Bluetooth Technology

Syed Masud Mahmud, Ph.D.
Electrical and Computer Engg. Dept.
Wayne State University
Detroit MI 48202
(313) 577-3855,
smahmud@eng.wayne.edu
http://ece.eng.wayne.edu/~smahmud

July 12, 2005
Outline of the Talk

- Bluetooth Basics
- Bluetooth Network
- Inquiry Procedure
- Paging Procedure
- Bluetooth Packets
- Bluetooth States
- Bluetooth Security
- Bluetooth Network for Vehicular Applications
Bluetooth Basics
A Short-Range Wireless Technology

• Bluetooth is a short-range wireless technology intended to **replace the cables** connecting various electronic devices.
• Most Bluetooth devices are designed for a range of 10 meters.
Bluetooth Applications

- Bluetooth is ideal for
  - wireless headsets,
  - wireless synchronization of PDAs with PCs,
  - wireless PC peripherals such as printers, keyboards, or mice,
  - wireless access to LANs
  - wireless access to internet from a laptop via cellular networks
Bluetooth Applications

In-Vehicle Wireless Network.

- Advantages of such a network are:
  - Hands free control of Lights, Radio, AC, Wiper, Cruise, and other features using a BT enabled headset.
  - Remote Engine Start using a BT enabled PDA
  - Remote monitoring of various parameters such as tire pressure, interior temperature, battery voltage, etc. using a BT enabled PDA.
  - Wireless connection to a dealer’s computer at a service station
Bluetooth Applications

Automatic Payment:
- Automatic payment at a BT enabled parking meter using a BT enabled cell phone.
- Automatic payment at a BT enabled vending machine using a BT enabled cell phone.
## Power Classes of Bluetooth Devices

<table>
<thead>
<tr>
<th>Class</th>
<th>Power</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Class 1</td>
<td>100 mW</td>
<td>100m (≈ 325ft)</td>
</tr>
<tr>
<td></td>
<td>(20 dBm)</td>
<td></td>
</tr>
<tr>
<td>Class 2</td>
<td>2.5 mW</td>
<td>10m (≈ 32ft)</td>
</tr>
<tr>
<td></td>
<td>(4 dBm)</td>
<td></td>
</tr>
<tr>
<td>Class 3</td>
<td>1mW</td>
<td>1m (≈ 3ft)</td>
</tr>
<tr>
<td></td>
<td>(0 dBm)</td>
<td></td>
</tr>
</tbody>
</table>
Outdoor Range Tests

- Test results show that in open space the range of Bluetooth Devices far exceeded the specification.

<table>
<thead>
<tr>
<th>Class</th>
<th>Maximum Range</th>
<th>Spec. Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Class 1 (20 dBm)</td>
<td>250 m (≈ 820ft)</td>
<td>100 m</td>
</tr>
<tr>
<td>Class 2 (4 dBm)</td>
<td>122 m (≈ 400ft)</td>
<td>10 m</td>
</tr>
</tbody>
</table>
Radio Frequency Bands

- The Bluetooth system operates in the **2.4 GHz** *ISM* band (unlicensed band).
- In USA, Bluetooth devices operate in **79 RF** channels.
- Channel spacing is **1MHz**.

* **ISM**: Industrial, Scientific and Medical.
## Operating and Guard Bands

### Operating frequency bands

<table>
<thead>
<tr>
<th>Regulatory Range</th>
<th>RF Channels</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.400-2.4835 GHz</td>
<td>f=2402+k MHz, k=0,…,78</td>
</tr>
</tbody>
</table>

### Guard Bands

<table>
<thead>
<tr>
<th>Lower Guard Band</th>
<th>Upper Guard Band</th>
</tr>
</thead>
<tbody>
<tr>
<td>2 MHz</td>
<td>3.5 MHz</td>
</tr>
</tbody>
</table>
Frequency Hopping

- Bluetooth devices continuously change channels to avoid interference from other devices such as **cordless phones** and **microwave oven** which also operate at the 2.4 GHz ISM band.
The Inventor of Frequency Hopping Technique

Hedy Lamarr
(1914 - 2000)

• Was a Movie Star and an Inventor
• Received a patent in 1942 on Frequency Hopping (FH)
• Did not make a dime from her patent because her patent expired in 1959 and nobody used FH until 1962.
The Inventor of Frequency Hopping Technique

Hedy Lamarr
(1914 - 2000)

SHE HOPPED HUSBANDS SIX TIMES
Hedy Lamarr was a remarkable combination of movie star and inventor. She was born in Vienna in 1914 as Hedwig Eva Maria Kiesler. She achieved fame in 1933 as the star of the scandalous Czech film *Extase*, which featured the first nude scene in cinematic history. The same year Lamarr married Fritz Mandl, one of the five leading European armament manufacturers. Mandl specialized in shells and grenades, but from the mid-thirties on he also manufactured military aircraft. He was interested in control systems and conducted research in the field. Mandl kept his young wife by his side as he attended hundreds of dinners and meetings with arms developers, builders, and buyers, where Lamarr clearly learned some things. The marriage broke up in 1937, when Lamarr escaped to London. She soon was signed by MGM
Frequency Hopping Rate for Bluetooth Devices

- During normal operations, the channels are changed **1600** times/second.
- During special procedures such as **inquiry** and **paging**, the channels are changed **3200** times/second.
GFSK Coding for 1Mbs Rate (Gaussian Frequency Shift Keying)

Ideal Zero Crossing

Zero Crossing Error

Bluetooth Technology

Syed Masud Mahmud, Ph.D.
**π/4-DQPSK Coding for 2Mbs Rate**

**π/4- DQPSK:**
Differential Quarternary Phase Shift Keying

<table>
<thead>
<tr>
<th>$b_{2k-1}$</th>
<th>$b_{2k}$</th>
<th>$\phi_k$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>$\pi/4$</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>$3\pi/4$</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>$-3\pi/4$</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>$-\pi/4$</td>
</tr>
</tbody>
</table>
8DPSK Coding for 3Mbs Rate

<table>
<thead>
<tr>
<th>$b_{3k-2}$</th>
<th>$b_{3k-1}$</th>
<th>$b_{3k}$</th>
<th>$\varphi_k$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>$\pi/4$</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>$\pi/2$</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>$3\pi/4$</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>$\pi$</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>$-3\pi/4$</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>$-\pi/2$</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>$-\pi/4$</td>
</tr>
</tbody>
</table>

8DPSK: 8-ary Differential Phase Shift Keying
Bluetooth Network
Master-Slave Protocol

- Bluetooth devices communicate among themselves using a Master-Slave Protocol.
- A Bluetooth network is called a Picoent.
- Each Piconet has one Master and up to seven active Slaves (earlier specifications).
- Latest specifications allow up to fourteen active Slaves on two different logical links together.
A Piconet With Two Devices

Master  Slave
A Piconet With Eight Devices

Master

7 Slaves
Scatternet: Multiple Interconnected Piconets

Master
Slave

PICONESET 1
PICONESET 2
PICONESET 3
Bluetooth Device Address

- Each Bluetooth unit has a unique **48-bit address** (BD_ADDR).

Format of BD_ADDR (an example)

<table>
<thead>
<tr>
<th>LSB</th>
<th>MSB</th>
</tr>
</thead>
<tbody>
<tr>
<td>company_assigned</td>
<td>company_id</td>
</tr>
<tr>
<td>LAP</td>
<td>UAP</td>
</tr>
<tr>
<td>0000 0001 0000 0000 0000</td>
<td>0001 0010 0111 1011 0011 0101</td>
</tr>
</tbody>
</table>

**LAP**: Lower Address Part  
**UAP**: Upper Address Part  
**NAP**: Non-Significant Address Part
64 Reserved LAP Values

- 64 LAP values are reserved for Inquiry Process and future use. These LAP values must not be used for BD-ADDR of any devices regardless of the values of UAP and NAP.

<table>
<thead>
<tr>
<th>LAP (24-bit Hex Num)</th>
<th>Reserved For</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x9E8B00</td>
<td>Dedicated Inquiry</td>
</tr>
<tr>
<td>0x9E8B01-0x9E8B32</td>
<td>Future use</td>
</tr>
<tr>
<td>0x9E8B33</td>
<td>General Inquiry</td>
</tr>
<tr>
<td>0x9E8B34-0x9E8B3F</td>
<td>Future use</td>
</tr>
</tbody>
</table>
64 Reserved LAP Values

<table>
<thead>
<tr>
<th>LSB</th>
<th>LAP</th>
<th>UAP</th>
<th>NAP</th>
<th>MSB</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0000</td>
<td>0000</td>
<td>1101</td>
<td>0001</td>
</tr>
<tr>
<td></td>
<td>1111</td>
<td>1100</td>
<td>1101</td>
<td>0001</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>XXXX</td>
<td>XXXX</td>
</tr>
</tbody>
</table>
Channel Hopping Sequence

- The Channel Hopping Sequence of a piconet depends on the least significant 28 bits of the BD_ADDR of the Master device.

An Example of Channel Hopping Sequence

0 → 77 → 45 → 31 → 1 → 59 → 21 → ...

Hop Sequence = Function(Entire LAP and 4 LSBs of UAP of the Master BD_ADDR)
Bluetooth Clock

- Each Bluetooth unit has a 28-bit Timer (Clock) which counts at the rate of 3.2KHz (3200 counts/sec)
- The clock wraps around in 23hr 18min.
Piconet Clock

- The master clock of a piconet is known as the piconet clock.
- All timing signals of a piconet are derived from the master clock.

**CLKN**: Native Clock of a Device

**CLK**: Piconet Clock
Phase of a Hop Sequence

- The **phase** (channel number) of a hop sequence at a particular time depends on the most significant 27 bits of the *Master Clock*.

\[
\text{CURRENT CHANNEL} = \text{Function(Master Clock)}
\]

0 → 77 → 45 → 31 → 1 → 59 → 21 → ...

**Hop Sequence**
Channel Selection

- UAP/LAP (bits: 0 - 27 of BD_ADDR)
- CLOCK (bits: 1 - 27)
- Other Parameters
- RF channel number (A pseudo-random sequence)
Time Slots

- The basic piconet physical channel is divided into time slots, each 625 µs in length.
- The time slots are numbered according to the most significant 27 bits of the Bluetooth clock of the piconet master.
Master uses Even-Numbered and Slaves use Odd-Numbered Time Slots

\[ k = \text{An Even Number (Time Slot number)} \]
\[ f(k) = \text{Frequency of Time Slot } k \]
Multi-Slot Packets

- Note that for the 3-slot packet the frequency remains the same for all three slots.
- The frequency changes only at the start of a packet.
• Frequency remains the same for all five slots.
Multi-Slave Operation

master

slave 1

slave 2

Bluetooth Technology
Slave-to-Slave Communications are not allowed

- In a Bluetooth system, slaves can communicate only with the Master of the piconet.
Adapted Frequency Hopping (AFH)

- An adapted frequency hopping sequence may use fewer (at least 20) than 79 channel frequencies.
- In this scheme the slave frequency is the same as the preceding master transmission frequency.
Same Channel Mechanism

![Diagram showing the same channel mechanism in Bluetooth Technology]

Syed Masud Mahmud, Ph.D.
Hop Sequence Switching

Hop sequence is changed when the master sends an AFH command and receives an acknowledge.
Logical Transports (Links)

- Synchronous Connection-Oriented (**SCO**) logical transport.
- Extended Synchronous Connection-Oriented (**eSCO**) logical transport.
- Asynchronous Connection-Oriented (**ACL**) logical transport.
- Active Slave Broadcast (**ASB**) logical transport.
- Parked Slave Broadcast (**PSB**) logical transport.
**SCO Logical Transport**

- **SCO** link is a point-to-point link between the master and a specific slave.
- For SCO communication, some time slots are **reserved** for both the master and the slave. Thus it can be considered as a circuit-switched communication.
- SCO packets are **never retransmitted**.

![SCO Link Diagram](image)
Maximum Three SCO Links

• The master can support up to three SCO links to the same slave or to different slaves.
**ACL Logical Transport**

- The **ACL** link is used for point-to-point asynchronous communication.
- Using the ACL link, the master can communicate with each active slave on a one-to-one basis.
Logical Transport Address (LT_ADDR)

- Each slave on the ACL link is given a 3-bit address (LT_ADDR) in the range 1 – 7.
- Thus, there could be maximum 7 slaves on the ACL link.
- LT_ADDR=0 is used for broadcast messages.
ACL Links with Basic and Enhanced Data Rate

- The master sends a *Packet_Type_Table (ptt)* parameter to a slave when an ACL link is setup for the slave.
- The parameter is sent via an LMP (Link Manager Protocol) message.
- \( ptt = 0 \)  \( \rightarrow \) Basic Data Rate (1 Mbps)
- \( ptt = 1 \)  \( \rightarrow \) Enhanced Data Rate (2 or 3 Mbps)
eSCO Logical Transport

• The eSCO link was not available in earlier Bluetooth Specifications.

• It can be used for voice and other synchronous data (**time bounded data**) such as streaming video.

• Unlike the SCO link, **retransmissions are allowed** on the eSCO link.

• The eSCO transport supports both the Basic and Enhanced Data Rate.
LT_ADDR for eSCO Link

- A second set of 3-bit LT_ADDR is used for eSCO slaves.
- Thus, in addition to the 7 slaves on the ACL link there could be another 7 slaves on the eSCO link.
- The LT_ADDR for eSCO link is used with retransmitted messages so that the slaves can identify their retransmitted messages.
**ASB Logical Transport**

- The master of a piconet uses the ASB transport to broadcast packets to all **active slaves** connected to the piconet.

- An acknowledgement is **not necessary** for the ASB packets.
PSB Logical Transport

- A piconet can have up to 255 parked slaves.
- The master uses the PSB transport to broadcast packets to all parked slaves.
- The PSB link is the only logical transport that exists between the master and the parked slaves.
## Bluetooth Layers

<table>
<thead>
<tr>
<th>Application Layer</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>L2CAP Layer</strong></td>
</tr>
<tr>
<td>(Logical Link Control and Adaptation Protocol Layer)</td>
</tr>
<tr>
<td>- Allows multiple applications to use a link between two devices.</td>
</tr>
<tr>
<td>- Accepts large packets and converts to smaller packets for Baseband Layer.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>LMP Layer</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Link Manager Protocol Layer)</td>
</tr>
<tr>
<td>- Provides functionality to attach/detach slaves and switch roles between the master and a slave.</td>
</tr>
<tr>
<td>- Handles low power modes of slaves, provides quality of service, authenticates devices, etc.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Baseband Layer</th>
</tr>
</thead>
<tbody>
<tr>
<td>- Controls the radio layer and provides frequency hop sequence.</td>
</tr>
<tr>
<td>- Synchronizes devices’ clocks and establishes connections.</td>
</tr>
<tr>
<td>- Handles packets over the radio layer and provides low-level encryption.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Radio Layer</th>
</tr>
</thead>
</table>

---

Wayne State University

Bluetooth Technology

Syed Masud Mahmud, Ph.D.

53
Bluetooth Technology

Syed Masud Mahmud, Ph.D.
Inquiry Procedure
Inquiry Procedure

• Bluetooth devices use the inquiry procedure to discover nearby devices.

• During the inquiry procedure, the **master** (inquiring device) transmits inquiry messages for the **general** or **dedicated** inquiry.
General and Dedicated Inquiry

- **General** inquiry message is used to discover any nearby BT devices.
- **Dedicated** inquiry message is used to discover a particular group of BT devices.
Discoverable Devices

- A BT device must be in discoverable mode for it to be detected by an inquiring device.
Access Code

- In BT systems all packets start with an **Access Code**. The Access Code indicates the message type.

**Basic Rate Packet Format**

<table>
<thead>
<tr>
<th>LSB 68/72</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>
</tr>
</tbody>
</table>

**Enhanced Rate Packet Format**

<table>
<thead>
<tr>
<th>LSB</th>
<th>ACCESS CODE</th>
<th>HEADER</th>
<th>GUARD</th>
<th>SYNC</th>
<th>ENHANCED DATA RATE PAYLOAD</th>
<th>TRAILER</th>
<th>MSB</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>GFSK</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>DPSK</td>
</tr>
</tbody>
</table>
Inquiry Message

- The inquiry message is very short, and it contains only an inquiry access code.

68 bits

<table>
<thead>
<tr>
<th>INQUIRY ACCESS CODE</th>
</tr>
</thead>
</table>

Inquiry Message
General Inquiry Access Code

- **GIAC**: General Inquiry Access Code (used for general inquiry) is generated using the reserved LAP (0x9E8B33) for General Inquiry.

![Diagram showing the process of generating GIAC]
Dedicated Inquiry Access Code

- **DIAC**: Dedicated Inquiry Access Code (used for dedicated inquiry) is generated using the reserved LAP (**0x9E8B00**) for Dedicated Inquiry.

![Diagram showing the process of generating DIAC using the reserved LAP](image_url)
Master TX/RX Cycle in Inquiry Mode

- The master sends a train of inquiry messages over 32 hop frequencies for a certain period of time.

Hop Rate = 3200 times/sec
Inquiry Response Hop Frequency

- For every $f(k)$ there is a corresponding response frequency $f'(k)$. 

Hop Rate = 3200 times/sec
Hop Sequence for Inquiry Message

- The reserved LAP (0x9E8B33) for general inquiry is used to generate the hop sequence for both the General and Dedicated Inquiry.
- The least significant 4 bits of UAP is used as 0000.
Inquiry and Inquiry Scan States

A BT device enters this state to discover other BT devices.

Other BT devices periodically enter this state to be discovered by an inquiring device.
Timing of Inquiry Response Packet on Successful Inquiry in First Half Slot

Master

Slave

hop f(k)

hop f(k+1)

hop f’(k)

68 μs

625 μs
Timing of Inquiry Response Packet on Successful Inquiry in Second Half Slot

Master-to-slave slot: hop $f(k)$, hop $f(k+1)$
Slave-to-master slot: hop $f'(k)$, hop $f'(k+1)$

68μs

625μs
## GAP Timers

<table>
<thead>
<tr>
<th>Timer name</th>
<th>Recommended value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$T_{GAP}(100)$</td>
<td>10.24 s</td>
<td>Normal time span that a Bluetooth device performs inquiry.</td>
</tr>
<tr>
<td>$T_{GAP}(101)$</td>
<td>10.625 ms</td>
<td>Minimum time in INQUIRY_SCAN.</td>
</tr>
<tr>
<td>$T_{GAP}(102)$</td>
<td>2.56 s</td>
<td>Maximum time between repeated INQUIRY_SCAN enterings.</td>
</tr>
</tbody>
</table>
Paging Procedure
Paging Procedure

• A BT device initiates a paging procedure to connect another BT device to it.

Master: Paging Device
Slave: Paged Device
Master TX/RX Cycle in Page Mode

- Similar to the inquiry mode, the master sends a train of short messages over 32 hop frequencies for a certain period of time.

Hop Rate = 3200 times/sec
Page Hopping Sequence

- The hop sequence depends on the 28 LSBs of **BD_ADDR** of the **Paged Device**.
Page Response Hop Frequency

- For every $f(k)$ there is a corresponding response frequency $f'(k)$.
Connection Steps when Slave Responds to the First Page Message

<table>
<thead>
<tr>
<th>Step</th>
<th>Master</th>
<th>Slave</th>
</tr>
</thead>
<tbody>
<tr>
<td>Page</td>
<td>$f(k)$</td>
<td>$f'(k)$</td>
</tr>
<tr>
<td>FHS</td>
<td>$f(k+1)$</td>
<td>$f'(k+1)$</td>
</tr>
<tr>
<td>1st Traffic</td>
<td>$g(m)$</td>
<td>$g(m+1)$</td>
</tr>
</tbody>
</table>

Bluetooth Technology

Syed Masud Mahmud, Ph.D.
Connection Steps when Slave Responds to the Second Page Message

**Master**
- Step 1: Page (f(k), f(k+1))
- Step 2: FHS (f(k+2))
- Step 3: 1st Traffic (g(m))

**Slave**
- Step 1: Response (f'(k+1))
- Step 2: Response (f'(k+2))
- Step 3: 1st Traffic (g(m+1))

Page hopping sequence: f(k), f(k+1), f(k+2), g(m), f'(k+1), f'(k+2), g(m+1)

Basic channel hopping sequence: f(k), f(k+1), f(k+2), g(m), f'(k+1), f'(k+2), g(m+1)
Page and Page Scan States

A BT device enters this state to connect another BT device to it.

Other BT devices periodically enter this state to check for paging messages.
# GAP Timers

<table>
<thead>
<tr>
<th>Timer name</th>
<th>Recommended value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$T_{\text{GAP}(101)}$</td>
<td>10.625 ms</td>
<td>Minimum time in PAGE_SCAN</td>
</tr>
<tr>
<td>$T_{\text{GAP}(102)}$</td>
<td>2.56 s</td>
<td>Maximum time between repeated PAGE_SCAN enterings.</td>
</tr>
</tbody>
</table>
State Transition During Paging and Inquiry Procedures
Bluetooth Packets
Basic Rate (1Mps) Packet Format

A packet may consist of:

- The shortened access code only.
- The access code and the packet header.
- The access code, the packet header and the payload.
Enhanced Rate Packet Format

<table>
<thead>
<tr>
<th>LSB</th>
<th>MSB</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACCESS CODE</td>
<td>HEADER</td>
</tr>
<tr>
<td>GUARD</td>
<td>SYNC</td>
</tr>
<tr>
<td>ENHANCED DATA RATE PAYLOAD</td>
<td>TRAILER</td>
</tr>
</tbody>
</table>

- **GFSK** - Gaussian Frequency-Shift Keying
- **DPSK** - Differential Phase-Shift Keying
- **GUARD** - The guard time is used for modulation change on the physical channel.
Bit Ordering

- The bit ordering when defining packets and messages in the Baseband Specification, follows the *Little Endian* format. The following rules apply:
  - The least significant bit (LSB) corresponds to $b_0$;
  - The LSB is the first bit sent over the air;
  - For instance, a 3-bit parameter $X=3$ is sent as: $b_0b_1b_2 = 110$
## Access Code

<table>
<thead>
<tr>
<th>LSB</th>
<th>4</th>
<th>64</th>
<th>MSB</th>
</tr>
</thead>
<tbody>
<tr>
<td>PREAMBLE</td>
<td></td>
<td>SYNC WORD</td>
<td>TRAILER</td>
</tr>
</tbody>
</table>

- If a packet header follows, the access code is 72 bits long, otherwise it is 68 bits long.
- It is used for synchronization, DC offset compensation and identification.
- All packets sent in the same piconet are preceded by the same access code.
## Access Code Types

<table>
<thead>
<tr>
<th>Code type</th>
<th>LAP</th>
<th>Code length</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>CAC</td>
<td>Master</td>
<td>72</td>
<td>Channel Access Code (Identifies a Piconet)</td>
</tr>
<tr>
<td>DAC</td>
<td>Paged device</td>
<td>68/72</td>
<td>Device Access Code</td>
</tr>
<tr>
<td>GIAC</td>
<td>Reserved</td>
<td>68/72*</td>
<td>General Inquiry Access Code</td>
</tr>
<tr>
<td>DIAC</td>
<td>Dedicated</td>
<td>68/72*</td>
<td>Dedicated Inquiry Access Code</td>
</tr>
</tbody>
</table>

*72 bits are used in the response message containing FHS packet.*
Preamble of the Access Code

- The preamble is either 1010 or 0101, depending on whether the LSB of the sync word is 1 or 0, respectively.

```
LSB | MSB | LSB | LSB | LSB | MSB | LSB
1   | 0   | 1   | --  | 0   | 1   | --
preamble | sync word
0101 | 0   | 1   | --  | 0   | 1   | --
preamble | sync word
```

Access Code

```
LSB | 4 | 64 |
--- |---|----|
PREAMBLE | | |
SYNC WORD |
TRAILER  |---|---|
```

Bluetooth Technology

Syed Masud Mahmud, Ph.D.
Sync Word of the Access Code

- The sync word is a 64-bit code word derived from a 24 bit address (LAP).
- The construction guarantees large Hamming distance ($d_{\text{min}}=14$) between sync words based on different LAPs.
# Trailer of the Access Code

- The trailer is used if a packet header follows.
- It is either 1010 or 0101 depending on whether the MSB of the sync word is 0 or 1, respectively.

<table>
<thead>
<tr>
<th>MSB</th>
<th>LSB</th>
<th>MSB</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0101</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>MSB</th>
<th>LSB</th>
<th>MSB</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1010</td>
<td></td>
</tr>
</tbody>
</table>
Packet Header

- The header consists of 18 bits, and is encoded with a rate 1/3 FEC resulting in a 54-bit header.
- In the rate 1/3 FEC coding each bit is copied 3 times.

Basic Rate Packet Format

<table>
<thead>
<tr>
<th>LSB 68/72</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>
</tr>
</tbody>
</table>
LT_ADDR (Logical Transport Address)

- The 3-bit LT_ADDR field indicates the destination slave for a packet in a master-to-slave transmission slot and indicates the source slave for a slave-to-master transmission slot.
TYPE

• The 4-bit TYPE code specifies which packet type (out of 16 different types) is used.
FLOW=0 $\rightarrow$ STOP

- It is used with ACL packets only. When the RX buffer in the recipient is full, a STOP indication (FLOW=0) shall be returned to stop the other device from transmitting data temporarily.

- Packets containing only link control information (ID, POLL, and NULL packets), SCO packets or eSCO packets can still be received.
FLOW=1 → GO

- When the RX buffer can accept data, a GO indication (FLOW=1) shall be returned.
ARQN

• The ARQN bit is used to inform the source of a successful transfer of payload data with CRC.

• $\text{ARQN} = 1 \rightarrow$ Positive Acknowledge.

• $\text{ARQN} = 0 \rightarrow$ Negative Acknowledge.

Packet Header

<table>
<thead>
<tr>
<th>LSB</th>
<th>MSB</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>8</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>LT_ADDR</th>
<th>TYPE</th>
<th>FLOW</th>
<th>ARQN</th>
<th>SEQN</th>
<th>HEC</th>
</tr>
</thead>
</table>
SEQN

- The SEQN bit provides a sequential numbering scheme to order the data packet stream.

- For each **new transmitted** packet that contains data with CRC, the **SEQN bit is inverted**.

---

Packet Header

<table>
<thead>
<tr>
<th></th>
<th>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>LT_ADDR</td>
<td>TYPE</td>
<td>FLOW</td>
<td>ARQN</td>
<td>SEQN</td>
<td>HEC</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

---

Bluetooth Technology

Syed Masud Mahmud, Ph.D.
HEC

• Each header has a header-error-check (HEC) to check the header integrity
PACKET TYPES

- The packet types have been divided into four segments.
- **Segment-1** is reserved for the four control packets common to all link types (SCO eSCO and ACL)
- **Segment-2** is reserved for packets occupying a single time slot.
- **Segment-3** is reserved for packets occupying three time slots.
- **Segment-4** is reserved for packets occupying five time slots.
## Segment-1 Packets

<table>
<thead>
<tr>
<th>Segment</th>
<th>TYPE code $b_3b_2b_1b_0$</th>
<th>Slot occupancy</th>
<th>SCO logical transport (1 Mbps)</th>
<th>eSCO logical transport (1 Mbps)</th>
<th>eSCO logical transport (2-3 Mbps)</th>
<th>ACL logical transport (1 Mbps) ptt=0</th>
<th>ACL logical transport (2-3 Mbps) ptt=1</th>
</tr>
</thead>
<tbody>
<tr>
<td>0000</td>
<td>0000</td>
<td>1</td>
<td>NULL</td>
<td>NULL</td>
<td>NULL</td>
<td>NULL</td>
<td>NULL</td>
</tr>
<tr>
<td>0001</td>
<td>0001</td>
<td>1</td>
<td>POLL</td>
<td>POLL</td>
<td>POLL</td>
<td>POLL</td>
<td>POLL</td>
</tr>
<tr>
<td>0010</td>
<td>0010</td>
<td>1</td>
<td>FHS</td>
<td>reserved</td>
<td>reserved</td>
<td>FHS</td>
<td>FHS</td>
</tr>
<tr>
<td>0011</td>
<td>0011</td>
<td>1</td>
<td>DM1</td>
<td>reserved</td>
<td>reserved</td>
<td>DM1</td>
<td>DM1</td>
</tr>
</tbody>
</table>

**ACL Enhanced Rate** is selected via LMP using the packet_type_table (ptt) parameter.

**eSCO Enhanced Rate** is selected when the eSCO link is established.
## Segment-2 Packets

<table>
<thead>
<tr>
<th>Segment</th>
<th>TYPE code $b_3b_2b_1b_0$</th>
<th>Slot occupancy</th>
<th>SCO logical transport (1 Mbps)</th>
<th>eSCO logical transport (1 Mbps)</th>
<th>eSCO logical transport (2-3 Mbps)</th>
<th>ACL logical transport (1 Mbps) $\text{ptt}=0$</th>
<th>ACL logical transport (2-3 Mbps) $\text{ptt}=1$</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>0100</td>
<td>1</td>
<td>undefined</td>
<td>undefined</td>
<td>undefined</td>
<td>DH1</td>
<td>2-DH1</td>
</tr>
<tr>
<td></td>
<td>0101</td>
<td>1</td>
<td>HV1</td>
<td>undefined</td>
<td>undefined</td>
<td>undefined</td>
<td>undefined</td>
</tr>
<tr>
<td></td>
<td>0110</td>
<td>1</td>
<td>HV2</td>
<td>undefined</td>
<td>2-EV3</td>
<td>undefined</td>
<td>undefined</td>
</tr>
<tr>
<td></td>
<td>0111</td>
<td>1</td>
<td>HV3</td>
<td>EV3</td>
<td>3-EV3</td>
<td>undefined</td>
<td>undefined</td>
</tr>
<tr>
<td></td>
<td>1000</td>
<td>1</td>
<td>DV</td>
<td>undefined</td>
<td>undefined</td>
<td>undefined</td>
<td>3-DH1</td>
</tr>
<tr>
<td></td>
<td>1001</td>
<td>1</td>
<td>undefined</td>
<td>undefined</td>
<td>undefined</td>
<td>undefined</td>
<td>AUX1</td>
</tr>
</tbody>
</table>
### Segment-3 and -4 Packets

<table>
<thead>
<tr>
<th>Segment</th>
<th>TYPE code $b_3b_2b_1b_0$</th>
<th>Slot occupancy</th>
<th>SCO logical transport (1 Mbps)</th>
<th>eSCO logical transport (1 Mbps)</th>
<th>eSCO logical transport (2-3 Mbps)</th>
<th>ACL logical transport (1 Mbps) ptt=0</th>
<th>ACL logical transport (2-3 Mbps) ptt=1</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>1010</td>
<td>3</td>
<td>undefined</td>
<td>undefined</td>
<td>undefined</td>
<td>DM3</td>
<td>2-DH3</td>
</tr>
<tr>
<td></td>
<td>1011</td>
<td>3</td>
<td>undefined</td>
<td>undefined</td>
<td>undefined</td>
<td>DH3</td>
<td>3-DH3</td>
</tr>
<tr>
<td></td>
<td>1100</td>
<td>3</td>
<td>undefined</td>
<td>EV4</td>
<td>2-EV5</td>
<td>undefined</td>
<td>undefined</td>
</tr>
<tr>
<td></td>
<td>1101</td>
<td>3</td>
<td>undefined</td>
<td>EV5</td>
<td>3-EV5</td>
<td>undefined</td>
<td>undefined</td>
</tr>
<tr>
<td>4</td>
<td>1110</td>
<td>5</td>
<td>undefined</td>
<td>undefined</td>
<td>undefined</td>
<td>DM5</td>
<td>2-DH5</td>
</tr>
<tr>
<td></td>
<td>1111</td>
<td>5</td>
<td>undefined</td>
<td>undefined</td>
<td>undefined</td>
<td>DH5</td>
<td>3-DH5</td>
</tr>
</tbody>
</table>
ID Packet

- In addition to the packet types listed in the previous table, the ID packet is also a common packet type.
- ID packet consists of the device access code (DAC) or inquiry access code (IAC).
- It has a fixed length of 68 bits.
Example: An ID Packet (DAC) is used during a Paging Process
NULL Packet

• The NULL packet has no payload and consists of the channel access code (CAC) and packet header only.

• The NULL packet may be used to return link information to the source regarding the success of the previous transmission (ARQN), or the status of the RX buffer (FLOW).

• The NULL packet may not have to be acknowledged.
POLL Packet

- This packet is used by the master to POLL the status of slaves in a piconet.
- It does not have a payload.
- Upon reception of a POLL packet the slave shall respond with a packet even when the slave does not have any information to send unless the slave has scatternet commitments in that timeslot.
- This return packet is an implicit acknowledgement of the POLL packet.
FHS (Frequency Hop Synchronization) Packet

- The FHS packet is used for frequency hop synchronization before a piconet is established.
- The FHS packet contains, among other things, the Bluetooth device address and the clock of the sender.
- The payload contains 144 information bits plus a 16-bit CRC code (total 160 bits).
FHS Packet Format

- The payload is coded with a rate 2/3 FEC with a gross payload length of 240 bits. (In the rate 2/3 coding 5 parity bits are used with every group of 10 information bits.)

<table>
<thead>
<tr>
<th>LSB</th>
<th>Parity bits</th>
<th>LAP</th>
<th>Undefined</th>
<th>SR</th>
<th>Reserved</th>
<th>UAP</th>
<th>NAP</th>
<th>Class of device</th>
<th>LT_ADDR</th>
<th>CLK_{27-2}</th>
<th>Page scan mode</th>
</tr>
</thead>
<tbody>
<tr>
<td>34</td>
<td></td>
<td>24</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>8</td>
<td>16</td>
<td>24</td>
<td>3</td>
<td>26</td>
<td>3</td>
</tr>
</tbody>
</table>

Bluetooth Technology

Syed Masud Mahmud, Ph.D.
SR Field

- This field indicates page scan repetition rate i.e. how often the sender visits the Page Scan state.

<table>
<thead>
<tr>
<th>SR bit format $b_1b_0$</th>
<th>SR mode</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>R0</td>
</tr>
<tr>
<td>01</td>
<td>R1</td>
</tr>
<tr>
<td>10</td>
<td>R2</td>
</tr>
<tr>
<td>11</td>
<td>reserved</td>
</tr>
</tbody>
</table>

FHS Payload

<table>
<thead>
<tr>
<th>LSB</th>
<th>FHS Payload</th>
<th>MSB</th>
</tr>
</thead>
<tbody>
<tr>
<td>34</td>
<td>Parity bits</td>
<td>3</td>
</tr>
<tr>
<td>24</td>
<td>LAP</td>
<td>26</td>
</tr>
<tr>
<td>2</td>
<td>Undef'ed</td>
<td>3</td>
</tr>
<tr>
<td>2</td>
<td>SR</td>
<td>26</td>
</tr>
<tr>
<td>2</td>
<td>Reserved</td>
<td>3</td>
</tr>
<tr>
<td>8</td>
<td>UAP</td>
<td>26</td>
</tr>
<tr>
<td>16</td>
<td>NAP</td>
<td>3</td>
</tr>
<tr>
<td>24</td>
<td>Class of device</td>
<td>26</td>
</tr>
<tr>
<td>3</td>
<td>LT_ADDR</td>
<td>3</td>
</tr>
<tr>
<td>26</td>
<td>CLK$_{27-2}$</td>
<td>3</td>
</tr>
<tr>
<td>3</td>
<td>Page scan mode</td>
<td>3</td>
</tr>
</tbody>
</table>
SR Field

<table>
<thead>
<tr>
<th>SR mode</th>
<th>$T_{\text{page scan}}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>R0</td>
<td>$\leq 1.28 \text{s}$ and $= T_w$ (Continuous Scan)</td>
</tr>
<tr>
<td>R1</td>
<td>$\leq 1.28 \text{s}$</td>
</tr>
<tr>
<td>R2</td>
<td>$\leq 2.56 \text{s}$</td>
</tr>
<tr>
<td>Reserved</td>
<td>-</td>
</tr>
</tbody>
</table>

$T_w$ = Amount of time a device stays in the Page Scan State

$T_{\text{page scan}}$ = Page scan interval
FHS Packet Format

Class of Device Field:

- The *Class of Device* field indicates sender’s device type such as headset, speaker, microphone, printer, modem, etc.

```
<table>
<thead>
<tr>
<th>LSB</th>
<th>FHS Payload</th>
<th>MSB</th>
</tr>
</thead>
<tbody>
<tr>
<td>34</td>
<td>24</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>8</td>
</tr>
<tr>
<td>16</td>
<td>24</td>
<td>3</td>
</tr>
<tr>
<td>26</td>
<td>3</td>
<td></td>
</tr>
</tbody>
</table>
```

- Parity bits
- LAP
- Undefined
- SR
- Reserved
- UAP
- NAP
- Class of device
- LT_ADDR
- CLK_{27-2}
- Page scan mode
**FHS Packet Format**

**LT_ADDR:**

- This is the *primary* LT_ADDR the recipient should use if the FHS packet came from the master at the time of connection setup or role switch.
- At the time of inquiry response, a device should send an FHS packet with LT_ADDR=0.

<table>
<thead>
<tr>
<th>LSB</th>
<th>FHS Payload</th>
<th>MSB</th>
</tr>
</thead>
<tbody>
<tr>
<td>34</td>
<td>Parity bits</td>
<td>3</td>
</tr>
<tr>
<td>24</td>
<td>LAP</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>Undefined</td>
<td>24</td>
</tr>
<tr>
<td>2</td>
<td>SR</td>
<td>8</td>
</tr>
<tr>
<td>2</td>
<td>Reserved</td>
<td>16</td>
</tr>
<tr>
<td>8</td>
<td>UAP</td>
<td>24</td>
</tr>
<tr>
<td>16</td>
<td>NAP</td>
<td>3</td>
</tr>
<tr>
<td>24</td>
<td>Class of device</td>
<td>26</td>
</tr>
<tr>
<td>3</td>
<td>LT_ADDR</td>
<td>3</td>
</tr>
<tr>
<td>26</td>
<td>CLK_{27:2}</td>
<td>3</td>
</tr>
<tr>
<td>3</td>
<td>Page scan mode</td>
<td></td>
</tr>
</tbody>
</table>

---

*Wayne State University College of Engineering*

*Bluetooth Technology*  
*Syed Masud Mahmud, Ph.D.*

110
FHS Packet Format

Page Scan Mode:

- This is a 3-bit field indicating which scan mode is used as default by the sender of the FHS packet.

<table>
<thead>
<tr>
<th>LSB</th>
<th>FHS Payload</th>
<th>MSB</th>
</tr>
</thead>
<tbody>
<tr>
<td>34</td>
<td>Parity bits</td>
<td>3</td>
</tr>
<tr>
<td>24</td>
<td>LAP</td>
<td>26</td>
</tr>
<tr>
<td>2</td>
<td>Undefined</td>
<td>3</td>
</tr>
<tr>
<td>2</td>
<td>SR</td>
<td>24</td>
</tr>
<tr>
<td>8</td>
<td>Reserved</td>
<td>3</td>
</tr>
<tr>
<td>16</td>
<td>UAP</td>
<td>3</td>
</tr>
<tr>
<td>24</td>
<td>NAP</td>
<td>3</td>
</tr>
<tr>
<td>3</td>
<td>Class of device</td>
<td>3</td>
</tr>
<tr>
<td>26</td>
<td>LT_ADDR</td>
<td>3</td>
</tr>
<tr>
<td>3</td>
<td>CLK_{27-2}</td>
<td>3</td>
</tr>
<tr>
<td>3</td>
<td>Page scan mode</td>
<td>3</td>
</tr>
</tbody>
</table>
SCO Packets

• The SCO packets do not include a CRC and are never retransmitted.

SCO Packet Types

• HV1 Packet: Voice
• HV2 Packet: Voice
• HV3 Packet: Voice
• DV Packet: Data and Voice
HV1 Packet

- HV stands for high quality voice.
- The HV1 packet carries 10 information bytes.
- The bytes are protected with a rate 1/3 FEC.
- The payload length is fixed at 240 bits.
- There is no payload header present.
- An HV1 packet can carry 1.25ms of speech at a 64 kb/s rate (64b/msec → 64*1.25 → 80bits (10 bytes) for 1.25msec of voice).
- Thus, an HV1 packet has to be sent after every two time slots.
Reserved Time Slots for HV1 Packets

- An SCO link with HV1 packets uses 100% of the Bluetooth Bandwidth.
HV2 Packet

- The HV2 packet carries 20 information bytes.
- The bytes are protected with a rate 2/3 FEC.
- The payload length is fixed at 240 bits.
- There is no payload header present.
- An HV2 packet can carry 2.5ms of speech at a 64 kb/s rate.
- Thus, an HV2 packet has to be sent after every four time slots.
Reserved Time Slots for HV2 Packets

- An SCO link with HV2 packets uses 50% of the Bluetooth Bandwidth.
HV3 Packet

- The HV3 packet carries 30 information bytes.
- The bytes are not protected by FEC.
- The payload length is fixed at 240 bits.
- There is no payload header present.
- An HV3 packet can carry 3.75ms of speech at a 64 kb/s rate.
- Thus, an HV3 packet has to be sent after every six time slots.
Reserved Time Slots for HV3 Packets

- An SCO link with HV3 packets uses 33.3% of the Bluetooth Bandwidth.
DV Packet

- The DV packet is a combined data-voice packet.
- The payload is divided into a voice field of 80 bits and a data field containing up to 150 bits.
- The voice field is not protected by FEC.
- The data field contains up to 9 information bytes, 1-byte payload header, and includes a 16-bit CRC.
DV Packet (contd.)

- The data field is encoded with a rate 2/3 FEC. The data field is checked for errors and is retransmitted if necessary.

**Reserved Slots for DV Packets**

<table>
<thead>
<tr>
<th>f(k)</th>
<th>f(k+1)</th>
<th>f(k+2)</th>
<th>f(k+3)</th>
</tr>
</thead>
</table>

Master

Slave

625 μs
## SCO Packet Summary

<table>
<thead>
<tr>
<th>Type</th>
<th>Payload Header (bytes)</th>
<th>User Payload (bytes)</th>
<th>FEC</th>
<th>CRC</th>
</tr>
</thead>
<tbody>
<tr>
<td>HV1</td>
<td>na</td>
<td>10</td>
<td>1/3</td>
<td>no</td>
</tr>
<tr>
<td>HV2</td>
<td>na</td>
<td>20</td>
<td>2/3</td>
<td>no</td>
</tr>
<tr>
<td>HV3</td>
<td>na</td>
<td>30</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>DV&lt;sup&gt;1&lt;/sup&gt;</td>
<td>1 D</td>
<td>10+(0-9) D</td>
<td>2/3 D</td>
<td>yes D</td>
</tr>
</tbody>
</table>

*Items followed by D relate to data field only*
Extended SCO (eSCO) Packets

• Similar to SCO, eSCO reserves slots between the master and a particular slave.
• The eSCO packets may be used for 64kb/s speech transmission as well as transparent data at 64kb/s and other rates.
• eSCO packets have a 16-bit CRC to detect errors.
Extended SCO (eSCO) Packets

• Unlike SCO packets, retransmissions are allowed for eSCO packets.
• A secondary set of LT_ADDR is assigned to eSCO slaves to select them during retransmission.
eSCO Payload Length

- The payload length of eSCO packets is variable.
- The payload length is set during the LMP eSCO setup and remains fixed until the link is removed or renegotiated.
eSCO Retransmission Window

- In addition to the reserved slots the eSCO transport may have a retransmission window after the reserved slots.
- The reserved slots and the retransmission window together form the complete eSCO window.
eSCO Window for Single-Slot Packets

- The master shall only retransmit in the retransmission window if there are enough slots left for both the master and slave packets to complete in the retransmission window.
eSCO Window for Asymmetric Traffic

- eSCO Instant
- Reserved Slots
- Retransmission Window
- eSCO Window

Syed Masud Mahmud, Ph.D.
## eSCO Packets

<table>
<thead>
<tr>
<th>Type</th>
<th>Slot occupancy</th>
<th>Data Rate (Mbps)</th>
<th>Payload Header (bytes)</th>
<th>User Payload (bytes)</th>
<th>FEC</th>
<th>CRC</th>
</tr>
</thead>
<tbody>
<tr>
<td>EV3</td>
<td>1</td>
<td>1</td>
<td>na</td>
<td>1-30</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>EV4</td>
<td>3</td>
<td>1</td>
<td>na</td>
<td>1-120</td>
<td>2/3</td>
<td>Yes</td>
</tr>
<tr>
<td>EV5</td>
<td>3</td>
<td>1</td>
<td>na</td>
<td>1-180</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>2-EV3</td>
<td>1</td>
<td>2</td>
<td>na</td>
<td>1-60</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>2-EV5</td>
<td>3</td>
<td>2</td>
<td>na</td>
<td>1-360</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>3-EV3</td>
<td>1</td>
<td>3</td>
<td>na</td>
<td>1-90</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>3-EV5</td>
<td>3</td>
<td>3</td>
<td>na</td>
<td>1-540</td>
<td>No</td>
<td>Yes</td>
</tr>
</tbody>
</table>
ACL Packets

- ACL packets have payload header and CRC.
- ACL packets are retransmitted if necessary.

Format of ACL Packets

<table>
<thead>
<tr>
<th>ACCESS CODE</th>
<th>HEADER</th>
<th>PAYLOAD</th>
<th>PAYLOAD</th>
</tr>
</thead>
<tbody>
<tr>
<td>68/72 bits</td>
<td>54 bit</td>
<td>8 or 16 bits</td>
<td>N bytes (N depends on Packet Type)</td>
</tr>
</tbody>
</table>

LSB

Bluetooth Technology

Syed Masud Mahmud, Ph.D.
# ACL Packets with Basic Data Rate

<table>
<thead>
<tr>
<th>Type</th>
<th>No of Slots</th>
<th>Payload Header (bytes)</th>
<th>User Payload (bytes)</th>
<th>FEC</th>
<th>CRC</th>
</tr>
</thead>
<tbody>
<tr>
<td>DM1</td>
<td>1</td>
<td>1</td>
<td>0-17</td>
<td>2/3</td>
<td>yes</td>
</tr>
<tr>
<td>DH1</td>
<td>1</td>
<td>1</td>
<td>0-27</td>
<td>no</td>
<td>yes</td>
</tr>
<tr>
<td>DM3</td>
<td>3</td>
<td>2</td>
<td>0-121</td>
<td>2/3</td>
<td>yes</td>
</tr>
<tr>
<td>DH3</td>
<td>3</td>
<td>2</td>
<td>0-183</td>
<td>no</td>
<td>yes</td>
</tr>
<tr>
<td>DM5</td>
<td>5</td>
<td>2</td>
<td>0-224</td>
<td>2/3</td>
<td>yes</td>
</tr>
<tr>
<td>DH5</td>
<td>5</td>
<td>2</td>
<td>0-339</td>
<td>no</td>
<td>yes</td>
</tr>
<tr>
<td>AUX1</td>
<td>1</td>
<td>1</td>
<td>0-29</td>
<td>no</td>
<td>no</td>
</tr>
</tbody>
</table>
## ACL Packets with Enhanced Data Rate

<table>
<thead>
<tr>
<th>Type</th>
<th>Slot occupancy</th>
<th>Data Rate (Mbps)</th>
<th>Payload Header (bytes)</th>
<th>User Payload (bytes)</th>
<th>FEC</th>
<th>CRC</th>
</tr>
</thead>
<tbody>
<tr>
<td>2-DH1</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>0-54</td>
<td>no</td>
<td>yes</td>
</tr>
<tr>
<td>2-DH3</td>
<td>3</td>
<td>2</td>
<td>2</td>
<td>0-367</td>
<td>no</td>
<td>yes</td>
</tr>
<tr>
<td>2-DH5</td>
<td>5</td>
<td>2</td>
<td>2</td>
<td>0-679</td>
<td>no</td>
<td>yes</td>
</tr>
<tr>
<td>3-DH1</td>
<td>1</td>
<td>3</td>
<td>2</td>
<td>0-83</td>
<td>no</td>
<td>yes</td>
</tr>
<tr>
<td>3-DH3</td>
<td>3</td>
<td>3</td>
<td>2</td>
<td>0-552</td>
<td>no</td>
<td>yes</td>
</tr>
<tr>
<td>3-DH5</td>
<td>5</td>
<td>3</td>
<td>2</td>
<td>0-1021</td>
<td>no</td>
<td>yes</td>
</tr>
</tbody>
</table>
Payload Header Format

Payload Header for Basic Rate Single-Slot ACL Packets

Payload Header for Multi-Slot and all EDR ACL Packets
## LLID Code of Payload Header

<table>
<thead>
<tr>
<th>LLID code $b_1 b_0$</th>
<th>Logical Link</th>
<th>Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>NA</td>
<td>undefined</td>
</tr>
<tr>
<td>01</td>
<td>ACL-U</td>
<td>Continuation fragment of an L2CAP message</td>
</tr>
<tr>
<td>10</td>
<td>ACL-U</td>
<td>Start of an L2CAP message or no fragmentation</td>
</tr>
<tr>
<td>11</td>
<td>ACL-C</td>
<td>LMP message</td>
</tr>
</tbody>
</table>
FLOW Bit of Payload Header

- The flow indicator in the payload is used to control the flow at the L2CAP level. FLOW=1 means GO and FLOW=0 means STOP.

- Real-time flow control shall be carried out at the packet level by the link controller via the flow bit in the packet header.

- With the payload flow bit, traffic from the remote L2CAP can be controlled.
Bluetooth States
States

- Unconnected State
  - Standby
    - Inquiry
      - Unknown Address
    - Page
      - Known Address
  - Transmit
    - Connected
      - Park
        - Releases LT_ADDR
      - Hold
        - Keeps LT_ADDR
      - Sniff
        - Keeps LT_ADDR

- Connecting States
- Active States
- Low Power State

Bluetooth Technology
Syed Masud Mahmud, Ph.D.
Devices in Sniff and Hold Modes are Still Connected to the Piconet
Sniff Mode

• In the sniff mode, the slave’s listen activity is reduced.

• If a slave participates on an ACL link, it still listens to that link but at a **reduced rate**.

• Devices in sniff mode may use the absent periods to engage in activity on another piconet, or to enter reduced power mode.

• Sniff mode **does not affect** the SCO or eSCO links that may be active.
Sniff Anchor Points

• When a slave is in sniff mode, the master shall transmit to the slave only in specified time slots starting at sniff anchor points.

• The sniff anchor points are spaced regularly with an interval of $T_{\text{sniff}}$. 
Hold Mode

- The ACL link is not supported in the Hold Mode.
- Already established SCO or eSCO links are still available.
- With the hold mode, capacity can be made free to do other things like scanning, paging, inquiring, or attending another piconet.
Hold Mode

- The device in hold mode can also enter a low-power sleep mode.
- Prior to entering hold mode, master and slave agree on the time duration the slave remains in hold mode.
Park State

- When a slave does not need to participate on the piconet channel, but still wants to remain synchronized to the channel, it can enter the low-power park mode.
- The slave gives up its LT_ADDR.
- Instead, it receives two new addresses to be used in the park mode
  - PM_ADDR: 8-bit Parked Member Address
  - AR_ADDR: 8-bit Access Request Address
Park State

• The PM_ADDR distinguishes a parked slave from the other parked slaves.
• This address is used in the master-initiated unpark procedure. In addition to the PM_ADDR, a parked slave can also be unparked by its 48-bit BD_ADDR.
• The AR_ADDR is used by the slave in the slave-initiated unpark procedure.
• All messages sent to the parked slaves have to be carried by broadcast packets (LT_ADDR=0).
Beacon Train

• A beacon train is used to provide regular synchronization to all parked slaves.
• The beacon train consists of one or more equidistant beacon slots which is transmitted periodically with a constant time interval ($T_B$).
Beacon Train Format

$N_B$ = Number of beacon slots in a beacon train

$T_B$ = Interval between beacon trains

$\Delta_B$ = Interval between beacon slots

The master sends these parameters to a slave when the slave has been parked.
Beacon Access Window

- In addition to the beacon slots, an access window is defined where the parked slaves can send requests to be unparked.
- To increase reliability, the access window may be repeated several times.
Access Procedure Using the Polling Technique

- The slave shall send an access request in the proper half slot if a broadcast packet has been received in the preceding slot.
Disturbance of Access Window by SCO Traffic

• In the following example, slaves with AR_ADDR=3 and 4 cannot send access requests due to the presence of reserved SCO slots.
Extended Sleep Interval of Parked Slaves

A parked slave may not wake up at every beacon instant. Instead, it may wake up after several beacon intervals.
Master Initiated Unpark Procedure

- The master can unpark a parked slave by broadcasting a message on the beacon slots.
- The master shall use either the slave’s PM_ADDR, or its BD_ADDR.
- The message also includes the logical transport address LT_ADDR the slave shall use after it has re-entered the piconet.
Multiple Parked Slaves May Be Unparked Simultaneously

- The unpark message may include a number of slave addresses so that multiple slaves may be unparked simultaneously.

- For each slave, a different LT_ADDR shall be assigned.
Bluetooth Security
Entities used in Authentication and Encryption Procedures

- Four different entities are used for maintaining security at the Bluetooth Link Layer: a **public address** which is unique for each Bluetooth device, **two secret keys**, and a **random number** which is different for each new transaction.
### Four Entities for Bluetooth Security

<table>
<thead>
<tr>
<th>Entity</th>
<th>Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>BD_ADDR</td>
<td>48 bits</td>
</tr>
<tr>
<td>Private user key, authentication</td>
<td>128 bits</td>
</tr>
<tr>
<td>Private user key, encryption</td>
<td>8-128 bits</td>
</tr>
<tr>
<td>configurable length (byte-wise)</td>
<td></td>
</tr>
<tr>
<td>RAND</td>
<td>128 bits</td>
</tr>
</tbody>
</table>
Authentication and Encryption Keys

- The secret keys are derived during initialization and are further never disclosed.
- The *encryption key* is derived from the *authentication key* during the authentication process.
- Each time encryption is activated, a new encryption key is generated.
- The authentication key, once established, will be more static.
Authentication and Encryption Keys

• The particular application running on the Bluetooth device decides when, or if, to change the authentication key.
• For the authentication algorithm, the size of the key used is always 128 bits.
• For the encryption algorithm, the key size may vary between 1 and 16 octets (8 - 128 bits).
• The size of the encryption key shall be configurable for two reasons.
Encryption Key

• The first has to do with many different requirements imposed on cryptographic algorithms in different countries – both with respect to export regulations and official attitudes towards privacy in general.

• The second reason is to facilitate a future upgrade path for the security without the need of a costly redesign of the algorithms and encryption hardware; increasing the effective key size is the simplest way to combat increased computing power at the opponent side.
Link Key

• The authentication key is also known as the *Link Key*.
• In order to accommodate for different types of applications, four types of link keys have been defined:
  - the initialization key $K_{\text{init}}$
  - the unit key $K_A$
  - the combination key $K_{AB}$
  - the temporary key $K_{\text{master}}$
Initialization Key $K_{\text{init}}$

- The initialization key, $K_{\text{init}}$, is used as the link key during the initialization process when no combination or unit keys have been defined and exchanged yet or when a link key has been lost.
- It is derived from four entities: the Bluetooth address $BD_{\text{ADDR}}_B$ of the claimant unit, a PIN code, the length of the PIN ($L$), and a random number $R_{\text{RAND}}_A$ issued (and created) by the verifier.
Generation of the Initialization Key when Unit B wants to Establish a Connection with Unit A
Unit A is Authenticating Unit B
Initialization Procedures

- All initialization procedures consist of the following five parts:
  - Generation of an initialization key
  - Authentication
  - Generation of a link key
  - Link key exchange
  - Generation of an encryption key in each unit
Other Link Keys

- There are three other types of links keys:
  - the unit key $K_A$
  - the combination key $K_{AB}$
  - the temporary key $K_{master}$
Unit Key $K_A$

- The unit key $K_A$ is generated in, and therefore dependent on, a single unit A.
- The unit key is generated once at installation of the Bluetooth unit.
- Thereafter, it is very rarely changed.
Generation of Unit Key $K_A$

- The unit key $K_A$ (of unit A) is generated using the BD_ADDR of unit A and a random number RAND.

![Diagram showing the generation of $K_A$]

- The BD_ADDR of unit A is processed with a key $E_{21}$ to generate $K_A$. The BD_ADDR is split into two parts: 48 bits and 128 bits, then XORed with the random number RAND to produce $K_A$.
Exchanging Unit Key $K_A$

- If Unit A wants Unit B to use its Unit Key $K_A$, then Unit A sends its unit key to Unit B as follows:

$K_A$ \rightarrow \text{UNIT A} \quad \text{UNIT B} \quad \rightarrow \quad K_{BA} = K_A$

When the unit key has been exchanged, the initialization key shall be discarded in both units.
Using Unit Key $K_A$

- Bluetooth units which have little memory to store keys, or, when installed in equipment that must be accessible to a large group of users, will preferably use their own unit key.
- In that case, they only have to store a single key.
Combination Key $K_{AB}$

- The combination key is derived from information in both units A and B, and is therefore always dependent on two units.
- The combination key is derived for each new combination of two Bluetooth units.
Combination Key $K_{AB}$

- Applications that require a higher security level preferably use the combination keys.
- These applications will require more memory since a combination key for each link to a different Bluetooth unit has to be stored.
Generation of Combination Key $K_{AB}$

- First, each unit generates a random number, say $LK_{RAND_A}$ and $LK_{RAND_B}$.
- Then, the two random numbers $LK_{RAND_A}$ and $LK_{RAND_B}$ are exchanged securely between Unit A and Unit B by XORing with the current link key, say $K$.

\[
C_A = LK_{RAND_A} \oplus K \\
LK_{RAND_B} = C_B \oplus K
\]

\[
\text{Unit A} \\
C_A \\
\text{Unit B} \\
C_B = LK_{RAND_B} \oplus K
\]
Generation of Combination Key $K_{AB}$

- Both Unit A and Unit B generate two random numbers $LK_{KA}$ and $LK_{KB}$ by using $E_{21}$ algorithm on the previously generated random numbers and their Bluetooth addresses.
- Note that both units know their Bluetooth addresses, which were exchanged during the inquiry and paging process.

$$LK_{KA} = E_{21}(LK_{RAND_A}, BD_{ADDR_A})$$

$$LK_{KB} = E_{21}(LK_{RAND_B}, BD_{ADDR_B})$$
Generation of Combination Key $K_{AB}$

![Diagram of Generation of Combination Key $K_{AB}$]

- $LK_{RAND}\_A$ → $E_{21}$ → $LK_{K}\_A$
- $BD\_ADDR\_A$ → $LK_{K}\_A$
- $LK_{RAND}\_B$ → $E_{21}$ → $LK_{K}\_B$
- $BD\_ADDR\_B$ → $LK_{K}\_B$

(Unit A)

(Unit B)

Bluetooth Technology

Syed Masud Mahmud, Ph.D.
Generation of Combination Key $K_{AB}$

- These random numbers $LK_{K_A}$ and $LK_{K_B}$ constitute the units’ contribution to the combination key that is to be created.
- The combination key $K_{AB}$ is generated by XORing $LK_{K_A}$ and $LK_{K_B}$

$$K_{AB} = LK_{K_A} \oplus LK_{K_B}$$

**Unit A**

$$K_{BA} = LK_{K_A} \oplus LK_{K_B} = K_{AB}$$

**Unit B**
Generation of Combination Key $K_{AB}$

\[
\begin{align*}
L_K_{K_A} &= E_{21}(L_K_{RAND_A}, BD_{ADDR_A}) \\
C_A &= L_K_{RAND_A} \oplus K \\
L_K_{RAND_B} &= C_B \oplus K \\
L_K_{K_B} &= E_{21}(L_K_{RAND_B}, BD_{ADDR_B}) \\
K_{AB} &= L_K_{K_A} \oplus L_K_{K_B}
\end{align*}
\]

Generating a combination key. The old link key (K) shall be discarded after the exchange of a new combination key has succeeded.
Master Key $K_{\text{master}}$

- The master key, $K_{\text{master}}$, is a link key only used during the current session.
- It will replace the original link key only temporarily.
- For example, this may be utilized when a master wants to reach more than two Bluetooth units simultaneously using the same encryption key.
Modifying the Link Keys

• In certain circumstances, it is desirable to be able to modify the link keys.
• A link key based on a unit key can be changed, but not very easily.
• The unit key is created once during the first use.
Modifying the Link Keys (contd.)

• Changing the unit key is a less desirable alternative, as several units may share the same unit key as link key (think of a printer whose unit key is distributed among all users using the printer’s unit key as link key).

• Changing the unit key will require re-initialization of all units trying to connect.
Modifying the Link Keys (contd.)

- In certain cases, this might be desirable; for example to deny access to previously allowed units.
- If the key change concerns combination keys, then the procedure is rather straightforward.
Modifying the Link Keys (contd.)

• The change procedure can be identical to the procedure used to generate the combination key, using the current value of the combination key as link key.
• This procedure can be carried out at any time after the authentication and encryption start.
Modifying the Link Keys (contd.)

• Since the combination key corresponds to a single link, it can be modified each time this link is established.

• This will improve the security of the system since then old keys lose their validity after each session.
Modifying the Link Keys (contd.)

- Of course, starting up an entirely new initialization procedure is also a possibility.
- In that case, user interaction is necessary since a PIN is required in the authentication and encryption procedures.
The PIN is the Cause of Security Problems in a Bluetooth System

• Since the initialization key $K_{\text{init}}$ is generated using a PIN and all other procedures depend on $K_{\text{init}}$, the PIN is the main cause of security problems in Bluetooth systems.

• For most systems the PIN is a 4-digit number (0000 – 9999).
The PIN is the Cause of Security Problems in a Bluetooth System

- For two Bluetooth devices to form an ad-hoc network (automatically), the PINs of the two devices must be same.
- In 50% of the devices it is always 0000, so that they can automatically form an ad-hoc network when they come close to each other.
The Problem with a 4-Digit PIN

- Since the PIN is a 4-digit number, the hackers may be able to get it using a *Brute Force* attack.
- If a user is required to manually enter a PIN, every time the user is going to use a Bluetooth device for a certain application, then it won’t be convenient for the user if the PIN is very long.
- As a result, the security threat still remains in Bluetooth devices.
Bluetooth Network for Vehicular Applications
An In-Vehicle Wireless Network

NDM (Wireless Access Point)

Remote Keyless Entry Device
FAX
PDA
Phone
Headphone
Notebook

Wireless Link

CAN Bus

Bluetooth Technology

Syed Masud Mahmud, Ph.D.
How to Build a Secure In-Vehicle Wireless Network?

1. Make the PIN a very large number so that it can’t be easily obtained through a Brute Force attack.

2. Make the PIN transparent to the user so that the user doesn’t have to remember it.
How to Build a Secure In-Vehicle Wireless Network?

3. *Change* the PIN from time to time so that it can’t be obtained by analyzing data recorded over a period of several communication sessions.

4. Use different PINs for different in-vehicle devices (e.g. laptop, PDA, cell phone, etc.) so that PINs of other devices can’t be obtained from a stolen device.
How to Build a Secure In-Vehicle Wireless Network?

5. Don’t allow two in-vehicle devices to start a communication without being authenticated by a *Gate-Way* device.

6. Register all devices (to be used in the vehicle) to the Gate-Way device, so that the Gate-Way device can know who is allowed to participate in a secured communication.
How to Build a Secure In-Vehicle Wireless Network?

7. The user should be able to *remove* a stolen device from the list of registered devices, maintained in the Gate-Way device.
THE END
Reference


(Most of the figures and materials, presented in this document, have been taken from the above article.)