

[0059] The mode 1 for the proposed system is conventional Bluetooth operation, which is described in detail in *Specification of the Bluetooth System*, Version 1.0A, Jul. 26, 1999, which is incorporated herein by reference.

[0060] FIG. 3 summarizes the parameters for mode 2 and also compares it to mode 1. An exemplary symbol rate for mode 2 is 0.65 Msymbols/sec. (other rates are also available) giving a bit rate of 2.6 Mbits/sec for 16 QAM (16-ary quadrature amplitude modulation) and 3.9 Mbits/sec. for 64 QAM (64-ary quadrature amplitude modulation). The transmit spectrum mask for mode 2 can be, for example, the same as Bluetooth, as shown in FIG. 4. For FIG. 4, the transmitter is transmitting on channel M and the adjacent channel power is measured on channel N. The FIG. 4 spectrum mask can be achieved, for example, by a raised cosine filter of  $\alpha=0.54$  and a 3 dB bandwidth of 0.65 MHz for the symbol rate of mode 2.

[0061] In one example of operation in mode 1 and mode 2, a Bluetooth master and slave first synchronize to each other and communicate using mode 1 and then enter mode 2 upon negotiation. FIG. 5 shows an exemplary transition diagram for the master and slave to enter and exit mode 2. The entry into and exit from mode 2 is negotiable between the master and slave.

[0062] An exemplary frame format structure for master to slave and slave to master transmissions in mode 2 is similar to mode 1 and is shown in FIG. 6. In one example the preamble consists of the pattern  $(1+j)^*$   $\{1, -1, 1, -1, 1, -1, 1, -1, 1, -1, 1, -1, 1, -1, 1, -1, 1, -1, 1, -1\}$ , which aids in the initial symbol timing acquisition of the receiver. The preamble is followed by the 64 bit Bluetooth sync. word transmitted using quadrature phase shift keying (QPSK), implying a 32 symbol transmission in mode 2. The sync. word is followed by the 54 bit Bluetooth header transmitted using QPSK, implying 27 symbols in mode 2. The farthest constellations in the 16/64 QAM are employed for the transmission of the preamble, sync. word and header as shown in FIG. 6A. The header is followed by a payload of 1 slot or up to 5 slots, similar to Bluetooth. The maximum number of bits in the payload is thus 7120 bits for 16 QAM transmission and 10680 bits for 64 QAM transmission.

[0063] The master can communicate with multiple slaves in the same piconet, some slaves in mode 2 and others in mode 1, as shown in the exemplary WPAN of FIG. 7. The timing diagram of FIG. 8 shows an example for a Bluetooth SCO HV1 link (i.e., mode 1) between the master M and slaves  $S_1$  and  $S_2$ , with slave  $S_2$  communicating with the master in mode 2 (see also FIG. 7).

[0064] A block diagram of exemplary receiver algorithms for acquisition and packet reception in mode 2 is shown in FIG. 9, and an exemplary receiver block diagram for supporting mode 2 is shown in FIG. 10. In FIG. 10, the A/D converter can sample the incoming symbols at, for example, 2 samples/symbol, implying a 1.3 MHz sampling rate. An exemplary transmitter block diagram for supporting mode 2 is shown in FIG. 11. Several blocks can be shared between the transmitter (FIG. 11) and the receiver (FIG. 10) to reduce the overall cost of a transceiver for mode 2. Similarly, several blocks of the mode 2 transmitter and mode 2 receiver can be used also for mode 1, thereby reducing the overall cost of implementing a transceiver for combined mode 1+mode 2.

[0065] A convolutional code of rate  $\frac{1}{2}$ ,  $K=5$  is used at 101 in the example of FIG. 10 to improve the packet error rate performance in the presence of automatic repeat requests (ARQ). Whenever the CRC of a packet is detected in error at 102, the transmitter sends the parity bits in the retransmission. The receiver combines the received data across packets in the Viterbi decoder to improve the overall performance of the receiver. A flow diagram of an exemplary scheme is shown in FIG. 12.

[0066] In the example of FIG. 12, the original data bits and corresponding CRC bits are encoded (e.g., using convolutional coding) at 120 to produce an encoded result that includes the original data bits and corresponding CRC bits, plus parity bits (redundant overhead bits) generated by the encoding algorithm. After the encoding operation at 120, only the original data bits and corresponding CRC bits are initially transmitted at 121. If the CRC at the receiver does not check correctly, then a retransmission is requested at 122. In response to the retransmission request, the parity bits associated with the previously transmitted data bits are transmitted at 123. At the receiver, the received parity bits are mapped into corresponding data and CRC bits using conventional techniques at 125. If the CRC of the data bits produced at 125 is correct at 124, these data bits are then passed to a higher layer. If the CRC does not check correctly at 124, then the received parity bits are combined with the associated data bits plus CRC bits (earlier-received at 121) for Viterbi decoding at 126. Thereafter, at 127, if the data bits and corresponding CRC bits generated by the Viterbi decoding algorithm produce a correct CRC result, then those data bits are passed to a higher layer. Otherwise, the data bits that were received at 121 are discarded, and a retransmission of those data bits is requested at 128.

[0067] The original data bits and corresponding CRC bits are then retransmitted at 129 and, if the CRC checks, the data bits are passed to higher layer. Otherwise, the retransmitted data bits plus CRC bits are combined with the parity bits (earlier-received at 123) for Viterbi decoding at 1200. If the data bits and corresponding CRC bits generated at 1200 by the Viterbi decoding algorithm produce a correct CRC result at 1201, then those data bits are passed to a higher layer. Otherwise, the parity bits that were transmitted at 123 are discarded, and retransmission of the parity bits is requested at 1202. Thereafter, the operations illustrated generally in the flow from 123 through 1202 in FIG. 12 can be repeated until the CRC for the data bits checks correctly or until a predetermined time-out occurs.

[0068] FIG. 12A diagrammatically illustrates pertinent portions of an exemplary transceiver embodiment which can implement receiver operations described above with respect to FIG. 12. The incoming packet data including, for example, the received version of the original data bits and corresponding CRC bits, is buffered at 1204 and is also applied to CRC decoder 1205. In response to the CRC decoding operation, a controller 1206 generates either a negative (NAK) or positive (ACK) acknowledgment in the form of an ARQ packet for transmission to the other end. If the CRC checks correctly (ACK), then the controller 1206 signals buffer 1204 to pass the buffered data to a higher layer. On the other hand, if the CRC did not check correctly (NAK), then, in response to the negative acknowledgement, the other end will transmit the parity bits, which are input to the controller 1206 and buffered at 1204. The controller