Logical Link Control and Adaptation Protocol

Definition of L2CAP

L2CAP takes data from higher protocol layers and applications and sends it over the lower layers. L2CAP passes packets either to HCI or in a host-less system, directly to LM. L2CAP utilities ACL connections. A separate control function is required to set up and close down these connections. L2CAP transfers data, not audio (voice or IP regard as data)

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 Services (QoS) management for higher layer protocols.

L2CAP Signaling

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 multi cast packets. 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)

Length	**Channel Identifier**
Data	

Structure of L2CAP packet

L2CAP Command

Structure of L2CAP command is divided into:

- - 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

OpCode	**Identifier**	**Length**
Data		

Structure of L2CAP command

Establishing a Connection

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 the 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.

Configuring a Connection

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.

(C) Copyright Nokia 2014. All rights reserved.

- **QoS** option can select best effort, or a guaranteed QoS

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.

Transferring Data

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.

Disconnection and Timeouts

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.

Connectionless Data Channels

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.

Summary

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.

(C) Copyright Nokia 2014. All rights reserved.