

---

---

# Bluetooth

## *Part 6: Logical Link Control and Adaptation Protocol*

**Kjell Jørgen Hole**

UiB



Last updated 03.03.09

Mail: [Kjell.Hole@ii.uib.no](mailto:Kjell.Hole@ii.uib.no)

URL: [www.kjhole.com](http://www.kjhole.com)

## Outline

[KJhole.com](http://KJhole.com)

- Logical Link Control and Adaptation Protocol (**L2CAP**) def.
- L2CAP signaling
- Establishing and configuring L2CAP connection
- Transferring data, disconnecting, and timeouts
- One-way transmission to group of Bluetooth devices
- Java L2CAP vs. RFCOMM programming

## Definition of L2CAP (1)

KJhole.com

**L2CAP** Takes data from higher protocol layers and *applications* and sends it over the lower layers. L2CAP passes packets either to the HCI, or in a host-less system, directly to LM

- L2CAP utilizes ACL connections. A separate control function is required to set up and close down these connections
- L2CAP transfers data, not audio (voice over IP regarded as data)

6.3

## Definition of L2CAP (2)

KJhole.com

- L2CAP has many functions:
  - multiplexing to allow several higher layer links, possibly based on different protocols, to pass across a single ACL connection
  - segmentation and reassembly to allow transfer of larger packets than lower layers support
  - Quality of Service (QoS) management for higher layer protocols
  - optional error control and retransmissions

6.4

## L2CAP Signaling (1)

KJhole.com

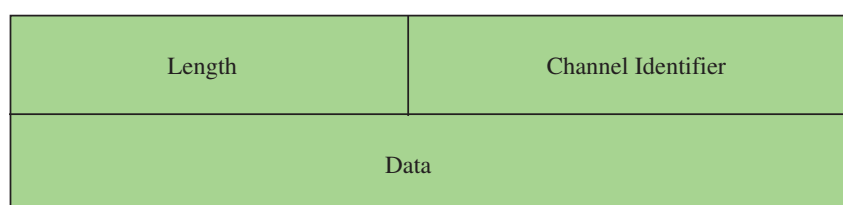
- L2CAP labels packets with channel numbers to differentiate between higher layer channels
- L2CAP entities communicate with each other using control channels with a special channel number to handle connecting, configuring, and disconnecting L2CAP connections
- A second channel number is reserved for receiving multicast packets

6.5

## L2CAP Signaling (2)

KJhole.com

- An L2CAP packet contains a **length** field (2 bytes), a **channel identifier**, or channel number, field (2 bytes), and a **data** field (0–65,535 bytes)



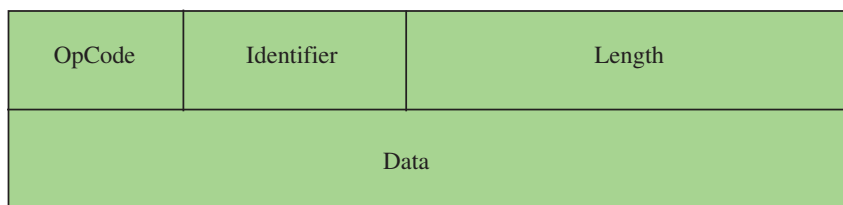
**Figure 6-1** Structure of L2CAP packet

6.6

- Structure of L2CAP command is shown in Figure 6-2
  - OpCode** (1 byte) identifying contents of command
  - Identifier** (1 byte) used to pair up requests and responses
  - Length** (2 bytes) of data field
- Many commands can be sent within the data field of one L2CAP packet

6.7

## L2CAP Command Structure



**Figure 6-2** Structure of L2CAP command

6.8

## Establishing a Connection

---

KJhole.com

- To establish a link, a higher layer protocol sends a request to the L2CA layer to connect
- If there is no ACL connection, then L2CA sends a request to a lower layer (HCI or LM) to connect
- The steps involved in setting up an ACL connection are many and quite complex (involving many HCI and LM commands)
- Once an ACL connection is established across the lower layers, L2CAP packets can be sent across it

6.9

## Configuring a Connection (1)

---

KJhole.com

- Once a connection has been established, it must be configured. Parameters which can be configured are:

**Maximum Transmission Unit (MTU)** maximum size in bytes of packet payload a device is willing to accept, no more than 65,535 bytes

**Flush timeout** is the amount 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

6.10

## Configuring a Connection (2)

---

KJhole.com

- Once the initiating device has configured the outbound channel going to the accepting device, the accepting device can configure the return channel
- If two devices have difficulty deciding on a mutually agreeable set of parameters, messages could be exchanged for maximum two minutes

6.11

## Transferring Data

---

KJhole.com

- Once a connection has been established and configured, it can be used to transfer data
- How the higher layers pass data to and from the L2CA layer is implementation dependent (standard contains suggestions)
- Segmentation and reassembly needed because
  - some higher layer protocols use packet sizes larger than those which Bluetooth can handle
  - some HCI implementations only support small packets. Must support packets carrying up to 255 bytes of data

6.12

## Disconnecting and Timeouts

KJhole.com

- There are 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) timer is started. If the RTX timer expires before a response is received, the channel may be disconnected

6.13

## Connectionless Data Channels

KJhole.com

- L2CAP provides connectionless channels to connect a device to a group of one or more other devices in a single direction
- Connectionless channels cannot be configured for QoS
- L2CAP defines messages to disable connectionless traffic

6.14

# Java ME

6.15

## L2CAP vs. RFCOMM Programming

- JSR-82 provides a *packet-based* API for L2CAP as opposed to the *stream-based* APIs available for RFCOMM and OBEX
- The L2CAP API is the right choice for an app if
  - the app implements a Bluetooth profile that uses the L2CAP protocol and that profile does not use RFCOMM or OBEX
  - the app implements a new custom protocol that is packet oriented

6.16

## L2CAP Flow Control

KJhole.com

- The programmer must provide some form of flow control because L2CAP does not provide any
- We will only use the RFCOMM API in this course

6.17

## Summary

KJhole.com

- All applications must use L2CAP to send data. It is also used by Bluetooth's higher layers such as RFCOMM
- L2CAP provides the facilities needed by higher layer protocols to communicate across a Bluetooth link:
  - establishing links across ACL channels using L2CAP commands
  - multiplexing between different higher layer entities
  - providing segmentation and reassembly facilities

6.18