Bluetooth is a standard for short range, low power, low cost wireless communication that uses radio technology. Although originally envisioned as a cable-replacement technology by Ericsson (a major cell phone manufacturer) in 1994, embedded Bluetooth capability is becoming widespread in numerous types of devices. They include intelligent devices (PDAs, cell phones, PCs), data peripherals (mice, keyboards, joysticks, cameras, digital pens, printers, LAN access points), audio peripherals (headsets, speakers, stereo receivers), and embedded applications (automobile power locks, grocery store updates, industrial systems, MIDI musical instruments).
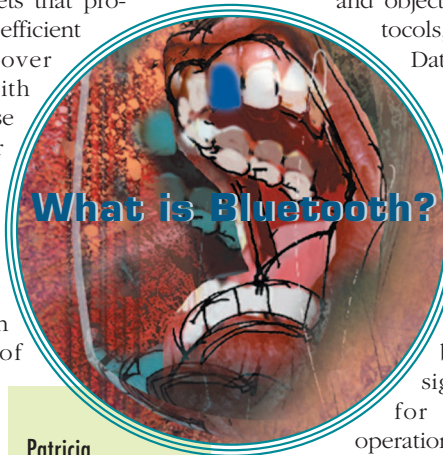
Ericsson joined forces with Intel Corporation, International Business Machines Corporation (IBM), Nokia Corporation, and Toshiba Corporation to form the Bluetooth Special Interest Group (SIG) in early 1998. 3Com Corporation, Lucent/Agere Technologies Inc., Microsoft Corporation and Motorola Inc. joined the group in late 1999. Joint work by the SIG members allowed the Bluetooth vision to evolve into open standards to ensure rapid acceptance and compatibility in the marketplace. The resulting Bluetooth specification, developed by the Bluetooth SIG, is open and freely available at the official Bluetooth website www.bluetooth.org. Bluetooth technology is already supported by over 2100 companies around the world. The Wireless Personal Area Network (WPAN) technology, based on the Bluetooth Specification, is now an IEEE standard under the denomination of 802.15 WPANs. In 2003, Cahners In-Stat estimated that Bluetooth-enabled equipment shipments would climb to just under 1 billion units by 2005.

The Bluetooth specification defines how Bluetooth devices will group themselves for the purposes of communication. A Bluetooth Wireless Personal Area Network (BT-WPAN) consists of *piconets*. Each piconet is a cluster of up to eight Bluetooth devices. One device is designated as the master, and the others are the slaves. Two piconets can be connected through a common Bluetooth device (a gateway or bridge) to form a *scatternet*. These interconnected piconets within the scatternet form a backbone for the Mobile Area Network (MANET), and can enable devices which are not directly communicating with each other, or which are

out of range of another device, to exchange data through several hops in the scatternet.

Current implementations of Bluetooth depend primarily on simple point-to-point data links between Bluetooth devices within direct range of each other. However, the Bluetooth specification defines not only a point-to-point link (connectivity) solution, but also a solution for more complex networking topologies. Therefore, the goal is to form Bluetooth scatternets that provide effective and efficient communication over multiple hops with acceptable response times and power consumption so that end-to-end solutions can be deployed.

This article presents an overview of Bluetooth communication. The following article (starting page 36) will focus on the formation of Bluetooth scatternets and some proposed scatternet models.

The Bluetooth protocol stack

The Bluetooth specification divides the Bluetooth protocol stack into three logical groups. They are the Transport Protocol group, the Middleware Protocol group and the Application group, as shown in Fig. 1.

The Transport group protocols allow Bluetooth devices to locate each other, and to manage physical and logical links with higher layer protocols and applications. Please note that the use of the word "transport" in the Transport protocol group does not indicate that it coincides with the Transport layer of the Open Systems Interconnection Reference Model (OSI) model. Rather, these protocols correspond to the Data-Link and Physical layers of the OSI model. The Radio, Baseband, Link Manager, Logical Link Control and Adaptation (L2CAP) layers and the Host Controller Interface (HCI) are included in the Transport Protocol group. These protocols support both asynchronous and synchronous transmission. All the protocols in this group are

required to support communications between Bluetooth devices.

The Middleware Protocol group includes third-party and industry-standard protocols, as well as Bluetooth SIG-developed protocols. These protocols allow existing and new applications to operate over Bluetooth links. Industry-standard protocols include Point-to-Point Protocol (PPP), Internet Protocol (IP), Transmission Control Protocol (TCP), wireless application protocols (WAP), and object exchange (OBEX) protocols, adopted from Infrared Data Association (IrDA).

Bluetooth SIG-developed protocols include 1) a serial port emulator (RFCOMM) that enables legacy applications to operate seamlessly over Bluetooth transport protocols, 2) a packet-based telephony control signaling protocol (TCS) for managing telephony operations, and 3) a service discovery protocol (SDP) that allows devices to obtain information about each other's available services. Reuse of existing protocols and seamless interfacing to existing applications was a high priority in the development of the Bluetooth specifications, as shown in Fig. 2.

The Application group consists of actual applications that use Bluetooth links. They can include legacy applications as well as Bluetooth-aware applications. A brief discussion of the layers in the Transport group follows.

Radio layer. The specification of the Radio layer is primarily concerned with the design of the Bluetooth transceivers, discussed in detail later.

Baseband layer. This layer defines how Bluetooth devices search for and connect to other devices. The master and slave roles that a device may assume are defined here, as are the fre-
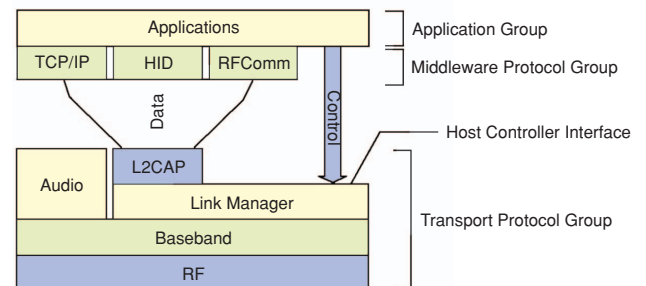

What is Bluetooth?

Patricia McDermott-Wells

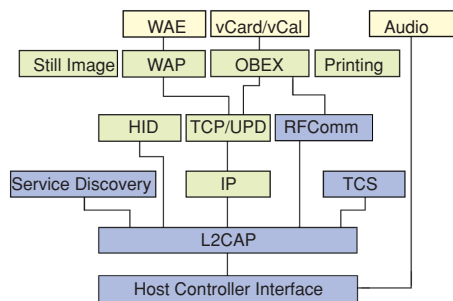Fig. 1 Bluetooth Core Protocol Groups

Fig. 2 Interoperability with existing protocols & applications

quency-hopping sequences used by devices. The devices use a time division duplexing (TDD), packet-based polling scheme to share the air-interface. The master and slave each communicate only in their pre-assigned time slots. Also, defined here are the types of packets, packet processing procedures and the strategies for error detection and correction, signal scrambling (whitening), encryption, packet transmission and retransmissions.

The Baseband layer supports two types of links: Synchronous Connection-Oriented (SCO) and Asynchronous Connection-Less (ACL). SCO links are characterized by a periodic, single-slot packet assignment, and are primarily used for voice transmissions that require fast, consistent data transfer. A device that has established a SCO link has, in essence, reserved certain time slots for its use. Its data packets are treated as priority packets, and will be serviced before any ACL packets. A device with an ACL link can send variable length packets of 1, 3 or 5 time-slot lengths. But it has no time slots reserved for it.

Link Manager layer. This layer implements the Link Manager Protocol (LMP), which manages the properties of the air-interface link between devices. LMP manages bandwidth allocation for general data, bandwidth reservation for audio traffic, authentication using challenge-response methods, trust relationships between devices, encryption of data and control of power usage. Power usage control includes the negotiation of low-

Fig. 3 Packet Format

power activity modes and the determination of transmission power levels.

L2CAP layer. The Logical Link Control and Adaptation Protocol (L2CAP) layer provides the interface between the higher-layer protocols and the lower-layer transport protocols. L2CAP supports mul-

tiplexing of several higher layer protocols, such as RFComm and SDP. This allows multiple protocols and applications to share the air-interface. L2CAP is also responsible for packet segmentation and reassembly, and for maintaining the negotiated service level between devices.

HCI layer. The Host Controller Interface (HCI) layer defines a standard interface for upper level applications to access lower layers of the stack. This layer is not a required part of the specification. Its purpose is to enable interoperability among devices and the use of existing higher-level protocols and applications.

Communication

A Bluetooth transceiver is a frequency-hopping spread-spectrum (FHSS) device that uses the unlicensed (worldwide) 2.4 GHz ISM (Industrial, Scientific, Medical) frequency band. In most countries, there are 79 channels available; however, some countries allow the use of only 23 channels. The nominal bandwidth for each channel is 1MHz.

FCC part 15.247 regulations restrict the maximum allowed peak power output to 1 watt and require that at least 75 of the 79 channels be used in a pseudo-random manner. A device cannot operate on a given channel for longer than 0.4 seconds within any 30-second period. These limits (or restrictions) were put into place to minimize the amount of interference in the ISM band, which is also used by 802.11 b/g devices, HomeRF devices, portable phones and microwave ovens.

When connected to other Bluetooth devices, a Bluetooth device hops (changes frequencies) at the rate of 1600 times per second for typical use, with a residence time of 625 μ sec . When in inquiry or page mode, it hops at 3200 hops per second with a residence time of 312.5 μ sec .

A Bluetooth transceiver uses all 79 channels, and hops pseudo-randomly across all channels at a rate of 1600 hops per second for standard transmissions. It has a range of approximately 10 meters, although ranges up to 100 meters can be achieved with amplifiers. Because the transceiver has an extremely small footprint, it is easily embedded into physical devices, making it a truly ubiquitous radio link.

The Bluetooth specification uses time division duplexing (TDD) and time division multiple access (TDMA) for device communication. A single time slot is 625 μ sec in length, representing the length of a

single-slot packet. At the Baseband layer, a packet consists of an access code, a header, and the payload, as shown in Fig. 3.

The access code contains the piconet address (to filter out messages from other piconets) and is usually 72 bits in length. The header contains link control data, encoded with a forward error-correcting code (FEC) with a 1/3 rate for high reliability. Such code is a repetition code and thus every bit in the header is transmitted three times. The header is usually 18 bits in length, and includes the active member address for a currently active slave. The payload can contain from 0 to 2745 bits of data, and may be protected by a 1/3 rate FEC (simple bit repetition, for SCO packets only), a 2/3 rate FEC (which is a (15,10) shortened Hamming code capable of correcting all one-bit errors and detecting all two-bit errors), or a 3/3/ rate (no FEC).

For SCO connections, packets must be exactly one time-slot in length. For ACL links, packets may be 1, 3, or 5 time slots in length.

Bluetooth uses polling-based packet transmission. All communication between devices takes place between a master and a slave, using time-division duplex (TDD), with no direct slave-to-slave communication. The master will poll each active slave to determine if it has data to transmit. The slave may only transmit data when it has been polled. Also, it must send its data in the time slot immediately following the one in which it was polled.

The master transmits only in even-numbered time slots, while the slaves transmit only in odd-numbered time slots. In each time slot, a different frequency channel f is used (a hop in the hopping sequence).

The piconet

The Bluetooth specification defines a piconet as an ad-hoc, spontaneous clustering of Bluetooth devices. In it, one device holds the role of master, while the rest of the devices are slaves. While there is no limit to the total number of slaves in a piconet, a maximum of seven slaves can be active in a piconet at any given point in time. If there are more than seven slaves, the rest of the slaves must be "parked." The maximum number of "parked" slaves is 255 per piconet with direct addressing via a *parked slave address* as defined by the SIG; however, indirect addressing of parked slaves by their specific *Bluetooth device address* is also permitted, effectively allowing any number of

parked slaves. To reactivate a parked slave, the master must first place a currently active slave into a parked state.

When two Bluetooth devices enter into communication range, they will attempt to communicate with each other. If no piconet is available at that time, a negotiation process will ensue. One device will become the master (usually the device which initiated the communication) and the other will become a slave.

Any Bluetooth device can function within a piconet as a master, a slave or a bridge. These roles are temporary and exist only as long as the piconet itself exists. The master device selects the frequency, the frequency-hopping sequence, the timing (when the hops will actually occur) and the polling order of the slaves. The master is also responsible for instructing the slave devices to switch to different device states for periods of inactivity.

A master and slave must exchange address and clock information in order for the slave to join the master's piconet. Bluetooth devices each have a unique Global ID used to create a hopping pattern. The master radio shares its Global ID and clock offset with each slave in its piconet, providing the offset into the hopping pattern. A slave must be able to recreate the frequency-hopping sequence of the piconet it has joined, must know which frequency to use at which time, and must synchronize itself with the master's clock. The slave device does not actually adjust its own clock. Rather it tracks the amount of clock drift between its clock and the master's, and adjusts its transmission schedule accordingly.

A Bluetooth bridge device (or gateway) interconnects two or more piconets for multi-hop communication. The bridge communicates with all the piconets connected to it by aligning itself with the clocking of each piconet when it is ready to communicate. However, it can only communicate with one piconet at a time. Because the bridge incurs additional overhead shifting from one clocking to another to communicate with each connected piconet, it has the potential to become a bottleneck.

A bridge device may be a slave in all of the piconets to which it is connected, or it may be a master in one piconet and a slave in the others. The interconnection of two or more piconets via bridge devices results in the formation of a Bluetooth *scatternet*.

A Bluetooth device can be in one of the following states: standby, inquiry, page, connected, transmit, hold, park or sniff, as shown in Fig. 4. A device is in *Standby* mode when it is powered on but has not yet joined a piconet. It enters the *Inquiry* state when it sends out requests to find other devices to which it might connect. A master of an existing piconet may also be in a *Page* state, sending out messages looking for devices that it can invite to join its piconet.

When successful communication is made between the master and the new device, the new device assumes the slave role, enters the *Connected* state, and receives an active address. While connected, the slave can transmit data when the master polls it. During the transmission of its data, the slave is in a *Transmit* state. At the end of its transmission, it returns to the *Connected* state.

The *Sniff* state is a low-power consumption state in which the slave "sleeps" for a pre-determined number of time slots. It wakes up at its appointed time slot for data transmission. It then returns to the inactive state until its next designated *Sniff* time slot arrives. The *Hold* state is another low-power state in which the slave is not active for a predetermined amount of time. However, there is no data transfer within the *Hold* state.

When a slave device has no data to send or receive, the master may instruct the slave to enter the *Park* state. When it enters a *Park* state, the slave relinquishes its active address in the piconet. The address will then be given to another slave that the master is reactivating from *Park* state.

Conclusions

Bluetooth wireless technology encompasses several key points that facilitate its widespread adoption: 1) it is an open specification that is publicly available and royalty free; 2) its short-range wireless capability allows peripheral devices to communicate over a single air-interface, replacing cables that use connectors with a multitude of shapes, sizes and numbers of pins; 3) Bluetooth supports both voice and data, making it an ideal technology to enable many types of devices to communicate; and 4) Bluetooth uses an unregulated frequency band available anywhere in the world.

To fully realize the Bluetooth vision, full networking of multiple Bluetooth devices is required. This leads to the investigation of Bluetooth scatternets, which must address scatternet formation and reconfiguration, scheduling, and routing issues.

Read more about it
• IEEE 802.15 Working Group for WPAN. [Online]. Available: http://www.ieee802.org/15/

• C. Bisdikian, S. Bouet, J. Inouye, R. Mettälä, B. Miller, K. Morley, et al. (1999, August 25). "Bluetooth Protocol Architecture - Version 1.0." Bluetooth White Paper - Bluetooth Special Interest Group. [Online]. Available: https://www.

Fig. 4 Device connection states

bluetooth.org/ foundry/sitecontent/document/Protocol_Architecture

• D. Chomienne and M. Eftimakis. (2001). *Bluetooth Tutorial*. [Online]. Available: http://www.newlogic.com/products/Bluetooth-Tutorial-2001.pdf

• J. Kardach, "Bluetooth architecture overview," *Intel Technology Journal*, Q2., '00

• R. Mettälä, J. Inouye, and T. Cooklev, "Bluetooth PC Card Transport Layer" Bluetooth White Paper - Bluetooth Special Interest Group. [Online] Available: https://www.bluetooth.org/foundry/sitecontent/document/PC_Card_Transport_Layer.pdf
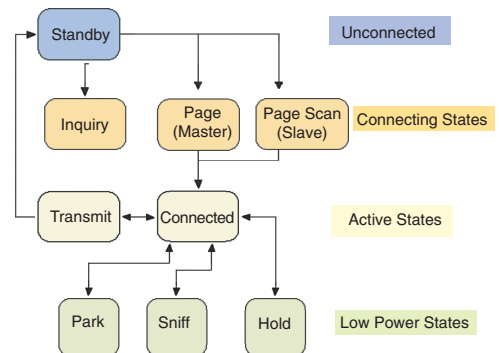
• B. A. Miller and C. Bisdikian, *Bluetooth Revealed*. Upper Saddle River, NJ: Prentice Hall PTR, 2001.

• J. Stripp, (2001), Bluetooth: Value Adds and Opportunities. MindBranch, Inc., N Adams, MA. [Online]. Available: http://www.palowireless.com/bluearticles/docs/bluetooth_value_adds.pdf

About the author
Patricia McDermott-Wells is an adjunct instructor in the Computer Science and Mathematics Depts. at Florida International University, Miami, FL, and a doctoral student in the School of Computer and Information Sciences at Nova Southeastern University, Davie, FL. She is president of Mega-Data Services, Inc., a small software development firm <www.mega.data.com>. She has a BS in Mathematics and a MS in Management Science/Operations Research from University of Miami. Contact her at <info@mega-data.com>.