Abstract - HCL Bluetooth Protocol Stack

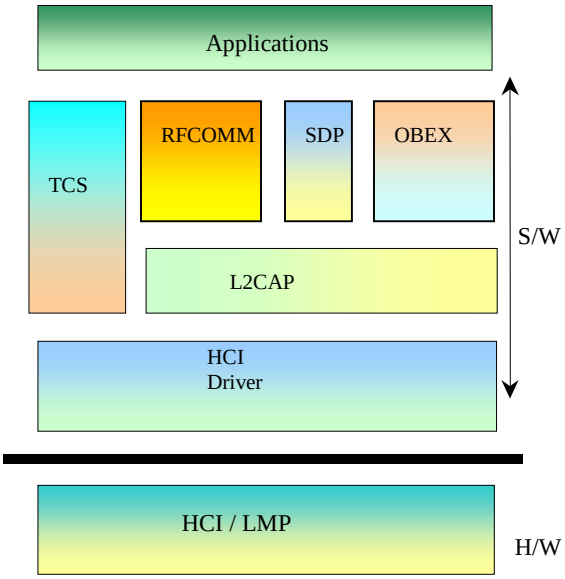
Tushar Vrind
Member Technical Staff, Microprocessor Based Software Group
HCL Technologies Limited India

1. Introduction

HCL Bluetooth Protocol Stack (HBPS) is a high performance, low footprint, and highly portable solution for the embedded market. It supports the Windows CE and VxWorks Operating Systems and the UART, USB, RS-232 and PCMCIA interfaces. It is written in ANSI C and uses an OS independent architecture to provide maximum portability across various operating systems along with very less time-to-market.

1.1 Main Product Features

- Compliant to Bluetooth specifications 1.0B and 1.1
- Comprises SDP, RFCOMM, TCS, L2CAP, HCI Driver layers.
- OS independent architecture – easily customizable across various platforms. First version is available for Windows CE based CEPC (x86) platform.
- High modularity – selected modules can be picked up to have a customized solution according to requirements.
- Very small footprint - suitable for embedded devices.
- Written in ANSI C for independence of development environment (compiler, debugger, linker etc.)
- Supports GAP, Serial and Dial up Networking profiles.
- Includes a baseband Simulator for testing and debugging Bluetooth Applications without using the actual hardware.
- Supports the PCMCIA, USB, UART and RS-232 interfaces.
- Easy to install and customize.



1.2 Demo Configurations

- Wireless Chat

This application establishes a Bluetooth link between two PCs using HBPS and allows the users to chat or send files to each other over the link. It can use any of the UART, USB, PCMCIA, RS232 or Simulator interfaces.

- Bluetooth Modem

This application demonstrates a Dial Up Networking connection using the DUN profile and allows the users to browse the Internet wirelessly.

2. Description and design of L2CAP

The Logical Link and Control Adaptation Protocol layer resides over the HCI Driver and provides services like Protocol Multiplexing, Segmentation and Reassembly, Group Management and Quality of Service.

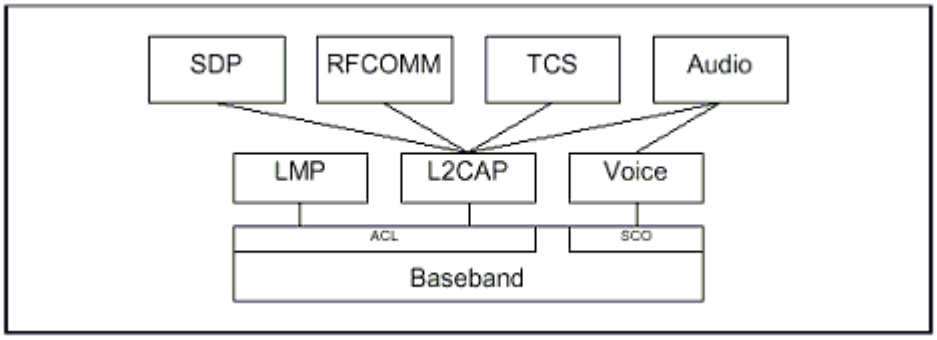


Figure 1. L2CAP in Bluetooth Protocol Architecture

L2CAP resides over Baseband Protocol in data link layer. It is only defined for Asynchronous Connection Less (ACL) Link and there is no support for Synchronous Connection Oriented (SCO) Link.

The use of AUX1 packet is prohibited, as it does not support data integrity (CRC) check and L2CAP depends on integrity checks in the Baseband.

The L_CH field in the packet header identifies the L2CAP packet. The 2-bit logical channel (L_CH) field distinguishes L2CAP packets from LM packets.

L_CH code	Logical Channel	Information
00	RESERVED	Reserved for future use
01	L2CAP	Continuation of L2CAP packet
10	L2CAP	Start of L2CAP packet
11	LMP	Link Manager Protocol

2.1 Protocol Architecture

- Connection Oriented
    1. Channel Identifier is used to label each connection.
    2. Channel is assumed to be full duplex.
    3. Qos Flow Specification is assigned to each direction, i.e. inbound as well as outgoing.
- Datagram – based, No Streams
    1. Packet boundaries are preserved.
- Relies on Baseband for data security and delivery in order.

The functional requirements for L2CAP include:

1. **Protocol Multiplexing**

   The Baseband protocol does not support any “type” field to identify the higher protocol being multiplexed over it. Thus to support protocol multiplexing L2CAP must be able to distinguish between upper layer protocols such as the SDP, RFCOMM, TCS.

2. **Segmentation And Reassembly (SAR)**

   L2CAP permits higher-level protocols and applications to transmit and receive L2CAP data packet upto 64 kilobytes in length. The data packets defined by Baseband Protocol are limited in size. Exporting a maximum transmission unit

(MTU) associated with the largest Baseband payload (341 bytes for DH5 packet) limits the efficient use of the bandwidth for the higher layer protocols. Large L2CAP packets must be segmented into multiple smaller Baseband packets prior to their transmission over the air. Similarly, multiple received Baseband packets may be reassembled into single larger L2CAP packet following a simple integrity check.

3. **Group Management**

   Bluetooth supports the concept of piconet, a group of devices synchronously hopping together on the same clock. Many protocols have the concept of group of addresses. The L2CAP group abstraction permits implementation to efficiently map protocol groups onto piconet. Thus the upper layers remain unexposed to the functionality of LM and Baseband.

4. **Quality of Service**

   The L2CAP connection establishment process allows the exchange of information regarding the quality of service (QoS) expected between two Bluetooth units. Each L2CAP implementation must monitor the resources used by the protocol and ensure that the QoS contracts are honored.

L2CAP is packet based but follows communication models based on channels. A channel represents a data flow between L2CAP entities in remote devices. Channels may be connection-oriented or connection-less.
CIDs are local names representing a logical channel endpoint on the device. CIDs can be dynamically allocated (except for reserved ones) in a manner best suited for a implementation, with the provision that the same CID is not used as a local L2CAP channel end point for simultaneous (multiple) L2CAP channels between local device and some remote device. CID assignment is relative to a particular device and a device can assign CIDs independently from other devices. Even if the same CID value has been assigned to (remote) channel endpoints by several remote devices connected to a single local device, the local device can still uniquely associate each remote CID with a different device.
The connection oriented data channels represent a connection between two devices, where a CID identifies each endpoint of the channel. The connectionless channels restrict data flow to a single direction. These channels are used to support a channel group where the CID on the source represents one or more devices.

CID	**Description**
0x0000	Null Identifier
0x0001	Signaling Channel
0x0002	Connectionless reception channel
0x0003 – 0x003F	Reserved
0x0040 – 0xFFFF	Dynamically allocated

Channel Type	**Local CID**	**Remote CID**
Connection-oriented	Dynamically allocated	Dynamically allocated
Connectionless data	Dynamically allocated	0x0002 (fixed)
Signaling	0x0001 (fixed)	0x0001 (fixed)

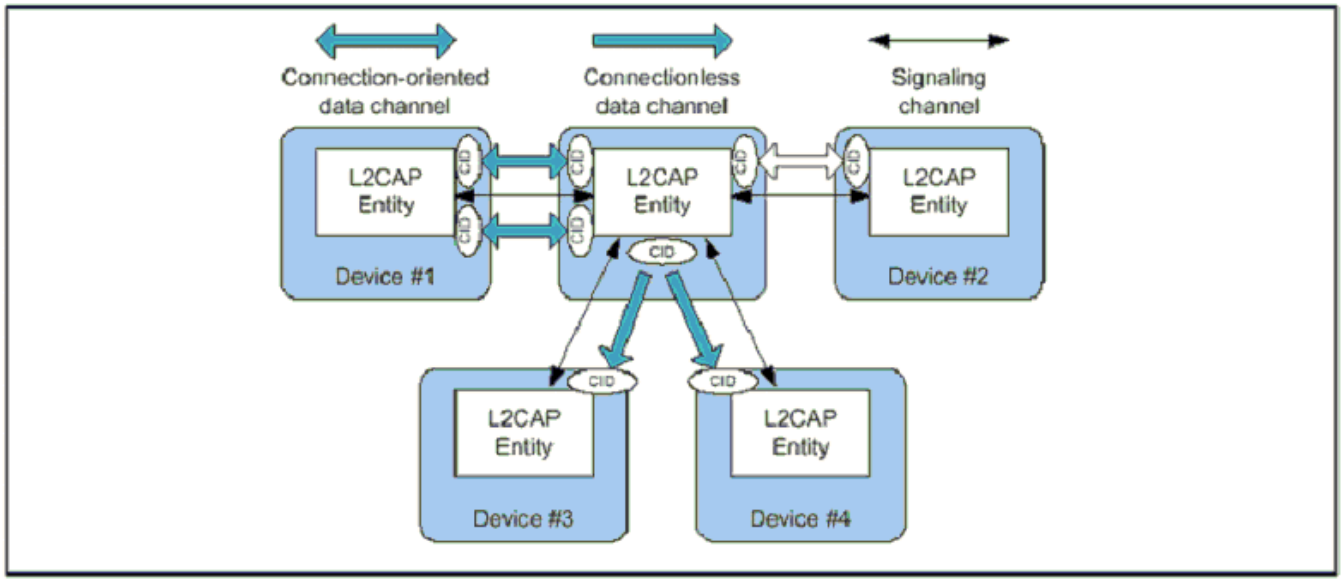


Figure 2. *Channels between devices*

2.1.1 Connection-Oriented Channel State Machine

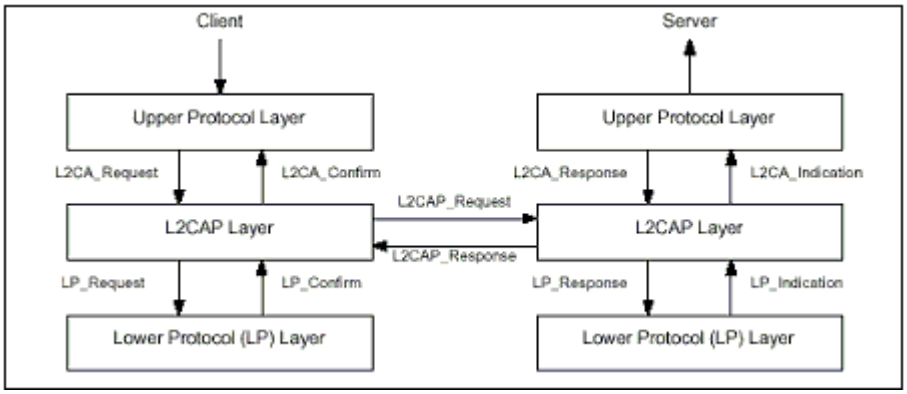


Figure 3. L2cap layer Interactions

The connection-oriented channel state machine is defines in terms of states, events causing state transitions, and the actions to be performed in response to events. This state machine is only pertinent to bi-directional CIDs and is not representative of the signaling channel or the uni-directional channel.

2.1.1.1 Channel Operational States:

- CLOSED

  In this state, there is no channel associated with this CID. This is the only state when a link level connection (Baseband) may not exist. Link disconnection forces all other states into the CLOSED State.

- W4_L2CAP_CONNECT_RSP

  In this state, the CID represents a local end-point and an L2CAP_ConnectReq message has been sent referencing this endpoint and it is now waiting for the corresponding L2CAP_ConnectRsp message.

- W4_L2CA_CONNECT_RSP

  In this state, the remote end-point exists and an L2CAP_ConnectReq has been received by the local L2CAP entity. An L2CA_ConnectInd has been sent to the upper layer and the part of the local L2CAP entity processing the received L2CAP_ConnectReq waits for the corresponding response. The response may require a security check to be performed.

- CONFIG

In this state, the connection has been established but both sides are still negotiating the channel parameters. The Configuration State may also be entered when the channel parameters are being renegotiated. Prior to entering the CONFIG State, all outgoing data traffic should be suspended since the traffic parameters of the data traffic are to be renegotiated. Incoming data traffic must be accepted until the remote channel endpoint has entered the CONFIG State.

In the CONFIG State, both sides must issue L2CAP_ConfigReq messages – if only defaults are being used, a null message should be sent. If a large amount of parameters need to be negotiated, multiple messages may be sent to avoid any MTU limitations and negotiate incrementally. Moving from the CONFIG State to the OPEN State requires both sides to be ready. An L2CAP entity is ready when it has received a positive response to its final request and it has positively responded to the final request from the remote device.

- OPEN

  In this state, the connection has been established and configured, and data flow may proceed.

- W4_L2CAP_DISCONNECT_RSP

  In this state, the connection is shutting down and an L2CAP_DisconnectReq message has been sent. This state is now waiting for the corresponding response.

- W4_L2CA_DISCONNECT_RSP

  In this state, the connection on the remote endpoint is shutting down and an L2CAP_DisconnectReq message has been received. An L2CA_DisconnectInd has been sent to the upper layer to notify the owner of the CID that the remote endpoint is being closed. This state is now waiting for the corresponding response from the upper layer before responding to the remote endpoint.

2.1.1.2 Events

Events are all incoming messages to the L2CA layer along with timeouts. These are partitioned into the following five categories

- Indications and Confirms from Lower Layers
- Requests and Responses from Higher Layers
- Data from Peers
- Signal Requests and Responses from peers
- Timer Expiration

2.1.1.3 Actions

Actions are all outgoing messages. These are partitioned into the following five categories.

- Confirms and Indications to higher layers
- Request and Responses to lower layers
- Requests and Responses to peers
- Data transmission to peers
- Setting timers.

2.1.2 Connection Less Data Channel

In addition to connection-oriented channels, L2CAP also exports the concept of a group-oriented channel. Data sent to the ’group’ channel is sent to all members of the group in a best-effort manner. Groups have no quality of service associated with them. Group channels are unreliable; L2CAP makes no guarantee that data sent to the group successfully reaches all members of the group. If reliable group transmission is required, it must be implemented at a higher layer.

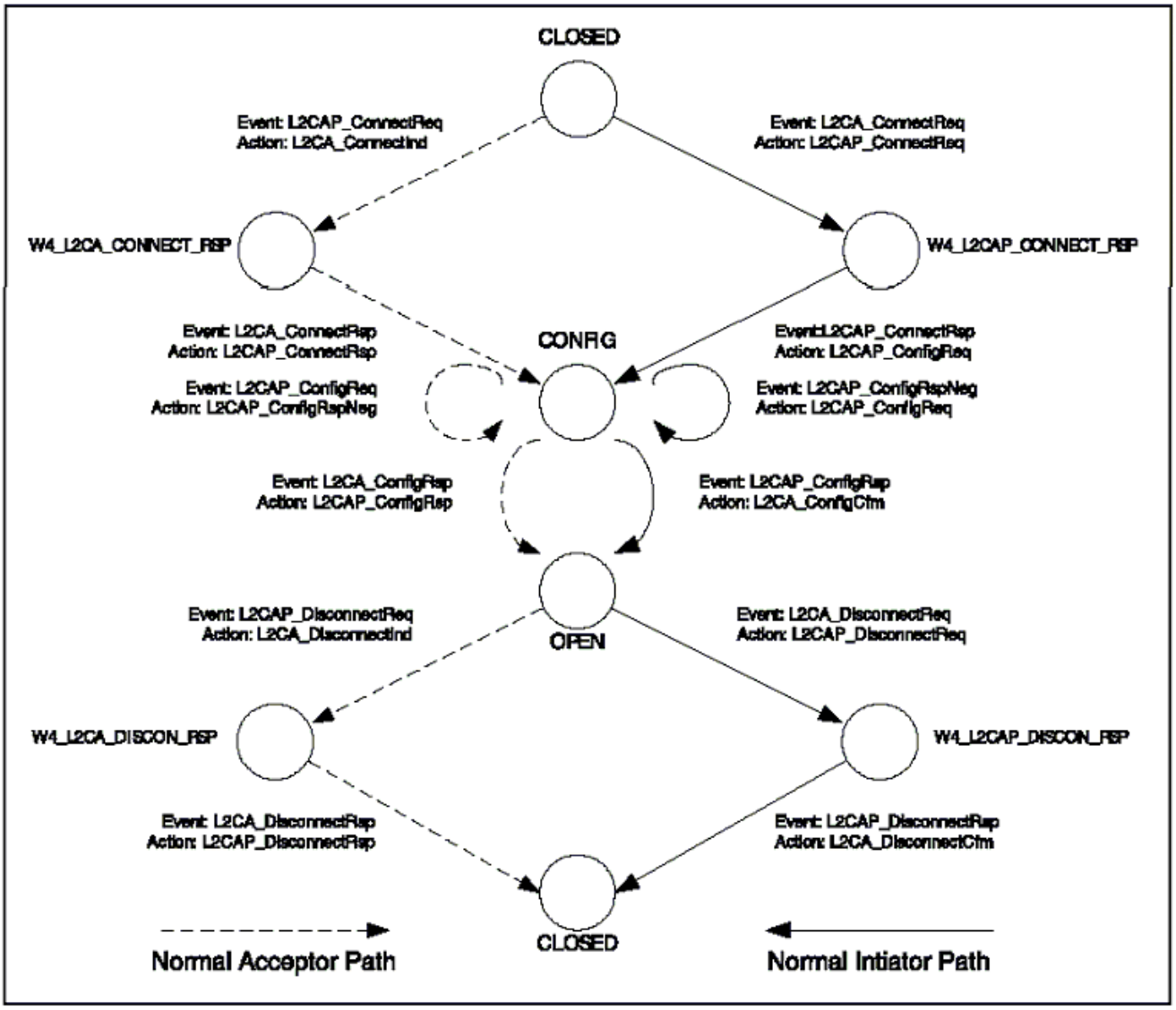


Figure 4. State Machine Example

Figure 4 above illustrates a simplified state machine and typical transition path taken by an initiator and acceptor. The state machine shows what events cause state transitions and what actions are also taken while the transitions occur. Not all the events listed in the specification are included in the simplified State Machine to avoid cluttering the figure.

Figure 5 below presents an illustration of the events and actions based around the messages sequences being communicated between two devices. In this example, the initiator is creating the first L2CAP channel between two devices. Both sides start in the CLOSED state. After receiving the request from the upper layer, the entity requests the lower layer to establish a physical link. If no physical link exists, LMP commands are used to create the physical link between the devices. Once the physical link is established, L2CAP signals may be sent over it. Figure 5 is an example and not all setup sequences will be identical to the one illustrated below.

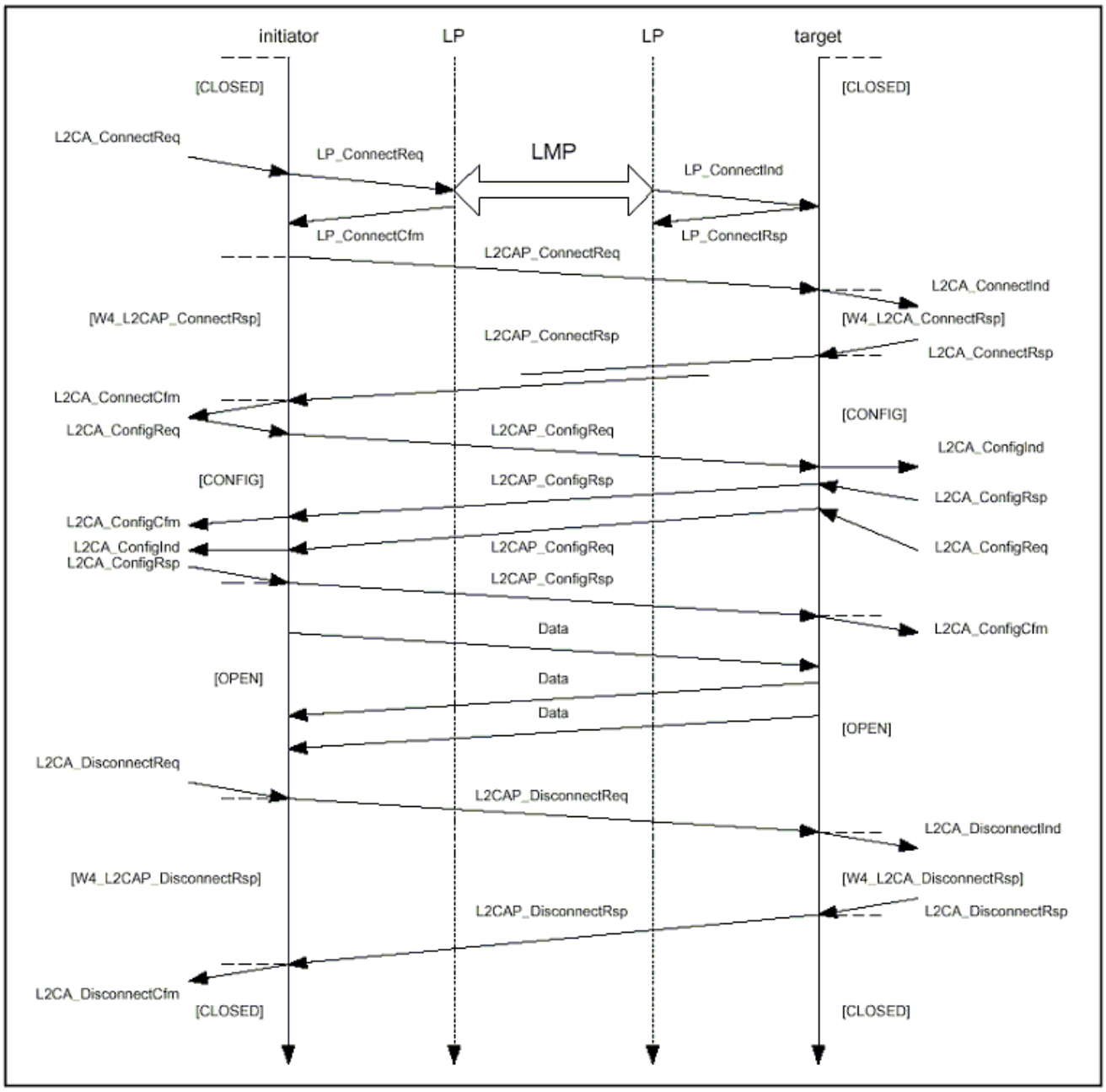
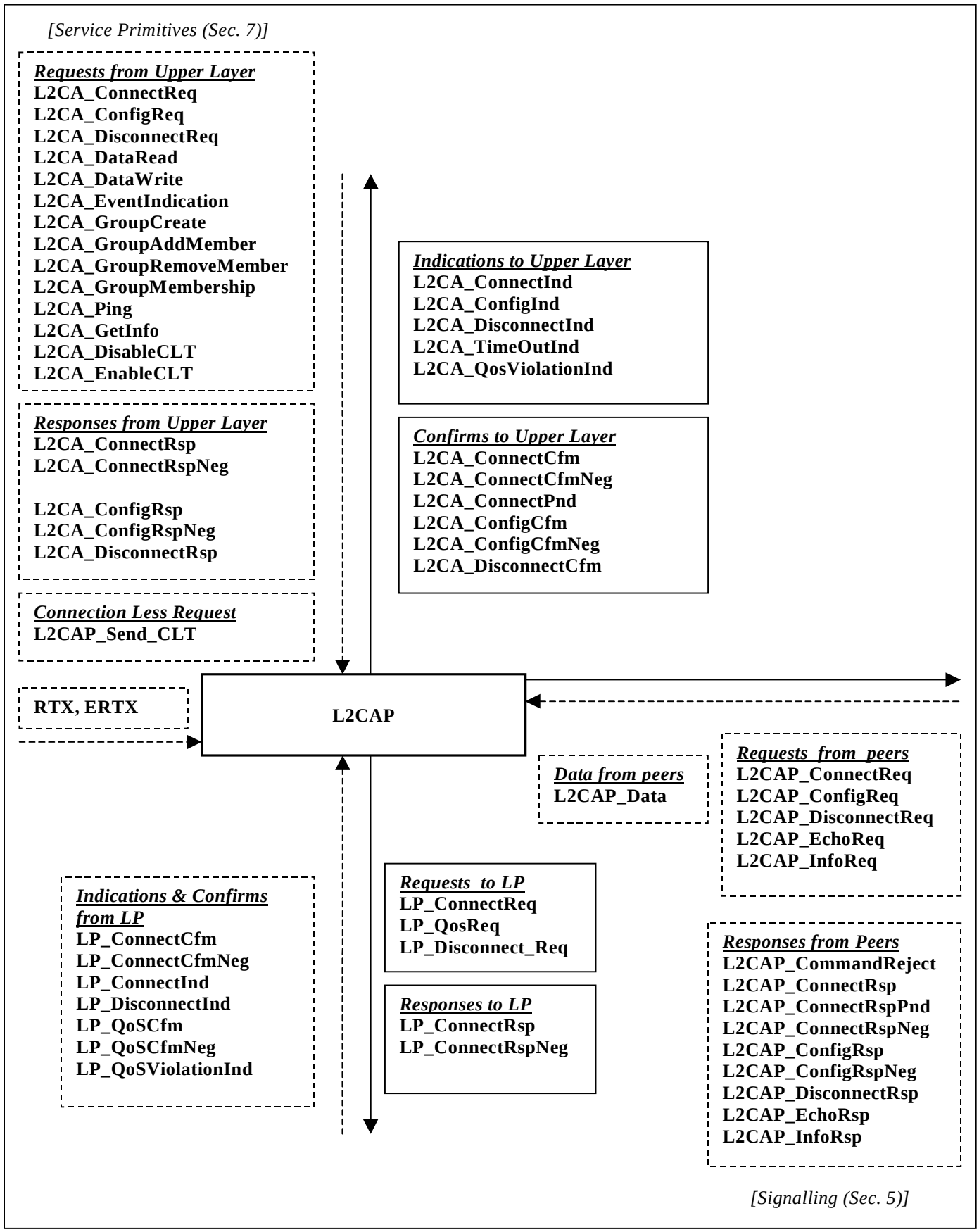


Figure 5. Message Sequence Chart of Basic Operation

2.2 Dependencies for support of Bluetooth Core Specifications ver 1.0B and 1.1



Packet Format

Connection Oriented Data Packet format

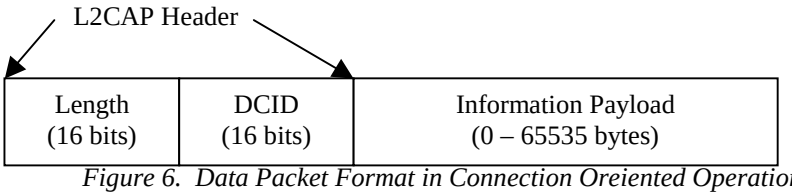


Figure 6. Data Packet Format in Connection Oreiented Operation

- Short L2CAP Packet Header (low overhead)
- Length (of payload)
- Destination Channel ID
- Payload
    1. Data received from and sent to the Network Layer
    2. Maximum Transmission Unit (MTU) limits the payload size.

Connection Less Data Packet Format

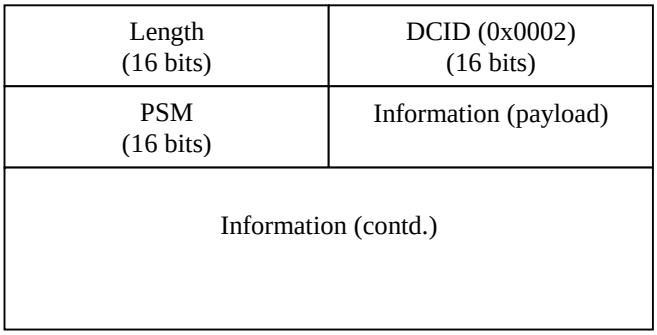


Figure 7. Data Packet Format in Connection less Operation

2.3 Segmentation and Reassembly

The Logical channel information is used from the Baseband to determine the start of the L2CAP Packet.

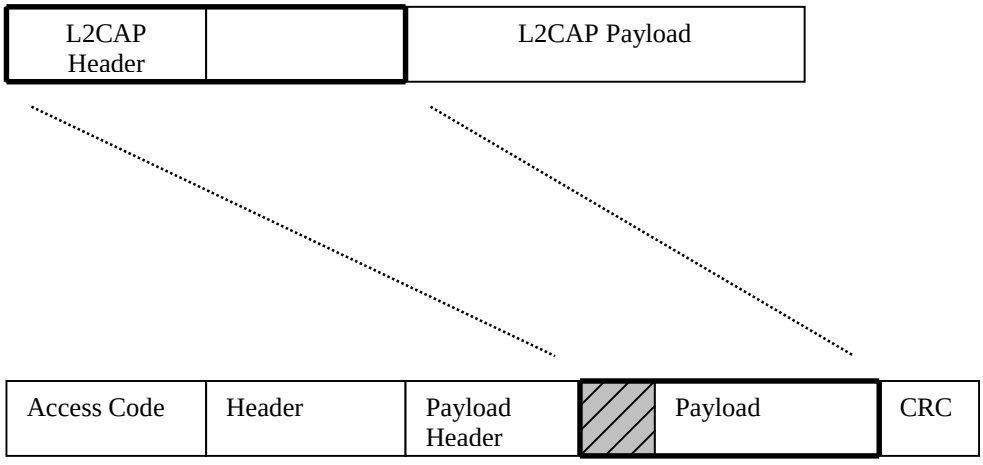


Figure 8. Segmentation Operation in L2CAP

An Example of SAR (Segmentation and Reassembly)

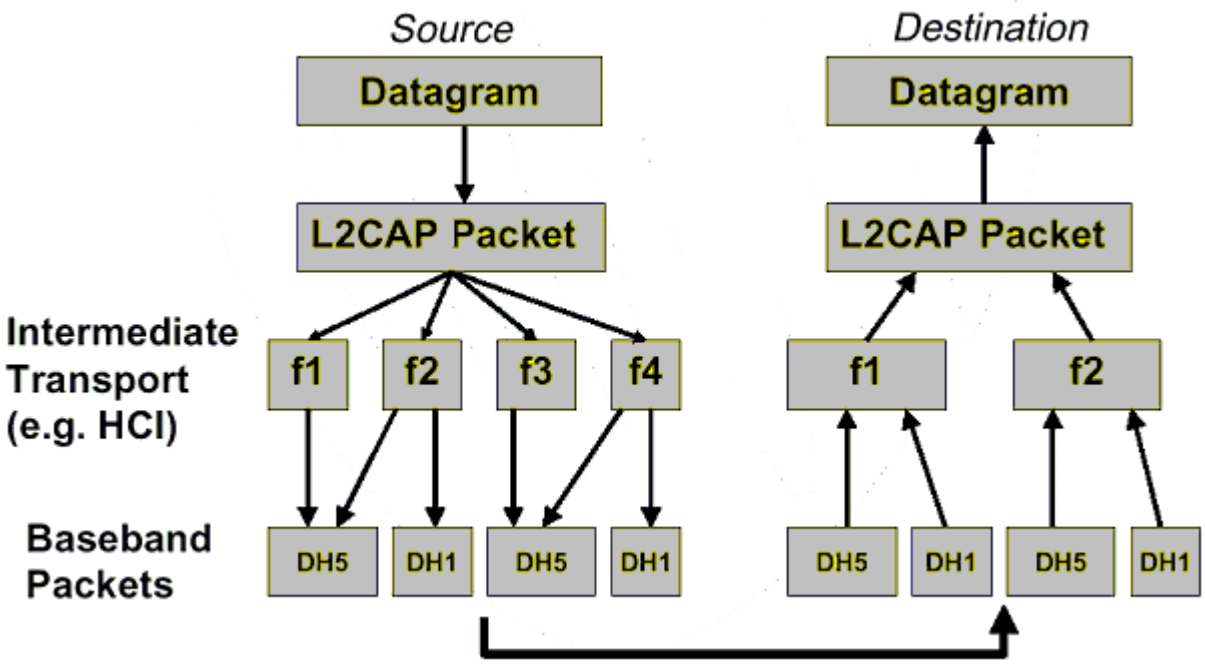


Figure 9. Segmentation and Reassembly Operation in L2CAP

3. Description and Design of Bluetooth Baseband Simulator

It simulates the Baseband and link layers of the Bluetooth Specification.
The Baseband simulator replaces the link between two Bluetooth Hardware modules with a serial cable (or Ethernet connection). It may be useful in two scenarios:

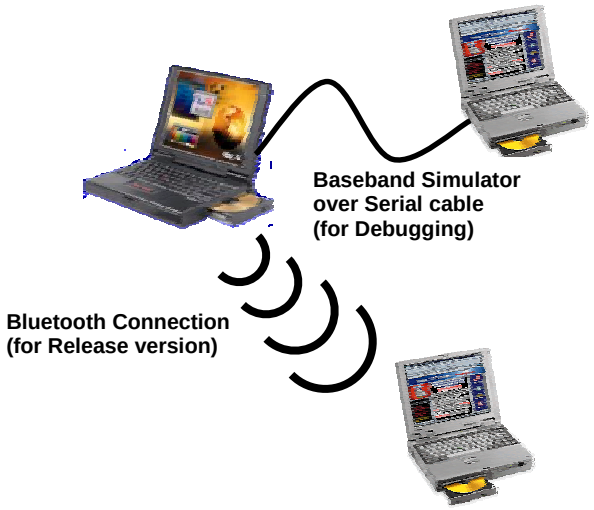
a) **During Bluetooth Hardware Development: A**s a reference for verifying the functionality of the hardware. The behaviour of the hardware may be verified against Simulator's behaviour for a given set of HCI commands.
b) **During Application Development:** The simulator can be used in place of expensive Bluetooth hardware during debugging phase. There is no need to buy expensive Sniffer equipment as the simulator provides packet capture and display feature too.

The Baseband simulator is written in ANSI C and provides a well-defined set of interfaces. It implements the LMP (Link Manager Protocol) layer, and provides interface to the HCI layer. It can either be linked with HCL Bluetooth Protocol Stack (HBPS) or any application that needs to transfer HCI packets.

The simulator uses an OS independent architecture. It supports most of the commonly used HCI commands. Currently it has been tested with Windows NT/98 and Windows CE.

Main Product Features

- Compliant to Bluetooth specification 1.0B.
- Can hook with either HBPS or any other application that needs to transfer HCI packets.
- OS independent architecture – easily customizable across various platforms. First version is available for Windows CE based CEPC (x86) platform and Windows NT/98
- Written in ANSI C for independence of development environment (compiler, debugger, linker etc.). Provides a well-defined interface that the applications can use to access the services provided.
- Provides packet display feature, obviating the need for expensive Sniffer equipment.
- Currently the simulator supports the RS232 interface at 115kbps and Ethernet.



Interfaces
The simulator provides very simple interfaces to the top layer, so that it is easy to link this with application code.

For Initialisation and De-Initialisation, the following two routines are used.

```
STATUS InitSim(void);
STATUS DeInitSim(void);
```

HCI packets can be sent and received through the following two functions.

```
STATUS SendSim(IN u8  pkt_type, IN u8 *pkt, IN u16  length);
STATUS RecvSim(OUT u8 *pkt_type, OUT u8 *pkt, OUT u16 *length);
```

The SendSim function sends a packet to the simulator. It takes as input the type of the packet, a pointer to the packet and the length of the packet to send. It returns either after the packet has been sent successfully (return value = 0) or an error occurs (return value indicates error code).

The RecvSim function receives a packet from the simulator interface. It returns the packet type, packet data and the length of the packet. This function blocks until a packet has been received and would return only when either a packet has been received successfully (return value is 0) or an error occurs (return value indicates error code).

Besides these functions, the simulator contains switches to turn packet display on and off.

These functions in turn use the serial (or Ethernet) interface provided by the OS to transport the packets from one end to another. The functions are coded in an OS Independent manner, so that for supporting a new operating system, only the routines that perform actual reading and writing to the serial (or Ethernet) port need to be replaced.

This simulator will process the HCI packets and send it over a physical link to the peer simlulator. It also gives support to the following events. These are a subset of the possible total events.

1. **Create_Connection**:: Command Status Event & Connection Complete Event after LM determines that a connection with remote BD_ADDR is established, No Command Complete Event
2. **Disconnect** ::Command Status Event & Disconnection Complete Event, No Command Complete Event
3. **Reject_Connection_Request**:: Command Status Event & Connection Complete Event, No Command Complete Event
4. **Accept_Connection_Request**:: Command Status Event & Connection Complete Event, No Command Complete Event

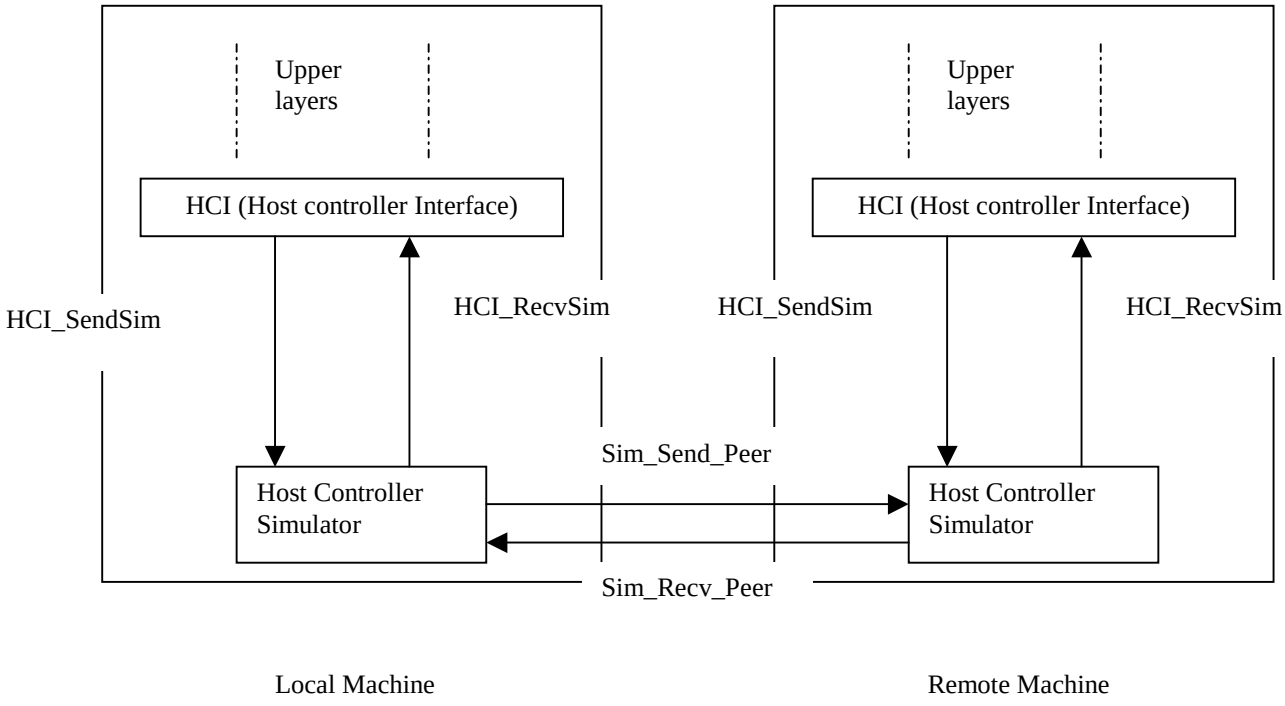


Figure 10. Baseband Simulator

4. Description and design of the TCS Protocol Layer

The TCS layer is based on the ITU-T Recommendations Q.931. TCS Binary is a protocol that provides support for cordless telephony functionality over the Bluetooth air interface. The signalling data of TCS Binary is transmitted across the Bluetooth air interface using L2CAP as the transport protocol. Voice transport utilizes SCO links, which are controlled from TCS Binary via the Bluetooth module control (BTMC) and/ or the Host Controller interface (HCI). The supplementary services included within the TCS Binary specification are support for Calling Line Identity Presentation (CLIP). TCS Binary also provides a mechanism for sending Dual-Tone Multi-Frequency (DTMF) and Register Recall signals across the air interface.
The TCS contains the following functionality:

- *Call Control* ***(CC)*** - Signalling for the establishment and release of speech and data calls between Bluetooth Devices.
- *Group Management* ***(GM)*** - Signalling to ease the handling of groups of Bluetooth devices.
- *Connectionless TCS* ***(CL)*** - Provisions to exchange signalling information not related to an ongoing call.

In The proposed design all the supplementary features will be supported, apart from supporting the mandatory features.
The Bluetooth specification includes two profiles that utilize the TCS Binary as defined by the Bluetooth Core specification

- *The Cordless Telephony Profile*: defines the interoperability requirements of Bluetooth devices acting as either a Gateway (GW) or as Terminals (TL), where a gateway is a 'base station' that connects to the ISDN or fixed telecommunications network, and provides cordless communications services to the Terminals. The Terminals are typically cordless handsets.
- *The Intercom Profile*: defines the interoperability requirements for direct Terminal-to-Terminal communication. A Cordless Telephony Application may support either or both profiles. It is however, unlikely that a Gateway will directly support the Intercom Profile, but it can include (group) management functionality that allows direct Terminal to Terminal communication to be achieved with reduced call set up times.

The design does not discriminate between user and network side, but merely between Outgoing Side (the party originating the call) and Incoming Side (the party terminating the call).
TCS uses point-to- point signaling and may use point-to-multi-point signaling. Point-to-point signaling is mapped towards a connection-oriented L2CAP channel, whereas point-to-multi-point signaling is mapped towards the connectionless L2CAP channel, which in turn is sent as broadcast information on the beacon channel (piconet broadcast).
For handling more calls simultaneously, multiple instances of TCS Binary may exist at the same time, i.e. the implementation will have a different connection record or object for different L2CAP channel identifier
The Call Control entity controls the LMP directly, for the purpose of establishing and releasing SCO links.
The Group Management entity controls the LMP and LC/Baseband directly during the initialization procedures to control (for example) the inquiry, paging and pairing.
The connection-oriented L2CAP channel between the Outgoing and Incoming side shall be available before any of the CC procedure can operate, Additionally, in a multi-point configuration, a connectionless L2CAP channel shall be available between the Outgoing and Incoming Side.
A message shall always be delivered in a single L2CAP packet. The start of the message (the LSB of the first octet) shall be aligned with the start of the L2CAP payload.
The internal structure of TCS Binary contains the functional entities Call Control, Group Management and ConnectionLess.

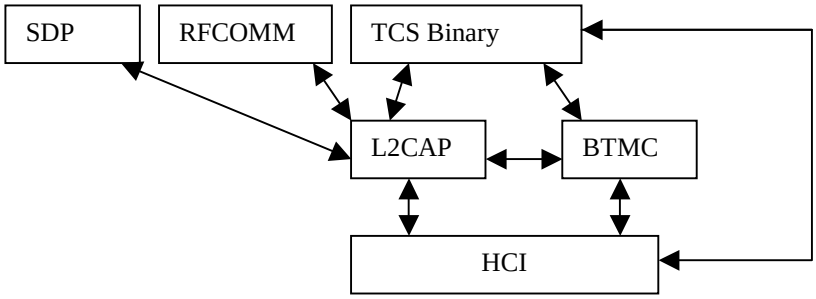


Figure 11. TCS in HBPS

The TCS Binary layer interacts with L2CAP, BTMC and HCI layer in the HCL Bluetooth protocol stack as shown in Figure 11.
TCS uses point-to-point signaling and may use point-to-multi-point signaling.
Point-to-point signaling is used when it is known to which side (Bluetooth device) a call needs to be established (single-point configuration).

Point-to-multi-point signaling may be used when there are more sides available for call establishment (multi-point configuration); e.g. when, for an incoming call, a home base station needs to alert all phones in range.
Point-to-point signaling is mapped towards a connection-oriented L2CAP channel, whereas point-to-multi-point signaling is mapped towards the connectionless L2CAP channel, which in turn is sent as broadcast information on the beacon channel (piconet broadcast).

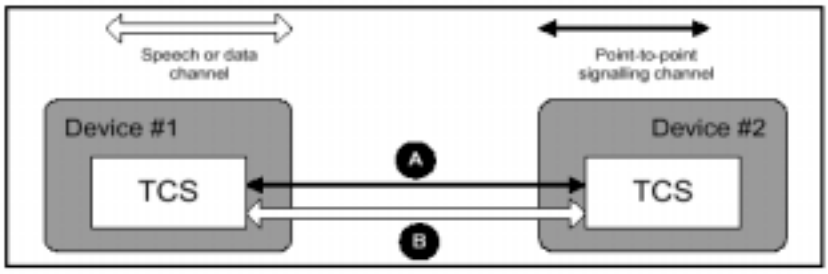


Figure 12. Point-to-point signaling in a single-point configuration

Figure 12 above illustrates point-to-point signalling to establish a voice or data call in a single-point configuration. First the other device is notified of the call request using the point-to-point signalling channel (A). Next, this signalling channel is used to further establish the speech or data channel (B).

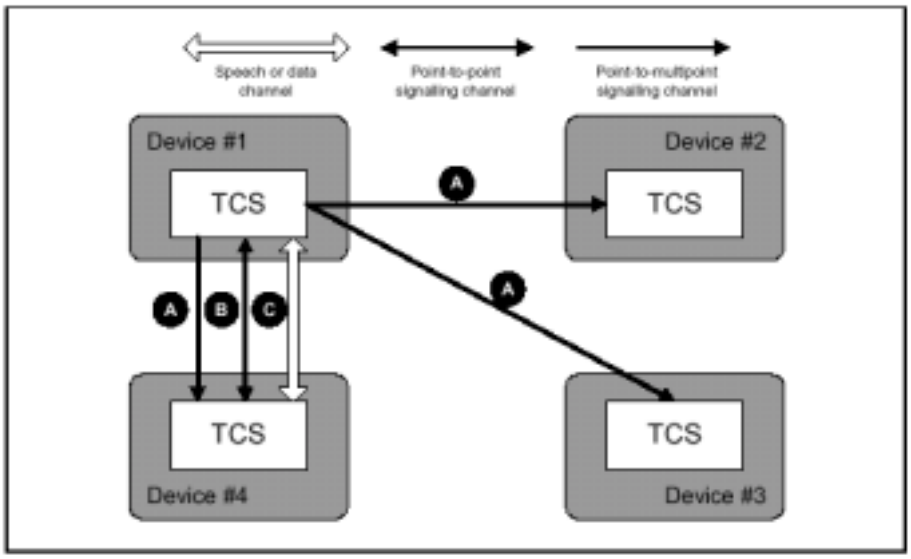


Figure 13. Signaling in a multi-point configuration

Figure 13 above illustrates how point-to-multi-point signalling and point-to-point signalling is used to establish a voice or data call in a multi-point configuration. First all devices are notified of the call request using point-to-multi-point signalling channel (A). Next, one of the devices answers the call on the point-to-point signalling channel (B); this signalling channel is used to further establish the speech or data channel (C). In such a scenario, when multiple devices respond to the Set-up Request, then the design proposes that further message exchange is done with the first device that responds positively to the Set-up Request, and all the remaining devices are sent a Release Complete message.

4.1.1 CALL CONTROL

A connection-oriented L2CAP channel between the Outgoing and Incoming Side shall be available before any of the CC procedures can operate. Additionally, in a multi-point configuration, a connectionless L2CAP channel shall be available between the Outgoing and Incoming Side.

4.1.1.1 Call States:

The call states used by the TCS are those identified in Q.931 [1], for the user side only. To allow for implementation within computing power- and memory- restricted devices, only a subset of the states is mandatory for TCS based implementations. This mandatory subset is termed **Lean TCS**. The states are named as follows. States in bold are mandatory states, part of Lean TCS:
General States
Null (0)
Active (10)
Disconnect request (11)
Disconnect indication (12)
Release request (19)
Outgoing Side States
Call initiated (1)
Overlap sending (2)
Outgoing call proceeding (3)
Call delivered (4)
Incoming Side States
Call present (6)
Call received (7)
Connect request (8)
Incoming call proceeding (9)
Overlap receiving (25)
The design supports both the Lean TCS and the Full version of TCS.

4.1.1.2 Call Establishment Message Flow:

Figure 14 below provides a complete view of the messages exchanged during successful Call Establishment, as described in the sections above. A solid arrow indicates the mandatory messages, part of the Lean TCS. A dotted arrow indicates the optional messages. A triangle indicates a running timer.

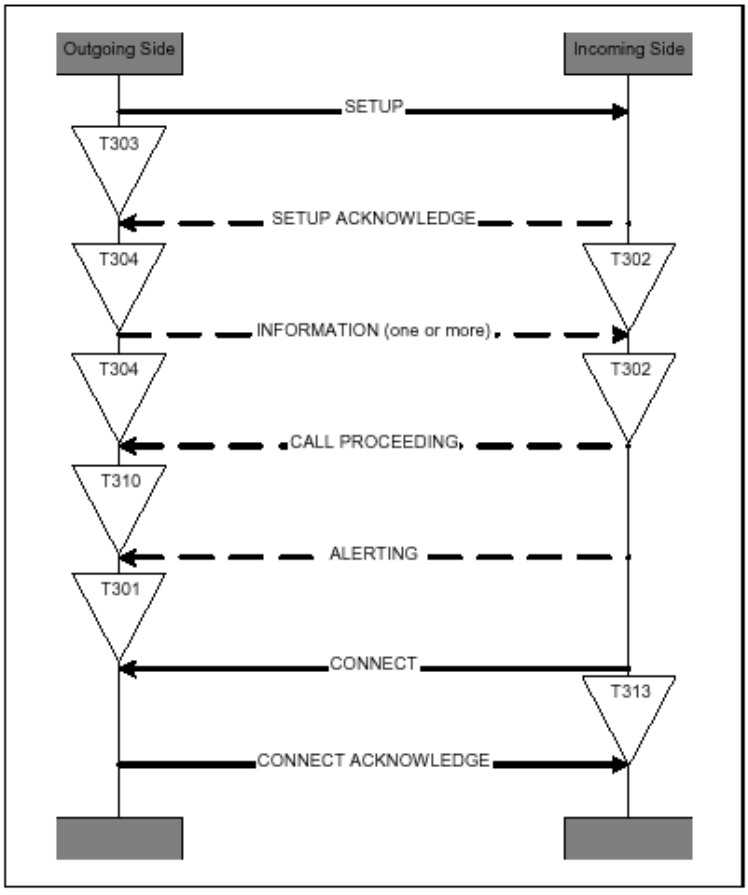


Figure 14. Call establishment message flow

4.1.1.3 Call Clearing Message Flow

Figure 15 below provides the complete view on the messages exchanged during normal Call Clearing. All messages are mandatory.

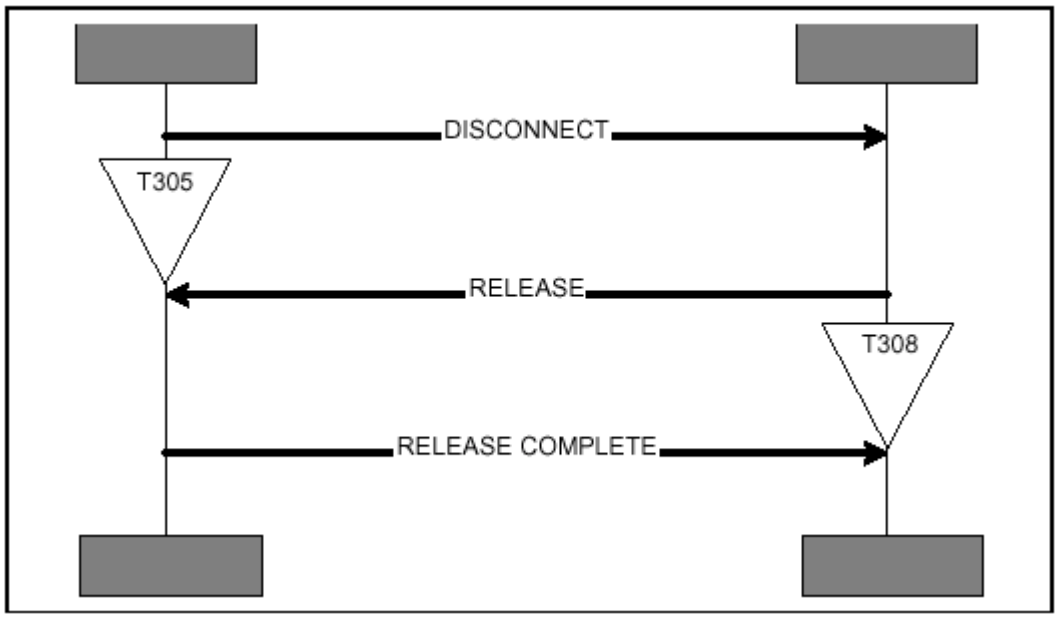


Figure 15. Call clearing message flow

4.1.1.4 TCS Call States

Figure 16 below explains the states and the message flows in each of the sates, before a connection has been established in case of a Full TCS implementation, i.e. when both the mandatory and the optional states are implemented. The figure is self – explanatory.

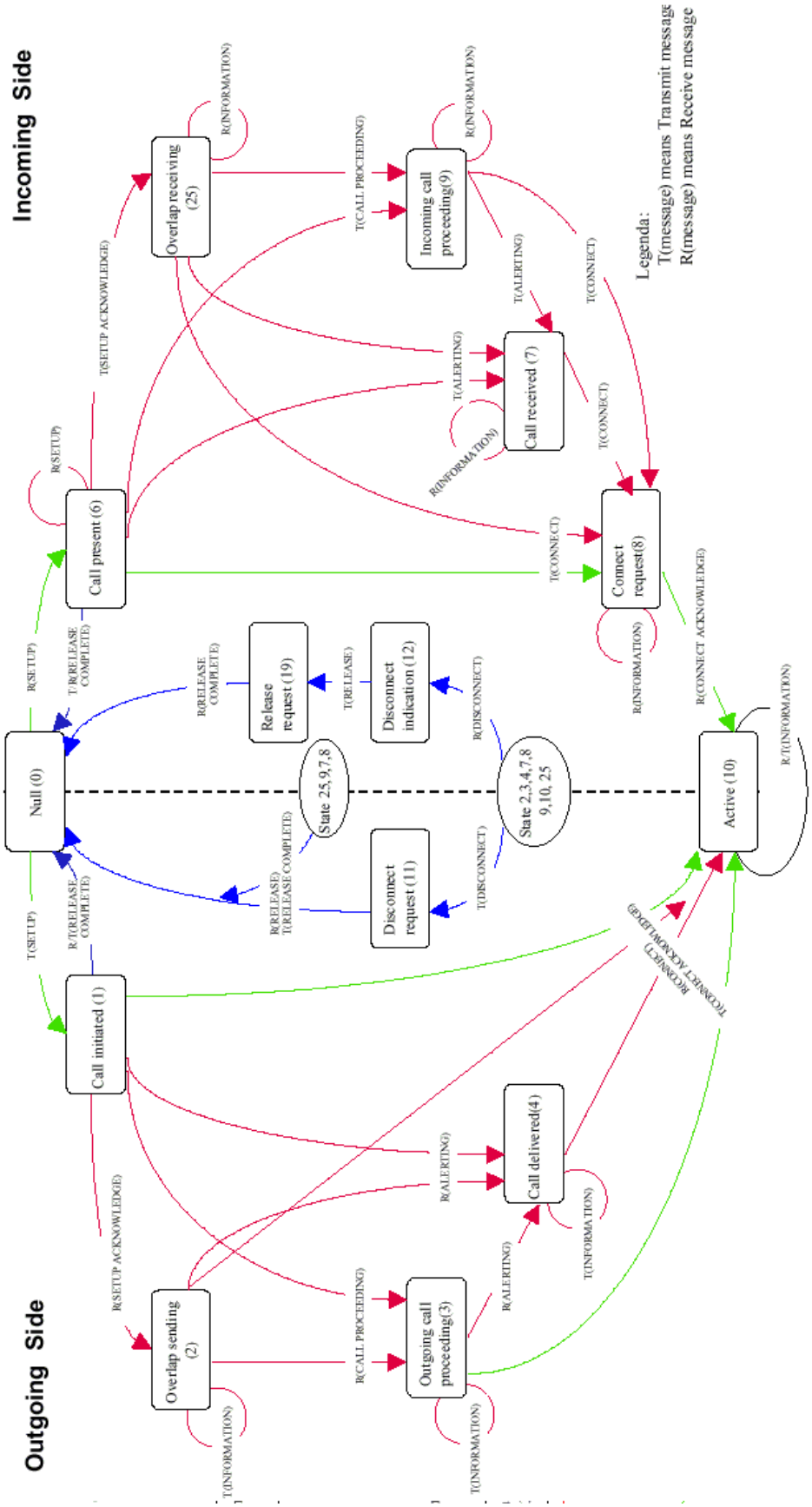

Incoming Side
Outgoing Side
Legenda:
T(message) means Transmit message
R(message) means Receive message
Null (0)
Call initiated (1)
Overlap sending (2)
Outgoing call proceeding(3)
Call delivered(4)
Call present (6)
Call received (7)
Connect request(8)
Active (10)
Incoming call proceeding(9)
Overlap receiving (25)
Release request (19)
Disconnect indication (12)
Disconnect request (11)
State 25,9,7,8
State 2,3,4,7,8 9,10, 25
R(SETUP)
T(SETUP)
T(RELEASE COMPLETE)
R(RELEASE COMPLETE)
T(RELEASE)
R(DISCONNECT)
T(DISCONNECT)
R(RELEASE) T(RELEASE COMPLETE)
R(SETUP ACKNOWLEDGE)
T(SETUP ACKNOWLEDGE)
R(CALL PROCEEDING)
T(CALL PROCEEDING)
R(ALERTING)
T(ALERTING)
R(CONNECT)
T(CONNECT)
R(CONNECT ACKNOWLEDGE)
T(CONNECT ACKNOWLEDGE)
R(INFORMATION)
T(INFORMATION)
R/T(INFORMATION)

Figure 16. Full TCS State Diagram

Figure 17 below explains the states and the message flows in each of the sates, before a connection has been established in case of a Lean TCS implementation, i.e. when only the mandatory states are implemented. The figure is self – explanatory.

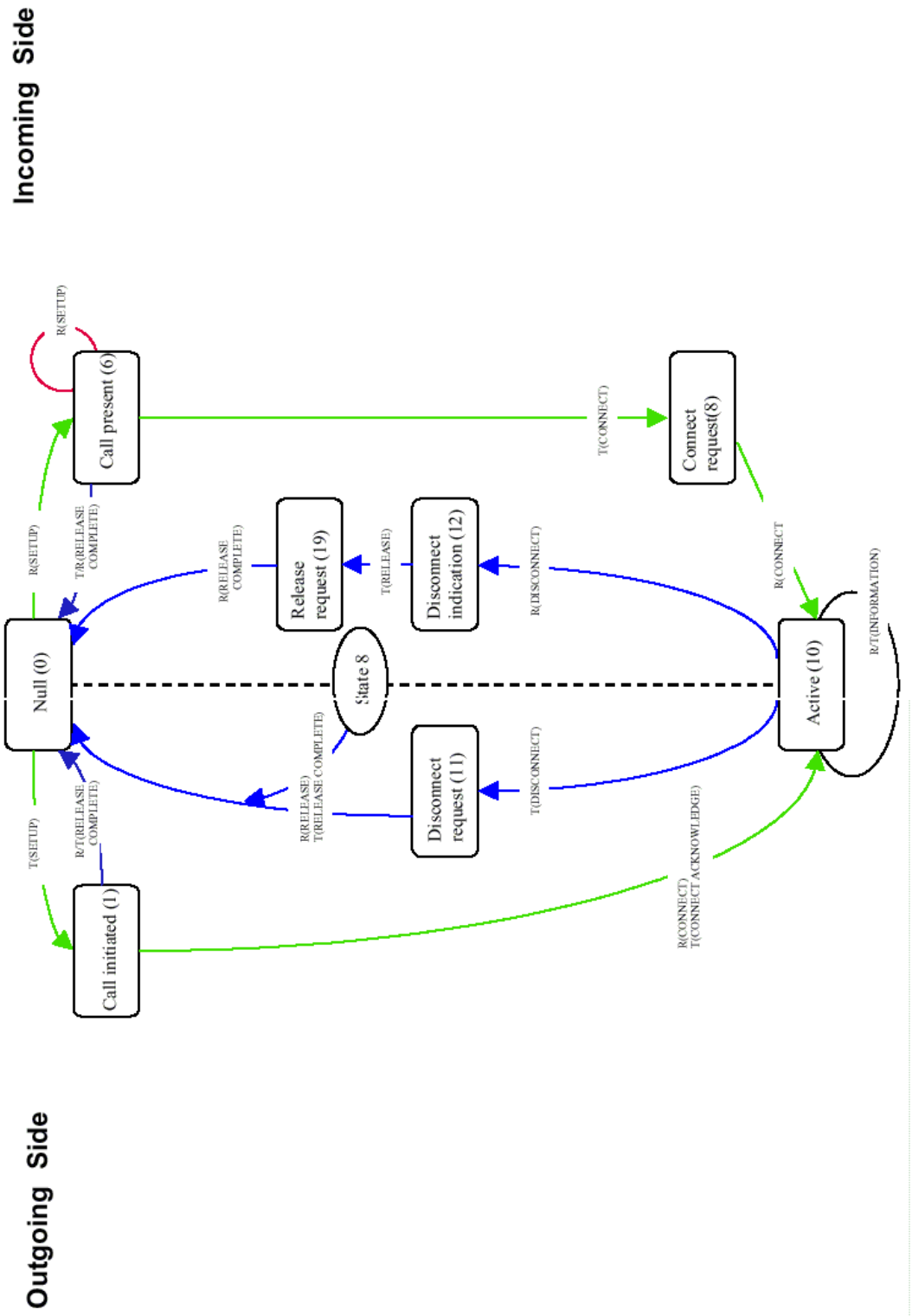


Figure 17. Lean TCS State Diagram

4.1.1.5 Message Sequence Chart

The Message Sequence Chart given below in Figure 18, explains a typical scenario where in, there are Telephony applications running over the TCS Binary Layer. The set of messages is not exhaustive, but indicative.

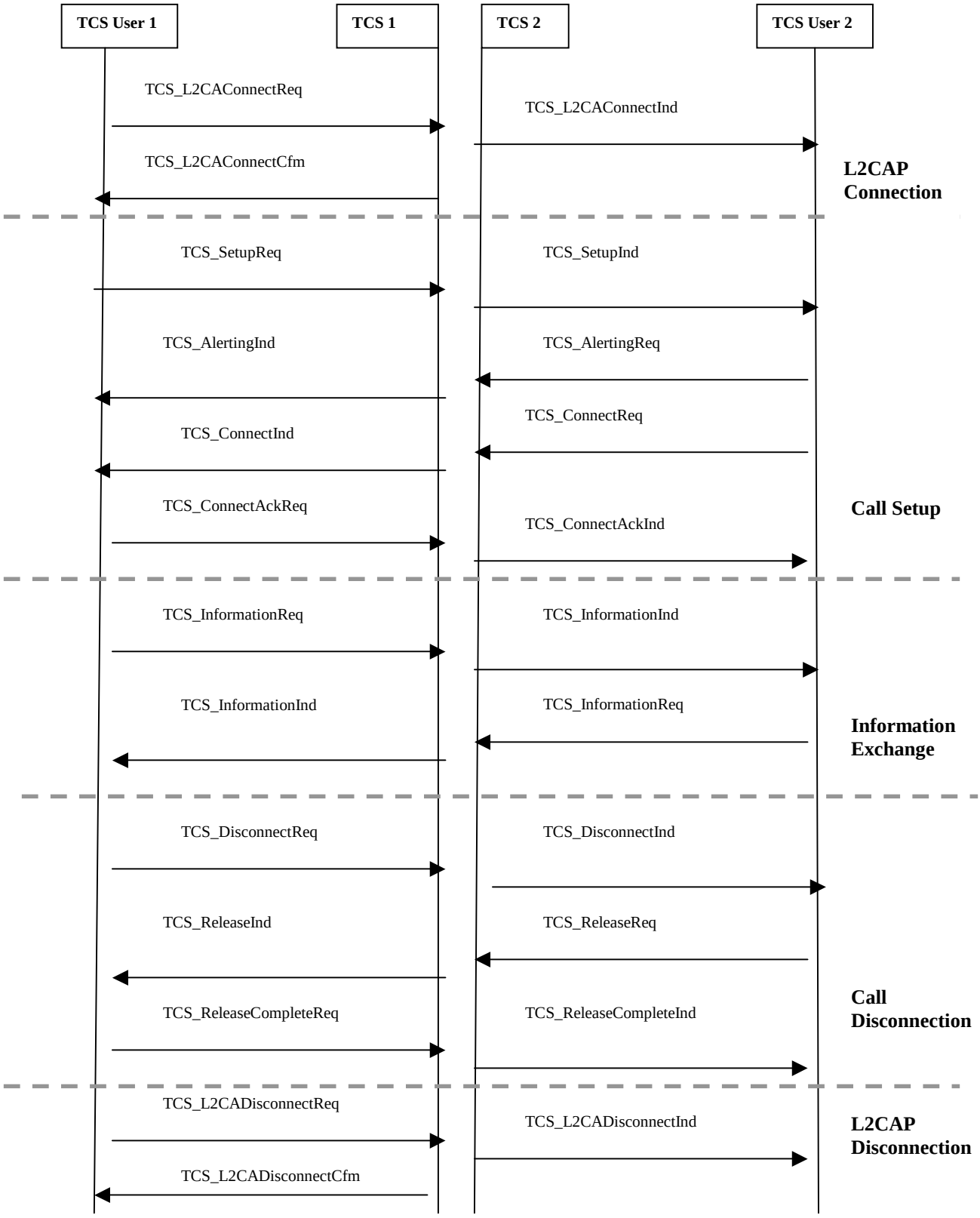


Figure 18. Message Sequence Chart

4.1.1.6 Protocol Timers

Figure 19 below, lists the value of the Timers that will be used by the TCS layer in the proposed design.

Timer name	*Value*
T301	Minimum 3 minutes
T302	15 seconds
T303	20 seconds
T304	30 seconds
T305	30 seconds
T308	4 seconds
T310	30 –120 seconds
T313	4 seconds
T401	8 second
T402	8 seconds
T403	4 second
T404	2.5 seconds
T405	2 seconds
T406	20 seconds

Figure 19. Timer Values

4.1.2 GROUP MANAGEMENT

The Group Management entity provides procedures for managing a group of devices.
The following procedures are supported:

- *Obtain access rights*: enables the requesting device to use the telephony services of another device, part of a group of devices
- *Configuration distribution*: facilitates the handling and operation of a group of devices
- *Fast inter-member access*: enables faster contact establishment between devices of the same group

A connection-oriented L2CAP channel between devices shall be available before any of the GM procedures can operate. For group management, the concept of Wireless User Group (WUG) is used.

4.1.2.1 Wireless User Group (WUG)

A WUG consists of a number of Bluetooth units supporting TCS. One of the devices is called the WUG master. The WUG master is typically a gateway, providing the other Bluetooth devices – called WUG members – with access to an external network. All members of the WUG in range are members of a piconet (active or parked). Master of this piconet is always the WUG master.
The main relational characteristics of a WUG are:

- All units that are part of a WUG know which unit is the WUG master and which other units are member of this WUG. WUG members receive this information from the WUG master.
- When a new unit has paired with the WUG master, it is able to communicate and perform authentication and encryption with any other unit part of the WUG without any further pairing/initialisation. The WUG master provides the required authentication and encryption parameters to the WUG members.

Both relational characteristics are maintained through the Configuration distribution procedure. However, the scatter net ability is not mandatory for support of WUG. For fast inter- member access, the devices detaches from the master of the WUG and forms a new pair with the other terminal with which it wants to establish connection.

Figure 20. Obtain access rights message flow

Figure 20 above provides the complete view on the messages exchanged during the Obtain access rights procedure.

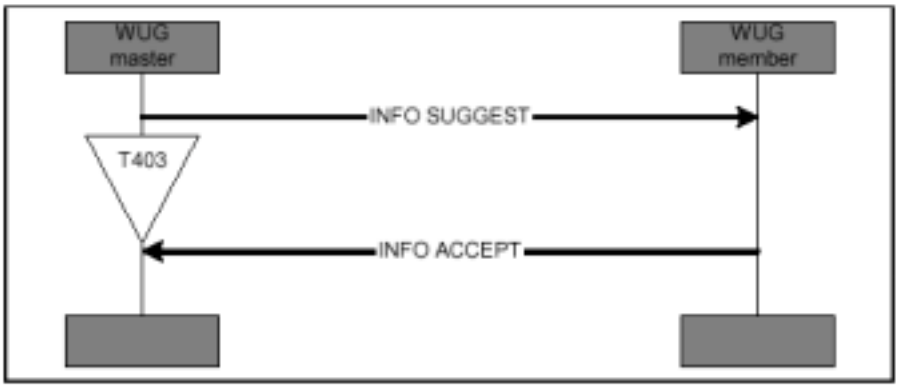


Figure 21. Configuration distribution message flow

Figure 21 above provides the complete view on the messages exchanged during the Configuration distribution procedure.

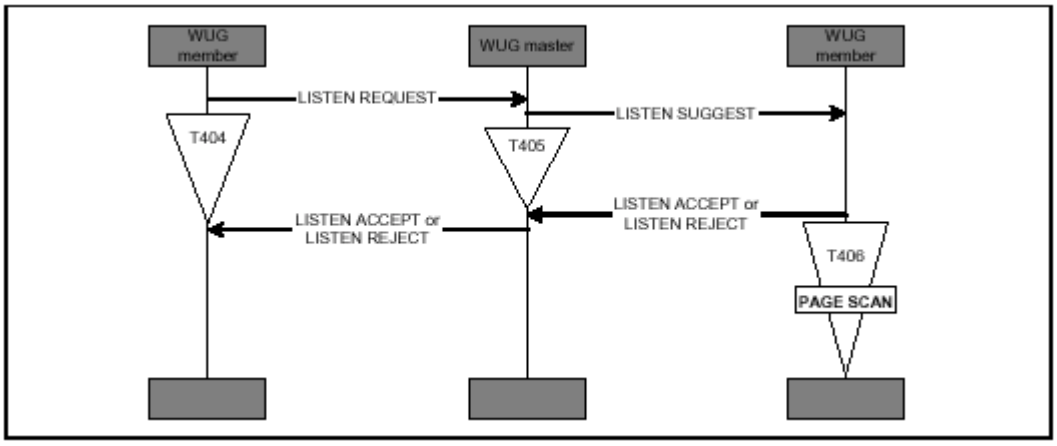


Figure 22. Fast inter-member access message flow

Figure 22 above provides a view of the messages exchanged during fast inter-member access, as described in the sections above. A successful Fast inter-member access procedure ends with the terminating WUG member going into page scan, thus allowing the originating WUG member to contact him directly.

4.1.3 CONNECTIONLESS TCS

A connectionless TCS message can be used to exchange signalling information without establishing a TCS call. It is thus a connectionless service offered by TCS. A connectionless TCS message is a CL INFO message. A connection-oriented L2CAP channel between the Outgoing and Incoming Side shall be available before a CL INFO message can be sent. In the case of a connection-oriented channel, it may choose to delay the termination of the channel for a defined period to exchange more CL INFO messages. Alternatively, in a multi-point configuration, a connectionless L2CAP channel may be used and, if so, shall be available before CL INFO can be sent.

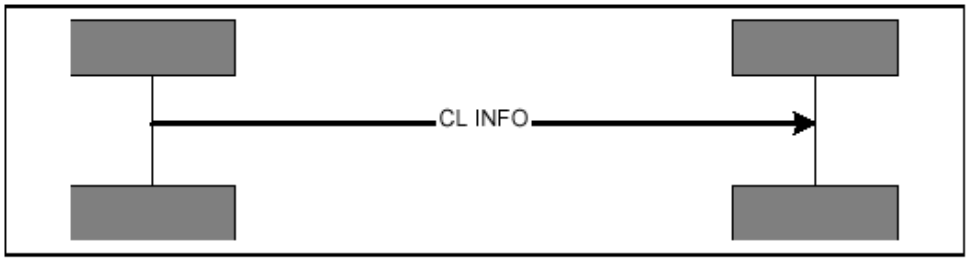


Figure 23. Connectionless TCS message flow

4.1.4 SUPPLEMENTARY SERVICES

The TCS provides explicit support for only one supplementary service, the Calling Line Identity. For supplementary services provided by an external network, using DTMF sequences for the activation/de-activation and interrogation of supplementary services, the DTMF start & stop procedure is supported. This procedure allows both finite and infinite tone lengths. For other means of supplementary service control, no explicit support is specified. Either using the service call, or use the company specific information element, or a combination may realize support.

4.1.4.1 Calling Line Identity:

To inform the Incoming Side of the identity of the originator of the call, the Out-going Side may include the calling party number information element in the SETUP message transferred as part of the call request.

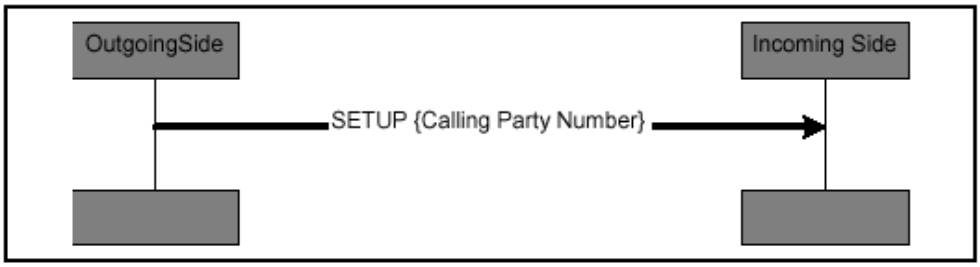


Figure 24. Calling line identity message flow

4.1.4.2 DTMF Start & Stop

The DTMF start & stop procedure is supported to provide supplementary service control on PSTN type of networks. In principle DTMF messages can be initiated by either (Outgoing or Incoming) Side; in practice, however, the Side (gateway) connected to the external PSTN network will be the recipient. DTMF messages can be transmitted only in the active state of a call. Tone generation shall end when the call is disconnected.

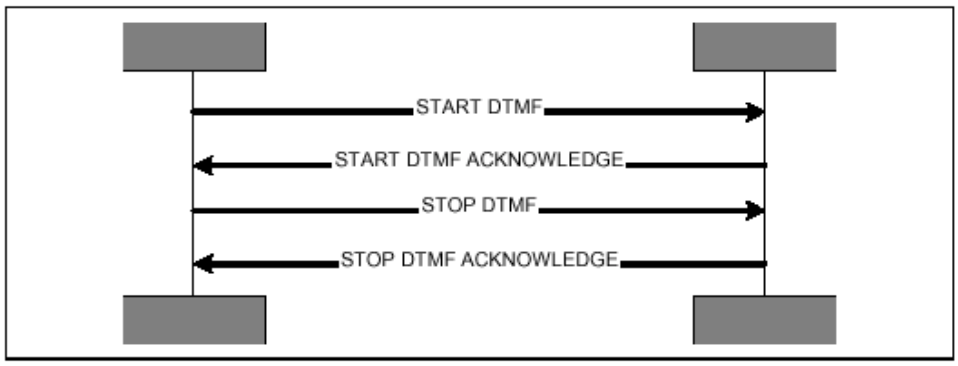


Figure 25. DTMF start & stop message flow

4.1.4.3 Register Recall

Register recall means to seize a register (with dial tone) to permit input of further digits or other action. In some markets, this is referred to as ’hook flash’. Sending an INFORMATION message with a keypad facility information element, indicating ‘register recall’ (value 16H) supports register recall.

Dependencies
For support of *Bluetooth Core Specifications ver 1.0B and 1.1*

4.1.5 Lower Layer Interfaces

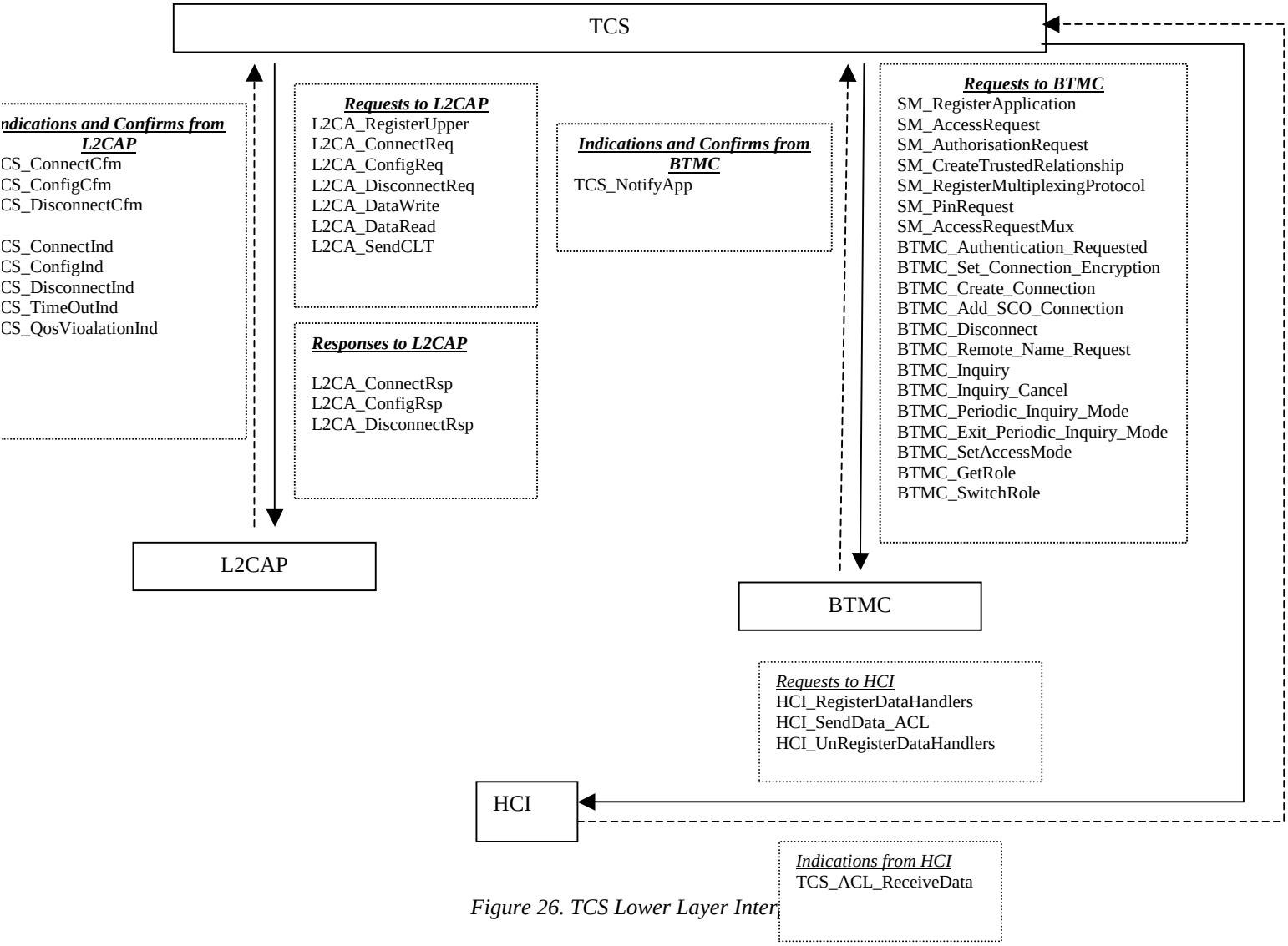


Figure 26. TCS Lower Layer Inter

4.1.6 Peer (TCS) Interface

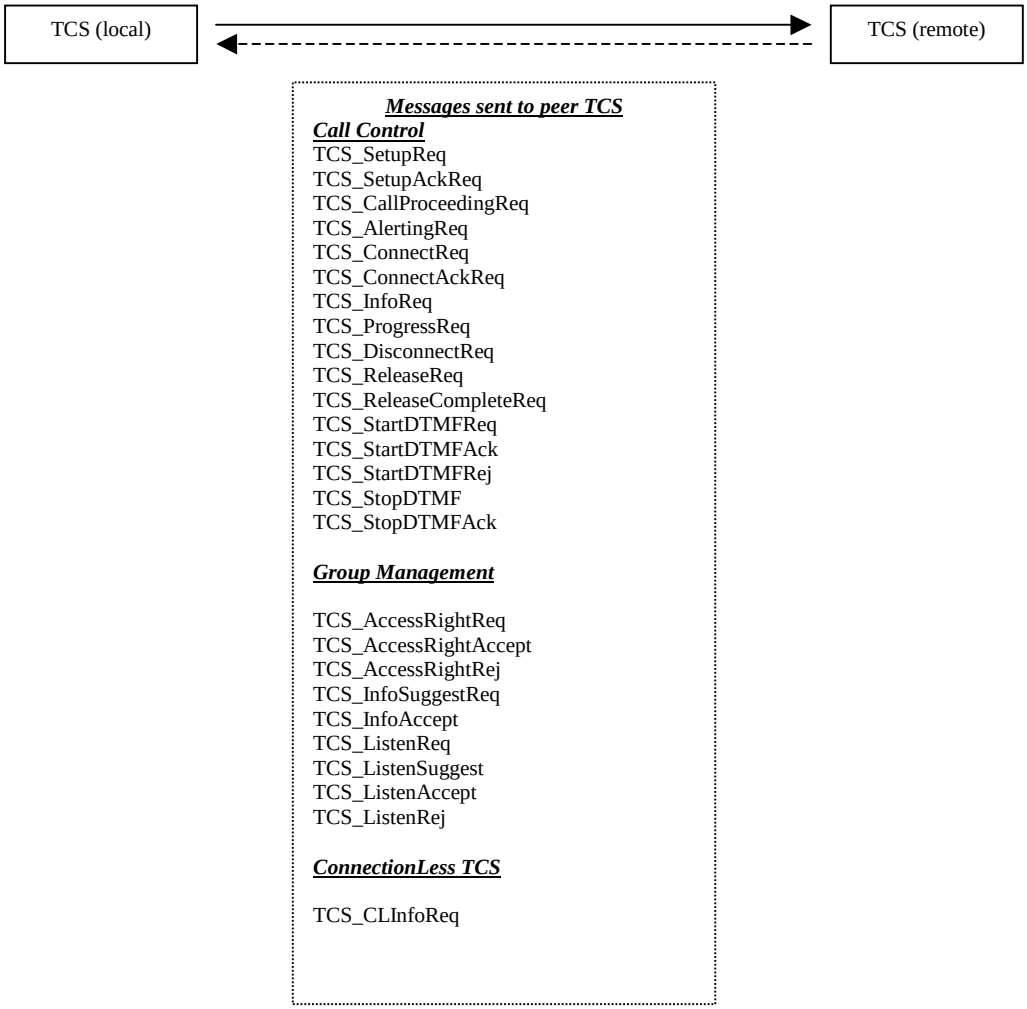


Figure 27. TCS Peer (TCS) Interface

4.1.7 Upper Layer Interface

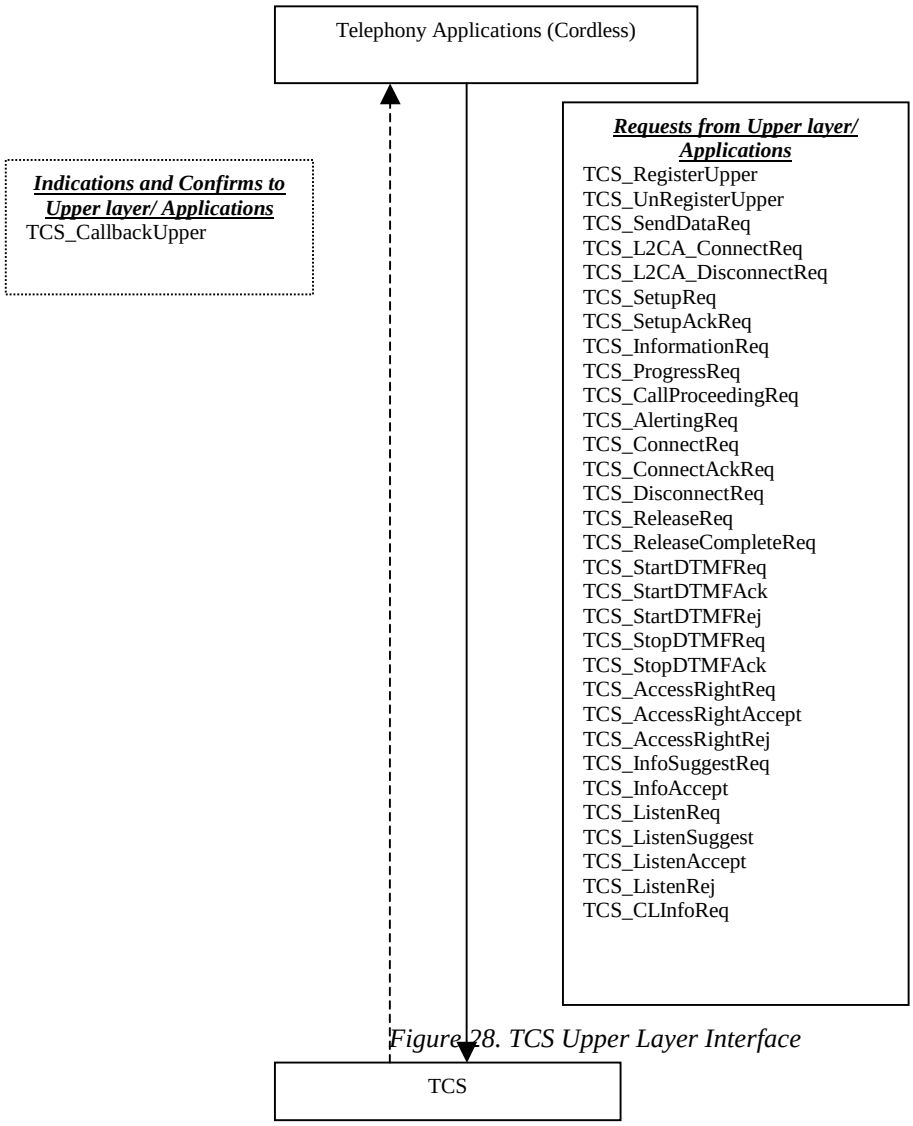


Figure 28. TCS Upper Layer Interface

4.2 Design Description

The TCS binary has six functional entities, as proposed in this design.

The first module takes care of the ***TCS interfaces***, where in there are function which are used for the initialisation and de-initialisation of the layer, during which the TCS Binary layer data handlers with HCI, and event indication call backs with L2CAP and BTMC. This module also handle the call-backs which it has registered with the lower layers, which in turn signals the processing thread of TCS by sending a predefined message to the TCS queue. Apart from this, it also exports a set of APIs to the upper layer/ application, which can be used by any Telephony application for establishment of speech or data channels.

There is another module, which handles the ***Call Control*** functionality as described in the TCS high-level design. The basic functionality of this module is to prepare TCS message packets which would be sent to the Peer TCS over the L2CAP connection oriented L2CAP channel, if it is already up, or send it after establishing the connection oriented L2CAP channel. This module also handles the messages or signalling data packets, which are received from the peer side. The messages that will be handled by this module are messages like.

1. SETUP
2. SETUP ACKNOWLEDGE
3. INFORMATION

4. SETUP ACKNOWLEDGE
5. CALL PROCEEDING
6. ALERTING
7. CONNECT
8. DISCONNECT
9. RELEASE
10. RELEASE COMPLETE
11. CONNECT ACKNOWLEDGE

The next module takes care of the ***Group Management*** functionality of the TCS Layer as explained in the TCS high-Level design. The basic functionality of this module is to prepare TCS message packets which would be sent to the Peer TCS over the L2CAP connection oriented L2CAP channel, if it is already up, or send it after establishing the connection oriented L2CAP channel. This module also handles the messages or signalling data packets, which are received from the peer side. The messages that will be handled by this module are messages like.

1. ACCESS RIGHT REQUEST
2. ACCESS RIGHT ACCEPT
3. ACCEPT RIGHT REJECT
4. INFO SUGGEST
5. INFO ACCEPT
6. LISTEN REQUEST
7. LISTEN ACCEPT
8. LISTEN REJECT

The next module takes care of the ***Connection Less*** functionality of the TCS layer as explained in the TCS high-Level design. The basic functionality of this module is to prepare TCS message packets which would be sent to the Peer TCS over the L2CAP connection oriented L2CAP channel, if it is already up, or send it after establishing the connection oriented L2CAP channel. This module also handles the messages or signalling data packets, which are received from the peer side. The messages that will be handled by this module are messages like.

1. CL INFO

The next module takes care of the ***Supplementary Services*** of the TCS layer as explained in the TCS high-Level design. The basic functionality of this module is to prepare TCS message packets which would be sent to the Peer TCS over the L2CAP connection oriented L2CAP channel, if it is already up, or send it after establishing the connection oriented L2CAP channel. This module also handles the messages or signalling data packets, which are received from the peer side. The messages that will be handled by this module are messages like.

1. START DTMF
2. START DTMF ACK
3. STOP DTMF
4. STOP DTMF ACK

The next module is the one, which is the ***Main Thread*** of the TCS Binary layer. This module processes the TCS queue and takes action upon the message that is received on the main queue. This is the module, which is involved in calling the other modules, so whenever a message is received on the TCS main queue, it invokes any one of the modules explained above which in turn takes the necessary action. The message ids are all predefined and the module, which takes care of the TCS Interfaces, will post all the messages to the message queue. So in general terms, this module will not do anything except for waiting on the message queue for the messages and invoking a module, which would in turn handle the message.

The Figure below describes how the modules interact amongst each other.

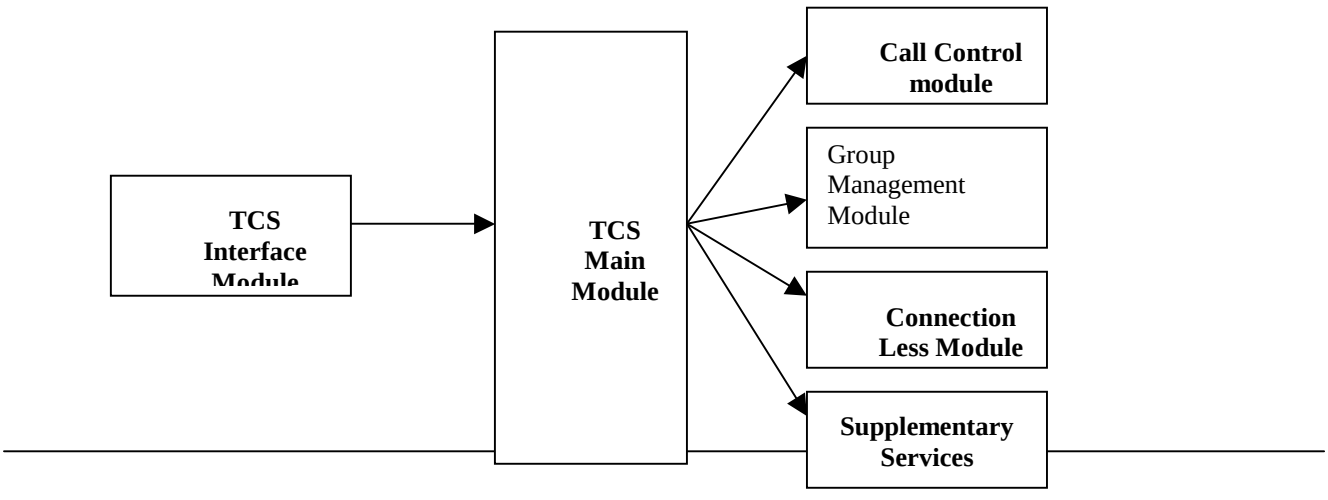
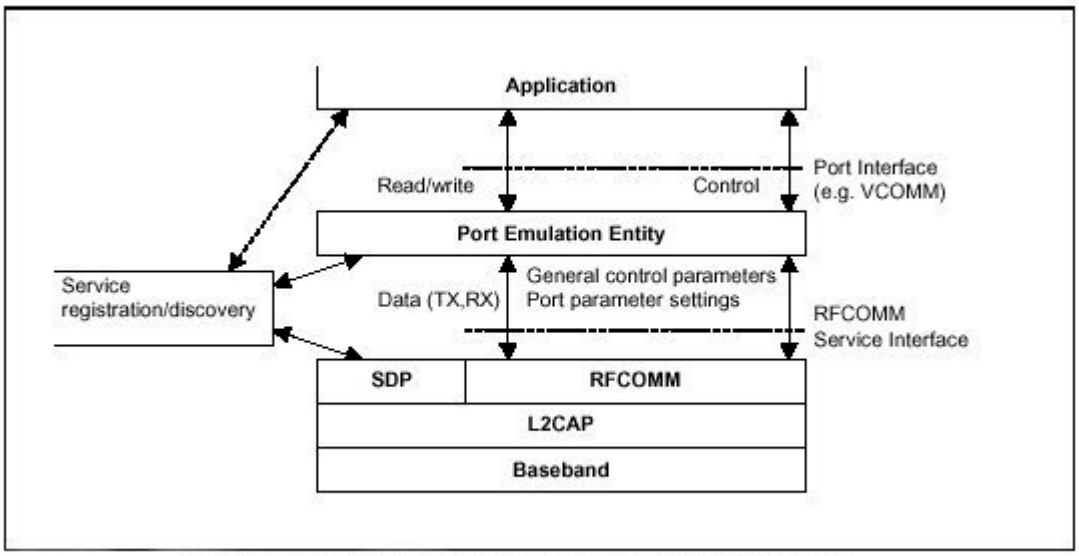


Figure29. TCS Modules (Interactions)

5. Description and Design of RFCOMM

RFCOMM layer resides over the L2CAP Layer and provides the service of emulating a serial port over the Bluetooth radio link, i.e. its purpose is to emulate the serial ports for legacy applications.
The ACL link between two units is set up using the *Link Manager Protocol*. The Baseband provides orderly delivery of data packets, although there might be individual packet corruption and duplicates. No more than one ACL link exists between any two devices. The RFCOMM layer depends on the baseband for reliable transfer of data and in it self doesn't support retransmission. So the RFCOMM only supports the best effort kind of traffic. The RFCOMM channels or DLCIs appear to be full duplex, but it doesn't imply that the baseband is full duplex too.



RFCOMM Protocol Layer in the Bluetooth Stack

The port emulation entity maps a system-specific communication interface (API) to the RFCOMM services. The port emulation entity plus RFCOMM make up a port driver.
RFCOMM relies on the lower layers for reliable transfer of data and hence it doesn't try to ensure the correctness of data but it does calculate the CRC for the header and footer fields. As a result it doesn't support those data frames of GSM 07.10 which ensure data correctness on their own through retransmissions.
RFCOMM is a complete communication path that involves two applications running on different devices with a communication segment between them. In this context, the term *application* may mean other things than end-user application, e.g. higher layer protocols or other services acting on behalf of end-user applications.

RFCOMM is a simple transport protocol, with additional provisions for emulating RS-232 serial ports. The protocol supports up to 60 simultaneous connections between two BT devices. Its connections are client-server based.