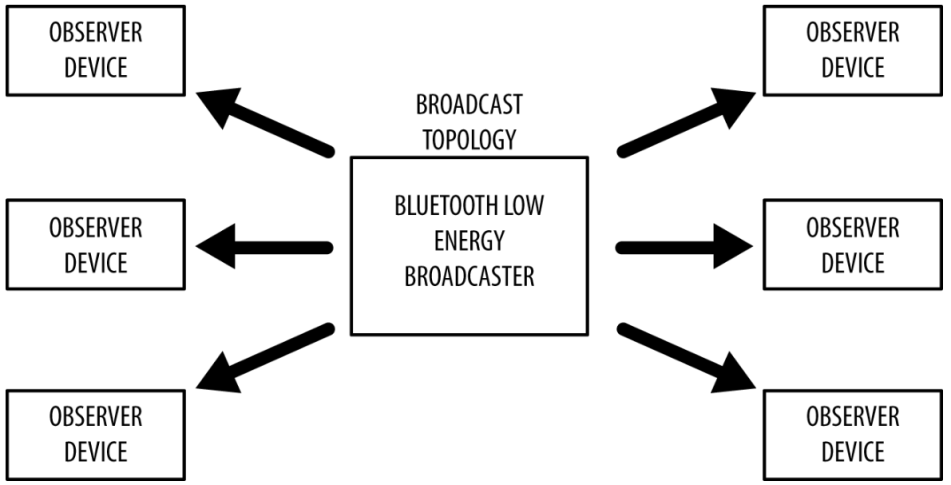
# 1. Overview

A Bluetooth Low Energy device can communicate with the outside world in two ways: broadcasting or connections. Each mechanism has its own advantages and limitations.

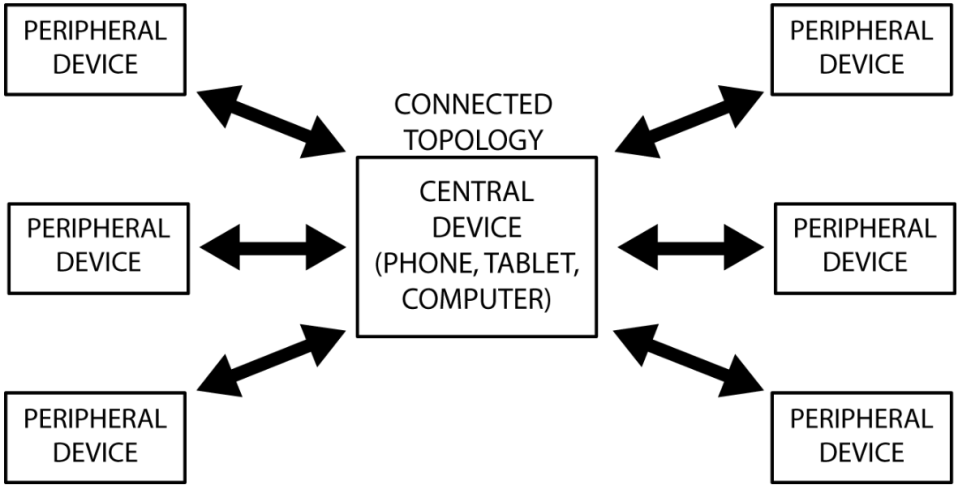
## 1.1 Broadcasting

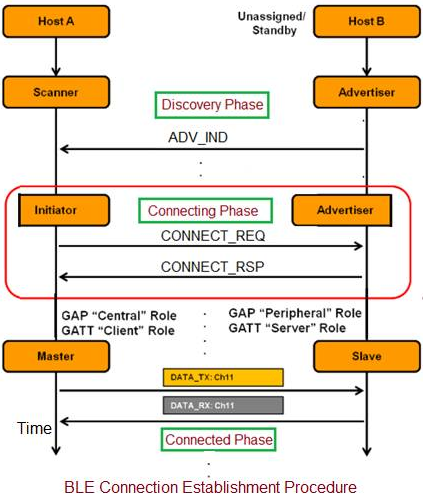
Using connectionless broadcasting, you can send data out to any scanning device or receiver in listening range.



## 1.2 Connection

If you need to transmit data in both directions, or if you have more data than the two advertising payloads can accommodate, you will need to use a connection. A connection is a permanent, periodical data exchange of packets between two devices. It is therefore inherently private (the data is sent to and received by only the two peers involved in a connection, and no other device unless it’s indiscriminately sniffing).





The Link Layer enters the Connection state when

(1) an initiator sends a CONNECT\_IND PDU on the primary advertising physical channel to an

advertiser,

(2) an advertiser receives a CONNECT\_IND PDU on the primary advertising physical channel from an initiator,

(3) an advertiser sends an AUX\_CONNECT\_RSP PDU on the secondary advertising physical channel to an initiator,

(4) or an initiator receives an AUX\_CONNECT\_RSP PDU on the secondary advertising physical channel from an advertiser.

A connection is only considered to be established once a data physical channel packet has been received from the peer device.

If the connection is first created using the CONNECT\_IND PDU on the primary advertising physical channel, it shall use the LE 1M PHY in both directions. If the connection is first created on the secondary channel using the AUX\_CONNECT\_REQ and AUX\_CONNECT\_RSP PDUs, it shall use the same PHY in both directions as was used for the AUX\_CONNECT\_REQ and AUX\_CONNECT RSP PDU.

When two devices are in a connection, the two devices act in different roles. A Link Layer in the Master Role is called a master. A Link Layer in the Slave Role is called a slave. The master controls the timing of a connection event. A connection event is a point of synchronization between the master and the slave.

The slave shall always send a packet if it receives a packet from the master regardless of a valid CRC matched or not.

# 2. Connection setup

## 2.1 Master Role

The master shall start to send the first packet within the transmit window. The first packet sent in the Connection State by the master determines the anchor point for the first connection event, and therefore the timings of all future connection events in this connection.

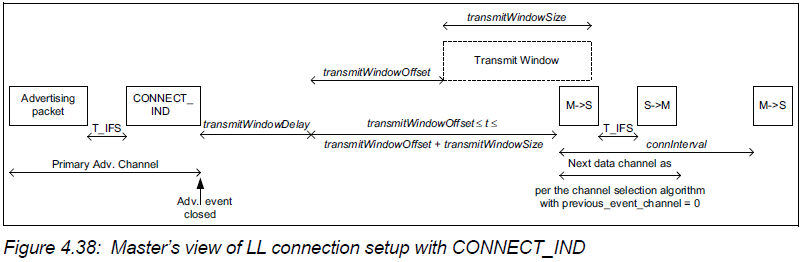
The second connection event anchor point shall be connInterval after the first connection event anchor point.

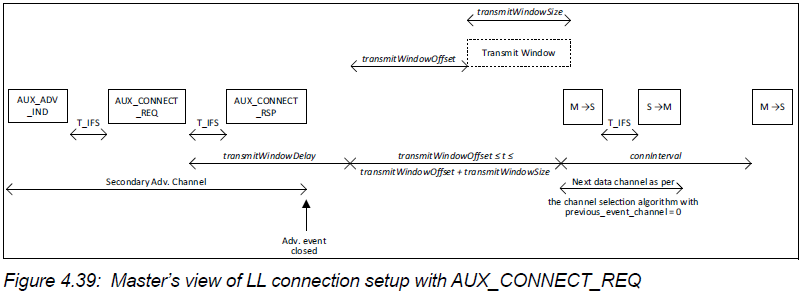
The transmit window starts at transmitWindowDelay + transmitWindowOffset after the end of the packet containing the CONNECT\_IND PDU or AUX\_CONNECT\_REQ PDU, and the transmitWindowSize parameter shall define the size of the transmit window. The connInterval is used in the calculation of the maximum offset and size of the transmit window.

The transmitWindowOffset and transmitWindowSize parameters are determined by the Link Layer. The transmitWindowOffset shall be a multiple of 1.25 ms in the range 0ms to connInterval.

The transmitWindowSize shall be a multiple of 1.25 ms in the range 1.25 ms to the lesser of 10 ms and (connInterval - 1.25 ms).

The value of transmitWindowDelay shall be 1.25 ms when a CONNECT\_IND PDU is used, 2.5 ms when an AUX\_CONNECT\_REQ PDU is used on an LE Uncoded PHY, and 3.75 ms when an AUX\_CONNECT\_REQ PDU is used on the LE Coded PHY.



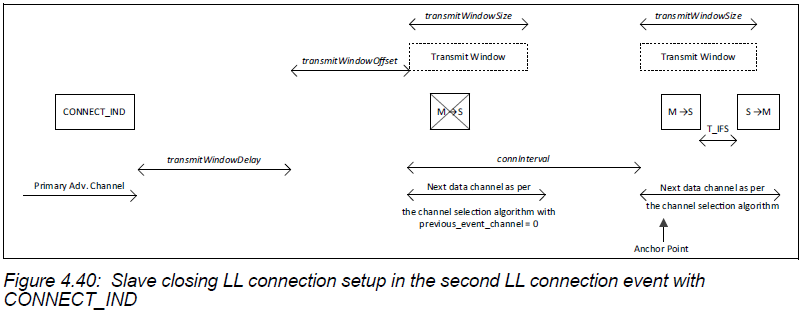


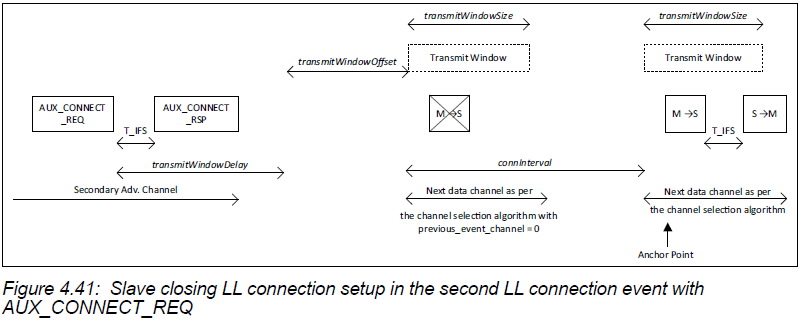
## 2.2 Slave Role

The slave shall start to listen for the first packet within the transmit window, while doing so, it should perform the window widening. It is permitted that the master’s first packet can extend beyond the transmit window, and therefore the slave must take this into account.

If a packet is not received in a transmit window, the slave shall attempt to receive a packet in a subsequent transmit window. A subsequent transmit window shall start connInterval after the start of the previous transmit window, with the same transmitWindowSize.

**The supervision timeout decide the max tolerance time for receiving packets loss**



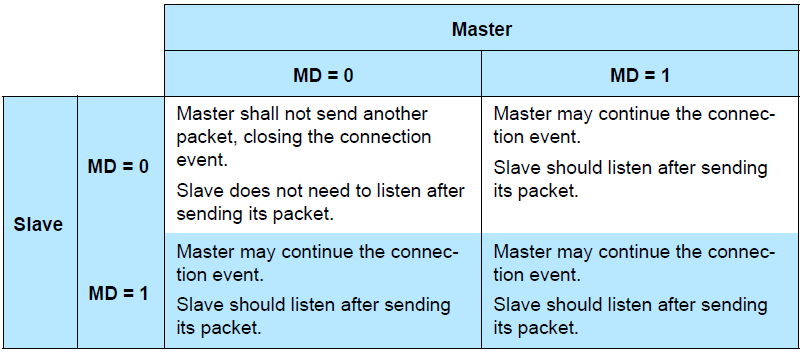


## 2.3 Close connection events

The MD bit of the Header of the Data Physical Channel PDU is used to indicate that the device has more data to send.

If either or both of the devices have set the MD bit, the master may continue the connection event by sending another packet, and the slave should listen after sending its packet. If neither device has set the MD bit in their packets, the packet from the slave closes the connection event.

Failure to receive a packet, or two consecutive packets received with an invalid CRC match within a connection event shall close the event.



## 2.4 channel group index selection

Link Layer shall classify the RF channels in the general-purpose group into used channels (used for transmitting data) and unused channels (not used for transmitting data). This is called the channel map. The minimum number of used channels shall be 2.

### 2.4.1 Channel Selection algorithm #1

Channel Selection Algorithm #1 only supports channel selection for connection events.

Channel Selection Algorithm #1 consists of two stages:

(1) calculation of the unmapped channel index;

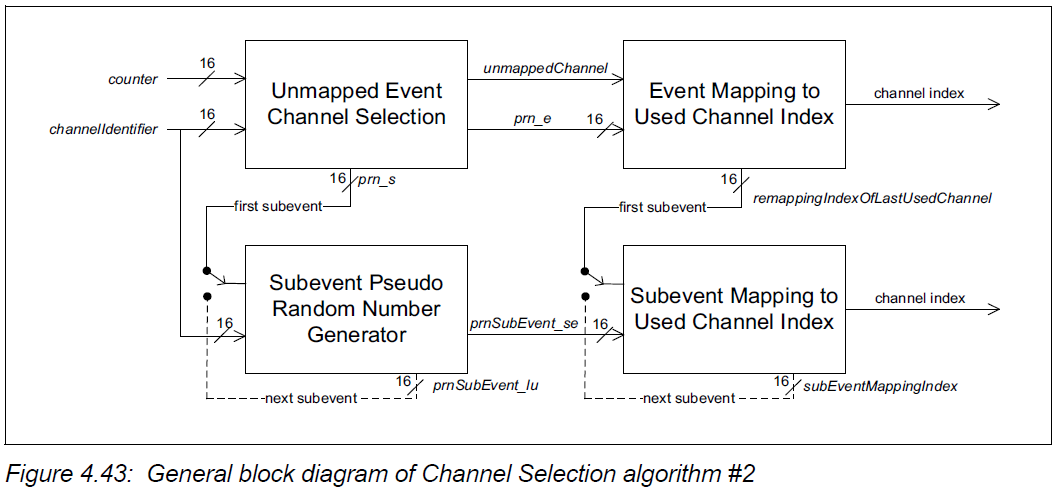
(2) mapping this index to a data channel index from the set of used channels.

unmappedChannel = (lastUnmappedChannel + hopIncrement) mod 37

The lastUnmappedChannel shall be 0 for the first connection event of a connection. When a connection event closes, the lastUnmappedChannel shall be set to the value of the unmappedChannel.

### 2.4.2 Channel Selection algorithm #2

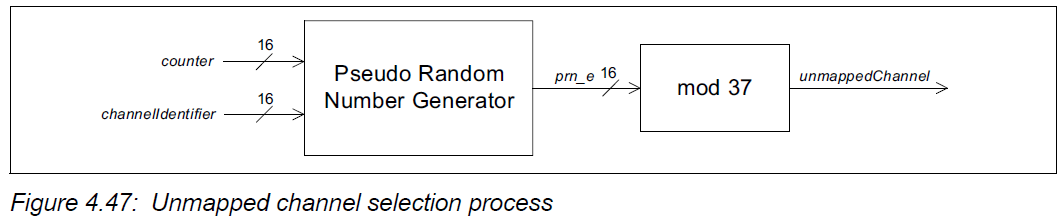
Channel Selection Algorithm #2 supports channel selection for connection events and for periodic advertising events and also supports subevent channel selection.



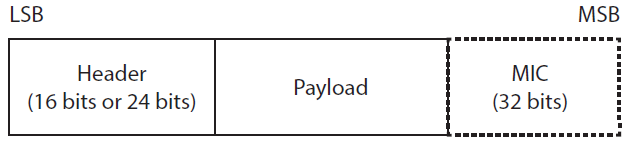
*channelIdentifier* = (Access Address31-16) XOR (Access Address15-0)

counter changes for each event. For ACL connections it is the connection event counter connEventCounter

unsigned pseudo-random numbers prn\_e and prn\_s

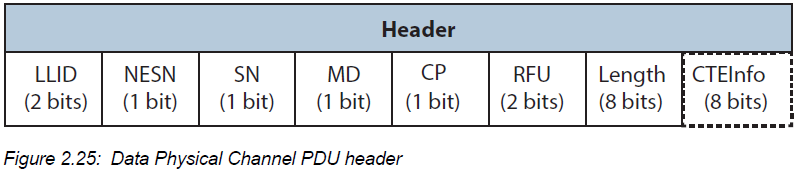


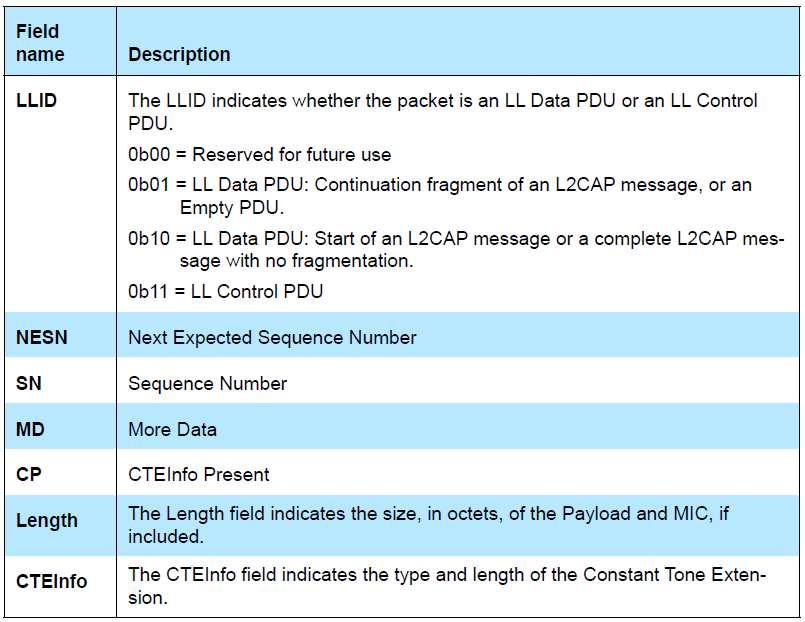
## 2.5 Data Physical Channel PDU

****

variable size payload, and may include a Message Integrity Check (MIC) field.

The MIC field shall not be included in an un-encrypted ACL connection, or in an encrypted ACL connection with a Data Channel PDU with a zero length Payload.





### 2.5.1 LL Data PDU

An LL Data PDU is a Data Channel PDU that is used to send L2CAP data.

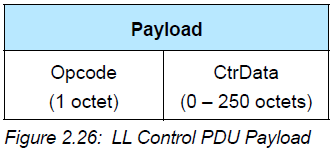
An LL Data PDU with the LLID field in the Header set to 0b01, and the Length field set to 0b00000000, is known as an Empty PDU. The master’s Link Layer may send an Empty PDU to the slave to allow the slave to respond with any Data Physical Channel PDU, including an Empty PDU.

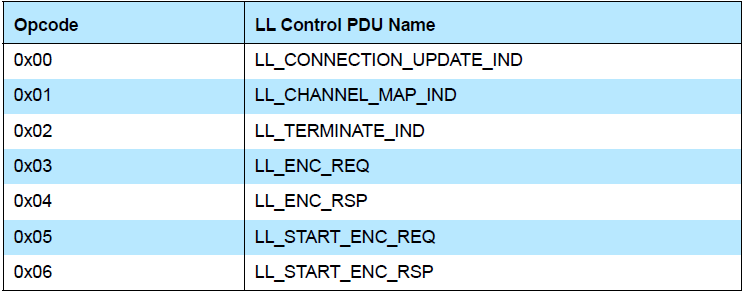
An LL Data PDU with the LLID field in the Header set to 0b10 shall not have the Length field set to 0b00000000.

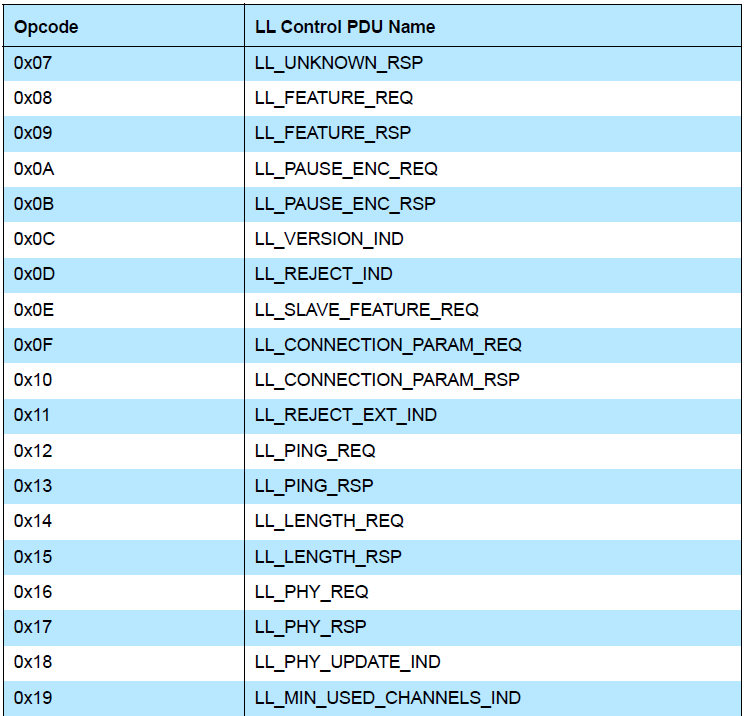
Note: If the Link Layer receives an HCI ACL Data Packet with Data\_Total\_Length equal to 0b00000000 and Packet\_Boundary\_Flag set to 0b00 (i.e., a start fragment), then the Link Layer cannot simply transmit the fragment over the air but, instead, must combine it with one or more of the following continuation fragments to form a PDU with LLID set to 0b10 and nonzero length.

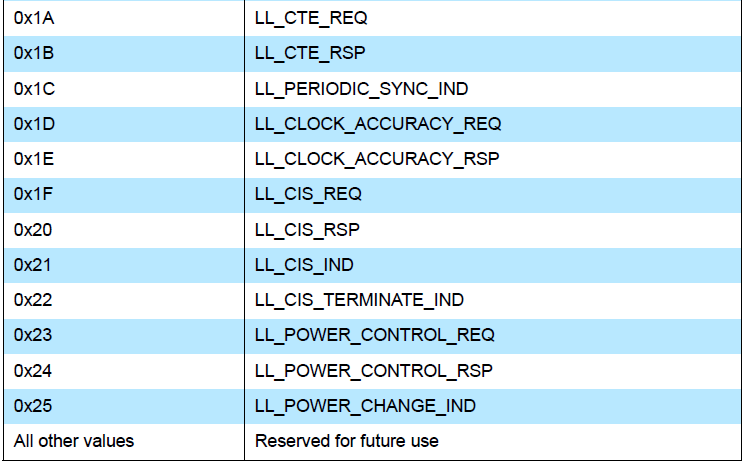
### 2.5.2 LL Control PDU

An LL Control PDU is a Data Physical Channel PDU that is used to control the Link Layer connection.









If an LL Control PDU is received that is not supported or reserved for future use, the Link Layer shall respond with an LL\_UNKNOWN\_RSP PDU.

# 3. ACL control procedures

## 3.1 Connection Update procedure

The Link Layer parameters for a connection (connInterval, connSlaveLatency and connSupervisionTimeout) may be updated after entering the Connection State.

The Link Layer shall notify its Host if any of the three connection parameters have changed. If no connection parameters are changed, the Host would not be notified;

## 3.2 Channel Map Update procedure

The Link Layer parameter for channel map (channelMap) may be updated after entering the Connection State. The master can update the channel map by sending an LL\_CHANNEL\_MAP\_IND PDU. The slave shall not send this PDU. The master Controller can update the channel map without being requested to by the Host.

## 3.3 Encryption procedure

The Link Layer, upon request from the Host, can enable the encryption of packets after entering the Connection state using the encryption start procedure.

Once the connection is encrypted, the Link Layer can change the encryption key by using the encryption pause procedure.

### 3.3.1 Encryption Start procedure

To enable encryption, two parameters must be exchanged, IV and SKD.

To start encryption, the Link Layer of the master shall generate the master’s part of the initialization vector (IVm) and the master’s part of the session key diversifier (SKDm).

IVm shall be a 32 bit random number generated by the Link Layer of the master. SKDm shall be a 64 bit random number generated by the Link Layer of the master.

The Link Layer of the master shall then send an LL\_ENC\_REQ PDU; the Rand and EDIV fields are provided by the Host. After the master receives the LL\_ENC\_RSP PDU in response, only PDUs required by this procedure are expected.

when the Link Layer of the slave receives an LL\_ENC\_REQ PDU it shall generate the slave’s part of the initialization vector (IVs) and the slave’s part of the session key diversifier (SKDs). The slave shall then send an LL\_ENC\_RSP PDU and then notify the Host with the Rand and EDIV fields.

SKD = SKDm || SKDs

IV = IVm || IVs

If the Host does not provide a Long Term Key, either because the event to the Host was masked out or if the Host indicates that a key is not available, the slave shall either send an LL\_REJECT\_IND with the ErrorCode set to PIN or Key Missing (0x06).

If the Host does provide a Long Term Key, the Link Layer of the slave shall calculate sessionKey using the encryption engine with LTK as the key, and SKD as the plain text input. The sessionKey parameter shall be set to the output of the encryption engine.

If, at any time during the encryption start procedure after the slave has received the LL\_ENC\_REQ PDU or the master has received the LL\_ENC\_RSP PDU, the Link Layer of the master or the slave receives an unexpected Data Physical Channel PDU from the peer Link Layer, it shall immediately exit the Connection state, and shall transition to the Standby state. The Host shall be notified that the link has been disconnected with the error code Connection Terminated Due to MIC Failure (0x3D).

### 3.3.2 Encryption Pause procedure

To enable a new encryption key to be used without disconnecting the link, encryption must be disabled and then enabled again.

## 3.4 Feature Exchange procedure

The Link Layer parameter for the current supported feature set (FeatureSet) may be exchanged after entering the Connection state. Both the master and slave can initiate this procedure.

## 3.5 Version Exchange procedure

The Link Layer parameters for version information (companyID, subVerNum, linkLayerVer) may be exchanged after entering the Connection State.

## 3.6 ACL Termination procedure

occurs when the Host requests the Link Layer to terminate the connection.

Either the Link Layer of the master or slave can initiate this procedure by sending an LL\_TERMINATE\_IND PDU.

## 3.7 Connection Parameters Request procedure

The master or slave may initiate a Connection Parameters Request procedure to request the remote device to have the Link Layer parameters for the connection (connInterval, connSlaveLatency and connSupervisionTimeout) updated any time after entering the Connection State.

The Connection Parameters Request procedure is initiated by issuing an

LL\_CONNECTION\_PARAM\_REQ PDU. The procedure can be initiated as a result of a Host initiated connection update procedure.

## 3.8 LE Ping procedure

The LE Ping procedure, when supported, can be used at the Link Layer to verify presence of the remote Link Layer. The procedure can also be used to verify message integrity on the LE ACL logical transport by forcing the remote device to send an LE ACL packet that contains a valid MIC.

Either the master or the slave Link Layer may initiate this procedure at any time after entering the Connection state by sending an LL\_PING\_REQ PDU. The responding Link Layer responds with the LL\_PING\_RSP PDU.

The Link Layer supporting this feature shall send an LL\_PING\_REQ PDU when the remote device has not sent a packet containing a payload protected by a MIC within the authenticated payload timeout set by the Host.

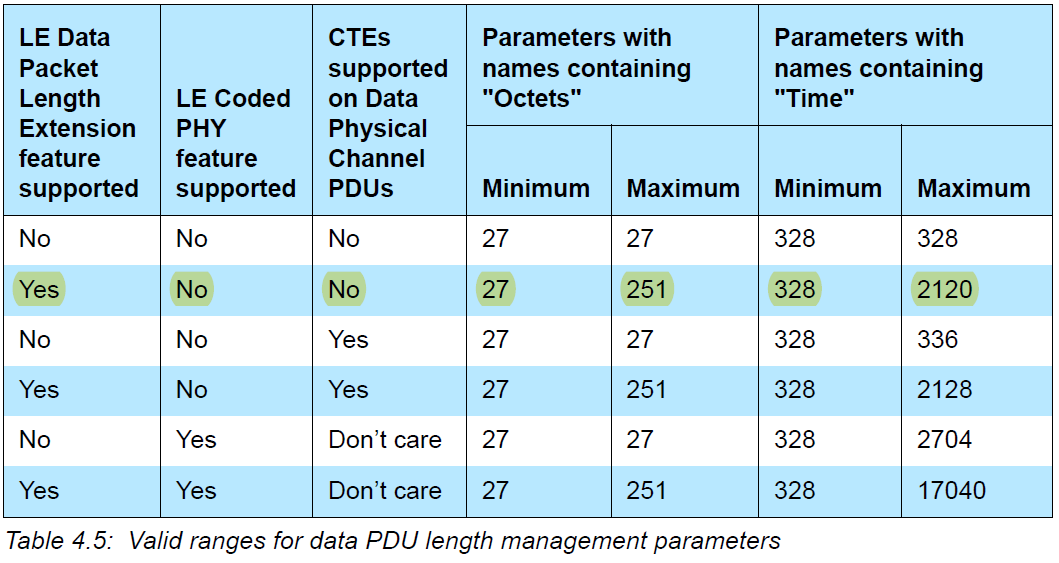
## 3.9 Data Length Update procedure

A Controller uses the Data Length Update Procedure to transmit the latest values of the current maximum Receive LL Data PDU Payload length and PDU Time (connMaxRxOctets and connMaxRxTime) and the current maximum Transmit LL Data PDU Payload length and PDU Time (connMaxTxOctets and connMaxTxTime) to the peer device.

Both the master and slave can initiate this procedure at any time after entering the Connection State by sending an LL\_LENGTH\_REQ PDU.

Upon receiving an LL\_LENGTH\_REQ PDU, the Link Layer shall respond with an LL\_LENGTH\_RSP PDU containing its own connMaxTxOctets, connMaxRxOctets, connMaxTxTime, and connMaxRxTime values for the connection.

If the peer device does not support the LE Coded PHY feature, then the MaxRxTime and MaxTxTime fields in the LL\_LENGTH\_REQ and LL\_LENGTH\_RSP PDUs shall be set to a value less than or equal to 2128 microseconds.



The procedure is completed when the initiating Controller receives an LL\_LENGTH\_RSP PDU.

## 3.10 PHY Update procedure

The PHY Update procedure, when supported, is used to change the transmit or receive PHYs, or both, of an ACL connection; it does not affect the transmit or receive PHY of any associated Connected Isochronous Streams.

When this procedure is initiated by the master, it sends an LL\_PHY\_REQ PDU. The slave responds with an LL\_PHY\_RSP PDU. The master then responds to this with an LL\_PHY\_UPDATE\_IND PDU. When this procedure is initiated by the slave, it sends an LL\_PHY\_REQ PDU. The master responds with an LL\_PHY\_UPDATE\_IND PDU.

The TX\_PHYS and RX\_PHYS fields of the LL\_PHY\_REQ and LL\_PHY\_RSP PDUs shall be used to indicate the PHYs that the sending Link Layer prefers to use. The M\_TO\_S\_PHY and S\_TO\_M\_PHY fields of the LL\_PHY\_UPDATE\_IND PDU shall indicate the PHYs that shall be used after the instant.

## 3.11 Minimum Number Of Used Channels procedure

A Controller uses the Minimum Number Of Used Channels Procedure to request that the peer device uses a minimum number of channels on a given PHY.

The slave can initiate this procedure at any time after entering the Connection State by sending an LL\_MIN\_USED\_CHANNELS\_IND PDU. The Master shall not send this PDU.

If the Link Layer receives an LL\_MIN\_USED\_CHANNELS\_IND PDU, it should ensure that the

connection uses at least the number of channels given in the MinUsedChannels field of the PDU.

## 3.12 Constant Tone Extension Request procedure

## 3.13 Periodic Advertising Sync Transfer procedure

## 3.14 Sleep Clock Accuracy Update procedure

## 3.15 Connected Isochronous Stream Creation procedure

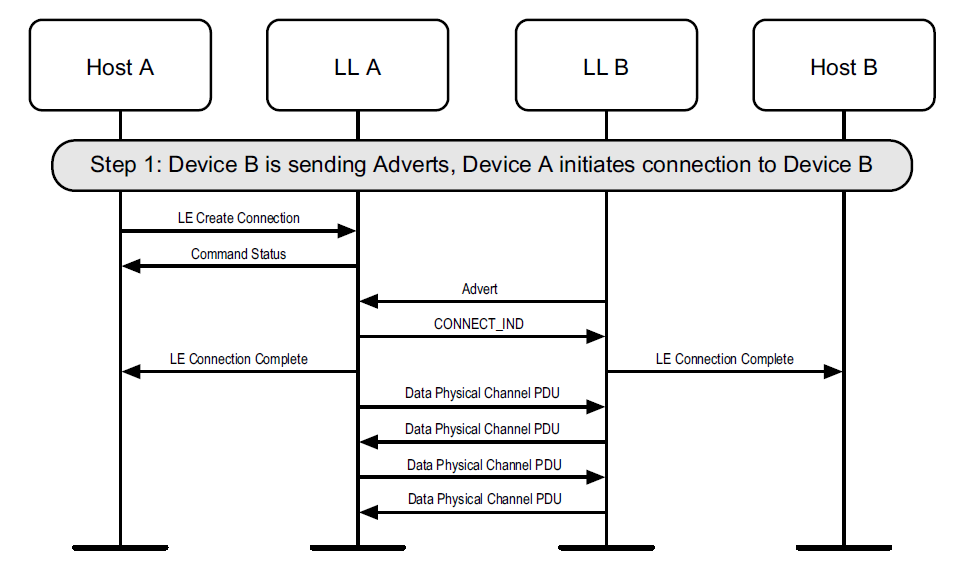
## 3.16 Connected Isochronous Stream Termination procedure

## 3.17 Power Control Request procedure

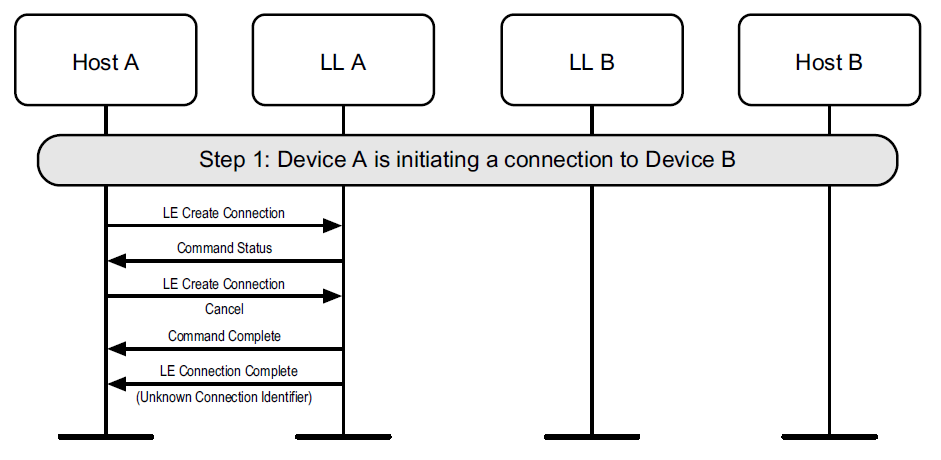
## 3.18 Power Change Indication procedure

# 4. Message sequence chart

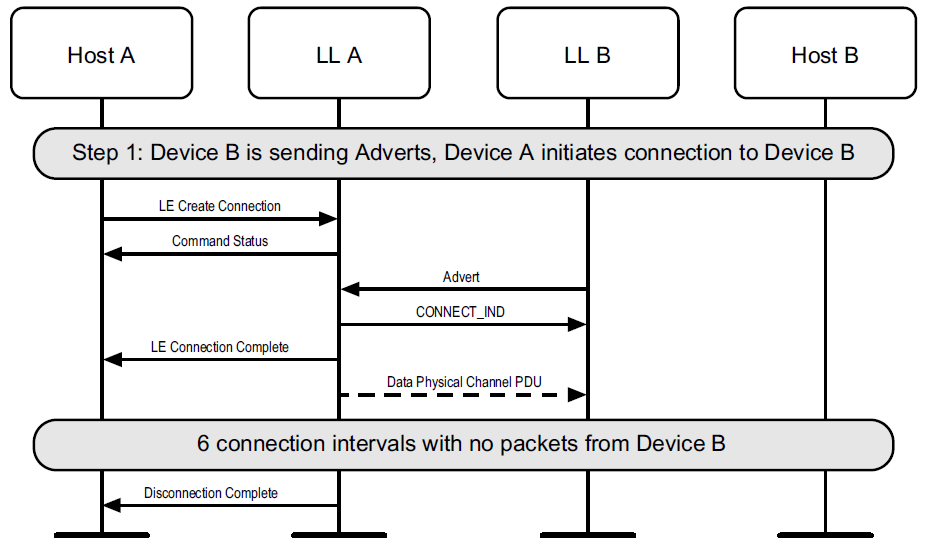
## 4.1 Initiate a connection



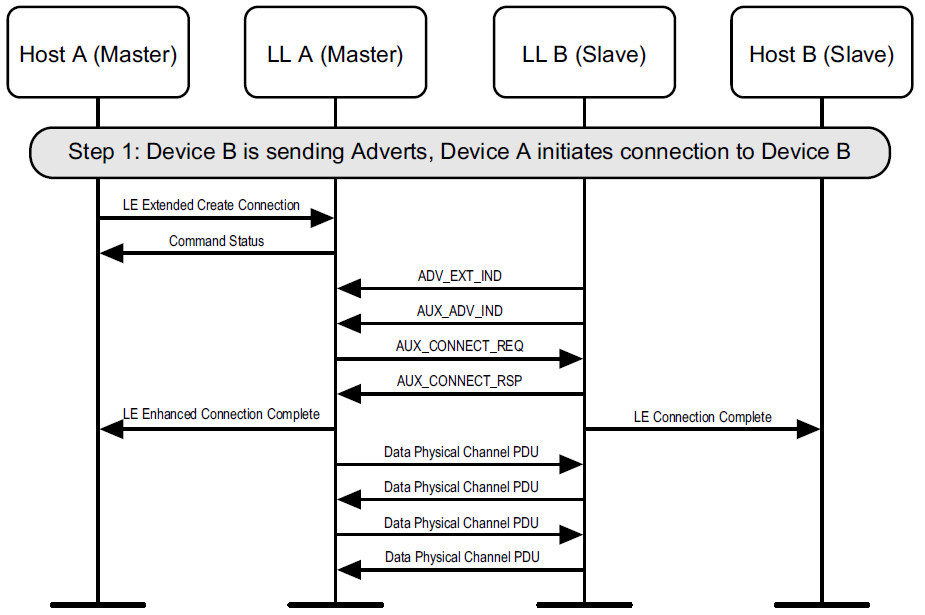
## 4.2 Canceling an initiation



## 4.3 Initiating a connection that fails to establish

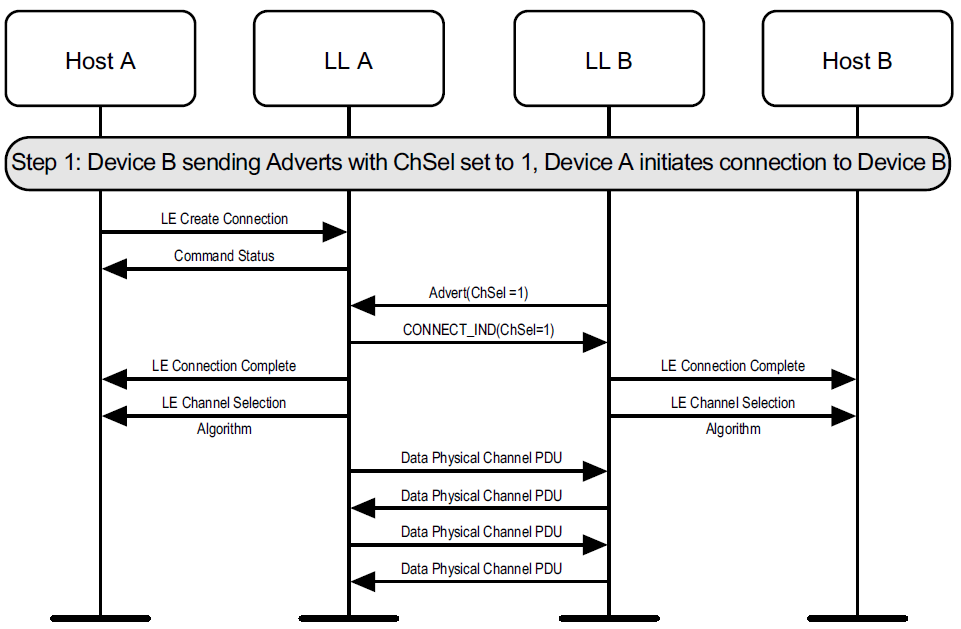


## 4.4 Initiating a connection on the secondary advertising physical channel

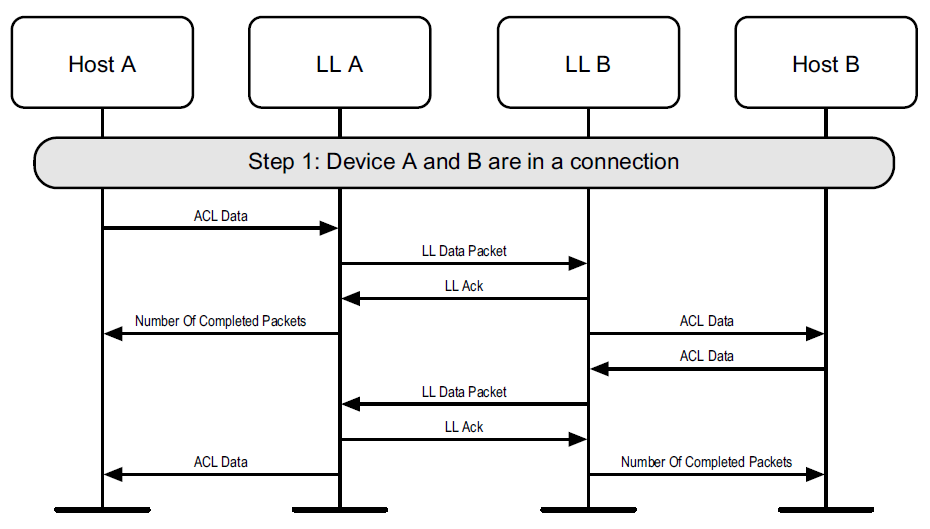


## 4.5 Initiating a Channel Selection algorithm #2 connection

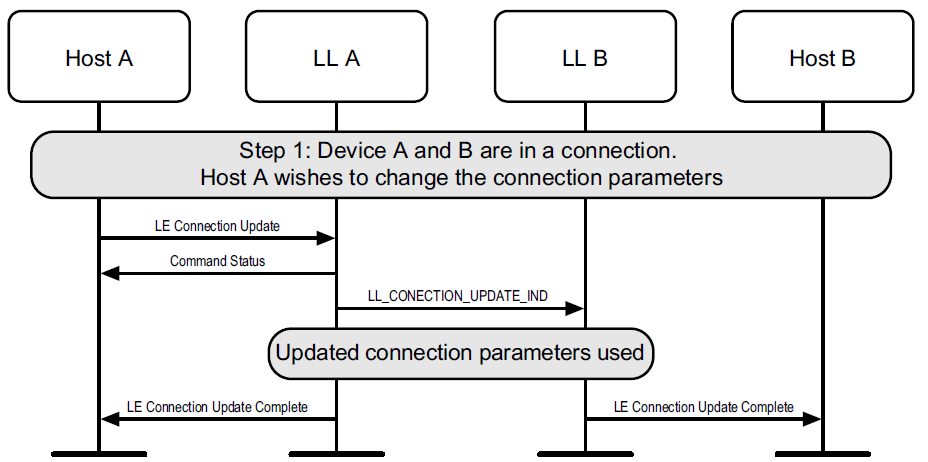
Where a device supports the Channel Selection Algorithm #2 feature, it can initiate a connection which will use Channel Selection Algorithm #2 to an advertiser who has the ChSel field of the advertising physical channel PDU set to 1.



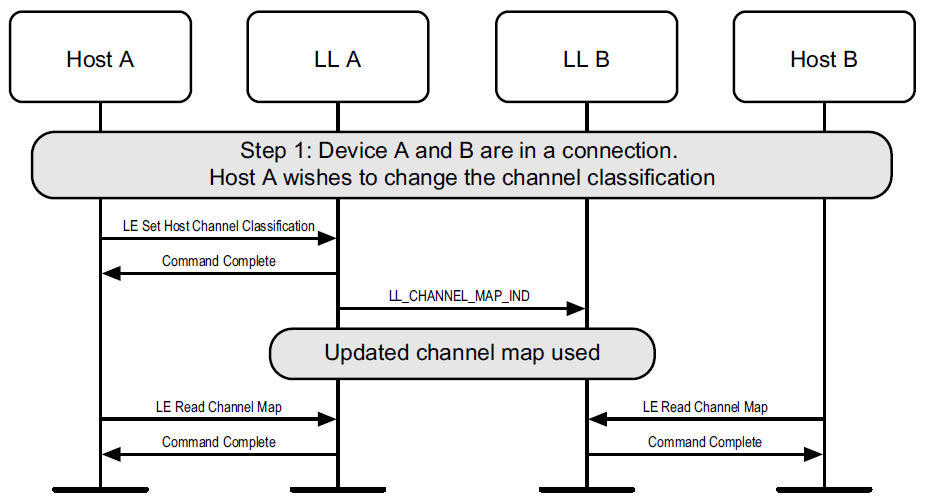
## 4.6 Sending data



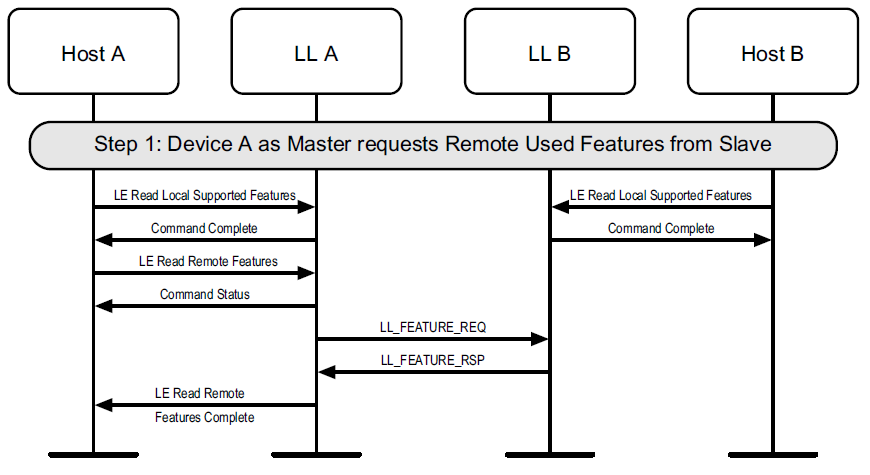
## 4.7 Connection update



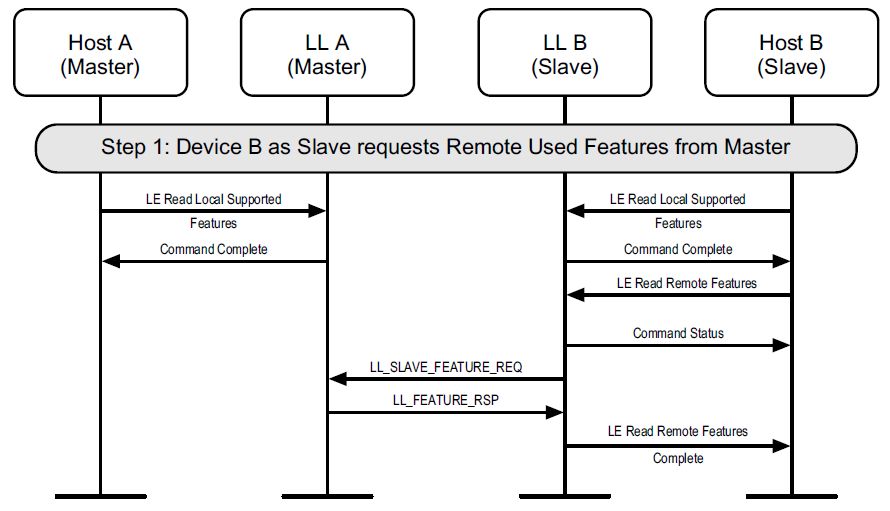
## 4.8 Channel map update



## 4.9 Features exchange

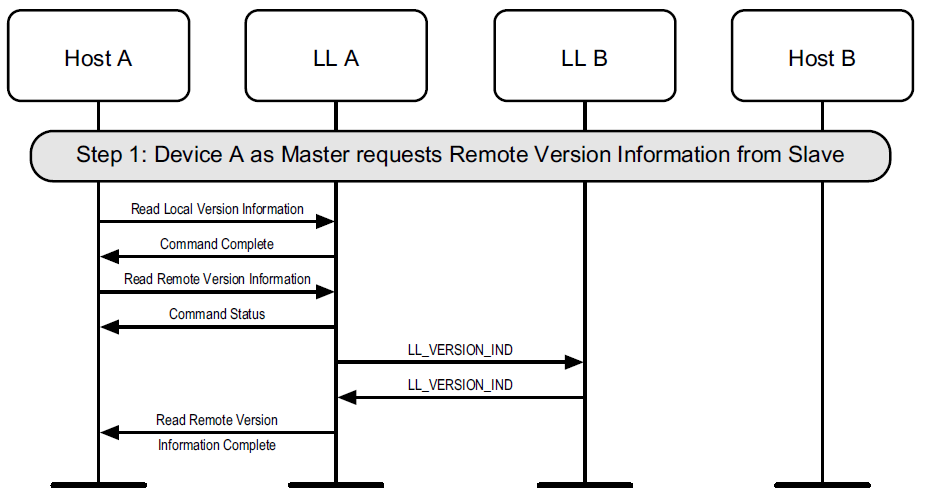


*Master-initiated features exchange*

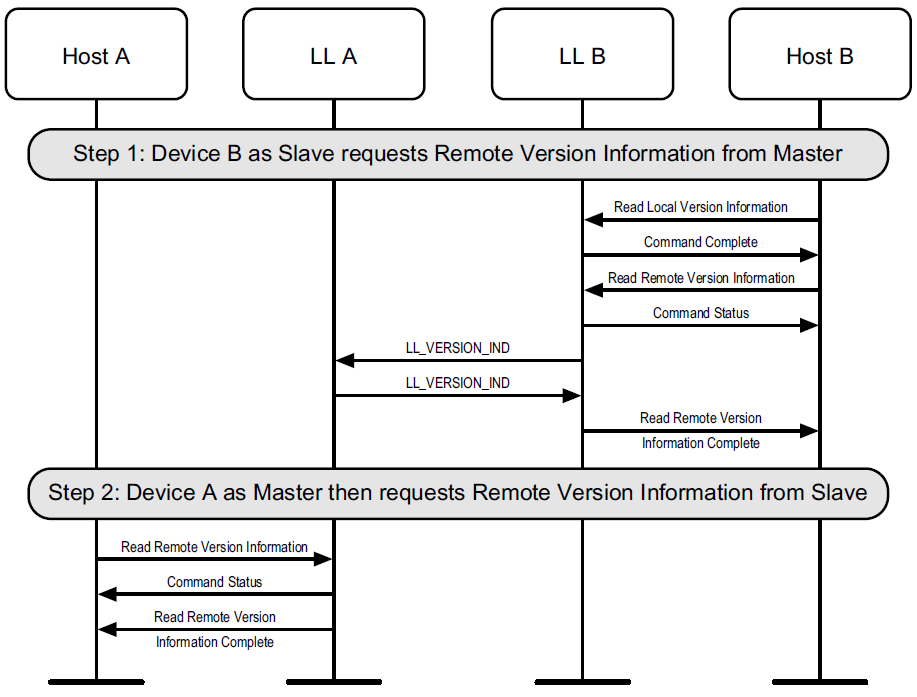


*Slave-initiated features exchange*

## 4.10 Version exchange

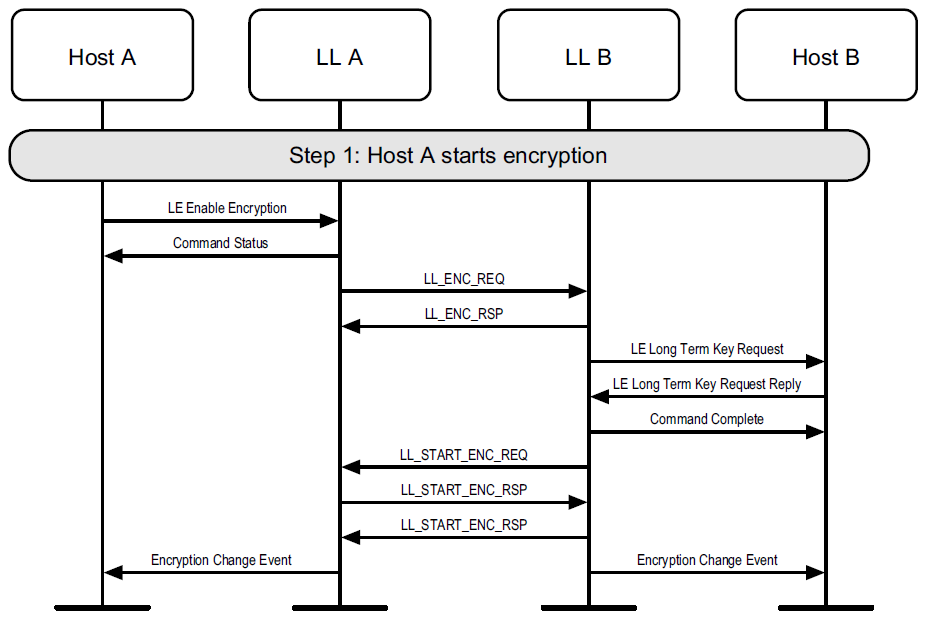


*Version exchange from master*

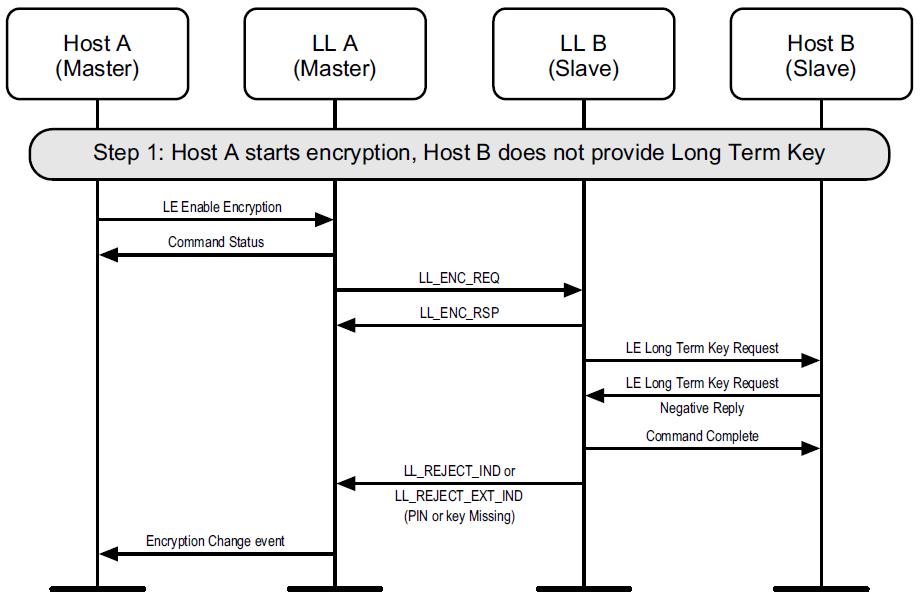


*Version exchange from slave*

## 4.11 Start encryption

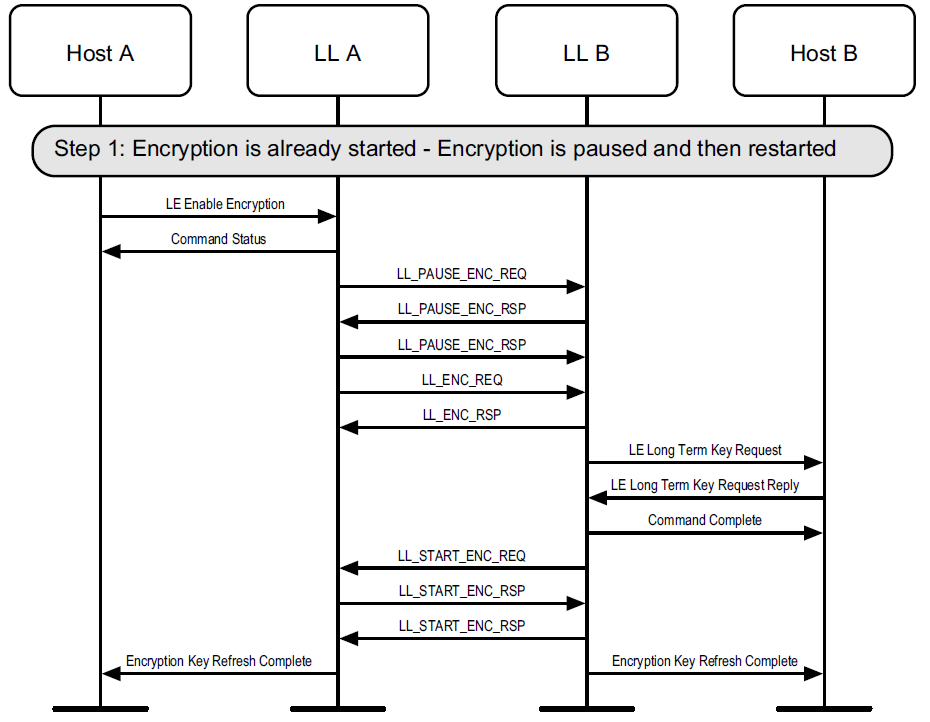


## 4.12 Start encryption without long-term key



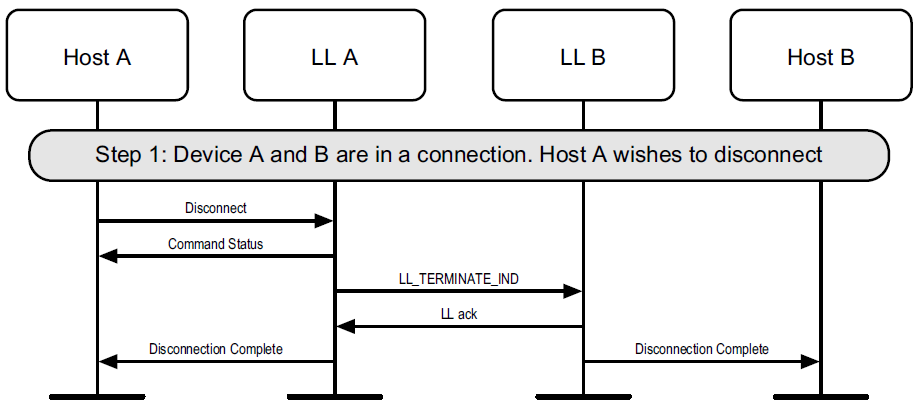
## 4.13 Restart encryption

If encryption has already been started on a connection, it may be restarted by the master. This may be required to use a stronger encryption as negotiated by the Security Manager Protocol.

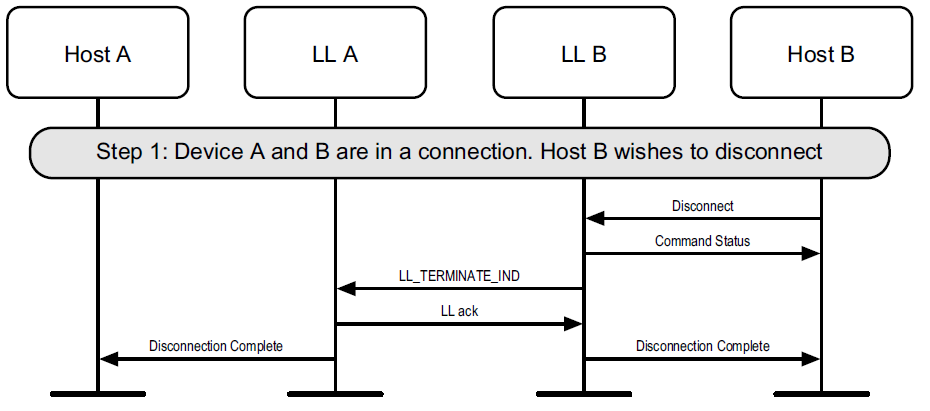


## 4.14 Disconnect

Once a connection has no need to be kept active, the Host can disconnect it.

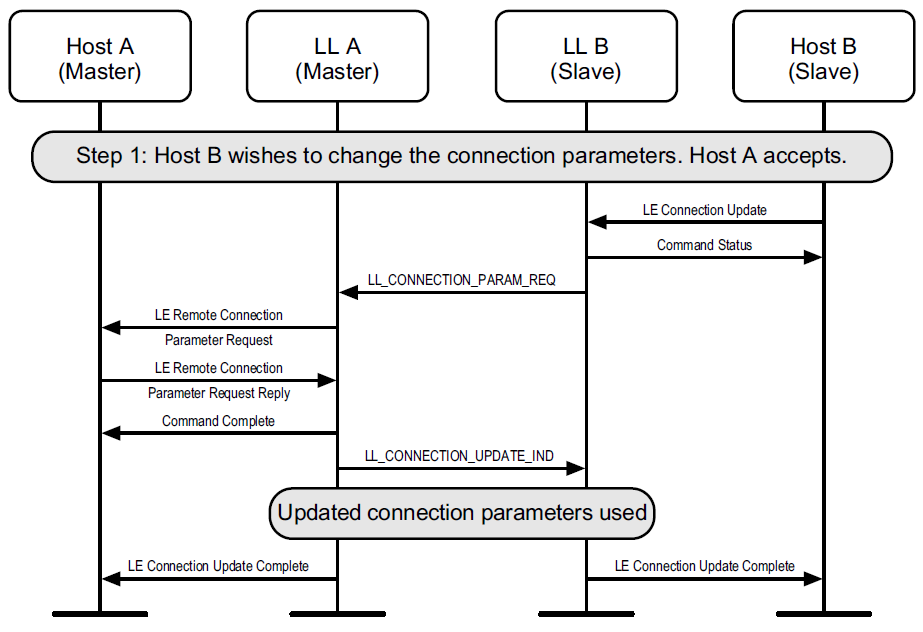


*Disconnect from master*

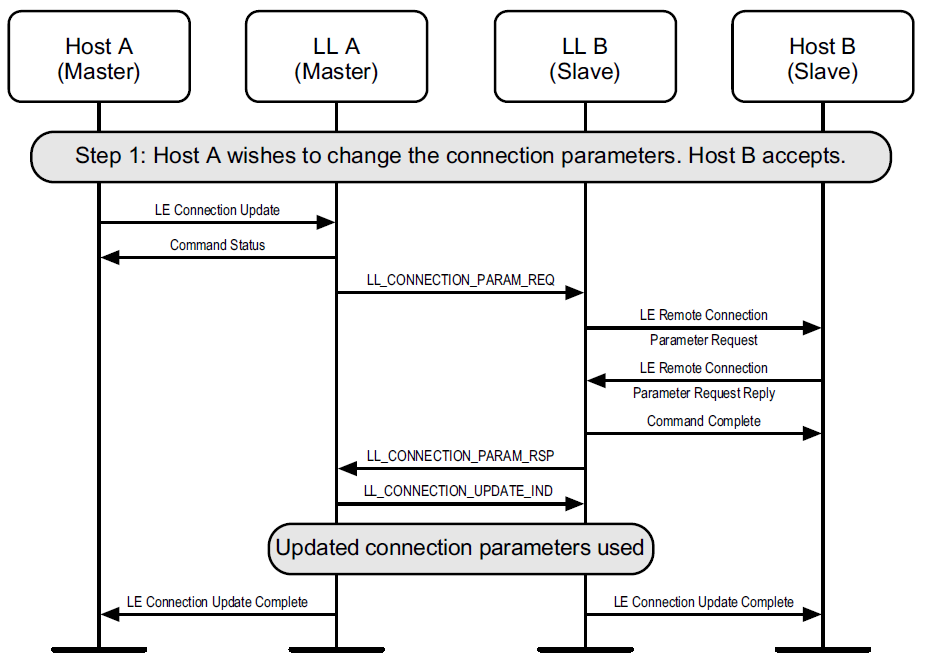


*Disconnect from slave*

## 4.15 Connection Parameters Request



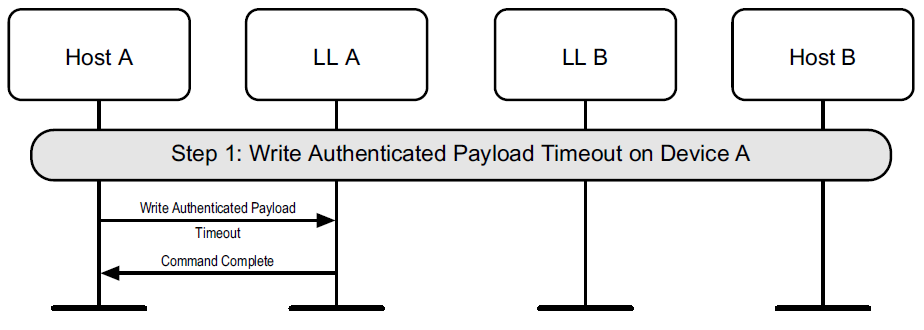
*Slave-initiated Connection Parameters Request procedure*

**

*Master-initiated Connection Parameters Request procedure*

## 4.16 Ping

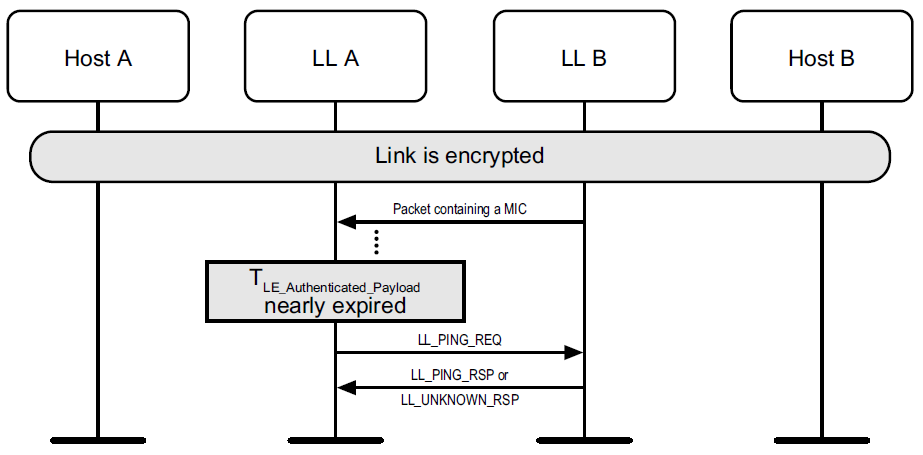
A Host may use the HCI\_Write\_Authenticated\_Payload\_Timeout command to change the maximum interval between packets containing a valid MIC that the Link Layer will enforce when encryption is used.



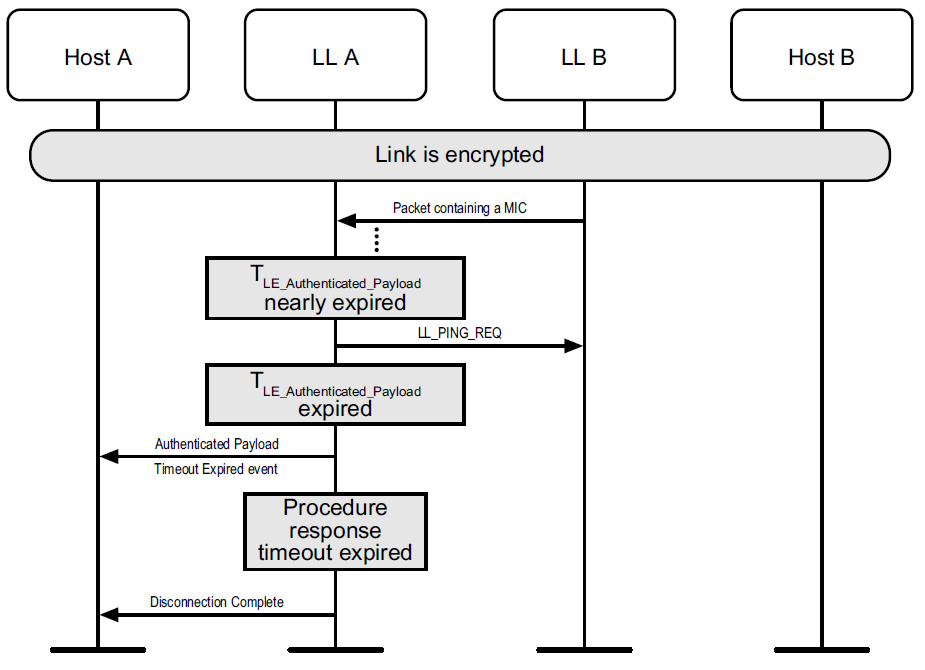
*Set LE authenticated payload timeout*

Either Link Layer can authenticate the remote device using the LE Ping procedure

LL A may be a master or a slave.

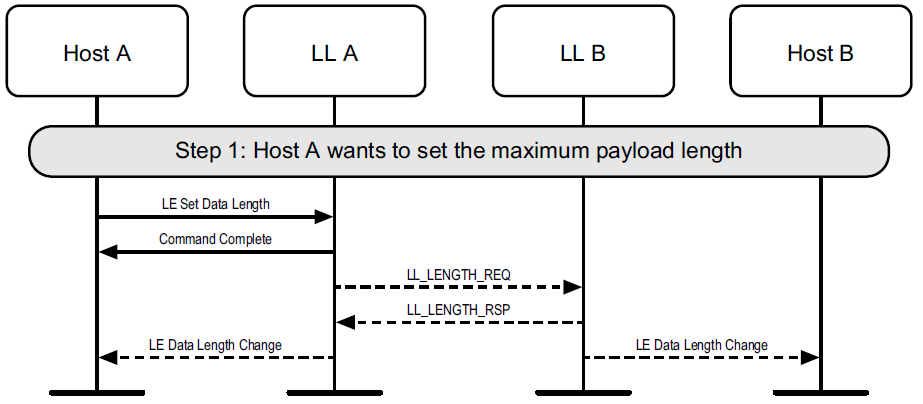


*Successful LE Ping*

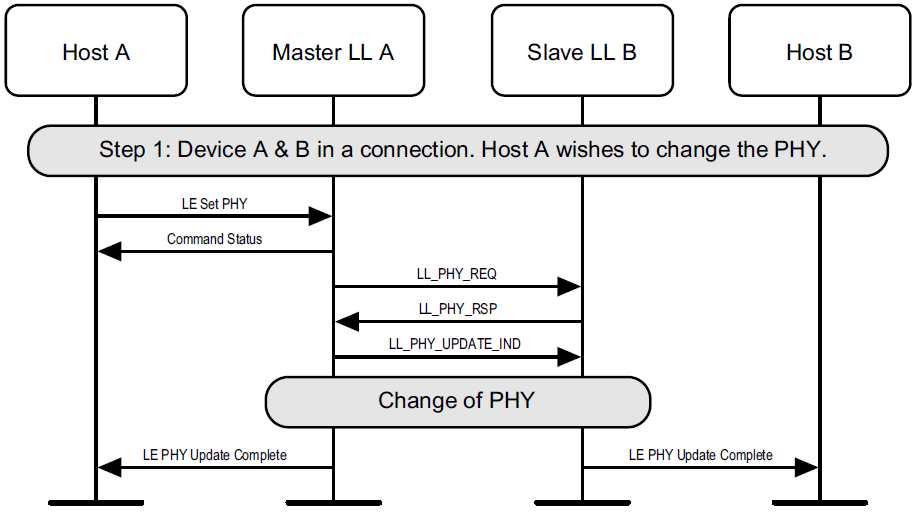


*Unsuccessful LE Ping*

## 4.17 Data length update

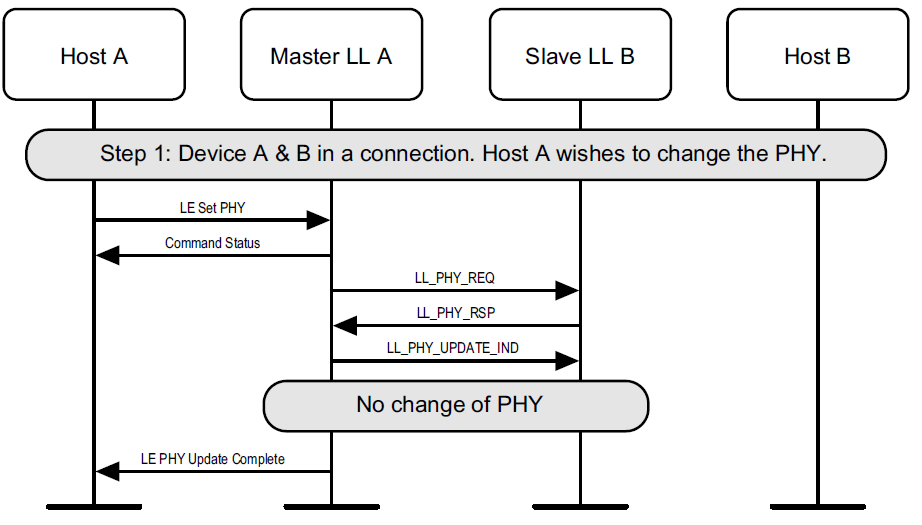


## 4.18 PHY Update

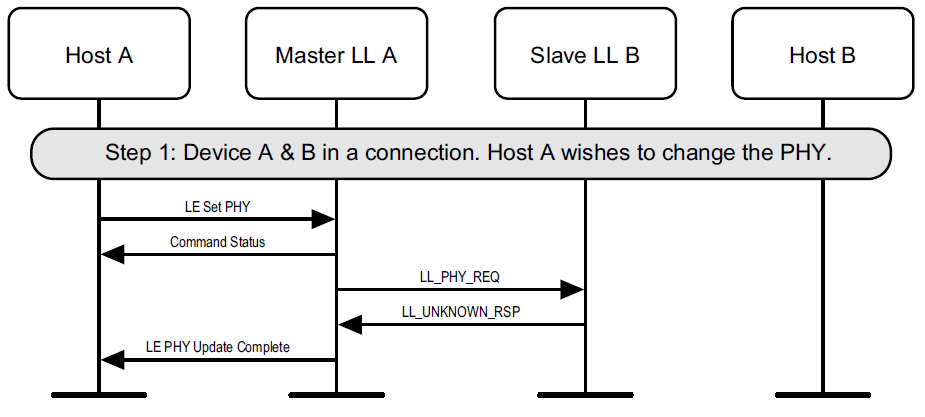


*Master-initiated PHY Update procedure – master requests a change of PHY, PHY*

*changed in at least one direction*



*Master-initiated PHY Update procedure – PHY not changed (either because slave doesn't specify PHYs that the master prefers, or because the master concludes that the current PHYs are still best)*

**

*Master-initiated PHY Update procedure – master requests a change of PHY, slave*

*does not support the feature*

## 4.19 Minimum number of used channels request

