FORUM NOKIA

Bluetooth Technology Overview

Version 1.0; April 4, 2003

NOKIA

Contents

Change History

April 4, 2003	V1.0	Document published

Disclaimer

The information in this document is provided ”as is,” with no warranties whatsoever, including any warranty of merchantability, fitness for any particular purpose, or any warranty otherwise arising out of any proposal, specification, or sample. Furthermore, information provided in this document is preliminary, and may be changed substantially prior to final release. This document is provided for informational purposes only.

Nokia Corporation disclaims all liability, including liability for infringement of any proprietary rights, relating to implementation of information presented in this document. Nokia Corporation does not warrant or represent that such use will not infringe such rights.

Nokia Corporation retains the right to make changes to this specification at any time, without notice.

The phone UI images shown in this document are for illustrative purposes and do not represent any real device.

License

Bluetooth Technology Overview

Version 1.0; April 4, 2003

1 Purpose

The following document provides an overview of Bluetooth technology. It emphasizes Bluetooth profiles and high-level protocols, as the Application Programming Interfaces (APIs) mostly rely on them. Thus, the information in this document should serve as a foundation for software application developers.

This document does not contain any information related to application programming interfaces or development tools. Bluetooth protocols and profiles introduced here do not necessarily refer to existing Bluetooth products on the market.

2 Basics of Bluetooth

Bluetooth is a short-range radio technology that enables wireless connectivity between mobile devices. Design-wise, the three main goals for Bluetooth were: small size, minimal power consumption, and low price. The technology was designed to be simple, and the target was to have it become a de facto standard in wireless connectivity.

Bluetooth radio operates in the unlicensed ISM band at 2.4 GHz [Core, p.19]. In some countries, this band is reserved for military use, but these countries have now begun freeing that band for general use. The maximum gross data rate is 1 Mbps [Core, p.41].

The range of Bluetooth depends on the power class of the radio. Most devices are expected to use the class 2 radio that provides 0 dBm nominal output power, resulting in a range of up to 10 meters in an obstacle-free environment. This range is sufficient for cable-replacement applications. When a longer range is needed (e.g., in access points), a more powerful radio (class 1) can be used. Larger power consumption is not a problem if the device is a piece of fixed equipment. With mobile devices such as mobile phones, power-consumption issues are crucial and therefore class 2 is the only feasible option.

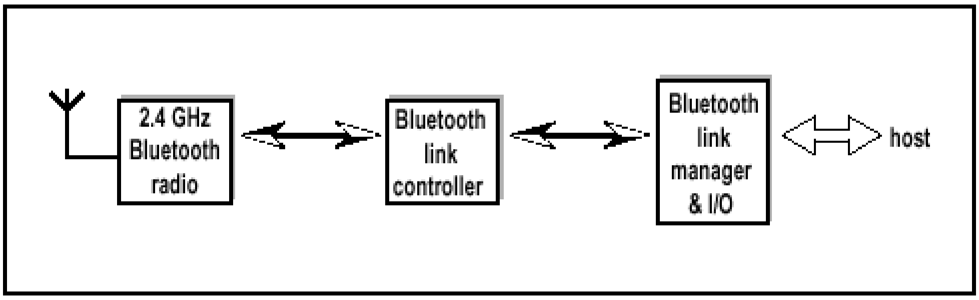

Figure 1. Bluetooth system blocks [Core, p.41]

The Bluetooth system consists of a radio unit, a link control unit, and a support unit for link management and host terminal interface functions (see Figure 1). The Host Controller Interface (HCI) provides the means for a host device to access Bluetooth hardware capabilities [Core, p. 543]. For example, a laptop computer could be the host device and a PC card inserted in the PC is the Bluetooth device. All commands from the host to the Bluetooth module and events from the module to the host go through the HCI interface. The protocol stack is above the radio and baseband hardware, partly residing in the Bluetooth unit and partly in the host device.

A Bluetooth solution can also be implemented as a one-processor architecture (embedded solution) where the application resides together with the Bluetooth protocols in the same hardware. In that case, the HCI is not needed. This is a feasible implementation for simple devices such as accessories or micro servers.

3 Air Interface

3.1 Piconet and Scatternet

The Bluetooth network is called a *piconet*. In the simplest case it means that two devices are connected (see Figure 2a). The device that initiates the connection is called a *master* and the other devices are called *slaves*. The majority of Bluetooth applications will be point-to-point applications. Bluetooth connections are typically ad hoc connections, which means that the network will be established just for the current task and then dismantled after the data transfer has been completed.

A master can have simultaneous connections (point-to-multipoint) to up to seven slaves (see Figure 2b). Then, however, the data rate is limited. One device can also be connected in two or more piconets. The set-up is called *scatternet* (see Figure 2c). A device can, however, only be a master to one piconet at a time. Support for hold, park, or sniff mode is needed for a device to be part of the scatternet. In these modes a device does not actively participate in a piconet, leaving time for other activities such as participating in another piconet, for example.

The master/slave roles are not necessarily fixed and can also be changed during the connection if, for example, the master does not have enough resources to manage the piconet. Master/slave switch is also needed in the scatternet. Master/slave switch support is not mandatory.

Most of current Bluetooth implementations support piconets only. Point-to-multipoint support depends on the implementation.

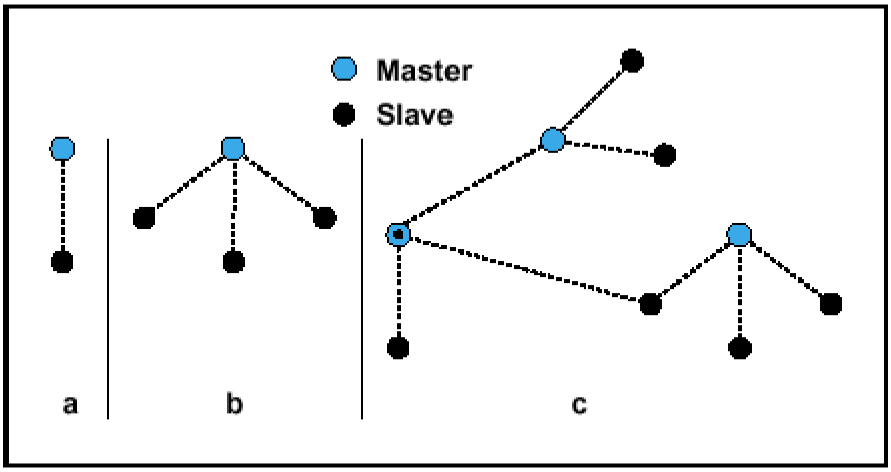

Figure 2. Bluetooth piconet and scatternet scenarios:

a) Point-to-point connection between two devices

b) Point-to-multipoint connection between a master and three slaves

c) Scatternet that consists of three piconets [Core, p.42]

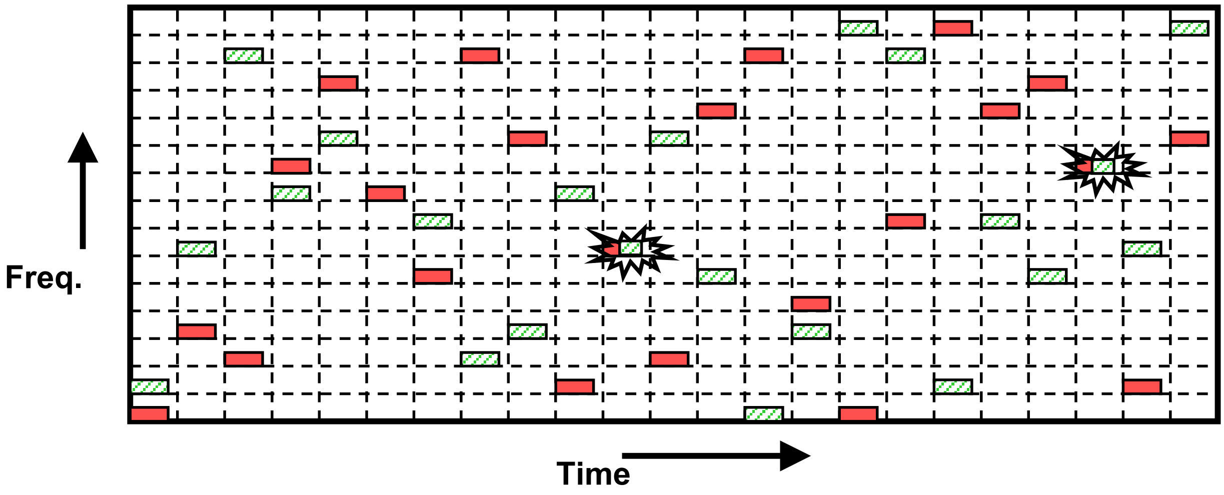

Figure 3. Example of Bluetooth frequency hopping

3.2 Frequency Hopping

Bluetooth technology uses a frequency hopping technique, which means that every packet is transmitted on a different frequency. In most countries, 79 channels can be used. With a fast hop rate (1600 hops per second), good interference protection is achieved. Another benefit is a short packet length. If some other device is jamming the transmission of a packet, the packet is resent in another frequency determined by the frequency scheme of the master. This scenario is depicted in Figure 3 where packets of device 1 (colored packets) and device 2 (banded packets) are trying to use the same frequency. Note that this case only refers to situations where there are two or more simultaneous active piconets or a non-Bluetooth device using the same frequency in range. The error correction algorithms are used to correct the fault caused by jammed transmissions [Whitepaper1, p.10].

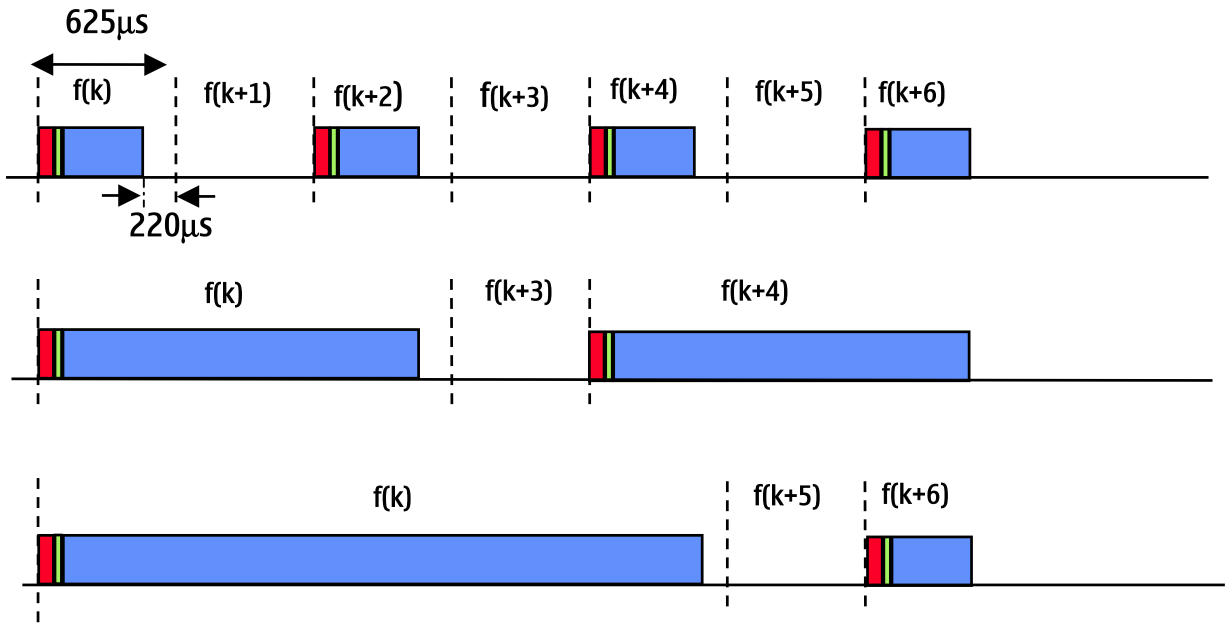

Figure 4. Three-slot and five-slot long packets reduce overhead compared to one-slot packets. 220 µs switching time after the packet is needed for changing the frequency.

Subsequent time slots are used for transmitting and receiving. The nominal slot length is 625 µs. A packet nominally covers a single slot, but can be extended to cover three or five slots, as depicted in Figure 4. In multi-slot packets the frequency remains the same until the entire packet is sent. When using a multi-slot packet, the data rate is higher because the header and a 220 µs long switching time after the packet are needed only once in each packet. On the other hand, the robustness is reduced: in a crowded environment the long packets will more probably be lost [Core, p.41].

3.3 Links and Packets

The **Asynchronous Connectionless (ACL) links** are defined for data transmission, primarily packet data. They support symmetrical and asymmetrical packet-switched connections. Multi-slot packets use the ACL link type and can reach the maximum data rate of 723 kbps in one direction and 57.6 kbps in the other direction. The master controls the ACL link bandwidth and decides how much of the bandwidth a slave can use in a piconet. Broadcast messages are supported in the ACL link, i.e., from the master to all slaves in the piconet [Whitepaper1, p.12].

The **Synchronous Connection Oriented (SCO) links** support symmetrical, circuit-switched, point-to-point connections and are therefore primarily used for voice traffic. Two consecutive time slots at fixed intervals are reserved for an SCO link [Whitepaper1, p.12]. The SCO link reserves every sixth slot for a transmitting channel and the subsequent slot for a receiving channel, so there can be up to three simultaneous SCO links. The data rate for SCO links is 64 kbps [Core, p.58].

Data is transmitted in packets. Each packet consists of three entities: the access code, the header, and the payload [BISG1, p.47]. The construction of the packet and the number of bits per entity are shown in Figure 5. The size of the access code and the header are fixed. The payload may range from 0 to 2745 bits per packet. The control packets may also consist of the access code only, or of the access code and header only. In ACL packets all three entities are needed.

Three methods are used for ensuring reliable data transfer in crowded environments. In the **Forward Error Correction (FEC)** scheme, additional check bits are added in the packet header or the payload. In the **Automatic Repeat Request (ARQ)** scheme, the data payload is retransmitted until the recipient sends an acknowledgment. Acknowledgement information is included in the header of the return packet. To determine whether the payload is correct or not, a **Cyclic Redundancy Check (CRC)** code is added to the packet [Profiles, pp.67-68].

The most commonly needed ACL packets are DM1, DH1, DM3, DH3, DM5, and DH5 (see **Table 1**). The numbers indicate the length of the packet (single-slot, triple-slot, or five-slot packets). In addition to information bytes, the payload contains a 16-bit CRC code. Retransmission is applied if no acknowledgement of proper reception is received. In DM (Data - Medium rate) packets the payload is 2/3 FEC encoded, i.e., additional check bits are added. In DH (Data - High rate) packets the information bytes are not encoded at all. AUX1 packet has no CRC code and it is not retransmitted [Core, pp.60-61].

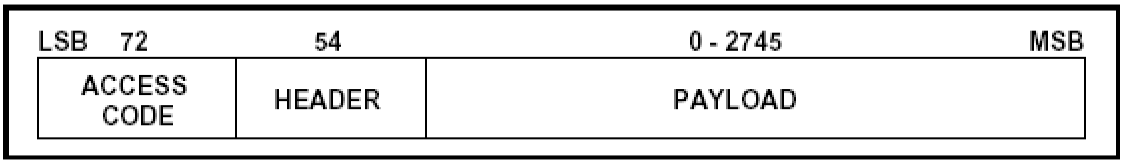

Figure 5. Standard packet format [BISG1, p.47]

Table 1 ACL packets: DM1, DM3, and DM5 use rate 2/3 FEC encoding [Core, p.65]

Type	Payload Header (bytes)	User Payload (bytes)	FEC	CRC	Symmetric Max. Rate (kb/s)	Asymmetric Max. Rate (kb/s)	
						Forward	Reverse
DM1	1	0-17	2/3	yes	108.8	108.8	108.8
DH1	1	0-27	no	yes	172.8	172.8	172.8
DM3	2	0-121	2/3	yes	258.1	387.2	54.4
DH3	2	0-183	no	yes	390.4	585.6	86.4
DM5	2	0-224	2/3	yes	286.7	477.8	36.3
DH5	2	0-339	no	yes	433.9	723.2	57.6
AUX1	1	0-29	no	no	185.6	185.6	185.6

3.4 Example: Mixed Links

The following example describes how multiple connections can be handled simultaneously (see Figure 6). Let us assume that the master is a mobile phone. The phone has already established connections to three slaves (wireless headset, printer, home-lighting system). In time slots 1 and 2 (black boxes), a call is ongoing (also in slots 7-8,13-14, and 19-20). In the following two slots, the user adjusts the volume and the headset acknowledges it. In slots 5 and 6, the user turns on the light in the room and receives an acknowledgement. In slots 9-12, the user starts printing out a note from her phone and the printer acknowledges it. And finally, in slots 21 and 22 the call will hang up.

For the call, both the phone and the headset have to reserve every sixth slot (the black boxes). That means that between phone/headset links only four slots are free for additional communication. Thus five-slot data packets cannot be sent while having a voice link.

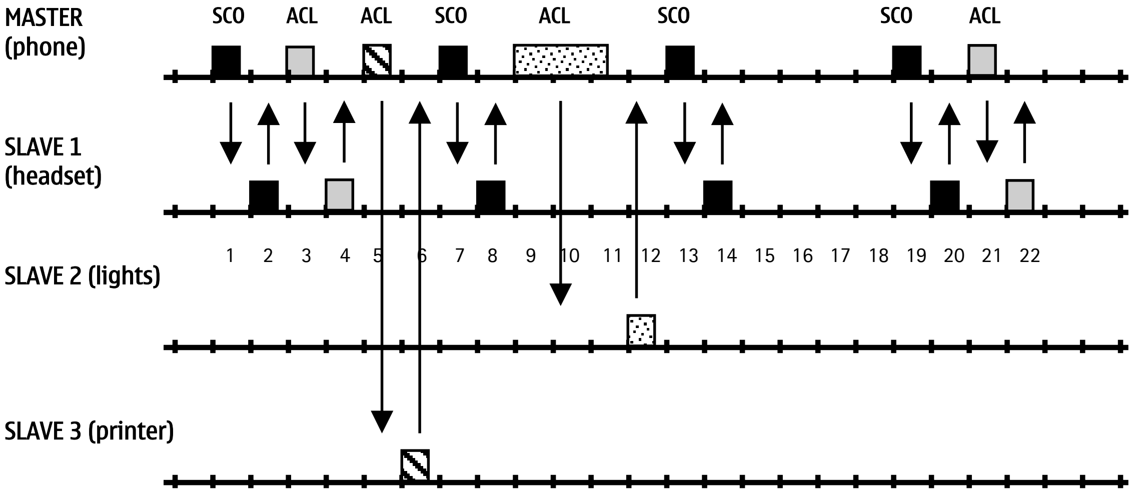

Figure 6. Example of multiple simultaneous links.

This theoretical scenario may not be feasible in practice but it gives a good idea of the multitasking properties of Bluetooth technology, supporting different applications simultaneously without any noticeable effect for the user. (In applications where data is transmitted in the background the data transfer speed is irrelevant.)

3.5 Connection Establishment and Inquiry

The connection to a desired device is made by a **page** message (Figure 7). If the address of the recipient is unknown, an **inquiry** message is needed before paging. Before any connections are made, all units are in **standby** mode. A unit in a standby mode wakes up every 1.28 seconds to listen to page/inquiry messages. Each time a unit wakes up, it listens on one of the 32 defined hop frequencies.

The page message will be sent on 32 different frequencies. Initially the message is sent on the first 16 frequencies, 128 times, and if no response is received, the master sends a page message on the remaining 16 frequencies, 128 times. The maximum connection time is 2.56 seconds [Whitepaper2, p.11].

When paging, the master must know the slave's Bluetooth address and system clock to calculate the proper access code and the wake-up sequence phase. That information was provided in the inquiry process [Whitepaper1, p.13].

In inquiry, the master sends an inquiry access code, and other devices respond with their identity and system clock. After that, the connection can be made with any of those devices using the paging procedure described earlier.

In **connection** state, the Bluetooth unit can be in several modes of operation. Sniff, hold, and park modes are used to save power or to free the capacity of a piconet:

Active mode: In the active mode, the Bluetooth unit actively participates on the channel.

Sniff mode: In the sniff mode, the duty cycle of the slave's listen activity can be reduced. This means that the master can only start transmission in specified time slots.

Hold mode: While in connection state, the ACL link to a slave can be put in a hold (possible SCO links are still supported). In hold mode, the slave can do other things, such as scanning, paging, inquiring, or attending another piconet (scatternet scenario, see Section 3.1).

Park mode: If a slave does not need to participate in the piconet but still wants to remain synchronized to the channel (to participate in the piconet again later), it can enter the park mode. It gives up its active member address. Park mode is useful if there are more than seven devices that occasionally need to participate in the same piconet. The parked slave wakes up regularly to listen to the channel in order to re-synchronize and to check for broadcast messages sent by the master.

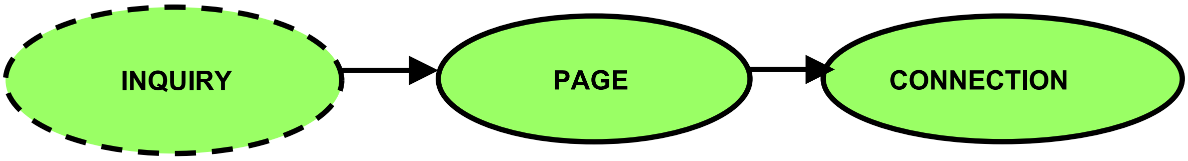

Figure 7. Inquiry and page procedures lead to connection state

4 Bluetooth Profiles

The Bluetooth Special Interest Group (SIG) has defined a number of usage models for Bluetooth technology. They describe the main Bluetooth applications and the intended devices, e.g., the synchronization between a handheld device and a PC, and connecting to the Internet wirelessly using a mobile phone or a cordless modem.

Profiles specify how the interoperable solution for the functions described in the usage models is provided; in other words, a profile defines the protocols and protocol features supporting a particular usage model. Bluetooth v1.1 profiles are shown in Figure 8. Some profiles are dependent on other profiles. For example, three profiles (File Transfer Profile, Object Push Profile, and Synchronization Profile) are dependent on the Generic Object Exchange Profile. All profiles are dependent on the Generic Access Profile, i.e., they are reusing it.

The usage models have gradually lost their significance and it is more illustrative to talk about corresponding profiles. Thus, usage models have been omitted here and the emphasis is on profiles.

Bluetooth products support different sets of profiles. In order to support a certain profile, mandatory features of the profile must be implemented.

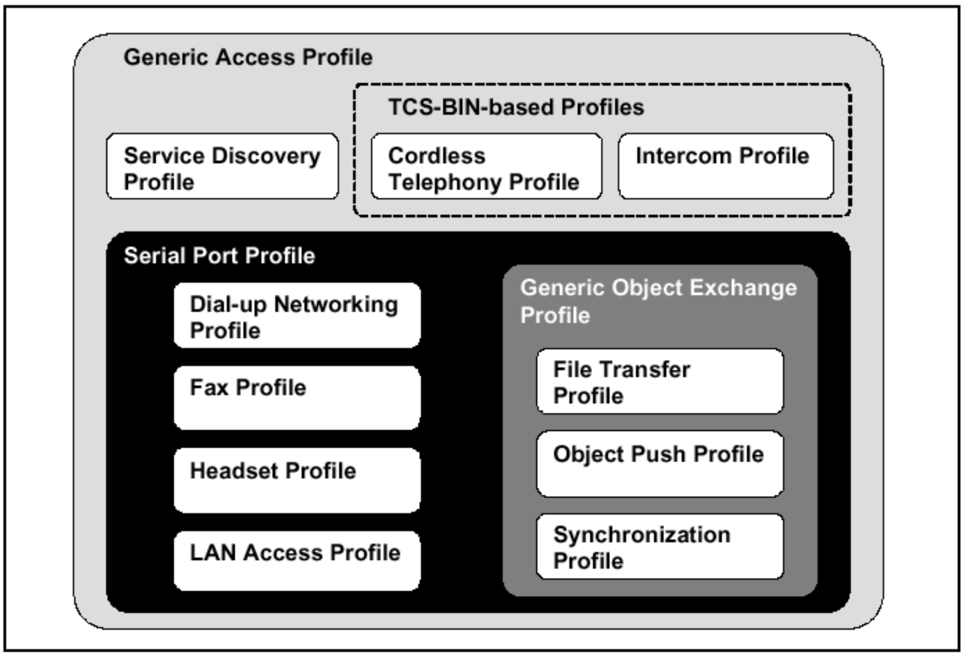

Figure 8. Bluetooth profile dependencies [Profiles, p.177]

4.1 The Four General Profiles in the Bluetooth Specification v1.1

Generic Access Profile defines the generic procedures related to discovery of Bluetooth devices (idle mode procedures) and link management aspects of connecting to Bluetooth devices (connecting mode procedures). It also defines procedures related to use of different security levels. In addition, this profile includes common format requirements for parameters accessible on the user interface level. Every Bluetooth device has to support the Generic Access Profile [Profiles, p.13].

Service Discovery Application Profile defines the features and procedures for an application in a Bluetooth device to discover services of another Bluetooth device [Profiles, p.63].

Serial Port Profile defines the requirements for Bluetooth devices necessary for setting up emulated serial cable connections using RFCOMM between two peer devices [Profiles, p.171].

Generic Object Exchange Profile defines the protocols and procedures that will be used by applications that need object exchange capabilities. Possible scenarios are synchronization, file transfer, and object push [Profiles, p.315].

4.2 Usage Model-Oriented Profiles

Cordless Telephony Profile and **Intercom Profile** define the features and procedures required for interoperability between different units active in the "three-in-one phone" usage model (the same phone can be used as a cordless phone, a walkie-talkie, and a cellular phone). The Cordless Telephony Profile is used when the phone is connected to a base station of fixed telephony network via Bluetooth and the Intercom Profile implements so-called "walkie-talkie" usage between Bluetooth phones [Profiles, pp.105, 452].

Dial-Up Networking Profile describes how to use a cellular phone or a modem beside a computer as a wireless modem to receive data calls, to connect to a dial-up Internet access server, or to use other dial-up services [Profiles, p.231].

Fax Profile defines how a computer can use a Bluetooth cellular phone or modem as a wireless fax modem to send or receive a fax message [Profiles, p.255].

Headset Profile defines the requirements for Bluetooth devices necessary to support the headset use case. Wireless headsets can be used with cellular phones and laptops [Profiles, p.201].

LAN Access Profile defines how Bluetooth-enabled devices can access the services of a local-area network using PPP (Point-To-Point Protocol) over RFCOMM (Bluetooth protocol that emulates RS-232 signal) and how the same PPP mechanisms are used to form a network consisting of two Bluetooth-enabled devices [Profiles, p.277].

File Transfer Profile covers the scenarios that enable the user to browse and edit objects (files and folders) in the file system of another Bluetooth device and to transfer objects between two Bluetooth devices. The most common devices in question are PCs, notebooks, and PDAs [Profiles, p.371].

Object Push Profile covers the scenarios that enable users to push, pull, and exchange simple objects such as business cards between two Bluetooth devices such as notebook PCs, PDAs, and mobile phones [Profiles, p.345].

Synchronization Profile covers the following scenarios: PIM data exchange between two devices and automatic synchronization of data (e.g., calendar items) when a device enters the proximity of the computer. Synchronization can be used between notebooks, PDAs, and mobile phones [Profiles, p.403].

4.3 Additional Profiles

The Bluetooth specification initially included thirteen profiles as described in Section 4.1 and 4.2. To guarantee interoperability in many application areas, the working groups of the Bluetooth SIG are specifying new profiles. These will be published independently. Twelve additional profiles have already been published [Profiles2]:

Generic Audio/Video Distribution Profile (GAVDP) defines a generic part of the protocols and procedures that realize distribution of audio/video content using ACL channels.

Advanced Audio Distribution Profile (A2DP) defines distributing of audio content of high quality in mono or stereo on ACL channels. A2DP is dependent upon GAVDP.

Audio/Video Remote Control Profile (AVRCP) defines transmission of a user-activated A/V control signal to a remote Bluetooth device.

Basic Imaging Profile (BIP) is an OBEX-based profile that enables devices to negotiate the size and encoding of imaging data to be exchanged.

Basic Printing Profile (BPP) is an OBEX-based profile that enables printing of text e-mails, short messages, and formatted documents from mobile devices.

Hardcopy Cable Replacement Profile (HCRP) is a lightweight profile implementation for printing and scanning any type of document. HCRP is implemented directly on top of L2CAP avoiding the overhead from OBEX, RFCOMM, or PAN.

Bluetooth Extended Service Discovery Profile (ESDP) for Universal Plug and Play™ (UPnP™) is a profile for discovering other devices that support UPnP services and retrieve information about the services.

Hands-Free Profile (HFP) defines a case where a mobile phone can be used in conjunction with a hands-free device (e.g., a car kit). HFP provides a wireless means for both remote control and voice connections

Human Interface Device Profile (HID) defines usage of wireless keyboards, pointing devices, gaming devices, and remote monitoring devices.

Common ISDN Access Profile defines how applications access ISDN over Bluetooth.

Personal Area Networking Profile (PAN) defines IP-based personal networking. PAN also provides support for network access points (e.g., LAN or GSM).

SIM Access Profile (SAP) defines how to access a SIM card via a Bluetooth link.

5 Bluetooth Protocols

Protocols are needed to implement different profiles and usage models. Every profile uses at least part of the protocol stack. In order to achieve interoperability between two Bluetooth devices, they both must have the same vertical profile of the protocol stack. The Bluetooth protocol stack is depicted in Figure 9.

Bluetooth products support different sets of protocols. In order to support a certain Bluetooth profile, the mandatory features of certain protocols must be implemented.

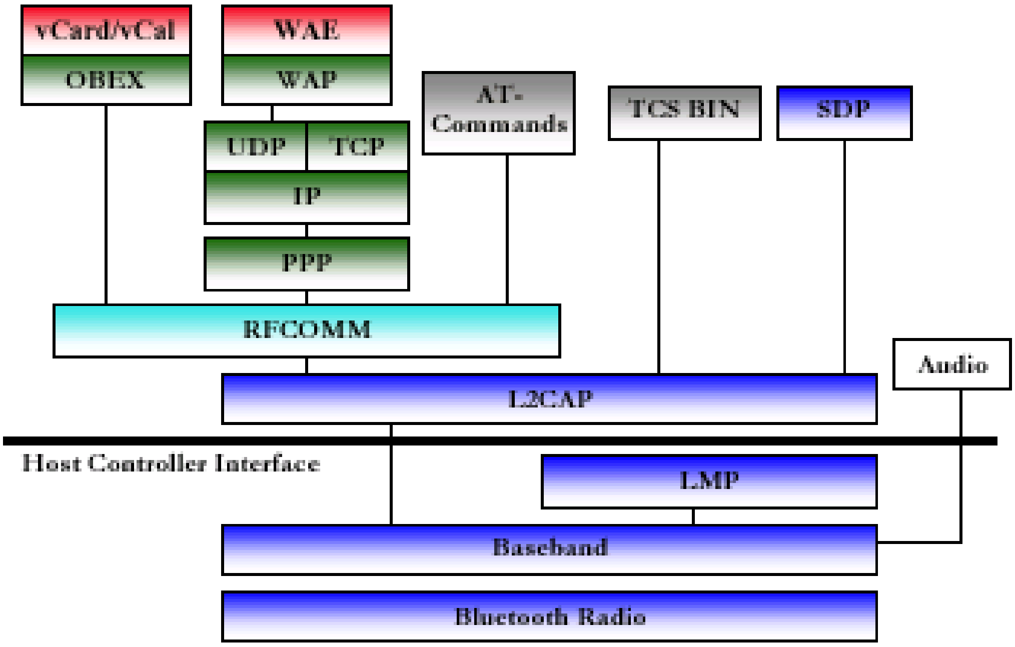

Figure 9. Bluetooth v1.1 protocol stack [BISG3, p.6]

5.1 Bluetooth Core Protocols

Baseband and **Link Control** together enable a physical RF link between Bluetooth units forming a piconet. This layer is responsible for synchronizing the transmission-hopping frequency and clocks of different Bluetooth devices [Whitepaper1, p.8].

Audio is routed directly to and from Baseband. Any two Bluetooth devices supporting audio can send and receive audio data between each other just by opening an audio link [Protocols, p.8].

Link Manager Protocol (LMP) is responsible for link set-up (authentication and encryption, control, and negotiation of baseband packets) between Bluetooth devices and for power modes and connection states of a Bluetooth unit [Protocols, p.8].

Logical Link Control and Adaptation Protocol (L2CAP) takes care of multiplexing, reassembly, and segmentation of packets [Protocols, p.8].

Service Discovery Protocol (SDP) is needed when requesting device information, services, and the characteristics of other devices. Devices have to support the same service in order to establish a connection with each other.

5.2 Cable Replacement Protocol

RFCOMM emulates RS-232 signals and can thus be used in applications that were formerly implemented with a serial cable (e.g., a connection between a laptop computer and a mobile phone).

5.3 Telephony Protocol

Telephony Control Protocol – Binary (TCS-BIN) defines the call control signaling for the establishment of speech and data call between Bluetooth devices [Protocols, p.9]. AT commands provide means for controlling a mobile phone or a modem.

5.4 Adopted Protocols

OBEX (Object Exchange) is adopted from IrDA. It is a session protocol that provides means for simple and spontaneous object and data transfer. It is independent of the transport mechanism and transport Application Programming Interface (API) [Whitepaper1, p.9].

TCP/UDP/IP is defined to operate in Bluetooth units allowing them to communicate with other units connected, for instance, to the Internet. The TCP/IP/PPP protocol configuration is used for all Internet Bridge usage scenarios in Bluetooth 1.0 and for OBEX in future versions. The UDP/IP/PPP configuration is available as transport for WAP [Whitepaper1, p.9].

PPP in the Bluetooth technology is designed to run over RFCOMM to accomplish point-to-point connections. PPP is a packet-oriented protocol and must therefore use its serial mechanisms to convert the packet data stream into a serial data stream [Whitepaper1, p.9].

The **Wireless Application Protocol (WAP)** stack can reside on top of RFCOMM (based on LAN Access Profile) or on top of L2CAP (based on PAN Profile). The latter reduces overhead and is likely to become the preferred solution for WAP over Bluetooth. In Figure 9, a LAN Access Profile-based implementation is depicted. Wireless Application Environment (WAE) hosts the WAP browser environment.

6 Example: Dial-Up Networking Profile

Let's take a closer look at the Dial-Up Networking Profile (DUN). As seen in Figure 8, it is inside the Serial Port Profile and therefore partly reuses the capabilities of the Serial Port Profile. For the DUN Profile, there are two device configurations (roles):

- Gateway (GW) is the device that provides access to the public network (typically mobile phones and modems)
- Data terminal (DT) is the device that uses the dial-up services of the gateway (typically PCs)

The protocol stack of the DUN Profile is depicted in Figure 10. The DUN Profile needs a two-piece protocol stack and an SDP branch. PPP over RFCOMM is needed for transferring payload data. AT commands are delivered over RFCOMM to control the modem (mobile phone).

The application on top of the stack is either a driver application on a PC (data terminal) or the modem emulation on a phone (gateway).

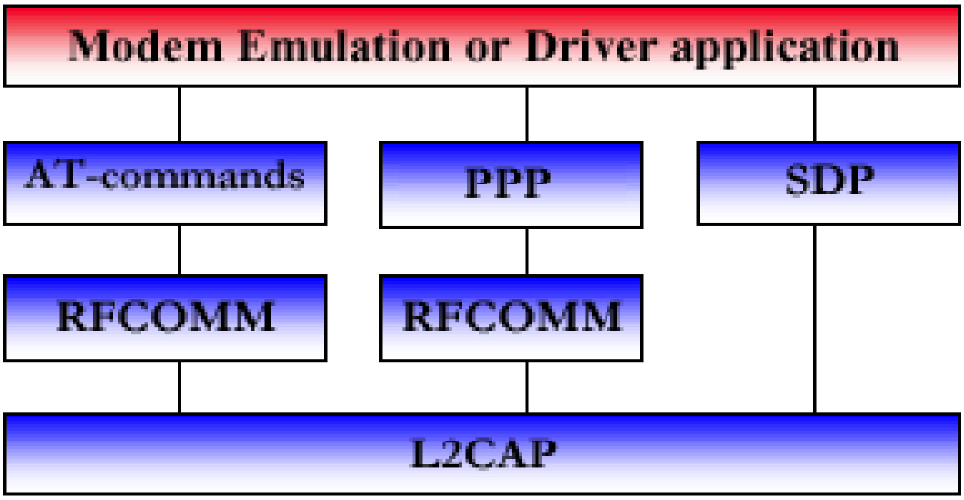

Figure 10. Dial-up networking profile protocol stack

7 Bluetooth Security

7.1 Security Methods

Authentication ensures the identity of Bluetooth devices. It authenticates the device at the other end of the link [Security, p.14]. Authentication is accomplished by using a stored link key or by pairing (entering a PIN).

Pairing is a procedure that authenticates two devices based on a common passkey, thereby creating a trusted relationship between those devices. An arbitrary but identical passkey must be entered on both devices. As long as both devices are paired, the pairing procedure is not required when connecting those devices again (the existing link key is used for authentication). Devices without any input method, like headsets, have fixed passkeys.

Authorization is a process of deciding if a device is allowed to have access to a specific service. User interaction may be required unless the remote device has been marked as "trusted." Usually the user can set authorization on/off to every remote device separately. Authorization always requires authentication [Security, p.14] .

Encryption protects communication against eavesdropping. For example, it ensures that nobody can listen to what a laptop transmits to a phone. The length of the encryption key can be between 8 and 128 bits.

7.2 Device Trust Levels

A **trusted device** has been previously authenticated, a link key is stored, and the device is marked as "trusted" in the security database of a device. The device can access Bluetooth services without user acceptance.

An **untrusted device** has been previously authenticated, a link key is stored, but the device is not marked as "trusted." Access to services requires acceptance of the user.

An **unknown device** means that there is no security information on this device. This is also an untrusted device.

7.3 Security Level of Services

Authorization required: Access is only granted automatically to trusted devices or untrusted devices after an authorization procedure ('Do you accept connection from remote device?'). Authentication is always required.

Authentication required: The remote device must be authenticated before connecting to the application.

Encryption required: The link must be changed to encrypted before accessing the service.

It is also possible that a service does not require any of these mechanisms. On the other hand, the application (service) might have its own user authentication mechanisms (a PIN code, for example).

8 Bluetooth Specifications and Qualification

Bluetooth SIG was founded in 1998 when several telecommunications and computing companies noticed that there was a need for a wireless technology to connect portable devices such as laptops and mobile phones. Infrared technology had its limitations, and therefore a technology based on radio links was conceived.

To avoid the chaos of incompatible proprietary solutions, Nokia, Ericsson, IBM, Intel, and Toshiba decided to create a common standard for wireless connectivity called Bluetooth. A consortium called Bluetooth SIG was established to create and publish specifications, promote the technology, and administer a qualification program to ensure interoperability. Today, with the addition of four more members in 1999 — 3COM, Agere (former Lucent), Microsoft, and Motorola — there are nine promoter member companies. Bluetooth SIG has more than 2,600 associate and adopter member companies. Widespread industry support has been achieved.

Soon after Bluetooth SIG was founded the companies realized that Bluetooth could offer a lot more than just cable replacement between wireless devices. Further usage models were created, and Bluetooth was considered to replace other planned short-range network technologies as well. However, current Bluetooth implementations mainly provide basic cable replacement functionality.

8.1 Specifications

Specification v1.0 was published in July 1999; specification v1.0B by the end of the same year. The first qualified Bluetooth products from autumn 2000 complied with specification v1.0B. Based on experiences with the first v1.0B-compliant products, the major ambiguities of the specification were corrected in the fall of 2000. The draft version was called v1.0B+CE (critical errata), and there were also some products qualified according to this release. For v1.1, all known contradictions were gathered and the specification was rewritten more clearly in some sections. In early 2001, v1.1 was finally published. The majority of current Bluetooth devices comply with specification v1.1.

Bluetooth working groups are specifying new profiles for different application areas. Twelve new profiles were published after the Bluetooth v1.1 Profiles release (see Section 4.3). Despite new application profiles, there are also plans to get the data transfer rate from the current 1Mbps up to 2 Mbps or even 5 to 10 Mbps. That would probably require major changes to the radio architecture, for example, to the modulation factor.

8.2 Guaranteeing Interoperability

Bluetooth SIG has taken actions to ensure interoperability. The **Bluetooth Qualification Program** makes sure that if a company wants to get its product Bluetooth qualified, it passes the qualification program–that is, that the manufacturer's product is tested and approved by a Bluetooth Qualified Test Facility (BQTF). Bluetooth qualification testing is based on testing that is conducted against a reference test system (Conformance testing) as well as functional testing against another operational Bluetooth product (Interoperability testing). The test specifications cover radio frequency qualification testing, protocol conformance testing, profile conformance testing, and profile interoperability testing.

Blue Units have been used as reference units. Blue Units were designated by Bluetooth SIG and supplied by Ericsson and Nokia. Manufacturers had to test their products against Blue Units in one of the BQTFs in order to get their Bluetooth device approved. Testing against Blue Units did not guarantee full compatibility against all of the Bluetooth devices, but in the beginning it was an important way to ensure some kind of interoperability.

UnPlugFests are testing events sponsored by Bluetooth SIG. These events take place three or four times a year. At UnPlugFests, different manufacturers (members of Bluetooth SIG) can test protocol functionality and profile interoperability of their products or prototypes with other manufacturers' devices. However, success in UnPlugFests does not guarantee that the device will be Bluetooth compliant.

8.3 Do Third-Party Applications Need to be Bluetooth Qualified?

Commercial Bluetooth products and Bluetooth components and subsystems that are offered for sale or distributed to a customer for resale or further distribution in a modified form or in combination with other products must be Bluetooth qualified.

Third-party software applications that take advantage of pre-qualified Bluetooth features of a product through the Application Programming Interface do not have to be qualified. Most third-party applications probably fall into this category. Applications that incorporate new protocol or profile functionality have to be Bluetooth qualified. Usually adding new Bluetooth protocols or profiles is only possible for the manufacturer of the product.

Developers creating a wireless Bluetooth accessory solution must qualify the hardware accessory in order to use the Bluetooth brand name. The possible software application on the phone that is using the accessory does not need qualification if it does not include new protocol or profile functionality.

9 Summary

In addition to presenting basic information on Bluetooth technology, this document introduces and explains terms that a Bluetooth application developer will probably encounter. The emphasis is on relevant topics for the application developer: data packets, profiles, and protocols. For further information on Bluetooth technology, developers should examine the Bluetooth v1.1 Core and Profiles specifications that can be downloaded from http://www.bluetooth.com.

For information on Bluetooth application development possibilities for Nokia phones, please refer to the Documents section at http://forum.nokia.com.

10 Terms and Abbreviations

Term or Abbreviation	Description
ACL	Asynchronous Connectionless packet
ARQ	Automatic Repeat Request scheme
Authentication	Procedure for ensuring the identity of a Bluetooth device by using a link key
Authorization	Procedure for trusting a Bluetooth device to allow it to access a Bluetooth service
CRC	Cyclic Redundancy Check
DUN	Dial-Up Networking profile
FEC	Forward Error Correction scheme
FTP	File Transfer Profile
GAP	Generic Access Profile
GOEP	Generic Object Exchange (OBEX) Profile
HCI	Host Controller Interface
ISM	Industrial, Scientific, Medical
L2CAP	Logical Link Control and Adaptation Protocol
OBEX	Object Exchange protocol, adopted from IrDA specification
OPP	Object Push Profile
Pairing	Exchanging passwords (as part of an authentication procedure for unpaired devices)
Piconet	Bluetooth network
PC	Personal Computer
PPP	Point-to-Point Protocol
RFCOMM	Protocol emulating RS-232
Scatternet	Network of multiple piconets
SCO	Synchronous Connection-Oriented packet
SDP	Service Discovery Protocol
SPP	Serial Port Profile
TCS BIN	Telephony Control Protocol – Binary
UPnP™	Universal Plug and Play™
WAE	Wireless Application Environment
WAP	Wireless Application Protocol

11 References

[Core]

Specification of the Bluetooth System, Core v1.1, Bluetooth SIG, February 2001

http://www.bluetooth.com/pdf/Bluetooth_11_Specifications_Book.pdf

[Profiles]

Specification of the Bluetooth System, Profiles v1.1, Bluetooth SIG, February 2001

http://www.bluetooth.com/pdf/Bluetooth_11_Profiles_Book.pdf

[Profiles2]

Bluetooth.org, Profile Specification, Bluetooth SIG, Inc., 2003

https://www.bluetooth.org/docman2/ViewCategory.php?group_id=53&category_id=214

[Protocols]

Bluetooth Protocol Architecture v1.0, Riku Mettälä, Bluetooth SIG, August 1999

https://www.bluetooth.org/foundry/sitecontent/document/whitepapers_presentations

[Security]

Bluetooth Security Architecture, Thomas Müller, Bluetooth SIG, July 1999

https://www.bluetooth.org/foundry/sitecontent/document/whitepapers_presentations

[Whitepaper1]

Bluetooth White Paper 1.1, AU-System, January 2000

http://www.cse.iitd.ernet.in/~csd97403/btwp.pdf

[Whitepaper2]

Comprehensive Description of the Bluetooth System v0.9p, Dan Sönnerstam, Bluetooth SIG, May 1998

http://info.nsu.ac.kr/cwb-data/data/ycra2/comprehensive_description_of_the_BT_system.pdf

Build Test Sell

Developing and marketing mobile applications with Nokia

1

Go to Forum.Nokia.com

Forum.Nokia.com provides the tools and resources you need for content and application development as well as the channels for sales to operators, enterprises, and consumers.

Forum.Nokia.com

2

Download tools and emulators

Forum.Nokia.com/tools has links to tools from Nokia and other industry leaders including Borland, Adobe, AppForge, Macromedia, Metrowerks, and Sun.

Forum.Nokia.com/tools

3

Get documents and specifications

The documents area contains useful white papers, FAQs, tutorials, and APIs for Symbian OS and Series 60 Platform, J2ME, messaging (including MMS), and other technologies. Forum.Nokia.com/devices lists detailed technical specifications for Nokia devices.

Forum.Nokia.com/documents
Forum.Nokia.com/devices

4

Test your application and get support

Forum Nokia offers free and fee-based support that provides you with direct access to Nokia engineers and equipment and connects you with other developers around the world. The Nokia OK testing program enables your application to enjoy premium placement in Nokia's sales channels.

Forum.Nokia.com/support
Forum.Nokia.com/ok

5

Market through Nokia channels

Go to Forum.Nokia.com/business to learn about all of the marketing channels open to you, including Nokia Tradepoint, an online B2B marketplace.

Forum.Nokia.com/business

6

Reach buyers around the globe

Place your applications in Nokia Tradepoint and they're available to dozens of buying organizations around the world, ranging from leading global operators and enterprises to regional operators and XSPs. Your company and applications will also be considered for the regional Nokia Software Markets as well as other global and regional opportunities, including personal introductions to operators, on-device and in-box placement, and participation in invitation-only events around the world.

Forum.Nokia.com/business