</td>
<td>LMP message</td>
</tr>
</tbody>
</table>
## Packet summary

<table>
<thead>
<tr>
<th>Type</th>
<th>User Payload (bytes)</th>
<th>FEC</th>
<th>CRC</th>
<th>Symmetric Max. Rate</th>
<th>Asymmetric Max. Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID</td>
<td>na</td>
<td>na</td>
<td>na</td>
<td>na</td>
<td>na</td>
</tr>
<tr>
<td>NULL</td>
<td>na</td>
<td>na</td>
<td>na</td>
<td>na</td>
<td>na</td>
</tr>
<tr>
<td>PULL</td>
<td>na</td>
<td>na</td>
<td>na</td>
<td>na</td>
<td>na</td>
</tr>
<tr>
<td>FHS</td>
<td>18</td>
<td>2/3</td>
<td>yes</td>
<td>na</td>
<td>na</td>
</tr>
</tbody>
</table>

## Packet summary (cont’d)

<table>
<thead>
<tr>
<th>Type</th>
<th>Payload Header (bytes)</th>
<th>User Payload (bytes)</th>
<th>FEC</th>
<th>CRC</th>
<th>Symmetric Max. Rate (kb/s)</th>
<th>Asymmetric Max. Rate (kb/s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>HV1</td>
<td>na</td>
<td>10</td>
<td>1/3</td>
<td>no</td>
<td>64.0</td>
<td></td>
</tr>
<tr>
<td>HV2</td>
<td>na</td>
<td>20</td>
<td>2/3</td>
<td>no</td>
<td>64.0</td>
<td></td>
</tr>
<tr>
<td>HV3</td>
<td>na</td>
<td>30</td>
<td>no</td>
<td>no</td>
<td>64.0</td>
<td></td>
</tr>
<tr>
<td>Dv*</td>
<td>1 D</td>
<td>10+(0-9)D</td>
<td>2/3</td>
<td>yes</td>
<td>64.0+57.6 D</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Type</th>
<th>Payload Header (bytes)</th>
<th>User Payload (bytes)</th>
<th>FEC</th>
<th>CRC</th>
<th>Symmetric Max. Rate (kb/s)</th>
<th>Asymmetric Max. Rate (kb/s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>DM1</td>
<td>1</td>
<td>0-17</td>
<td>2/3</td>
<td>yes</td>
<td>108.8</td>
<td>106.8</td>
</tr>
<tr>
<td>DH1</td>
<td>1</td>
<td>0-27</td>
<td>no</td>
<td>yes</td>
<td>172.8</td>
<td>172.8</td>
</tr>
<tr>
<td>DM3</td>
<td>2</td>
<td>0-121</td>
<td>2/3</td>
<td>yes</td>
<td>258.1</td>
<td>387.2</td>
</tr>
<tr>
<td>DH3</td>
<td>2</td>
<td>0-183</td>
<td>no</td>
<td>yes</td>
<td>390.4</td>
<td>585.6</td>
</tr>
<tr>
<td>DM5</td>
<td>2</td>
<td>0-224</td>
<td>2/3</td>
<td>yes</td>
<td>286.7</td>
<td>477.8</td>
</tr>
<tr>
<td>D5H</td>
<td>2</td>
<td>0-339</td>
<td>no</td>
<td>yes</td>
<td>433.0</td>
<td>723.2</td>
</tr>
<tr>
<td>AUX1</td>
<td>1</td>
<td>0-29</td>
<td>no</td>
<td>no</td>
<td>185.6</td>
<td>185.6</td>
</tr>
</tbody>
</table>
Logical channels

- LC channel (link control): mapped to the packet header. Carries low level functions including ARQ, flow control, payload characterization.
- LM channel (link manager): carries control information exchanged between the link managers of the master and slave(s). Indicated by the L_CH code 11.
- UA/UI channel (User Asynchronous/Isochronous data): carries L2CAP transparent asynchronous user data.
- US channel (User Synchronous data): carries transparent synchronous user data. Carried over SCO link.

Operational modes for Bluetooth (1)

- Four modes defined: active, sniff, hold, and park.
- In active mode, the Bluetooth unit actively participates on the channel:
  - The master schedules the transmission based on traffic demands.
  - Active slaves listen in the master-to-slave slots for packets.
  - If not addressed, it may sleep until the next new master transmission.
Operational modes for Bluetooth (2)

- **Sniff mode**: the duty cycle of the slave’s listen activity is reduced
  - In sniff mode, the master will only transmit packets in sniff slots
  - Sniff slots are spaced regularly with an interval of $T_{\text{sniff}}$
  - The slave will listen at slot $D_{\text{sniff}}$ (the timing of this first sniff slot is negotiated between master and slave) for $N_{\text{sniff attempt}}$ consecutive slots every sniff period
  - If a packet received in one of the $N_{\text{sniff attempt}}$ slots, the slave continues listening as long as it receives packets. Once it stops receiving, it should continue listening for $N_{\text{sniff timeout}}$ slots or for the rest of the $N_{\text{sniff attempt}}$ number of slots, whichever is greater
  - To enter sniff mode, the master should issue a sniff command containing $T_{\text{sniff}}$ and $D_{\text{sniff}}$

Operational modes for Bluetooth (3)

- **Hold mode**: a slave in hold mode does not support ACL packets any more
  - Prior to entering the hold mode, the master and slave agree on the time duration the slave remains in hold mode
  - A timer is set with the holdTO value. When timer expires, the slave wakes up
Operational modes for Bluetooth (4)

- Park mode: the slave gives up its active member address AM_ADDR
  - Two new addresses received:
    - PM_ADDR: 8-bit parked member address, used in master initiated unpark procedure
    - AR_ADDR: 8-bit access request address, used in slave initiated unpark procedure
  - The parked slave wakes up at regular intervals to listen to the channel to synchronize and check for broadcast messages
    - Beacon channel is used for synchronization and channel access for parked slaves
  - Park mode is also used for supporting more slaves in piconet

Beacon channel

- Master transmits to parked slaves using a periodic beacon, which begins at a beacon instant.
- Beacon channel consists of one beacon slot or a train of equidistant beacon slots transmitted periodically with a constant interval
- $T_B$ and $N_B$ are chosen such that there are sufficient beacon slots for a parked slave to synchronize to
- Beacon channels can be used to unpark one or more parked slaves
- Or for carrying general broadcast messages to parked slaves
Beacon access window (cont’d)

- Beacon access window is defined for the parked slaves to send unpark requests
  - Access window is repeated $M_{\text{access}}$ times for increased reliability
  - The width of the window is $T_{\text{access}}$
- Polling access is supported
  - TDD used
  - The slave-to-master slot is divided into two half slots, each with $312.5\,\mu s$
  - The parked slave is only allowed to send access request in the proper half slot
Unparking

- Master-activated unparking
  - The master sends a dedicated LMP unpark command including the parked slave’s PM_ADDR on beacon slots
  - The new AM_ADDR of the unparked slave should also be included
  - An unpark command can unpark several parked slaves

- Slave-activated unparking
  - A parked slave can send access request in the access window via the polling procedure
  - After access request is sent, the parked slave listens for an unpark message from the master
  - Before an unpark message is received, the parked slave can try several times

Scatternet

- Several piconets covering the same area may form a scatternet where two units belonging to different piconets can communicate
- A unit can be slaves in several piconets, but can only be the master in one piconet
- In case of ACL links only, a slave can participate in another piconet by entering hold or park mode. It can also join another piconet in sniff slots of the sniff mode
- A slave can join another piconet if there is only one SCO link using HV3 packets
Chapter 4 Bluetooth

Part 2  LMP and L2CAP

Link Manager Protocol (LMP)

- LMP is used for link setup, security and control
- Transferred in the payload and distinguished by a reserved value in the L_CH
- LM messages have higher priority than user data
- Do not need to explicitly acknowledge the messages in LMP
Format of LMP

- LM PDUs are always sent as single-slot packets: DV or DM1
- Payload header is 1 byte
- FLOW bit is always 1 and ignored on the receiving side
- Each PDU is assigned a 7-bit opcode to uniquely identify different types of PDUs
- The transaction ID is positioned in the LSB and is 0 if the PDU belongs to a transaction initiated by the master

```
+---+---+
| LSB | MSB |
+---+---+
| OpCode and transaction id | Content |
+---+---+
```

Hold mode

- The ACL link of a connection between two Bluetooth devices can be placed in hold mode for a specified hold time
  - No ACL packets will be transmitted
  - Transceiver can be turned off
- The LMP_hold (opcode: 20) and LMP_hold_req (opcode: 21) PDUs both contain a parameter, hold instant, which specifies the instant at which the hold will become effective
  - At least $6 \times T_{poll}$ in the future

```
<table>
<thead>
<tr>
<th>M/O</th>
<th>PDU</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>LMP_hold</td>
<td>hold time, hold instant</td>
</tr>
<tr>
<td>0</td>
<td>LMP_hold_req</td>
<td>hold time, hold instant</td>
</tr>
</tbody>
</table>
```
Master forces hold mode

- If there has previously been a request for hold mode that has been accepted by the slave
  - hold time cannot be longer than any hold time that slave has previously accepted
- The master first finalizes the transmission of the current ACL packet with L2CAP information and stops L2CAP transmission
- Selects the hold instant and queues the LMP_hold for transmission
- A timer starts until the hold instant occurs
- When the slave LM receives the LMP_hold it checks the hold instant
  - If in the future, starts a timer until the instant
- When the master LM exits from hold mode it re-enables L2CAP transmission

Slave forces hold mode

- If there has previously been a request for hold mode that has been accepted by master
  - hold time cannot be longer than any hold time that master has previously accepted
- The slave first finalizes the transmission of the current ACL packet with L2CAP information and stops L2CAP transmission
- Selects the hold instant and queues the LMP_hold for transmission. Then waits for the LMP_hold from master
- When the master LM receives the LMP_hold, it finalizes the transmission of the current ACL packet with L2CAP information and stops L2CAP transmission. Then it sends the LMP_hold to the slave
- When the master LM and slave LM exit from hold mode they re-enable L2CAP transmission
Master or slave requests hold mode

- The initiating LM finalizes L2CAP transmission first. Then send LMP_hold_req with hold instant at least 9*T_poll
- If no LMP_not_accepted or LMP_hold_req received before this hold instant, it enters hold mode
- The receiving LM can modify the LMP_hold_req and sends it to the other end
- If an agreement is seen, LMP_accepted is sent; otherwise, LMP_not_accepted sent

Connection establishment

- When the paging device wishes to create a connection involving layers above LM, it sends LM_connection_req
- When the other side receives this message, the host is informed
- The remote device can decide to accept or reject this request
- If this request is accepted, LMP security procedures can be invoked
- Finally, LMP_setup_complete is exchanged
Logical link control and adaptation protocol (L2CAP)

- L2CAP provides connection-oriented and connectionless data services to upper layer protocols
  - Multiplexing, segmentation and reassembly, group abstractions
- L2CAP is only over ACL links
  - AUX1 not permitted: no CRC
- Two payload headers:

L2CAP functional requirements

- Protocol multiplexing
  - L2CAP must be able to distinguish between upper layer protocols
- SAR
  - Upper layer protocols are allowed to use larger packets
- Quality of service
  - The L2CAP connection establishment process allows the exchange of information regarding the quality of service. Each L2CAP implementation must monitor the resources used
- Groups
  - Efficiently map protocol groups on to piconets
Channel identifiers

- Channel identifiers are local names representing a logical channel endpoint on the device

<table>
<thead>
<tr>
<th>CID</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x0000</td>
<td>Null identifier</td>
</tr>
<tr>
<td>0x0001</td>
<td>Signalling channel</td>
</tr>
<tr>
<td>0x0002</td>
<td>Connectionless reception channel</td>
</tr>
<tr>
<td>0x0003-0x003F</td>
<td>Reserved</td>
</tr>
<tr>
<td>0x0040-0xFFFF</td>
<td>Dynamically allocated</td>
</tr>
</tbody>
</table>

Segmentation and reassembly

- SAR is used for improving efficiency by supporting a maximum transmission unit size larger than the largest baseband packet
- All L2CAP packets may be segmented for transfer over baseband packets and use L_CH to indicate the start of a new L2CAP packet
- The baseband protocol delivers ACL packets in sequence and protects the integrity of data using a 16-bit CRC
- The length field in the header of L2CAP packets is used as a consistency check
Packet format for connection-oriented channel

- Length: 2 octets (16 bits): to indicate the size of information payload in bytes, excluding the length of the L2CAP header. Can be up to 65535 bytes. Serves as a simple integrity check of the reassembled L2CAP packet at the receiving side.
- Channel ID: 2 octets. Identifies the destination channel end points of the packet.
- Information: 0 to 65535 octets
Packet format for connectionless channel

- Data sent to the group channel is sent in best-effort manner
- Length: 2 octets. Indicates the size of the information payload plus the PSM field in bytes
- Channel ID: 2 octets
- Protocol/service multiplexer (PSM): 2 octets (minimum)
- Information: 0 to 65533 octets

Packet format for connectionless channel (cont’d)

<table>
<thead>
<tr>
<th>LSB</th>
<th>byte0</th>
<th>byte1</th>
<th>byte2</th>
<th>byte3</th>
<th>MSB</th>
</tr>
</thead>
<tbody>
<tr>
<td>Length</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Channel ID (0x0002)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PSM</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Information (payload)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Information (cont.)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>PSM value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x0001</td>
<td>Service Discovery Protocol</td>
</tr>
<tr>
<td>0x0003</td>
<td>RFCOMM</td>
</tr>
<tr>
<td>0x0005</td>
<td>Telephony Control Protocol</td>
</tr>
<tr>
<td>&lt;0x1000</td>
<td>RESERVED</td>
</tr>
</tbody>
</table>
Collision probability between two piconets

- Consider two piconets in the same area
- For simplicity, Bluetooth transmits in whole time slot in single timeslot format
- Consider a particular packet sent in piconet 1. The probability that this packet will not be corrupted by the transmission in piconet 2 is $P_{\text{non-corrupt}} = \left(\frac{78}{79}\right)^2 = 0.9748$
- The probability of corruption is $P_{\text{corrupt}} = 1 - P_{\text{non-corrupt}} = 0.0252$