Bluetooth in Wireless Communication

K. V. S. S. S. S. Sairam, University of Madras, Dr. M. G. R. Engineering College

N. Gunasekaran, Anna University

S. Rama Reddy, Jerusalem College of Engineering, Dr. M. G. R. Engineering College

ABSTRACT

Bluetooth is a method for data communication that uses short-range radio links to replace cables between computers and their connected units. Industry-wide Bluetooth promises very substantial benefits for wireless network operators, end workers, and content developers of exciting new applications. This article delves into the implementation and architecture of Bluetooth. It also describes the functional overview and applications of Bluetooth, and deals with the development of a model for recording, printing, monitoring, and controlling of eight process variables at the same time, using a distributed control system. We explain industrial automation via Bluetooth using IISS. Industrial automation is one of the major applications of Bluetooth technology. Industrial automation, in terms of controlling or monitoring a factory, office, or industrial process, means to install machines that can do the work instead of human workers. Industrial plants consists of many devices interconnected in different ways ranging from simple data collection units (I/O) to more intelligent devices such as sensors, one-loop controllers, or programmable controllers, and a supervisory system used as a human–machine interface (HMI) for data logging and supervisory control. An IISS is a controlling device that monitors the devices in a company. It basically communicates via the interface card in the PC; the hardware is connected parallel across the device, and it is interfaced with the PC via a transceiver. The device can be accessed both manually via the switches and remotely via the PC. A simulation of connecting a PC with the machines in a company was executed. Also, we wrote a software program using C language; we will show how the remote monitoring takes place between the control room and the PC. These details in the article establish the growing need for Bluetooth technology.

INTRODUCTION

Bluetooth is an open standard for wireless connectivity with supporters mostly from the PC and cell phone industries. Not surprisingly, its primary market is for data and voice transfer between communication devices and PCs. In this way, it is similar in purpose to the IrDA protocol. Bluetooth, however, is a radio frequency (RF) technology utilizing the unlicensed 2.5 GHz industrial, scientific, and medical (ISM) band. Target applications include PC and peripheral networking, hidden computing, and data synchronization such as for address books and calendars. Other applications could include home networking and home appliances of the future such as smart appliances, heating systems, and entertainment devices.

BLUETOOTH HISTORY

Bluetooth was invented in 1994 by L. M. Ericsson of Sweden. The standard is named after Harald Blaatand "Bluetooth" II, king of Denmark 940–981A.D. A runic stone has been erected in his capitol city Jelling (Jutland) that depicts the chivalry of Harald, and the "runes" say:

- Harald christianized the Danes.
- Harald controlled Denmark and Norway.
- Harald thinks notebooks and cellular phones should seamlessly communicate.

The Bluetooth Special Interest Group (SIG) was founded by Ericsson, IBM, Intel, Nokia, and Toshiba in February 1998 to develop an open specification for short-range wireless connectivity. The group is now also promoted by 3COM, Microsoft, Lucent, and Motorola. More than 1900 companies have joined the SIG.

The following section describes some of the requirements for the Bluetooth system and in essence suggests the functionalities planned for it.

WHY BLUETOOTH?

Bluetooth attempts to provide significant advantages over other data transfer technologies, such as IrDA and HomeRF, vying for similar markets. Despite comments from the Bluetooth SIG indicating that the technology is complementary to IrDA, it is clearly a competitor for PC-to-peripheral connection. IrDA is already popular in PC peripherals, but is severely limited by the short connection distance of 1 m and the line-of-sight

0163-6804/02/$17.00 © 2002 IEEE

requirement for communication. This limitation eliminates the feasibility of using IrDA for hidden computing, where the communicating devices are nearby but not visible to one another.

Due to its RF nature, Bluetooth is not subject to such limitations. In addition to wireless device connections up to 10 m (up to 100 m if the transmitter's power is increased), devices need not be within line of sight and may even connect through walls or other nonmetal objects. This allows for applications such as a cell phone in a pocket or a briefcase acting as a modem for a laptop or PDA.

Bluetooth is designed to be low cost, eventually under $10/unit. On the flip side, however, are the limited connection distance and, even more damaging, the transmission speeds. Bluetooth supports only 780 kb/s, which may be used for 721 kb/s unidirectional data transfer (57.6 kb/s return direction) or up to 432.6 kb/s symmetric data transfer. These rates are comparable to the 1–2 Mb/s supported by HomeRF and, although live digital video is still beyond the capability of any RF technology, perfectly adequate for file transfer and printing applications.

Finally, Bluetooth's main strength is its ability to simultaneously handle both data and voice transmissions. It is capable of supporting one asynchronous data channel and up to three synchronous voice channels, or one channel supporting both voice and data. This capability combined with ad hoc device connection and automatic service discovery make it a superior solution for mobile devices and Internet applications. This combination allows such innovative solutions as a mobile hands-free headset for voice calls, print to fax capability, and automatically synchronizing PDA, laptop, and cell phone address book applications (Fig. 1).

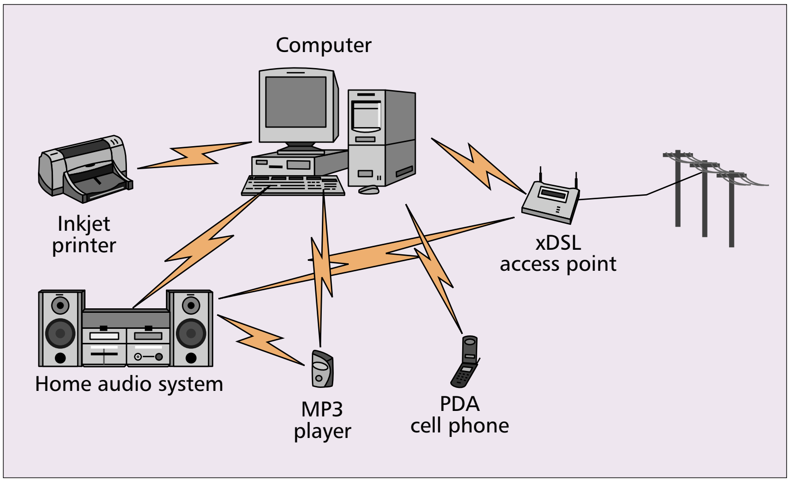

Figure 1. *A Bluetooth network.*

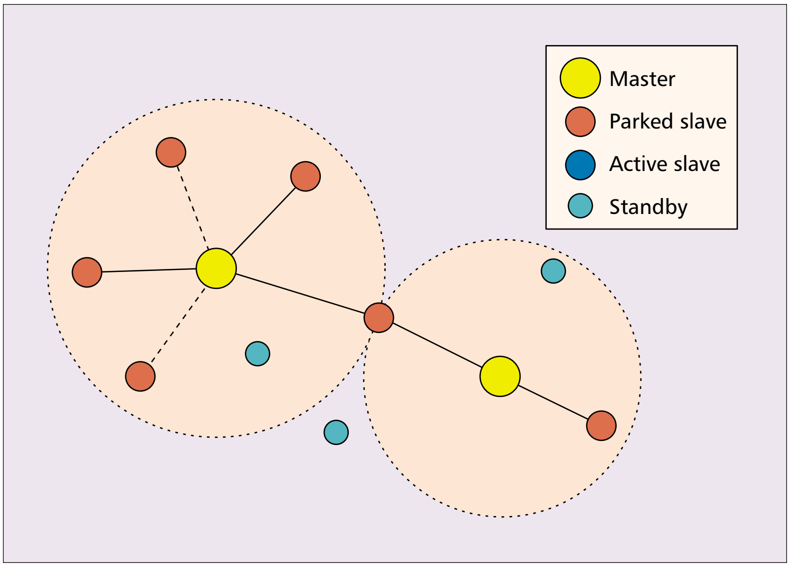

Figure 2. *Bluetooth scatternet diagram.*

ARCHITECTURE OVERVIEW

Bluetooth link control hardware, integrated as either one chip or a radio module and a baseband module, implements the RF, baseband, and link manager portions of the Bluetooth specification. This hardware handles radio transmission and reception as well as required digital signal processing for the baseband protocol. Its functions include establishing connections, support for asynchronous (data) and synchronous (voice) links, error correction, and authentication. The link manager firmware provided with the baseband CPU performs low-level device discovery, link setup, authentication, and link configuration. Link managers on separate devices communicate using the Link Management Protocol, which utilizes the services of the underlying link controller (baseband). The link control hardware may also provide a host controller interface (HCI) as a standard interface to the software.

NETWORK TOPOLOGY

Bluetooth devices are generally organized into groups of two to eight devices called ***piconets***, consisting of a single master device and one or more slave devices. A device may additionally belong to more than one piconet, either as a slave in both or as a master of one piconet and a slave in another. These bridge devices effectively connect piconets into a *scatternet*. A diagram of a Bluetooth scatternet is shown in Fig. 2.

Bluetooth operates in the unlicensed ISM frequency band, generally cluttered with signals from other devices: garage door openers, baby monitors, and microwave ovens, to name just a few. To help Bluetooth devices coexist and operate reliably alongside other ISM devices, each Bluetooth piconet is synchronized to a specific frequency hopping pattern. This pattern, moving through 1600 different frequencies per second, is unique to the particular piconet. Each frequency hop is a time slot during which data packets are transferred. A packet may actually span up to five time slots, in which case the frequency remains constant for the duration of that transfer.

BASEBAND STATE MACHINE

Piconets may be static or formed dynamically as devices move in and out of range of one another. A device leaves standby (the low-power default state) by initiating or receiving an inquiry or a page command. An inquiry may be used if

The Bluetooth baseband provides transmission channels for both data and voice, and is capable of supporting one asynchronous data link and up to three synchronous voice links (or one link supporting both). Synchronous connection-oriented links are typically used for voice transmission.

the address of a targeted device is unknown; it must be followed by a page command. A page command containing a specific DeviceAccessCode is used to connect to a remote device. Once the remote device responds, both devices enter the connected state, with the initiating device becoming the master and the responding device acting as a slave.

Once in the connected state, the slave device will synchronize to the master's clock and to the correct frequency-hopping pattern. At this point, link managers exchange commands in order to set up the link and acquire device information. The master will then initiate regular transmissions in order to keep the piconet synchronized. Slaves listen on every master-transmit time slot in order to synchronize with the master and to determine if they have been addressed.

Each active slave is assigned an active member address (AM_ADDR) and participates actively on the piconet, listening to all master time slots to determine if it is being addressed by the master. In addition, there are three lower-power slave states: *sniff*, *hold*, and *park*. A master can only transmit to devices in sniff mode during particular sniff-designated time slots. Therefore, these devices listen only during these special time slots and sleep the rest of the time. A slave in hold mode, alternately, does not receive any asynchronous packets and listens only to determine if it should become active again. Finally, a device in park mode not only stops listening, but also gives up its active member address. It is only a member of the piconet in that it remains synchronized with the frequency hopping pattern.

Baseband Links

The Bluetooth baseband provides transmission channels for both data and voice, and is capable of supporting one asynchronous data link and up to three synchronous voice links (or one link supporting both). Synchronous connection-oriented (SCO) links are typically used for voice transmission. These are point-to-point symmetric connections that reserve time slots in order to guarantee timely transmission. The slave device is always allowed to respond during the time slot immediately following an SCO transmission from the master. A master can support up to three SCO links to a single or multiple slaves, but a single slave can support only two SCO links to different masters. SCO packets are never retransmitted.

Asynchronous connectionless (ACL) links are typically used for data transmission. Transmissions on these links are established on a per-slot basis (in slots not reserved for SCO links). ACL links support point-to-multipoint transfers of either asynchronous or isochronous data. After an ACL transmission from the master, only the addressed slave device may respond during the next time slot, or if no device is addressed, the packet is considered a broadcast message. Most ACL links include packet retransmission.

Link Manager

The baseband state machine is controlled largely by the link manager. This firmware, generally provided with the link control hardware, handles link setup, security, and control. Its capabilities include authentication and security services, quality of service monitoring, and baseband state control. The link manager controls paging, changing slave modes, and handling required changes in master/slave roles. It also supervises the link and controls handling of multislot packets.

Link managers communicate with each other using the *Link Management Protocol* (LMP), which uses the underlying baseband services. LMP packets, which are sent in the ACL payload, are differentiated from logical link control and adaptation protocol (L2CAP) packets by a bit in the ACL header. They are always sent as single-slot packets and are higher priority than L2CAP packets. This helps ensure the integrity of the link under high traffic demand.

Host Controller Interface

Some link controller hardware may include an HCI layer above the link manager. This firmware layer is used to isolate the Bluetooth baseband and link manager from a transport protocol such as USB or RS-232. This allows a standard host processor interface to Bluetooth hardware. An HCI driver on the host is used to interface a Bluetooth application with the transport protocol. Currently three transport mechanisms are supported: USB, RS-232, and UART. The HCI layer is illustrated in Fig. 3. Using HCI, a Bluetooth application can access Bluetooth hardware without knowledge of the transport layer or other hardware implementation details.

Software Protocols

The remaining Bluetooth protocols are implemented in software. L2CAP, the lowest layer, provides the interface to the link controller and allows for interoperability between Bluetooth devices. It provides protocol multiplexing, which allows support for many third-party upper-level protocols such as TCP/IP and vCard/vCalendar. In addition, L2CAP provides group management mapping upper protocol groups to Bluetooth piconets, segmentation and reassembly of packets between layers, and negotiation and monitoring quality of service between devices.

Several Bluetooth protocols interface to the L2CAP link layer. SDP provides service discovery specific to the Bluetooth environment without inhibiting the use of other service discovery protocols. RFCOMM is a simple transport protocol providing serial data transfer. A Port Emulation Entity is used to map the communication API to RFCOMM services, effectively allowing legacy software to operate on a Bluetooth device. Telephony Control Protocol Specification (TCS) is provided for voice and data call control, providing group management capabilities and connectionless TCS, which allows for signaling unrelated to an ongoing call. Both point-to-point and point-to-multipoint signaling are supported using L2CAP channels, although actual voice or data is transferred directly to and from the baseband — bypassing L2CAP — over SCO links.

Bluetooth also supports IrDA Object Exchange Protocol (IrOBEX), a session protocol defined by IrDA. This protocol may run over other transport layers, including RFCOMM and

TCP/IP. For Bluetooth devices, only connection-oriented OBEX is supported. Three application profiles have been developed using OBEX. These include synchronization functionality for phone books, calendars, messaging, and so on; file transfer functionality; and object push for business card support.

Finally, Bluetooth may be used as a *Wireless Application Protocol* (WAP) bearer. The specification outlines the interoperability requirements for implementing this capability.

Logical Link Control and Adaptation Protocol

The L2CAP link layer operates over an ACL link provided by the baseband. A single ACL link, set up by the link manager using LMP, is always available between the master and any active slave. This provides a point-to-multipoint link supporting both asynchronous and isochronous data transfer. L2CAP provides services to upper-level protocols by transmitting data packets over L2CAP channels. Three types of L2CAP channels exist: *bidirectional signaling channels* that carry commands; *connection-oriented channels* for bidirectional point-to-point connections; and *unidirectional connectionless channels* that support point-to-multipoint connections, allowing a local L2CAP entity to be connected to a group of remote devices.

Channels

Figure 4 shows L2CAP entities with various types of channels between them. Every L2CAP channel includes two endpoints referred to by a logical channel identifier (CID). Each CID may represent a channel endpoint for a connection-oriented channel, a connectionless channel, or a signaling channel. Since a bi-directional signaling channel is required between any two L2CAP entities before communication can take place, every L2CAP entity will have one signaling channel endpoint with a reserved CID of 0x0001. All signal channels between the local L2CAP entity and any remote entities use this one endpoint.

Each connection-oriented channel in an L2CAP entity will have a local CID that is dynamically allocated. All connection-oriented CIDs must be connected to a single channel, and that channel must be configured before data transfer can take place. Note that the channel will at that point be bound to a specific upper-level protocol. In addition, a quality of service (QoS) agreement for the channel will be established between the two devices. QoS is negotiated for each channel during configuration and includes data flow parameters such as peak bandwidth, as well as the transmission type: best effort, guaranteed, or no traffic.

Connectionless channels are unidirectional and used to form groups. A single outgoing connectionless CID on a local device may be logically connected to multiple remote devices. The devices connected to this outgoing endpoint form a logical group. These outgoing CIDs are dynamically allocated. The incoming connectionless CID, however, is fixed at 0x0002. Although multiple outgoing CIDs may be created to form multiple logical groups, only one incoming connectionless CID is provided on each L2CAP entity. All incoming connectionless data arrives via this endpoint. These channels do not require connection or configuration. Therefore, any required configuration information, such as upper-level protocol, is passed as part of the data packet.

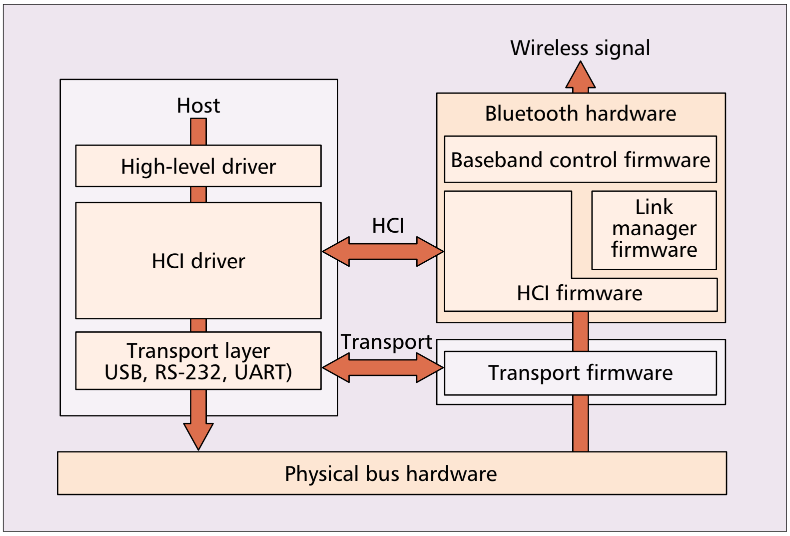

Figure 3. *The HCI layer.*

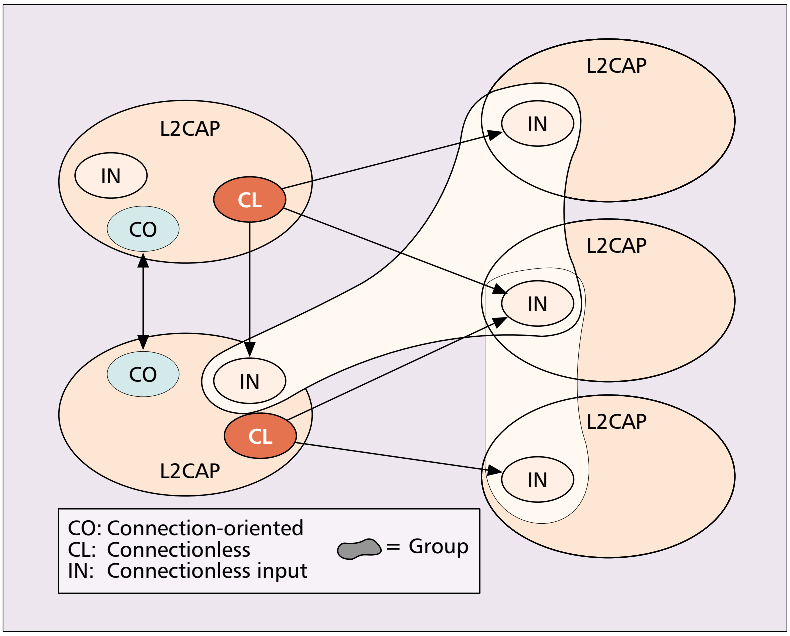

Figure 4. *A diagram of L2CAP channels.*

Channel State Machine

An L2CAP connection-oriented channel endpoint may be in one of several possible states, with data transfer only possible in the OPEN state. Initially, an endpoint is CLOSED, indicating that no channel is associated with the CID. This is the only state in which a baseband is not required, and it is the state an endpoint will default to if the link is disconnected.

The Service Discovery Protocol provides a means of determining what Bluetooth services are available on a particular device. A Bluetooth device may act as an SDP client querying services, an SDP server providing services, or both.

CONNECTION

In order to open a channel, the channel endpoint must be connected and configured. A connection occurs when either the local L2CAP entity requests connection to a remote device or an indication has been received indicating that a remote L2CAP entity is requesting connection to a local CID. In the first case, the request has originated from the upper-level protocol, has been passed on to the remote device, and the local entity enters the W4_L2CAP_Connect_RSP state to await a response. In the latter case, the indication is recognized as a connection request, the request has been passed on to the upper layer, and the local entity enters the W4_L2CA_Connect_RSP state to await a response. In either case, when the expected response is received, the local device enters the CONFIG state.

CONFIGURATION

A connection-oriented channel must be configured before data can be transmitted. Configuration involves a negotiation between both sides of the connection until all options are agreed upon. This is done using Configuration Request and Configuration Response commands. Supported configuration option types include a maximum transmission unit (MTU), a flush timeout, and a QoS agreement. The MTU option reflects the largest L2CAP packet payload the local device can handle. The flush timeout determines the amount of time the link controller will attempt to transmit an L2CAP segment before flushing the packet. Finally, the QoS agreement is used to negotiate a flow specification for a single transmission direction. L2CAP implementations are only required to support best dffort service, but no traffic or guaranteed service may also be negotiated. Other parameters in the flow specification include the token rate, token bucket size, peak bandwidth, latency, and delay variation. The requesting device indicates all of the nondefault options it will accept, to which the responding device either agrees or provides alternate settings. This process continues until all options are agreed upon. This configuration is for a single transfer direction, however, and the process must then be repeated for the opposite transfer direction. After all configuration parameters have been determined, both L2CAP entities enter the OPEN state, at which point data transfer may begin.

DISCONNECTION

To close a channel, one L2CAP entity must send a disconnection request to the other. If an entity receives a disconnect request from the upper-level protocol, it passes the request onto the remote device, and the local entity enters the W4_L2CAP_DISCONNECT_RSP state to await a response. If the local entity receives an indication that the remote device is requesting disconnection, it sends a disconnection request to the upper layer and then enters the W4_L2CA_DISCONNECT_RSP state to await a response. In either case, when the expected response is received, the local device enters the CLOSED state.

PACKETS

Data is transmitted across channels using packets. A connection-oriented channel uses packets with a 32-bit header followed by a payload of up to 65,535 bytes. The header includes a 16-bit length of payload to use for integrity check and the 16-bit destination CID. The payload contains information received from or being sent to the upper-level protocol. Connectionless channel packets also include a header but always use 0x0002 for the remote CID. In addition, the header is followed by a 16-bit (minimum) *protocol/service multiplexer* (PSM), which is used to indicate from which upper-level protocol the packet originated. This allows for packet reassembly on the remote device. The PSM field is not required for connection-oriented channels since they are bound to a specific protocol during connection.

SERVICE DISCOVERY PROTOCOL

The Service Discovery Protocol (SDP) provides a means to determine which Bluetooth services are available on a particular device. A Bluetooth device may act as an SDP client querying services, an SDP server providing services, or both. A single Bluetooth device will have no more than one SDP server, but may act as a client to more than one remote device. SDP provides access only to information about services; utilization of those services must be provided via another Bluetooth or third-party protocol. In addition, SDP provides no notification mechanism to indicate that an SDP server, or any specific service, has become available or unavailable as may occur when the services available on a device change, or when a device comes in or out of RF proximity. This would be a common occurrence in a network supporting mobile devices. The client may, of course, poll a server to detect unavailability, but other means are required to detect a server or service that has recently become available.

SERVICE RECORDS

In SDP, a service may provide information, perform an action, or control a resource. SDP servers maintain service records to catalog all available services provided by the device. Each service is represented by a single service record with a dynamically allocated service record handle that is unique within the server. A special service record, with a service record handle of 0x00000000, is provided to describe the SDP server itself and its supported protocols. Service attributes within a service record describe and define the supported service including the service type, a service ID, the protocols supported, the service name, a service description, and so on. These attributes are composed of a 16-bit ID and a variable length value. Attribute values in turn include a header field, with a data type and data size, and a data field. A range of data types are supported: null, unsigned integer, signed twos-complement integer, *Universally Unique Identifier* (UUID), text string, Boolean, data element sequence (set), data element alternative (select one), and URL. The interpretation of this data is dependent on the attribute ID and the service class to which the service belongs.

Connection-oriented packet format					
Length	(16 bits)	Dest CID	(16 bits)	Payload	(0–65,535 bytes)
Connectionless packet format					
Length	(16 bits)	Dest CID	0x0002	PSM	Payload (0–65,535 bytes)
Baseband packet format					
Access	Code header	Payload header	Payload	CRC	

DISCOVERING SERVICES

The purpose of SDP is to discover, not access, services. Two processes are supported: searching and browsing. Searching is based on UUIDs. A service record is returned by a search only if all of the UUIDs in the service search pattern are found within service record attribute values. It does not matter in which attribute a UUID is found, or whether the UUID is only one element in a list, as long as all the search pattern UUIDs are contained somewhere among the attribute values for the service.

PROTOCOL

SDP is a packet-based protocol utilizing a request-response architecture. The SDP packet is referred to as a protocol data unit (PDU), which includes a header followed by a variable number of parameters. The length of the parameter field is specified in the header, as is the type (PDU ID), which may indicate a request or response for searches or attribute queries. The header also includes a transaction ID that is used to match a request with a corresponding response. Every request must be acknowledged with a response. If for some reason the server cannot handle the request, it may send a response of type ErrorResponse (PDU ID 0x01).

It is possible that the response may be too large to fit into a single PDU. To accommodate this, a continuation state parameter is supported by most PDUs. In a response, this parameter indicates the number of bytes that are outstanding. The client may then resend the original request, with a new transaction ID, but with the continuation state parameter set. This alerts the server to send the continuation of the response. At what point a response is split is determined by the server.

Three categories of transactions (PDU IDs) are supported: *service search transactions*, *service attribute transactions*, and *service search attribute transactions*. Service search transactions are used to request a list of service record handles for service records that have attributes containing all of the UUIDs in a service search pattern. There is no mechanism to request all the service records, although browsing is supported as already described. Service attribute transactions are used to request specific attribute values from a service record. Service search attribute transactions combine the service search and service attribute transactions, which allows getting specific attribute values for all service records that match a service search pattern.

FRAME FORMATS

The Bluetooth core protocols consist of baseband, LMP, L2CAP, and SDP. The baseband and link control layer enables the physical RF link between Bluetooth units forming a piconet. As the Bluetooth RF system is a frequency hopping spread-spectrum system in which packets are transmitted in defined time slots on defined frequencies, this layer uses inquiry and paging procedures to synchronize the transmission hopping frequency and clock of different Bluetooth devices (see box this page).

The link manager protocol is responsible for link setup between Bluetooth devices. This includes security aspects like authentication and encryption by generating, exchanging, and checking link and encryption keys, and the control and negotiation of baseband packet sizes.

L2CAP provides connection-oriented and connectionless data services to the upper-layer protocols with protocol multiplexing capability, segmentation and reassembly, and group abstractions. Discovery services are crucial to the Bluetooth framework. These services provide the basis for all the usage models.

BLUETOOTH TODAY AND TOMORROW

With the bulk of the work developing the Bluetooth specification complete, the Bluetooth SIG is now working on improvements and analyzing feedback from the industry. In addition to their work investigating improvements in speed, security, noise immunity, and so on, the SIG continues to develop Bluetooth profiles.

As more and more manufacturers adopt Bluetooth and create devices that support it, developers will find new, previously unimagined ways of applying its power.

Together with other industry initiatives, such as WAP and Symbian, Bluetooth will have tremendous effects on everyday life. Bluetooth is one of the key technologies that can make the mobile information society possible, blurring the boundaries between home, office, and outside world.

CONCLUSION

In the future, Bluetooth is likely to be standard in tens of millions of mobile phones, PCs, laptops, and a whole range of other electronic devices. As a result, the market is going to demand new innovative applications, value-added services, end-to-end solutions, and much more. The possibilities opened up really are limitless, and because the radio frequency used is

With the bulk of the work developing the Bluetooth specification complete, the Bluetooth SIG is now working on improvements and analyzing feedback from the industry. In addition to their work investigating improvements in speed, security, noise immunity, and so on, the SIG continues to develop Bluetooth profiles.

The possibilities opened up really are limitless, and because the radio frequency used is globally available, Bluetooth can offer fast and secure access to wireless connectivity all over the world.

globally available, Bluetooth can offer fast and secure access to wireless connectivity all over the world. With potential like that, it's no wonder that Bluetooth is set to become the fastest adopted technology in history.

BIBLIOGRAPHY

[1] J. C. Haartsen *et al.*, "The Bluetooth Radio System" *IEEE Pers. Commun.*, Feb. 2000, pp. 28–36.

[2] M. Albrecht *et al.*, "IP Services Over Bluetooth: Leading the Way to a New Mobility," *IEEE Local Comp. Net.*, 1999, pp. 2–11.

[3] O. Miklos *et al.*, "Performance Aspects of Bluetooth Scatternet Formation" *1st Mobihoc*, Aug. 2000 pp 147–48.

[4] P. Bhagwat, L. Tassiulas, and R. LaMaire, "Distributed Topology Construction of Bluetooth Personal Area Networks," *IEEE INFOCOM 2001*, Tel Aviv, Israel, Mar. 2000.

BIOGRAPHIES

K. V. S. S. S. S. Sairam (s5kanduri@rediffmail.com) is working as a senior lecturer in the ECE Department, Dr. M. G. R. Engineering College, University of Madras, India. He is pursuing his Ph.D degree in fiber optics communication under the guidance of Dr. N. Gunasekaran, professor, SECE, Anna University. His research interests include fiber optics networking, switching, routing, and wireless communication. He got his Master's degree in industrial electronics from Sri Jasyachama Rajendra College of Engineering, University of Mysore, India, in May 1998. He got his Bachelor's degree from Karnatak University, Dharwad, India, in February 1996. He has guided M.Tech projects, and has published 2 Journal papers and 10 papers for international and national conferences.

N. GUNASEKARAN is working as a professor, School of Electronics and Comunication Engineering, Anna University, India. He has published several papers in journals and contributed to international and national conferences. He has guided several Ph.D and Master degree projects.

S. RAMA REDDY is currently working as a professor in Jerusalem College of Engineering, India. He has published several papers in journals and contributed to international and national conferences. He has guided several Ph.D and Master degree projects.