information of 14 bytes that take 14×8/11=10.2 μs to transmit. **FIG. 5** depicts an IEEE 802.11 packet transmission.

[**0080**] As shown in **FIG. 6**, an IEEE 802.11 forward data packet **500** consists of a preamble **504**, a MAC header **506** and a data field **508**. If received correctly, the receiver, responds with an acknowledgement packet **502** after a SIFS period. The latter packet consists of a preamble **510** and an acknowledgement field **512** comprising MAC information.

[**0081**] There are thus 4 scenarios to consider: there are two possible IEEE preamble lengths (96 and 192 μs); and there are either two or four Bluetooth "idle" periods (two and four slots).

[**0082**] The scenario where two Bluetooth slots are available for transmission for IEEE transmissions having a long preamble is considered.

[**0083**] The overhead due to preambles, SIFS, and MAC overhead amounts to [2×192]+10+[(28+14)×(8/11)]=424.5 μs. Of the two idle slots, it is permissible only to use 625+366=991 μs according to the Bluetooth specification. This is to leave 625−366=259 μs to allow the radio system to hop to the frequency of the next slot. Subtract 424.5 from 991, to get 566.5, which is the time left for actual data transmission at 11 Mbit/s. In this time 566.5/(8/11)=779 IEEE 802.11 bytes can be transmitted. This data can be transmitted every 4 slots. Hence the effective bit rate is equal to (8×779)/(4×625)=2.5 Mbit/s.

[**0084**] The scenario where four Bluetooth slots are available for transmission for IEEE transmissions having a long preamble is now considered.

[**0085**] If four Bluetooth slots are available, then the time for payload transmission is equal to payload time 625×3+366−424.5=1817. This Equates to 1817/(8/11)=2498 IEEE 802.11 CCK bytes. The equivalent bit rate is now (8×2498)/(6×625)=5.33 Mbit/s.

[**0086**] If the calculations are repeated for short IEEE 802.11 preambles, the bit rates are 3.33 Mbit/s for an HV2 connection or for two HV3 connections. For a single HV3 connection the bit rate is 5.89 Mbit/s. The results are summarised in Table 1.

TABLE 1

IEEE 802.11 throughput	Two Slots	Four Slots
Short preamble	3.33 Mbit/s	5.89 Mbit/s
long preamble	2.49 Mbit/s	5.33 Mbit/s

[**0087**] Table 1 shows IEEE 802.11 user throughputs if IEEE 802.11 packets are transmitted in slots that are left idle by Bluetooth. If there is one HV2 connection or two HV3 connections, there are 2 idle slots to transmit. If there is one HV3 connection, there are 4 idle slots to transmit. If there is on HV1 or DV1 connection there are no idle slots. If there is no SCO connection at all, then all slots are available for transmission, and the theoretical IEEE 802.11 maximum of 11 Mbit/s can be achieved.

[**0088**] If a Bluetooth ACL packet must be transmitted, the interoperability device **106** simply holds back IEEE 802.11 packets. As the ACL packets are none real time data packets, they can be held back. When a Bluetooth ACL packet is to be transmitted, an IEEE 802.11 packet transmission will not be in progress, as the ACL connection would be in PARK mode if an IEEE transmission was in progress, as discussed hereinabove.

[**0089**] In an alternative formulation, if a Bluetooth ACL packet transmission or reception is in progress, the IEEE 802.11 transmission is held back until the Bluetooth transmission/reception is completed. Then the Bluetooth ACL connection is put in HOLD or PARK mode, and the IEEE802.11 transmission can be scheduled and organised around SCO transmissions, as described above.

[**0090**] Optionally, the interoperability device has a further mode in which it will not allow the IEEE 802.11 devices and Bluetooth device to receive in parallel. By not allowing this, only one radio will be operating at a given time, which implies that the radio hardware can be reused. This again results in an architecture as shown in **FIG. 2**. In this mode Bluetooth SCO slots are always received. If neither the Bluetooth nor the IEEE 802.11 transmitter need to transmit, the common receiver listens to either Bluetooth or IEEE 802.11 packets, according to an algorithm.

[**0091**] Such an algorithm may be static; for instance the receiver listens to IEEE 802.11 in odd slots and to Bluetooth packets in even slots. Also given the distribution of traffic between Bluetooth and IEEE802. 11, the algorithm could give preference to one over the other.

[**0092**] Finally, the receiver may have a dual synchronisation mode, where it listens to the channel, detects on the fly what type of packet is in the medium (Bluetooth or IEEE 802.11), and reports this to the receiver, which will switch to the appropriate reception mode.

[**0093**] Both IEEE 802.11 and Bluetooth Packets may be longer than a single slot. In that case the receiver attempts to receive the packet until completion.

[**0094**] In a typical embodiment of the invention, the MAC controller of the IEE802.11 device and the baseband controller of the Bluetooth device may be implemented in separate, dedicated processor chips. The interoperability device's functionality may be implemented in an additional chip. Alternatively, the functionality of the interoperability device can be added to the controller chips of either the Bluetooth or the IEE802.11 device. In a still further alternative, it is possible to integrate the IEEE 802.11 MAC control functions and the Bluetooth control function in a single chip and add the interoperability functionality to the same chip as well. Other arrangements of chips and division of interoperability functionality are also possible.

[**0095**] **FIG. 6** illustrates an example of a "system on a chip" implementation of a combined IEEE 802.11 MAC controller and a Bluetooth Baseband controller. The chip **600** includes a DMA (Direct Memory Access) **610**, an interrupt controller (Int. Ctrl) **612**, timers **614**, RAM (Random Access Memory) **616** all connected to a CPU (central processor unit) **622** via an internal bus **624**, which elements are all required for both the IEEE 802.11 and Bluetooth functions. An external bus (Ext. Bus) block **608** is also required for both the IEEE 802.11 and Bluetooth functions, and is connected to the CPU **622** via internal bus **624** and to an external flash memory and/or ROM via lines **626**. A USB