Crossroads
The ACM Student Magazine

Bluetooth: A Technical Overview

by ***Myra Dideles***

Introduction

Bluetooth is a low-power, short-range wireless technology originally developed for replacing cables when connecting devices like mobile phones, headsets and computers. It has since evolved into a wireless standard for connecting electronic devices to form Personal Area Networks (PANs) as well as ad hoc networks. Not only will cables be unnecessary for connecting devices, but connections will also be done seamlessly without the need for installations and software drivers. With this technology, devices will be able to discover any other Bluetooth-enabled device, determine its capabilities and applications, and establish connections for data exchange.

During Bluetooth's inception, its developers envisioned several usage scenarios [**12**, **18**]. The following examples illustrate five of these:

- *Three-in-one phone - use the same phone everywhere*

  This is a wireless phone that will use the best telecommunication technology available [**18**]. At the office, it will use Bluetooth technology to communicate with other phones thereby acting as an intercom or a walkie-talkie. At home, it will function as a cordless phone, incurring fixed-line charges. When the user is on the move, it can function as a mobile (cellular-like) phone.

- *Internet bridge - surf the Internet regardless of the connection*

The user will be able to connect to the Internet anywhere, regardless of whether it is through a wireless connection using a Bluetooth link with a mobile phone or a wired connection such as a local area network (LAN), a public switched telephone network (PSTN) or a digital subscriber line (DSL).

- *Interactive conference - connect every participant for instant data exchange*

  In conferences or meetings, participants will be able to instantly exchange information such as business cards or presentation slides using their Bluetooth-enabled devices.

- *The ultimate headset - a cordless headset keeps your hands free*

  This is a Bluetooth-enabled headset that allows users to connect wirelessly to their mobile phones or mobile PCs for a hands-free connection, giving users the flexibility to concentrate on more important matters.

- *Automatic synchronization*

  Personal devices such as a desktop computer, hand-held, mobile phone and notebook belonging to the same user will perform automatic synchronization of their Personal Information Management (PIM) applications. When the user enters the office, the calendar application on the user's mobile phone or handheld automatically synchronizes with the scheduler in the office, alerting the user of any conflicts in his schedule or upcoming meetings.

Today, scenarios such as those depicted in **Figure 1**, illustrate a vision for ad hoc networks connected by Bluetooth links. In the figure, devices belonging to one user can interconnect with each other and they can also connect to local information points - in this example, to get updates on flight arrivals and departures.

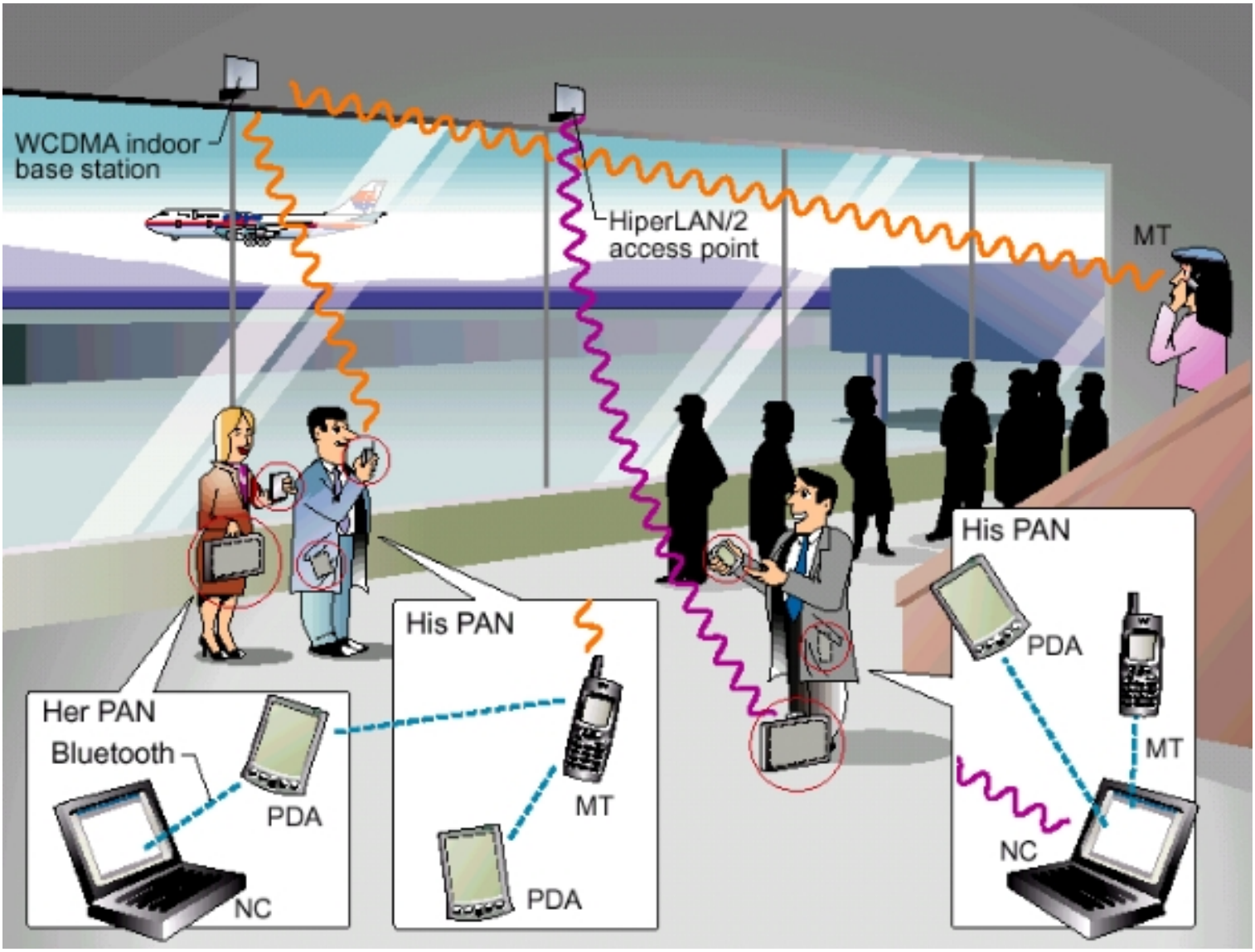

Figure 1: Ad hoc networks at an airport scenario [10].

This article will look more closely at how Bluetooth works, what protocols and profiles it uses as well as some of its security aspects. A simple connection process will be shown to demonstrate how the protocols work and interact with one another. Before concluding with a discussion, some issues and concerns regarding this technology will be presented including a comparison between IEEE 802.11b and Bluetooth.

How Bluetooth works

Bluetooth operates on the unlicensed Industrial Scientific Medical (ISM) band at 2.4 GHz, which ensures worldwide communication compatibility. Since the ISM band is open to anyone, systems operating on this band must deal with several unpredictable sources of interference, such as microwave ovens, baby monitors and 802.11 wireless networks [12]. Hence, to minimize the risk of such interference, Bluetooth uses a Frequency Hopping Spread Spectrum (FHSS) technology for its air interface. During a connection, radio transceivers hop from one channel to another. This means that after one packet is sent on a

channel, the two devices retune their frequencies (hop) to send the next packet on a different channel. When the transmission encounters a disturbance due to interference, the packet will simply be retransmitted on a different channel. Hence, if one frequency channel is blocked, there will be a limited disturbance to the Bluetooth communication. This allows several Bluetooth networks to run concurrently without interrupting one other. The link rate offered by Bluetooth is 1 Mbps, but with overhead, this effectively becomes 721 kbps. The typical range for Bluetooth is 10m, but it can reach up to 100m depending on the power class of the device.

Piconets and Scatternets

Two or more Bluetooth units sharing the same channel form a piconet. One device acts as a master and the devices connected to it act as slaves. The slaves in a piconet can only have links to the master. Slaves cannot directly transmit data to one another. In effect, the master acts as a switch for the piconet and all traffic must pass through the master. Any device can be either a master or a slave within a piconet and they can change roles at any point in a connection when a slave wants to take over a master's role. There can be up to 7 active slaves in a piconet but only one master.

Every Bluetooth device has its own clock and can be uniquely identified by its Bluetooth device address. Slaves in a piconet use the master's Bluetooth device address and clock to determine the frequency hopping sequence. Offsets are added to the native clocks of each of the slaves to synchronize with the master's clock for the duration of the connection [**5**]. Furthermore, the master also controls when devices transmit data, since slaves can only transmit when scheduled by the master. Hence, in deciding when and how often it communicates with the slaves, the master effectively controls how the total available bandwidth is distributed among the slaves [**8**].

A set of two or more interconnected piconets form scatternets. **Figure 2** shows an illustration of piconets and scatternets. A Bluetooth unit can be a slave in two or more piconets, but it can be a master in only one. Devices that participate in two or more piconets may act as gateways, forwarding traffic from one piconet to another [**14**]. Moreover, since Bluetooth units can only transmit and receive data in one piconet, its participation in several piconets is on a Time Division Multiplex (TDM) basis. This means that although devices can participate in several piconets, they may be active in only one piconet at any one time. Hence, devices participating in multiple piconets divide their time between the piconets, spending some time slots in one and some time slots in another. Piconets may be identified by the master's identity and clock. A device wishing to be active in another piconet will have to notify the master of its current piconet that it will be inactive for a predetermined length of time. The device will then have to re-synchronize its clock (by adding an offset) with its

other master. When a slave becomes inactive in a piconet, communications between masters and the other active slaves go on as normal. On the other hand, when a master becomes inactive in its piconet, the slaves will have to wait for it to be active again before communication can resume.

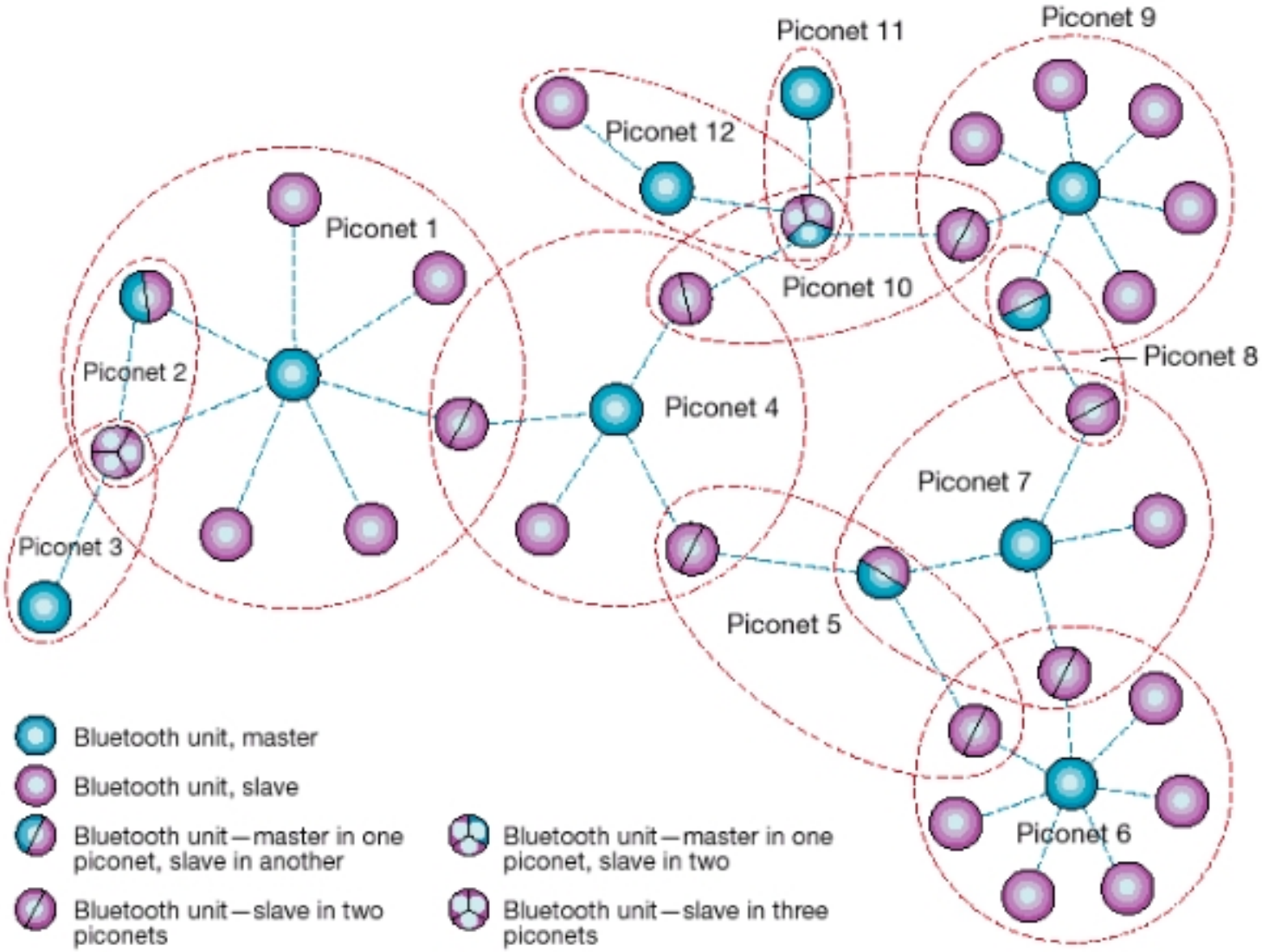

Figure 2: Bluetooth Piconets and Scatternets [10].

The Bluetooth Protocols

The Bluetooth protocols contain the standard procedures for connections and data exchange between Bluetooth devices. **Figure 3** shows the Bluetooth protocol stack [5, 8]. The Radio is the interface between the on-air channel medium and the Baseband. The Baseband layer is responsible for channel coding and decoding. It digitizes the signals received by the radio for passing up the stack and it formats the data it receives from the Link Controller for transmission over the channel. The Link Controller is responsible for establishing and maintaining the links between Bluetooth units. The Link Manager Protocol (LMP) handles

piconet management and link configuration. It also includes procedures for enforcing link security, such as encryption and authentication procedures.

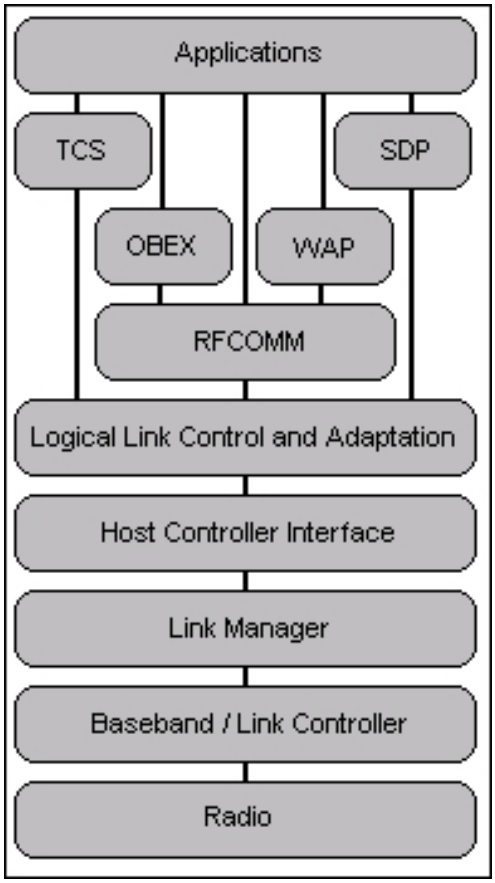

Figure 3: The Bluetooth protocol Stack [**8**].

The Host Controller Interface (HCI) defines uniform methods for accessing and controlling the lower layers of the protocol stack, namely the baseband and the link manager. Directly above it, the Logical Link Control and Adaptation Protocol (L2CAP) provides connection-oriented and connectionless data services to the other higher level protocol layers. Its protocol multiplexing capabilities allow different protocols and services to use one baseband link. The remaining protocols all utilize L2CAP links (and hence are positioned on top of that protocol). The Service Discovery Protocol (SDP) defines procedures for discovering services of other devices as well as determining the characteristics of those services. The RFCOMM protocol defines a transport protocol for emulating RS-232 serial ports. The Telephony Control Protocol Specification (TCS) defines call control signaling for establishing speech and data calls between Bluetooth devices, providing them with telephony services. The Object Exchange Protocol (OBEX) is a specification for object data exchange over infrared (IR) links. Examples for using OBEX include exchanging business cards and synchronizing calendar applications. The Bluetooth technology uses the IrOBEX protocol specification to allow applications to function over both short-range RF and IR, allowing applications to choose to use either. In the same way, the Wireless Application Protocol (WAP) includes interoperability requirements for Bluetooth as a WAP user. This allows one device to use WAP over Bluetooth links providing value-added services.

There are three ways of implementing a Bluetooth protocol stacks as illustrated in **Figure 4**. It may use the standard two-processor architecture, the embedded architecture, or the single-processor architecture [**9**]. In the two-processor architecture (**Figure 4a**), the Bluetooth host resides on the PC while the lower level protocols are encapsulated in a Bluetooth module. This architecture is usually implemented in add-on Bluetooth modules or PC-cards for personal computers and notebook computers. The second approach (**Figure 4b**) still uses two processors but most of the protocol layers are on the target processor. This architecture is usually used in resource-limited devices, such as mobile phones and handhelds. The third architecture (**Figure 4c**) is the single-processor architecture, used in system-on-a-chip or single-chip solutions.

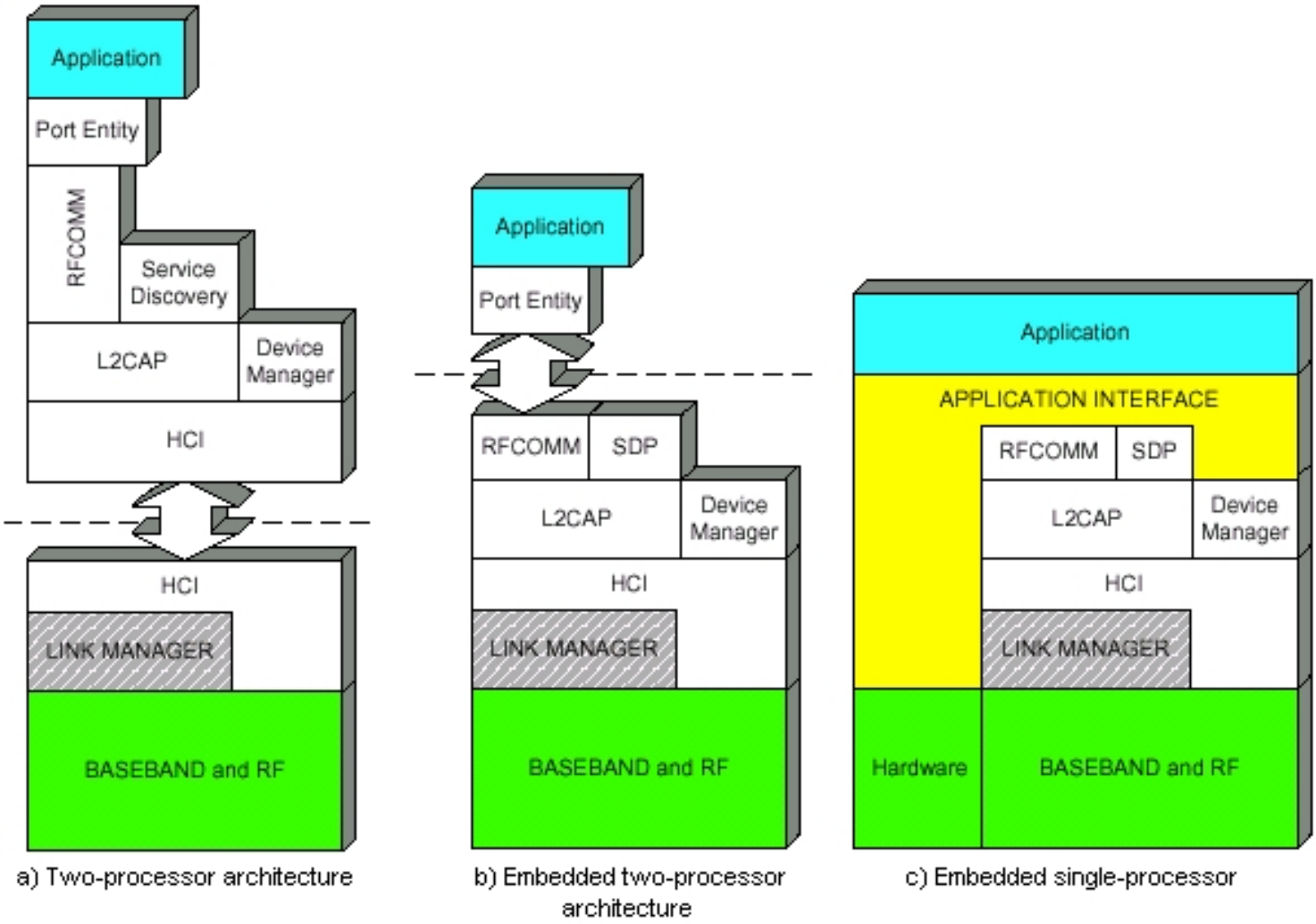

Figure 4: Bluetooth Stack Partitioning [**16**].

The Bluetooth Profiles

The Bluetooth profiles [**6**] contain guidelines on how applications should use the Bluetooth protocol stack. **Figure 5** shows an illustration of the Bluetooth profiles. The guidelines

defined in the profiles ensure interoperability between devices regardless of their manufacturers. All features defined in profiles are process-mandatory. This means that if a device implements a certain feature, the device implements it in a specified, standard manner.

Figure 5: Bluetooth Profiles [1].

Profiles can build upon other profiles allowing reuse of features and functionalities defined in other profiles. As seen in the figure, all profiles build upon the Generic Access Profile (GAP). For example, the Serial Port Profile builds upon the guidelines for device discovery and connectability when defining its own guidelines for setting up RFCOMM links for serial port emulation.

Devices do not need to implement all the profiles. A device only needs to implement the ones needed to support its applications. An exception however, is the Generic Access Profile, which is required in all devices. This profile defines the following (from [8]):

- Requirements for features which must be implemented in all devices
- Generic procedures for discovering Bluetooth devices
- Link management facilities for connecting to Bluetooth devices
- Procedures related to use of different security levels
- Common format requirements for device parameters accessible on the user interface level

Discovery and connection procedures ensure that all Bluetooth devices will recognize other Bluetooth devices and be able to connect to them. Having required features for Bluetooth units will allow developers to make assumptions about other devices when designing their applications. On the other hand, terminologies to be used at the user interface level will help users to recognize the features and functions across different devices, platforms and interface designs.

When version 1.1 of the Bluetooth Profile Specification came out, thirteen profiles were defined (as seen in **Figure 5**). Since then, several new profiles have been developed, including the Human Interface Device Profile, the Personal Area Networking Profile, and the Basic Printing Profile. The Human Interface Device Profile [3] defines guidelines for devices such as keyboards, joysticks, as well as controls used for games and simulation devices such as data gloves and steering wheels. The Personal Area Networking Profile [4] describes how two or more Bluetooth devices can form ad hoc networks as well as how they can use a

network access points to access remote networks. The Basic Printing Profile [2] defines the requirements and procedures to support the Basic Printing usage model. This usage model is defined for mobile phones and PDAs for printing text messages, short email messages, business cards (vCards), and other formatted documents. Considerable effort is still being given to the development of other Bluetooth profiles as more usage scenarios are built for Bluetooth networks.

Security of Bluetooth

The frequency hopping scheme used by the Bluetooth technology already makes listening in on Bluetooth links very difficult. In fact, the U.S. military considers a communication link using frequency hopping over 79 channels to be secure [8]. Nevertheless, Bluetooth offers encryption and authentication using an algorithm based on the SAFER+ (Secure And Fast Encryption Routine) cipher algorithm. This algorithm [8] generates 128 bit cipher keys from a 128 bit plaintext input. When initializing a security procedure, a 128 bit key is generated from a Personal Identification Number (PIN), the Bluetooth device address of the claimant, and a random number shared between the claimant and the verifier. The authentication procedure checks whether the two devices are using the same 128 bit key to verify that the same PIN number was entered on the two devices. If the authentication procedure is successful, a new 128 bit key is generated using a new random number from each unit, the Bluetooth device addresses of the two units, and the current 128 bit key. This key is used to produce the cipher stream to cipher and decipher the bitstream data.

Bluetooth also introduces three security modes, which may be used by applications depending on their security requirements [6, 8]. Mode 1 is not secure. A device in mode 1 never initiates any security procedures. A device in mode 2 enforces security procedures at service-level. Only after a L2CAP channel has been established is any security procedure performed. Depending on the application, this can include authorization, authentication, and encryption. On the other hand, a device in security mode 3 enforces security procedures at link level. Hence, when a device fails the security measures performed by a connecting device, no link is established between the two. As in security mode 2, security measures include authentication, encryption, and authorization.

In addition to these security measures, devices may also be "invisible" should they wish to stay private. Bluetooth allows a device to stay in an undiscoverable mode, where a device does not respond to inquiry scans. This means that even if a device is in range, it will never be discovered by any device performing an inquiry scan. A device may also stay in a non-connectable mode where although it responds to an inquiry, no device can ever establish a link with it, except when the device itself initiates the link.

The Connection Process

To demonstrate how Bluetooth devices discover and connect with one another, let us use the scenario of a laptop connecting wirelessly to a mobile phone to use dial-up networking (DUN) (adapted from [**8**]). This process is shown in **Figure 6**. Before any device can connect to another device, it has to initially look for devices that it might connect to. In Bluetooth, this is called the Inquiry Process. The inquiring device, which we call A, sends out an inquiry packet or repeated inquiry packets and waits to receive responses back. Discoverable devices in range respond to an inquiry by sending a Frequency Hop Synchronization (FHS) packet, which contains all the information device A needs to connect to the responding device, including the Bluetooth device's address, page scan modes, and clock offset. All devices that respond to the inquiry are reported to the host controller of device A. Whether or not the list of all devices discovered is presented to the user is application-dependent.

At this point, device A knows which devices are in range, but it does not yet know which devices support dial-up networking. Using the information retrieved from the inquiry process, device A now attempts to connect to the different devices that responded to its inquiry in order to find out what services they support. Depending on the application, device A may either 1) establish links to all the devices that responded to its inquiry and get the information about their services and later on reconnect with the one that supports dial-up networking; or 2) upon seeing that a device supports dial-up networking, directly proceed to setting up a connection with that device without finding out the services from the rest of the devices in the list. In **Figure 6**, the second option is adopted.

In order to find out the services of a device, device A sends out paging packets. A connectable device will respond and a baseband link can be established between the two devices. Following that, a L2CAP connection will be established before they can exchange service information. This information exchange is handled by the Service Discovery Protocol. Say a device B has responded that it has the dial-up networking service. A RFCOMM connection can then be established across the already existing L2CAP link. When this has been set, a dial-up networking connection can then be established on top of the RFCOMM connection, after which the laptop can then start using the cell phone to access the phone network without any cables being needed for connections.

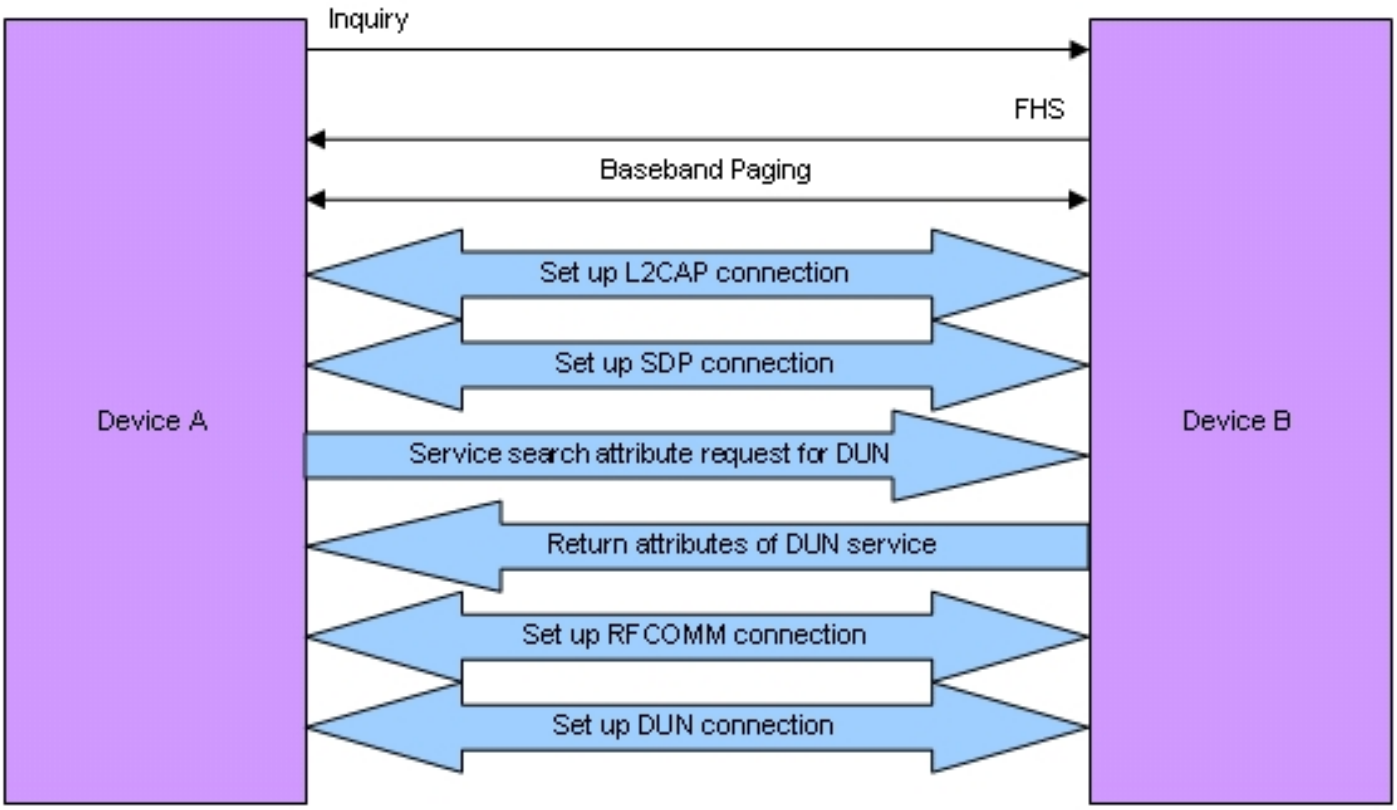

Figure 6: Setting up a Bluetooth connection.

Some Issues and Concerns

Since Bluetooth is a relatively new technology, parts of it are still under development. Two research themes often emerge. The first issue concerns the development of effective scheduling algorithms for Bluetooth scatternets. As discussed previously, a Bluetooth unit can only be active in one piconet but it can be a member in a number of piconets. Hence, a unit that participates in multiple piconets needs to divide its time among the different piconets. To schedule communication with such units, their availability in other piconets needs to be taken into consideration. This then becomes a scatternet-wide coordination problem which can easily become a bottleneck in scatternet organization. Several recent papers [**13**, **17**] have proposed scheduling algorithms that begin to address this concern.

The second issue looks at possible interference problems with IEEE 802.11 wireless networks. Because both IEEE 802.11 and Bluetooth wireless networks operate on the same frequency band, it is expected that some interference will occur when these two networks are present in the same environment. Indeed, simulations and experiments in [**11**, **15**] have shown that significant packet losses and access delays do occur. Hence, some co-existence mechanism may need to be developed to reduce the performance degradation when these two technologies are present in one area.

Aside from interference issues, it is also interesting to look at intersections in the 802.11b

wireless local area network (WLAN) technology and Bluetooth. Although IEEE 802.11b was originally designed for providing network access, it can also operate in ad hoc mode, making it possible to form ad hoc personal area networks (PANs) which is the niche Bluetooth is aiming for. Because of the widespread use of the IEEE 802.11b standard even in small handheld devices, there is reasonable concern that this technology might surpass Bluetooth in terms of usage and acceptance even in PANs. A study by Johansson, Kapoor, Kazantzidis and Gerla [**14**] compares the PAN capabilities of the two technologies. The principal difference between the two standards is that Bluetooth is connection-oriented while 802.11b is connectionless [**14**]. This implies that Bluetooth units need to set up connections before they can send any data, while for 802.11b, units can directly send data to any other unit in range. Table 1 shows a summary of the more detailed comparisons between the two technologies from [**14**].

	Bluetooth	**IEEE 802.11b**
Media Access Control	Based on controlling unit (master)	Random-access-oriented
Neighbor Discovery	Standardized discovery using the INQUIRY procedure	No defined way to discover unknown devices (may use broadcasting), but known devices can be directly addressed
Multihop PAN's	Involves scatternets - interconnected piconets	Straightforward - no piconet architecture, all nodes are peers
Power Consumption	Uses polling between masters and slaves; offers power saving modes	Units may receive packets from other units at any time, so receivers need to be active for long periods.

Table 1: A summary of comparisons between Bluetooth and IEEE 802.11.

The differences between these two standards directly affect how they perform against each other. Some of the advantages offered by the piconet architecture of Bluetooth are support for Quality of Service (QoS), heavy traffic management, and energy efficiency. Bluetooth has the capability to handle network traffic with Quality of Service (QoS) requirements because the master has control over the allocation of resources. However, for traffic that does not require QoS support, IEEE 802.11b might be a better choice since it offers a higher bit rate (11 Mbps) than Bluetooth (1 Mbps).

The Frequency Hop Spread Spectrum approach used in Bluetooth avoids packet collisions between piconets, although it offers a lower capacity per channel. This gives Bluetooth the capability to actually add capacity as more units / piconets are added to the network. In IEEE 802.11b, the units have access to wider channels, but interference will cause units to share a single channel, which leads to increased packet collisions. Moreover, in the piconet, the slaves can only send data when replying to data sent by the master. This scheme also reduces packet collisions and saves power since the receivers of the slaves only need to be active for specific time slots. In addition to this, Bluetooth offers power saving modes where the slaves may choose not to receive packets from the master for certain time intervals. On the other hand, IEEE 802.11b units should be able to receive data from other units, so their receivers need to be active for long periods of time. However, once the two units start exchanging data, they may announce the time expected for the data exchange. During the duration of this exchange, the units may also enter sleep mode.

One of the more interesting results in the experiments from [**14**] is that IEEE 802.11 performed better in low loads but Bluetooth handled dense network conditions better. Not only did the capacity of the Bluetooth network increase as more piconets were added, it also demonstrated fair bandwidth allocation. This is in contrast to the performance of IEEE 802.11b, where the number of collisions increased with the increasing number of nodes and where the allocation of bandwidth was unfairly distributed. Moreover, the results have also shown that Bluetooth is more energy efficient than IEEE 802.11 when the number of piconets is greater than three. In fact, while the energy efficiency of IEEE 802.11 wireless networks decreases as the number of piconets increases, the energy efficiency of Bluetooth stays constant. An important note the authors make is that when choosing the technology for PANs, the deciding factors are not so clear-cut. Bluetooth may be the choice in PANs that operate in dense environments with high interference given its increasing capacity and energy efficiency. However, if PANs were to operate in sparse environments and low interference levels, then IEEE 802.11 could be a better choice.

Conclusion

Bluetooth is a fast growing technology. More and more devices are coming out with Bluetooth capabilities built into the hardware and applications utilizing the Bluetooth technology are starting to come out as well. Although Bluetooth-enabled devices still cost considerably more than devices without Bluetooth capabilities, they do seem to be finding their niche in the market.

Parallel to market growth, Bluetooth is also undergoing continuous technical development. Even more profiles will be defined as new usage models emerge. There is also on-going research into improving Bluetooth network performance, not only alongside other wireless

networks, such as IEEE 802.11, but also among other Bluetooth networks as well. Furthermore, Bluetooth v2.0 is already under construction. Expected improvements from the current version include higher data rates and faster response times.

However, in order to become a widely accepted technology, several factors need to come into play. Being primarily a cable replacement technology, the price of enabling a device to use the Bluetooth technology should not cost significantly more than the cable it replaces. Furthermore, using Bluetooth should not be any more complicated than using cables. In particular, the connection process should be as seamless as plug and play technologies. Minimal configuration should be left to users and dependable service should be provided. This is certainly lacking in current Bluetooth solutions where a complicated set-up process and upnpredictable service has been known to frustrate potential users [**7**].

Another important driver for this technology will be killer applications. Many of the available Bluetooth hardware in the market today are already starting to address interoperability issues which plagued early releases. However, applications available today are mostly single solutions such as wireless headsets for mobile phones and wireless keyboards, which usually work with only their targeted hardware. Users need to see integrated solutions and ones that deliver on the promise of interoperability. With applications that actually support the usage models, and single-chip solutions that actually meet the targeted $5 price (at high volumes), we might very well see Bluetooth in the mainstream.

Finally, regarding issues concerning the IEEE 802.11 network, the trend seems to be that both technologies are falling into their own categories. Bluetooth is emerging as the PAN technology as initially envisioned while IEEE 802.11 is settling into its WLAN category. In the near future, where users will be having a plethora of mobile devices around them, Bluetooth's inherent capabilities for handling high network loads while maintaining low-power operation and higher throughput might turn out highly valuable. However, peaceful co-existence between the two technologies will have to be addressed.

References

1

Arfwedson, H. and Sneddon, R. *Ericsson's Bluetooth modules.* Ericsson Review No. 4, 1999.

2

Bluetooth SIG. *Basic Printing Profile Interoperability Specification.* 2001. <**http://www.bluetooth.com/dev/specifications.asp**>

3

Bluetooth SIG. *Human Interface Device Profile.* 2001. <**http://www.bluetooth.com/dev/specifications.asp**>

4

Bluetooth SIG. *Personal Area Networking Profile.* 2001. <**http://www.bluetooth.com/dev/specifications.asp**>

5

Bluetooth SIG. *Specification of the Bluetooth System, Core v1.1.* 2001. <**http://www.bluetooth.com/dev/specifications.asp**>.

6

Bluetooth SIG. *Specification of the Bluetooth System, Profiles v1.1.* 2001. <**http://www.bluetooth.com/dev/specifications.asp**>

7

Bray, H. *Teething Pains.* The Boston Globe, November 11, 2002.

8

Bray, J. and Sturman, C.F. *Bluetooth Connect Without Cables.* Prentice Hall PTR 2001.

9

Caccam, A., Dideles, M., Galang, B. and Wong, I. *Bluetooth Host-side Protocol Stack Development using Formal Design Techniques.* Philippine Engineering Journal, 2001.

10

Frodigh, M., Johansson, P. and Larsson, P. *Wireless ad hoc networking - The art of networking without a network.* Ericsson Review No. 4, 2002.

11

Golmie, N., Van Dyck, R.E. and Soltanian, A. *Interference of Bluetooth and IEEE 802.11: Simulation Modeling and Performance Evaluation.* The Fourth ACM International Workshop on Modeling, Analysis and Simulation of Wireless and Mobile Systems, 2001.

12

Haartsen, J. *BLUEOOTH - The universal radio interface for ad hoc, wireless connectivity.* Ericsson Review No. 3, 1998.

13

Johansson, N., Alriksson, F., Jonsson, U. *JUMP Mode - A Dynamic Window-based Scheduling Framework for Bluetooth Scatternets.* The Second ACM International Symposium on Mobile Ad Hoc Networking and Computing 2001.

14

Johansson, P., Kapoor, R., Kazantzidiz, M. and Gerla, M. *Personal Area Networks: Bluetooth or IEEE 802.11?* International Journal of Wireless Information Networks, Vol. 9, No. 2. 2002.

15

Kapoor, R., Kazantzidis, M., Gerla, M. and Johansson, P. *Multimedia Support Over Bluetooth Piconets.* First Workshop on Wireless Mobile Internet, July 2001.

16

Mezoe. *A Brief Introduction to BlueStack.* <**http://www.mezoe.com**>.

17

Racz, A., Mikos, G., Kubinszky, F. and Valko, A. *A Pseudo Random Coordinated*

Scheduling Algorithm for Bluetooth Scatternets. The Second ACM International Symposium on Mobile Ad Hoc Networking and Computing 2001.

Steward, W., Mann, B. and Gilster, R. *Wireless Devices End to End.* Hungry Minds, Inc, 2002.

Biography

Myra Dideles (**cdideles@ccs.neu.edu**) is a graduate student at Northeastern University in Boston, Massachusetts working on her Master of Science degree in Computer Science. She worked at the Advanced Science and Technology Institute and was involved in the development of a Bluetooth protocol stack prototype. Her current interests include mobile applications, pervasive and ubiquitous computing, context-aware computing, and more generally, the human aspects of computing.

Acknowledgements

The author would like to thank her advisor, Prof. Peter Tarasewich for his guidance throughout the writing process and Bien Galang for his help in clarifying some Bluetooth issues.