Bluetooth Technology Overview: An Application Programmer's Primer

Kenneth Steck, AnywhereYouGo.com

Bluetooth is a technology specification that utilizes the 2.4 GHz Industrial, Scientific, and Medical (ISM) frequency band in order to achieve wireless connectivity among devices. The technology is aimed at small mobile devices because the components are small, require very little power, and are inexpensive. The Bluetooth specification is a result of the collaborated efforts of over 1800 companies involved in the Bluetooth Special Interest Group (SIG).

With a Bluetooth wireless connection, Bluetooth enabled devices will be able to use each other's services and create ad hoc networks. Possible scenarios include: a wireless headset for a mobile phone; a PDA and PC synchronizing data automatically and wirelessly; a PDA accessing the Web via dial-up networking services provided by a Bluetooth enabled mobile phone.

To venture beyond the standard usage models found in the Bluetooth specifications, imagine a scenario where you are out traveling and in an airport. Your PDA has your spouse's birthday listed, along with a list of possible gift ideas. As you walk past the shopping section of the terminal, the information in your PDA is logged by a Bluetooth access point in the stores. Any store that offers items on your spouse's wish list may be able to wirelessly send you a coupon or notification of a sale on some items. While on the plane you browse through your PDA, notice these messages, and complete an e-mail order form while responding to other business-related e-mail. When you land and depart from the plane, you turn on your Bluetooth enabled mobile phone. Your PDA notices you have several e-mails waiting in your outbox, so it connects to your mobile phone for dial-up Internet access and sends them. Now your business e-mail and a birthday gift for your spouse is all handled automatically as you walk to the auto rental counter. The possible scenarios are virtually limitless for commercial and consumer use.

Bluetooth Network Topology

Bluetooth networks at the first level are called **piconets**. A piconet is defined as up to 255 devices (of which only eight may be active) linked together, all using a specific frequency-hop sequence defined by the **master device**. In a piconet, there can be only one master device and seven active slave devices. The device that initiates a connection is considered the master of the piconet; it is possible for a device to be a slave in two different piconets, or to be a master of one piconet and slave to another. Piconet communication is in the form of point-to-point or point-to-multipoint connections.

One or more piconets may be linked together to form a **scatternet**. A scatternet is defined as multiple piconets joined together, where each piconet has a unique frequency-hop sequence. There may be a maximum of ten fully loaded piconets in a scatternet. This topology allows inter-piconet communication, as well as range adaptation. The nominal range limit for two devices to communicate is 10 meters, but in the context of a scatternet this range may be extended to 100 meters by using inter-piconet communication.

Technical Features

Bluetooth operates in the 2.4 GHz ISM band and utilizes 74 MHz of the spectrum. This range varies across different countries, based on government regulations. Bluetooth uses a combination of several standards to ensure a robust connection. A **frequency-hopping** scheme is used to avoid radio interference and add a level of security to RF connections by using a different frequency within the available ISM

band for each data packet transmission. Each piconet is defined by its unique frequency-hopping pattern that is defined by an algorithm based on the master device address.

Time Division Duplex (TDD) is actually a fancy way of describing the fact that the radio unit does not transmit and receive at the same time. TDD means that the radio must set itself to transmit for one time slot, and then switch itself to receive for the next time slot.

Speech Coding (CVSD) is the particular voice data transmission algorithm used by Bluetooth — it's simply a way of digitally encoding the characteristics of analog data. Bluetooth also uses small data packets, which are protected with Forward Error Correction (FEC) or Automatic Repeat Request (ARQ), and Cyclic Redundancy Check (CRC) for error correction. More information on these algorithms is available at AnywhereYouGo.com or in the Bluetooth Core Specifications.

Bluetooth devices may communicate at a range of 10 meters. This range may be extended to 100 meters when using a scatternet topology to transmit messages, or by increasing the radio unit's transmission power.

Bluetooth devices may communicate at a speed of up to 1 Mbps, and may maintain simultaneous voice and data connections. Data connections may be synchronous or asynchronous, with data rates of 432 Kbps and 721 Kbps respectively. Up to three simultaneous voice connections may be maintained at a data rate of 64 Kbps. Voice connections are always synchronous.

Bluetooth Security

Bluetooth provides three mutually exclusive modes of built-in security: unsecured, service secure, and link secure. Unsecured connections are *completely* unsecured, and are used to provide services to any Bluetooth device. Service secure connections provide authentication for services that require knowledge of 'who' is at the other end of the link. Link secure connections provide authentication, authorization, and encryption (ciphering of plain text).

Authentication in Bluetooth is used during the establishment of a connection, while encryption is used during the lifetime of a connection. Encryption is configurable to meet local government regulations.

> *A full description and discussion of Bluetooth security is beyond the scope of this paper. For further information on this topic, visit www.AnywhereYouGo.com or www.bluetooth.com*

Bluetooth Qualification Program

The Bluetooth Qualification Program is a process by which a product is extensively tested to prove Bluetooth Specification compliance. Before any device may be marketed with the Bluetooth logo, it must pass the qualification program. This program exists to ensure that all Bluetooth devices will be able to interact with each other, despite being developed by different corporations. Again, a full discussion of this program is outside the scope of this paper.

Bluetooth Components

Any Bluetooth solution consists of four major components: antenna/RF component, Bluetooth hardware and firmware (baseband and Link Controller), Bluetooth software protocol stack, and the application itself. Each of these components is a product in itself, and companies exist that have entire business models based around solving only one of these four areas.

Antenna/RF

The antenna and RF design portion is interesting in that it requires a unique solution for each device. When purchasing a Bluetooth module for Ericsson, for instance, the antenna is not provided. Bluetooth silicon manufacturers cannot effectively provide an antenna with the hardware. Even single chip solutions require specialized antenna design, depending on the device. Antenna design requires specialized skills to ensure that the Bluetooth radio will operate within its specification.

Bluetooth Radio and Baseband

The Bluetooth radio is the hardware transceiver unit that implements the Bluetooth radio specification. The purpose of the specification is to provide compatibility between Bluetooth devices that operate in the 2.4 GHz ISM band, and to define the quality of the system. Further information on the Bluetooth radio specifications may be found in the Bluetooth core specification document.

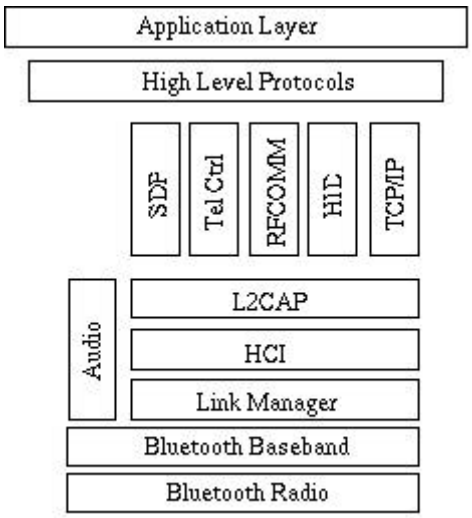
The Bluetooth baseband consists mainly of a Link Controller (LC) that carries out baseband protocols and low-layer link routines. Protocols defined within the scope of the baseband specification include (among others) physical channels and links, data packet definitions, error correction and detection, logical channels, channel control, and hop selection. For more information about the Bluetooth baseband specification, see the Bluetooth core specification document.

An example implementation of the Bluetooth radio and baseband is the Ericsson Bluetooth Module. In addition to the hardware, this module contains the firmware that implements the baseband specifications. As you'd expect, there are a number of other manufacturers developing Bluetooth modules too.

Bluetooth Software Protocol Stack

The Bluetooth software protocol stack can be thought of as driver code. This code allows the application software to send and receive information from the Bluetooth module. Several implementations of this currently exist, and vary from GNU licensed code to commercial products targeted at various operating systems.

Major components of the protocol stack are the Link Manager (LM), the Logical Link Control and Adaptation Protocol (L2CAP), the Host Control Interface (HCI), the Service Discovery Protocol (SDP), Audio/Telephony Control, RFCOMM, Human Interface Device (HID), TCP/IP, and other high level protocols. These are all described in the subsequent sections.



Link Manager (LM)

The LM manages link setup, link configuration, and link packet control and transfer. The LM also manages link security during the initialization of the connection and throughout its existence (where applicable). The LM handles synchronous and asynchronous packet communication within the piconet, as well as the timing parameters used during communication. The LM also handles master/slave role switching between devices. Sniff, hold, and park mode behavior is controlled by the LM too.

> *'Sniff', 'hold', and 'park' are power saving modes in which a Bluetooth device may operate. They allow for varying levels of participation and communication within a piconet. The use of each of these modes is broadly determined by the type of device, its function, and overall need for immediate service.*

Logical Link Control and Adaptation Protocol (L2CAP)

The services provided by L2CAP include protocol multiplexing, segmentation and reassembly, and quality of service. The L2CAP protocol architecture is connection-oriented, with connections labeled by a channel identifier. Each channel is assumed to be a full-duplex connection, with a Quality-of-Service (QoS) flow specification being applied to each channel direction.

Protocol multiplexing enables an application to use several higher-layer protocols simultaneously — RFCOMM, TCP/IP, etc. This service also passes packets used by the higher layer protocols to the appropriate handlers. Protocol identifiers have a one-to-many mapping with channel identifiers. For example, a master device may provide a TCP/IP service and have more than one slave unit using that service.

Segmentation and reassembly is a service by which packets from higher layer protocols are segmented into appropriate-sized Bluetooth packets and reassembled again after transmission. This service is transparent to the higher layer protocols.

The L2CAP also negotiates and enforces Quality Of Service (QoS) contracts, which are based on the type of service provided, with a 'best effort' contract used by default. QoS regulates the token size (bytes), the rate and peak bandwidth (bytes/sec), and other key elements to ensure a robust connection.

Host Control Interface (HCI)

The HCI provides a standard interface to the Bluetooth module and link manager services that is independent from the host hardware implementation. This layer provides transparency between the host controller and the Bluetooth hardware. There is an addendum to the HCI specification for each physical bus (USB, PCI, RS232, etc.) that further defines the interface functions based on which physical bus is used.

Service Discovery Protocol (SDP)

The SDP is the layer that exposes high-level services such as LAN access or printer services to users and other applications. This layer also provides information to implement a plug-and-play solution, such as a laptop computer using a printer. This layer could be implemented using a higher SDP such as Java JINI.

Audio and Telephony Control

These two protocols are linked, because in the Bluetooth specification, telephony Control contains Call Control and Audio Control. This protocol defines the interface needed to connect and disconnect a call, including signaling the devices desired to participate in the connection. Telephony audio links are established with

synchronous links, and therefore do not go through the same L2CAP-to-LM path that asynchronous links go through. Audio links may be thought of as direct baseband to baseband links.

RFCOMM

RFCOMM provides a protocol to emulate cables with Bluetooth, enabling compatibility with a large base of applications that currently use the serial port as their main communication bus. RFCOMM conveys all of the RS232 control signals, and supports remote port configuration. RFCOMM borrows from the IrComm in the IrDA protocol stack.

Human Interface Device (HID)

HID is a protocol that enables the concept of a cordless computer. HID describes keyboards, mice and joysticks. This layer would enable plug and play support for such devices when used with a PC.

TCP/IP

TCP/IP over Bluetooth presents a powerful way to link devices. TCP/IP is a network and transport layer that's widely supported by applications and APIs across almost every operating system. The problems with using TCP/IP over Bluetooth include, among others, handling ad hoc networking, DNS name resolution, and broadcasting. Better profiles for networking with Bluetooth are currently being developed by the SIG.

Other Protocols

Other protocols include things such as WAP, object exchange, still image, IR, etc. These protocols would be used by an application that sends its native packets through Bluetooth, just as it would use any other transport layer. All of these possibilities exist within the scope of Bluetooth. All that is missing is the development of the applications.

Application Software and Bluetooth Profiles

The application software is the highest-level component of a Bluetooth solution: it includes the user interface, as well as the implementation of any profile requirements. The application software discovers, broadcasts, and provides services to any device within range. Application software in the Bluetooth world is referred to as a **usage model**; these are made up of one or more **Bluetooth profiles**.

Bluetooth profiles specify a set of basic standards to ensure device interoperability. For the user, this ensures a common experience across Bluetooth devices. For the application programmer, a profile specifies the use of combined procedures from many basic standards, and reduces set parameters in basic standards.

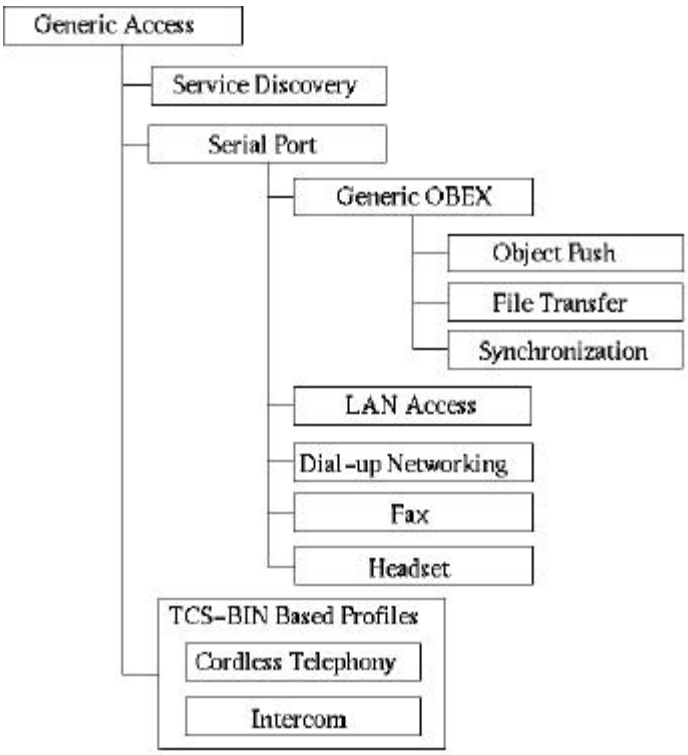
Profiles ensure interoperability on three major levels: radio, protocol, and usage:

- Radio interoperability ensures devices can contact each other in the first place, and keep in contact.
- Protocol interoperability ensures devices can communicate with and understand each other.
- Usage interoperability ensures the device can actually run applications and the user can interact with the device.

Each profile requires the implementation of basic standards that involve the Bluetooth stack (Baseband, LM, L2CAP), higher layer Bluetooth protocols (SDP, RFCOMM, etc.), higher-layer adapted protocols (TCP/IP, WAP, etc.), and considerations of device usage and human interaction. From the application programmer's perspective, what need to be implemented in the application

software are the profile requirements. The software stack (baseband firmware) is already implemented in the Bluetooth hardware. Application programmers should be able to limit their concerns to the API provided by whichever Bluetooth module manufacturer they've selected.

Listed below are high-level descriptions of each of the profiles found in the 1.0b specifications. For further information on these profiles, it is advisable to read the profile specifications thst are freely downloadable from `www.bluetooth.com`.



Generic Access Profile (GAP)

This profile covers user interface, modes, security, idle mode, and link establishment topics. This profile is the root of the profile tree, and all other profiles are derived from and expand on its basic requirements. The GAP is the only profile that all devices *must* implement. This profile defines the most basic set of intercommunication requirements.

The purpose of the GAP is to provide definitions and common requirements for modes and access procedures used by transport and application profiles. The GAP also describes how devices behave in standby and connecting states. This guarantees that links and channels always can be established between Bluetooth devices. The GAP is especially concerned with defining requirements for discovery, link establishment, and security procedures. User interface aspects are also addressed (mostly in order to guarantee a consistent user experience).

Service Discovery Profile (SDP)

The general expectation of the Bluetooth industry is that the number and types of services that can be offered over Bluetooth links will increase rapidly. In preparation for this situation, procedures need to be established to aid a user of a Bluetooth device in sorting out and selecting from these many services. This is the goal of SDP: to provide a standardized procedure to locate and identify services.

The SDP is used to locate services that are available on or via devices in the vicinity of a Bluetooth enabled device. Once the services are discovered, a user may then select to use one or more of them. The selection, access, and use of a service is

outside the scope of SDP. Rather, the SDP provides services used to discover services. The SDP defines the protocols and procedures that are used by a service discovery application on a device to locate services in other Bluetooth-enabled devices.

The SDP does not provide a definition for *automatic* service discovery. This means that the SDP defines only user initiated service discovery applications and procedures. There are other profiles (discussed later) that *do* define automatic service discovery requirements, which are initiated by application-level actions.

Serial Port Profile

This profile defines protocols and procedures to be used by devices using Bluetooth technology to implement an RS232 or similar serial cable emulation. This profile is dependent on procedures implemented by the GAP. It supports the scenario of setting up two virtual serial ports on two devices, and connecting with Bluetooth in order to emulate a serial cable connection.

The primary purpose of creating this profile is to support legacy applications requiring serial cable connections (with RS232 signaling). The serial port profile requires support for one-slot packets only, which is meant to ensure that data rates up to 128 kbps may be achieved. Support for higher rates is optional.

Headset Profile

The headset profile defines protocols and procedures to be used by devices such as the "Ultimate Headset", and is dependent on procedures implemented by the GAP and the serial port profile. The requirements defined by this profile address end-user services and define procedures required to guarantee interoperability between devices in the Headset use case. The headset profile does not require any fixed master/slave role.

The headset profile does not require any level of security or encryption when establishing and maintaining a voice data connection. The use of security is left up to the discretion of the user and to the capabilities of the device in use.

FAX Profile

The FAX profile defines protocols and procedures to be used by devices implementing FAX services, as described in the Data Access Point use model. A Bluetooth mobile phone or modem may be used by a computer or PDA to send and receive fax messages. The FAX profile is dependent on procedures implemented by the GAP and the serial port profile. The FAX profile does not require any fixed master/slave role, although the device using the FAX services (data terminal) will always initiate the connection. The device providing the FAX services (gateway) may need to request a master/slave role switch if it is providing services to more than one data terminal. The FAX profile *does* require a secure connection and encryption of all user data.

Dial-up Networking Profile

The dial-up networking profile defines protocols and procedures to be used by devices implementing Internet access services (Internet Bridge). A Bluetooth mobile phone or modem may be used to provide dial-up Internet access. The dial-up networking profile is dependent on procedures implemented by the GAP and the serial port profile. The dial-up networking profile does not require any fixed master/slave role, although the device using the dial-up networking services (data terminal) will always initiate the connection.

As in the FAX profile, the device providing the dial-up networking services (gateway) may need to request a master/slave role switch if it is providing services

to more than one data terminal. The dial-up networking profile does require a secure connection and encryption of all user data, and also requires an exchange of PINs at connection initiation.

LAN Access Profile

The LAN access profile defines requirements for devices providing PPP connections over RFCOMM. The LAN access profile does not require any particular networking protocol, however IP is the most widely used and is therefore discussed more throughout the profile specifications. LAN access to one or more devices is defined, yet LAN emulation, ad hoc network creation, and conferencing are not addressed. Those topics are currently being addressed by Working Groups in the SIG, and will be solved with additional profiles.

Generic Object Exchange Profile (GOEP)

The GOEP defines the protocols and procedures used by applications providing object exchange capabilities. The possible usage models may include synchronization or file transfer. Devices most likely to implement GOEP are PDAs, laptop PCs, and mobile phones. GOEP does not define a fixed master/slave role because the profile assumes only two devices are involved with object exchange. GOEP requires that bonding and pairing procedures be supported (see Baseband specification for more information); however the application is not required to use the procedures.

Object Push Profile

The object push profile defines requirements for applications that exchange data objects. These objects most likely will be in the form of business cards or calendar appointments. Bonding procedures must be supported, but use by the application is optional. This is because the devices may only exchange data once, such as the first time you make a business contact, and therefore have no need to maintain a lasting knowledge of each other. On the client side, user interaction is always required to initiate object push. This type of application is not defined to be "always on".

File Transfer Profile

The file transfer profile defines requirements for applications that need to be able to browse, transfer, and/or manipulate remote files and folders. A great analogy is to think about browsing the contents of other computers through the "Network Neighborhood" icon on your Windows desktop.

Synchronization Profile

The synchronization profile defines requirements for applications that need to keep data between devices synchronized. A great parallel is the Palm HotSync concept. This profile does require bonding and pairing to be supported and used, because users will want their data transfers to be secure, and they will generally only use one PDA and PC to perform synchronization.

Cordless Telephony Profile

The cordless telephony profile defines requirements for devices that use landline access through a base station. An example of this profile in action is the specific mode of the 3-in-1-phone usage model that enables the user to make phone calls over a landline when in range of a base station. This profile requires authentication and encryption of user data. The cordless telephony profile defines that the base station or **gateway** is the master device, while the handsets or **terminals** are the slaves.

Intercom Profile

The intercom profile defines requirements for devices that provide direct audio links between devices. An example of this profile in action is the specific mode of the 3-in-1-phone usage model that enables the user to use the handset as a walkie-talkie. This is a fairly simple profile, as it only defines the creation of a single audio link between two devices.