Bluetooth—A New Low-Power Radio Interface Providing Short-Range Connectivity

JAAP C. HAARTSEN, MEMBER, IEEE, AND SVEN MATTISSON, MEMBER, IEEE

In the past decades, progress in microelectronics and VLSI technology has fostered the widespread use of computing and communication applications in portable electronic devices. In this paper, we review the Bluetooth technology, a new universal radio interface enabling electronic devices to connect and communicate wirelessly via short-range connections. Motivations for the air interface design and radio requirement decisions are discussed. Frequency hopping, interference resistance, and the concepts of ad hoc connectivity and scatternets are explained in detail. Furthermore, Bluetooth characteristics enabling low-cost single-chip implementations and supporting low power consumption are discussed.

Keywords*—*Ad hoc radio networks, low-power transceivers, short-range radio.

I. INTRODUCTION

Imagine a cheap, power-efficient radio chip that is small enough to fit inside any electronic device or machine, provides local connectivity, and creates a (worldwide) microscale web. What applications might you use it in? Current portable devices use infrared links (IrDA) to communicate with each other. Although infrared transceivers are inexpensive, they have a limited range (1–2 m), require direct line-of-sight, are sensitive to direction, and can in principle only be used between two devices. In contrast, radios have much greater range, can propagate around objects and through various materials, and connect to many devices simultaneously. In addition, radio interfaces do not require user interaction: connections can be established without requiring any particular user knowledge (hidden computing, automatic synchronization of files, calendars, and so on).

At the end of 1997, several companies in the communications and PC industries identified the desire for local connectivity between electronic devices. A single standard for short-range radio connectivity will ensure interoperability between devices of different manufacturers. In February 1998, five major telecom and PC companies—Ericsson, Nokia, IBM, Toshiba, and Intel—formed a special interest group (SIG) to create a standard radio interface to fulfil this desire. The radio interface was named Bluetooth after a Danish Viking king Harald Blatand from the tenth century who united Denmark and Norway. This group was further expanded in December 1999 with 3Com, Lucent, Microsoft, and Motorola. In addition to these nine promoter companies, more than a thousand companies have joined as adopters of the Bluetooth technology. A year and half after its foundation, the Bluetooth SIG published the first version of the Bluetooth specification incorporating both radio protocols and control software [1], enabling manufacturers to start designing radio equipment and applications. The first Bluetooth products will emerge in mid-2000 and focus on mobile applications (mobile phones, notebook computers, and accessories; see Fig. 1). Conservative estimates foresee several hundred million Bluetooth-enabled devices in the next five years.

II. BLUETOOTH AIR INTERFACE

The focus of user scenarios envisioned for first-generation products is typically on traveling business people: portable devices that contain Bluetooth radios would enable them to leave cables and connectors at home. Before the air interface for Bluetooth could be designed, however, certain requirements had to be satisfied.

- The system must operate globally.
- The system must support peer connectivity—i.e., there is no wired infrastructure to provide call setup and networking functions: connections are made on an *ad hoc* basis.
- The connection must support voice and data—e.g., for multimedia applications.
- The radio transceiver must be small and operate at low power—i.e., the radio must fit into small, portable devices, such as mobile phones, headsets, and personal digital assistants (PDAs).

A. License-Free Band

To operate worldwide, the required frequency band must be available globally. Further, it must be license-free and open to any radio system. One frequency band that satisfies such requirements is at 2.45 GHz—the Industrial-Scientific-

---

Manuscript received December 31, 1999; revised June 5, 2000.

J. C. Haartsen is with Ericsson Radio Systems B.V., 7801 CA, Emmen, The Netherlands (e-mail: jaap.haartsen@erh.ericsson.se).

S. Mattisson is with Ericsson Mobile Communications AB, SE-221 83 Lund, Sweden (e-mail: sven.mattisson@ecs.ericsson.se).

Publisher Item Identifier S 0018-9219(00)09137-4.

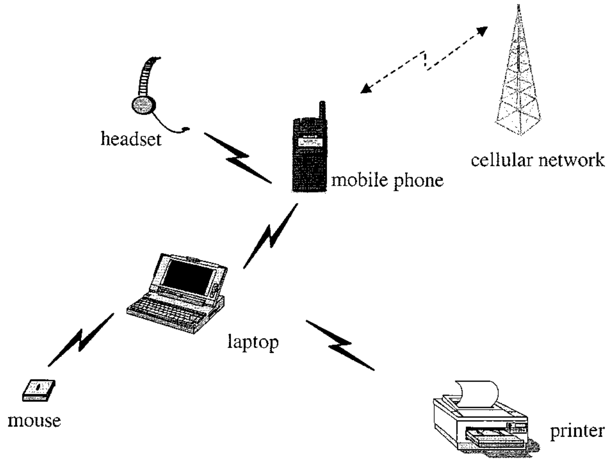


Fig. 1. Bluetooth applications envisioned for the near future.

Medical (ISM) band, which ranges from 2,400 to 2,483.5 MHz in the United States, Japan, and Europe (note, Spain still only allows part of the ISM band to be used for unlicensed operations but is expected to harmonize with the European rules soon).

B. Frequency Hopping

Since the ISM band is open to anyone, radio systems operating in this band must cope with several unpredictable sources of interference, such as baby monitors, garage door openers, cordless phones and microwave ovens (the strongest source of interference). Interference can be avoided using an adaptive scheme that finds an unused part of the spectrum, or it can be suppressed by means of spectrum spreading. In the United States, radios operating in the 2.45-GHz ISM band are required to apply spectrum-spreading techniques if their transmitted power level exceeds 0 dBm [2]. Bluetooth radios use frequency-hop (FH) spread spectrum, since it better supports low-cost, low-power radio implementations. In addition, they better cope with near-far problems: a nearby jammer is effectively suppressed by the narrow channel filter as long as its jammer TX spectrum does not coincide with the selected hop channel. FH systems divide the frequency band into several hop channels. During a connection, radio transceivers hop from one channel to another in a pseudorandom fashion. The instantaneous (hop) bandwidth is small in FH radios, but spreading is obtained over the entire frequency band. This results in low-cost narrow-band transceivers with maximum immunity to interference. Occasionally, interference jams a hop channel, causing faulty reception. When this occurs, error-correction schemes in the link restore the bit errors.

C. Channel Definition

Bluetooth channels use a FH/time division duplex (FH/TDD) scheme (Fig. 2). The channel is divided into consecutive slots, each slot lasting 625 μs; a different hop channel is used for each slot. This gives a nominal hop rate of 1600 hops/s. One packet can be transmitted per slot. Subsequent slots are alternately used for transmitting and receiving, which results in a TDD scheme. TDD has been selected since 1) it better matches the peer communication concept; i.e., there is no apparent uplink and downlink or other distinguishing factors that would support frequency division duplex (FDD); 2) only a single, consecutive 79-MHz-wide radio band at 2.45 GHz is defined; and 3) it simplifies implementations (see, also, Section V).

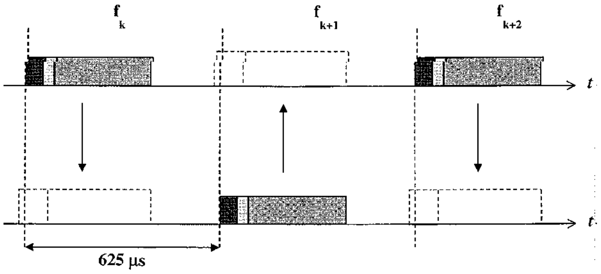


Fig. 2. Bluetooth frequency-hop/time division duplex channel.

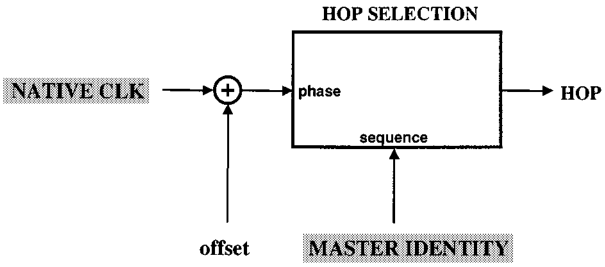


Fig. 3. Hop channel selection scheme: In the selection box, the master identity selects the sequence and the clock selects the phase to result in the current hop channel to be used.

Two or more units sharing the same channel form a piconet (piconet operation is discussed in greater detail below). One unit acts as a master, controlling the traffic on the piconet. The other units are slaves. The FH channel is determined by the FH sequence (the order in which the hop channels are visited) and by the phase in this sequence. In Bluetooth, the sequence is determined by the identity of the piconet master and phase is determined by the master unit's system clock (Fig. 3). In order to recreate the master clock in the slave unit, the slave may add an offset to its own native clock. The FH sequence is very long and its repetition time exceeds 23 h. If every participant on a given channel uses the same identity and clock as input to the hop-selection box, then each unit will consistently select the same hop channel and remain synchronized. Every piconet has a unique set of master parameters, which create a unique channel.

The channel makes use of several, equally spaced, 1-MHz hop channels. With Gaussian-shaped frequency shift keying (FSK) modulation, a symbol rate of 1 Mb/s is achieved. In the 2.45-GHz ISM band, 79 hop channels have been defined (Table 1). On average, the FH sequence visits each carrier with equal probability.

D. Packet Definition

In each slot, a packet can be exchanged between the master unit and one of the slaves. Packets have a fixed format (Fig. 4). Each packet begins with a 72-bit access

Table 1
Bluetooth Radio Parameters

PARAMETERS	VALUES
MODULATION	G-FSK, $h \leq 0.35$
PEAK DATA RATE	1 Mb/s
RF BANDWIDTH	220 kHz (-3dB), 1 MHz (-20 dB)
RF BAND	2.4 GHz, ISM band
HOP CHANNELS	79
CARRIER SPACING	1 MHz
PEAK TX POWER	≤ 20 dBm

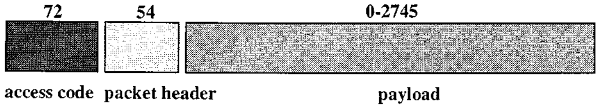


Fig. 4. Bluetooth air packet format.

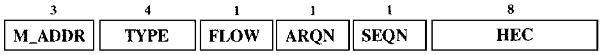


Fig. 5. Packet header fields.

code, which is derived from the master identity and is unique for the channel. Every packet exchanged on the channel is preceded by this access code. Recipients on the piconet match the incoming signals against the access code. If the two do not match, the received packet is not considered valid on the channel and the rest of its contents are ignored. Besides packet identification, the access code is also used for synchronization and compensating for frequency offset. The access code is very robust and resistant to interference. Correlation of the access code by recipients provides similar processing gains as direct-sequence spreading.

A header trails the access code. It contains important control information, such as a 3-bit slave address, packet type, flow control bits, bits for the automatic retransmission query (ARQ) scheme, and a header-error-check (HEC) field (Fig. 5). The ARQ is based on a stop-and-wait scheme with a minimal wait period: one slot. The success or failure of a packet is directly revealed in the header of the return packet. The information in the 1-bit ARQN field (acknowledge, ACK, or not-acknowledge, NAK) in the header of the received packet indicates whether the payload just transmitted was successful or not (Fig. 6). Based on this information, the transmitter can send a new payload or has to retransmit the previous payload. The payload of the received packet is checked for errors in the CRC check. Based on the success or failure of this received payload, the ARQN field in the header of the return packet is set at ACK or NAK. Since there is only about 220-μs delay between the reception of the last bit of the received packet and the transmission of the first bit of the return packet, the ARQ processing must be carried out in real-time (if a piconet contains several slaves, the delay between ARQ reception and transmission in the master may be longer, depending on the traffic scheduling). The selection of transmitting a new payload or retransmitting the previous payload, based on the received ACK/NAK information, and the setting of the ACK/NAK field in the transmitted packet, based on the success of the received payload, is illustrated in Fig. 6. In the ARQ protocol, a 1-bit SEQN field is included to distinguish retransmissions from new transmissions: for every new transmission, a SEQN bit is inverted. Retransmissions of already correctly received packets can then be filtered out at the recipient. This stop-and-wait protocol is as effective as a selective-repeat ARQ protocol in that only erroneous packets are retransmitted. Yet, the overhead is minimal. The header, whose length is fixed at 54 bits, is protected by a one-third rate forward-error-correction (FEC) code.

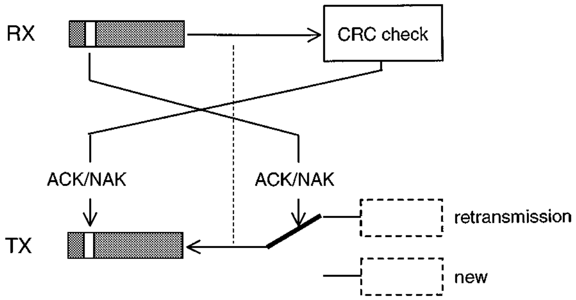


Fig. 6. Illustration of procedures applied in the ARQ protocol.

Payload may or may not trail the header. The length of the payload may vary from 0 to 2745 bits. To support high data rates, multislot packets have been defined. A packet can cover one slot, three slots, or five slots. Packets are always sent on a single hop channel. For multislot packets, the hop channel is used as applied in the first slot. After the multislot packet, the channel continues on the hop channel as dictated by the master clock. The type of bits in the header indicate what type of packet is used (i.e., traffic or control, one-slot or multislot, FEC applied or not).

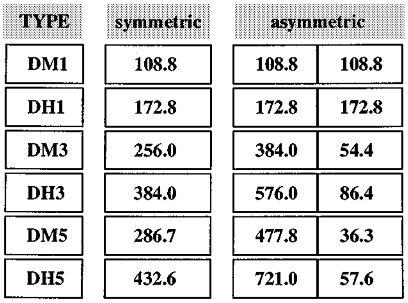
E. Physical Link Definition

Two types of links have been defined to support multimedia applications that mix voice and data:

- synchronous connection-oriented (SCO) link;
- asynchronous connectionless (ACL) link.

SCO links support symmetrical, circuit-switched, point-to-point connections typically used for voice. These links are defined on the channel by reserving two consecutive slots (forward and return slots) with a fixed period. Reservation is carried out by the master and the slave when the link is set up. ACL links support symmetrical or asymmetrical, packet-switched, point-to-multipoint connections typically used for bursty data transmission. Master units use a polling scheme to control the ACL connections and to prevent collisions on the channel when multiple slaves should transmit. All SCO and ACL traffic on the channel is scheduled by the master. Local slave addresses in the packet header indicate the recipient and only this recipient is allowed to respond in the next slave-to-master slot. The ACL link is constantly present between the master and the slave as long as the piconet exists; the ACL link conveys

Table 2
Achievable Data Rates (kb/s) on the ACL Link Using Different Packet Types



TYPE	symmetric	asymmetric	
DM1	108.8	108.8	108.8
DH1	172.8	172.8	172.8
DM3	256.0	384.0	54.4
DH3	384.0	576.0	86.4
DM5	286.7	477.8	36.3
DH5	432.6	721.0	57.6

both control information and asynchronous data services. In contrast, SCO links can be set up and released dependent of the need for synchronous services.

A set of packets has been defined for each physical link; the type of bits in the header indicate what packet is used.

- For SCO links, three kinds of single-slot voice packet have been defined, each of which carries voice at a rate of 64 kb/s. Voice is sent unprotected, but if the SCO period is decreased, a forward-error-correction rate of two-thirds or one-third can be selected.
- For ACL links, single-slot, 3-slot, and 5-slot data packets have been defined. Data can be sent either unprotected or protected by a two-thirds forward-error-correction rate. The maximum data rate—723.2 kb/s in one direction and 57.6 kb/s in the reverse direction—is obtained from an unprotected, 5-slot packet. Table 2 summarizes the data rates that can be obtained from ACL links. DMx represents x-slot, FEC-encoded data packets and DHx represents unprotected data packets.

Fig. 7 depicts mixed SCO and ACL links on a piconet with one master and two slaves. Slave 1 supports an ACL link and an SCO link with a six-slot SCO period. Slave 2 only supports an ACL link. Note, slots may be empty when no data is available.

F. *Interference Immunity*

The Bluetooth radio must operate in an open band that is subject to uncontrolled interference. Interference immunity is provided by the following features.

- Frequency hopping techniques are applied with a high hopping rate and short packet lengths (1600 hops/s for single-slot packets). If a packet is lost, only a small portion of the message is lost.
- Packets can be protected by forward error control.
- Data packets carried by the ACL link are protected by an ARQ scheme in which lost data packets are automatically retransmitted. The recipient checks each received packet for errors. If errors are detected, it indicates this in the header of the following return packet. This results in a fast ARQ scheme—delays are only one slot in duration, and only lost packets are retransmitted.
- Voice carried over the SCO link is never retransmitted. Instead, a robust voice-encoding scheme is used. The scheme, which is based on continuous variable slope delta (CVSD) modulation, follows the audio waveform and is very resistant to bit errors.

III. *AD HOC* CONNECTIVITY

Bluetooth is based on peer connectivity: a device carrying a Bluetooth radio can make a connection to any other device carrying a Bluetooth radio. There is no wired infrastructure with base stations or access points that can support the call setup or can provide low-power modes. This puts special demands on the design of the connection establishment procedures, which must combine a short setup time with a low-power standby mode.

A. *Establishing Connections*

When units are not connected to any other unit, they are in the standby mode. In this mode, they do not transmit but only listen to the hop channels with a very low duty cycle. A unit starts to transmit when it desires to make a connection (which may be induced by a user interaction). In that case, it enters the page or inquiry state in which it broadcasts page or inquiry messages as explained later. In the standby mode, the Bluetooth radio periodically listens for these page or inquiry messages. At wake-up, the unit listens for about 11 ms on a particular hop channel. The interval between wake-up events ranges between 0 and 3.84 s, so the duty cycle in standby is well under 1%. From the total set of 79 hop channels, a subset of 32 unique wake-up channels has been defined. This reduces the frequency uncertainty between pager and standby unit by more than 50% (from 79 hop channels to 32 hop channels). The subset is chosen pseudorandomly. Over these wake-up carriers, a wake-up sequence visits each hop channel once: the sequence length is 32 hop channels. Both the hop channel set and the sequence are determined by the standby unit identity; therefore, each unit has a different wake-up sequence with a different set of hop channel. In consecutive wake-up events, consecutive hop channels in the wake-up sequence are used: after 32 wake-up events, the standby unit has cycled through all wake-up hop channels. The native clock of the standby unit determines the phase in the wake-up sequence. During the wake-up period, the unit listens on a single wake-up hop channel and correlates the incoming signal with the access code derived from its own identity. If the correlator triggers—that is, if most of the received bits match the access code—the unit is activated and invokes a connection-setup procedure. Otherwise, the unit returns to sleep until the next wake-up event.

Units trying to connect to a unit in standby mode must send the standby unit's access code on the proper hop channel to be heard. This access code forms the page message. Therefore, the pager must know the standby unit's identity and preferably its native clock

- to generate the required access code;
- to derive the wake-up sequence and the associated hop channels;
- to predict the phase in this sequence.

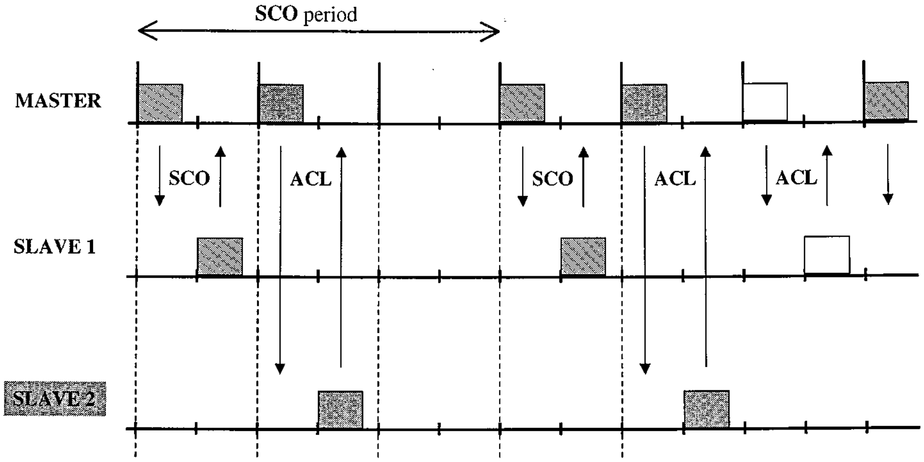


Fig. 7. Example of SCO and ACL link mixing on a single piconet channel (each slot is on a different hop channel).

Since paging units cannot accurately know the native clock of a recipient, they must resolve the time-frequency uncertainty. This uncertainty resolution is intentionally placed at the pager since it requires quite some time and power. Since a unit is in standby most of the time and only occasionally enters the page state to connect to another unit, placing the burden at the pager is preferred from a power consumption point of view. That units only listen while in standby and do not send beacons or other control information supports both the power savings issue and avoids wasteful interference. Although the pager knows the wake-up hop sequence of the recipient (since this sequence is associated with its address), it may not know where the recipient is in the sequence, or when the next wake-up event is scheduled. Therefore, the pager transmits the access code continuously—not only in the hop channel they expect the recipient to wake up in, but also in the hop channels before and after. For a period of 10 ms, paging units transmit the same access code sequentially on different hop channels centered around the expected hop channel. Since the access code is a short message, this message can be repeated many times on different hop channels during the 10-ms period. This 10-ms train of access codes is transmitted continuously until the recipient responds or a time-out is exceeded. When a paging unit and recipient simultaneously select the same wake-up carrier, the recipient receives the access code and returns an acknowledgment. The paging unit then sends a packet containing its identity and its current clock. After the recipient acknowledges this packet, each unit uses the paging unit's parameters for hop channel selection—a piconet has thus been established, where the paging unit acts as the master.

Before a connection can be established, the paging unit must know the identity of the recipient. To obtain the identity of units within transmission range, the paging unit may execute an inquiry procedure prior to any connection establishment. This procedure is similar to the paging procedure: the inquiring unit transmits an inquiry access code on the inquiry wake-up channels according to an inquiry sequence. The inquiry access code, the inquiry wake-up channels, and the inquiry sequence are all common to all Bluetooth devices; i.e., they are derived from a single inquiry identity, which is a reserved identity in the Bluetooth specification. The inquiry wake-up interval ranges between 0 and 2.56 s. When a recipient receives the inquiry message (i.e., the inquiry access code), it returns a packet containing its identity and clock—the very opposite of the paging procedure. After having gathered each response, the inquiring unit can then select a specific unit to page (Fig. 8).

B. Piconets

Bluetooth units that are within range of each other can set up *ad hoc* connections using the procedures described in the previous paragraph. Two or more Bluetooth units that share a FH channel form a piconet. To regulate traffic on the channel, one of the participating units becomes a master of the piconet. Any unit can become a master, but by definition, the paging unit that establishes the piconet is assigned the master role. All other participants are slaves. Participants may change roles, if one of the slave units wants to take over the master role. This is accomplished via a master-slave switch procedure [1]. At any one time, there can be only a single master per piconet since the master identity and clock specify the piconet channel parameters like hop sequence and the access code preceding all packets. When the piconet is released, the master and slave roles are lost. Every unit in the piconet uses the master identity and clock to track the hopping channel. Each unit also has its own, free-running, native clock. When a connection is established, a clock offset is added to synchronize the slave clock with the master clock. Each time a slave receives a packet from the piconet master carrying a valid access code, it readjusts the offset value to remain synchronized to the master clock. The native clock is never adjusted, however, and offsets are solely valid for the duration of the connection.

Master units control all traffic on a channel. They allocate capacity for SCO links by reserving slots. For ACL links, they use a polling scheme. A slave is only permitted to send in the slave-to-master slot when it has been addressed by its slave address in the preceding master-to-slave slot. A master-to-slave packet implicitly polls the slave. If no infor-

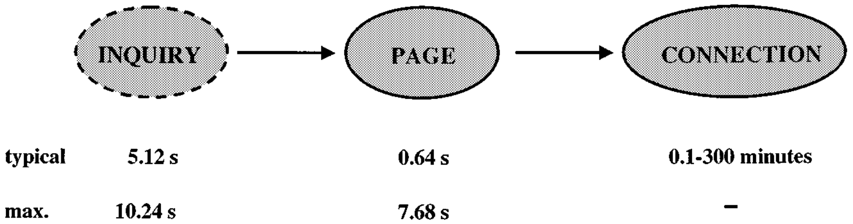


Fig. 8. Connection establishment procedure and response times.

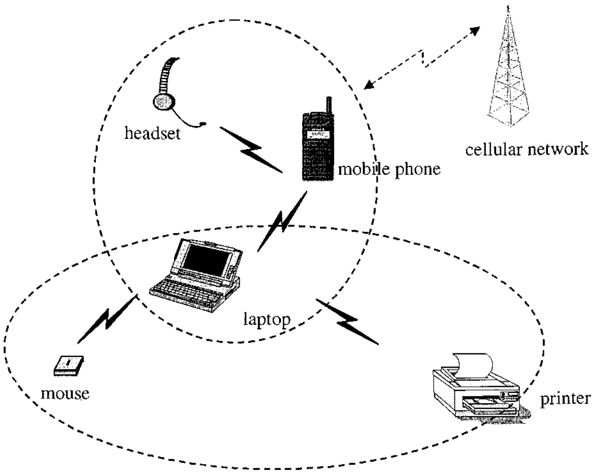


Fig. 9. Scatternet of two piconets. The laptop participates in both piconets.

mation is available to send to the slave, the master can use a POLL packet to poll the slave explicitly. POLL packets consist of an access code and header only. This central polling scheme prevents collisions between slave transmissions on the piconet channel. The way the master polls the slaves and schedules the traffic depends on the services and traffic demands for each slave. Slaves with demanding applications regarding response time and throughput will be polled more often than slaves with less requiring services. The master can dynamically adjust its scheduling algorithm.

C. Scatternet

Users on the same channel must share capacity. Since the channel capacity is only 1 MHz, as more and more users are added, throughput per user quickly drops to less than some 10 kb/s. The spectral bandwidth available is 79 MHz, but cannot be used effectively when every unit must share the same 1-MHz hop channel. Therefore, another solution has been adopted. Units that share the same area and that are within range of one another can potentially establish *ad hoc* connections between themselves. However, solely units that truly want to exchange information share the same 1-MHz channel of a piconet. This solution permits several piconets to be created with overlapping coverage areas. Each piconet channel applies its own pseudorandom hopping sequence through the 79-MHz medium. The piconets are uncoordinated and hop independently. Within the piconet, the participants have to share the 1 MHz, but multiple piconets share the entire 79 MHz, thus increasing the aggregate capacity.

A collection of multiple piconets is called a scatternet. Aggregate and individual throughput of users in a scatternet is much greater than if all users would participates on the same piconet with a 1-Mb/s channel. In the scatternet, the radio medium of 79 MHz is shared; in a single piconet, the 1-Mb/s channel is shared. Additional gains are obtained by statistically multiplexing packet transmissions (i.e., the instantaneous occupancy of the hop channels is determined by the traffic characteristics) and by the geographical separation of piconet masters and slaves. As the number of piconets increases, performance in the FH system degrades gracefully. Simulations of a scatternet made up of 10 piconets indicate that reduction in throughput per piconet is less than 10%. A strongly simplified throughput calculation is given by

$$TH = \left(1 - \tfrac{1}{79}\right)^{N-1} \tag{1}$$

which is the normalized throughput TH per piconet assuming N piconets co-located. Equation (1) assumes a worst-case scenario wherein each collision is catastrophic (i.e., all transmissions are lost) and power and distance dependencies are ignored.

The information exchanged on the piconet channel is only shared by the piconet participants, not by all members in the scatternet. The maximum number of units that can actively participate on a single piconet is eight: one master and seven slaves. The slave address in the packet header, which is used to distinguish each unit, is limited to 3 bits. Although the piconets in the scatternet form separate entities, connections can be made between piconets. A unit can participate in different piconets by time division duplexing. It can be a master in one piconet and a slave in another. It can also be slave in several other piconets. Instantaneously, it can only be present on a single channel. Careful packet scheduling is required when a unit participates in multiple piconets. Fig. 9 illustrates the scatternet approach applied to the scenario shown in Fig. 1. In this example, the mobile phone is master to the headset and the laptop in one piconet; the laptop is master to the mouse and the printer in another piconet. Therefore, the laptop acts both as slave and as master in the first and second piconet, respectively.

IV. POWER MANAGEMENT

Intended for providing a universal interface to portable, battery-driven equipment, the Bluetooth interface should require as little power as possible. In the Bluetooth specification, several provisions have been included to save power. First, the hopping mechanism is rather robust in that master and slave remain synchronized even if no packets are exchanged over the channel for several hundreds of milliseconds (i.e., the hopping channel is virtually present even when no transmissions take place). Therefore, no dummy data has to be exchanged to keep synchronization between master and slaves. Second, a receiver can decide quickly whether a packet is present or not. At the beginning of the receive slot, the receiver correlates the incoming signal in a sliding correlator, which is matched to the access code. Since the access code only lasts for a little more than 70 μs, after a scan duration of about 100 μs (to compensate for some timing jitter and drift) the receiver can decide to continue to listen or return to sleep. If the access code is not received (i.e., the correlator does not trigger) within the scan window, apparently no packet was sent, or was so corrupted that further reception does not make sense. The Bluetooth recipient can then sleep for the rest of the receive slot, and transmit slot if the unit is a slave. If the proper access code is received, the receiver will continue to demodulate the packet header. Checking the slave address in the header, a slave can then determine whether the packet is intended for him. If not, he does not have to read the payload. Alternatively, if the slave address matches, the packet type indication tells the slave whether there is a payload present and how long it may last.

The above-described measures help to reduce the power consumption at the micro level. However, Bluetooth also supports power-saving techniques at the macro level. If communication is expected to be postponed, a master can put a slave into the HOLD mode. During the HOLD period, no communication is possible between the master and the slave. When the HOLD period expires, the slave returns to the channel instantaneously, i.e., it remains synchronized to the hop channel. Alternatively, the slave can be put in the PARK mode. In this case, the slave enters a low-duty cycle mode where it periodically listens to the master at large intervals. Again, the parked slave remains synchronized and locked to the piconet channel, but can only continue communicating when unparked by the master. To support the PARK mode, the master transmits a beacon signal at regular intervals. Since the parked slave remains locked, a much lower power consumption can then be obtained in the unlocked state when the unit is not connected to any piconet channel. Finally, a SNIFF mode has been defined where the slave does not listen to the master every receive slot, but can skip some slots just to save power. When a slave is placed into the SNIFF mode, master and slave agree on which slots the slave will sniff. Only in these slots, communication can continue. The SNIFF mode can be used in applications that require little bandwidth.

When the unit is in standby, it only wakes up for about 11 ms on a single frequency (see also Section III-A describing the connection establishment). The burden of finding the unit in standby both in time and frequency is placed at the pager: Paging only occurs occasionally, whereas the periodic scanning in standby must be carried out continuously to enable connection establishment at any time.

V. IMPLEMENTATION CONSIDERATIONS

It was recognized early on that the Bluetooth system only makes sense if it supports low-cost, low-power implementations. That is, the radio and air-interface specifications must not be too restrictive from an implementation point of view and the entire transceiver must be possible to integrate in a cost-effective way on a single chip. The focus has been on providing a reasonable minimum performance with large design flexibility that can be exploited should the application require better performance than what is stipulated by the minimum requirements. Then, only demanding applications will require implementations that are more expensive while simple ones can be made very low cost.

The air interface is packet-oriented, but does not use interleaving and, thus, supports streaming signal processing in the base band, which simplifies the hardware. Duplexing is achieved by time division and in conjunction with a 220 μs TX-RX turnaround time; the need for separate TX and RX oscillators is eliminated. Further, no duplex filter is needed, which also lowers cost [4]. Finally, separating the TX and RX procedures in time prevents crosstalk from the transmitter into the receiver and allows highly integrated implementations.

The most demanding characteristics of an integrated transceiver is the sensitivity, selectivity, and output power while staying within a tight power consumption budget in a mainstream process [5]. In the design of Bluetooth, all these issues were weighted against overall performance. The initial range goal was set to 10 m, which is reasonable for a cable replacement. Further, the transmitter power should be minimized not to compromise the operational time of a battery-operated host (e.g., a cellular telephone).

A. *Link Budget*

To define the different radio parameters, the radio link budget has to be examined. This link budget is the relation between the transmitted power, P_{TX}, and the received signal power, P_{RX}. Several parameters influence the signal on its path from the transmitter to the receiver. The signal is subject to attenuation, L_{path}, due to propagation losses (path loss), and the transmitter and receiver antennas has characteristic gains relative to an ideal point source. These antenna gains are denoted by G_{TX}, and G_{RX}, respectively. Finally, the signal is also subject to multipath reflections, which will cause cancellation if the net path difference is close to an odd multiple of half a wavelength. Likewise, obstacles may also reduce the received signal strength. These losses are called fading losses, L_{fade}, and will vary substantially depending on the environment. Collecting all these parameters into one equation, where we assume decibel units throughout, we get the received signal power to be

$$P_{RX} = P_{TX} - L_{\text{path}} - L_{\text{fade}} + G_{TX} + G_{RX}. \quad (2)$$

To properly detect and decode the incoming radio signal, the receiver needs a certain minimum signal strength relative to interfering signals. If we consider thermal noise as the primary source of interference, we can define the receiver sensitivity, RX_{sens}. The received noise power depends on the noise spectral density, N_0, and the receiver noise bandwidth, BW. In addition to this noise power seen by the transmitter, the radio itself adds thermal noise due to implementation losses. These implementation losses are denoted by NF. Finally, the demodulation process requires a certain carrier-to-noise ratio, C/I_{AWGN}, and collecting all the receiver parameters, again in decibel units, we get the receiver sensitivity as

$$RX_{\text{sens}} = N_0 + BW + C/I_{AWGN} + NF. \tag{3}$$

By letting $P_{RX} = RX_{\text{sens}}$ and by letting P_{TX} assume its maximum value, the above two equations will reveal the relations between implementation parameters, like NF, and the maximum range.

Depending on the application, the path loss will vary substantially with distance, but for Bluetooth, the following relations were assumed [6]:

$$L_{\text{path}} = 20 \cdot \log\left(\frac{4\pi R}{\lambda}\right) \approx 40 + 20\log(R) \qquad R \leq 8.5 \text{ m}, \tag{4}$$

and

$$L_{\text{path}} = 36 \cdot \log\left(\frac{4\pi R}{\lambda}\right) - 46.7 \text{ dB} \approx 25.3 + 36\log(R) \qquad R > 8.5 \text{ m} \tag{5}$$

where R is the range and λ is the wavelength, both in meters. Equation (4) is equivalent to line-of-sight conditions where the direct signal path dominates. For ranges exceeding 8.5 m, it is necessary to account for objects obscuring the path (e.g., office desks and walls) and the attenuation will increase more rapidly beyond this range. For example, a range of 10 m yields a path loss of some 62 dB.

A fading margin, L_{fade}, of 8 dB was added yielding a total path loss of 70 dB for a 10-m link. This fading margin does not account for very deep fading dips. Such deep fading dips will only occur at distinct frequencies and because of the diversity provided by the frequency hopping, this will only cause occasional interruptions. If a packet is lost in a deep fade, it will be retransmitted immediately and then at a different frequency with other fading properties.

The choice of 8 dB as a fading margin is somewhat arbitrary as Bluetooth is intended for many different applications. A margin of 8 dB is comparable to what is used in some cellular systems, but in some environments (e.g., non-line-of-sight with no supporting reflections), an 8-dB fading margin will not suffice and then range will be shorter. However, for typical indoor applications, 8 dB have been considered a reasonable compromise between power consumption and versatility.

For a system like Bluetooth, the antenna characteristics will vary from one application to the next. To make the antenna requirements nonrestrictive, antenna gains of 0 dBi were assumed. If directional antennas are feasible, the range may be increased but high-gain antennas are not required. Antenna gain is also restricted in the ISM band by the regulations [2]. Inserting the above propagation loss and antenna gains into (2), we get the simplified receiver sensitivity formula in decibels to

$$RX_{\text{sens}} = P_{TX} - 70. \tag{6}$$

This relation must be balanced against (3) to get the receiver noise figure, that is the implementation margin. A higher noise figure offers more implementation margin and lower power consumption but shorter range.

The modulation scheme chosen for Bluetooth is similar to the one used for FH systems according to the IEEE 802.11 standard [3], that is GFSK with a BT product of 0.5 and a modulation index of 0.32. GFSK is a constant envelop modulation enabling limiting receivers and class C transmitters based on a direct modulated high-power VCO; both factors contributing to low-power simple receivers. A high-power VCO will efficiently deliver enough output power to directly drive the class C transmitter, and as any high-frequency gain-stage or up-conversion mixer would add significantly to the power consumption, the use of a high-power VCO lowers the total power consumption. The class C transmitter devices only conduct during a fraction of the carrier waveform period and can, thus, achieve a higher efficiency than a linear power amplifier.

For the GFSK modulation, the required detector C/I_{AWGN} is assumed to be 21 dB for 0.1% bit error rate (BER). With a 1-MHz receiver noise bandwidth, we get the receiver input noise to -174 dBm/Hz $+60$ dBHz $= -114$ dBm, and inserting this value into (3) and combining (3) and (6) we get, in decibel units

$$P_{TX} = 70 \text{ dB}-114 \text{ dBm}+21 \text{ dB}+NF = -23 \text{ dB}+NF. \tag{7}$$

A typical cellular receiver achieves a noise figure of some 8 dB and it was assumed that at least another 10 dB were needed for two reasons. First, inexpensive antenna filters, connectors, switches and matching networks add extra losses compared to components that are more expensive. Second, single-chip integration will add substantial amounts of substrate and supply interference to the receiver. The latter contributions are still largely unknown. A 1-mW antenna output power (0 dBm) is easy to accomplish, and this will result in a 23-dB receiver noise figure, for a 70-dB path loss plus fading margin, which is deemed to be sufficient for implementation losses and on-chip interference. This transmit power will not require high-capacity batteries or add substantially to the current consumption of its host. Thus, a reference sensitivity level of -70 dBm was defined in conjunction with a nominal 0-dBm transmit power.

Should an application call for a longer range or have a higher path loss, a more sensitive receiver (lower noise figure) may be employed but the specification also supports higher output power as a means to extend the range. Providing power control is used, an antenna output power up to 20 dBm may be used. This upper limit is set by ISM band restrictions in Europe and by the near-far problem in a scatternet. A high output power in, for example, a cordless handset may completely block the receiver in a Bluetooth enabled mouse or another application intended for short range only. To minimize such near-far related interference, the high-power transmitters need to regulate their power level such that the received signal strength at the destination is within some 10 dB of -50 dBm (the wide span is to provide margin for path-loss fluctuations). Should this received signal strength not suffice for good reception, the path loss has to be lowered (e.g., the range shortened) as a higher output level would just increase interference.

The power control does not eliminate the near-far problem but does reduce the unnecessary interference caused by high-power transmitters when used for a short-range connection. As a side effect, the transmitter power consumption will also be lower when the output power is regulated.

B. Dynamic Range

In many implementation studies, focus is on achieving a very low noise figure. This makes sense in range-limited applications. Bluetooth will, however, typically be interference limited rather than range limited, and in this case, it is more important to consider the spurious-free dynamic range (SFDR) [7].

In an interference-limited scenario, there are three major sources of interference. One is when a nearby transmitter is using a frequency close to the desired signal. Then, the transmitted signal may contain enough power, close to the frequency of the desired signal, to cause jamming. The selectivity filters will limit the impact of such interference. Another source of interference is when two signals mix and cause intermodulation distortion. If the distance between the two interferers and the distance between the closest interferer and the desired signal is the same, the third-order intermodulation products, IM_3, will fall on the desired signal. The third form of interference results when a nearby transmitter is strong enough to saturate the receiver. The receiver is then completely blocked and the desired signal cannot be detected.

Third-order intermodulation products will be proportional to the interference level raised to the third power. Thus, when the interference is strong enough, the intermodulation products will be approaching the levels of the fundamental tones of the interfering signals. An intercept point can be found by extrapolating intermodulation power plotted versus interference power. This intercept point is called the third-order intercept point, IP_3, and is an important parameter when describing the linearity of a radio.

By finding the interference level (assuming both interferers are received with the same signal strength) when the intermodulation products reach the same level as the receiver noise floor, N_{RX}, we get the SFDR. Assuming decibel units, we then get

$$\begin{aligned} SFDR &= \tfrac{2}{3}\left(IP_3 - N_{RX}\right) \\ &= \tfrac{2}{3}\left(IP_3 - \left(RX_{SENS} - C/I_{AWGN}\right)\right). \end{aligned} \tag{8}$$

From the path-loss discussion, we already know the noise floor and we need to find an intercept point that is compatible with realistic user scenarios and implementation limits. A 0-dBm interferer at a 1-m distance will result in a -40-dBm interference level, P_I. Two such interferers transmitting at $f_0 + \Delta f$ and $f_0 + 2\Delta f$, respectively, will generate IM_3 products at the desired radio channel, f_0. The power level of the on-channel intermodulation product will then be, in decibel units, [8]

$$P_{IM_3} = IP_3 + 3\left(P_I - IP_3\right). \tag{9}$$

When P_{IM_3} is at least C/I_{AWGN} below the desired signal, intermodulation will not cause the BER to exceed 0.1% (a desired signal weaker than RX_{SENS} will be noise limited rather than intermodulation limited). When the desired signal is 3 dB stronger than the receiver sensitivity level (i.e., at -67 dBm), we require $IP_3 \geq -16$ dBm for the BER to be 0.1% or lower. Two interferers, each at a 1 m distance (actually a geometric mean distance of 1 m), were, thus, selected as a worst case scenario. If only one interferer is close, then no significant IM_3 products will be generated, and the receiver will work properly until the interference is strong enough to cause blocking (see the section on selectivity for further details on blocking).

The intermodulation performance of a receiver is largely determined by the input stages, that is the low-noise amplifier through the first mixer. After the first mixer, the operating frequency is low enough for feedback to be employed to increase linearity ($IP_3 \propto \sqrt{1+T}$, where T is the loop gain [9]). An input-referred IP_3 of -16 dBm is typical for some cellular handheld terminals and would, thus, seem to be a reasonable choice. Furthermore, a single-ended bipolar input transistor has, due to its dominant exponential transfer characteristic, an input IP_3 of some 70 mV at room temperature (i.e., $IP_3 = V_{th}\sqrt{8} \approx 70$ mV$_p$), which corresponds to -13 dBm in a 50-Ω resistor [10]. This suggests that a -16 dBm intercept point is reasonable from an implementation point of view. Losses in the antenna filter, antenna switch, and matching network will relax this further while the choice of impedance level in the matching network also offers some design flexibility. After the first selectivity filter, the interfering signals will be attenuated sufficiently for intermodulation to be a nonissue for stages following this filter.

Evaluating (8) with the chosen parameter values, we get an SFDR of 50 dB, which is quite reasonable from a low-power implementation point of view. The last stage where interferers are not substantially attenuated will be the stage with the highest signal level. This is typically the first stage in the selectivity filter. For example, with a few volts of supply voltage, a compression point of 1 V is typical, corresponding to an IP_3 of at least some 3 V, and with this compression point we require a thermal noise floor below 500 μV to meet the demanded SFDR of 50 dB. Such a noise floor is achievable also for very complex low-power analog filters [11] [a ten-pole filter with a 1 pF capacitor per pole and a pole noise factor F of 10 dB would roughly generate 200 μV of noise, i.e., $v_n^2 \approx FkT/C = 10 \cdot 1.38 \cdot 10^{-23} \cdot 300/1 \cdot 10^{-12} = (200\ \mu\text{V})^2$].

In addition to signal-path noise sources, on-chip logic circuits will generate substrate and supply rail interference. With a low supply voltage and modest power supply and substrate rejection ratios, 500 μV seems to be a reasonable noise limit. Furthermore, increasing the compression point above 1 V is not compatible with low supply voltages and the IF filter will, thus, be the SFDR-limiting stage.

Equations (8) and (9) show that we cannot improve the range (lower RX_{SENS}) unless either we increase SFDR or sacrifice intermodulation performance (we assume C/I_{AWGN} to be fixed). Thus, increasing range will result in either higher power consumption or more intermodulation from nearby interferers. This will aggravate the near-far problem and for Bluetooth, it was decided to promote low power and robustness to interference rather than range. Again, should it prove to be desirable, the designer is free to exceed the minimum requirements.

Early capacity simulations indicated that in a 10-m cell with 10 simultaneous links, a receiver noise figure of more than 40 dB would be sufficient for getting very close to the maximum average throughput. That is, only for applications with little traffic does a very good noise figure make a difference. There are two reasons for this. First, the interference levels in a reasonably populated cell will be higher than the thermal noise levels. Second, when the range exceeds 8.5 m, the path-loss increases sharply and for every decibel in noise figure improvement, range is only increased by some 7% because of an increased number of objects obscuring the path. For best possible line-of-sight conditions, the range increases some 12% for every decibel of noise figure improvement.

C. Selectivity

To maximize capacity, high adjacent-channel protection (ACP) ratios are desirable. Unlike in the case with static frequency allocation, it is possible in Bluetooth to avoid interference by frequency hopping. Still, the probability of being jammed increases when ACP is lowered, as it is increasingly likely that a sufficiently strong interferer will occupy the closest adjacent channels. System simulations have indicated that protection ratios of 0 dB, 30 dB, and 40 dB for the closest, second, and third-and-beyond-adjacent channels, respectively, suffice. These ACP values result in very little additional throughput degradation as the co-channel interference will be the limiting factor for the throughput performance.

To enable low-cost implementation, external filters are to be avoided. It is not possible, however, to get sufficient selectivity at a high IF (i.e., an IF where the image frequency falls out-of-band, or some 100 MHz). This problem can be solved with sophisticated, and more costly, receiver architectures. It turns out, however, that an in-band image does not degrade system throughput substantially (less than 5%), providing the image rejection is better than 20 dBc. Such an image rejection is well within the capabilities of modern IC processes and it was decided to add a waiver for an in-band image in the specification to promote integrated low-IF receivers. With a low-IF receiver, the intermediate frequency may be chosen rather freely and, in particular, it may be close to be the IF filter bandwidth.

An active filter's power consumption is roughly proportional to its center frequency, and its complexity is proportional to its highest Q-values (i.e., to the filter order and the reciprocal of the relative bandwidth). By specifically enabling a low-IF, we promote low-cost, low-power implementations. An alternative to a low-IF heterodyne solution would be a zero-IF, or homodyne, receiver [4]. These receivers are sensitive to high levels of in-band interference, and it was considered too risky to rely on such a receiver to reach a low-cost goal.

With random frequency hopping, co-channel interference will occur with some probability. Because the co-channel protection ratio is much less than the adjacent channel protection ratios, the co-channel interference will be the dominating source of interference. In Bluetooth, a $C/I_{\text{co}-\text{channel}}$ of 11 dB is required and this has to be balanced against the suppression of the first adjacent channel; better co-channel performance degrades the adjacent channel performance and vice versa. The in-band image also appears as a co-channel interferer, but 20-dBc image suppression is sufficiently high to make the image insignificant with respect to $C/I_{\text{co}-\text{channel}}$.

Selectivity also sets phase-noise requirements on the local oscillator VCO. The most demanding phase-noise requirement will be the third adjacent channel protection ratio (40 dB). That is, the LO noise power at the channel of an interferer will mix down the interferer to the desired channel [4]. This reciprocal mixing will have to be lower than the ACP for the interferer channel. In decibel units, we have the following phase noise requirement:

$$L_{\Delta f} \leq -3 - C/I_{AWGN} - BW - ACP_{\Delta f} \quad \text{dBc/Hz} \tag{10}$$

where 3 dB is an extra margin to make the phase noise smaller than the selectivity filter contribution. Further, a rectangular phase-noise distribution within the channel is assumed. This translates to a phase-noise requirement $L_{\geq 3\text{ MHz}} \leq -124$ dBc/Hz. This phase noise is less demanding than what is found in typical cellular systems and will enable on-chip resonators to be used [12], [13], providing the resonator Q is on the order of 10, as well as lower VCO power.

To promote fast development of Bluetooth receivers, a few parameters in the Bluetooth specification are relaxed in the shorter term. Specifically, we have ACP_1 MHz $= -4$ dB and $C/I_{\text{co}-\text{channel}} = 14$ dB. After a convergence period of 18 months, these initial values will be tightened to their final values—$ACP_{1\text{ MHz}} = 0$ dB and $C/I_{\text{co}-\text{channel}} = 11$ dB. These long-term tighter values are still tentative pending the evaluation after the convergence period.

Finally, an in-band blocking level of 40 dB was selected. This blocking level is the same as $ACP_{3\text{ MHz}}$. Such a blocking level corresponds to a ratio of the maximum signal path loss to interferer path loss of 100 : 1 (i.e., a ratio of 40 dB). That is, interferers more than 30 cm away will not substantially increase the BER for links at the range limit according to (4) and (5).

VI. CONCLUSION

Bluetooth is a system for providing short-range, wireless connectivity between portable devices. The design of the air interface has been optimized for low power, interference immunity, and *ad hoc* connectivity. The link provides both synchronous services like voice and asynchronous services like file transfer. Relaxed requirements on the radio side allow the implementation of single-chip low-power and low-cost Bluetooth transceivers with very few external components. The system is supported by several leading manufacturers of personal computers and telecommunications equipment. The first consumer products to support Bluetooth are expected to appear on the market around mid-2000.

REFERENCES

[1] (1999, July). *Specification of the Bluetooth System v1.0A* [Online] Available: http:/www.bluetooth.com.

[2] FCC, Part 15–Radio Frequency Devices.

[3] IEEE 802.11, *Wireless LAN MAC and Physical Layer Specification*, June 1997.

[4] B. Razavi, *RF Microelectronics*. Englewood Cliffs, NJ: Prentice-Hall, 1998.

[5] E. M. Cherry and D. E. Hooper, *Amplifying Devices and Low-Pass Amplifier Design*. New York: Wiley, 1968.

[6] GEC Plessey, ““RF propagation considerations for WLAN applications,” GEC Plessey Semiconductors”, Tech. Rep. AN145, 1995.

[7] J. Davidse, *Analog Electronic Circuit Design*: Prentice Hall, 1991.

[8] P. Wambacq and W. Sansen, *Distortion Analysis of Analog Integrated Circuits*. Boston, MA: Kluwer, 1998.

[9] D. O. Pederson and K. Mayaram, *Analog Integrated Circuits for Communication; Principles, Simulation and Design*. Boston, MA: Kluwer, 1991.

[10] W. Sansen, “Distortion in elementary transistor circuits,” *IEEE Trans. Circuits Syst. II*, vol. 46, pp. 315–325, Mar. 1999.

[11] J. Silva-Martinez, M. Steyaert, and W. Sansen, *High-Performance CMOS Continuous-Time Filters*. Boston, MA: Kluwer, 1993.

[12] T. H. Lee, *The Design of CMOS Radio-Frequency Integrated Circuits*. Cambridge, MA: Cambridge Univ. Press, 1998.

[13] A. A. Abidi, “Low-power radio-frequency IC’s for portable communications,” *Proc. IEEE*, pp. 544–569, Apr. 1995.

Jaap C. Haartsen (Member, IEEE) received the M.Sc. and Ph.D. degrees in electrical engineering from the Delft University of Technology, The Netherlands.

He joined Ericsson Mobile Communications in 1991 and has since worked in RTP, USA, and in Lund, Sweden, in the area of wireless technology. Currently, he is in Emmen, The Netherlands, where he is working with the Bluetooth system both for current and future applications. Recently, he was appointed as part-time Professor at the Twente University of Technology, The Netherlands, in the area of mobile communications. He holds more than 30 patents.

Sven Mattisson (Member, IEEE) received the Ph.D. degree in applied micro electronics from Lund University, Lund, Sweden, in 1986.

From 1987 through 1994, he was an Associate Professor in Applied Micro Electronics, Lund, where his research was focused on circuit simulation and analog ASIC design. 1995, he joined Ericsson Mobile Communications, Lund, where he is an Expert in Analog System Design and Deputy Manager of research. Since 1996, he has also been an Adjunct Professor at Lund University. He has been involved in Bluetooth development since joining Ericsson.