The data on the piconet channel is conveyed in packets. The general packet format is shown in Figure 15. Each packet consists of three entities: the access code, the header, and the payload. In the figure, the number of bits per entity is indicated.

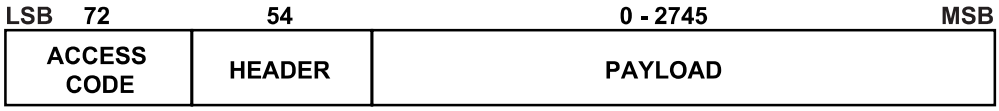


Figure 15—Standard packet format

The access code and header are of fixed size: 72 bits and 54 bits, respectively. The payload can range from zero to a maximum of 2745 bits. Different packet types have been defined. Packets may consist of the (shortened) access code only (see 8.4.4.1.1), of the access code - header, or of the access code - header - payload.

8.4.2 Access code

Each packet starts with an access code. If a packet header follows, the access code is 72 bits long, otherwise the access code is 68 bits long. This access code is used for synchronization, DC offset compensation and identification. The access code identifies all packets exchanged on the channel of the piconet: all packets sent in the same piconet are preceded by the same channel access code. In the receiver of the Bluetooth unit, a sliding correlator correlates against the access code and triggers when a threshold is exceeded. This trigger signal is used to determine the receive timing.

The access code is also used in paging and inquiry procedures. In this case, the access code itself is used as a signalling message and neither a header nor a payload is present.

The access code consists of a preamble, a sync word, and possibly a trailer; see Figure 16. For details see 8.4.2.1.

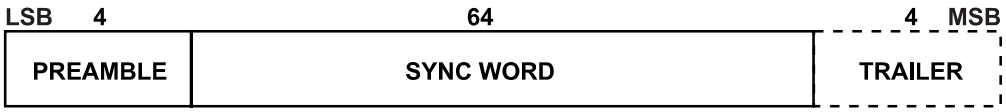


Figure 16—Access code format

8.4.2.1 Access code types

There are three different types of access codes defined:

— Channel access code (CAC)

— Device access code (DAC)

— Inquiry access code (IAC)

The respective access code types are used for a Bluetooth unit in different operating modes. The channel access code identifies a piconet. This code is included in all packets exchanged on the piconet channel. The device access code is used for special signalling procedures, e.g., paging and response to paging. For the inquiry access code there are two variations. A general inquiry access code (GIAC) is common to all devices. The GIAC can be used to discover which other Bluetooth units are in range. The dedicated inquiry

Copyright © 2002 IEEE. All rights reserved.

access code (DIAC) is common for a dedicated group of Bluetooth units that share a common characteristic. The DIAC can be used to discover only these dedicated Bluetooth units in range.

The CAC consists of a preamble, sync word, and trailer and its total length is 72 bits. When used as self-contained messages without a header, the DAC and IAC do not include the trailer bits and are of length 68 bits.

The different access code types use different Lower Address Parts (LAPs) to construct the sync word. The LAP field of the Bluetooth device address is explained in 8.13.1. A summary of the different access code types can be found in Table 12.

Table 12—Summary of access code types

Code type	LAP	Code length	Comments
CAC	Master	72	See also 8.13.2
DAC	Paged unit	68/72[a]	
GIAC	Reserved	68/72*	
DIAC	Dedicated	68/72*	

[a]length 72 is only used in combination with FHS packets

8.4.2.2 Preamble

The preamble is a fixed zero-one pattern of four symbols used to facilitate dc compensation. The sequence is either 1010 or 0101, depending whether the LSB of the following sync word is 1 or 0, respectively. The preamble is shown in Figure 17.



Figure 17—Preamble

8.4.2.3 Sync Word

The sync word is a 64-bit code word derived from a 24 bit address (LAP); for the CAC the master's LAP is used; for the GIAC and the DIAC, reserved, dedicated LAPs are used; for the DAC, the slave unit LAP is used. The construction guarantees large Hamming distance between sync words based on different LAPs. In addition, the good auto correlation properties of the sync word improve on the timing synchronization process. The derivation of the sync word is described in 8.13.2.

Copyright © 2002 IEEE. All rights reserved.

LINKPLAY EXHIBIT 1008
Page 68 of 1169

8.4.2.4 Trailer

The trailer is appended to the sync word as soon as the packet header follows the access code. This is typically the case with the CAC, but the trailer is also used in the DAC and IAC when these codes are used in FHS packets exchanged during page response and inquiry response procedures.

The trailer is a fixed zero-one pattern of four symbols. The trailer together with the three MSBs of the syncword form a 7-bit pattern of alternating ones and zeroes which may be used for extended dc compensation. The trailer sequence is either 1010 or 0101 depending on whether the MSB of the sync word is 0 or 1, respectively. The choice of trailer is illustrated in Figure 18.



Figure 18—Trailer in CAC when MSB of sync word is 0 (a), and when MSB of sync word is 1 (b)

8.4.3 Packet header

The header contains link control (LC) information and consists of six fields:

- AM_ADDR: 3-bit active member address
- TYPE: 4-bit type code
- FLOW: 1-bit flow control
- ARQN: 1-bit acknowledge indication
- SEQN: 1-bit sequence number
- HEC: 8-bit header error check

The total header, including the HEC, consists of 18 bits, see Figure 19, and is encoded with a rate 1/3 FEC (not shown but described in 8.5.1) resulting in a 54-bit header. Note that the AM_ADDR and TYPE fields are sent with their LSB first. The function of the different fields will be explained in 8.4.3.1 through 8.4.3.6.

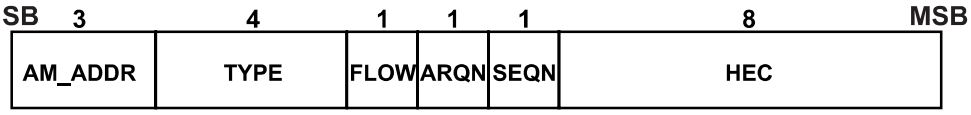


Figure 19—Header format

8.4.3.1 AM_ADDR

The AM_ADDR represents a member address and is used to distinguish between the active members participating on the piconet. In a piconet, one or more slaves are connected to a single master. To identify each slave separately, each slave is assigned a temporary 3-bit address to be used when it is active. Packets exchanged between the master and the slave all carry the AM_ADDR of this slave; that is, the AM_ADDR

Copyright © 2002 IEEE. All rights reserved.
LINKPLAY EXHIBIT 1008
Page 69 of 1169

of the slave is used in both master-to-slave packets and in the slave-to-master packets. The all-zero address is reserved for broadcasting packets from the master to the slaves. An exception is the FHS packet which may use the all-zero member address but is *not* a broadcast message (see 8.4.4.1.4). Slaves that are disconnected or parked give up their AM_ADDR. A new AM_ADDR has to be assigned when they re-enter the piconet.

8.4.3.2 TYPE

Sixteen different types of packets can be distinguished. The 4-bit TYPE code specifies which packet type is used. Important to note is that the interpretation of the TYPE code depends on the physical link type associated with the packet. First, it shall be determined whether the packet is sent on an SCO link or an ACL link. Then it can be determined which type of SCO packet or ACL packet has been received. The TYPE code also reveals how many slots the current packet will occupy. This allows the non-addressed receivers to refrain from listening to the channel for the duration of the remaining slots. In 8.4.4, each packet type will be described in more detail.

8.4.3.3 FLOW

This bit is used for flow control of packets over the ACL link. When the RX buffer for the ACL link in the recipient is full and is not emptied, a STOP indication (FLOW = 0) is returned to stop the transmission of data temporarily. Note, that the STOP signal only concerns ACL packets. Packets including only link control information (ID, POLL and NULL packets) or SCO packets can still be received. When the RX buffer is empty, a GO indication (FLOW = 1) is returned. When no packet is received, or the received header is in error, a GO is assumed implicitly. In this case, the slave can receive a new packet with CRC although its RX buffer is still not emptied. The slave shall then return a negative acknowledgment (NAK) in response to this packet even if the packet passed the CRC check.

8.4.3.4 ARQN

The 1-bit acknowledgment indication ARQN is used to inform the source of a successful transfer of payload data with CRC, and can be positive acknowledge ACK or negative acknowledge NAK. If the reception was successful, an ACK (ARQN = 1) is returned, otherwise a NAK (ARQN = 0) is returned. When no return message regarding acknowledge is received, a NAK is assumed implicitly. NAK is also the default return information.

The ARQN is piggy-backed in the header of the return packet. The success of the reception is checked by means of a cyclic redundancy check (CRC) code. An unnumbered ARQ scheme, which means that the ARQN relates to the latest received packet from the same source, is used. See 8.5.3 for initialization and usage of this bit.

8.4.3.5 SEQN

The SEQN bit provides a sequential numbering scheme to order the data packet stream. For each new transmitted packet that contains data with CRC, the SEQN bit is inverted. This is required to filter out retransmissions at the destination; if a retransmission occurs due to a failing ACK, the destination receives the same packet twice. By comparing the SEQN of consecutive packets, correctly received retransmissions can be discarded. See 8.5.3.2 for initialization and usage of the SEQN bit. For broadcast packets, a modified sequencing method is used, see 8.5.3.5.

8.4.3.6 HEC

Each header has a header-error-check (HEC) to check the header integrity. The HEC consists of an 8-bit word generated by the polynomial `647` (octal representation). Before generating the HEC, the HEC generator is initialized with an 8-bit value. For FHS packets sent in master page response state, the slave

Copyright © 2002 IEEE. All rights reserved.

LINKPLAY EXHIBIT 1008
Page 70 of 1169

upper address part (UAP) is used. For FHS packets sent in inquiry response, the default check initialization (DCI, see 8.5.4) is used. In all other cases, the UAP of the master device is used. For the definition of Bluetooth device addresses, see 8.13.1.

After the initialization, a HEC is calculated for the 10 header bits. Before checking the HEC, the receiver shall initialize the HEC check circuitry with the proper 8-bit UAP (or DCI). If the HEC does not check, the entire packet is disregarded. More information can be found in 8.5.4.

8.4.4 Packet types

The packets used on the piconet are related to the physical links they are used in. This standard has defined two physical links: the SCO link and the ACL link. For each of these links, 12 different packet types can be defined. Four control packets will be common to all link types: their TYPE code is unique irrespective of the link type.

To indicate the different packets on a link, the 4-bit TYPE code is used. The packet types have been divided into four segments. The first segment is reserved for the four control packets common to all physical link types; all four packet types have been defined. The second segment is reserved for packets occupying a single time slot; six packet types have been defined. The third segment is reserved for packets occupying three time slots; two packet types have been defined. The fourth segment is reserved for packets occupying five time slots; two packet types have been defined. The slot occupancy is reflected in the segmentation and can directly be derived from the type code. Table 13 summarizes the packets defined for the SCO and ACL link types.

Table 13—Packets defined for SCO and ACL link types

Segment	TYPE code $b_3b_2b_1b_0$	Slot occupancy	SCO link	ACL link
1	0000	1	NULL	NULL
	0001	1	POLL	POLL
	0010	1	FHS	FHS
	0011	1	DM1	DM1
2	0100	1	undefined	DH1
	0101	1	HV1	undefined
	0110	1	HV2	undefined
	0111	1	HV3	undefined
	1000	1	DV	undefined
	1001	1	undefined	AUX1
3	1010	3	undefined	DM3
	1011	3	undefined	DH3
	1100	3	undefined	undefined
	1101	3	undefined	undefined
4	1110	5	undefined	DM5
	1111	5	undefined	DH5

Copyright © 2002 IEEE. All rights reserved.
LINKPLAY EXHIBIT 1008
Page 71 of 1169

8.4.4.1 Common packet types

There are five packet types defined, which are independent of the underlying physical link type. Additionally, the ID packet type is defined which is used in the paging and inquiry procedures, described in 8.10.6 and 8.10.7, respectively. These packet types are defined as follows.

8.4.4.1.1 ID packet

The identity or ID packet consists of the device access code (DAC) or inquiry access code (IAC). It has a fixed length of 68 bits. It is a very robust packet since the receiver uses a bit correlator to match the received packet to the known bit sequence of the ID packet. The packet is used, for example, in paging, inquiry, and response routines.

8.4.4.1.2 NULL packet

The NULL packet has no payload and therefore consists of the channel access code and packet header only. Its total (fixed) length is 126 bits. The NULL packet is used to return link information to the source regarding the success of the previous transmission (ARQN), or the status of the RX buffer (FLOW). The NULL packet itself does not have to be acknowledged.

8.4.4.1.3 POLL packet

The POLL packet is very similar to the NULL packet. It does not have a payload either. In contrast to the NULL packet, it requires a confirmation from the recipient. It is not a part of the ARQ scheme. The POLL packet does not affect the ARQN and SEQN fields. Upon reception of a POLL packet the slave must respond with a packet. This return packet is an implicit acknowledgement of the POLL packet. This packet can be used by the master in a piconet to poll the slaves, which must then respond even if they do not have information to send.

8.4.4.1.4 FHS packet

The FHS packet is a special control packet revealing, among other things, the Bluetooth device address and the clock of the sender. The payload contains 144 information bits plus a 16-bit CRC code. The payload is coded with a rate 2/3 FEC, which brings the gross payload length to 240 bits. The FHS packet covers a single time slot.

Figure 20 illustrates the format and contents of the FHS payload. The payload consists of eleven fields (see Table 14). The FHS packet is used in page master response, inquiry response and in master slave switch. In page master response or master slave switch, it is retransmitted until its reception is acknowledged or a timeout has exceeded. In inquiry response, the FHS packet is not acknowledged. The FHS packet contains real-time clock information. Subsequent transmissions of the FHS packet contain updated clock information. Each transmission always contains the most recent native clock data from the sender. The FHS packet is used for frequency hop synchronization before the piconet channel has been established, or when an existing piconet changes to a new piconet. In the former case, the recipient has not been assigned an active member address yet, in which case the AM_ADDR field in the FHS packet header is set to all-zeroes; however, the FHS packet should not be considered as a broadcast packet. In the latter case the slave already has an AM_ADDR in the existing piconet, which is then used in the FHS packet header.

Copyright © 2002 IEEE. All rights reserved.

LINKPLAY EXHIBIT 1008
Page 72 of 1169

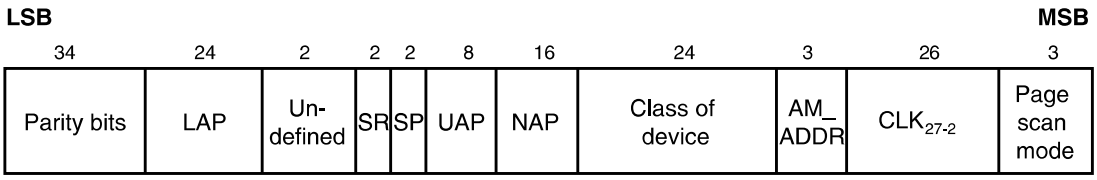


Figure 20—Format of the FHS payload

Each field is described in more detail below:

Table 14—Description of the FHS payload

Parity bits	This 34-bit field contains the parity bits that form the first part of the sync word of the access code of the unit that sends the FHS packet. These bits are derived from the LAP as described in 8.13.2.
LAP	This 24-bit field contains the lower address part of the unit that sends the FHS packet.
Reserved	This 2-bit field is reserved for future use and shall be set to zero.
SR	This 2-bit field is the scan repetition field and indicates the interval between two consecutive page scan windows, see also Table 15 and Table 23.
SP	This 2-bit field is the scan period field and indicates the period in which the mandatory page scan mode is applied after transmission of an inquiry response message, see also Table 16 and Table 28.
UAP	This 8-bit field contains the upper address part of the unit that sends the FHS packet.
NAP	This 16-bit field contains the non-significant address part of the unit that sends the FHS packet (see also 8.13.1 for LAP, UAP, and NAP).
Class of device	This 24-bit field contains the class of device of the unit that sends the FHS packet. The field is defined in 2.4.3.
AM_ADDR	This 3-bit field contains the member address the recipient shall use if the FHS packet is used at call setup or master-slave switch. A slave responding to a master or a unit responding to an inquiry request message shall include an all-zero AM_ADDR field if it sends the FHS packet.
CLK_{27-2}	This 26-bit field contains the value of the native system clock of the unit that sends the FHS packet, sampled at the beginning of the transmission of the access code of this FHS packet. This clock value has a resolution of 1.25ms (two-slot interval). For each new transmission, this field is updated so that it accurately reflects the real-time clock value.
Page scan mode	This 3-bit field indicates which scan mode is used by default by the sender of the FHS packet. The interpretation of the page scan mode is illustrated in Table 17. Currently, the standard supports one mandatory scan mode and up to three optional scan modes (see also Annex D).

Copyright © 2002 IEEE. All rights reserved.

Table 15—Contents of SR field

SR bit format b_1b_0	SR mode
00	R0
01	R1
10	R2
11	reserved

Table 16—Contents of SP field

SP bit format b_1b_0	SP mode
00	P0
01	P1
10	P2
11	reserved

Table 17—Contents of page scan mode field

Bit format $b_2b_1b_0$	Page scan mode
000	Mandatory scan mode
001	Optional scan mode I
010	Optional scan mode II
011	Optional scan mode III
100	Reserved for future use
101	Reserved for future use
110	Reserved for future use
111	Reserved for future use

The LAP, UAP, and NAP together form the 48-bit IEEE address of the unit that sends the FHS packet. Using the parity bits and the LAP, the recipient can directly construct the channel access code of the sender of the FHS packet.

Copyright © 2002 IEEE. All rights reserved.

LINKPLAY EXHIBIT 1008
Page 74 of 1169

8.4.4.1.5 DM1 packet

DM1 serves as part of segment 1 in order to support control messages in any link type. However, it can also carry regular user data. Since the DM1 packet is recognized on the SCO link, it can interrupt the synchronous information to send control information. Since the DM1 packet can be regarded as an ACL packet, it will be discussed in 8.4.4.3.

8.4.4.2 SCO packets

SCO packets are used on the synchronous SCO link. The packets do not include a CRC and are never retransmitted. SCO packets are routed to the synchronous I/O (voice) port. This standard has defined three pure SCO packets. In addition, an SCO packet is defined which carries an asynchronous data field in addition to a synchronous (voice) field. The SCO packets defined so far are typically used for 64 kb/s speech transmission.

8.4.4.2.1 HV1 packet

The HV1 packet carries 10 information bytes. The bytes are protected with a rate 1/3 FEC. No CRC is present. The payload length is fixed at 240 bits. There is no payload header present.

HV packets are used for transmission of voice and transparent synchronous data (see also 9.3.21.1). HV stands for High-quality Voice. The voice packets are never retransmitted and need no CRC. An HV1 packet carries 1.25 ms of speech at a 64 kb/s rate. An HV1 packet has to be sent every two time slots ($T_{SCO} = 2$).

8.4.4.2.2 HV2 packet

The HV2 packet carries 20 information bytes. The bytes are protected with a rate 2/3 FEC. No CRC is present. The payload length is fixed at 240 bits. There is no payload header present.

An HV2 packet carries 2.5 ms of speech at a 64 kb/s rate. An HV2 packet has to be sent every four time slots ($T_{SCO} = 4$).

8.4.4.2.3 HV3 packet

The HV3 packet carries 30 information bytes. The bytes are not protected by FEC. No CRC is present. The payload length is fixed at 240 bits. There is no payload header present.

An HV3 packet carries 3.75ms of speech at a 64 kb/s rate. An HV3 packet has to be sent every six time slots ($T_{SCO} = 6$).

8.4.4.2.4 DV packet

The DV packet is a combined data - voice packet. The payload is divided into a voice field of 80 bits and a data field containing up to 150 bits; see Figure 21. The voice field is not protected by FEC. The data field contains up to 10 information bytes (including the 1-byte payload header) and includes a 16-bit CRC. The data field is encoded with a rate 2/3 FEC. If necessary, extra zeroes are appended to assure that the total number of payload bits is a multiple of 10 prior to FEC encoding. Since the DV packet has to be sent at regular intervals due to its synchronous (voice) contents, it is listed under the SCO packet types. The voice and data fields are treated separately. The voice field is handled like normal SCO data and is never retransmitted; that is, the voice field is always new. The data field is checked for errors and is retransmitted if necessary.

Copyright © 2002 IEEE. All rights reserved.
LINKPLAY EXHIBIT 1008
Page 75 of 1169

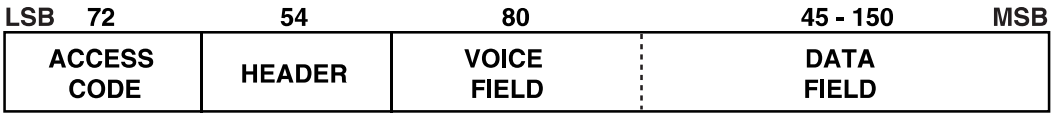


Figure 21—DV packet format

8.4.4.3 ACL packets

ACL packets are used on the asynchronous links. The information carried can be user data or control data. Including the DM1 packet, seven ACL packets have been defined. Six of the ACL packets contain a CRC code and retransmission is applied if no acknowledgement of proper reception is received (except in case a flush operation is carried out, see 8.5.3.3). The 7th ACL packet, the AUX1 packet, has no CRC and is not retransmitted.

8.4.4.3.1 DM1 packet

The DM1 packet is a packet that carries data information only. DM stands for Data – Medium rate. The payload contains up to 18 information bytes (including the 1-byte payload header) plus a 16-bit CRC code. The DM1 packet may cover up to a single time slot. The information plus CRC bits are coded with a rate 2/3 FEC which adds 5 parity bits to every 10-bit segment. If necessary, extra zeros are appended after the CRC bits to get the total number of bits (information bits, CRC bits, and tail bits) equal a multiple of 10. The payload header in the DM1 packet is only 1 byte long; see Figure 22. The length indicator in the payload header specifies the number of user bytes (excluding payload header and the CRC code).

8.4.4.3.2 DH1 packet

This packet is similar to the DM1 packet, except that the information in the payload is not FEC encoded. As a result, the DH1 frame can carry up to 28 information bytes (including the 1 byte payload header) plus a 16-bit CRC code. DH stands for Data – High rate. The DH1 packet may cover up to a single time slot.

8.4.4.3.3 DM3 packet

The DM3 packet is a DM1 packet with an extended payload. The DM3 packet may cover up to three time slots. The payload contains up to 123 information bytes (including the 2-byte payload header) plus a 16-bit CRC code. The payload header in the DM3 packet is 2 bytes long, see Figure 23. The length indicator in the payload header specifies the number of user bytes (excluding payload header and the CRC code). When a DM3 packet is sent or received, the radio frequency (RF) hop frequency shall not change for a duration of three time slots (the first time slot being the slot where the channel access code was transmitted).

8.4.4.3.4 DH3 packet

This packet is similar to the DM3 packet, except that the information in the payload is not FEC encoded. As a result, the DH3 packet can carry up to 185 information bytes (including the 2-byte payload header) plus a 16-bit CRC code. The DH3 packet may cover three time slots. When a DH3 packet is sent or received, the hop frequency shall not change for a duration of three time slots (the first time slot being the slot where the channel access code was transmitted).

8.4.4.3.5 DM5 packet

The DM5 packet is a DM1 packet with an extended payload. The DM5 packet may cover up to five time slots. The payload contains up to 226 information bytes (including the 2-byte payload header) plus a 16-bit

Copyright © 2002 IEEE. All rights reserved.

LINKPLAY EXHIBIT 1008
Page 76 of 1169

CRC code. The payload header in the DM5 packet is 2 bytes long. The length indicator in the payload header specifies the number of user bytes (excluding payload header and the CRC code). When a DM5 packet is sent or received, the hop frequency shall not change for a duration of five time slots (the first time slot being the slot where the channel access code was transmitted).

8.4.4.3.6 DH5 packet

This packet is similar to the DM5 packet, except that the information in the payload is not FEC encoded. As a result, the DH5 packet can carry up to 341 information bytes (including the two bytes payload header) plus a 16-bit CRC code. The DH5 packet may cover five time slots. When a DH5 packet is sent or received, the hop frequency shall not change for a duration of five time slots (the first time slot being the slot where the channel access code was transmitted).

8.4.4.3.7 AUX1 packet

This packet resembles a DH1 packet but has no CRC code. The AUX1 packet can carry up to 30 information bytes (including the 1-byte payload header). The AUX1 packet may cover up to a single time slot.

8.4.5 Payload Format

In the previous packet overview, several payload formats were considered. In the payload, two fields are distinguished: the (synchronous) voice field and the (asynchronous) data field. The ACL packets only have the data field and the SCO packets only have the voice field – with the exception of the DV packets, which have both.

8.4.5.1 Voice field

The voice field has a fixed length. For the HV packets, the voice field length is 240 bits; for the DV packet the voice field length is 80 bits. No payload header is present.

8.4.5.2 Data field

The data field consists of three segments: a payload header, a payload body, and possibly a CRC code (only the AUX1 packet does not carry a CRC code).

1) **Payload header**:

   Only data fields have a payload header. The payload header is 1 or 2 bytes long. Packets in segments one and two have a 1-byte payload header; packets in segments three and four have a 2-bytes payload header. The payload header specifies the logical channel (2-bit L_CH indication), controls the flow on the logical channels (1-bit FLOW indication), and has a payload length indicator (5 bits and 9 bits for 1-byte and 2-bytes payload header, respectively). In the case of a 2-bytes payload header, the length indicator is extended by four bits into the next byte. The remaining four bits of the second byte are reserved for future use and shall be set to zero. The formats of the 1-byte and 2-bytes payload headers are shown in Figure 22 and Figure 23.

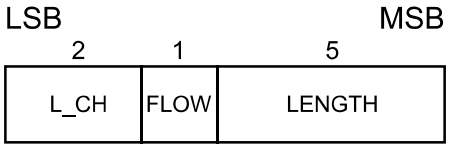


Figure 22—Payload header format for single-slot packets

Copyright © 2002 IEEE. All rights reserved.

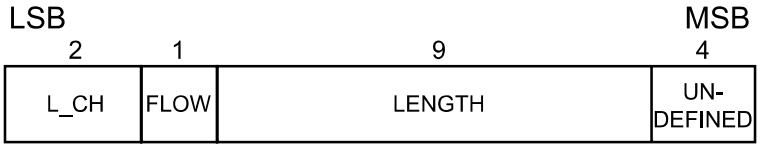


Figure 23—Payload header format for multislot packets

The L_CH field is transmitted first, the length field last. In Table 18, more details about the contents of the L_CH field are listed.

Table 18— Logical channel L_CH field contents

L_CH code b_1b_0	Logical Channel	Information
00	NA	undefined
01	UA/UI	Continuation fragment of an L2CAP message
10	UA/UI	Start of an L2CAP message or no fragmentation
11	LM	LMP message

An L2CAP message can be fragmented into several packets. Code 10 is used for an L2CAP packet carrying the first fragment of such a message; code 01 is used for continuing fragments. If there is no fragmentation, code 10 is used for every packet. Code 11 is used for LMP messages. Code 00 is reserved for future use.

The flow indicator in the payload is used to control the flow at the L2CAP level. It is used to control the flow per logical channel (when applicable). FLOW = 1 means flow-on (“OK to send") and FLOW = 0 means flow-off ("stop"). After a new connection has been established the flow indicator should be set to FLOW = 1. When a Bluetooth unit receives a payload header with the flow bit set to "stop" (FLOW = 0), it shall stop the transmission of ACL packets before an additional amount of payload data is sent. This amount can be defined as the flow control lag, expressed with a number of bytes. The shorter the flow control lag, the less buffering the other Bluetooth device may dedicate to this function. The flow control lag shall not exceed 1792 bytes (7*256 bytes), but in order to allow units to optimize the selection of packet length and buffer space, the flow control lag of a given implementation is provided in the LMP_features_res message (9.3.11).

If a packet containing the payload flow bit of "stop" is received with a valid packet header but bad payload, the payload flow control bit will not be recognized. The packet level ACK contained in the packet header will be received and a further ACL packet can be transmitted. Each occurrence of this situation allows a further ACL packet to be sent in spite of the flow control request being sent via the payload header flow control bit. It is recommended that Bluetooth units that use the payload header flow bit should ensure that no further ACL packets are sent until the payload flow bit has been correctly received. This can be accomplished by simultaneously turning on the flow bit in the packet header and keeping it on until an ACK is received back (ARQN = 1). This will typically be only one round trip time. Since they lack a payload CRC, AUX1 packets should not be used with a payload flow bit of "stop".

Copyright © 2002 IEEE. All rights reserved.

LINKPLAY EXHIBIT 1008
Page 78 of 1169

The link manager is responsible for setting and processing the flow bit in the payload header. Real-time flow control is carried out at the packet level by the link controller via the flow bit in the packet header (see 8.4.3.3). With the payload flow bit, traffic from the remote end can be controlled. It is allowed to generate and send an ACL packet with payload length zero irrespective of flow status. L2CAP start- and continue-fragment indications (L_CH = 10 and L_CH = 01) also retain their meaning when the payload length is equal to zero (i.e. an empty start-fragment should not be sent in the middle of an on-going L2CAP packet transmission). It is always safe to send an ACL packet with payload length = 0 and L_CH = 01. The payload flow bit has its own meaning for each logical channel (UA/I or LM); see Table 19. On the LM channel, no flow control is applied and the payload flow bit is always set at one.

Table 19—Use of payload header flow bit on the logical channels

L_CH code b_1b_0	Usage and semantics of the ACL payload header FLOW bit
00	Not defined, reserved for future use.
01 or 10	Flow control of the UA/I channels (which are used to send L2CAP messages)
11	Always set FLOW = 1 on transmission and ignore the bit on reception

The length indicator indicates the number of bytes (i.e. 8-bit words) in the payload excluding the payload header and the CRC code; i.e. the payload body only. With reference to Figure 22 and Figure 23, the MSB of the length field in a 1-byte header is the last (right-most) bit in the payload header; the MSB of the length field in a 2-byte header is the fourth bit (from left) of the second byte in the payload header.

2) **Payload body**:

The payload body includes the user host information and determines the effective user throughput. The length of the payload body is indicated in the length field of the payload header.

3) **CRC code generation**:

The 16-bit cyclic redundancy check code in the payload is generated by the CRC-CCITT polynomial `210041` (octal representation). It is generated in a way similar to the HEC. Before determining the CRC code, an 8-bit value is used to initialize the CRC generator. For the CRC code in the FHS packets sent in master page response state, the UAP of the slave is used. For the FHS packet sent in inquiry response state, the DCI is used (see 8.5.4). For all other packets, the UAP of the master is used.

The 8 bits are loaded into the 8 least significant (left-most) positions of the LFSR circuit (see Figure 33). The other 8 bits are at the same time reset to zero. Subsequently, the CRC code is calculated over the information. Then the CRC code is appended to the information; the UAP (or DCI) is disregarded. At the receive side, the CRC circuitry is in the same way initialized with the 8-bit UAP (DCI) before the received information is checked. More information can be found in 8.5.4.

Copyright © 2002 IEEE. All rights reserved.

8.4.6 Packet summary

A summary of the packets and their characteristics is shown in Table 20, Table 21, and Table 22. The user payload represents the packet payload excluding FEC, CRC, and payload header.

Table 20—Link control packets

Type	User payload (bytes)	FEC	CRC	Symmetric max. rate	Asymmetric max. rate
ID	na	na	na	na	na
NULL	na	na	na	na	na
POLL	na	na	na	na	na
FHS	18	2/3	yes	na	na

Table 21—ACL packets

Type	Payload header (bytes)	User payload (bytes)	FEC	CRC	Symmetric max. rate (kb/s)	Asymmetric max. rate (kb/s)	
						Forward	Reverse
DM1	1	0–17	2/3	yes	108.8	108.8	108.8
DH1	1	0–27	no	yes	172.8	172.8	172.8
DM3	2	0–121	2/3	yes	258.1	387.2	54.4
DH3	2	0–183	no	yes	390.4	585.6	86.4
DM5	2	0–224	2/3	yes	286.7	477.8	36.3
DH5	2	0–339	no	yes	433.9	723.2	57.6
AUX1	1	0–29	no	no	185.6	185.6	185.6

Table 22—SCO packets

Type	Payload Header (bytes)	User Payload (bytes)	FEC	CRC	Symmetric Max. Rate (kb/s)
HV1	na	10	1/3	no	64.0
HV2	na	20	2/3	no	64.0
HV3	na	30	no	no	64.0
DV[a]	1 D	10+(0–9) D	2/3 D	yes D	64.0+57.6 D

[a]Items followed by ‘D’ relate to data field only.

Copyright © 2002 IEEE. All rights reserved.

LINKPLAY EXHIBIT 1008
Page 80 of 1169

8.5 Error Correction

There are three error correction schemes defined for Bluetooth:

— 1/3 rate FEC

— 2/3 rate FEC

— ARQ scheme for the data

The purpose of the FEC scheme on the data payload is to reduce the number of retransmissions. However, in a reasonably error-free environment, FEC gives unnecessary overhead that reduces the throughput. Therefore, the packet definitions given in 8.4 have been kept flexible to use FEC in the payload or not, resulting in the DM and DH packets for the ACL link and the HV packets for the SCO link. The packet header is always protected by a 1/3 rate FEC; it contains valuable link information and should be able to sustain more bit errors.

Correction measures to mask errors in the voice decoder are not included in this clause. This matter is discussed in 8.12.3.

8.5.1 FEC Code: Rate 1/3

A simple 3-times repetition FEC code is used for the header. The repetition code is implemented by repeating the bit three times, see the illustration in Figure 24. The 3-bit repetition code is used for the entire header, and also for the voice field in the HV1 packet.

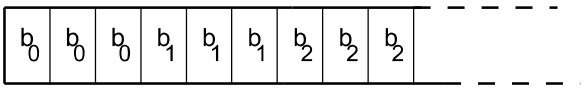


Figure 24—Bit-repetition encoding scheme

8.5.2 FEC Code: Rate 2/3

The other FEC scheme is a (15,10) shortened Hamming code. The generator polynomial is $g(D) = (D + 1)(D^4 + D + 1)$. This corresponds to 65 in octal notation. The LFSR generating this code is depicted in Figure 25. Initially all register elements are set to zero. The 10 information bits are sequentially fed into the LFSR with the switches S1 and S2 set in position 1. Then, after the final input bit, the switches S1 and S2 are set in position 2, and the five parity bits are shifted out. The parity bits are appended to the information bits. Consequently, each block of 10 information bits is encoded into a 15 bit codeword. This code can correct all single errors and detect all double errors in each codeword. This 2/3 rate FEC is used in the DM packets, in the data field of the DV packet, in the FHS packet, and in the HV2 packet. Since the encoder operates with information segments of length 10, tail bits with value zero may have to be appended after the CRC bits. The total number of bits to encode, i.e., payload header, user data, CRC, and tail bits, shall be a multiple of 10. Thus, the number of tail bits to append is the least possible that achieves this (i.e., in the interval 0...9). These tail bits are not included in the payload length indicator.

Copyright © 2002 IEEE. All rights reserved.

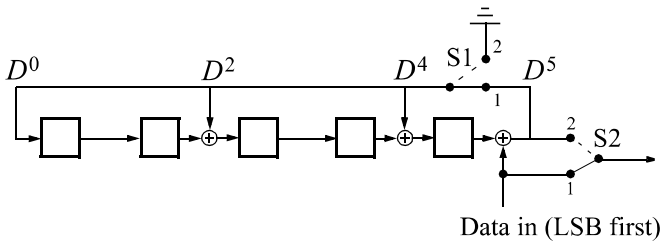


Figure 25—LFSR generating the (15,10) shortened Hamming code

8.5.3 ARQ Scheme

With an automatic repeat request scheme, DM, DH and the data field of DV packets are transmitted and retransmitted until acknowledgement of a successful reception is returned by the destination (or timeout is exceeded). The acknowledgement information is included in the header of the return packet, so-called piggy-backing. To determine whether the payload is correct or not, a cyclic redundancy check (CRC) code is added to the packet. The ARQ scheme only works on the payload in the packet (only that payload which has a CRC). The packet header and the voice payload are not protected by the ARQ scheme.

8.5.3.1 Unnumbered ARQ

Bluetooth uses a fast, unnumbered acknowledgment scheme: an ACK (ARQN = 1) or a NAK (ARQN = 0) is returned in response to the receipt of previously received packet. The slave will respond in the slave-to-master slot directly following the master-to-slave slot; the master will respond in its next transmission to the same slave. In a multislave piconet operation, between two successive transmissions from the master to the same slave, the master may transmit to other slaves as well. Hence, an ACK to a slave's transmission may not always be included in the master transmission that immediately follows the particular slave's transmission. For a packet reception to be successful, at least the HEC must check. In addition, the CRC must check if present.

At the start of a new connection which may be the result of a page, page scan, master-slave switch or unpark, the master sends a POLL packet to verify the connection. In this packet the master initializes the ARQN bit to NAK. The response packet sent by the slave also has the ARQN bit set to NAK. The subsequent packets use the following rules.

The ARQN bit is affected by data packets containing CRC and empty slots only. As shown in Figure 26, upon successful reception of a CRC packet, the ARQN bit is set to ACK. If, in any receive slot in the slave, or, in a receive slot in the master following transmission of a packet, one of these events applies:

1) no access code is detected
2) the HEC fails
3) the CRC fails

then the ARQN bit is set to NAK.

Packets that have correct HEC but that are addressed to other slaves, or packets other than DH, DM, or DV packets, do not affect the ARQN bit. In these cases the ARQN bit is left as it was prior to reception of the packet. If a CRC packet with a correct header has the same SEQN as the previously received CRC packet, the ARQN bit is set to ACK and the payload is disregarded without checking the CRC.

Copyright © 2002 IEEE. All rights reserved.

LINKPLAY EXHIBIT 1008
Page 82 of 1169

The ARQN bit in the FHS packet is not meaningful. Contents of the ARQN bit in the FHS packet should not be checked.

Broadcast packets are checked on errors using the CRC, but no ARQ scheme is applied. Broadcast packets are never acknowledged.

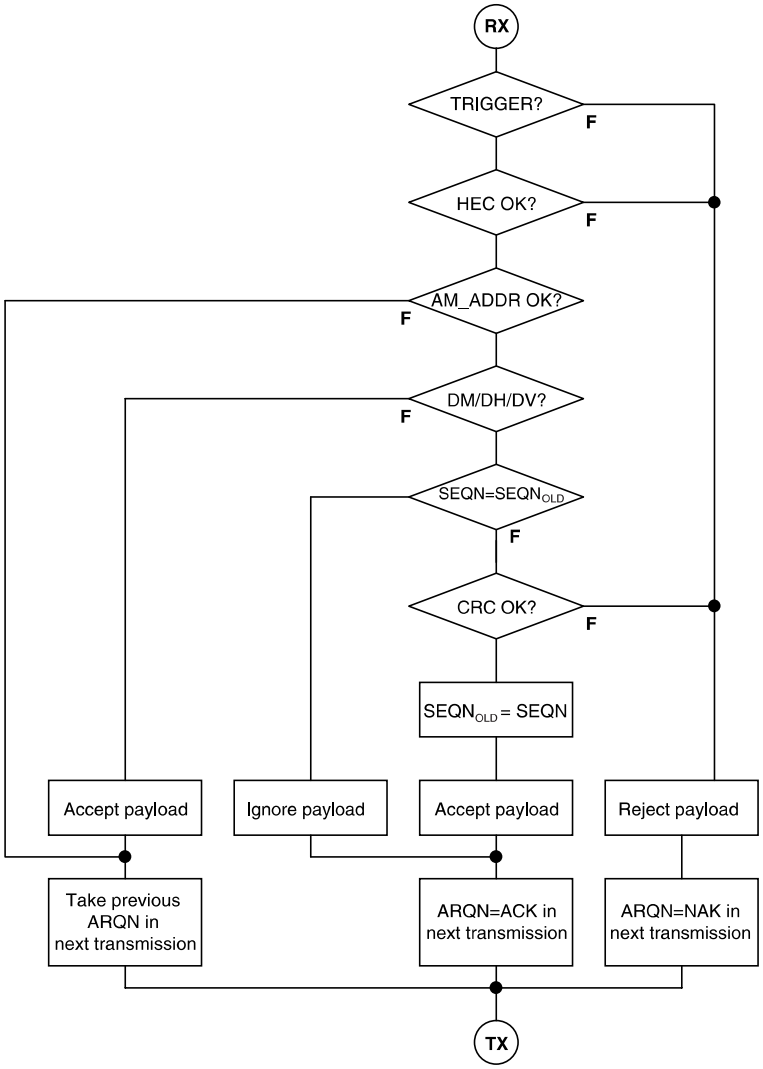


Figure 26—Receive protocol for determining the ARQN bit

8.5.3.2 Retransmit filtering

The data payload is retransmitted until a positive acknowledgment is received (or a timeout is exceeded). A retransmission is carried out either because the packet transmission itself failed, or because the piggy-backed acknowledgment transmitted in the return packet failed (note that the latter has a lower failure probability since the header is more heavily coded). In the latter case, the destination keeps receiving the same payload over and over again. In order to filter out the retransmissions in the destination, the SEQN bit is added in the header. Normally, this bit is alternated for every new CRC data payload transmission. In case of a retransmission, this bit is not changed. So the destination can compare the SEQN bit with the previous SEQN value. If different, a new data payload has arrived; otherwise it is the same data payload and can be discarded. Only new data payloads are transferred to the link manager. Note that CRC data payloads can be carried only by DM, DH, or DV packets.

Copyright © 2002 IEEE. All rights reserved.

At the start of a new connection which may be the result of a page, page scan, master slave switch or unpark, the master sends a POLL packet to verify the connection. The slave responds with a packet. The SEQN bit of the first CRC data packet, on both the master and the slave sides, is set to 1. The subsequent packets use the rules given below.

The SEQN bit is affected only by the CRC data packets as shown in Figure 27. It is inverted every time a new CRC data packet is sent. The CRC data packet is retransmitted with the same SEQN number until an ACK is received or the packet is flushed. When an ACK is received, the SEQN bit is inverted and a new payload is sent. When the packet is flushed (see 8.5.3.3), a new payload is sent. The SEQN bit is not necessarily inverted. However, if an ACK is received before the new packet is sent, the SEQN bit is inverted. This procedure prevents loss of the first packet of a message (after the flush command has been given) due to the retransmit filtering.

If a device decides to flush, and it has not received an acknowledgement for the current packet, it replaces the current packet with an ACL L2CAP continuation packet with length zero. It transmits this packet with the same sequence number as the packet it is trying to flush until it does get an ACK. Only then can it move on to the next packet. If a flush is needed for the next packet in the transmit queue before the zero-length packet has been transmitted, that next packet can be removed from the queue directly without it also being replaced by a zero-length packet. The described flushing procedure is considered optional, although strongly recommended.

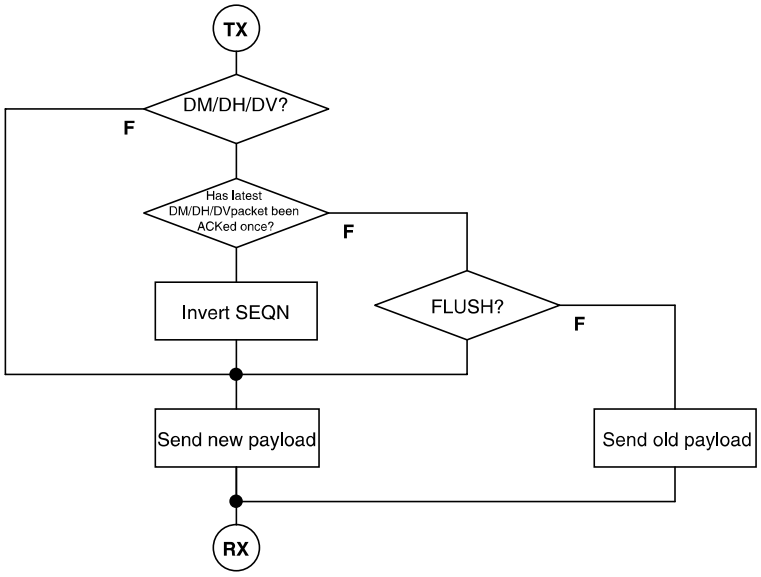


Figure 27—Retransmit filtering for packets with CRC

The SEQN bit in the FHS packet is not meaningful. This bit can be set to any value. Contents of the SEQN bit in the FHS packet should not be checked. During transmission of all other packets the SEQN bit remains the same as it was in the previous packet.

8.5.3.3 Flushing payloads

The ARQ scheme can cause variable delay in the traffic flow since retransmissions are inserted to assure error-free data transfer. For certain communication links, only a limited amount of delay is allowed: retransmissions are allowed up to a certain limit at which the current payload must be disregarded and the next payload must be considered. This data transfer is indicated as isochronous traffic. This means that the retransmit process shall be overruled in order to continue with the next data payload. Aborting the retransmit

Copyright © 2002 IEEE. All rights reserved.

LINKPLAY EXHIBIT 1008
Page 84 of 1169

scheme is accomplished by *flushing* the old data and forcing the Bluetooth controller to take the next data instead.

Flushing results in loss of remaining portions of an L2CAP message. Therefore, the packet following the flush will have a start packet indication of L_CH = 10 in the payload header for the next L2CAP message. This informs the destination of the flush (see 8.4.5). Flushing will not necessarily result in a change in the SEQN bit value (see 8.5.3.2).

8.5.3.4 Multislave considerations

In case of a piconet with multiple slaves, the master carries out the ARQ protocol independently to each slave.

8.5.3.5 Broadcast packets

Broadcast packets are packets transmitted by the master to all the slaves simultaneously. A broadcast packet is indicated by the all-zero AM_ADDR. (Note that the FHS packet is the only packet which may have an all-zero address but is not a broadcast packet.) Broadcast packets are not acknowledged (at least not at the LC level).

Since broadcast messages are not acknowledged, each broadcast packet is repeated for a fixed number of times. A broadcast packet is repeated N_{BC} times before the next broadcast packet of the same broadcast message is repeated (see Figure 28). However, time-critical broadcast information may abort the ongoing broadcast repetition train. For instance, unpark messages sent at beacon instances may do this (see 8.10.8.4.5).

Broadcast packets with a CRC have their own sequence number. The SEQN of the first broadcast packet with a CRC is set to SEQN = 1 by the master and it is inverted for each new broadcast packet with CRC thereafter. Broadcast packets without a CRC have no influence on the sequence number. The slave accepts the SEQN of the first broadcast packet it receives in a connection and checks for change in SEQN for consequent broadcast packets. Since there is no acknowledgement of broadcast messages and there is no end packet indication, it is important to receive the start packets correctly. To ensure this, repetitions of the broadcast packets that are L2CAP start packets and LMP packets will not be filtered out. These packets are indicated by L_CH=1X in the payload header as explained in 8.4.5. Only repetitions of the L2CAP continuation packets will be filtered out.

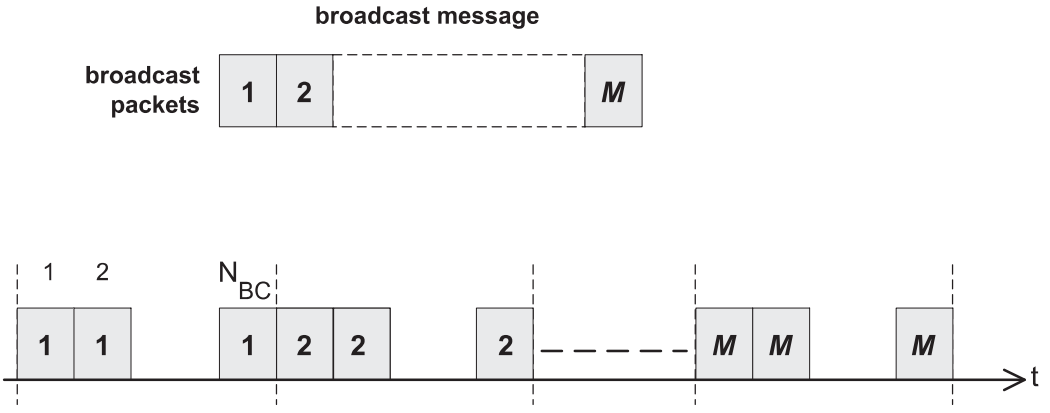


Figure 28—Broadcast repetition scheme

Copyright © 2002 IEEE. All rights reserved.

8.5.4 Error checking

Errors in packets or wrong delivery can be checked using the channel access code, the HEC in the header of a packet, and the CRC in payload of a packet. At packet reception, first the access code is checked. Since the 64-bit sync word in the channel access code is derived from the 24-bit master LAP, this checks if the LAP is correct, and prevents the receiver from accepting a packet of another piconet.

The HEC and CRC are used to check both on errors and on a wrong address: to increase the address space with 8 bits, the UAP is normally included in the HEC and CRC checks. Then, even when a packet with the same access code (i.e., an access code of a device owning the same LAP, but different UAP) passes the access code test, it will be discarded after the HEC and CRC tests when the UAP bits do not match. However, there is an exception where no common UAP is available in the transmitter and receiver. This is the case when the HEC and CRC are generated for the FHS packet in inquiry response state. In this case the default check initialization (DCI) value is used. The DCI is defined to be `0x00` (hexadecimal).

The generation and check of the HEC and CRC are summarized in Figure 31 and Figure 34. Before calculating the HEC or CRC, the shift registers in the HEC/CRC generators are initialized with the 8-bit UAP (or DCI) value. Then the header and payload information is shifted into the HEC and CRC generators, respectively (with the LSB first).

The HEC generating LFSR is depicted in Figure 79. The generator polynomial is $g(D) = (D+1)(D^7 + D^4 + D^3 + D^2 + 1) = D^8 + D^7 + D^5 + D^2 + D + 1$. Initially this circuit is preloaded with the 8-bit UAP such that the LSB of the UAP (denoted UAP_0) goes to the left-most shift register element, and, UAP_7 goes to the right-most element. The initial state of the HEC LFSR is depicted in Figure 30. Then the data is shifted in with the switch S set in position 1. When the last data bit has been clocked into the LFSR, the switch S is set in position 2, and, the HEC can be read out from the register. The LFSR bits are read out from right to left (i.e., the bit in position 7 is the first to be transmitted, followed by the bit in position 6, etc.).

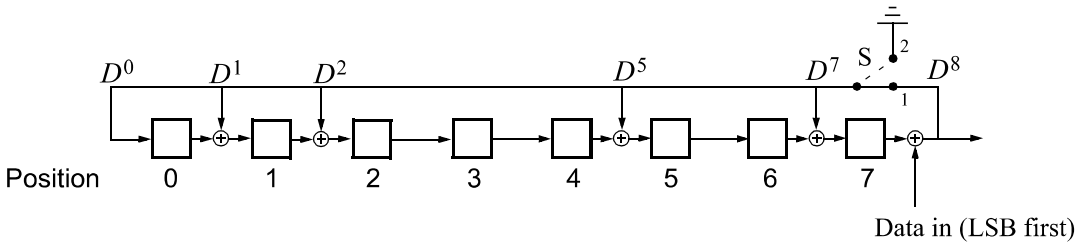


Figure 29—LFSR circuit generating the HEC

Position:	0	1	2	3	4	5	6	7
LFSR:	UAP_0	UAP_1	UAP_2	UAP_3	UAP_4	UAP_5	UAP_6	UAP_7

Figure 30—Initial state of the HEC generating circuit

Copyright © 2002 IEEE. All rights reserved.

LINKPLAY EXHIBIT 1008
Page 86 of 1169

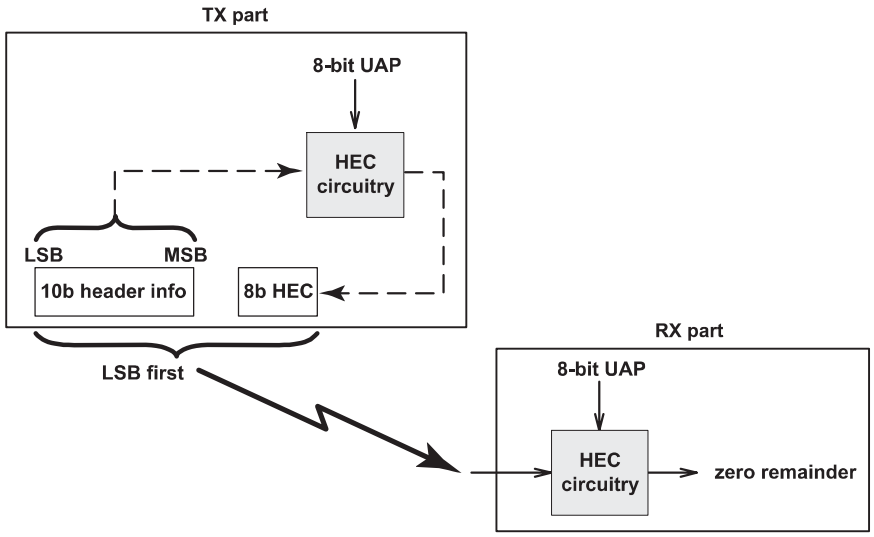


Figure 31—HEC generation and checking

The 16-bit LFSR for the CRC is constructed similarly using the CRC-CCITT generator polynomial $g(D) = D^{16} + D^{12} + D^5 + 1$ (see Figure 32). For this case, the 8 left-most bits are initially loaded with the 8-bit UAP (UAP_0 to the left and UAP_7 to the right) while the 8 right-most bits are reset to zero. The initial state of the 16 bit LFSR is depicted in Figure 33. The switch S is set in position 1 while the data is shifted in. After the last bit has entered the LFSR, the switch is set in position 2, and, the register's contents are transmitted, from right to left (i.e., starting with position 15, then position 14, etc.).

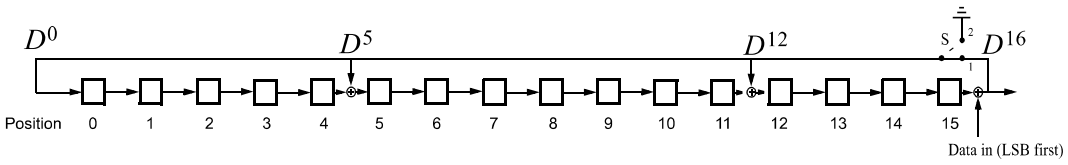


Figure 32—LFSR circuit generating the CRC

Position:	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
LFSR:	UAP_0	UAP_1	UAP_2	UAP_3	UAP_4	UAP_5	UAP_6	UAP_7	0	0	0	0	0	0	0	0

Figure 33—Initial state of the CRC generating circuit

Copyright © 2002 IEEE. All rights reserved.
LINKPLAY EXHIBIT 1008
Page 87 of 1169

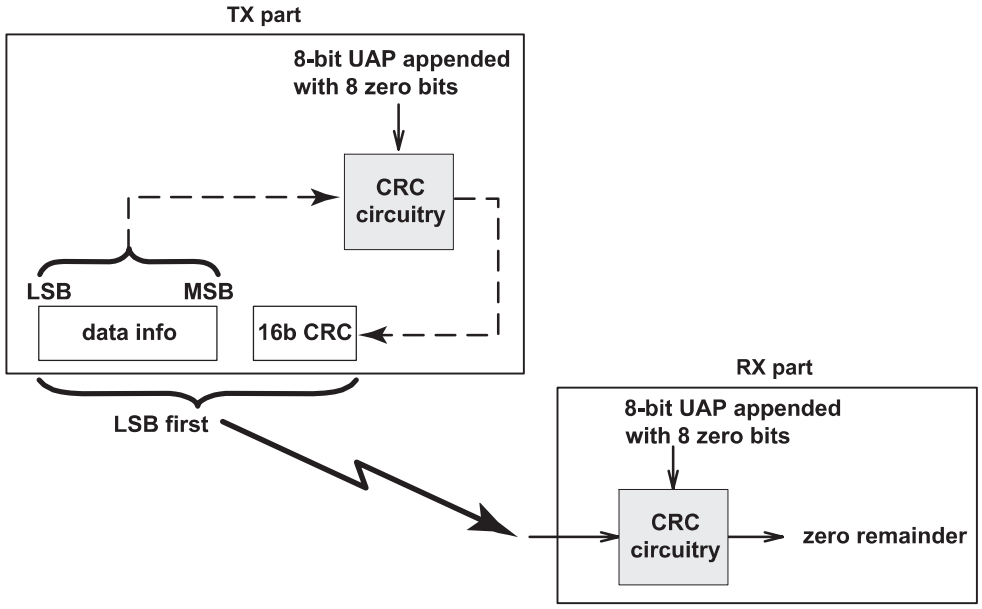


Figure 34—CRC generation and checking

8.6 Logical channels

In the Bluetooth system, five logical channels are defined:

- LC control channel
- LM control channel
- UA user channel
- UI user channel
- US user channel

The control channels LC and LM are used at the link control level and link manager level, respectively. The user channels UA, UI, and US are used to carry asynchronous, isochronous, and synchronous user information, respectively. The LC channel is carried in the packet header; all other channels are carried in the packet payload. The LM, UA, and UI channels are indicated in the L_CH field in the payload header. The US channel is carried by the SCO link only. The UA and UI channels are normally carried by the ACL link; however, they can also be carried by the data in the DV packet on the SCO link. The LM channel can be carried either by the SCO or the ACL link.

8.6.1 LC channel (Link control)

The LC channel is mapped onto the packet header. This channel carries low-level link control information like ARQ, flow control, and payload characterization. The LC channel is carried in every packet except in the **ID** packet which has no packet header.

Copyright © 2002 IEEE. All rights reserved.

LINKPLAY EXHIBIT 1008
Page 88 of 1169

8.6.2 LM channel (Link manager)

The LM control channel carries control information exchanged between the link managers of the master and the slave(s). Typically, the LM channel uses protected DM packets. The LM channel is indicated by the L_CH code 11 in the payload header.

8.6.3 UA/UI channel (User asynchronous/Isochronous data)

The UA channel carries L2CAP transparent asynchronous user data. This data may be transmitted in one or more baseband packets. For fragmented messages, the start packet uses an L_CH code of 10 in the payload header. Remaining continuation packets use L_CH code 01. If there is no fragmentation, all packets use the L2CAP start code 10.

Isochronous data channel is supported by timing start packets properly at higher levels. At the baseband level, the L_CH code usage is the same as the UA channel.

8.6.4 US Channel (User Synchronous data)

The US channel carries transparent synchronous user data. This channel is carried over the SCO link.

8.6.5 Channel mapping

The LC channel is mapped onto the packet header. All other channels are mapped onto the payload. The US channel can only be mapped onto the SCO packets. All other channels are mapped on the ACL packets, or possibly the SCO DV packet. The LM, UA, and UI channels may interrupt the US channel if it concerns information of higher priority.

8.7 Data whitening

Before transmission, both the header and the payload are scrambled with a data whitening word in order to randomize the data from highly redundant patterns and to reduce DC bias in the packet. The scrambling is performed prior to the FEC encoding.

At the receiver, the received data is descrambled using the same whitening word generated in the recipient. The descrambling is performed after FEC decoding.

The whitening word is generated with the polynomial $g(D) = D^7 + D^4 + 1$ (i.e., 221 in octal representation) and is subsequently EXORed with the header and the payload. The whitening word is generated with the linear feedback shift register shown in Figure 35. Before each transmission, the shift register is initialized with a portion of the master Bluetooth clock, $CLK_{6\text{-}1}$, extended with an MSB of value one. This initialization is carried out with CLK_1 written to position 0, CLK_2 written to position 1, etc. An exception exists when forming the FHS packet sent during frequency hop acquisition. In this case, initialization of the whitening register is carried out differently. Instead of the master clock, the X-input used in the inquiry or page response (depending on current state) routine is used; see Table 31 and Table 32 for the 79-hop and 23-hop systems, respectively. In case of a 79-hop system, the 5-bit values is extended with two MSBs of value one. In case of a 23-hop system, the 4-bit value is extended with three bits; the two MSBs are set to one and the third most significant bit is set to zero. During register initialization, the LSB of X (i.e., X_0) is written to position 0, X_1 is written to position 1, etc.

After initialization, the packet header and the payload (including the CRC) are scrambled. The payload whitening continues from the state the whitening LFSR had at the end of HEC. There is no re-initialization of the shift register between packet header and payload. The first bit of the “Data In” sequence is the LSB of the packet header.

Copyright © 2002 IEEE. All rights reserved.
LINKPLAY EXHIBIT 1008
Page 89 of 1169

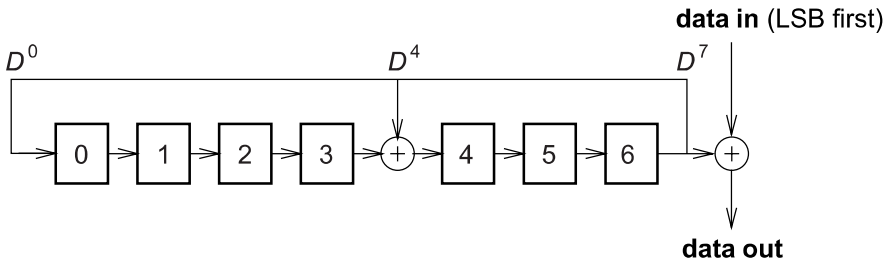


Figure 35—Data whitening LFSR

8.8 Transmit/Receive routines

This subclause describes the way to use the packets as defined in 8.4 in order to support the traffic on the ACL and SCO links. Both single-slave and multi-slave configurations are considered. In addition, the use of buffers for the TX and RX routines are described.

Note that the TX and RX routines described in 8.8.1 and 8.8.2 are of an informative character only. The final implementation may be carried out differently.

8.8.1 TX routine

The TX routine is carried out separately for each ACL link and each SCO link. Figure 36 shows the ACL and SCO buffers as used in the TX routine. In this figure, only a single TX ACL buffer and a single TX SCO buffer are shown. In the master, there is a separate TX ACL buffer for each slave. In addition there may be one or more TX SCO buffers for each SCO slave (different SCO links may either reuse the same TX SCO buffer, or each have their own TX SCO buffer). Each TX buffer consists of two FIFO registers: one current register which can be accessed and read by the Bluetooth controller in order to compose the packets, and one next register that can be accessed by the Bluetooth Link Manager to load new information. The positions of the switches S1 and S2 determine which register is current and which register is next; the switches are controlled by the Bluetooth Link Controller. The switches at the input and the output of the FIFO registers can never be connected to the same register simultaneously.

Of the packets common on the ACL and SCO links (ID, NULL, POLL, FHS, DM1) only the DM1 packet carries a payload that is exchanged between the Link Controller and the Link Manager; this common packet makes use of the ACL buffer. All ACL packets make use of the ACL buffer. All SCO packets make use of the SCO buffer except for the DV packet where the voice part is handled by the SCO buffer and the data part is handled by the ACL buffer. In 8.8.1.1 through 8.8.1.3, the operation for ACL traffic, SCO traffic, and combined data-voice traffic on the SCO link will be considered.

Copyright © 2002 IEEE. All rights reserved.

LINKPLAY EXHIBIT 1008
Page 90 of 1169

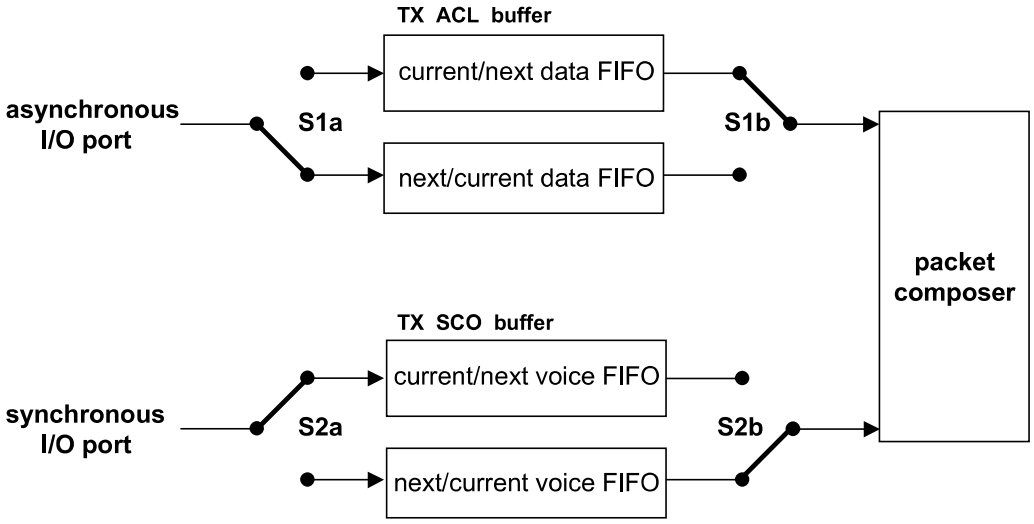


Figure 36—Functional diagram of TX buffering

8.8.1.1 ACL traffic

In the case of pure (asynchronous) data, only the TX ACL buffer in Figure 36 has to be considered. In this case, only packet types DM or DH are used, and these can have different lengths. The length is indicated in the payload header. The selection of high-rate data or medium-rate data shall depend on the quality of the link. When the quality is good, the FEC in the data payload can be omitted, resulting in a DH packet. Otherwise, DM packets must be used.

The default TYPE in pure data traffic is NULL. This means that, if there is no data to be sent (the data traffic is asynchronous, and therefore pauses occur in which no data is available) or no slaves need to be polled, NULL packets are sent instead – in order to send link control information to the other Bluetooth unit (e.g. ACK/STOP information for received data). When no link control information is available either (no need to acknowledge and/or no need to stop the RX flow) no packet is sent at all.

The TX routine works as follows. The Bluetooth Link Manager loads new data information in the register to which the switch S1a points. Next, it gives a flush command to the Bluetooth Link Controller, which forces the switch S1 to change (both S1a and S1b switch in synchrony). When the payload needs to be sent, the packet composer reads the current register and, depending on the packet TYPE, builds a payload which is appended to the channel access code and the header and is subsequently transmitted. In the response packet (which arrives in the following RX slot if it concerned a master transmission, or may be postponed until some later RX slot if it concerned a slave transmission), the result of the transmission is reported back. In case of an ACK, the switch S1 changes position; if a NAK (explicit or implicit) is received instead, the switch S1 will not change position. In that case, the same payload is retransmitted at the next TX occasion.

As long as the Link Manager keeps loading the registers with new information, the Bluetooth Link Controller will automatically transmit the payload; in addition, retransmissions are performed automatically in case of errors. The Link Controller will send NULL or nothing when no new data is loaded. If no new data has been loaded in the next register, during the last transmission, the packet composer will be pointing to an empty register after the last transmission has been acknowledged and the next register becomes the current register. If new data is loaded in the next register, a flush command is required to switch the S1 switch to the proper register. As long as the Link Manager keeps loading the data and type registers before each TX slot, the data is automatically processed by the Link Controller since the S1 switch is controlled by the ACK information received in response. However, if the traffic from the Link Manager is interrupted once and a default packet is sent instead, a flush command is required to continue the flow in the Link Controller.

Copyright © 2002 IEEE. All rights reserved.

The flush command can also be used in case of time-bounded (isochronous) data. In case of a bad link, many retransmission are necessary. In certain applications, the data is time-bounded: if a payload is retransmitted all the time because of link errors, it may become outdated, and the system might decide to continue with more recent data instead and skip the payload that does not come through. This is accomplished by the flush command as well. With the flush, the switch S1 is forced to change and the Link Controller is forced to consider the next data payload and overrules the ACK control.

8.8.1.2 SCO traffic

In case of an SCO link, we only use HV packet types. The synchronous port continuously loads the next register in the SCO buffer. The S2 switches are changed according to the T_{sco} interval. This T_{sco} interval is negotiated between the master and the slave at the time the SCO link is established.

For each new SCO slot, the packet composer reads the current register after which the S2 switch is changed. If the SCO slot has to be used to send control information with high priority concerning a control packet between the master and the considered SCO slave, or a control packet between the master and any other slave, the packet composer will discard the SCO information and use the control information instead. This control information shall be sent in a DM1 packet. Data or link control information can also be exchanged between the master and the SCO slave by using the DV or DM1 packets. Any ACL type of packet can be used to sent data or link control information to any other ACL slave. This is discussed in 8.8.1.3.

8.8.1.3 Mixed data/voice traffic

In 8.4.4.2, a DV packet has been defined that can support both data and voice simultaneously on a single SCO link. When the TYPE is DV, the Link Controller reads the data register to fill the data field and the voice register to fill the voice field. Thereafter, the switch S2 is changed. However, the position of S1 depends on the result of the transmission like on the ACL link: only if an ACK has been received will the S1 switch change its position. In each DV packet, the voice information is new, but the data information might be retransmitted if the previous transmission failed. If there is no data to be sent, the SCO link will automatically change from DV packet type to the current HV packet type used before the mixed data/voice transmission. Note that a flush command is required when the data stream has been interrupted and new data has arrived.

Combined data-voice transmission can also be accomplished by using separate ACL links in addition to the SCO link(s) if channel capacity permits this.

8.8.1.4 Default packet types

On the ACL links, the default type is always NULL both for the master and the slave. This means that if no user information needs to be sent, either a NULL packet is sent if there is ACK or STOP information, or no packet is sent at all. The NULL packet can be used by the master to allocate the next slave-to-master slot to a certain slave (namely the one addressed). However, the slave is not forced to respond to the NULL packet from the master. If the master requires a response, it has to send a POLL packet.

The SCO packet type is negotiated at the LM level when the SCO link is established. The agreed packet type is also the default packet type for the SCO slots.

8.8.2 RX routine

The RX routine is carried out separately for the ACL link and the SCO link. However, in contrast to the master TX ACL buffer, a single RX buffer is shared among all slaves. For the SCO buffer, it depends how the different SCO links are distinguished whether extra SCO buffers are required or not. Figure 37 shows the ACL and SCO buffers as used in the RX routine. The RX ACL buffer consists of two FIFO registers: one register that can be accessed and loaded by the Bluetooth Link Controller with the payload of the latest RX

Copyright © 2002 IEEE. All rights reserved.

LINKPLAY EXHIBIT 1008
Page 92 of 1169

packet, and one register that can be accessed by the Bluetooth Link Manager to read the previous payload. The RX SCO buffer also consists of two FIFO registers: one register that is filled with newly arrived voice information, and one register that can be read by the voice processing unit.

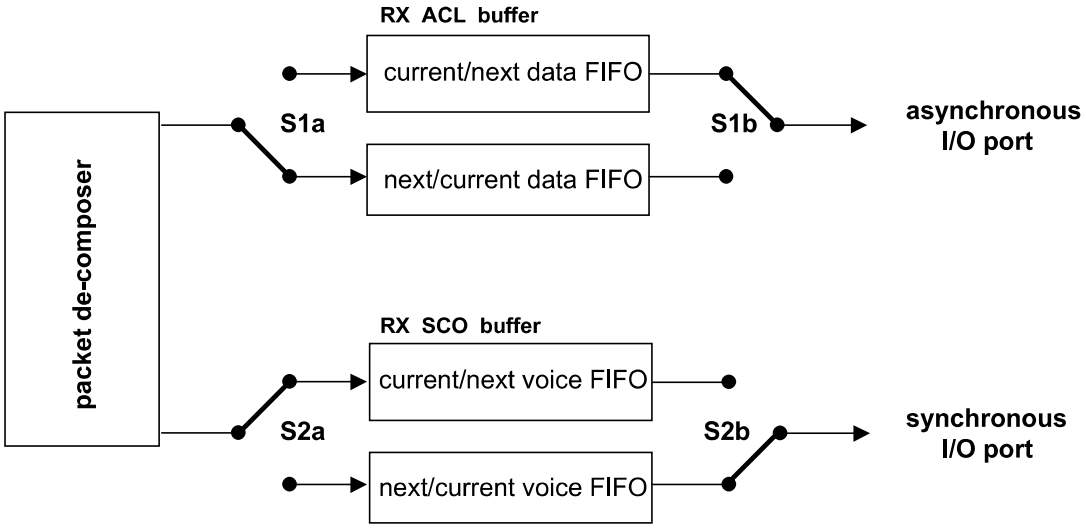


Figure 37—Functional diagram of RX buffering

Since the TYPE indication in the header of the received packet indicates whether the payload contains data and/or voice, the packet decomposer can automatically direct the traffic to the proper buffers. The switch S1 changes every time the Link Manager has read the old register. If the next payload arrives before the RX register is emptied, a STOP indication must be included in the packet header of the next TX packet that is returned. The STOP indication is removed again as soon as the RX register is emptied. The SEQN field is checked before a new ACL payload is stored into the ACL register (flush indication in L_CH and broadcast messages influence the interpretation of the SEQN field see 8.5.3).

The S2 switch is changed every T_{SCO}. If – due to errors in the header – no new voice payload arrives, the switch still changes. The voice processing unit then has to process the voice signal to account for the missing speech parts.

8.8.3 Flow control

Since the RX ACL buffer can be full while a new payload arrives, flow control is required. As was mentioned in 8.4.3.3, the header field FLOW in the return TX packet can use STOP or GO in order to control the transmission of new data.

8.8.3.1 Destination control

As long as data cannot be received, a STOP indication is transmitted which is automatically insertbed by the Link Controller into the header of the return packet. STOP is returned as long as the RX ACL buffer is not emptied by the Link Manager. When new data can be accepted again, the GO indication is returned. GO is the default value. Note that all packet types not including data can still be received. Voice communication, for example, is not affected by the flow control. Also note that although a Bluetooth unit cannot receive new information, it can still continue to transmit information: the flow control is separate for each direction.

Copyright © 2002 IEEE. All rights reserved.

8.8.3.2 Source control

On the reception of a STOP signal, the Link Controller will automatically switch to the default packet type. The ACL packet transmitted just before the reception of the STOP indication is kept until a GO signal is received. It is retransmitted as soon as a GO indication is received. Default packets are sent as long as the STOP indication is received. When no packet is received, GO is assumed implicitly. Note that the default packets contain link control information (in the header) for the receive direction (which may still be open) and may contain voice (HV packets). When a GO indication is received, the Link Controller resumes to transmit the data as is present in the TX ACL buffers.

In a multi-slave configuration, only the transmission to the slave that issued the STOP signal is stalled. This means that the previously described routine implemented in the master only concerns the TX ACL buffer that corresponds to the slave that cannot accept data momentarily.

8.8.4 Bitstream processes

Before the user information is sent over the air interface, several bit manipulations are performed in the transmitter to increase reliability and security. To the packet header, an HEC is added, the header bits are scrambled with a whitening word, and FEC coding is applied. In the receiver, the inverse processes are carried out. Figure 38 shows the processes carried out for the packet header both at the transmit and the receive side. All header bit processes are mandatory.

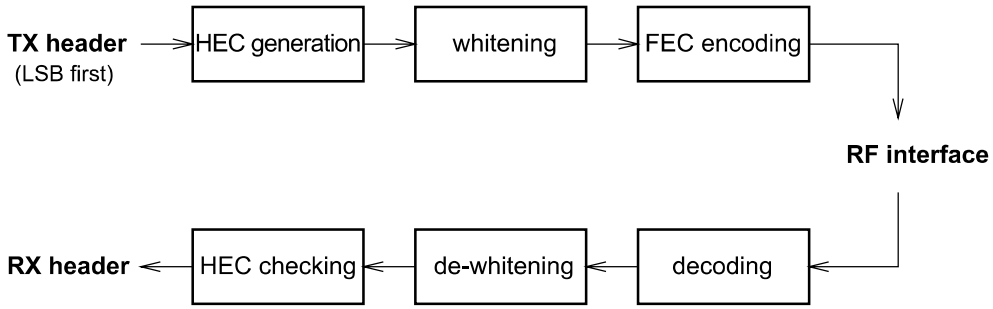


Figure 38—Header bit processes

For the payload, similar processes are performed. It depends on the packet type, which processes are carried out. Figure 39 shows the processes that may be carried out on the payload. In addition to the processes defined for the packet header, encryption can be applied on the payload. Only whitening and de-whitening, as explained in 8.7, are mandatory for every payload; all other processes are optional and depend on the packet type and the mode enabled. In Figure 39, optional processes are indicated by dashed blocks.

Copyright © 2002 IEEE. All rights reserved.

LINKPLAY EXHIBIT 1008
Page 94 of 1169

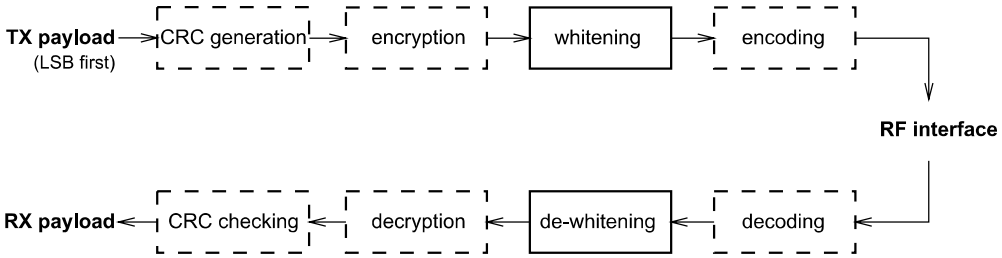


Figure 39—Payload bit processes

8.9 Transmit/receive timing

The Bluetooth transceiver applies a time-division duplex (TDD) scheme. This means that it alternately transmits and receives in a synchronous manner. It depends on the mode of the Bluetooth unit what the exact timing of the TDD scheme is. In the normal connection mode, *the master transmission shall always start at even numbered time slots (master CLK1 = 0) and the slave transmission shall always start at odd numbered time slots (master CLK1 = 1).* Due to packet types that cover more than a single slot, master transmission may continue in odd numbered slots and slave transmission may continue in even numbered slots.

All timing diagrams shown in this subclause are based on the signals as present at the antenna. The term "exact" when used to describe timing refers to an ideal transmission or reception and neglects timing jitter and clock frequency imperfections.

The average timing of master packet transmission shall not drift faster than 20 ppm relative to the ideal slot timing of 625 µs. The instantaneous timing shall not deviate more than 1 µs from the average timing. Thus, the absolute packet transmission timing t_k of slot boundary k shall fulfill the equation:

$$t_k = \left(\sum_{i=1}^{k} (1 + d_i) T_N \right) + j_k + \text{offset}, \quad (1)$$

where T_N is the nominal slot length (625 µs), j_k denotes jitter ($|j_k| \leq 1$ µs) at slot boundary k, and, d_k, denotes the drift ($|d_k| \leq 20$ ppm) within slot k. The jitter and drift may vary arbitrarily within the given limits for every slot, while "offset" is an arbitrary but fixed constant. For hold, park, and sniff mode the drift and jitter parameters as described in 9.3.9 apply.

8.9.1 Master/slave timing synchronization

The piconet is synchronized by the system clock of the master. The master never adjusts its system clock during the existence of the piconet. The slaves adapt their native clocks with a timing offset in order to match the master clock. This offset is updated each time a packet is received from the master: by comparing the exact RX timing of the received packet with the estimated RX timing, the slaves correct the offset for any timing misalignments. Note that the slave RX timing can be corrected with any packet sent in the master-to-slave slot, since only the channel access code is required to synchronize the slave.

The slave TX timing shall be based on the most recent slave RX timing. The RX timing is based on the latest successful trigger during a master-to-slave slot. For ACL links, this trigger must have occurred in the master-to-slave slot directly preceding the current slave transmission; for SCO links, the trigger may have

Copyright © 2002 IEEE. All rights reserved.

occurred several master-to-slave slots before since a slave is allowed to send an SCO packet even if no packet was received in the preceding master-to-slave slot. The slave shall be able to receive the packets and adjust the RX timing as long as the timing mismatch remains within the ±10 µs uncertainty window.

The master TX timing is strictly related to the master clock. The master shall keep an exact interval of $M \times 1250$ µs (where M is a positive integer larger than 0) between the start of successive transmissions; the RX timing is based on this TX timing with a shift of exactly $N \times 625$ µs (where N is an odd, positive integer larger than 0). During the master RX cycle, the master will also use the ±10 µs uncertainty window to allow for slave misalignments. The master will adjust the RX processing of the considered packet accordingly, but will not adjust its RX/TX timing for the following TX and RX cycles. During periods when an active slave is not able to receive any valid channel access codes from the master, the slave may increase its receive uncertainty window and/or use predicted timing drift to increase the probability of receiving the master's bursts when reception resumes.

Timing behavior may differ slightly depending on the current state of the unit. The different states are described in 8.9.2 through 8.9.5.

8.9.2 Connection state

In the connection mode, the Bluetooth transceiver transmits and receives alternately (see Figure 40 and Figure 41). In the figures, only single-slot packets are shown as an example. Depending on the type and the payload length, the packet size can be up to 366 µs. Each RX and TX transmission is at a different hop frequency. For multislot packets, several slots are covered by the same packet, and the hop frequency used in the first slot will be used throughout the transmission.

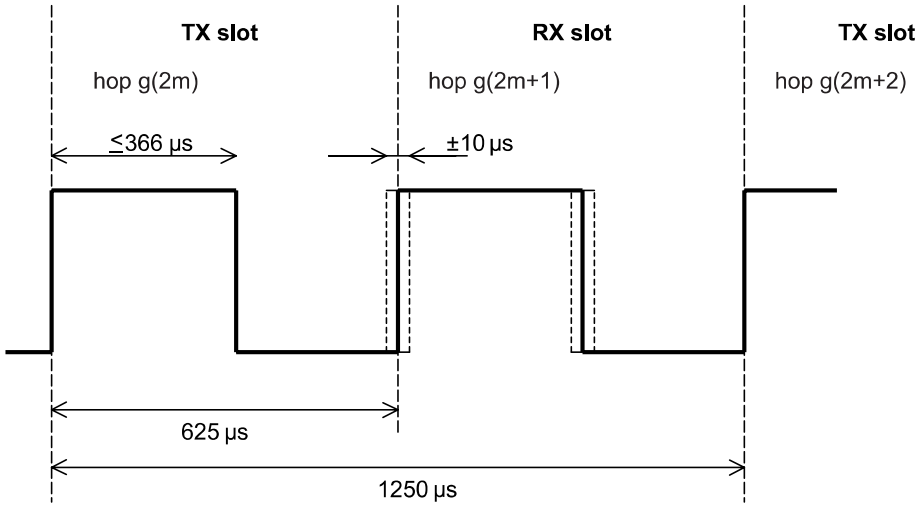


Figure 40—RX/TX cycle of Bluetooth master transceiver in normal mode for single-slot packets

Copyright © 2002 IEEE. All rights reserved.

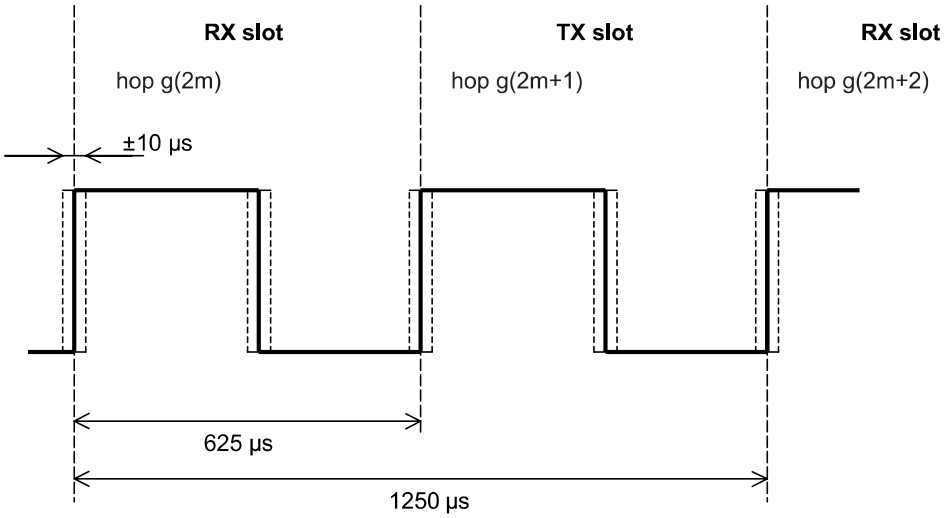


Figure 41—RX/TX cycle of Bluetooth slave transceiver in normal mode for single-slot packets

The master TX/RX timing is shown in Figure 40. In Figure 40 through Figure 45, $f'(k)$ is used for the frequencies of the page hopping sequence and $f'(k)$ denotes the corresponding page response sequence frequencies. The channel hopping frequencies are indicated by *g(m)*. After transmission, a return packet is expected $N \times 625$ µs after the start of the TX burst where N is an odd, positive integer. N depends on the type of the transmitted packet. To allow for some time slipping, an uncertainty window is defined around the exact receive timing. During normal operation, the window length is 20 µs, which allows the RX burst to arrive up to 10 µs too early or 10 µs too late. During the beginning of the RX cycle, the access correlator searches for the correct channel access code over the uncertainty window. If no trigger event occurs, the receiver goes to sleep until the next RX event. If in the course of the search, it becomes apparent that the correlation output will never exceed the final threshold, the receiver may go to sleep earlier. If a trigger event does occur, the receiver remains open to receive the rest of the packet.

The current master transmission is based on the previous master transmission: it is scheduled $M \times 1250$ µs after the start of the previous master TX burst where M depends on the transmitted and received packet type. Note that the master TX timing is not affected by time drifts in the slave(s). If no transmission takes place during a number of consecutive slots, the master will take the TX timing of the latest TX burst as reference.

The slave's transmission is scheduled $N \times 625$ µs after the start of the slave's RX burst. If the slave's RX timing drifts, so will its TX timing. If no reception takes place during a number of consecutive slots, the slave will take the RX timing of the latest RX burst as reference.

8.9.3 Return from hold mode

In the connection state, the Bluetooth unit can be placed in a hold mode (see 8.10.8). In the hold mode, a Bluetooth transceiver neither transmits nor receives information. When returning to the normal operation after a hold mode in a slave Bluetooth unit, the slave must listen for the master before it may send information. In that case, the length of the search window in the slave unit may be increased from 20 µs to a larger value X µs as illustrated in Figure 42. Note that only RX hop frequencies are used: the hop frequency used in the master-to-slave (RX) slot is also used in the uncertainty window extended into the preceding time interval normally used for the slave-to-master (TX) slot.

If the length of search window (X) exceeds 1250 µs, consecutive windows shall not be centered at the start of RX hops $g(2m)$, $g(2m + 2)$, ... $g(2m + 2i)$ (where "i" is an integer) to avoid overlapping search windows.

Copyright © 2002 IEEE. All rights reserved.

Consecutive windows should instead be centred at $g(2m)$, $g(2m + 4)$, ... $g(2m + 4i)$, which gives a maximum value $X = 2500$ µs, or even at $g(2m)$, $g(2m + 6)$, ... $g(2m + 6i)$ which gives a maximum value $X = 3750$ µs. The RX hop frequencies used shall correspond to the RX slot numbers.

It is recommended that single slot packets are used upon return from hold to minimize the synchronization time, especially after long hold periods that require search windows exceeding 625 µs.

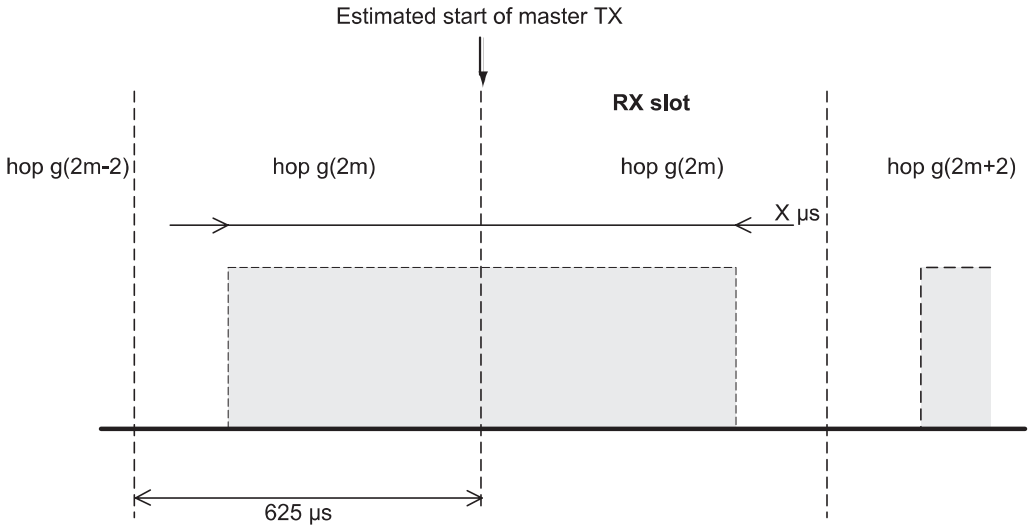


Figure 42—RX timing of slave returning from hold state

8.9.4 Park and sniff modes wake-up

The park and sniff modes are similar to the hold mode. A slave in park or sniff mode periodically wakes up to listen to transmissions from the master and to re-synchronize its clock offset. As in the return from hold mode, a slave in park or sniff mode when waking up may increase the length of the search window from 20 µs to a larger value X µs as illustrated in Figure 42.

8.9.5 Page state

In the page state, the master transmits the device access code (ID packet) corresponding to the slave to be connected, rapidly on a large number of different hop frequencies. Since the ID packet is a very short packet, the hop rate is increased from 1600 hops/s to 3200 hops/s. In a single TX slot interval, the paging master transmits on two different hop frequencies. In a single RX slot interval, the paging transceiver listens on two different hop frequencies; see Figure 43. During the TX slot, the paging unit sends an ID packet at the TX hop frequencies $f(k)$ and $f(k + 1)$. In the RX slot, it listens for a response on the corresponding RX hop frequencies $f'(k)$ and $f'(k + 1)$. The listening periods are exactly timed 625 µs after the corresponding paging packets, and include a ±10 µs uncertainty window.

Copyright © 2002 IEEE. All rights reserved.

LINKPLAY EXHIBIT 1008
Page 98 of 1169

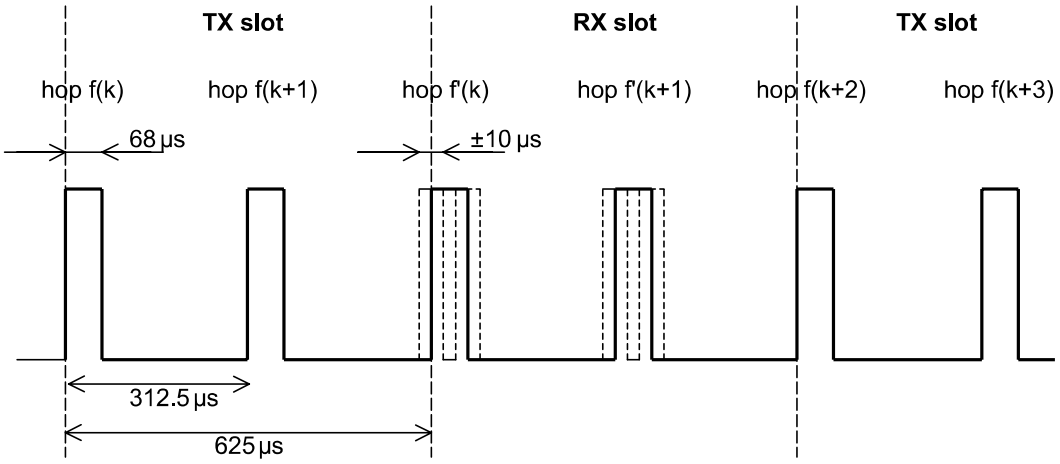


Figure 43—RX/TX cycle of Bluetooth transceiver in PAGE mode

8.9.6 FHS packet

At connection setup and during a master-slave switch, an FHS packet is transferred from the master to the slave. This packet will establish the timing and frequency synchronization (see also 8.4.4.1.4). After the slave unit has received the page message, it will return a response message which again consists of the ID packet and follows exactly 625 µs after the receipt of the page message. The master will send the FHS packet in the TX slot following the RX slot in which it received the slave response, according to the RX/TX timing of the master. The time difference between the response and FHS message will depend on the timing of the page message the slave received. In Figure 44, the slave receives the paging message sent first in the master-to-slave slot. It will then respond with an ID packet in the first half of the slave-to-master slot. The timing of the FHS packet is based on the timing of the page message sent first in the preceding master-to-slave slot: there is an exact 1250 µs delay between the first page message and the FHS packet. The packet is sent at the hop frequency *f(k + 1)* which is the hop frequency following the hop frequency *f(k)* the page message was received in. In Figure 45, the slave receives the paging message sent on the second frequency of the preceding master-to-slave slot. It will then respond with an ID packet in the second half of the slave-to-master slot exactly 625 µs after the receipt of the page message. The timing of the FHS packet is still based on the timing of the page message sent first in the preceding master-to-slave slot: there is an exact 1250 µs delay between the first page message and the FHS packet. The packet is sent at the hop frequency *f(k + 2)* which is the hop frequency following the hop frequency *f(k + 1)* the page message was received in.

Copyright © 2002 IEEE. All rights reserved.

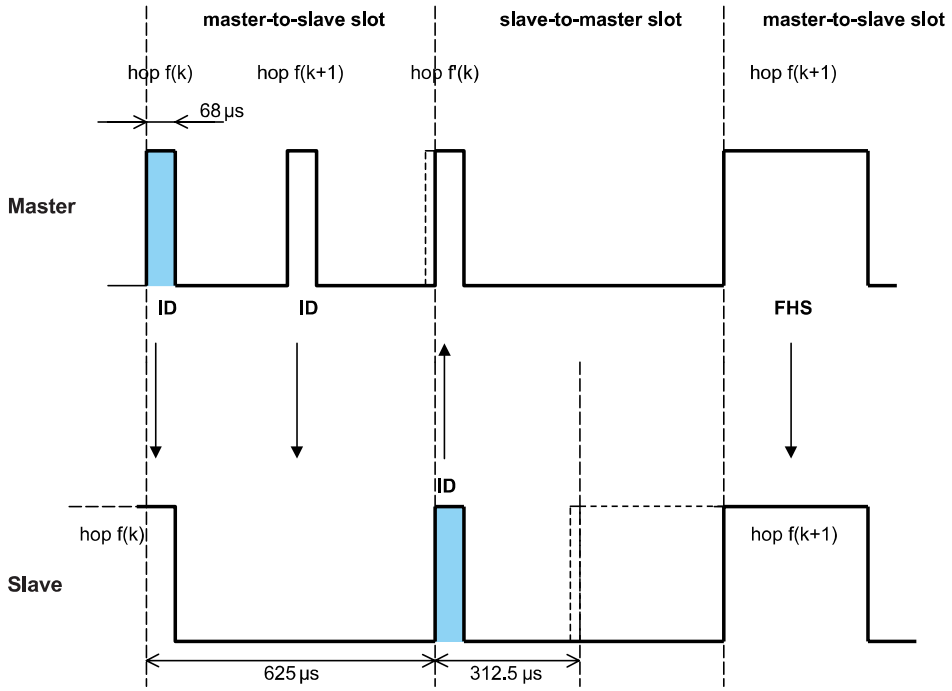


Figure 44—Timing of FHS packet on successful page in first half slot

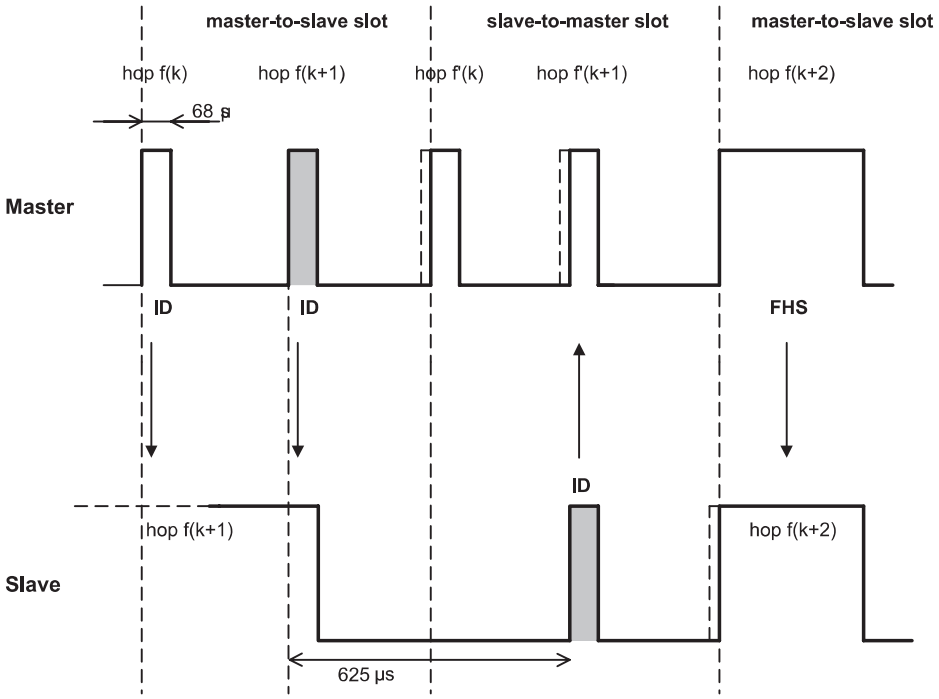


Figure 45—Timing of FHS packet on successful page in second half slot

Copyright © 2002 IEEE. All rights reserved.

LINKPLAY EXHIBIT 1008
Page 100 of 1169

The slave will adjust its RX/TX timing according to the reception of the FHS packet (and not according to the reception of the page message). The response sent 625 µs after the start of the FHS packet acknowledges receipt of the FHS packet.

8.9.7 Multislave operation

As was mentioned in the beginning of this clause, the master always starts the transmission in the even-numbered slots whereas the slaves start their transmission in the odd-numbered slots. This means that the timing of the master and the slave(s) is shifted by one slot (625 µs); see Figure 46.

Only the slave that is addressed by its AM_ADDR can return a packet in the next slave-to-master slot. If no valid AM_ADDR is received, the slave may only respond if it concerns its reserved SCO slave-to-master slot. In case of a broadcast message, no slave is allowed to return a packet (an exception is the access window for access requests in the park mode; see 8.10.8.4).

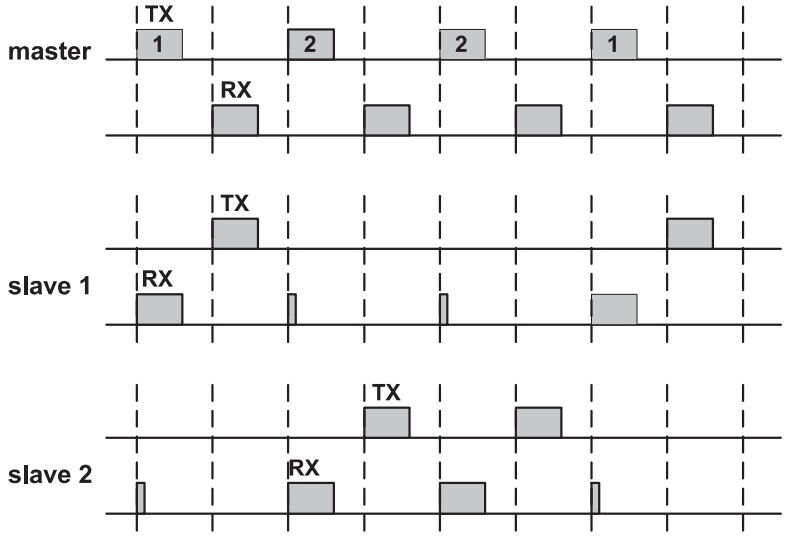


Figure 46—RX/TX timing in multislave configuration

8.10 Channel control

8.10.1 Scope

This section describes how the channel of a piconet is established and how units can be added to and released from the piconet. Several states of operation of the Bluetooth units are defined to support these functions. In addition, the operation of several piconets sharing the same area, the so-called scatternet, is discussed. A special section is attributed to the Bluetooth clock which plays a major role in the FH synchronization.

8.10.2 Master-slave definition

The channel in the piconet is characterized entirely by the master of the piconet. The Bluetooth device address (BD_ADDR) of the master determines the FH hopping sequence and the channel access code; the system clock of the master determines the phase in the hopping sequence and sets the timing. In addition, the master controls the traffic on the channel by a polling scheme.

Copyright © 2002 IEEE. All rights reserved.
LINKPLAY EXHIBIT 1008
Page 101 of 1169

By definition, the master is represented by the Bluetooth unit that initiates the connection (to one or more slave units). Note that the names “master” and “slave” only refer to the protocol on the channel: the Bluetooth units themselves are identical; that is, any unit can become a master of a piconet. Once a piconet has been established, master-slave roles can be exchanged. This is described in more detail in 8.10.9.3.

8.10.3 Bluetooth clock

Every Bluetooth unit has an internal system clock which determines the timing and hopping of the transceiver. The Bluetooth clock is derived from a free running native clock which is never adjusted and is never turned off. For synchronization with other units, only offsets are used that, added to the native clock, provide temporary Bluetooth clocks which are mutually synchronized. It should be noted that the Bluetooth clock has no relation to the time of day; it can therefore be initialized at any value. The Bluetooth clock provides the heart beat of the Bluetooth transceiver. Its resolution is at least half the TX or RX slot length, or 312.5 µs. The clock has a cycle of about a day. If the clock is implemented with a counter, a 28-bit counter is required that wraps around at 2^{28}–1. The LSB ticks in units of 312.5 µs, giving a clock rate of 3.2 kHz.

The timing and the frequency hopping on the channel of a piconet is determined by the Bluetooth clock of the master. When the piconet is established, the master clock is communicated to the slaves. Each slave adds an offset to its native clock to be synchronized to the master clock. Since the clocks are free-running, the offsets have to be updated regularly.

The clock determines critical periods and triggers the events in the Bluetooth receiver. Four periods are important in the Bluetooth system: 312.5 µs, 625 µs, 1.25 ms, and 1.28 s; these periods correspond to the timer bits CLK_0, CLK_1, CLK_2, and CLK_{12}, respectively (see Figure 47). Master-to-slave transmission starts at the even-numbered slots when CLK_0 and CLK_1 are both zero.

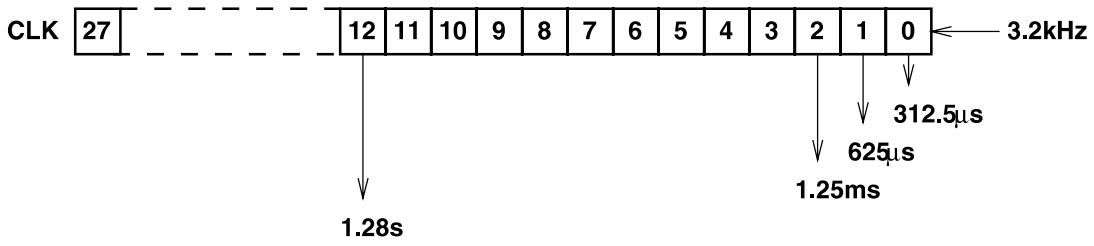


Figure 47—Bluetooth clock

In the different modes and states a Bluetooth unit can reside in, the clock has different appearances:

— CLKN native clock
— CLKE estimated clock
— CLK master clock

CLKN is the free-running native clock and is the reference to all other clock appearances. In states with high activity, the native clock is driven by the reference (e.g., crystal) oscillator with worst case accuracy of ±20 ppm. In the low power states, like STANDBY, HOLD, PARK and SNIFF, the native clock may be driven by a low power oscillator (LPO) with relaxed accuracy (±250 ppm).

CLKE and CLK are derived from the reference CLKN by adding an offset. CLKE is a clock estimate a paging unit makes of the native clock of the recipient; i.e. an offset is added to the CLKN of the pager to approximate the CLKN of the recipient (see Figure 48). By using the CLKN of the recipient, the pager speeds up the connection establishment.

Copyright © 2002 IEEE. All rights reserved.

LINKPLAY EXHIBIT 1008
Page 102 of 1169

CLK is the master clock of the piconet. It is used for all timing and scheduling activities in the piconet. All Bluetooth devices use the CLK to schedule their transmission and reception. The CLK is derived from the native clock CLKN by adding an offset (see Figure 49). The offset is zero for the master since CLK is identical to its own native clock CLKN. Each slave adds an appropriate offset to its CLKN such that the CLK corresponds to the CLKN of the master. Although all CLKNs in the Bluetooth devices run at the same nominal rate, mutual drift causes inaccuracies in CLK. Therefore, the offsets in the slaves are regularly updated such that CLK is approximately CLKN of the master.

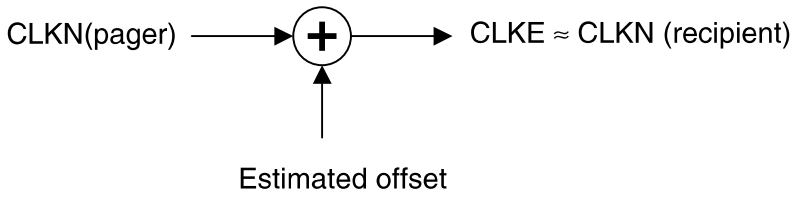


Figure 48—Derivation of CLKE

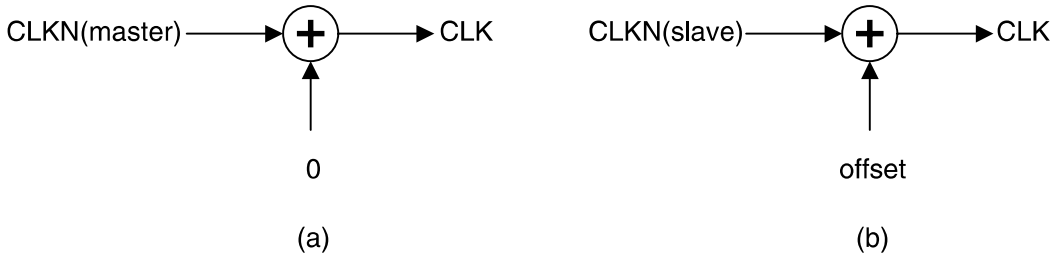


Figure 49—Derivation of CLK in master (a) and in slave (b)

8.10.4 Overview of states

Figure 50 shows a state diagram illustrating the different states used in the Bluetooth link controller. There are two major states: STANDBY and CONNECTION; in addition, there are seven substates, page, page scan, inquiry, inquiry scan, master response, slave response, and inquiry response. The substates are interim states that are used to add new slaves to a piconet. To move from one state to the other, either commands from the Bluetooth link manager are used, or internal signals in the link controller are used (such as the trigger signal from the correlator and the timeout signals).

Copyright © 2002 IEEE. All rights reserved.

.

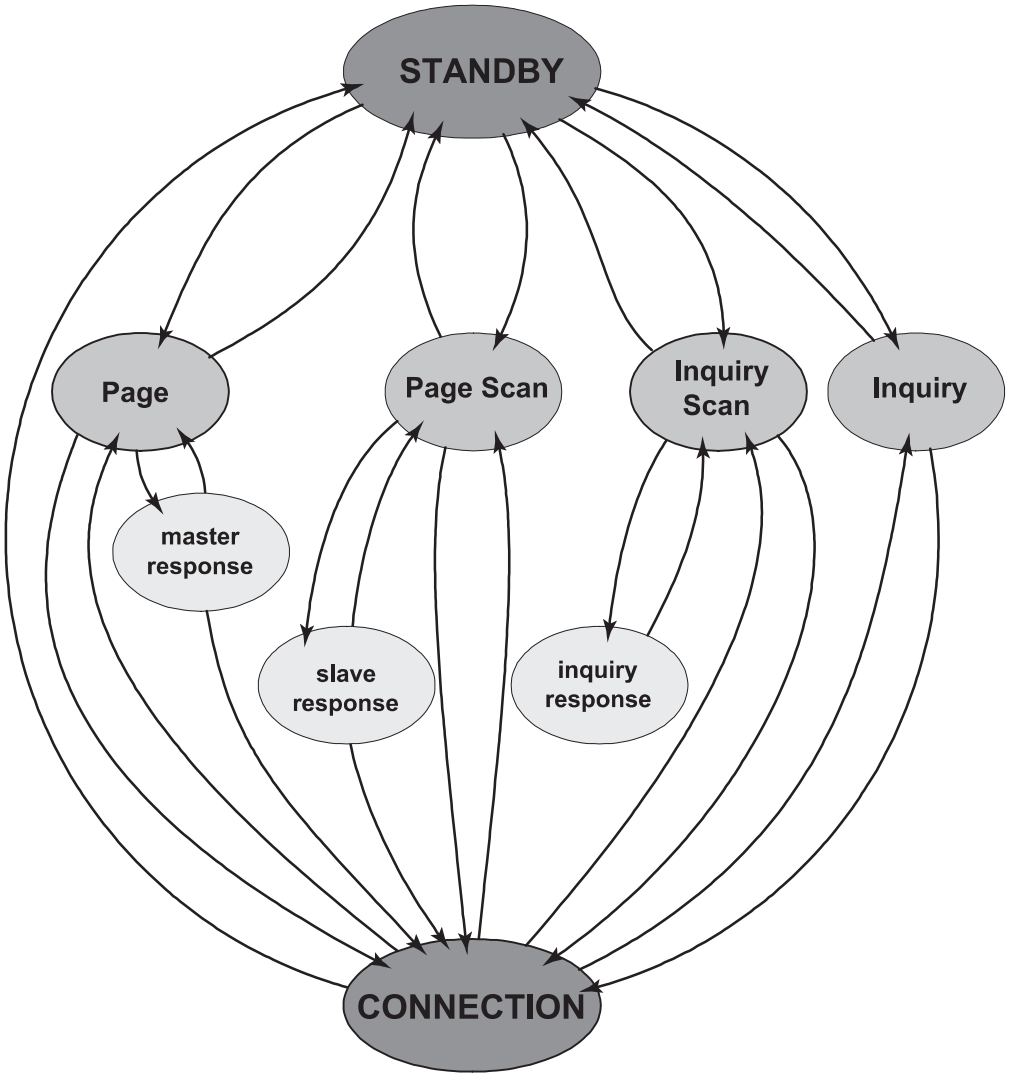


Figure 50—State diagram of Bluetooth link controller

8.10.5 Standby state

The STANDBY state is the default state in the Bluetooth unit. In this state, the Bluetooth unit is in a low-power mode. Only the native clock is running at the accuracy of the LPO (or better).

The controller may leave the STANDBY state to scan for page or inquiry messages, or to page or inquiry itself. When responding to a page message, the unit will not return to the STANDBY state but enter the CONNECTION state as a slave. When carrying out a successful page attempt, the unit will enter the CONNECTION state as a master. The intervals with which scan activities can be carried out are discussed in 8.10.6.2 and 8.10.7.2.

Copyright © 2002 IEEE. All rights reserved.

LINKPLAY EXHIBIT 1008
Page 104 of 1169

8.10.6 Access procedures

8.10.6.1 General

In order to establish new connections the procedures inquiry and paging are used. The inquiry procedure enables a unit to discover which units are in range, and what their device addresses and clocks are. With the paging procedure, an actual connection can be established. Only the Bluetooth device address is required to set up a connection. Knowledge about the clock will accelerate the setup procedure. A unit that establishes a connection will carry out a page procedure and will automatically be the master of the connection.

In the paging and inquiry procedures, the device access code (DAC) and the inquiry access code (IAC) are used, respectively. A unit in the page scan or inquiry scan substate correlates against these respective access codes with a matching correlator.

For the paging process, several paging schemes can be applied. There is one mandatory paging scheme which has to be supported by each Bluetooth device. This mandatory scheme is used when units meet for the first time, and in case the paging process directly follows the inquiry process. Two units, once connected using a mandatory paging/scanning scheme, may agree on an optional paging/scanning scheme. Optional paging schemes are discussed in Annex D. In the current chapter, only the mandatory paging scheme is considered.

8.10.6.2 Page scan

In the page scan substate, a unit listens for its own device access code for the duration of the scan window $T_{w\ page\ scan}$. (11.2.7.20) During the scan window, the unit listens at a single hop frequency, its correlator matched to its device access code. The scan window shall be long enough to completely scan 16 page frequencies.

When a unit enters the page scan substate, it selects the scan frequency according to the page hopping sequence corresponding to this unit (see 8.11.3.1). This is a 32-hop sequence (or a 16-hop sequence in case of a reduced-hop system) in which each hop frequency is unique. The page hopping sequence is determined by the unit's Bluetooth device address (BD_ADDR). The phase in the sequence is determined by $CLKN_{16\text{-}12}$ of the unit's native clock ($CLKN_{15\text{-}12}$ in case of a reduced-hop system); that is, every 1.28s a different frequency is selected.

If the correlator exceeds the trigger threshold during the page scan, the unit will enter the slave response substate, which is described in 8.10.6.4.1.

The page scan substate can be entered from the STANDBY state or the CONNECTION state. In the STANDBY state, no connection has been established and the unit can use all the capacity to carry out the page scan. Before entering the page scan substate from the CONNECTION state, the unit preferably reserves as much capacity for scanning. If desired, the unit may place ACL connections in the HOLD mode or even use the PARK mode, see 8.10.8.3 and 8.10.8.4. SCO connections are preferably not interrupted by the page scan. In this case, the page scan may be interrupted by the reserved SCO slots which have higher priority than the page scan. SCO packets should be used requiring the least amount of capacity (HV3 packets). The scan window shall be increased to minimize the setup delay. If one SCO link is present using HV3 packets and T_{SCO}=6 slots, a total scan window $T_{w\ page\ scan}$ of at least 36 slots (22.5ms) is recommended; if two SCO links are present using HV3 packets and T_{SCO}=6 slots, a total scan window of at least 54 slots (33.75ms) is recommended.

The scan interval $T_{page\ scan}$ is defined as the interval between the beginnings of two consecutive page scans. A distinction is made between the case where the scan interval is equal to the scan window $T_{w\ page\ scan}$ (continuous scan), the scan interval is maximal 1.28s, or the scan interval is maximal 2.56s. These three cases determine the behavior of the paging unit; that is, whether the paging unit shall use R0, R1 or R2

Copyright © 2002 IEEE. All rights reserved.
LINKPLAY EXHIBIT 1008
Page 105 of 1169

(see also 8.10.6.3). Table 23 illustrates the relationship between $T_{page\ scan}$ and modes R0, R1 and R2. Although scanning in the R0 mode is continuous, the scanning may be interrupted by, for example, reserved SCO slots. The scan interval information is included in the SR field in the FHS packet.

During page scan the Bluetooth unit may choose to use an optional scanning scheme. (An exception is the page scan after returning an inquiry response message. See 8.10.7.4 for details.)

Table 23—Relationship between scan interval, train repetition, and paging modes R0, R1, and R2

SR mode	$T_{page\ scan}$	N_{page}
R0	continuous	≥1
R1	≤ 1.28s	≥ 128
R2	≤ 2.56s	≥ 256
Reserved	—	—

8.10.6.3 Page

The page substate is used by the master (source) to activate and connect to a slave (destination) which periodically wakes up in the page scan substate. The master tries to capture the slave by repeatedly transmitting the slave's device access code (DAC) in different hop channels. Since the Bluetooth clocks of the master and the slave are not synchronized, the master does not know exactly when the slave wakes up and on which hop frequency. Therefore, it transmits a train of identical DACs at different hop frequencies, and listens in between the transmit intervals until it receives a response from the slave.

The page procedure in the master consists of a number of steps. First, the slave's device address is used to determine the page hopping sequence (see 8.11.3.2). This is the sequence the master will use to reach the slave. For the phase in the sequence, the master uses an estimate of the slave's clock. This estimate can for example be derived from timing information that was exchanged during the last encounter with this particular device (which could have acted as a master at that time), or from an inquiry procedure. With this estimate CLKE of the slave's Bluetooth clock, the master can predict on which hop channel the slave will start page scan.

The estimate of the Bluetooth clock in the slave can be completely wrong. Although the master and the slave use the same hopping sequence, they use different phases in the sequence and will never meet each other. To compensate for the clock drifts, the master will send its page message during a short time interval on a number of wake-up frequencies. It will also transmit on hop frequencies just before and after the current, predicted hop frequency. During each TX slot, the master sequentially transmits on two different hop frequencies. (Since the page message is the ID packet which is only 68 bits in length, there is ample time (224.5 µs minimal) to switch the synthesizer.) In the following RX slot, the receiver will listen sequentially to two corresponding RX hops for ID packet. The RX hops are selected according to the page_response hopping sequence. The page_response hopping sequence is strictly related to the page hopping sequence; that is: for each page hop there is a corresponding page_response hop. The RX/TX timing in the page substate has been described in 8.9, see also Figure 43. In the next TX slot, it will transmit on two hop frequencies different from the former ones. The synthesizer hop rate is increased to 3200 hops/s.

A distinction must be made between the 79-hop systems and the 23-hop systems. First the 79-hop systems are considered. With the increased hopping rate as described above, the transmitter can cover 16 different hop frequencies in 16 slots or 10 ms. The page hopping sequence is divided over two paging trains A and B

Copyright © 2002 IEEE. All rights reserved.

LINKPLAY EXHIBIT 1008
Page 106 of 1169

of 16 frequencies. Train A includes the 16 hop frequencies surrounding the current, predicted hop frequency $f(k)$, where k is determined by the clock estimate $CLKE_{16\text{-}12}$. So the first train consists of hops $f(k-8)$, $f(k-7)$, ..., $f(k)$, ..., $f(k+7)$.

When the difference between the Bluetooth clocks of the master and the slave is between -8×1.28 s and $+7 \times 1.28$ s, one of the frequencies used by the master will be the hop frequency the slave will listen to. However, since the master does not know when the slave will enter the page scan substate, it has to repeat this train **A** N_{page} times or until a response is obtained. If the slave scan interval corresponds to R1, the repetition number is at least 128; if the slave scan interval corresponds to R2, the repetition number is at least 256.

Note that $CLKE_{16\text{-}12}$ changes every 1.28 s; therefore, every 1.28 s, the trains will include different frequencies of the page hopping set.

When the difference between the Bluetooth clocks of the master and the slave is less than -8×1.28 s or larger than $+7 \times 1.28$ s, more distant hops should be probed. Since in total, there are only 32 dedicated wake-up hops, the more distant hops are the remaining hops not being probed yet. The remaining 16 hops are used to form the new 10 ms train B. The second train consists of hops

$f(k-16), f(k-15), ..., f(k-9), f(k+8), ..., f(k+15)$

Train B is repeated for N_{page} times. If still no response is obtained, the first train A is tried again N_{page} times. Alternate use of train A and train B is continued until a response is received or the timeout *pageTO* is exceeded. If during one of the listening occasions, a response is returned by the slave, the master unit enters the master response substate.

The description for paging and page scan procedures given here has been tailored towards the 79-hop systems used in the US and Europe. For the 23-hop systems as used in France, the procedure is slightly different. In the 23-hop case, the length of the page hopping sequence is reduced to 16. As a consequence, there is only a single train (train A) including all the page hopping frequencies. The phase to the page hopping sequence is not $CLKE_{16\text{-}12}$ but $CLKE_{15\text{-}12}$. An estimate of the slave's clock does not have to be made.

The page substate can be entered from the STANDBY state or the CONNECTION state. In the STANDBY state, no connection has been established and the unit can use all the capacity to carry out the page. Before entering the page substate from the CONNECTION state, the unit shall free as much capacity as possible for scanning. To ensure this, it is recommended that the ACL connections are put on hold or park. However, the SCO connections shall not be disturbed by the page. This means that the page will be interrupted by the reserved SCO slots which have higher priority than the page. In order to obtain as much capacity for paging, it is recommended to use the SCO packets which use the least amount of capacity (HV3 packets). If SCO links are present, the repetition number N_{page} of a single train shall be increased, see Table 24. Here it has been assumed that the HV3 packets are used with an interval $T_{SCO} = 6$ slots, which would correspond to a 64 kb/s voice link.

Copyright © 2002 IEEE. All rights reserved.
LINKPLAY EXHIBIT 1008
Page 107 of 1169

Table 24—Relationship between train repetition and paging modes R0, R1, and R2 when SCO links are present

SR mode	No SCO link	One SCO link (HV3)	Two SCO links (HV3)
R0	$N_{page} \geq 1$	$N_{page} \geq 2$	$N_{page} \geq 3$
R1	$N_{page} \geq 128$	$N_{page} \geq 256$	$N_{page} \geq 384$
R2	$N_{page} \geq 256$	$N_{page} \geq 512$	$N_{page} \geq 768$

The construction of the page train is independent of the presence of SCO links; that is, SCO packets are sent on the reserved slots but do not affect the hop frequencies used in the unreserved slots (see Figure 51).

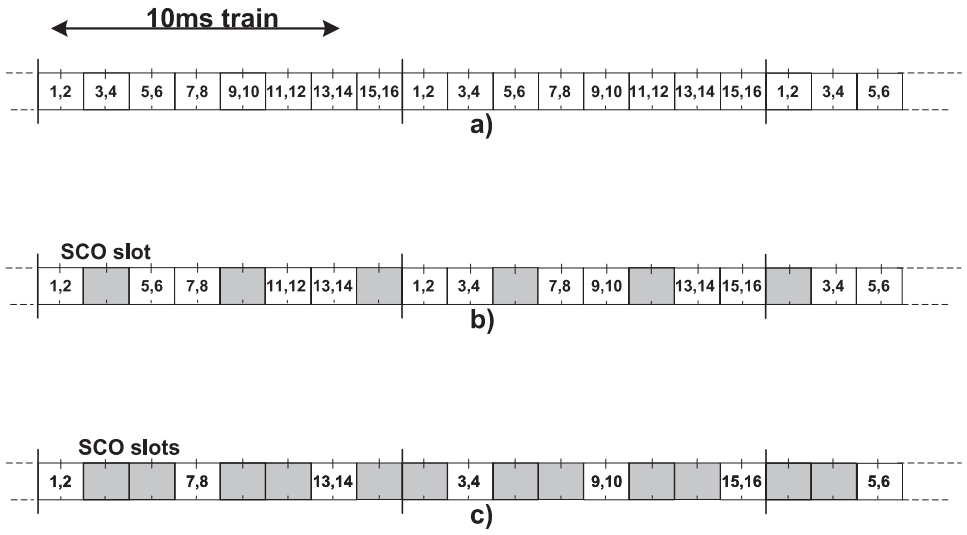


Figure 51—Conventional page (a), page while one SCO link present (b), page while two SCO links present (c)

For the descriptions of optional paging schemes see Annex D.

8.10.6.4 Page response procedures

When a page message is successfully received by the slave, there is a coarse FH synchronization between the master and the slave. Both the master and the slave enter a response routine to exchange vital information to continue the connection setup. Important for the piconet connection is that both Bluetooth units use the same channel access code, use the same channel hopping sequence, and that their clocks are synchronized. These parameters are derived from the master unit. The unit that initializes the connection (starts paging) is defined as the master unit (which is thus only valid during the time the piconet exists). The channel access code and channel hopping sequence are derived from the Bluetooth device address (BD_ADDR) of the master. The timing is determined by the master clock. An offset is added to the slave's native clock to temporarily synchronize the slave clock to the master clock. At start-up, the master parameters have to be transmitted from the master to the slave. The messaging between the master and the slave at start-up will be considered in this section.

Copyright © 2002 IEEE. All rights reserved.

LINKPLAY EXHIBIT 1008
Page 108 of 1169

The initial messaging between master and slave is shown in Table 25 and in Figure 52 and Figure 53. In those two figures frequencies $f(k)$, $f(k + 1)$, etc. are the frequencies of the page hopping sequence determined by the slave's BD_ADDR. The frequencies $f'(k)$, $f'(k + 1)$, etc. are the corresponding page_response frequencies (slave-to-master). The frequencies $g(m)$ belong to the channel hopping sequence.

In step 1 (see Table 25), the master unit is in page substate and the slave unit in the page scan substate. Assume in this step that the page message (in this case the slave's device access code) sent by the master is received by the slave. On recognizing its device access code, the slave enters the slave response in step 2. The master waits for a reply from the slave and when this arrives in step 2, it will enter the master response in step 3. Note that during the initial message exchange, all parameters are derived from the slave's BD_ADDR, and that only the page hopping and page_response hopping sequences are used (which are also derived from the slave's BD_ADDR). Note that when the master and slave enter the response states, their clock input to the page and page_response hop selection is frozen as is described in 8.11.3.3.

Table 25—Initial messaging during startup

Step	Message	Direction	Hopping sequence	Access code and clock
1	slave ID	master to slave	page	slave
2	slave ID	slave to master	page response	slave
3	FHS	master to slave	page	slave
4	slave ID	slave to master	page response	slave
5	1st packet master	master to slave	channel	master
6	1st packet slave	slave to master	channel	master

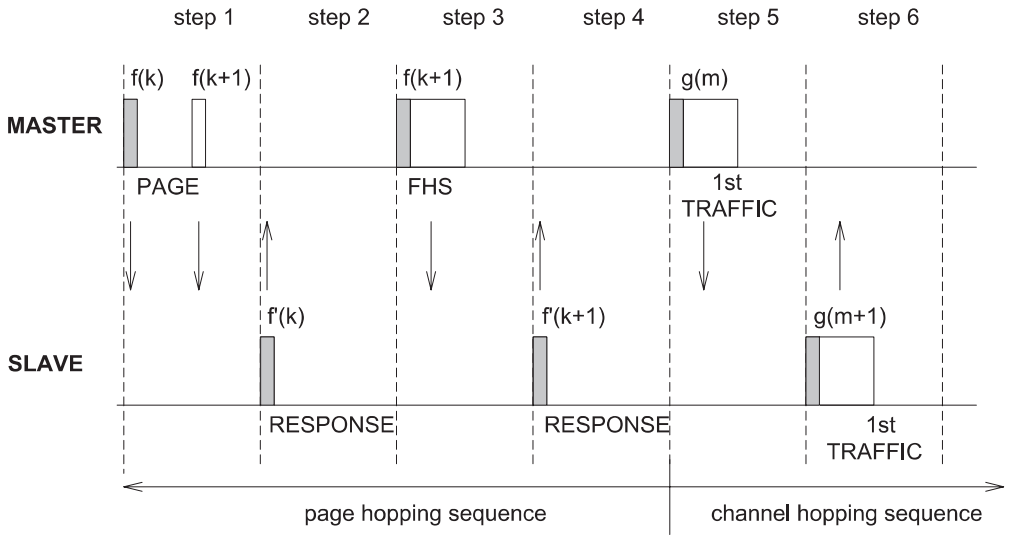


Figure 52—Messaging at initial connection when slave responds to first page message

Copyright © 2002 IEEE. All rights reserved.

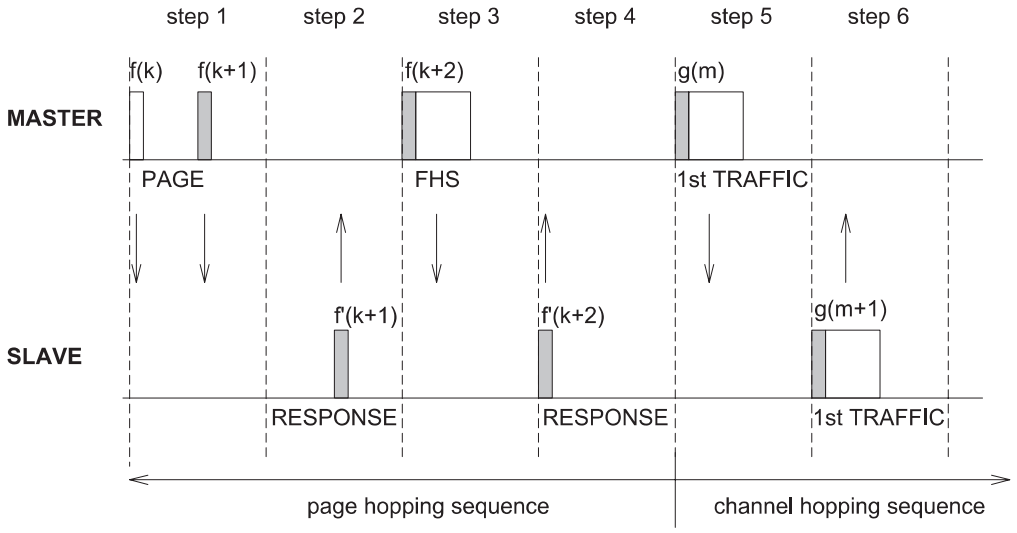


Figure 53—Messaging at initial connection when slave responds to second page message

8.10.6.4.1 Slave response

After having received its own device access code in step 1, the slave unit transmits a response message in step 2. This response message again only consists of the slave's device access code. The slave will transmit this response 625 µs after the beginning of the received page message (slave ID packet) and at the response hop frequency that corresponds to the hop frequency in which the page message was received. The slave transmission is therefore time aligned to the master transmission. During initial messaging, the slave still uses the page response hopping sequence to return information to the master. The clock input $CLKN_{16\text{-}12}$ is frozen at the value it had at the time the page message was received.

After having sent the response message, the slave's receiver is activated (312.5 µs after the start of the response message) and awaits the arrival of an FHS packet. Note that an FHS packet can arrive 312.5 µs after the arrival of the page message as shown in Figure 53, and not after 625 µs as is usually the case in the RX/TX timing. More details about the timing can be found in 8.9.6.

If the setup fails before the CONNECTION state has been reached, the following procedure is carried out. The slave will keep listening as long as no FHS packet is received until *pageRespTO* is exceeded. Every 1.25 ms, however, it will select the next master-to-slave hop frequency according to the page hop sequence. If nothing is received after *pageRespTO*, the slave returns back to the page scan substate for one scan period. Length of the scan period depends on the SCO slots present. If no page message is received during this additional scan period, the slave will resume scanning at its regular scan interval and return to the state it was in prior to the first page scan state.

If an FHS packet is received by the slave in the slave response substate, the slave returns a response (slave's device access code only) in step 4 to acknowledge the reception of the FHS packet (still using the page response hopping sequence). The transmission of this response packet is based on the reception of the FHS packet. Then the slave changes to the channel (master's) access code and clock as received from the FHS packet. Only the 26 MSBs of the master clock are transferred: the timing is assumed such that CLK_1 and CLK_0 are both zero at the time the FHS packet was received as the master transmits in even slots only. From the master clock in the FHS packet, the offset between the master's clock and the slave's clock is determined and reported to the slave's link manager.

Copyright © 2002 IEEE. All rights reserved.

LINKPLAY EXHIBIT 1008
Page 110 of 1169

Finally, the slave enters the CONNECTION state in step 5. From then on, the slave will use the master's clock and the master BD_ADDR to determine the channel hopping sequence and the channel access code. The connection mode starts with a POLL packet transmitted by the master. The slave responds with any type of packet. If the POLL packet is not received by the slave, or the response packet is not received by the master, within *newconnectionTO* number of slots after FHS packet acknowledgement, the master and the slave will return to page and page scan substates, respectively (see 8.10.8).

8.10.6.4.2 Master response

When the master has received a response message from the slave in step 2, it will enter the master response routine. It freezes the current clock input to the page hop selection scheme. Then the master will transmit an FHS packet in step 3 containing the master's real-time Bluetooth clock, the master's 48-bit BD_ADDR address, the BCH parity bits, and the class of device. The FHS packet contains all information to construct the channel access code without requiring a mathematical derivation from the master device address. The FHS packet is transmitted at the beginning of the master-to-slave slot following the slot in which the slave has responded. So the TX timing of the FHS is not based on the reception of the response packet from the slave. The FHS packet may therefore be sent 312.5 μs after the reception of the response packet like shown in Figure 53 and not 625 μs after the received packet as is usual in the RX/TX timing (see also 8.9.6).

After the master has sent its FHS packet, it waits for a second response from the slave in step 4 which acknowledges the reception of the FHS packet. Again this is only the slave's device access code. If no response is received, the master retransmits the FHS packet, but with an updated clock and still using the slave's parameters. It will retransmit (the clock is updated every retransmission) until a second slave response is received, or the timeout of *pageRespTO* is exceeded. In the latter case, the master turns back to the page substate and sends an error message to the link manager. During the retransmissions of the FHS packet, the master keeps using the page hopping sequence.

If the slave's response is indeed received, the master changes to the master parameters for the channel access code and the master clock. The lower clock bits CLK_0 and CLK_1 are zero at the start of the FHS packet transmission and are not included in the FHS packet. Finally, the master enters the CONNECTION state in step 5. The master BD_ADDR is used to change to a new hopping sequence, the *channel hopping sequence*. The channel hopping sequence uses all 79 hop channels in a (pseudo) random fashion (see also 8.11.3.6).

The master can now send its first traffic packet in a hop determined with the new (master) parameters. This first packet will be a POLL packet (see 8.10.8). This packet will be sent within *newconnectionTO* number of slots after reception of the FHS packet acknowledgement. The slave will respond with any type of packet. If the POLL packet is not received by the slave or the POLL packet response is not received by the master within *newconnectionTO* number of slots, the master and the slave will return to page and page scan substates, respectively.

8.10.7 Inquiry procedures

8.10.7.1 General

In the Bluetooth system, an inquiry procedure is defined which is used in applications where the destination's device address is unknown to the source. One can think of public facilities like printers or facsimile machines, or access points to a LAN. Alternatively, the inquiry procedure can be used to discover which other Bluetooth units are within range. During an inquiry substate, the discovering unit collects the Bluetooth device addresses and clocks of all units that respond to the inquiry message. It can then, if desired, make a connection to any one of them by means of the previously described page procedure.

The inquiry message broadcast by the source does not contain any information about the source. However, it may indicate which class of devices should respond. There is one general inquiry access code (GIAC) to inquire for any Bluetooth device, and a number of dedicated inquiry access codes (DIAC) that only inquire

Copyright © 2002 IEEE. All rights reserved.
LINKPLAY EXHIBIT 1008
Page 111 of 1169

for a certain type of devices. The inquiry access codes are derived from reserved Bluetooth device addresses and are further described in 8.4.2.1.

A unit that wants to discover other Bluetooth units enters an inquiry substate. In this substate, it continuously transmits the inquiry message (which is the ID packet, see 8.4.4.1.1) at different hop frequencies. The inquiry hop sequence is always derived from the LAP of the GIAC. Thus, even when DIACs are used, the applied hopping sequence is generated from the GIAC LAP. A unit that allows itself to be discovered, regularly enters the inquiry scan substate to respond to inquiry messages. The following subclauses (8.10.7.2 through 8.10.7.4) describe the message exchange and contention resolution during inquiry response. The inquiry response is optional: a unit is not forced to respond to an inquiry message.

8.10.7.2 Inquiry scan

The inquiry scan substate is very similar to the page scan substate. However, instead of scanning for the unit's device access code, the receiver scans for the inquiry access code long enough to completely scan for 16 inquiry frequencies. The length of this scan period is denoted $T_{w\_inquiry\_scan}$. The scan is performed at a single hop frequency. As in the page procedure, the inquiry procedure uses 32 dedicated inquiry hop frequencies according to the *inquiry hopping sequence*. These frequencies are determined by the general inquiry address. The phase is determined by the native clock of the unit carrying out the inquiry scan; the phase changes every 1.28 s.

Instead or in addition to the general inquiry access code, the unit may scan for one or more dedicated inquiry access codes. However, the scanning will follow the inquiry scan hopping sequence which is determined by the general inquiry address. If an inquiry message is recognized during an inquiry wake-up period, the Bluetooth unit either performs a backoff in CONNECTION or STANDBY state before reentering the inquiry scan substate or enters the inquiry response substate if a random backoff was performed before entering the inquiry scan substate.

The inquiry scan substate can be entered from the STANDBY state or the CONNECTION state. In the STANDBY state, no connection has been established and the unit can use all the capacity to carry out the inquiry scan. Before entering the inquiry scan substate from the CONNECTION state, the unit preferably reserves as much capacity as possible for scanning. If desired, the unit may place ACL connections in the HOLD mode or even use the PARK mode (see 8.10.8.3). SCO connections are preferably not interrupted by the inquiry scan. In this case, the inquiry scan may be interrupted by the reserved SCO slots which have higher priority than the inquiry scan. SCO packets should be used requiring the least amount of capacity (HV3 packets). The scan window, $T_{w\ inquiry\ scan}$, shall be increased to increase the probability to respond to an inquiry message. If one SCO link is present using HV3 packets and $T_{SCO} = 6$ slots, a total scan window of at least 36 slots (22.5 ms) is recommended; if two SCO links are present using HV3 packets and $T_{SCO} = 6$ slots, a total scan window of at least 54 slots (33.75 ms) is recommended.

The scan interval $T_{inquiry\ scan}$ is defined as the interval between two consecutive inquiry scans. The inquiry scan interval shall be at most 2.56 s.

8.10.7.3 Inquiry

The inquiry substate is used by the unit that wants to discover new devices. This substate is very similar to the page substate, the same TX/RX timing is used as used for paging (see 8.9.6 and Figure 43). The TX and RX frequencies follow the inquiry hopping sequence and the inquiry response hopping sequence, and are determined by the general inquiry access code and the native clock of the discovering device. In between inquiry transmissions, the Bluetooth receiver scans for inquiry response messages. When found, the entire response packet (which is in fact a FHS packet) is read, after which the unit continues with the inquiry transmissions. So the Bluetooth unit in an inquiry substate does not acknowledge the inquiry response messages. It keeps probing at different hop channels and in between listens for response packets. Like in the page substate, two 10 ms trains A and B are defined, splitting the 32 frequencies of the inquiry hopping

Copyright © 2002 IEEE. All rights reserved.

LINKPLAY EXHIBIT 1008
Page 112 of 1169

sequence into two 16-hop parts. A single train shall be repeated for at least $N_{inquiry}$ = 256 times before a new train is used. In order to collect all responses in an error-free environment, at least three train switches should have taken place. As a result, the inquiry substate may have to last for 10.24 s unless the inquirer collects enough responses and determines to abort the inquiry substate earlier. If desired, the inquirer can also prolong the inquiry substate to increase the probability of receiving all responses in an error-prone environment. If an inquiry procedure is automatically initiated periodically (say a 10 s period every minute), then the interval between two inquiry instances must be determined randomly. This is done to avoid two Bluetooth units to synchronize their inquiry procedures.

The inquiry substate is continued until stopped by the Bluetooth link manager (when it decides that it has sufficient number of responses), or when a timeout has been reached (*inquiryTO*).

The inquiry substate can be entered from the STANDBY state or the CONNECTION state. In the STANDBY state, no connection has been established and the unit can use all the capacity to carry out the inquiry. Before entering the inquiry substate from the CONNECTION state, the unit shall free as much capacity as possible for scanning. To ensure this, it is recommended that the ACL connections are put on hold or park. However, the SCO connections shall not be disturbed by the inquiry. This means that the inquiry will be interrupted by the reserved SCO slots which have higher priority than the inquiry. In order to obtain as much capacity for inquiry, it is recommended to use the SCO packets which use the least amount of capacity (HV3 packets). If SCO links are present, the repetition number $N_{inquiry}$ shall be increased, see Table 26. Here it has been assumed that the HV3 packet is used with an interval T_{SCO}=6 slots, which would correspond to a 64 kb/s voice link.

Table 26—Increase of train repetition when SCO links are present

	No SCO link	One SCO link (HV3)	Two SCO links (HV3)
$N_{inquiry}$	≥ 256	≥ 512	≥ 768

8.10.7.4 Inquiry response

For the inquiry operation, there is only a slave response, no master response. The master listens between inquiry messages for responses, but after reading a response, it continues to transmit inquiry messages. The slave response routine for inquiries differs completely from the slave response routine applied for pages. When the inquiry message is received in the inquiry scan substate, a response message containing the recipient's address shall be returned. This response message is a conventional FHS packet carrying the unit's parameters. However, a contention problem may arise when several Bluetooth units are in close proximity to the inquiring unit and all respond to an inquiry message at the same time. First of all, every Bluetooth unit has a free running clock; therefore, it is highly unlikely that they all use the same phase of the inquiry hopping sequence. However, in order to avoid collisions between units that do wake up in the same inquiry hop channel simultaneously, the following protocol in the slave's inquiry response is used. If the slave receives an inquiry message, it generates a random number RAND between 0 and 1023. The slave then returns to the CONNECTION or STANDBY state for the duration of RAND time slots. Before returning to the CONNECTION or STANDBY state, the unit may go through the page scan substate; this page scan shall use the mandatory page scan scheme. After at least RAND slots, the unit will return to the inquiry scan substate. On the first inquiry message received in this substate the slave goes into the inquiry response substate and returns an FHS response packet to the master 625 μs after the inquiry message was received. If during the scan no trigger occurs within a timeout period of *inqrespTO*, the slave returns to the STANDBY or CONNECTION state. If the unit does receive an inquiry message and returns an FHS packet, it adds an offset of 1 to the phase in the inquiry hop sequence (the phase has a 1.28 s resolution) and enters the inquiry scan substate again. If the slave is triggered again, it repeats the procedure using a new RAND. The offset to

Copyright © 2002 IEEE. All rights reserved.
LINKPLAY EXHIBIT 1008
Page 113 of 1169

the clock accumulates each time a FHS packet is returned. During a 1.28 s probing window, a slave on average responds four times, but on different frequencies and at different times. Possible SCO slots should have priority over response packets; that is, if a response packet overlaps with an SCO slot, it is not sent but the next inquiry message is awaited.

The messaging during the inquiry routines is summarized in Table 27. In step 1, the master transmits an inquiry message using the inquiry access code and its own clock. The slave responds with the FHS packet which contains the slave's device address, native clock, and other slave information. This FHS packet is returned at a semi-random time. The FHS packet is not acknowledged in the inquiry routine, but it is retransmitted at other times and frequencies as long as the master is probing with inquiry messages.

Table 27—Messaging during inquiry routines

Step	Message	Direction	Hopping sequence	Access code
1	ID	master to slave	inquiry	inquiry
2	FHS	slave to master	inquiry response	inquiry

If the scanning unit uses an optional scanning scheme, after responding to an inquiry with an FHS packet, it will perform page scan using the mandatory page scan scheme for $T_{mandatory\ pscan}$ period. Every time an inquiry response is sent the unit will start a timer with a timeout of $T_{mandatory\ pscan}$. The timer will be reset at each new inquiry response. Until the timer times out, when the unit performs page scan, it will use the mandatory page scanning scheme in the SR mode it uses for all its page scan intervals. Using the mandatory page scan scheme after the inquiry procedure enables all units to connect even if they do not support an optional paging scheme (yet). In addition to using the mandatory page scan scheme, an optional page scan scheme can be used in parallel for the $T_{mandatory\ pscan}$ period.

The $T_{mandatory\ pscan}$ period is included in the SP field of the FHS packet returned in the inquiry response routine (see 8.4.4.1.4). The value of the period is indicated in the Table 28.

Table 28—Mandatory scan periods for P0, P1, and P2 scan period modes

SP mode	$T_{mandatory\ pscan}$
P0	≥20s
P1	≥ 40s
P2	≥ 60s
Reserved	—

8.10.8 Connection state

In the CONNECTION state, the connection has been established and packets can be sent back and forth. In both units, the channel (master) access code and the master Bluetooth clock are used. The hopping scheme uses the *channel hopping sequence*. The master starts its transmission in even slots ($CLK_{1-0} = 00$), the slave starts its transmission in odd slots ($CLK_{1-0} = 10$).

Copyright © 2002 IEEE. All rights reserved.

LINKPLAY EXHIBIT 1008
Page 114 of 1169

The CONNECTION state starts with a POLL packet sent by the master to verify the switch to the master's timing and channel frequency hopping. The slave can respond with any type of packet. If the slave does not receive the POLL packet or the master does not receive the response packet for *newconnectionTO* number of slots, both devices will return to page/page scan substates.

The first information packets in the CONNECTION state contain control messages that characterize the link and give more details regarding the Bluetooth units. These messages are exchanged between the link managers of the units. For example, it defines the SCO links and the sniff parameters. Then the transfer of user information can start by alternately transmitting and receiving packets.

The CONNECTION state is left through a detach or reset command. The detach command is used if the link has been disconnected in the normal way. All configuration data in the Bluetooth link controller is still valid. The reset command is a hard reset of all controller processes. After a reset, the controller has to be reconfigured.

The Bluetooth units can be in several modes of operation during the CONNECTION state: active mode, sniff mode, hold mode, and park mode. These modes are now described in more detail.

8.10.8.1 Active mode

In the active mode, the Bluetooth unit actively participates on the channel. The master schedules the transmission based on traffic demands to and from the different slaves. In addition, it supports regular transmissions to keep slaves synchronized to the channel. Active slaves listen in the master-to-slave slots for packets. If an active slave is not addressed, it may sleep until the next new master transmission. From the type indication in the packet, the number of slots the master has reserved for its transmission can be derived; during this time, the nonaddressed slaves do not have to listen on the master-to-slave slots. A periodic master transmission is required to keep the slaves synchronized to the channel. Since the slaves only need the channel access code to synchronize with, any packet type can be used for this purpose.

8.10.8.2 Sniff mode

In the sniff mode, the duty cycle of the slave's listen activity can be reduced. If a slave participates on an ACL link, it has to listen in every ACL slot to the master traffic. With the sniff mode, the time slots where the master can start transmission to a specific slave is reduced; that is, the master can only start transmission in specified time slots. These so-called sniff slots are spaced regularly with an interval of $T_{\text{sniff.}}$

The slave starts listening at the sniff slots for $N_{\text{sniff attempt}}$ consecutive receive slots unless a packet with matching AM_ADDR is received. After every reception of a packet with matching AM_ADDR, the slave continues listening at the subsequent $N_{\text{sniff timeout}}$ or remaining of the receive slots, whichever is greater. So, for $N_{\text{sniff timeout}} > 0$, the slave continues listening as long as it receives packets with matching AM_ADDR.

Note that Receive slots here are every odd-numbered slots, in which the master may start sending a packet.

Note that $N_{\text{sniff attempt}} = 1$ and $N_{\text{sniff timeout}} = 0$ cause the slave to listen only at the first sniff slot, irrespective of packets received from the master.

Note that $N_{\text{sniff attempt}} = 0$ is not allowed.

To enter the sniff mode, the master or slave shall issue a sniff command via the LM protocol. This message will contain the sniff interval T_{sniff} and an offset $D_{\text{sniff.}}$ The timing of the sniff mode is then determined similar as for the SCO links. In addition, an initialization flag indicates whether initialization procedure 1 or 2 is being used. The device uses initialization 1 when the MSB of the current master clock (CLK_{27}) is 0; it uses initialization 2 when the MSB of the current master clock (CLK_{27}) is 1. The slave shall apply the initialization method as indicated by the initialization flag irrespective of its clock bit value CLK_{27}. The

Copyright © 2002 IEEE. All rights reserved.
LINKPLAY EXHIBIT 1008
Page 115 of 1169

master-to-slave sniff slots determined by the master and the slave shall be initialized on the slots for which the clock satisfies the following equation

$$CLK_{27\text{-}1} \bmod T_{sniff} = D_{sniff} \text{ for initialization 1}$$

$$(\overline{CLK}_{27}, CLK_{26\text{-}1}) \bmod T_{sniff} = D_{sniff} \text{ for initialization 2}$$

The slave-to-master sniff slot determined by the master and the slave shall be initialized on the slots after the master-to-slave sniff slot defined above. After initialization, the clock value CLK(k+1) for the next master-to-slave SNIFF slot is found by adding the fixed interval T_{sniff} to the clock value of the current master-to-slave sniff slot:

$$CLK(k+1) = CLK(k) + T_{sniff}$$

8.10.8.3 Hold mode

During the CONNECTION state, the ACL link to a slave can be put in a hold mode. This means that the slave temporarily does not support ACL packets on the channel any more (note: possible SCO links will still be supported). With the hold mode, capacity can be made free to do other things like scanning, paging, inquiring, or attending another piconet. The unit in hold mode can also enter a low-power sleep mode. During the hold mode, the slave unit keeps its active member address (AM_ADDR).

Prior to entering the hold mode, master and slave agree on the time duration the slave remains in the hold mode. A timer is initialized with the *holdTO* value. When the timer is expired, the slave will wake up, synchronize to the traffic on the channel and will wait for further master instructions.

8.10.8.4 Park mode

When a slave does not need to participate on the piconet channel, but still wants to remain synchronized to the channel, it can enter the park mode which is a low-power mode with very little activity in the slave. In the park mode, the slave gives up its active member address AM_ADDR. Instead, it receives two new addresses to be used in the park mode

— PM_ADDR: 8-bit Parked Member Address

— AR_ADDR: 8-bit Access Request Address

The PM_ADDR distinguishes a parked slave from the other parked slaves. This address is used in the master-initiated unpark procedure. In addition to the PM_ADDR, a parked slave can also be unparked by its 48-bit BD_ADDR. The all-zero PM_ADDR is a reserved address: if a parked unit has the all-zero PM_ADDR it can only be unparked by the BD_ADDR. In that case, the PM_ADDR has no meaning. The AR_ADDR is used by the slave in the slave-initiated unpark procedure. All messages sent to the parked slaves have to be carried by broadcast packets (the all-zero AM_ADDR) because of the missing AM_ADDR.

The parked slave wakes up at regular intervals to listen to the channel in order to re-synchronize and to check for broadcast messages. To support the synchronization and channel access of the parked slaves, the master supports a beacon channel described in the next section. The beacon structure is communicated to the slave when it is being parked. When the beacon structure changes, the parked slaves are updated through broadcast messages.

In addition for using it for low power consumption, the park mode is used to connect more than seven slaves to a single master. At any one time, only seven slaves can be active. However, by swapping active and parked slaves out respectively in the piconet, the number of slave virtually connected can be much larger

Copyright © 2002 IEEE. All rights reserved.

LINKPLAY EXHIBIT 1008
Page 116 of 1169

(255 if the PM_ADDR is used, and even a larger number if the BD_ADDR is used). There is no limitation to the number of slaves that can be parked.

8.10.8.4.1 Beacon channel

To support parked slaves, the master establishes a beacon channel when one or more slaves are parked. The beacon channel consists of one beacon slot or a train of equidistant beacon slots which is transmitted periodically with a constant time interval. The beacon channel is illustrated in Figure 54. A train of N_B ($N_B \geq 1$) beacon slots is defined with an interval of T_B slots. The beacon slots in the train are separated by Δ_B. The start of the first beacon slot is referred to as the beacon instant and serves as the beacon timing reference. The beacon parameters N_B and T_B are chosen such that there are sufficient beacon slots for a parked slave to synchronize to during a certain time window in an error-prone environment.

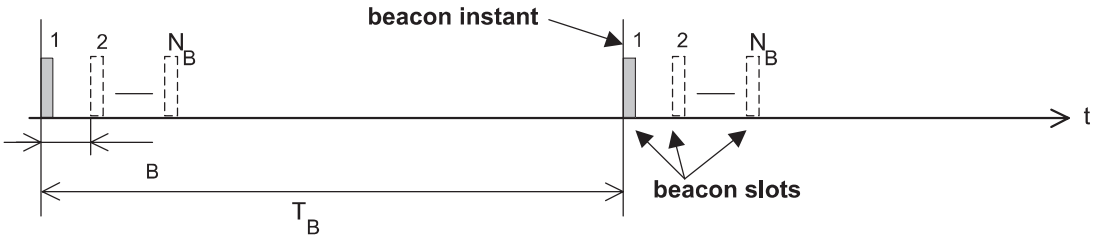


Figure 54—General beacon channel format

When parked, the slave will receive the beacon parameters through an LMP command. In addition, the timing of the beacon instant is indicated through the offset D_B. Like for the SCO link (see 8.3.2), two initialization procedures 1 or 2 are used. The master uses initialization 1 when the MSB of the current master clock (CLK_{27}) is 0; it uses initialization 2 when the MSB of the current master clock (CLK_{27}) is 1. The chosen initialization procedure is also carried by an initialization flag in the LMP command. The slave shall apply the initiations method as indicated by the initialization flag irrespective of its clock bit CLK_{27}. The master-to-slave slot positioned at the beacon instant shall be initialized on the slots for which the clock satisfies the following equation

$$CLK_{27-1} \bmod T_B = D_B \text{ for initialization 1}$$

$$(\overline{CLK}_{27}, CLK_{26-1}) \bmod T_B = D_B \text{ for initialization 2}$$

After initialization, the clock value CLK(k+1) for the next beacon instant is found by adding the fixed interval T_B to the clock value of the current beacon instant:

$$CLK(k+1) = CLK(k) + T_B$$

The beacon channel serves four purposes:

1) Transmission of master-to-slave packets which the parked slaves can use for re-synchronization
2) Carrying messages to the parked slaves to change the beacon parameters
3) Carrying general broadcast messages to the parked slaves
4) Unparking of one or more parked slaves

Since a slave can synchronize to any packet which is preceded by the proper channel access code, the packets carried on the beacon slots do not have to contain specific broadcast packets for parked slaves to be

Copyright © 2002 IEEE. All rights reserved.
LINKPLAY EXHIBIT 1008
Page 117 of 1169

able to synchronize; any packet can be used. The only requirement placed on the beacon slots is that there is master-to-slave transmission present. If there is no information to be sent, NULL packets can be transmitted by the master. If there is indeed broadcast information to be sent to the parked slaves, the first packet of the broadcast message shall be repeated in every beacon slot of the beacon train. However, synchronous traffic like on the SCO link, may interrupt the beacon transmission.

The master communicates with parked slaves using broadcast messages. Since these messages can be time critical, an ongoing repetition train of broadcast message can be prematurely aborted by broadcast information destined to parked slaves in beacon slots and in access windows (see 8.10.8.4.2).

8.10.8.4.2 Beacon access window

In addition to the beacon slots, an access window is defined where the parked slaves can send requests to be unparked. To increase reliability, the access window can be repeated M_{access} times ($M_{access} \geq 1$), see Figure 55. The access window starts a fixed delay D_{access} after the beacon instant. The width of the access window is T_{access}.



Figure 55—Definition of access window

The access window may support different slave access techniques, like polling, random access, or other forms of access. At this stage, only the polling technique has been defined. The format of the polling technique is shown in Figure 56. The same TDD structure is used as on the piconet channel, i.e. master-to-slave transmission is alternated by slave-to-master transmission. The slave-to-master slot is divided into two half slots of 312.5 µs each. The half slot a parked slave is allowed to respond in corresponds to its access request address (AR_ADDR), see also 8.10.8.4.6. For counting the half slots to determine the access request slot, the start of the access window is used (see Figure 56). The slave is only allowed to send an access request in the proper slave-to-master half slot if in the preceding master-to-slave slot a broadcast packet has been received. In this way, the master polls the parked slaves.

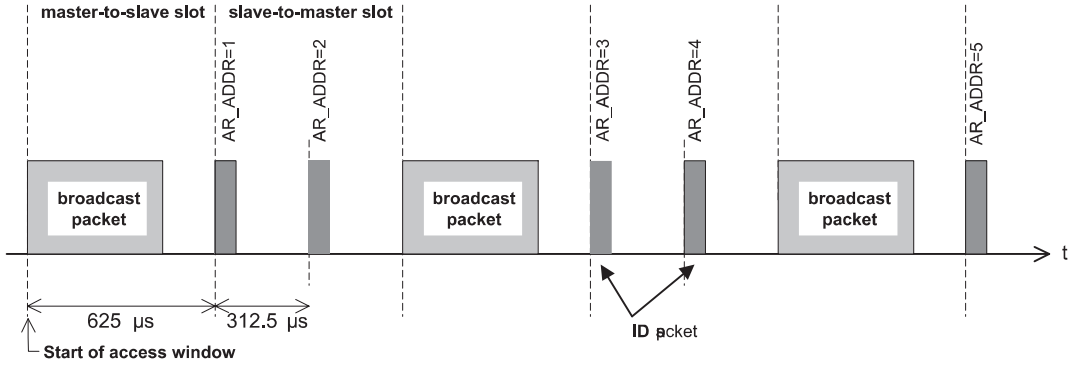


Figure 56—Access procedure applying the polling technique

Copyright © 2002 IEEE. All rights reserved.

LINKPLAY EXHIBIT 1008
Page 118 of 1169

However, the slots of the access window can also be used for traffic on the piconet if required. For example, if an SCO connection has to be supported, the slots reserved for the SCO link may carry SCO information instead of being used for access requests, i.e. if the master-to-slave slot in the access window contains a packet different from a broadcast packet, the following slave-to-master slot cannot be used for slave access requests. Slots in the access window not affected by traffic can still be used according to the defined access structure; an example is shown in Figure 57. The access procedure is continued as if no interruption had taken place.

When the slave is parked, it is indicated what type of access scheme will be used. For the polling scheme, the number of slave-to-master access slots N_{acc_slot} is indicated.

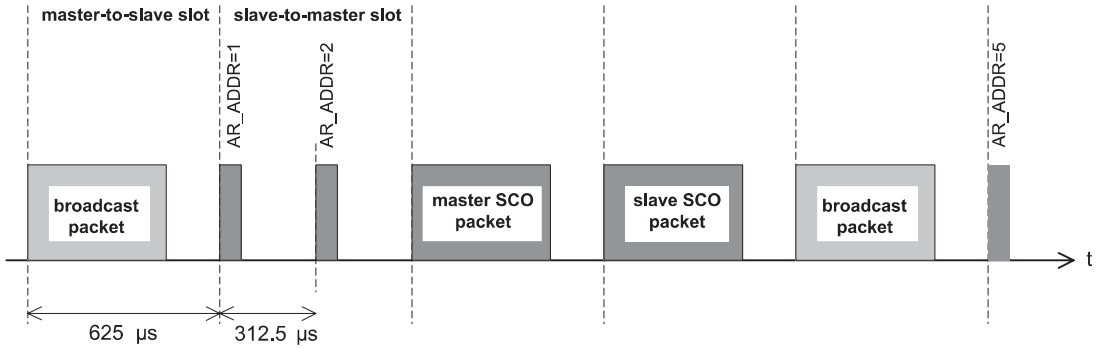


Figure 57—Disturbance of access window by SCO traffic

By default, the access window is always present. However, its activation depends on the master sending broadcast messages to the slave at the appropriate slots in the access window. A flag in a broadcast LMP message within the beacon slots may indicate that the access window(s) belonging to this instant will not be activated. This prevents unnecessary scanning of parked slaves that want to request access.

8.10.8.4.3 Parked slave synchronization

Parked slaves sleep most of the time. However, periodically they wake up to re-synchronize to the channel. Any packet exchanged on the channel can be used for synchronization. Since master transmission is mandatory on the beacon slots, parked slaves will exploit the beacon channel to re-synchronize. A parked slave will wake-up at the beacon instant to read the packet sent on the first beacon slot. If this fails, it will retry on the next beacon slot in the beacon train; in total, there are N_B opportunities per beacon instant to re-synchronize. During the search, the slave may increase its search window (see also 8.9.4). The separation between the beacon slots in the beacon train Δ_B is chosen such that consecutive search windows will not overlap.

The parked slave does not have to wake up at every beacon instant. Instead, a sleep interval can be applied which is longer than the beacon interval T_B (see Figure 58). The slave sleep window shall be a multiple N_{B_sleep} of T_B. The precise beacon instant the slave shall wake up on is indicated by the master with D_{B_sleep} which indicates the offset (in multiples of T_B) with respect to the beacon instant ($0 < D_{B\_sleep} < N_{B\_sleep} - 1$). To initialize the wake-up period, the following equations are used:

$$CLK_{27-1} \bmod (N_{B\_sleep} \bullet T_B) = D_B + D_{B\_sleep} \bullet T_B \text{ for initialization 1}$$

$$(\overline{CLK}_{27}, CLK_{26-1}) \bmod (N_{B\_sleep} \bullet T_B) = D_B + D_{B\_sleep} \bullet T_B \text{ for initialization 2}$$

where initialization 1 is chosen by the master if the MSB in the current master clock is 0 and initialization 2 is chosen if the MSB in the current master clock is 1.

Copyright © 2002 IEEE. All rights reserved.

When the master wants to send broadcast messages to the parked slaves, it may use the beacon slots for these broadcast messages. However, if $N_B < N_{BC}$, the slots following the last beacon slot in the beacon train shall be used for the remaining $N_{BC} - N_B$ broadcast packets. If $N_B > N_{BC}$, the broadcast message is repeated on all N_B beacon slots.

A parked slave shall at least read the broadcast messages sent in the beacon slot(s) it wakes up in; the minimum wake-up activity is to read the channel access code for re-synchronization and the packet header to check for broadcast messages.

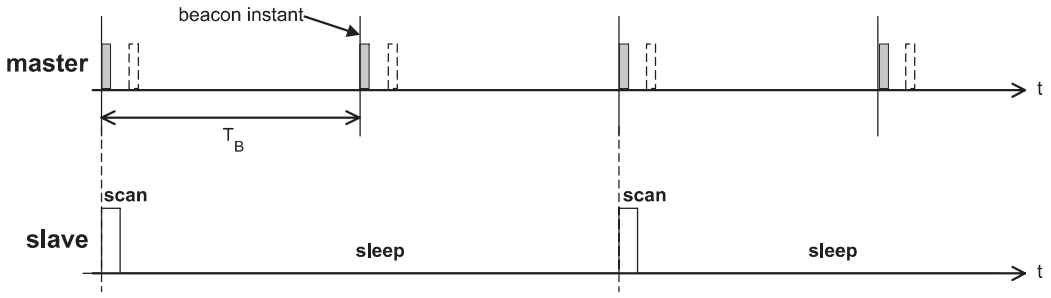


Figure 58—Extended sleep interval of parked slaves

8.10.8.4.4 Parking

A master can park an active slave through the exchange of one or a few LMP commands. Before put into the park mode, the slave is assigned a PM_ADDR and an AR_ADDR. Every parked slave has a unique PM_ADDR; however, the AR_ADDR is not necessarily unique. Also, the beacon parameters are given by the master when the slave is parked. The slave then gives up its AM_ADDR and enters the park mode. A master can park only a single slave at a time. The park message is carried with a normal data packet and addresses the slave through its AM_ADDR.

8.10.8.4.5 Master-activated unparking

The master can unpark a parked slave by sending a dedicated LMP unpark command including the parked slave's address. This message is sent in a broadcast packet on the beacon slots. Either the slave's PM_ADDR is used, or its full BD_ADDR is used. The message also includes the active member address AM_ADDR the slave will use after it has re-entered the piconet. The unpark message can include a number of slave addresses so that multiple slaves can be unparked simultaneously. For each slave, a different AM_ADDR is assigned.

After having received the unpark message, the parked slave matching the PM_ADDR or BD_ADDR will leave the park mode and enter the active mode. It will keep listening to the master until it is addressed by the master through its AM_ADDR. The first packet sent by the master shall be a POLL packet. The return packet in response to the POLL packet confirms that the slave has been unparked. If no response packets from the slave is received for *newconnectionTO* number of slots after the end of beacon repetition period, the master will unpark the slave again. If the slave does not receive the POLL packet for *newconnectionTO* number of slots after the end of beacon repetition period, it will return to park, with the same beacon parameters. After confirming that the slave is active, the master decides in which mode the slave will continue.

Copyright © 2002 IEEE. All rights reserved.

LINKPLAY EXHIBIT 1008
Page 120 of 1169

8.10.8.4.6 Slave-activated unparking

A slave can request access to the channel through the access window defined in 8.10.8.4.2. As shown in Figure 56, the access window includes several slave-to-master half slots where the slave can send an access request message. The specific half slot the slave is allowed to respond in, corresponds to its access request address (AR_ADDR) which it has received when it was parked. The order of the half slots (in Figure 56 the AR_ADDR numbers linearly increase from 1 to 5) is not fixed: an LMP command sent in the beacon slots may reconfigure the access window. When a slave desires access to the channel, it sends an access request message in the proper slave-to-master half slot.The access request message of the slave is the ID packet containing the device access code (DAC) of the master (which is in this case the channel access code without the trailer). The parked slave is only allowed to transmit an access request message in the half slot when in the preceding master-to-slave slot, a broadcast packet has been received. This broadcast message can contain any kind of broadcast information not necessarily related to the parked slave(s). If no broadcast information is available, a broadcast NULL or broadcast POLL packet shall be sent.

After having sent an access request, the parked slave will listen for an unpark message from the master. As long as no unpark message is received, the slave will repeat the access requests in the subsequent access windows. After the last access window (there are M_{access} windows in total, see 8.10.8.4.2), the parked slave shall listen for an additional N_{poll} time slots for an unpark message. If no unpark message is received within N_{poll} slots after the end of the last access window, the slave may return to sleep and retry an access attempt after the next beacon instant.

After having received the unpark message, the parked slave matching the PM_ADDR or BD_ADDR will leave the park mode and enter the active mode. It will keep listening to the master until it is addressed by the master through its AM_ADDR. The first packet sent by the master shall be a POLL packet. The return packet in response to the POLL packet confirms that the slave has been unparked. If no response packet from the slave is received for *newconnectionTO* number of slots after N_{poll} slots after the end of the last access window, the master will send the unpark message to the slave again. If the slave does not receive the POLL packet for *newconnectionTO* number of slots after N_{poll} slots after the end of the last access window, it will return to park, with the same beacon parameters. After confirming that the slave is active, the master decides in which mode the slave will continue.

8.10.8.4.7 Broadcast scan window

In the beacon train, the master can support broadcast messages to the parked slaves. However, it may extend its broadcast capacity by indicating to the parked slaves that more broadcast information is following after the beacon train. This is achieved by a special LMP command ordering the parked slaves (as well as the active slaves) to listen to the channel for broadcast messages during a limited time window. This time window starts at the beacon instant and continues for the period as indicated in the LMP command sent in the beacon train.

8.10.8.5 Polling schemes

8.10.8.5.1 Polling in active mode

The master always has full control over the piconet. Due to the stringent TDD scheme, slaves can only communicate with the master and not to other slaves. In order to avoid collisions on the ACL link, a slave is only allowed to transmit in the slave-to-master slot when addressed by the AM_ADDR in the packet header in the preceding master-to-slave slot. If the AM_ADDR in the preceding slot does not match, or an AM_ADDR cannot be derived from the preceding slot, the slave is not allowed to transmit.

On the SCO links, the polling rule is slightly modified. The slave is allowed to transmit in the slot reserved for its SCO link unless the (valid) AM_ADDR in the preceding slot indicates a different slave. If no valid

Copyright © 2002 IEEE. All rights reserved.