stonestreet one

Sending L2CAP Data Packets

VERSION 2.0

Sending L2CAP packets with the help of two or more HCI data packets considering the flags and other headers

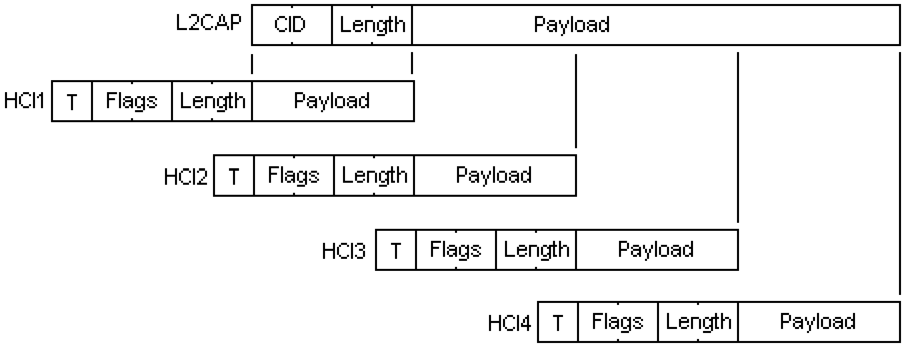
Bluetooth™

Sending L2CAP Data Packets

Sending the L2CAP packets with help of two or more HCI data packets considering the flags and other headers

Sending an L2CAP packet within two or more HCI packets

L2CAP packets can carry large data units. To send this large packet, the host software has to divide it into smaller quantities. These small chunks are sent to the link manager aided by the HCI interface. The small HCI packets need to know if they are the first part of the big L2CAP packet or a following part in order to be puzzled together again at the receiving end.



Take the data that you want to transmit and add four preceding bytes. These bytes are the header of the L2CAP packet and consists of a two byte Channel Identifier (CID) and two bytes of total L2CAP payload length (Total Length). The CID codes are up to the implementation to allocate for different channels and can be between 0x0040 – 0xFFFF.

The complete L2CAP packet is then divided upon a number of HCI data packets and the data is put into the payload of the ACL data packets. It is important that the packet boundary flag of the first HCI packet is set to start packet and the following HCI packets is set to continues packets.

Let’s take an example

Suppose we want to send 10 bytes of payload. Of some reason we want these to be sent in two HCI packets. First we create the L2CAP packet. The bytes in the packets are listed in the order they should be sent to the module. In this example we choose The CID to be 0x0040.

The L2CAP data packet with 10 bytes of payload:

```
0x0A      //L2CAP Length LSB
0x00      //L2CAP Length MSB
0x40      //CID LSB
0x00      //CID MSB
0x01      //Start of payload…..
0x02
0x03
0x04
0x05
0x06
0x07
0x08
0x09
0x0A
```

This L2CAP packet will now be divided into two HCI data packets. The active connection handle in this example is 0x0001.

This first HCI packet will have the packet boundary flag set to “Start packet” and contain eight bytes of payload; four of them will be the header of the L2CAP packet.

First HCI packet details:

```
0x02      //Transport HCI ACL data
0x01      //Connection handle LSB
0x20      //Connection handle MSB + Flag: START + Flag: point to point
0x08      //HCI data length LSB
0x00      //HCI data length MSB
0x0A      //L2CAP total length LSB;    Start of HCI payload
0x00      //L2CAP total length MSB
0x40      //L2CAP CID LSB
0x00      //L2CAP CID MSB
```

```
0x01        //Start of  L2CAP payload
0x02
0x03
0x04
```

The second HCI packet will have the packet boundary flag set to “Continued packet” and contain six bytes of payload. This will be enough to hold the last part of the L2CAP packet.

Second HCI packet details:

```
0x02        //Transport HCI ACL data
0x01        //Connection handle LSB
0x10        //Connection handle MSB + Flag: CONTINUED + Flag: point to
point
0x06        //HCI data length LSB
0x00        //HCI data length MSB
0x05        //Continuation of L2CAP payload; Start of HCI payload
0x06
0x07
0x08
0x09
0x0A
```

When an ACL connection is present and these two HCI data packets are sent to one of the boards, then it will arrive on the other board as one single HCI data packet. Of course, it will arrive in separate HCI packets in another case where the L2CAP packet is larger than the actual HCI data buffer size.