Quality of Service in Bluetooth Networking

Part I

Summary

The Quality of Service functions and procedures included in the Bluetooth 1.0 specification have been reviewed. Next issues associated with providing Quality of Service over a wireless link in general and Bluetooth in particular have been investigated. Allthough the Bluetooth 1.0 specification provides some Quality of Service support, some deficiencies have been identified. The requirements for QoS enhancements to the Bluetooth 1.0 specification and a Quality of Service Framework are presented in this document. The document is concluded with issues for future work.

Contents

1 INTRODUCTION 4

2 BLUETOOTH OVERVIEW 5

2.1 INTRODUCTION 5
2.2 USAGE SCENARIOS 7
2.3 STANDARDISATION 8
2.4 THE BLUETOOTH PROTOCOL SUITE 9
2.5 BASEBAND 10
2.6 LINK MANAGER PROTOCOL 17
2.7 HOST CONTROLLER INTERFACE 18
2.8 LOGICAL LINK CONTROL AND ADAPTATION PROTOCOL 20
2.9 BLUETOOTH NETWORK ENCAPSULATION PROTOCOL 23

3 INTRODUCTION BLUETOOTH QUALITY OF SERVICE 25

3.1 USING THE ACL LINK FOR DELAY SENSITIVE APPLICATIONS 25
3.2 RESOURCE CONTROL 26
3.3 SERVICE DIFFERENTIATION 26
3.4 BLUETOOTH CONFIGURATION 26
3.5 BLUETOOTH BITPIPE 27
3.6 WIRELESS LINK CHARACTERISTICS 27
3.7 QUALITY OF SERVICE IN PICONET AND SCATTERNETS 28
3.8 QOS GUARANTEES 28
3.9 QOS PARAMETERS 29

4 BLUETOOTH 1.0 DEFICIENCIES TO SUPPORT QUALITY OF SERVICE 31

4.1 L2CAP SEQUENTIAL TRANSMISSION RULE 31
4.2 ACL SERVICE 31
4.3 FLOW CONTROL 35
4.4 SETUP DELAYS 36
4.5 LAYER 2 FORWARDING VS. LAYER 3 ROUTING 36
4.6 TRAFFIC CONTROL 37

5 BLUETOOTH QUALITY OF SERVICE FRAMEWORK 38

5.1 REQUIREMENTS 38
5.2 GENERAL QOS FRAMEWORK AND DESIGN ALTERNATIVES 40

6 CONCLUSIONS 51

7 FUTURE WORK 52

8 ABBREVIATIONS 54

9 REFERENCES 56

1 Introduction

In this report it is investigated how Bluetooth can support applications that require Quality of Service. As a result of this investigation, enhancements to the Bluetooth 1.0 specification are defined.

First a brief overview of the Bluetooth protocol is given in Chapter 2. Next the issues to provide Quality of Service in Bluetooth are discussed in Chapter 3. The deficiencies of the Bluetooth 1.0 specification to support Quality of Service is described in Chapter 4. The Quality of Service Framework are presented in Chapter 5. Finally the conclusions are drawn in Chapter 6 and issues for future work are listed in Chapter 7.

The ideas described in this report do not imply any position w.r.t. Bluetooth standardisation.

2 Bluetooth overview

In this chapter the functions and procedures of the Bluetooth 1.0 protocol specification, which are relevant from a Quality of Service point of view, are explained. This explanation serves the understanding of the QoS enhancements of the Bluetooth 1.0 specification discussed in the remainder of this document. The reader who is familiar with the Bluetooth protocol suite may skip this chapter. Introductory articles and books to Bluetooth can also be found in [HAART1], [HAART2], [ARFWED], [MILLER].

2.1 Introduction

The Bluetooth technology has originally been developed as a wireless replacement for cables between electronic devices e.g. between a mobile phone and a Laptop computer. However the Bluetooth interface is gradually developing into a full wireless networking solution. The Bluetooth interface has a gross bitrate of 1 Mbps to support simultaneous voice and data communications. Bluetooth provides both authentication and encryption. Authentication provides the ability to verify the identity of the claimant through a challenge response algorithm. The Encryption procedure enables encryption of the data sent over the air-interface to prevent unintended eavesdropping. The Bluetooth interface supports both Point-to-Point and Point-to-Multipoint communications.

The Bluetooth architecture supports ad-hoc networking. Bluetooth does not rely on an existing infrastructure to allow wireless communications between devices. Bluetooth devices can discover each other and establish a link, optionally without user interaction. The Bluetooth architecture differs from a cellular network architecture in the sense that there are no specialized nodes, but in principle each Bluetooth device provides the same functionality. In a cellular network different nodes can be identified, where each node has different functions and capabilities (e.g. mobile phone and base station). The ad-hoc networking architecture has obvious advantages, but it is also the main reason that setup times in Bluetooth are long (order of seconds) and handovers are poorly supported.

The Bluetooth features can be summarized in a few words as a Low-cost, Low-power, Small-sized, Short-range, Robust wireless technology.

Some predict that the cost of a single chip solution, including the radio, may go down to $5. The radio design is such that a low cost single chip implementation is possible [HAART2]. The low cost makes Bluetooth suitable even to be integrated into low cost devices such as a wireless computer mouse. However the cost price will very much depend on the success of Bluetooth and the subsequent volumes in which the Bluetooth chip will be produced. To maintain the cost advantage, the QoS enhancements, discussed in this document, should have no significant increase in the cost of Bluetooth.

The Bluetooth technology is specifically targeted for devices which require a low power consumption, such as mobile phones, PDAs, Laptops, computer mouse, keyboards, etc. The Bluetooth architecture is sometimes referred to as low power architecture. In Idle mode the duty cycle is less than 1%[1]. The expected power consumption figures are listed in Table 1. There are three Low Power modes: PARK, HOLD and SNIFF. In PARK mode the duty cycle can even be lower than 1%!. The QoS enhancements should have no significant increase in the power consumption.

[1] In Idle mode the device is only active 10 ms. every scan window T = 1.28 seconds (<1% duty cycle). In Active mode the device only transmits when there is data to send. In Active mode the slave scans every master-to-slave slot to inspect the packet header. When the packet is not addressed to the slave (AM_ADDR), the slave can go to sleep until the next master-to-slave slot. The next master-to-slave slot depends on the current packet size. When only Link control information needs to be exchanged, then a NULL packet (header only) is used.

Mode	Power consumption	Duration (600 mAh battery)
Idle	0.3 mA	> 3 months
Voice (one channel)	10 mA	> 60 hours
Data (20% utilization)	6 mA	> 100 hours

Table 1 Power consumption.

The Bluetooth technology is prepared for a single chip implementation. This enables it to be integrated into small devices, such as the Chatpen and Headset depicted in Figure 1. Usually for small and simple devices the cost and power consumption of the Bluetooth chip are critical parameters.

Figure 1 Bluetooth enables Chatpen and Headset to connect to mobile phone.

Bluetooth provides a short range wireless interface, sometime referred to as a 'personal bubble'. There are three power classes defined with a maximum output power of 0 dBm, 4 dBm and 20 dBm respectively. The 0 dBm power classes provides a range of several ten's of meters, while the 20 dBm power classes has a range of 100 – 200 meters. The short range version is typically used for headphones, keyboard, etc. while the longer range version is typically used for mobile phones, PDAs, Laptops, etc. For the 20 dBm version, power control based on closed loop Received Signal Strength Indication (RSSI) is mandatory. The required receiver sensitivity is -70 dBm at 0.1 BER. Bluetooth does not require Line of Sight to operate. The power level of Bluetooth is significantly lower than the transmit power of current cellular phones.

The Bluetooth interface operates in the world-wide available and license free Industrial, Scientific and Medical (ISM) band. Regulations for this band vary, but in general aim to enable fair access to each user regardless of the used technique [HAART2]. The regulations generally require a spreading technique of the transmitted signal energy. Bluetooth uses frequency hopping to spread the signal. The Bluetooth hopping frequency is 1,600 hops/second over 79 frequencies[2]. In the ISM band interference immunity from other sources is important. The interference characteristics of other sources is difficult to predict. Other equipment operating in the ISM band are WLAN (802.11/b), micro-wave ovens, baby phones, garage door openers, etc. The Co-existence Working Group has the goal to investigate the detrimental effects of simultaneous operation of Bluetooth and other devices in the ISM band. It should be noted that the frequency hopping scheme, Forward Error Correction (FEC), re-transmissions (ARQ), short transmission unit and fast acknowledgements make Bluetooth robust against interference. Furthermore the low power level (i.e. short range), power control and frequency hopping scheme limits the interference Bluetooth causes to other systems.

[2] The 79 hops frequencies apply for the USA, Europe and most other countries. Spain and France allow only 23 frequencies.

2.2 Usage scenarios

To evaluate the most likely usage scenarios for Bluetooth, one should first consider how Bluetooth compares to other wireless technologies. In Figure 2 Bluetooth is compared to other wireless technologies w.r.t. bandwidth and coverage.

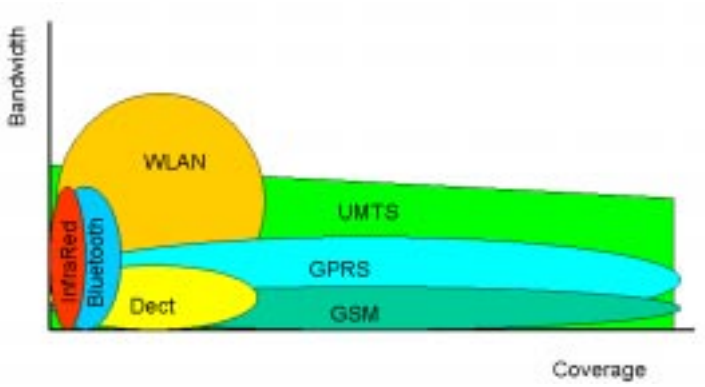

Figure 2 Bluetooth vs. other wireless technologies.

The InfraRed (IR) technology also provides a low cost wireless link, however it has a shorter range (1 - 2 m.) than Bluetooth. Furthermore IR is sensitive to direction and requires Line of Sight (LOS) and can only provide Point-to-Point communications. Bluetooth has much larger range, can go around objects and through various materials and can connect many devices at the same time.

WLAN both provides a larger range and higher bandwidth than Bluetooth. The WLAN is most likely to provide high speed access in the office environment and possibly in certain public areas and the home environment as well. The WLAN technology is currently more expensive than Bluetooth and is less power efficient.

The DECT technology provides voice and data communications. The range for DECT is larger than for Bluetooth. However the DECT technology does not very efficiently support packet-switched data due to fast circuit switching.

The cellular technologies provide more or less ubiquitous coverage, which makes them the most likely technology to provide ‘always on’ connectivity. The current GSM system provides rather low data access (9.6 kbps), but the GPRS system (< 160 kbps) and the future UMTS system (< 2 Mbps) will provide higher data rates[3].

It should be noted that for Bluetooth many usage scenarios can be defined. However taking into account the strong and weak points of competing technologies, three important usage scenarios are depicted in Figure 3. This is only a limited set of usage scenarios, i.e. those where Internet Access is the objective. The WLAN technology is most likely to find application in the office environment. Here Bluetooth provides the ability to connect different devices to the WLAN device e.g. PDA or Headset. Similarly the cellular device provides connectivity while on the move. In this scenario Bluetooth can be used to connect the Laptop to the cellular device. Finally Bluetooth can provide cordless multi-media connectivity in the home environment.

[3] It should be noted that the Radio 2 Working Group is developing a high speed radio (8 – 12 Mbps) for Bluetooth. There will also be a 2 Mbps radio, for compliance with 3G systems. See also Section 2.3.

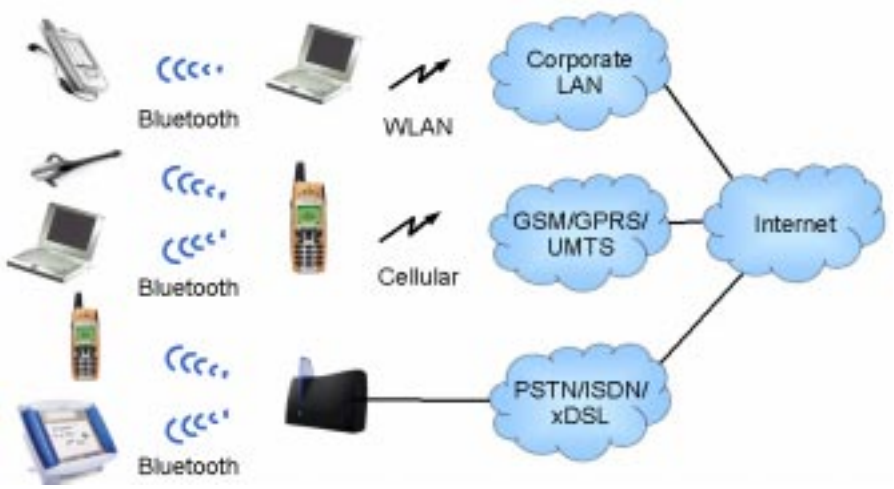

Figure 3 Bluetooth usage scenarios.

2.3 Standardisation

The Bluetooth technology has gained rapid acceptance, since its conception in 1995. The Bluetooth Special Interest Group (SIG), including Ericsson, Nokia, IBM, Intel and Toshiba, was formed in 1998 to develop an industry standard. The Bluetooth 1.0 specification became available in 1999, which contains the Bluetooth Core protocols and a set of Profiles. The Profiles guarantee interoperability between Bluetooth devices in certain usage scenarios, by specifying which options of the Core protocols should be implemented.

To extend the Bluetooth specification beyond that of a cable replacement, the SIG2 was formed. The number of Bluetooth promoters was extended from five to nine, including Motorola, 3COM, Lucent and Microsoft. The SIG2 basically aims to provide more functionality and higher speeds. There are 11 Working Groups within the SIG2, which aim to include more functionality through the specification of new Profiles[4], as listed in Table 2.

Besides Working Groups there are Expert Groups, Study Groups and Improvements Studies, which do not specify new Profiles, but come up with recommendations on specific topics. There is the Security and Automotive Expert Groups. There are the ATM (Automatic Teller Machine!), ISDN, HCI, BlueRF and QoS Study Groups. Finally there is the Voice Improvement Studies, which evaluates different voice codecs (e.g. AMR), the effect of tandem codecs, echo cancellation, error concealment and protection methods for voice communications.

The QoS Study Group aims to define QoS enhancements for the Bluetooth 1.0 specification from Baseband up to the L2CAP layer. There has been a kick-off meeting during the Bluetooth Developer Conference in December this year. There is a QoS mailinglist on which QoS issues are being discussed.

[4] The Radio2 and Co-existence Working Groups are the exception to this rule. They do not specify new Profiles.

Working group	Goals
Radio 2	Development of high speed radio (8 – 12 Mbps), which is backward compatible with 1.0 radio. There will also be a 2 Mbps radio, for compliance with 3G systems.
Personal Area Networking (PAN)	To enable peer to peer TCP/IP networking between Bluetooth devices (First phase: single Piconet scenario only. Second phase: multiple piconets). There are three subgroups: Inter-Piconet Scheduling, Access Point Roaming (including handovers), and Application Verification.
Human Interface Devices (HID)	Define how best to use Bluetooth as a USB interface to Human Interface Devices (mice, keyboard, game controllers, remote sensors, remote controls). The Cost effectiveness is critical and Bluetooth must show added value over existing low cost wireless input devices.
Audio/Video (AV)	To define low cost, low power worldwide wireless Audio/Video standard for consumer electronics. CD quality audio headphones; Still picture and video; MPEGX, MP3, etc.
Unrestricted Digital Interface (UDI)	Provide a Bluetooth profile which enables the UDI transfer service[5] of the 3G cellular phone system to external devices that are connected to the 3G cellular phone handsets through Bluetooth.
Printing	Specification to ensure that Bluetooth printers will always interoperate with client devices to some level of negotiated printing capability.
Still Image	Define the minimal requirements and generic functions necessary to enable the exchange of digital images and related data between two Bluetooth enabled digital imaging devices.
Enhanced Service Discovery Protocol (ESDP)	Develop Bluetooth profiles that provide mappings of industry Service Discovery Protocols over Bluetooth.
Local Positioning	Develop a location descriptor that provides information that can be used by other applications to determine location.
Automotive	Ensure device interoperability in the car environment by wireless connecting portable and car-embedded devices.
Co-existence	To quantify the detrimental effects of simultaneous operation of Bluetooth and other systems in ISM band. To develop methods of Bluetooth operation to improve coexistence. To evaluate coexistence issues for new Bluetooth radio designs.

Table 2 SIG2 Working Groups.

The Qualification program ensures interoperability between Bluetooth devices that comply to a certain profile. Only products that pass the Qualification program are allowed to use the Bluetooth trademark, as depicted in Figure 4.

Figure 4 Bluetooth trademark.

2.4 The Bluetooth protocol suite

In this section the Bluetooth protocol suite is briefly discussed. The Quality of Service features of the different protocol layers will be discussed in more detail in the following sections.

The Bluetooth protocol suite consists of a number of Core protocols and an Adaptation protocol, as depicted in Figure 5. On top of the Adaptation layer Standard protocols such as TCP/IP can be used. The PAN Working Group is in the process of specifying the Bluetooth Network Encapsulation Protocol (BNEP). The BNEP protocol emulates an Ethernet type of broadcast medium on top of Bluetooth, to support TCP/IP.

[5] UDI is a synchronous data transfer interface which is defined as an unrestricted digital data transfer service in the 3GPP technical specification.

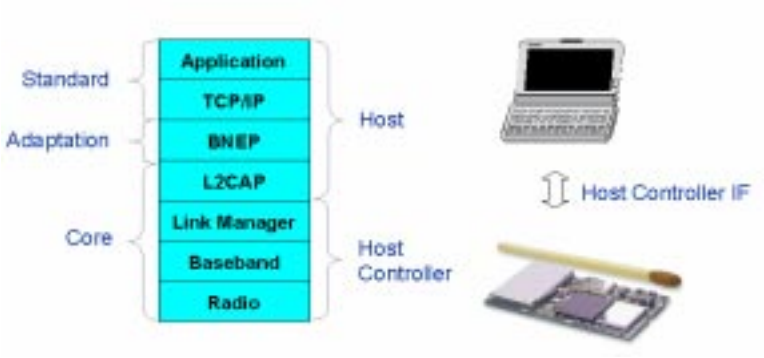

Figure 5 Bluetooth protocol suite.

The Logical Link Control and Adaptation Layer (L2CAP) provides Protocol Multiplexing, Segmentation and Re-assembly, Quality of Service and Group addressing. Voice packets can bypass the L2CAP layer, as the Baseband layer provides the ability to transmit voice packets directly.

The Link Manager provides functions to setup, maintain and release Baseband links. The Link Manager provides Link Configuration functions such as Power control, Quality driven FEC, Link Supervision, Quality of Service, Multi-slot control and Master-slave switch. Furthermore the Link Manager provides Link information (e.g. version, features, name request and timing & clock), security functions (authentication and encryption) and manages the Low Power modes (HOLD, PARK and SNIFF).

The BaseBand layer provides timing, framing, packet transmission, flow control, error detection and error correction. The entity in the Baseband, which carries out the low-level link routines, is referred to as Link Controller (LC).

The Radio layer provides modulation and demodulation of the physical layer signal. The Bluetooth radio uses Gaussian Frequency Shift Keying (GFSK). The Radio part specifies the frequency bands and the transceiver characteristics.

The Radio, Baseband and Link Manager are usually integrated into a Bluetooth module, also referred to as Host Controller. The higher layer software, i.e. L2CAP layer and higher, usually runs on a Host processor. There is a standard interface defined between the Host and Host Controller, referred to as Host Controller Interface (HCI).

2.5 Baseband

The Baseband layer provides basic transmission capabilities. Transmission on the air-interface into time slots, as depicted in Figure 6. When two Bluetooth devices establish a connection one becomes master and the other becomes the slave. The master transmits in the even numbered slots and the slave responds in the odd numbered slots, establishing a duplex link through Time Division Duplex.

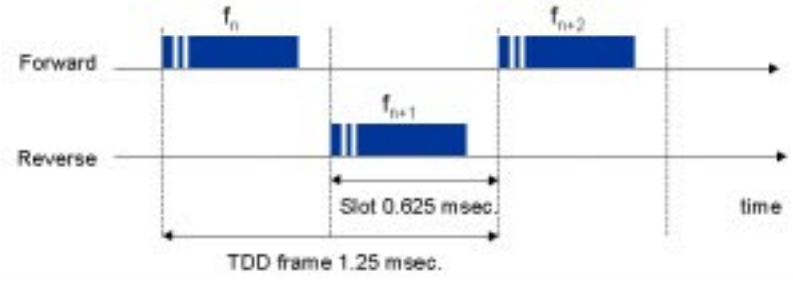

Figure 6 Time slots with Time Division Duplex.

In a time slot a single-slot packet can be transmitted. There are three packet lengths defined: single-, three- and five-slot packets, as depicted in Figure 7. Each new time slot the transmitter switches to a new frequency, except during packet transmission the frequency is maintained, also for multi-slot packets.

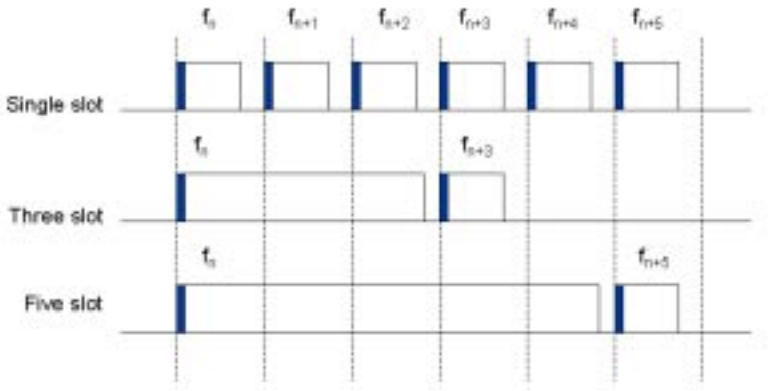

Figure 7 Single-, Three-, and Five-slot Baseband packets.

When two Bluetooth devices establish a connection one becomes master, the other becomes the slave. At that time a Piconet is established consisting of two Bluetooth devices. More Bluetooth devices can join the Piconet. A single Piconet can accommodate up to seven active slaves. Communication in a Piconet is always between the master and one of the slaves. Direct slave to slave communication is not possible. The master determines the frequency hop sequence in the Piconet. The Bluetooth address of the master determines the hop sequence and the clock of the master determines the phase of the sequence, as depicted in Figure 8. The hop sequence is unique for the Piconet.

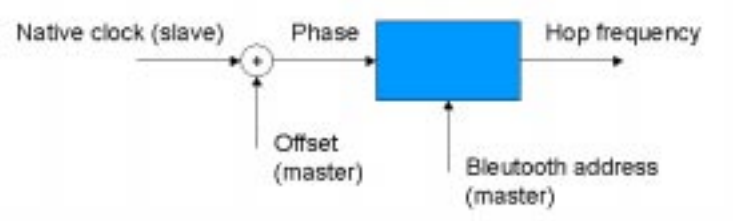

Figure 8 Frequency hop selection.

Each new slave has to synchronise to the master's clock, as depicted in Figure 9. During connection establishment the address and clock of the master are exchanged.

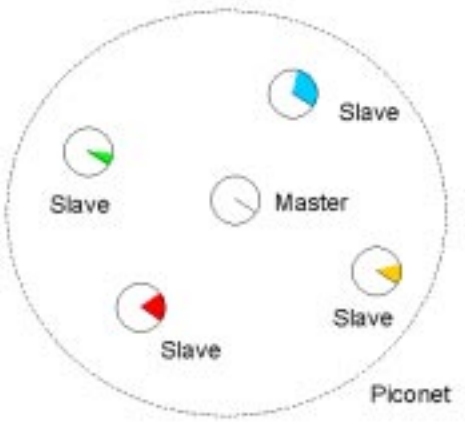

Figure 9 Master – slave synchronisation.

A slave is allowed to participate in more than one Piconet, as depicted in Figure 10. The slave participating in multiple Piconets time multiplexes between those Piconets. The Low Power mode SNIFF can support this type of multiplexing. Multiple Piconets connected together is called a Scatternet. The Piconets are not synchronised i.e. hop on a different frequency and slot timing is not aligned. When a switch to another Piconet is made, at least one TDD frame is wasted, as different Piconets are not slot synchronised.

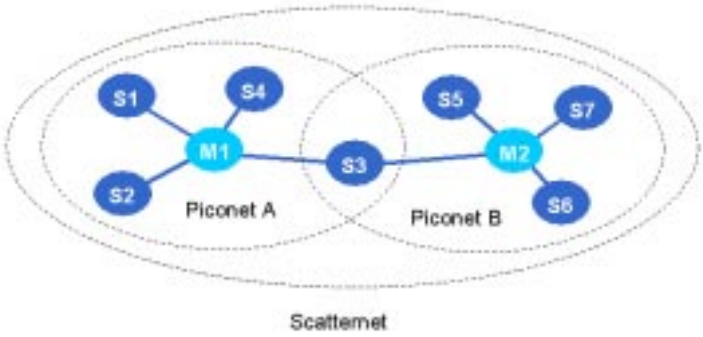

Figure 10 Piconets and Scatternets.

The transmission in a Piconet is preceded by an Access Code (AC), as depicted in Figure 11. The Access Code is derived from the master's Bluetooth address and uniquely identifies the Piconet. This allows Bluetooth devices to distinguish the transmissions of one Piconet from transmissions of another Piconet.

When a Bluetooth device joins a Piconet it is assigned a 3-bit MAC address, as depicted in Figure 11. This 3-bit MAC address is referred to as the Active Member Address (AM_ADDR). The 3-bit MAC address allows up to seven slaves and one master to be active in a single Piconet. It should be noted that each Bluetooth devices has a 48 bit Bluetooth Device Address (BD_ADDR). This address is used in the link setup procedure i.e. when the device is paged. When the master of the Piconet wants to send data to a slave in the Piconet, it uses the slave's MAC address. When a slave sends data to the master, it uses its own MAC address in the transmission.

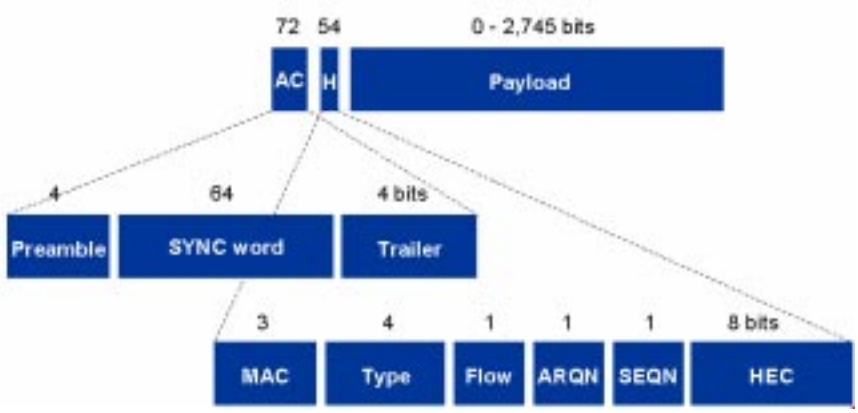

Figure 11 Packet header.

The TYPE field in the packet header identifies the type of packet that is sent. The Bluetooth packet types are listed in Table 3. The list of packet types is divided into segments. The first segment contains control packets, the second segment single-slot packets, the third segment three-slot packets and the fourth segment five-slot packets.

Segment	TYPE code	SCO Packet Type	ACL Packet Type
1	0000	NULL	NULL
1	0001	POLL	POLL
1	0010	FHS	FHS
1	0011	DM1	DM1
2	0100	-	DH1
2	0101	HV1	-
2	0110	HV2	-
2	0111	HV3	-
2	1000	DV	-
2	1001	-	DM1
3	1010	-	DM3
3	1011	-	DH3
3	1100	-	-
3	1101	-	-
4	1110	-	DM5
4	1111	-	DH5

Table 3 Packet types.

The Baseband layer supports two link types: Synchronous Connection Oriented (SCO) and Asynchronous Connectionless Link (ACL). The SCO link has characteristics typically found with circuit switched type of service, while the ACL link has more packet switched service characteristics.

On the ACL link there is Flow Control, but on the SCO link there is no Flow Control. The Flow Control is a Stop-and-Wait flow control. Flow bit zero stops the transmission of all ACL traffic, but control packets and SCO packets may still be sent.

The Baseband layer provides error control through Forward Error Correction (FEC) and re-transmissions (ARQ).

There are two types of FEC in Bluetooth: 1/3 rate and 2/3 rate FEC. With 1/3 rate FEC, each bit is transmitted three times. Decoding uses majority voting. With 2/3 rate FEC five parity bits are added to each block of ten bits. The code distance is 4 which allows the correction of a single bit error and detection of double bit errors. The header of each packet is protected by 1/3 rate FEC.

To support re-transmissions the packet header contains a SEQN bit and an ARQN bit, as depicted in Figure 11. For each new transmission the SEQN bit is toggled. The ARQN bit indicates in the reverse slot whether the transmission in the forward slot was successful (ACK) and not (NACK).

The SCO link is typically suitable to carry real-time traffic such as audio and video. The SCO link uses a slot reservation mechanism, which allows the periodic transmission of SCO data. The SCO payload is not protected by a Cyclic Redundancy Check (CRC) and there are no re-transmissions. The SCO link optionally uses Forward Error Correction (FEC) to provide robustness against bit errors. The SCO payload can be encoded with either 1/3 or 2/3 rate FEC. The bandwidth of the SCO link is negotiated by means of the period, which defines the time between consecutive SCO packets. The bandwidth is at most 64 kbps. The SCO link therefore provides a guaranteed bandwidth and delay, however it does not provide reliability.

The ACL link is typically suitable to support data applications. The ACL link is based on a polling mechanism between a master and up to seven active slaves in the Piconet. The ACL link can provide

both symmetric and asymmetric bandwidth[6]. The ACL bandwidth is determined by the ACL packet type and the frequency with which the device is polled. The ACL payload is optionally protected by 2/3 rate FEC. Each ACL payload is protected by a 16-bit CRC code, except for the AUX packet type. When the CRC check fails a re-transmission is performed, unless the time period, during which re-transmissions are allowed, has expired. The time period during which re-transmissions are allowed is denoted as the Flush Timeout setting. This time period starts when the transmission of the First segment of the L2CAP packet starts, as depicted in Figure 12. When the Flush Timeout timer expires the L2CAP packet is flushed from the Host Controller buffer and a new L2CAP packet is transmitted.

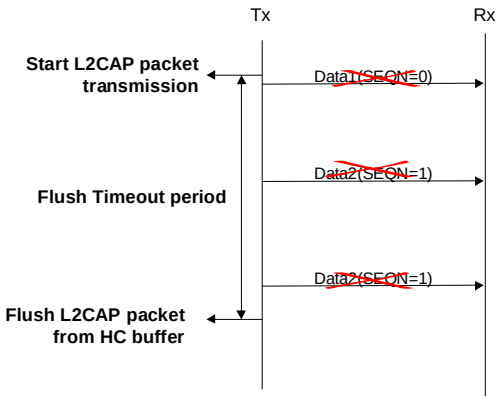

Figure 12 Flush Timeout.

The simultaneous transmission of SCO and ACL traffic is depicted in Figure 13. The SCO traffic is transmitted at periodic intervals in the slots reserved for SCO. The ACL traffic is transmitted in between the SCO transmissions. The ACL traffic may not use the reserved timeslots for SCO traffic. The ACL transmission may use single- or multi-slot packets. The SCO transmission is always single-slot. On the ACL link a polling scheme is used to control the transmissions between master and slave. A slave may only transmit in the slave-to-master slot when 'polled' in the preceding master-to-slave slot. The master polls a slave when it transmit data to the slave. When the master has no data to send to the slave, the master may send an explicit POLL packet without payload[7].

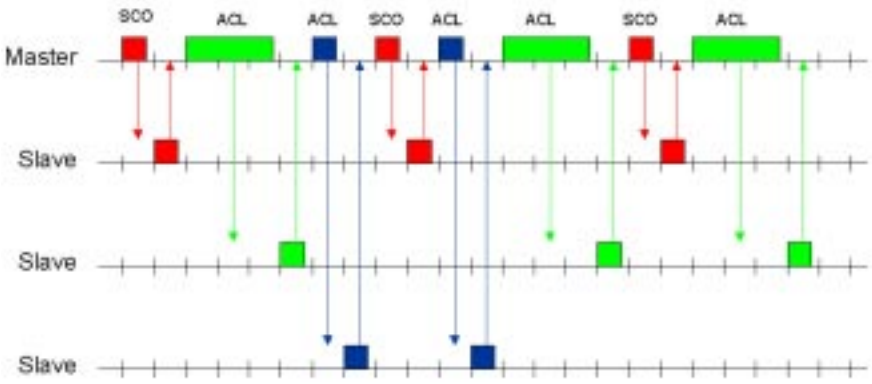

Figure 13 Simultaneous ACL and SCO transmissions.

[6] A pure asymmetric bandwidth cannot be supported, however the asymmetry ratio can be up to 1:20.
[7] A potential conflict can arise when a slave is polled in the TDD frame before the start of a TDD frame reserved for SCO traffic and the slave is allowed to use multi-slot packets. The slave is not aware of the reserved SCO TDD frame when this is an SCO link of another slave. The slave may decide to use a multi-slot packet and thus transmit in the TDD frame reserved for SCO traffic. It is the responsibility of the master to avoid this type of conflict.

The ACL transmission rates are listed in Table 4. It should be noted that the transmission rates are the maximum rates that can be obtained, assuming that the master continuously polls the slave and there are no re-transmissions on the air-interface.

Packet	Timeslots	CRC	FEC	Symmetric (kbps)	Asymmetric (kbps)	
					Forward	Reverse
DM1	1	Yes	Yes	108	108	108
DH1	1	Yes	-	172	172	172
DM3	3	Yes	Yes	258	387	54
DH3	3	Yes	-	390	585	86
DM5	5	Yes	Yes	286	477	36
DH5	5	Yes	-	433	723	57
AUX	1	-	-	185	185	185

Table 4 ACL transmission rates.

An ACL packet has a header in the payload, the SCO packets do not, as depicted in Figure 14.

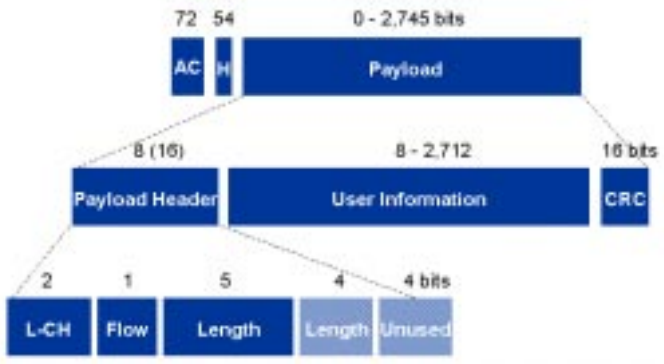

Figure 14 Payload header ACL packet.

The Payload header contains the L-CH code, which is used to separate LMP signaling from normal ACL traffic and to allow L2CAP re-assembly. The L-CH coding is listed in Table 5.

L_CH code	Information
00	Undefined
01	Continuation segment of an L2CAP packet
10	First segment of an L2CAP packet
11	LMP message

Table 5 L_CH code in ACL payload header.

The Flow Control bit controls the flow of traffic on the User Asynchronous (UA) and User Isochronous (UI) Logical Channels. When the Flow Control bit is one, the sender should stop the transmission of ACL packets before an additional amount of payload data is sent. The additional amount, denoted as flow control lag, should not exceed 1,792 bytes. The flow control lag can further be reduced by means of the LMP_features_res message.

The Length field indicates the length of the ACL payload. The payload is protected by a 16-bit CRC check. The Payload header is one byte for single slot and two bytes for multi-slot packets.

When the device is in ACTIVE mode it listens to the traffic in the Piconet. To save power the device can be put in a low power mode. There are three low power modes: PARK, SNIFF and HOLD.

When the device is in PARK mode it remains synchronised to the master, but does not participate in the traffic in the Piconet. When a device is parked it gives up its Active Member Address (AM_ADDR). The master assigns an 8-bit Parked Member Address (PM_ADDR) to the parked slave. The parked device periodically listens to the Beacon channel transmitted by the master of the Piconet, as depicted in Figure 16. This allows the device to re-synchronise and listen for broadcast messages.

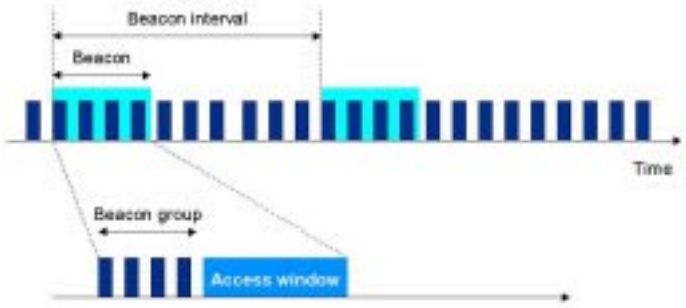

Figure 16 PARK mode.

By means of LMP signaling the broadcast messages are transmitted and devices are parked and unparked. The Beacon channel structure is signalled to the slave when it is parked. The Beacon channel consists of one ore more Beacon slots, denoted by N_B, which are transmitted at a regular interval, the Beacon interval, denoted by T_B. In addition to the Beacon slots, an Access Window is defined, where a parked slave can request to be unparked. The Access Window may support different forms of access techniques such as polling or random access, but at this stage only the polling scheme has been defined.

The SNIFF mode allows a device to be periodically present in different piconets i.e. the SNIFF mode enables Scatternets.

It should be noted that when a device switches between Piconets, at least one frame is lost, because different Piconets are not slot synchronised. Guard space in the scheduling algorithm has to allow for slot misalignment of the different Piconets. When a device jumps to another Piconet, then the channel parameters such as hop sequence and clock need to be adjusted. Note that a device cannot be master in multiple Piconets. But it can be master in one, and slave(s) in other Piconets.

The LMP protocol is used to negotiate SNIFF parameters. The request to enter SNIFF mode can be initiated from either sides, however the master may force a slave into SNIFF mode. The parameters involved with SNIFF are: SNIFF interval T_{SNIFF}, SNIFF attempt $N_{SNIFF\ attempt}$, SNIFF timeout $N_{SNIFFtimeout}$ and SNIFF offset D_{offset}, as depicted in Figure 17.

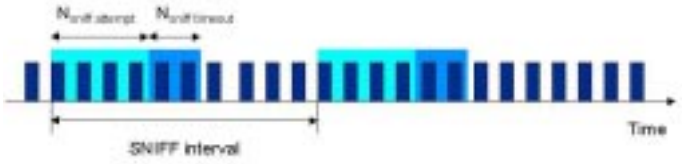

Figure 17 SNIFF mode.

The SNIFF offset D_{offset} specifies the beginning of the first SNIFF slot, i.e. specifies the absolute beginning of the SNIFF mode. The SNIFF interval T_{SNIFF} specifies the time between the first slots of consecutive SNIFF periods. The SNIFF attempt parameter determines for how many slots the slave must listen for incoming traffic. The slave sniffs during the Packet header for Access Code and Active Member Address. When there is a packet (data, POLL or NULL), the packet is received and the slave is allowed to transmit a packet in the reverse slot. The SNIFF timeout parameter determines for how many additional slots the slave must listen, as long as it receives packets with its own Active Member address. When the ACL link is in SNIFF mode, the master can only transmit to the slave in the 'SNIFF slots'.

The HOLD mode enables a slave to negotiate a period of time during which the transmission between master and slave is postponed. The slave transceiver can be turned off during the hold period to save power. The HOLD mode can be used to save power, but can also be used to discover other Bluetooth devices and/or other Piconets. In HOLD mode the slave retains its Active Member address.

2.6 Link Manager Protocol

The Link Manager Protocol (LMP) provides basic functions for ACL/SCO link setup/release, configuration and information. Furthermore LMP provides security functions and management of Low Power modes. The Link Manager Protocol (LMP) signaling messages shall not be delayed by L2CAP packets, i.e. LMP signaling has priority over L2CAP data [BLUESPEC]. However LMP signaling messages can be delayed by re-transmission on the Baseband layer.

An HCI QoS_Setup command triggers the Link Manager in the local device to send an LMP_quality_of_service_req (local device = slave) or LMP_quality_of_service (local device = master) to the remote side, as depicted in Figure 18. The LMP Quality of Service message contains a maximum time between consecutive polls between master and slave and the number of repetitions of broadcast packets. A slave has to accept the LMP Quality of Service request from the master. The master may accept or reject an LMP Quality of Service request from a slave.

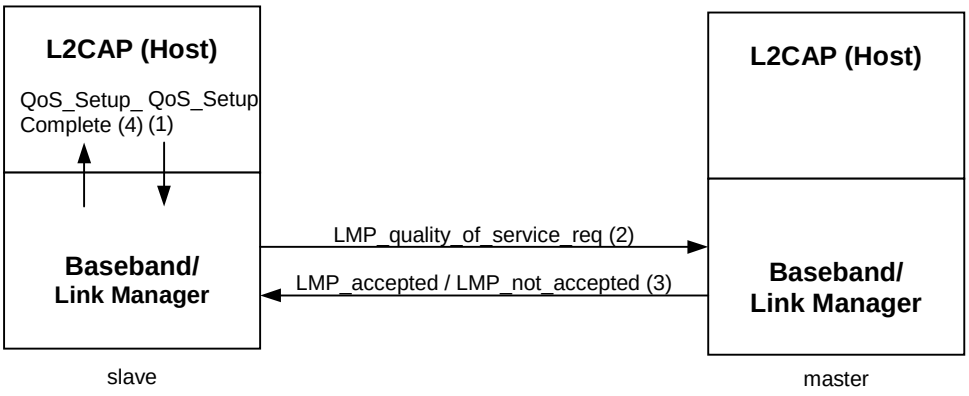

Figure 18 Link Manager Quality of Service signaling.

The set of Baseband packet types supported by the local Link Manager can be exchanged with the remote Link Manager with the LMP Supported Features procedure, as depicted in Figure 19. The remote device responds with the set of Baseband packet types it support in the LMP_features_res. The Link Manager shall use the intersection of the packet types supported by both sides.

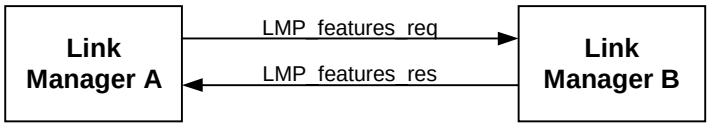

Figure 19 LMP Supported Features procedure.

The Link Manager is allowed to use single-slot packets by default. A slave is only allowed to use multi-slot packets when the master of the Piconet has allowed this through LMP signaling. A slave requests multi-slot usage by LMP_max_slot_req. The master responds with an LMP_accepted or LMP_not_accepted. The master may notify a slave of multi-slot usage by LMP_max_slot.

Within the limitations imposed by the remote Link Manager, Multi-Slot Control and the packet types specified by the HCI Create_Connection and HCI Change_Connection_Packet_Type the Link Manager is free to select the packet type for transmission. This implies that the Link Manager may wait for more data to arrive to fill e.g. a five-slot packet or to send the data immediately in a single-slot packet. At the transmitting side the Link Manager may fill a Baseband packet with data received in multiple HCI Data Packets, as long as the L2CAP packet boundaries are preserved i.e. an L2CAP First segment has to be transmitted in a new Baseband packet. At the receiving side the Link Manager may fill an HCI Data Packet with data received in multiple Baseband packets of the same ACL link as long as the L2CAP packet boundaries are preserved i.e. an L2CAP First segment has to be transferred in a new HCI Data Packet.

2.7 Host Controller Interface

The Host Controller Interface (HCI) is the standardised interface between L2CAP and the Baseband layer, as depicted in Figure 5. In the following text the L2CAP layer implementation is denoted as Host. The HCI interface enables the Host to send Commands to the Baseband, and the Baseband to send Events to the Host. Furthermore Data Packets can be transferred between Host and Baseband, as depicted in Figure 20.

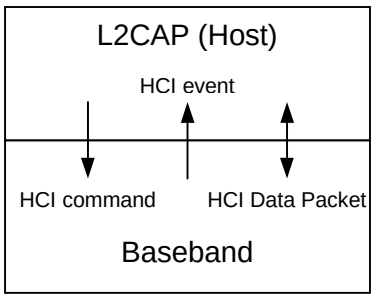

Figure 20 HCI Command, Events and Data Packets.

2.7.1 Baseband packet types

With the Create_Connection command the underlying Link Manager is instructed to setup an ACL link to a remote Bluetooth device. The Create_Connection command contains among other a Packet_Type field, which defines a set of Baseband packet types the Link Manager shall use for transmission on the ACL link. Each ACL packet format can be specified individually: DM1, DH1, DM3, DH3, DM5 and DH5. A Create_Connection command at the local device causes an Connection_Request event at the remote side, which contains the BD_ADDR of the initiating side, the Class_of_Device of the initiating side and the Link_Type (SCO or ACL). The Host shall not specify packet types the local device cannot support.

This implies that an Read_Local_Supported_Features command should be issued before an Create_Connection command is given.

With the Read_Local_Supported_Features command the Host can learn about the Baseband packet types the local Link Manager can support. In the response it is indicated whether the Baseband supports three-slot and/or five-slot packets. Single-slot packets are supported by default.

The Host can learn about the packet types the remote Link Manager can support via the Read_Remote_Supported_Features command. This command triggers the LMP Supported Features procedure, explained in Section 2.6, when this information is not already known in the local device.

The Change_Connection_Packet_Type command allows the Host to change dynamically the set of Baseband packets the Link Manager shall use on the ACL link. This Command shall not specify packet types the local device cannot support. Each Baseband packet can be specified individually, as with the Create_Connection command. The Connection_Packet_Type_Changed event notifies the Host when the set of packets the Link Manager shall use has changed.

2.7.2 HCI Data Packets

The data between L2CAP and the Baseband layer is transferred in HCI Data Packets, as depicted in Figure 22. In the Baseband layer there is a Host Controller buffer, where HCI Data Packets are buffered for transmission. There is a mandatory flow control mechanism from L2CAP to the Baseband to avoid overflow in the Host Controller buffer. There is an optional flow control mechanism from Baseband to the L2CAP layer, which prevents overflow of the L2CAP buffers.

Before the L2CAP layer transfers any data to the Baseband it should issue an Read_Buffer_Size command. This command returns separately for ACL and SCO links the maximum allowed size of Data Packets and the maximum number of Data Packets that can be stored in the Host Controller buffer. L2CAP shall not offer Data Packets larger than this maximum size. The minimum HCI Data Packet size that should be supported is 255 bytes payload. With the Number_Of_Completed_Packets event the Baseband layer informs L2CAP about the number of Data Packets that have completed transmission i.e. either successfully or being flushed. The Baseband layer should inform L2CAP periodically as long as there are packets in the Host Controller buffer. The frequency with which L2CAP is informed is implementation dependent. The unlikely event of Host Controller buffer overflow is indicated to L2CAP with an Data_Buffer_Overflow event.

The flow control mechanism from Host Controller to L2CAP is optional. With the Set_Host_Controller_To_Host_Flow_Control command flow control is switched on and off. Flow control is performed in a similar way as Baseband to L2CAP flow control is performed: With Host_Buffer_Size command the Baseband learns about the maximum size of Data Packets L2CAP can support and the size of the L2CAP receive buffer. With the Host_Number_Of_Completed_Packets event L2CAP informs the Baseband about the occupation of the L2CAP receive buffer.

An HCI Data Packet does not have to match a Baseband packet format. When an HCI Data Packet is larger than the Baseband format, the HCI Data Packet is transmitted in multiple Baseband packets. The Link Manager performs the segmentation of HCI Data Packet into Baseband packets (transmitting side) and re-assembly of Baseband packets into HCI Data Packets (receiving side). The maximum payload length of an HCI Data Packet is 65,536 bytes (2 bytes length field). The HCI interface has to preserve the L2CAP packet boundaries i.e. an HCI Data Packet contains at most one L2CAP First segment and if so this First segment starts at the beginning of the HCI Payload.

2.7.3 Flush Timeout

The Flush Timeout specifies the maximum time period during which the transmitting Baseband layer performs re-transmission attempts before the L2CAP packet is discarded i.e. flushed. It should be noted

that the Flush Timeout is set per ACL link. When there is a Flush Timeout all HCI Data Packets, belonging to the L2CAP packet which failed transmission, are deleted from the Host Controller buffer. When the First segment of the next L2CAP packet is already stored in the Host Controller buffer, then the First segment is scheduled for transmission. Otherwise all Continuation segments stored in the Host Controller buffer after the Flush Timeout are flushed from the Host Controller buffer, until there is a First segment.

There is a HCI command to flush the whole content of the Host Controller buffer. The Flush Timeout setting can be set with the Write_Automatic_Flush_Timeout command and read with the Read_Automatic_Flush_Timeout command. When a Flush of the Host Controller buffer has occurred the L2CAP layer can be notified with a Flush_Occurred event.

2.7.4 QoS Setup

The Service type, Traffic and QoS parameters, negotiated with the L2CAP Configuration Parameters (See Table 7), can be made available to the underlying Baseband layer with the QoS_Setup command. It should be noted that the Token Bucket Size is not included in the QoS_Setup command.

When the local device is a slave, the QoS_Setup command triggers an LMP_quality_of_service_req to the master to request the required QoS (See Section 2.6). It should be noted that the parameters included in the LMP_quality_of_service_req are limited to the maximum time between consecutive polls by the master and the number of broadcast repetitions. When the local device is the master, the QoS_Setup command triggers an LMP_quality_of_service indication to the remote slave (See Section 2.6). The response of the Baseband layer to a QoS_Setup command is given by a QoS_Setup_Complete event. In the status of this event it is indicated whether the QoS setup has been successful or not. It includes the same parameters as the original QoS_Setup command. The Baseband layer may indicate a successful QoS setup with modified QoS parameter values. The parameter values indicate the QoS level the Link Manager has accepted. The Link Manager in the Baseband layer can also indicate that the current QoS requirements cannot be met with a QoS_Violation event.

2.8 Logical Link Control and Adaptation Protocol

2.8.1 Introduction

The Logical Link Control and Adaptation Protocol (L2CAP) adapts higher layer protocols to the Bluetooth Baseband layer. L2CAP provides a basic datalink layer on top of the Bluetooth Baseband layer. The L2CAP provides the ability to transmit SDUs up to 65,535 bytes. An L2CAP channel is locally identified by a Channel Identifier (CID). A separate CID, i.e. L2CAP channel, is defined to carry L2CAP signaling. There is only one L2CAP signaling channel between two Bluetooth devices. There can be multiple L2CAP channels between two Bluetooth devices carrying user data. All L2CAP channels between two Bluetooth devices make use of the same underlying ACL link.

L2CAP provides both a Connection-Oriented and a Connectionless service. For the Connectionless L2CAP channel no Quality of Service is defined and data is sent to the members of the group in a Best Effort manner. The Connectionless L2CAP channel is unreliable i.e. there is no guarantee that each member of the group receives the L2CAP packets correctly. For the Connection-Oriented channel Quality of Service is defined and the reliability of the underlying Baseband layer is used to provide reliability.

2.8.2 Segmentation and Re-assembly

The L2CAP packet format is depicted in Figure 21. The Length field specifies the size of the payload

(bytes). The maximum size of the L2CAP payload is denoted by the Maximum Transmission Unit[8] (MTU). The minimum MTU for L2CAP signaling packets is 48 bytes. The default MTU is 672 bytes. The minimum MTU for Connection-Oriented packet is negotiated during channel Configuration (See Section 2.8.3). There is a separate MTU for each direction.

Figure 21 L2CAP packet format.

An L2CAP packet is segmented into one or more HCI Data Packets, as depicted in Figure 22. The first HCI Data Packet contains the L2CAP Length and Channel ID fields. The number of HCI Data Packets an L2CAP is segmented into, depends on the implementation. However the underlying Baseband layer specifies a limit on the size of the HCI Data Packet (See Section 2.7.2). Each HCI Data Packet contains an indication whether it is a 'First' segment or a 'Continuation' segment in the Packet Boundary Flag. The Connection handle identifies the underlying ACL link (e.g. SCO or ACL link).

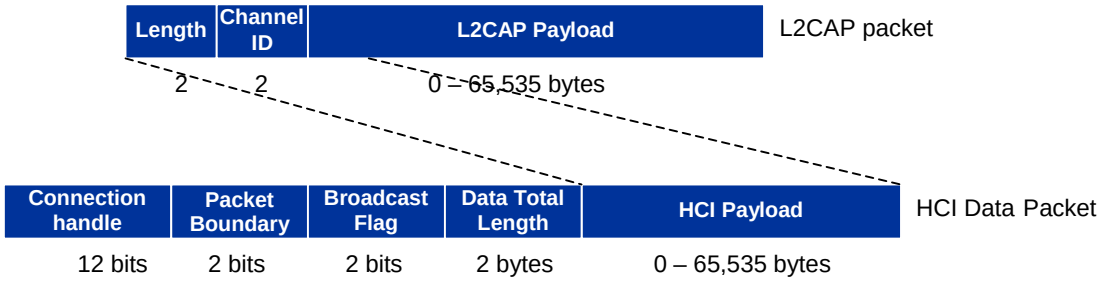

Figure 22 L2CAP packet segmentation.

The L2CAP transmission rules specify that an L2CAP packet transmission must be completed before a new L2CAP transmission is started over the same ACL link. This implies that L2CAP segments may not be transmitted 'interleaved' over the HCI interface. The reason for this requirement is that the 're-assembly' information is limited to 'First' or 'Continuation'. The underlying ACL link maintains the ordering of the segments, as depicted in Figure 23.

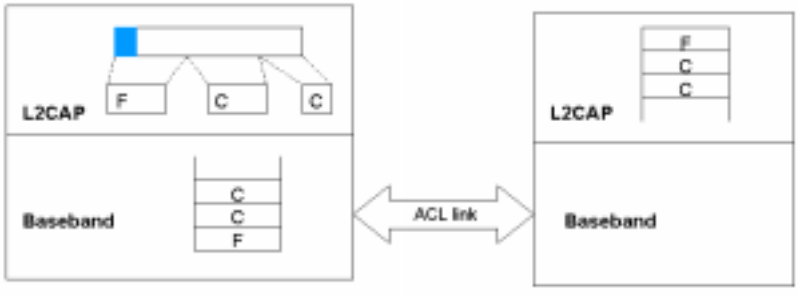

Figure 23 Sequential transmission rule of L2CAP packets.

The maximum HCI Payload is 65,536 bytes. Thus a complete L2CAP packet can be transferred in an HCI Data Packet to the Link Manager in the Baseband layer. The Packet Boundary Flag is listed in Table 6.

[8] It should be noted that the L2CAP MTU is not the same as the IP MTU. The IP header is included in the payload of the L2CAP packet. And if there is a layer inbetween IP and L2CAP e.g. BNEP, then the BNEP header is also encapsulated in the L2CAP payload.

Value	Parameter Description
00	Reserved for future use
01	Continuation segment of an L2CAP packet
10	First segment of an L2CAP packet
11	Reserved for future use

Table 6 Packet_Boundary_Flag field of HCI Data Packet [BLUESPEC].

The HCI Data Packets are transferred over the HCI interface to the Baseband layer. The HCI Data Packet format does not have to match the packet format used on the Baseband layer i.e. an HCI Data Packet may have to be further segmented into multiple Baseband packets for transmission. L2CAP uses the reliability provided by the underlying Baseband ACL link. For this purpose the ACL link carries out re-transmissions. The unit of re-transmission is the packet format used on the Baseband layer. Furthermore the Baseband layer preserves the ordering of the stream of segments offered by the L2CAP layer.

The re-assembly of L2CAP packets makes use of the First and Continuation information carried in the HCI Data Packets, assuming that the Baseband layer preserves the ordering of the segments. The length field of the L2CAP packet is used as a consistency check for re-assembly. It should be noted that the HCI Data Packet at the sender does not necessarily coincide with the same HCI Data Packet at the receiver. The receiver may decide to transfer the received data in smaller or larger HCI Data Packets to the higher layer[9].

2.8.3 Configuration Parameters

First an ACL link is established by means of LMP signaling, as depicted in Figure 24. Next an L2CAP channel is established by means of the L2CAP_ConnectReq. Once the L2CAP channel has been established, it can be configured by means of the L2CAP_ConfigReq. Configuration of the L2CAP channel is optional.

There are three L2CAP Configuration Parameters: MTU, Flush Timeout and Quality of Service. The Configuration Parameter must be negotiated for each direction separately.

The MTU in the Configuration Request specifies the maximum size of the L2CAP payload the local device can support. The remote device can respond with a lower MTU in the Configuration Response when it is not able to support the MTU in the request.

[9] The Link Manager at the receiving side constructs HCI Data Packets based on the received Baseband packets. Each Baseband packet contains an L-CH code in the Payload Header which is used to construct the HCI Data Packets. Strictly speaking the construction of HCI Data Packets is not similar to re-assembly as the original HCI Data Packet may not be reconstructed.

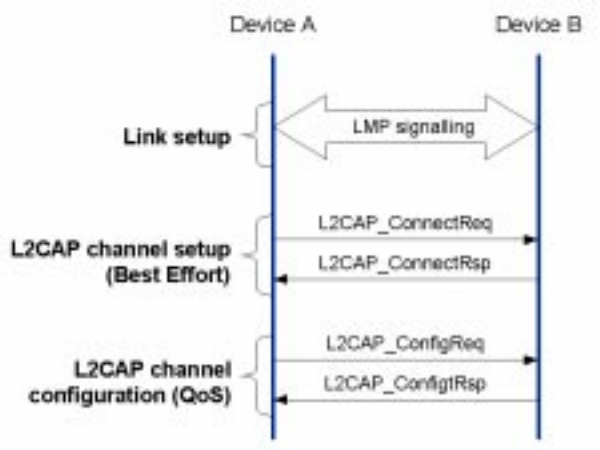

Figure 24 L2CAP Channel Configuration.

The Flush Timeout specifies the maximum time period during which the transmitting Baseband layer performs re-transmission attempts before the L2CAP packet is discarded i.e. flushed. The Flush Timeout period starts with the transmission of the first Baseband packet of the L2CAP packet. The Flush Timeout is specified in milliseconds. A Flush Timeout of 1 represents no re-transmission attempt and a Flush Timeout of all one's represents infinite re-transmission attempts (i.e. until the ACL link is lost). The infinite Flush Timeout setting is also referred to as 'reliable link'.

The Quality of Service (QoS) option allows the negotiation of Service type, Traffic and QoS parameters. The Service type, Traffic and QoS parameters refer to the traffic flow originating from the local device. The QoS parameters, that can be negotiated, are listed in Table 7.

Configuration Parameter	**Unit**	**Parameter Description**
Flags	-	Reserved for future use
Service Type	No traffic, Best Effort (default), Guaranteed.	The Service Type identifies the service level. Default Service Type is Best Effort.
Token Rate	Bytes/second	Represent average traffic load
Token Bucket Size	Bytes	Represents the maximum burst size
Peak Bandwidth	Bytes/second	Represents the maximum transmission rate of the source
Latency	Microseconds	Maximum delay between packet generation and start of packet transmission on the air-interface. The precise interpretation depends on the Service Type.
Delay Variation	Microseconds	Difference between maximum and minimum delay. Can be used to determine the buffer size at the receiver.

Table 7 QoS Configuration Parameter [BLUESPEC].

2.9 Bluetooth Network Encapsulation Protocol

The Personal Area Networking (PAN) Profile specifies how the IP protocol can be used on top of Bluetooth. The PAN tries to emulate an Ethernet broadcast medium over Bluetooth, as depicted in Figure 25. The Bluetooth Network Encapsulation Protocol (BNEP) is the protocol used to encapsulate IP packets. It should be noted that the current PAN version is limited to a single Piconet.

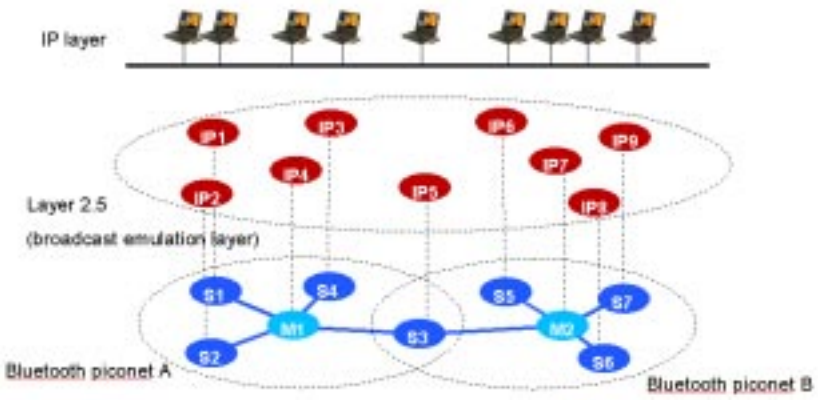

Figure 25 Bluetooth Network Encapsulation Protocol (BNEP).

In Figure 26 the protocol encapsulation and segmentation procedures in the different layers is depicted. First an IP packet is encapsulated into a BNEP packet. The BNEP packet is segmented into HCI Data Packets in the L2CAP layer. The HCI Data Packets are transferred over the HCI interface to the Host Controller buffer in the Baseband. The Link Manager then selects the Baseband packet type for transmission over the air-interface.

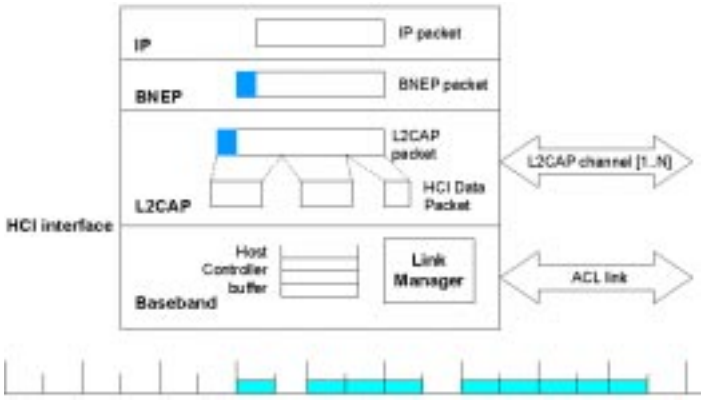

Figure 26 Packet transmission through different layers.

3 Introduction Bluetooth Quality of Service

3.1 Using the ACL link for delay sensitive applications

The Bluetooth protocol suite defines two link layer types: Synchronous Connection Oriented (SCO) and Asynchronous Connectionless Link (ACL). The SCO link provides a circuit switched type of service, while the ACL link provides a packet switched type of service, as explained in Section 2.5.

The ACL link can be configured to provide Quality of Service by means of the HCI QoS Setup command. Through this command the Traffic and Quality of Service requirements are specified together with the required service. Currently only the Guaranteed service is defined, similar to the Guaranteed defined within the Integrated Service architecture. However current Baseband implementations do not support this type of service. New polling algorithms need to be defined that support this type of service. Research into new polling algorithms that support Quality of Service is going on, however on short term it is unlikely that a polling algorithm is developed that supports the Guaranteed service.

Thus the Baseband [BLUESPEC] supports two types of concurrent services, SCO and ACL service, as depicted in Figure 27. A separate Connection Handle is associated with the SCO and ACL service. The ACL service is by default Best Effort, but can be configured to provide a Guaranteed service, as depicted in Figure 27. The Link Manager Protocol (LMP) provides the ability to setup, maintain and release Baseband links. The LMP signaling messages have priority over ACL traffic, however the LMP messages can be delayed by ACL re-transmissions.

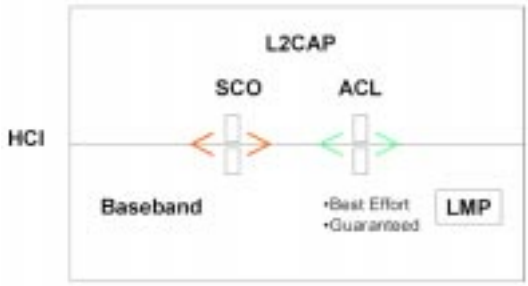

Figure 27 Bluetooth 1.0 Baseband service support [BLUESPEC].

Although the SCO link is designed to support real-time applications, it also has some deficiencies to support these type of applications. The SCO link can only provide a fixed symmetric bandwidth. Certain delay sensitive applications, e.g. streaming voice and video, however may require a variable and asymmetric bandwidth. Furthermore the FEC error detecting and error correcting capabilities are limited to more or less single bit errors. In the case of an external interferer, such as a WLAN or micro-wave oven, bit errors usually occur in bursts. FEC encoding in such a case is ineffective. Thus the voice and video quality can be significantly impaired in case of interference. Another important issue with respect to burst errors is the inability of the receiver to detect corrupted payload. The quality may be significantly reduced when audio/video frames containing multiple bit errors are fed into a decoder. Preferably a muting or error concealment algorithm is applied to improve the quality when there are multiple bit errors.

The ACL can provide reliability in the presence of interference, also when the bit errors occur in bursts. The delay involved with re-transmissions on the ACL link are small i.e. an acknowledgement can be received within 1.25 mseconds. This opens the possibility to perform re-transmissions for delay sensitive applications such as interactive real-time and streaming audio/video applications. The re-transmission period can be configured with the Flush Timeout setting, which prevents re-transmissions when it is no longer useful. Thus the ACL link can improve the quality in the presence of interference compared to the SCO link. Furthermore the ACL link can support variable and asymmetric bandwidth required by certain

applications. These are the main advantages of using the ACL link instead of the SCO link for real real-time applications. However the default Best Effort service provided by the ACL link does not provide the delay and bandwidth requirements that in most cases Conversational and Streaming applications require. The Guaranteed service does provide the delay and bandwidth requirements, however as stated before, is difficult to implement at this point in time. Thus to support Conversational and Streaming applications over the ACL link new ACL services need to be defined.

3.2 Resource control

Between any two Bluetooth devices there is at most one ACL link. This implies that applications running on the same device have to share the ACL link. The traffic flows, generated by each application, compete for resources over the ACL link. These traffic flows however may have different Quality of Service (QoS) requirements in terms of bandwidth and delay. Furthermore devices within a Piconet have to share the available air-interface bandwidth among each other. Therefore there is also contention for resources between devices in a Piconet. There is no guarantee that when multiple traffic flows contend for resources, that these flows will share the available resources in such a way that the Quality of Service requirements of each flow is satisfied. Therefore control over resource allocation is required to guarantee that the QoS requirements of each flow is satisfied.

A usage scenario where there is contention between traffic flows on the same device and contention between devices in the Piconet is exemplified in Figure 28. Without resource control, there is the risk that the bursty WWW traffic transmission interferes with the audio transmission on either the same devices or on a different device. It should be noted that QoS mechanisms do not increase the system's capacity as such, but in many cases improve the Quality of Service for one traffic flow at the expense of another traffic flow.

Figure 28 Usage scenario with resource contention between traffic flows.

3.3 Service differentiation

When there is contention for resources, the QoS functions enable to provide service differentiation. Service differentiation provides the ability to provide a 'better' service for one traffic flow at the expense of the service offered to another traffic flow. Service differentiation is only applicable when there is a mix of traffic. The service level is specified by means of QoS parameters such as bandwidth and delay. In many cases there is a trade-off between QoS parameters e.g. higher bandwidth provides lower delay. It should be noted that service differentiation does not improve the capacity of the system. It only gives control over how the limited amount of resources that satisfy the needs of the different traffic flows.

3.4 Bluetooth configuration

Quality of Service plays a role when there is contention for resources between traffic flows. But also in case of a single flow, configuration of the Bluetooth link can be required to meet the QoS requirements of the flow. Examples of these configuration issues are: Flush Timeout setting, Multi-slot control and Maximum poll interval. However it can be argued that the control over certain functions in the Bluetooth layer should be left to the Bluetooth layer. An example is the selection of Baseband packet types for

transmission. The Bluetooth layer has the information concerning the wireless link conditions needed to select the optimum packet type for transmission i.e. whether to use FEC or multi-slot packets.

3.5 Bluetooth bitpipe

The Bluetooth technology has been developed to provide a low cost wireless interface. Simplicity has been an important design objective for the Bluetooth interface. Therefore the question arises whether elaborate and complex QoS mechanisms are appropriate to be included in Bluetooth. Why not consider the Bluetooth datalink layer as a simple bitpipe and solve any QoS problem at the higher layers? For example QoS architectures at the network layer, such as Integrated Services and Differentiated Services, provide means to provide different services to traffic, as depicted in Figure 29.

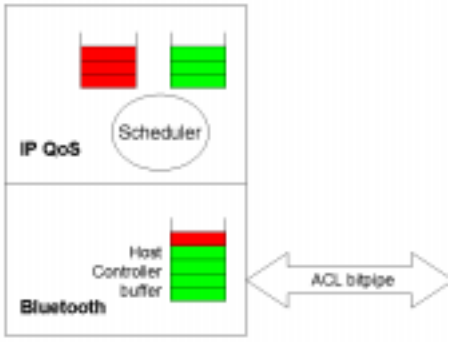

Figure 29 IP QoS mechanism to provide Quality of Service over Bluetooth.

Whether service differentiation at the higher layers is sufficient depends on the particular scenario and cannot be answered in a general sense. To give a feeling for this problem let's consider the scenario where both a traffic flow, generated by a WWW browser, and an audio flow, generated by a Voice over IP application, share the ACL link. The WWW traffic flow generates large packets in bursts and the audio flow periodically generates small audio frames. The WWW traffic flow uses five-slot packets and the audio flow uses single-slot packets. The audio flow generates a 20 bytes frame every 20 msec. (8 kbps). This requires a single slot each 20 msec. This leaves at most 5 DH5 packets downstream for the WWW traffic per 20 msec. The maximum downstream WWW bandwidth is thus 5 * 339 bytes / 20 msec. is 678 kbps. Let's assume that the Host Controller buffer can store 10 HCI Data Packets of 800 bytes long i.e. 8,000 bytes. In the worst case scenario the Host Controller buffer is completely filled with WWW traffic when an audio frame is ready for transmission. This causes the audio frame to be delayed around 94 msec.! This is an unacceptable high delay for a Voice over IP application. It should be noted that the delay is even higher when smaller packets are used, when FEC is applied, when the bandwidth has to be shared with other devices and when there are re-transmissions on the air-interface. Basically the problem is the Host Controller (HC) buffering delay. The buffering delay can be reduced by reducing the HC buffer size, however there are practical limitations how small the HC buffer can be. The Link Manager schedules tasks based on the content of the HC buffer. The Host Controller buffer should have a minimum size for the Link Manager to schedule tasks efficiently. Therefore a solution is preferred that does not rely on the HC buffer size, but allows traffic in the HC buffer to be treated differently. The QoS enhancements, evaluated in Chapter 5, are based on the latter approach.

3.6 Wireless link characteristics

Bluetooth, similar to other wireless interfaces, has different characteristics in terms of bandwidth and delay compared to fixed link interfaces. Roughly speaking the Bluetooth link has limited bandwidth, relatively high delay and high delay variation compared to fixed line interfaces. Furthermore the Bluetooth link has a relatively long link setup delay and Bluetooth has no specific features that support handovers efficiently. Although the Bluetooth interface has been designed to operate in a environment where there is interference, the presence of interference can have an effect on the effective bandwidth and delay characteristics. Although the Bluetooth link makes an attempt to hide the wireless link

characteristics to the higher layers, in certain cases it may fail to do so. Therefore the higher layers need to take into account these underlying wireless link characteristics. An example of the higher layers taking into account wireless link characteristics is the application of header compression on the wireless link [ROHC]. The most important implication of the wireless link characteristics for the Quality of Service is the type of guarantees that can be provided, as explained in Section 3.8.

3.7 Quality of Service in Piconet and Scatternets

From a Quality of Service point of view it is preferred to minimise the number of wireless hops in the end-to-end path. However there can be certain scenarios where multiple wireless hops cannot be avoided. The Quality of Service enhancements that are discussed in this document are limited to a single Piconet where the number of wireless hops is at most two (i.e. a slave-to-master and a master-to-slave hop). The Scatternet scenario, where there can be more than two wireless hops, is outside the scope of this document.

3.8 QoS guarantees

A Best Effort type of service does not provide any guarantees w.r.t. Quality of Service (QoS) parameters such as bandwidth, delay, delay variation, bit error ratio, loss probability, ordering, etc. A better than Best Effort type of service typically provides some kind of QoS guarantee. QoS guarantees can be divided into four categories, as listed in Table 1.

QoS guarantee	**Example**
Quantitative	Delay < 10 ms.
Statistical	In 95% cases delay < 10 ms.
Qualitative	Delay as in lightly loaded network
Relative	Delay better than another QoS class in the same system

Table 30 QoS guarantees.

Quantitative

A Quantitative QoS guarantee, also denoted as hard guarantee, specifies the QoS parameter quantitatively. The Quantitative QoS guarantee is the strongest guarantee that can be provided. However over a wireless link no quantitative QoS guarantee can be provided because the user may move out of range or the radio link conditions may deteriorate due to interference or adverse radio propagation conditions. However when the radio link conditions are ideal, the QoS guarantee over a wireless link approximates the QoS guarantee which can be provided over a fixed link.

Statistical

The Statistical QoS guarantee specifies the QoS parameter with the help of probability calculus. A statistical QoS parameter allows for some relaxation compared to a quantitative parameter.

Qualitative

The Qualitative QoS guarantee gives a certain qualification to the QoS parameter. The Qualitative QoS guarantee leaves more implementation freedom compared to a Quantitative QoS guarantee, but it also introduces some uncertainty about which exact QoS level to expect.

Relative

> A Relative QoS guarantee specifies the QoS parameters relative to other QoS guarantees in the same system. The Relative QoS guarantee can be considered the weakest form of QoS guarantee. A Relative QoS guarantee only makes sense when there is a mix of traffic classes.

The Bluetooth layer is not able to provide a Quantitative QoS guarantee i.e. hard guarantees, due to the wireless characteristics, as explained in Section 3.6. In case the Bluetooth interface is enhanced with QoS features, e.g. priority handling, then the Bluetooth interface can provide Relative guarantees. The same applies for the Statistical and Qualitative guarantees, provided that the error conditions on the air-interface can somehow be controlled.

3.9 QoS parameters

Different QoS parameters can be identified, such as bandwidth, delay, delay variation, reliability, and ordering. In the following sections these QoS parameters will be reviewed in the context of Bluetooth.

3.9.1 Bandwidth

For an application to run satisfactory, a certain amount of bandwidth needs to be available on the Bluetooth link. The amount of bandwidth influences the queueing delay experienced in the transfer of data. An application may either specify the amount of bandwidth it requires explicitly or the amount of bandwidth may be derived from a traffic specification and the associated delay requirement(s). The Guaranteed service [RFC2212], defined within the Integrated Services architecture, provides an example of how the required bandwidth is derived from a traffic specification and the delay bound. Furthermore the bandwidth requirements may be constant (e.g. peak bandwidth) or variable (e.g. average bandwidth).

Basically the air-interface bandwidth in Bluetooth is determined by the polling algorithm executed by the master of the Piconet, and the Baseband packet type, selected by the Link Manager for transmission.

The wireless link may suffer from interference and adverse propagation conditions, which necessitate re-transmissions to provide reliability. The amount of air-interface bandwidth should include a ‘re-transmission bandwidth’, which suits the application needs.

3.9.2 Delay and delay variation

Delay sensitive applications such as interactive real-time and streaming audio/video applications, may have strict requirements on the delay. Applications which employ a playback buffer, have a clear delay bound determined by the playback time i.e. depth of the playback buffer. The delay variation, also denoted as jitter, in most practical cases is defined as the difference between the minimum and maximum delay.

Delay and delay variations can have different causes in Bluetooth. In the following text a list of possible causes is given. The primary cause of delay is the amount of bandwidth available on the air-interface (see previous section) and the time when this bandwidth is available. Furthermore re-transmissions can cause additional delay. The flow control mechanism on the air-interface causes delay when the receiving device cannot process the data in time. Low power modes and the fact that a device, participating in multiple Piconets, has to time-multiplex between those Piconet, can cause delay. The connection establishment takes several seconds in Bluetooth, which is a possible source for delay. SCO traffic has priority over ACL traffic, therefore SCO traffic can delay ACL traffic. The ACL link is shared between both L2CAP user data and L2CAP / LMP signaling, which is a possible cause for delay. The L2CAP packets need to be transmitted sequentially over the same ACL link. Thus the ongoing transmission of a large L2CAP packet can delay the transmission of a new ACL packet. The Link

Manager is free in the selection of the Baseband packet type for transmission (within certain limitations imposed by the higher layer and the remote device). This implies that it may wait for more data to arrive, to send a multi-slot packet or send a single-slot packet immediately. It should be noted that this list of possible causes for delay is not exhaustive.

3.9.3 Reliability

Any application prefers the data to arrive correctly, however some applications are more tolerant to errors than others. In general audio and video applications can tolerate some errors, however typical data applications such as WWW browsing and email require the data to arrive correctly.

In general FEC encoding provides robustness against bit errors. Thus FEC can be used to support a wireless link with a low link budget or FEC can be used to extend the range of the wireless link. FEC adds redundancy bits to the data bits to detect bit errors and possibly correct bit errors. However by adding redundancy the effective throughput of the ACL link is reduced. The effectiveness of FEC depends on the bit error ratio and the distribution of bit errors. With 2/3 rate FEC single bit errors can be corrected and double bit errors can be detected. Once the number of bit errors exceeds this limit, FEC becomes ineffective. Thus FEC alone cannot provide a ‘reliable’ channel. FEC is more effective when bit errors are more or less randomly distributed compared to the case where bit errors occur in bursts. As a general rule, external interferers, such as a micro-wave ovens or WLAN, cause bit errors to occur in bursts, while a low link budget causes bit errors to occur more randomly. Bluetooth provides the ability to apply FEC dependent on the quality of the radio link, thereby allowing flexibility in the usage of FEC. The decision to use FEC is made by the Link Manager, where information concerning the actual radio link conditions is available. As a general rule the Link Manager should be allowed to use as many ACL packet types (with and without FEC) as possible, to operate in an efficient way.

The other method to provide reliability is to apply ARQ i.e. when the packet is not acknowledged in the return packet, it is re-transmitted. Re-transmissions also require bandwidth and thus reduce the effective throughput. Furthermore re-transmissions add to the transfer delay of packets. Re-transmissions are performed when either the actual data packet is lost, or when the acknowledgement is lost. The maximum number of re-transmissions is determined by the Flush Timeout setting. The Flush Timeout defines the maximum time period (msec.) during which re-transmission attempts are allowed. There is a single Flush Timeout setting per ACL link. When there is a Flush Timeout, the pending segments for transmission over the ACL link are being flushed from the Host Controller buffer. Furthermore all subsequent Continuation segments, which are stored in the Host Controller buffer for the ACL link, are flushed, until a new First segment for this ACL link is received. Dependent on the content of the Host Controller buffer, multiple L2CAP packets may be flushed.

Clearly, there is a trade-off between reliability and bandwidth/delay. Improving the reliability reduces the effective bandwidth and increases the delay. For data applications such as WWW browsing, email and file transfer, reliability has priority over delay. The data should be transferred reliably, and preferably as fast as possible. However for interactive real-time and streaming applications delay has priority over reliability. In general these applications use a play-back buffer which renders the information useless when it arrives after the playback time. For these applications the data should arrive in time, and preferably with the highest reliability.

3.9.4 Ordering

Applications that make use of the Internet, have to be robust against packet re-ordering, as the Internet may not preserve the ordering of IP packets. For this purpose TCP/IP and RTP add sequencing information to the data packets to restore the order at the receiver. However Bluetooth preserves the order of data packets. This feature is provided by the Baseband layer.

4 Bluetooth 1.0 deficiencies to support Quality of Service

There are several reasons why the Bluetooth 1.0 specification lacks in the support of Quality of Service. These deficiencies are discussed in this chapter.

4.1 L2CAP sequential transmission rule

The L2CAP packets transmitted over the ACL link must be transmitted sequentially i.e. the L2CAP packet transmission must be completed before a new L2CAP transmission starts. This implies that HCI Data Packets of different L2CAP packets may not be transferred 'interleaved' to the Host Controller buffer. The Baseband maintains the ordering of the L2CAP segments i.e. transmits the content of the Host Controller buffer in FIFO[10].

In case multiple applications with different QoS requirements run on the same Bluetooth device, the sequential transmission rule can cause problems to provide delay guarantees to applications that require this (see Section 2.8). For example when there is both a WWW browser and a voice application running on the device, then a large WWW L2CAP packet transmission must be completed before a new Voice frame transmission can start.

4.2 ACL service

There are several drawbacks of the current ACL service, which are discussed in this section.

4.2.1 ACL (QoS) service support

The default service provided by the ACL links is a Best Effort service with no bandwidth delay guarantees. The Guaranteed service is also defined for the ACL link, but a polling algorithm has not yet been defined which supports this type of service (see also Section 3.1). But there is an immediate need to an ACL service that provides bandwidth and delay guarantees. Furthermore there is the need to support different ACL services simultaneously, to support multiple applications with different QoS requirements on the same Bluetooth device.

4.2.2 Re-transmit filtering

Bluetooth re-transmit filtering rule

The SEQN bit in the header of an ACL packet with CRC check is used to detect and discard duplicates at the receiver (see Section 2.5). The receiver keeps track of the SEQN bit associated with the last correctly received payload, denoted as $SEQN_{old}$. When a packet is received, the SEQN of this packet is compared to $SEQN_{old}$. When they are the same the payload is designated as duplicate and is discarded. When the SEQN is different from $SEQN_{old}$ then the payload is designated as new and is accepted. This rule is denoted as the re-transmit filtering rule and included in the Bluetooth receiver protocol depicted Figure 31 (see Section 5.3.2 [BLUESPEC]).

[10] The advantage of the sequential transmission rule is that only a single L2CAP re-assembly buffer per ACL link is required.

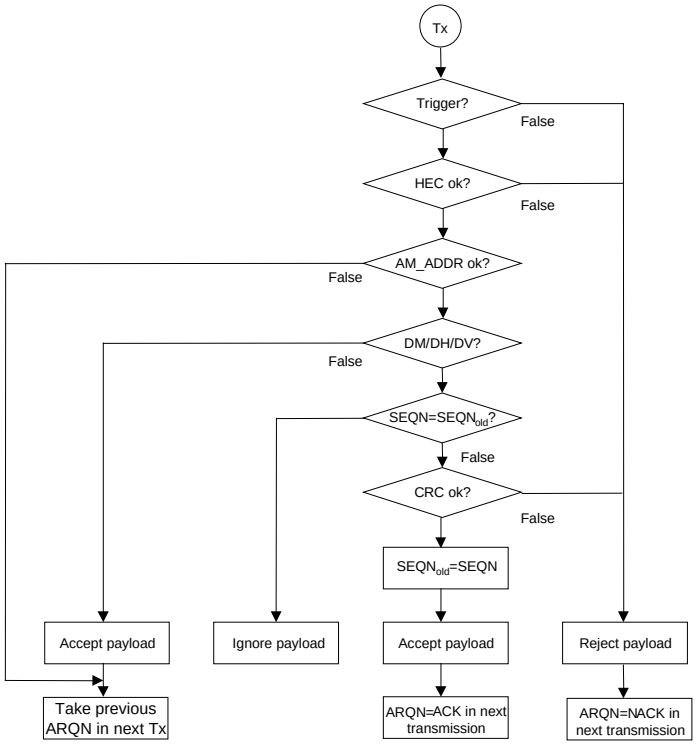

Figure 31 Receive protocol Bluetooth 1.0 (See Figure 5.3 pp 70 [BLUESPEC]).

The re-transmit filtering rule ensures that duplicates are discarded at the receiver. Duplicates are caused by re-transmissions. A re-transmission causes a duplicate when the acknowledgement of the previous payload transmission, which has been received correctly, is lost, see Figure 32. A re-transmission does not cause a duplicate when the previous transmitted payload was not received correctly.

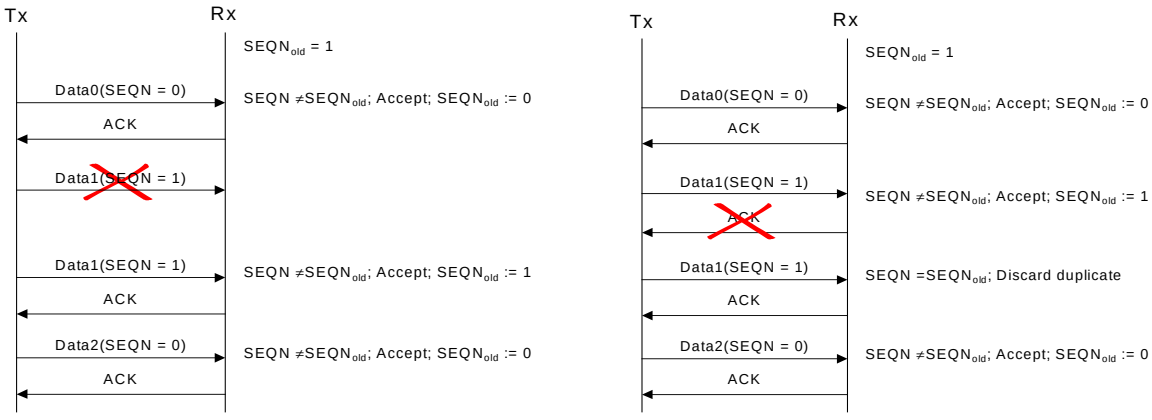

Figure 32 The re-transmit filtering ensures that a duplicate is discarded.

It should be noted that the payload is only accepted when the header is received correctly and the CRC check of the payload is ok. Thus the probability that the payload is lost is greater than the probability that the acknowledgement is lost. Furthermore the header is always encoded with 1/3 rate FEC, while the payload is optionally FEC encoded.

Re-transmit filtering rule deficiency

The re-transmit filtering rule is not failsafe i.e. the receiver may interpret a new payload as a duplicate and discard a new payload under certain conditions. This condition can arise when there has been a Flush Timeout at the transmitter: the transmitter has not received a positive acknowledgement within the Flush Timeout period and has flushed the Host Controller buffer. After the Flush the transmitter sends a new payload. As the transmitter does not know whether the receiver has received the previous payload correctly, it basically does not know whether to invert the SEQN bit or not. But as it is most likely that the previous payload has been lost instead of the acknowledgement, the SEQN bit is not inverted. The protocol for setting the SEQN bit at the transmitter is depicted in Figure 33.

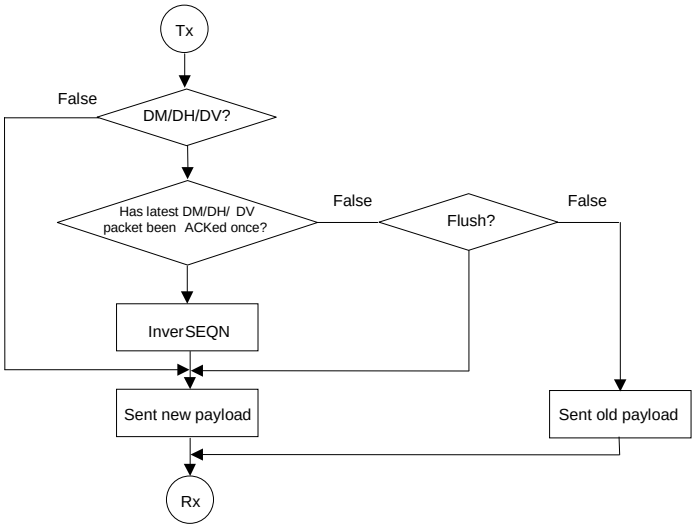

Figure 33 Rules for setting the SEQN bit (Section 5.3.2 [BLUESPEC]).

In case the payload has been received correctly but the acknowledgement is lost, then the transmitter will set the SEQN bit of the new payload transmission after a Flush such, that the receiver will designate the new payload as duplicate and discard it. However the more likely scenario is that the payload was not received correctly and the receiver will accept the new payload transmission, see Figure 34. Thus a new payload transmission after a Flush is discarded when the previous acknowledgement has been lost, which is denoted as the re-transmit filtering problem in this document.

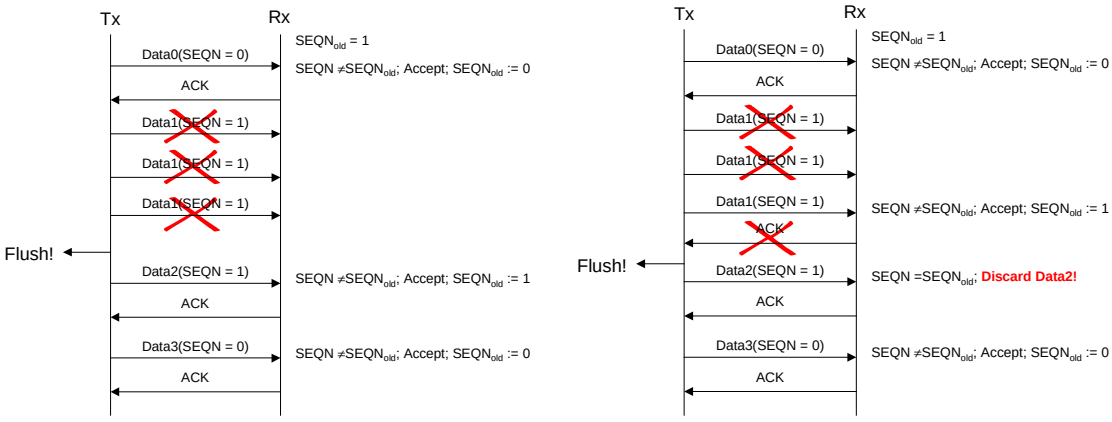

Figure 34 SEQN is not inverted after Flush.

The re-transmit filtering deficiency can arise when there has been a Flush, i.e. under conditions were there are problems to transmit the information over the Bluetooth interface correctly. It is argued that under such conditions it becomes important to prevent that once information is received correctly, this information is not erroneously discarded at the receiver.

It should be noted that the re-transmit filtering problem does not occur when the Flush Timeout setting is infinite. In such a case re-transmissions are carried out 'indefinitely' i.e. until the link is lost.

4.2.3 Packet type selection

The Packet type selection algorithm selects the Baseband packet type to be used for transmission over the air-interface. To run the selection algorithm efficiently the largest possible set of Baseband packets should be made available (see limitations Section 2.7.1). The selection algorithm thus decides whether to use FEC and/or multi-slot packets. The optimisation criteria of the algorithm is not specified in the Bluetooth specification. There are two important optimisation criteria: effective throughput and delay. The two parameters are correlated but do not necessarily lead to the same selection decisions.

Dependent on the dynamic error conditions on the air-interface, an optimum selection algorithm may be difficult to implement.

Once the selection algorithm has made a decision to use a certain packet type it is not allowed to change this decision until the next transmission. This implies that during re-transmissions it is not allowed to change packet type. The reason for this is that the sender does not know whether the payload was correctly received when it does not receive an acknowledgement.

4.2.4 ACL delay guarantees

There are several potential problems to provide bandwidth and delay guarantees for ACL service.

Re-transmissions

The error conditions on the air-interface may call for re-transmissions, which cause delay to successfully deliver ACL traffic.

SCO traffic

The timeslots reserved for SCO traffic may not be used for ACL traffic (or LMP signaling). This may cause ACL traffic to be delayed. Furthermore it limits the bandwidth that can be assigned to ACL traffic.

LMP signaling

The LMP channel uses the ACL link for transmission. The LMP signaling has priority over ACL traffic, but may not interrupt ongoing re-transmissions.

Polling

The polling algorithm used in Bluetooth has the inherent problem that the master is not aware of the instantaneous traffic demand of the slave. This implies that the master may poll a slave which has nothing to send (i.e. wasting bandwidth). Furthermore a slave, which has something to send, has to wait for the poll of the master.

Multi-slot usage and SCO traffic

The ACL traffic may use the timeslot in between those reserved for SCO traffic. However a slave is not aware of SCO timeslot reserved by another slave. Therefore a problem arises when a slave is allowed to use multi-slot packets and the slave is polled just before the SCO timeslots. In such a case the slave may respond with a multi-slot packet overlapping the reserved SCO timeslots. The scheduling algorithm should prevent this situation, which may imply that polling the slave is postponed until after the SCO timeslots.

Packetisation

The Packet selection algorithm is free to select the Baseband packet type and may wait for more data to arrive to fill a multi-slot packet. The same packetisation delay arises when HCI Data Packet are 're-assembled' at the Baseband receiver.

Low power modes

The typical usage for the different Low power modes is listed in Table 8.

Low power mode	Typical usage
HOLD	Detect and connect new devices (Inquiry/Paging)
SNIFF	Periodically be present in different Piconets (Scatternet support)
PARK	Power saving mode (listening to broadcast channel)

Table 8 Low power modes.

When a device enters a Low power mode, the delay and bandwidth characteristics are significantly changed.

Master-slave switch

When a master-slave switch is performed, there can be no data transfer for several TDD frames.

4.3 Flow Control

Host Controller buffer starvation / lockout

When multiple applications runs on the same Bluetooth device, then the traffic flows belonging to the different applications end up together in the Host Controller buffer. When no special action is taken it is possible that one traffic flow locks out another traffic, as depicted in Figure 35. For example the Best Effort traffic flow could occupy the whole Host Controller buffer preventing QoS traffic to enter. This can seriously affect the delay guarantees for the QoS flow, even in the case when the Baseband layer would give priority to QoS traffic.

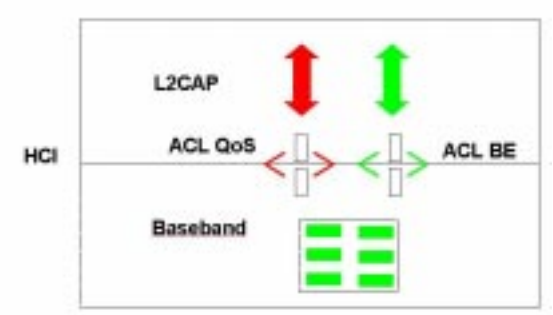

Figure 35 Host Controller buffer lockout.

Beside buffer lockout, there is the issue of buffer starvation. When the L2CAP layer does not provide the data in time, the Host Controller buffer may become empty. Furthermore when the Baseband employs a window mechanism, in which it schedules its tasks in the future, then the Host Controller buffer content should be at least equal to this window, otherwise the Baseband will run idle.

Flow Control over multiple wireless interfaces

In a Piconet there can only be direct communication between the master and a slave. To transmit from slave to slave, the information needs to be sent via the master. Thus the number of 'wireless hops' in a Piconet is at most two. In a Scatternet the number of 'wireless hops' is not limited. In a Piconet/Scatternet scenario there can be different bandwidth available on each link, which can cause congestion on the link that is the bottleneck, as depicted in Figure 36.

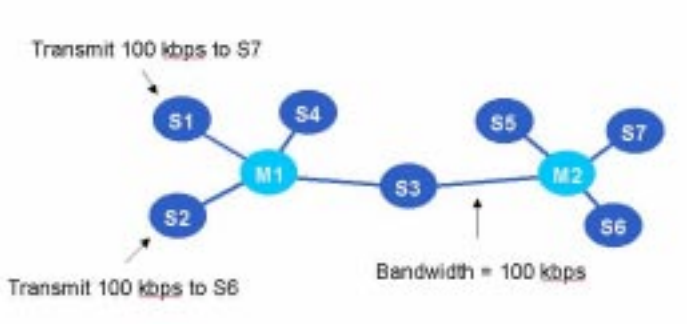

Figure 36 Congestion in Scatternet.

The Bluetooth Flow Control mechanisms should prevent buffers to overflow. There is a Stop-and-Wait Flow Control mechanism on the ACL link, stopping all ACL data transfer. Furthermore there is the mandatory HCI Flow Control mechanism in the Host to Host Controller direction and the optional HCI Flow Control mechanism in the Host Controller to Host direction, as depicted in Figure 37.

These three Flow Control mechanisms, when connected, can exercise backpressure when traffic cannot be forwarded on the congested link. However this backpressure mechanism can cause the ACL traffic to be stopped on multiple links in the Scatternet. Furthermore the Flow Control mechanism stops all ACL traffic on the link and not only the traffic flow (i.e. application) that causes the congestion.

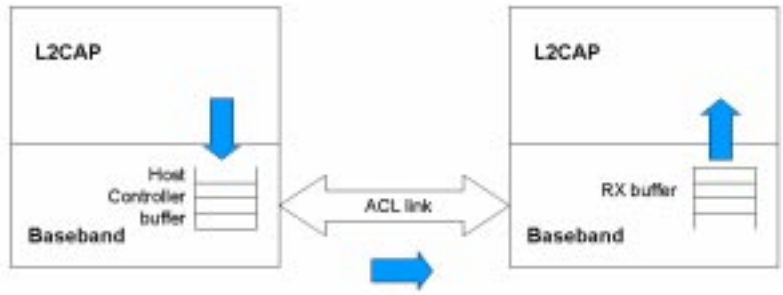

Figure 37 Bluetooth Flow Control.

4.4 Setup delays

The Inquiry and Paging procedures are relatively long in Bluetooth. The main reason for this is the absence of a common control channel in Bluetooth. Also handovers are poorly supported in Bluetooth. When a new Bluetooth device is connected to an Access Point with already connected devices, special attention should be paid that the Quality of Service of the ongoing connections is not affected.

4.5 Layer 2 forwarding vs. layer 3 routing

To support TCP/IP over Bluetooth there are two alternative solutions: Layer 3 routing or Layer 2 forwarding. With Layer 3 routing, each Bluetooth device is capable of routing IP packets. On each hop

the traffic goes up to the IP layer where the packet is either routed to the next hop, or passed to the higher layers. With Layer 2 forwarding not each Bluetooth device performs IP routing, but the Bluetooth layer itself supports the forwarding of packets within the Bluetooth domain.

The decision to support layer 2 forwarding or layer 3 routing has an impact on the QoS support for Bluetooth, as depicted in Figure 38. In case of layer 3 routing, the QoS support only has to consider a single wireless interface. After each interface the IP packets go up to the IP layer where a new reservation is made on the next wireless interface. Thus when the IP layer requests a bandwidth reservation, then this bandwidth reservation only concerns a single wireless interface. In case of layer 2 forwarding then a bandwidth request from the IP layer can entail the bandwidth reservation on multiple wireless interfaces. A bandwidth reservation on multiple wireless interfaces is more complex to implement compared to a bandwidth reservation on a single wireless interface. A reservation mechanism over multiple wireless interfaces is not further evaluated in this document. The reservation mechanism evaluated in this document will be limited to a single wireless interface.

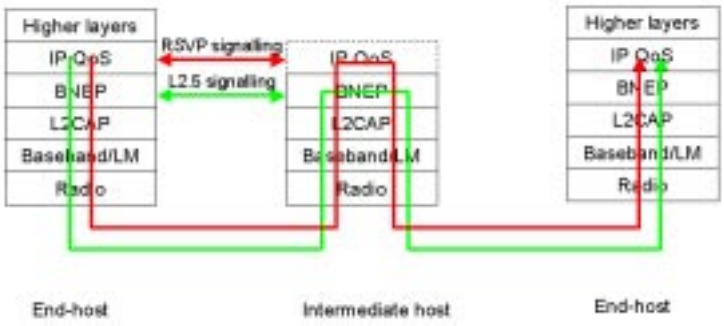

Figure 38 Layer 3 routing vs. layer 2 forwarding.

4.6 Traffic Control

Traffic Control has the function to protect the network from an (excessive) use other than that negotiated between network and user. In the Differentiated Services architecture Traffic Control is applied at the entrance of the Diffserv domain and in the Integrated Service architecture Traffic Control is applied in each network node by means of Policing. Traffic Control is important when Quality of Service guarantees need to be provided, as is discussed in Section 3.8.

The Traffic Control function can be implemented by the polling algorithm used to control traffic over the ACL link.

Special attention should be paid to the fact that when priorities are introduced in a Scatternet, priority traffic over a congested link in the Scatternet can starve Best Effort traffic from that link. Traffic Control can prevent priority traffic to load a congested link above a certain threshold to leave room for Best Effort traffic to be transferred.

5 Bluetooth Quality of Service Framework

In this chapter enhancements of the Bluetooth 1.0 specification are evaluated to improve the Quality of Service support in Bluetooth. First the requirements for these enhancements are defined in Section 5.1. Next a general framework, that defines the basic functions required to support Quality of Service, is discussed in Section 5.2. In this section also different implementation alternatives for these functions are evaluated.

5.1 Requirements

First general requirements on the QoS enhancements are defined. These general requirements are defined taking into account the specific characteristics of Bluetooth. Next requirements specific for L2CAP, HCI and Baseband are defined.

5.1.1 General

- Strive for simplicity! The Bluetooth technology is and should remain a simple and low cost technology. The Quality of Service (QoS) enhancements should be kept as simple as possible while serving the applications needs. The QoS enhancements should have no significant increase in the cost and power consumption.

- The requirements imposed on processing capacity, memory size and power consumption should be limited[11]. Furthermore the signaling load in support of QoS enhancements should be moderate. If the QoS functions require user configuration or management support, this should be easy.

- The QoS enhancements should preferably be defined as extensions to the current Bluetooth specification. Devices not requiring QoS enhancements should continue to operate in the same way. Before the QoS enhancements can be used between two devices this should be negotiated[12]. The QoS enhancements should allow future extensions.

- The QoS enhancements should facilitate the Bluetooth Low Power modes. Furthermore the QoS enhancements should be able to co-exist with the Inquiry and Paging procedures. The QoS enhancements should be applicable in both a Piconet and Scatternet scenario, although the full set of QoS enhancements may be not available in a Scatternet scenario.

- The QoS enhancements should allow multiple applications with different QoS requirements to run on the same Bluetooth device and/or in the same Piconet.

- The QoS enhancements should allow the roaming between different wireless interfaces such as Bluetooth, WLAN and cellular systems (e.g. GPRS and UMTS). This includes support to run Mobile IP efficiently over Bluetooth.

- The QoS enhancements should be able to interwork with other QoS mechanisms and architectures such as Differentiated Service, Integrated Services (RSVP) and Load Control.

- The QoS enhancements should include the support for header compression [ROHC] to more efficiently support the transfer of IP based applications.

---

[11] It should be noted that the processing, memory and power limitations can different from Bluetooth device to Bluetooth device. Thus for some Bluetooth devices the requirement is more stringent than for others.

[12] In the Bluetooth 1.0 specification some features are mandatory (e.g. single slot packet support) and some features are optional (e.g. Multi-slot packet support). For these optional features first a negotiation must be performed (e.g. LMP Supported Features) before these features can be used. The QoS enhancements discussed in this chapter can be considered as the negotiated features.

5.1.2 L2CAP

- The L2CAP Configuration option should allow the negotiation of the Traffic, QoS and Service type requirements for each new ACL service (e.g. ACL Isochronous and ACL Priority service).

- The L2CAP layer should allow L2CAP re-assembly per ACL Logical link, i.e. ACL Connection Handle.

- L2CAP packets transmitted over the same ACL Logical link, i.e. the same ACL Connection Handle, should be transmitted sequentially. L2CAP packets transmitted over different Logical ACL links, i.e. different ACL Connection Handles, need not be transmitted sequentially i.e. the HCI Data Packets may be transmitted interleaved.

5.1.3 HCI interface

- The HCI interface should provide the ability to negotiate the Traffic, QoS and Service type requirements for each new ACL service (e.g. ACL Isochronous and ACL Priority service). When the ACL Logical link establishment is successful a Connection Handle for this service at both the local and remote device is established.

5.1.4 Baseband

- The Baseband should support the establishment of multiple simultaneous ACL Logical links. With each ACL Logical link a Baseband service is associated. The default ACL service is the Best Effort service.

- The Baseband should support a new Isochronous service over the ACL link which provides bandwidth and delay guarantees.

- The Baseband should support a new Priority service over the ACL link, which gives scheduling priority over ACL Best Effort traffic, but does not provide bandwidth and delay guarantees.

- The Baseband should support the ACL Best Effort service efficiently.

- The Baseband should support the ACL Guaranteed service by introducing new scheduling algorithms.

- The Baseband layer should support a new Low Bitrate Low Delay service. This service allows higher layer information to be transported in the payload of LMP signaling message. The Baseband layer should support a Policing function to limit the use of the LMP signaling channel by higher layer information.

- Dependent on the required bandwidth the Baseband layer should allow the simultaneous support of one or more SCO links and one or more ACL Logical links.

- The ACL Logical links should be independent. The ACL Logical links are handled independently w.r.t. to Flow Control, ARQ / SEQN, Flush Timeout setting, Flush of Host Controller buffer and Packet type selection and Multi Slot Control. This allows one Logical link to interrupt the (re-) transmission of another Logical link.

- The ACL Logical links can be symmetric and (pure) asymmetric.

- The Baseband should allow the mapping of LMP signaling onto an ACL Logical Link.

- The Baseband layer should allow the negotiation to use an improved CRC checksum of 32-bits.

- The Baseband layer should provide a means to solve the ‘re-transmit filtering’ problem i.e. avoid that the recipient incorrectly discards new data because it interprets this new data as duplicate.

5.2 General QoS framework and design alternatives

5.2.1 General QoS Framework

In this section the basic functions required to provide Quality of Service in Bluetooth are discussed. The functions are discussed in a general way, without going into implementation details. The functions included in the QoS framework are depicted in Figure 39. The QoS functions are explained in the following text. The Bluetooth layers, where these functions can be implemented, will be discussed as well. No assumptions are made about the layers above Bluetooth. In the following text IP is used as an example, however the IP interoperability and service mapping is not further discussed in this document.

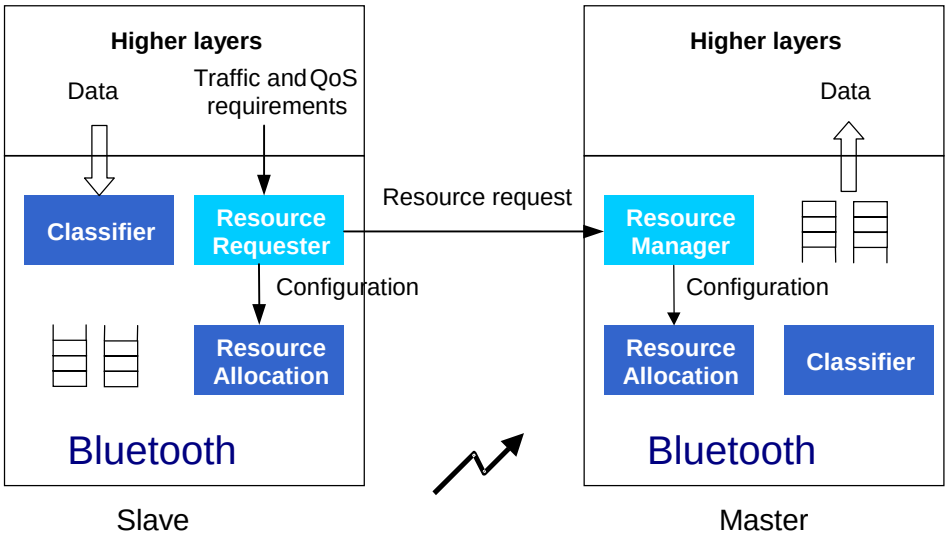

Figure 39 General QoS framework.

5.2.2 Resource reservation

The Resource Requester (RR) and Resource Manager (RM), depicted in Figure 39, enable to reserve air-interface bandwidth for a QoS flow in the Piconet. A (QoS) flow is defined as a uni-directional flow of traffic between two Bluetooth devices with the same Quality of Service requirements. The RR makes a request for resources on behalf of the QoS flow to the RM. The RM manages the air-interface resources of the Piconet and either admits or rejects resource requests from the RR. Resource reservation is done before the start of the QoS data transmission and the reservation is cancelled when the QoS flow is terminated.

The RR receives the Traffic and QoS requirements of the QoS flow from the higher layer. For example the Traffic specification contained in an RSVP message (Tspec) could be passed to the RR. Based on these Traffic and QoS requirements the RR generates a request for resources to the RM. When the request is accepted the RR entity may provide configuration parameters to the local Resource Allocation (RA) entity. The RA actually assigns the reserved resources to the traffic flow e.g. applies a scheduling mechanism that satisfies the QoS requirements.

Resource reservation in a Piconet is relatively easy as the master of the Piconet can handle the resource requests. However Resource reservation in a Scatternet is more complex, because different Piconets are involved. In this scenario the reservation mechanism may have to recover from the situation where the reservation over a single wireless hop fails.

Resource reservation on behalf of a QoS flow can also be found in the Integrated Service architecture. In the Integrated Services architecture an explicit reservation of bandwidth at the intermediate network elements is made. A 'soft' state is installed per QoS flow, which enables Classification and Scheduling of incoming IP packets belonging to the QoS flow. In each network element there is an Admission Control entity that manages the bandwidth of the network element. It either accepts or rejects the resource request of a QoS flow. Only because a certain amount of bandwidth is reserved specifically for the QoS flow the Guaranteed service [RFC2212] is able to provide hard QoS guarantees i.e. assured level of bandwidth, maximum end-to-end IP datagram transfer delay and no packet loss due to buffer overflow. In the Differentiated Services architecture there is no bandwidth reservation on a per flow basis. In the Differentiated Service architecture only Traffic Aggregates are considered, which is a large bundle of individual (micro)flows. In a Diffserv network traffic is not handled per flow but per Differentiated Service behaviour aggregate i.e. the set of IP datagrams with the same Differentiated Service Code-Point (DSCP) value in the IP header. Dependent on the DSCP value, an IP packet receives a certain forwarding treatment in the intermediate network elements. The externally observable forwarding behavior of the network elements is also referred to as Per Hob Behavior (PHB). The Expedited Forwarding (EF) PHB defines Qualitative QoS guarantees: low loss, low latency and low jitter and a Quantitative QoS guarantee with respect to an assured banwidth. The Assured Forwarding (AF) PHB defines Relative QoS guarantees: within each AF class different drop precedence levels are identified. However it should be noted that Both the EF and AF PHB require that there are sufficient resources, such as bandwidth and buffer space, in the network elements within the Differentiated Service domain to handle the associated behavior aggregates. So also in the Differentiated Service architecture bandwidth must be provided. However this bandwidth is not reserved on a per flow basis, but managed on a longer time scale by the operator. The Traffic Control functions are performed at the border of the Diffserv domain where traffic enters the network. The Traffic Control functions ensure that traffic entering the network abides the rules agreed in a Service Level Agreement (SLA). The SLA is an agreement between a customer and the operator of the Diffserv domain about the traffic characteristics of the offered traffic and the associated QoS requirements. The Traffic Control measures i.e. meters the traffic, possible marks the traffic and may drop packets that do not meet the criteria of the SLA. Traffic Control ensures that the network resources are not overloaded i.e. that the QoS requirements of the SLA can be met.

Resource reservation, Admission Control and QoS Load Control provide similar functions. Resource reservation allows the reservation of resources for a traffic flow such that its QoS requirements are satisfied. Admission either accepts or rejects a new QoS flow based on the available network resources. When a QoS flow is admitted, this implies that sufficient resources are there to serve the QoS flow. Furthermore by rejecting a new QoS flow a QoS Load Control function is provided. The rejected QoS flow possibly has to resort to a Best Effort type of service.

The primary function of Admission Control is to guarantee that the QoS level is maintained during the lifetime of the QoS flow. Without Admission Control there can be no guarantee that the QoS level is maintained, as the QoS traffic into the network is not controlled. Without Admission Control there is the risk that the system resorts to a Best Effort type of service when the QoS load is too high. However in usage scenarios where the system is 'over-provisioned' i.e. there are always sufficient resources to satisfy a 'QoS flow' request, Admission Control can be omitted. In such a usage scenario the Admission Control will always accepted the request.

A secondary function of Admission Control is that is can provide configuration parameters for Resource Allocation. Although there are sufficient network resources, to make efficient use of them, configuration of Resource Allocation may be needed. An example is a scenario where there are a data terminal, providing WWW browsing, and an audio device, with a low bitrate coded, which require access through a Data Access Point. In case the DAP is the master, the DAP could be configured to poll the audio device with a low frequency and assign the remaining bandwidth to the data terminal. Without configuration the DAP could possibly assign half the bandwidth to each device, which results in inefficient use of resources.

Design alternatives

In this section two alternatives to enable Resource reservation signaling are described. Resource reservation requires the signaling of Traffic and QoS requirements over the air-interface in case of a Slave-to-Master flow. A Slave-to-Master flow is a flow that originates at the slave. The basic operation for Resource reservation for a Slave-to-master QoS flow is depicted in Figures 40. The numbers in the figure indicate the sequence in which the signaling messages are be transferred.

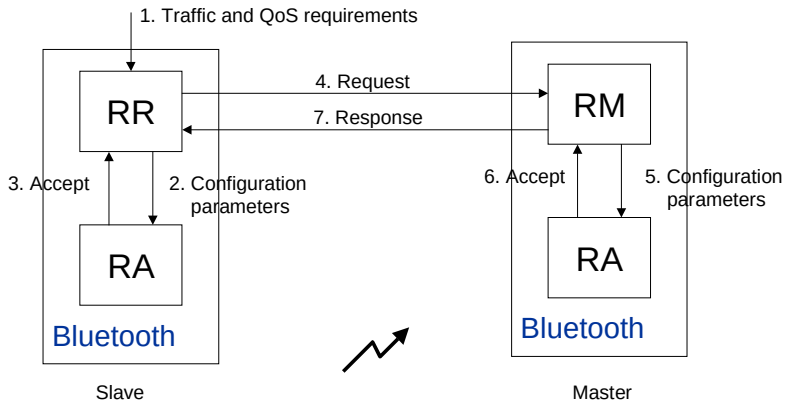

Figure 40 Resource reservation for Slave-to-Master QoS flow.

Basically there are two options to implement the Resource Reservation mechanism with the existing Bluetooth procedures: L2CAP QoS option or LMP quality of Service, as depicted in Figure 41 and 42 respectively.

The L2CAP QoS option has the advantage of providing extensive information concerning the Traffic and QoS requirements. With this information intelligent and elaborate Resource Allocation schemes can be defined. Furthermore multiple reservations can be made by configuration of multiple L2CAP channels. However the HCI interface does not support 'multiple' HCI QoS Setups. The disadvantage of the L2CAP QoS options is that complexity of Resource Allocation is shifted to the Baseband layer, where Resource Allocation is performed.

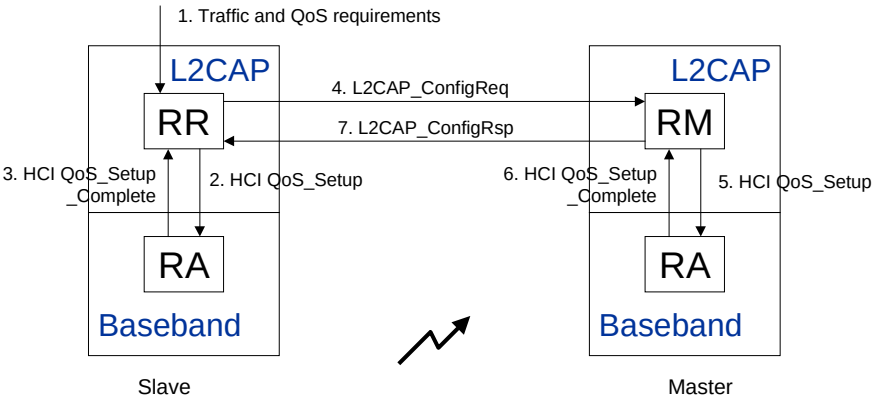

Figure 41 Resource reservation with L2CAP QoS option for Slave-to-Master QoS flow.

It should be noted that the HCI QoS_Setup command contains the same parameters except the Token Bucket Size field is not included.

With the LMP Quality of Service, the Traffic and QoS requirements are translated into a simple configuration parameter for Resource Allocation i.e. maximum poll interval. This limits the configuration

of (new) Resource Allocation schemes i.e. polling algorithms. Furthermore the LMP Quality of Service does not allow the reservation for different service classes identified on the ACL link.

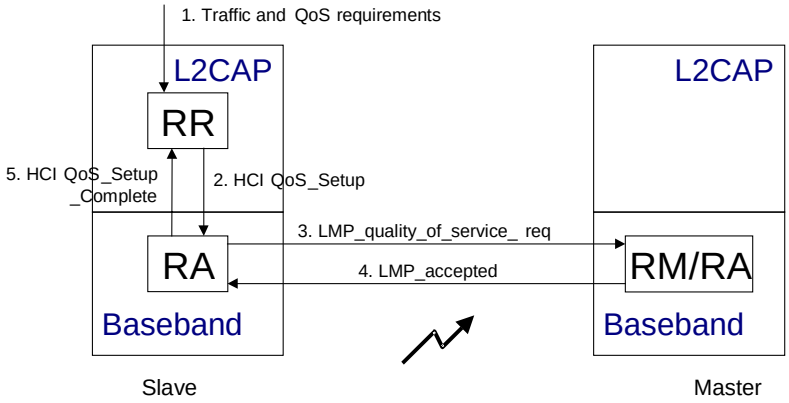

Figure 42 Resource reservation with LMP Quality of Service for Slave-to-Master QoS flow.

5.2.3 Resource Allocation

The primary function, needed to support Quality of Service, is control over the usage of resources. When an application receives the appropriate amount of resources, its Quality of Service requirements will be satisfied. The QoS framework should enable the QoS flow to receive the appropriate amount of resources. Different traffic flows have to share the air-interface resources and the resources on the local and remote Bluetooth devices, as depicted in Figure 31.

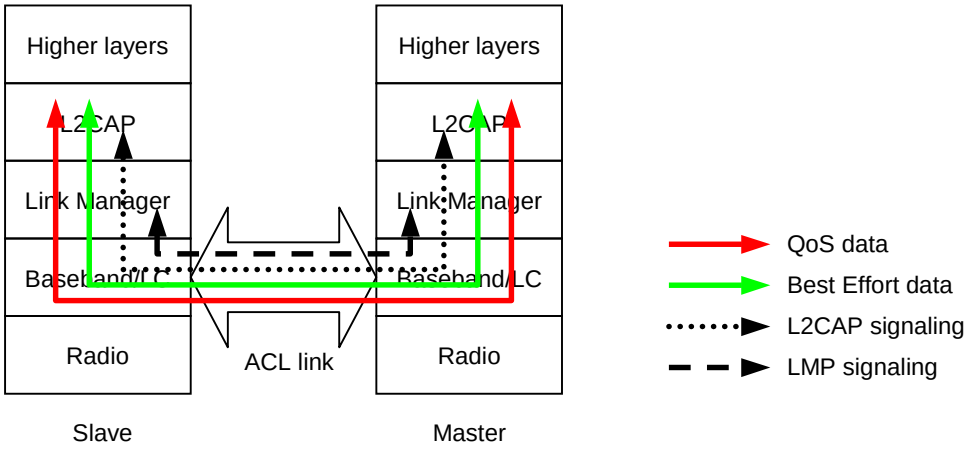

Figure 43 Resources required to provide Quality of Service.

The following functions and procedures in Bluetooth can be identified, that determine the amount of resources assigned to a traffic flow:

Polling algorithm:

> The polling algorithm which is executed by the master of the Piconet. The polling algorithm decides which Bluetooth device is polled next in the Piconet. The frequency with which the master polls a slave determines the bandwidth that is assigned to that slave.

Inter-Piconet scheduler:

Inter-Piconet scheduling algorithm, used by a master/slave participating in more than one Piconet. The Inter-Piconet Scheduler divides the limited amount of bandwidth of a Bluetooth device over the Piconets the device is active in.

Air-interface scheduler:

The Air-interface scheduler in a slave decides which data to send when it is polled.

L2CAP Packet and HCI Data Packet transfer rules:

There is a mandatory flow control mechanism in the host to Baseband direction and an optional flow control mechanism in the other direction on the HCI interface. Within the constraint imposed by these flow control mechanisms, HCI Data is transferred between host and Baseband. L2CAP packets over the same ACL link are required to be transmitted sequentially.

Baseband packet type:

The Link Manager selects the Baseband packet type for transmission. The Link Manager in the master of the Piconet controls the usage of multi-slot packets.

Flush Timeout setting:

The Flush Timeout setting which determines the amount of ‘re-transmission’ bandwidth and the maximum delay involved with re-transmissions.

L2CAP and Host Controller buffer space:

Dependent on the buffer space the L2CAP and Host Controller are able to absorb bursts of data.

Basically there are three different approaches for Resource Allocation in general and the polling algorithm in particular:

Static configuration

The polling algorithm is pre-configured, which means that no configuration parameters are supplied before or during the operation of the polling algorithm. An example of a pre-configured polling algorithm is Round Robin. In general this method is simple, but is limited in the flexibility to suit the particular QoS flow needs. It is unlikely that the Static configuration provides the required QoS in an environment where there is contention for air-interface resources. However in a over-provisioned environment it may be sufficient.

Dynamic configuration during QoS setup

The polling algorithm is configured during the ‘QoS’ setup i.e. configuration parameters derived from the QoS flow requirements are provided to the polling algorithm. An example of this approach is when a maximum poll interval is defined for the QoS flow. The polling algorithm is configured to satisfy the QoS flow’s requirements. This method is flexible in the sense that Resource Allocation can be configured to the specific needs of the QoS flow. In general this approach can be implemented efficiently when the traffic source, for which Resource Allocation is configured, is predictable e.g. constant bitrate sources. However when the traffic source is unpredictable, e.g. variable bitrate sources, this approach poses problems to be implemented efficiently.

Dynamic configuration during data transfer

> During data transfer the decisions how to perform Resource Allocation are made, possibly with the help of information obtained via signaling or other means. For example the polling algorithm estimates the actual traffic demand during the execution of the polling algorithm. The traffic demand may either be signalled explicitly e.g. by providing feedback information about the amount of data waiting for transmission, or signalled implicitly by the fact whether data or a NULL packet is returned upon a POLL. This approach can provide an efficient solution when the traffic sources are unpredictable e.g. variable bitrate sources. However some bandwidth may be needed to signal the traffic demand.

Combinations of the above Resource Allocation approaches can be identified e.g. a polling algorithm that is configured with a minimum poll frequency but uses 'feedback' information to poll more efficiently. It should be noted that the configuration during setup approach is associated with Resource reservation, which is discussed in Section 5.2.2. Furthermore it should be noted that Resource Allocation applies both to the ACL and SCO links.

5.2.4 Classification

A prerequisite to enable service differentiation is that certain traffic can be separated from other traffic. This function of separating traffic is performed by Classification. The Classification function should at least separate QoS traffic from Best Effort traffic and possibly identify different types of QoS traffic. For example when the higher layer is an IP layer that supports Differentiated Services, then classification could be performed based on the Differentiated Service Code-Point (DSCP) value contained in the IP header[13]. Another function performed by Classification is the marking of traffic. It is unlikely that the Bluetooth layer uses the higher layer marking (e.g. DSCP value in IP header) because the higher layer data is segmented into smaller pieces in the Bluetooth layer. Marking is done by encoding the Bluetooth header, which is used to encapsulate the higher layer data. Marking should be performed to enable QoS traffic to receive preferential treatment at both the transmitting and receiving side of the Bluetooth link and to enable to establish a 'QoS' connection over multiple Bluetooth hops. Two different approaches for Classification can be identified, which will be discussed next.

Pre-defined and fixed number of QoS service classes:

> There is a limited amount of pre-defined QoS service classes and each 'QoS enabled' Bluetooth device is aware of these classes and knows how they should be handled. Each Bluetooth PDU contains a marking that identifies to which QoS service class this PDU belongs and thus identifies how it should be handled. The marker itself identifies the semantics of the service. This approach does not require the establishment of a connection before the service can be used (connectionless). This approach is similar to the Differentiated Services architecture where the DSCP value identifies the requested Per Hob Behavoir (PHB) for that IP packet (see also Section 5.2.2). A very simple example of how this approach could be used in Bluetooth is that each packet is assigned a priority level which indicates how it should be handled relative to other traffic.

Classification based on installed state information:

> Prior to data transfer, state is installed in the Bluetooth device, which identifies how the traffic should be handled. The installed state is usually associated with the connection over which the data is transferred. This implies that the connection is actually configured according to the specific requirements of the QoS data. Each PDU contains an identification to which connection it belongs and thus how it should be handled. The installed state defines the service semantics, not the identification carried in the PDU.

[13] This implies that Classification is performed both at the IP and Bluetooth layer.

This approach requires signaling prior and after the transmission of QoS traffic to install and release the associated state in the Bluetooth layer. This approach is similar to the Integrated Services architecture, where Classification state is installed in the intermediate network elements to identify QoS traffic.

The advantages and disadvantages of both options for Classification are listed in Table 44.

	Pre-defined QoS service classes	**Installed flow state information**
Advantages	• Does not require signaling to install/release state • More easily supported over multiple wireless hops • Less signaling required with mobility (to install new flow state at new point of attachment)	• Granularity of separation is per flow, which allows traffic control functions to be applied per flow (e.g. policing) • Configurable to application needs i.e. flexible • Enables Statistical and Qualitative QoS guarantee
Disadvantages	• Bluetooth remote capabilities must be established before marked packet may be used • Granularity is per QoS service class (e.g. priority level), which allows traffic control functions per class and not per flow • Provides good QoS as long as QoS load is not too high • Without Admission Control (AC) only relative QoS guarantees and risk of Best Effort type of service • With AC the advantage of simplicity is diminished • Limited number of QoS service classes i.e. limited flexibility • Implementation of AC requires traffic control functions (e.g. meters, shapers and droppers) • More difficult to handle Resource Reservation	• Requires signaling to install/release state for each flow in each node • Complex over multiple wireless hops (e.g. scatternet) • Complex with mobility

Table 44 Advantages and disadvantages of the Classification options.

When considering the Bluetooth protocol suite it is clear that the L2CAP layer is suitable to support the 'installed state' approach for Classification. Between two Bluetooth devices multiple L2CAP channels can be established, which subsequently can be configured to the specific needs through the L2CAP Configuration Option (see Section 2.8.3). At the Baseband layer however there is currently a single ACL link, which has to carry the different traffic flows. In the following text it is explored how the ACL link can be used to support multiple traffic flows with different QoS requirements. The higher layer traffic flows are classified and mapped onto separate L2CAP channels, as depicted in Figure 45.

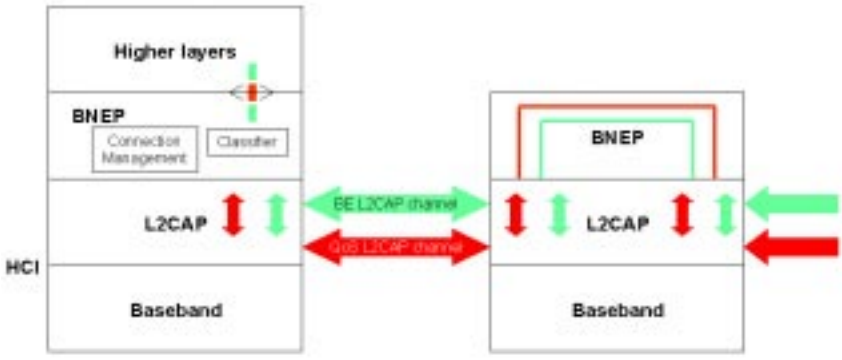

Figure 45 Classification L2CAP per channel.

The L2CAP Channel Identifier (CID) is used for Classification. For example Best Effort traffic is mapped onto one L2CAP channel, while QoS traffic is mapped onto another L2CAP channel which is configured by means of the L2CAP QoS option.

To establish a Bluetooth QoS connection over multiple hops, the Bluetooth Network Encapsulation Protocol (BNEP) has the task to map incoming L2CAP channels onto an appropriate outgoing L2CAP channel, as depicted in Figure 45. There is no need to include Classification information in BNEP packets. The mapping of incoming L2CAP channels onto outgoing L2CAP channels can be done locally (e.g. based on CID) and does not require BNEP signaling.

In case there is only a single QoS flow on the Bluetooth device, but there are other (QoS) flows within the Piconet, then only contention between flows in the Piconet can arise and not between flows on the same device. The problem is then not that different flows have to share the same ACL link, but that different flows have to co-exist in the same Piconet. In such a scenarion mapping of the L2CAP channel onto the ACL link which is configured by means of the HCI QoS Setup can be used to solve the contention within the Piconet, as depicted in Figure 46. In this scenario there is no need for service differentiation on the ACL link.

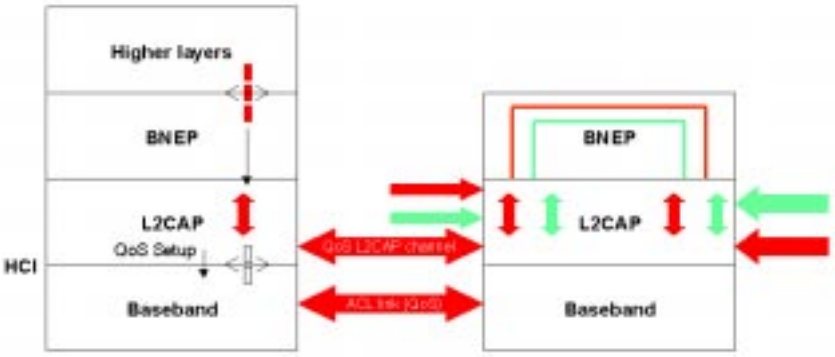

Figure 46 L2CAP QoS channel mapped onto ACL link configured by means of HCI QoS Setup.

In case there are multiple (QoS) flows per device then these flows have to share the same ACL link. Service differentiation at the Baseband layer is required to satisfy the requirements of each flow. To allow service differentiation at the Baseband layer, the Classification information needs to be carried over the HCI interface. Two options are identified:

- Establish separate Connection handles (Figure 47). A possible solution is that the HCI QoS Setup command returns a new Connection handle.
- Carry Classification information in HCI Data Packet (Figure 48). This solution has the disadvantage that the existing HCI Data Packet format needs to be changed.

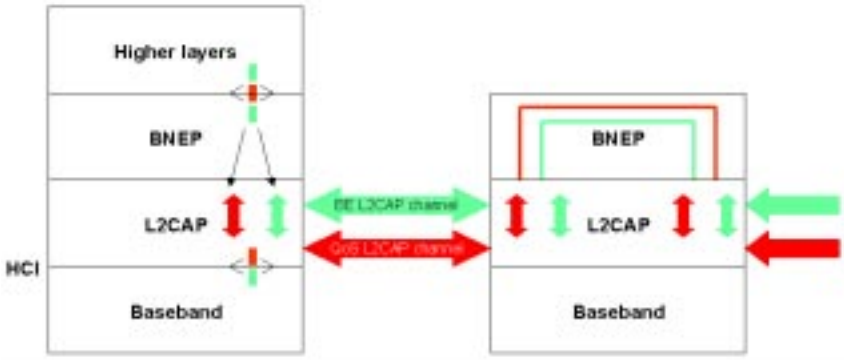

Figure 47 Classification based on information in Data Packet format on HCI interface.

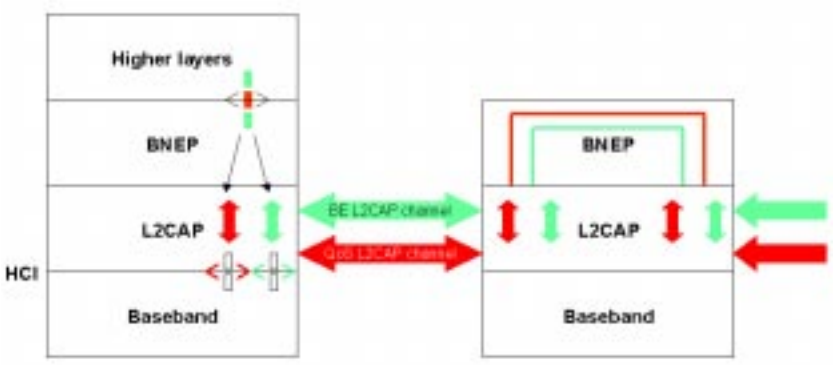

Figure 48 Classification based on Connection Handle on HCI interface.

When there are separate Connection Handles, then the existing flow control mechanism of the HCI interface applies to the set of Connection Handles defined. In the following text it is assumed that there is a separate Connection Handle to carry QoS traffic over the HCI interface. First it is investigated what can be achieved without marking Baseband traffic and next what is achieved when marking Baseband traffic.

Dependent on the implementation the Host Controller buffer can contain a large number of Baseband packets. Thus the queueing delay in the Host Controller buffer can be a major delay in the Baseband. Especially the FIFO transmission of the data stored in the Host Controller buffer, can cause QoS traffic, that is appended at the end of the queue, to be delayed by Best Effort traffic stored at the head of the queue. When there are separate Connection Handles for QoS and Best Effort traffic, then the Baseband can separate this traffic locally, as depicted in Figure 49.

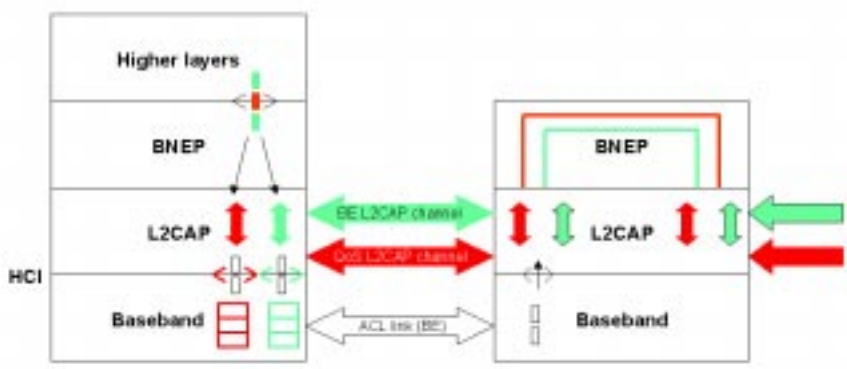

Figure 49 No marking of ACL traffic.

This information can be used by the Baseband to give priority to QoS traffic at the expense of the Best Effort queueing delay. But this requires the Baseband layer to be aware of the L2CAP packet boundaries. This can be achieved by inspecting the HCI Data Packet header, which indicates if this segment is a First or Continuation segment of an L2CAP packet. Assuming that the L2CAP layer transfers the L2CAP packets sequentially, then there are two options to establish the end of an L2CAP packet. Either the next First segment indicates the end of the previous L2CAP packet or the L2CAP header of the First segment is inspected to read the length field. In the first method, delay may be involved with establishing the end of an L2CAP packet. When the Baseband has established the L2CAP packet boundaries two options are identified:

- The Baseband gives priority to QoS L2CAP packets stored in the Host Controller buffer, but L2CAP packets are still transmitted sequentially. There are no changes required to the receiving Baseband or L2CAP re-assembly procedure. Implementation of this procedure eliminates the major part of the Host Controller queueing delay. However an L2CAP QoS packet has to wait for the completion of an

ongoing L2CAP Best Effort packet before transmission can start. Suppose that the Best Effort L2CAP packets are 1,500 bytes long, then in worst case (assuming a transmission rate of 100 kbps) an L2CAP QoS packet has to wait 120 msec. for the Best Effort packet to complete, before transmission of the QoS packet can start. This is an unacceptable high delay for many audio and video applications. Of course the transmission rate can be higher than 100 kbps, but even with the highest possible transmission rate on the ACL, the delay is in the order of 15 msec. From this analysis it is concluded that it is not sufficient to give priority to L2CAP packet transmission to provide Quality of Service.

- The Baseband may interrupt the transmission of a lower priority L2CAP packet to transmit a higher priority L2CAP packet. Interruption in this context means that Baseband packets of a higher priority packet are transmitted first before the remaining Baseband packets of the 'interrupted' L2CAP packet are transmitted. Therefore an L2CAP QoS packet does not have to wait for the completion of an ongoing L2CAP Best Effort packet, however it may have to wait for an ongoing re-transmission to be completed. When a high priority L2CAP packet is allowed to interrupt a low priority L2CAP packet transmission, then special attention should be paid to the re-assembly procedure of the high and low priority. At the receiver there is a single re-assembly buffer where all the L2CAP segments of high and low priority packets are stored. A re-assembly procedure that allows an L2CAP transmission to be interrupted and resumed is described below. This procedure requires that the transmission of a higher priority L2CAP packet is completed before the transmission of a lower L2CAP packet is resumed. Re-assembly of L2CAP packets at the receiving side is enabled by 'stacking' the received L2CAP segments, as depicted in Figure 50. In Figure 50 a low priority packet, consisting of six segments, is interrupted by a high priority packet transmission of three segments. In the low part the development of the L2CAP re-assembly buffer in time is depicted. First the two segments of the low priority packet are stored, next the high priority packet is appended. When the last segment of the high priority packet is received, the re-assembly attempt of the high priority packet will be successful and the high priority packet is removed from the re-assembly buffer. Finally the remaining segments of the low priority packet will arrive and the low priority packet will be re-assembled successfully.

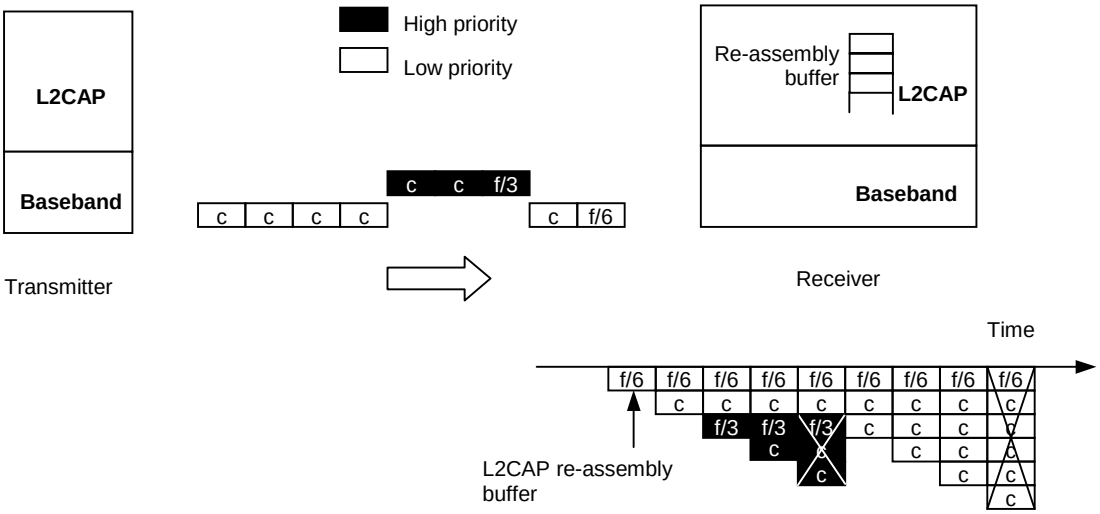

Figure 50 High priority packet interrupt low priority packet.

However this method has a performance penalty when there is a Flush Timeout for an L2CAP packet which interrupted another L2CAP packet. In such a case not only the interrupting L2CAP packet is flushed, but the interrupted L2CAP packet is lost as well. The frequency with which this type of event occurs and the effect on the higher layer applications should be evaluated further. Note that at the receiving L2CAP side there is still a single re-assembly buffer. To clean-up the L2CAP re-assembly buffer after a Flush, it is possible to use a L-CH coding as listed in Table 9.

L_CH code	Information
00	First segment of an L2CAP packet and a Flush Timeout occurred at the transmitting side
01	Continuation segment of an L2CAP packet
10	First segment of an L2CAP packet
11	LMP message

Table 9 New L_CH code in ACL payload header.

An alternative solution is to use the number of priority levels as a means to flush the re-assembly buffer. When there are for example two priority level then the L2CAP buffer can be flushed partially when a third 'First' segment is enqueued, see Figure 51. Then only the oldest content until the first First segment is flushed.

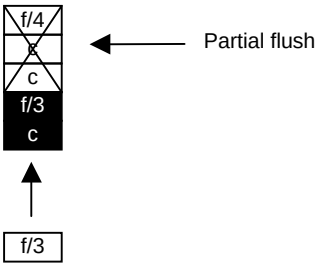

Figure 51 Flush policy L2CAP re-assembly buffer.

It should be noted that transmission of a 'Flush occurred' indication over the air-interface does not solve the problem that the Flush of the interrupting packet causes the interrupted packet to be lost as well. To prevent this service differentiation over the ACL is required, which is discussed next.

To recover more efficiently from a Flush, Baseband packets should be marked as e.g. high and low priority. This allows the implementation of a separate re-assembly buffer at the receiver for the high and low priority packets. In this case when there is a Flush for a high priority packet transmission, the low priority packet transmission will not be affected. This actually implies that separate Logical links are identified over the ACL link. The operation of one Logical link w.r.t. error control is independent from the operation of another Logical link. Thus the aim is to establish separate Logical links on the ACL link, as depicted in Figure 52.

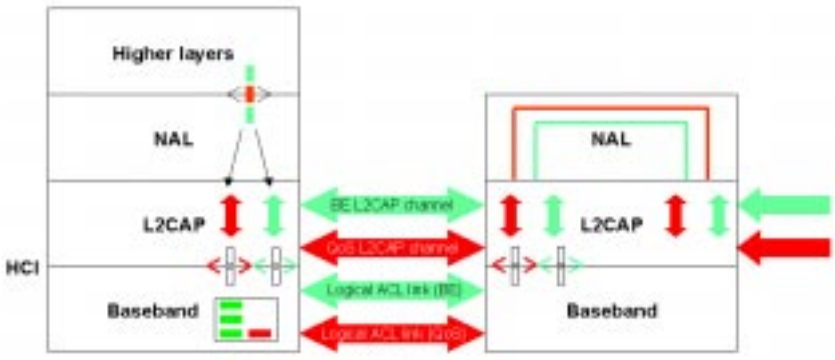

Figure 52 Establishment of Logical links over the ACL link.

6 Conclusions

The Quality of Service functions and procedures included in the Bluetooth 1.0 specification have been reviewed. Next issues associated with providing Quality of Service over a wireless link in general and Bluetooth in particular have been investigated. Allthough the Bluetooth 1.0 specification provides some Quality of Service support, some deficiencies have been identified. Especially the need to establish multiple concurrent Logical links over the same ACL link is considered important. Based on this analysis requirements for QoS enhancements of the Bluetooth 1.0 specification have been defined. Furthermore a Quality of Service Framework for Bluetooth has been presented.

7 Future work

UMTS/GPRS – Bluetooth service mapping

The UMTS system defines four service classes: Conversational, Streaming, Interactive and Background service classes. Associated with these service classes, Radio Access Bearers (RAB) can be established with different attributes such as: guaranteed bitrate, maximum bitrate, transfer delay, SDU error ratio, etc. The Bluetooth architecture also defines different service classes with different QoS parameters. To provide a seamless end-to-end service, a service mapping of UMTS to Bluetooth services needs to be defined. Possibly this mapping is done by mapping both services to IP QoS classes. Different possible service mappings may exist, but at least a recommended service mapping should be defined. Furthermore the mapping of Bluetooth QoS parameters onto UMTS RAB attributes must be evaluated.

UMTS/GPRS – Bluetooth QoS signalling interworking

The establishment, maintenance and release of radio bearers both within Bluetooth and UMTS requires signalling. The establishment of radio bearers that satisfy the QoS requirements requires the signalling of QoS requirements as well. To efficiently establish a radio bearer over the Bluetooth and UMTS air-interface, requires the interworking between Bluetooth and UMTS signalling. Again, this signalling interworking might be done by mapping both types of signalling to IP QoS signalling.

Impact of UMTS/GPRS control algorithms

The UMTS system is a complex system with different control algorithms at different protocol layers and interfaces. The Radio Link Control (RLC) provides error control procedures and flow control. There are the resource control algorithms such as Admission Control, Congestion Control, Packet Scheduling and Channel Switching. Furthermore there are the power control algorithm and the header compression algorithm used on the air-interface.

There is the difficult task of the UMTS designers to implement the control algorithms such that the requirements of the selected Service class with the associated QoS parameters can be met. In practical cases the requirements may not be met. Therefore it is good to have some understanding of the impact of the UMTS control algorithms on the service offering.

IP Quality of Service support

Two different QoS architectures are defined on the IP layer: Integrated Services and Differentiated Services. Furthermore there is the signalling protocol RSVP that enables resource reservation in the Integrated Service architecture. The IP Quality of Service mapping and RSVP signalling support for Bluetooth needs to be investigated.

High speed radio

The current Bluetooth interface provides a raw bitrate of 1 Mbps. An enhanced modulation scheme is proposed that increases this bitrate to 2 Mbps. for 3G support. A new radio design is proposed to enable a Bluetooth high-rate mode in excess of 7 Mbps. The Ericsson proposal for this new radio design has been accepted which provides 4 Msymbols/second with an adaptive modulation scheme of 8/4/2-PSK.

The Quality of Service features specified for this new radio should be studied and the support for high speed multi-media communications should be evaluated.

Impact of Power saving, Inquiry and Paging modes on Quality of Service

The Bluetooth interface has been defined to support devices with limited power capabilities. The Bluetooth interface provides three different modes to lower the power consumption: PARK, HOLD and SNIFF. The use of power saving modes however reduces the bandwidth and increases the delay for data transfer.

The same modes are used to allow devices to participate in several Piconets at the same time (e.g., to from a Scatternet). The impact of switching between Piconets (i.e., inter-Piconet scheduling) on QoS is part of this study area.

To establish a Bluetooth connection Inquiry and Paging procedures are used. Due to the ad-hoc network architecture of Bluetooth and power consumption limitations, the setup times with the current procedures are relatively high. Furthermore when an Access Point has already an active connection with one device, bandwidth is needed to establish a new connection with another device. Currently work is going on to improve the setup delay.

The impact of Power saving, Inquiry and Paging modes on the support of applications that require Quality of Service should be evaluated.

Intra-Piconet scheduling

The Round Robin polling algorithm is an easy to implement polling algorithm for Intra-Piconet scheduling. However the RR algorithm does not provide Quality of Service guarantees. A polling algorithm called Predictive Fair Polling (PFP) is being developed which aims to provide QoS guarantees. The PFP algorithm tries to predict the instantaneous traffic demand on the slave to improve the efficiency of the polling algorithm. The PFP algorithm also takes a fairness measure into account to assign the available resources to the slaves in a ‘fair’ way.

L2CAP channel management

In case the end-to-end connection includes multiple consecutive Bluetooth interfaces then at the intermediate Bluetooth nodes, an incoming L2CAP QoS channel need to be mapped onto an outgoing L2CAP QoS channel. This requires a channel management function in the L2CAP layer.

8 Abbreviations

AC	Access Code
ACL	Asynchronous Connectionless Link
AF	Assured Forwarding
AM_ADDR	Active Member Address
AMR	Adaptive Multi-Rate
ARQ	Automatic Re-transmission reQuest
AV	Audio Video
BD_ADDR	Bluetooth Device Address
BE	Best Effort
BER	Bit Error Rate
BNEP	Bluetooth Network Encapsulation Protocol
CID	Channel Identifier
CRC	Cyclic Redundancy Check
DAP	Data Access Point
DECT	Digital European Cordless Telecommunications
DSCP	Differentiated Services Code-Point
EF	Expedited Forwarding
ESDP	Enhanced Service Discovery Protocol
FEC	Forward Error Correction
FEP	Fair Exhaustive Polling
FIFO	First In First Out
3GPP	Third Generation Partnership Project
GFSK	Gaussian Frequency Shift Keying
GPRS	General Packet Radio System
GSM	Global System for Mobile communications
HC	Host Controller
HCI	Host Controller Interface
HID	Human Input Devices
IP	Internet Protocol
IR	Infrared
ISM	Industrial Scientific Medical
LBLD	Low Bitrate Low Delay
L2CAP	Logical Link Control and Adaptation Protocol
L-CH	Logical Channel
LC	Link Control
LMP	Link Manager
LOS	Line of Sight
MAC	Medium Access Control
MTU	Maximum Transmission Unit
PAN	Personal Area Network
PDA	Personal Digital Assistant
PDU	Protocol Data Unit
PFP	Predictive Fair Polling
PHB	Per Hop Behavoir
PM_ADDR	Parked Member Address
QoS	Quality of Service
RA	Resource Allocation
RR	Resource Requester
RSSI	Received Signal Strength Indicator
RSVP	Resource Reservation Protocol
RTCP	RTP Control Protocol
RTP	Real-time Protocol

RM	Resource Manager
SCO	Synchronous Connection-Oriented
SIG	Special Interest Group
SLA	Service Level Agreement
TCP	Transport Control Protocol
TDD	Time Division Duplex
UA	User Asynchronous
UDI	Unrestricted Digital Interface
UI	User Isochronous
UMTS	Universal Mobile Telecommunications System
USB	Universal Serial Bus
VAD	Voice Activity Detection
WLAN	Wireless Local Area Network
WWW	World Wide Web

9 References

[BLUESPEC] Title: Specification of the Bluetooth System; Core
Author: Bluetooth SIG
Doc. no.: Version 1.0 B
Date: December 1st 1999

[ROHC] Title: RObust Header Compression (ROHC)
Author: Carsten Burmeister, Christopher Clanton, Mikael Degermark, Hideaki Fukushima, Hans Hannu, Lars-Erik Jonsson, Rolf Hakenberg, Tmima Koren, Khiem Le, Zhigang Liu, Anton Martensson, Akihiro Miyazaki, Krister Svanbro, Thomas Wiebke, Takeshi Yoshimura, Haihong Zheng,
Doc. no.: Internet Draft (May 2001)
Date: November 24, 2000

[RFC2212] Title: Specification of Guaranteed Quality of Service
Author: S. Shenker, C. Partridge, R. Guerin
Doc. no.: RFC 2212
Date: September 1997

[HAART1] Title: Bluetooth
Author: J. Haartsen
Doc. no.: Ericsson Review, No. 3
Date: 1998

[HAART2] Title: The Bluetooth Radio System
Author: J. Haartsen
Doc. no.: IEEE Personal Communications
Date: February 2000

[ARFWED] Title: Ericsson's Bluetooth modules
Author: H. Arfwedson, R. Sneddon
Doc. no.: Ericsson Review, No. 4
Date: 1999

[MILLER] Title: Bluetooth Revealed
Author: B.A. Miller, C. Bisdikian
Publisher: Prentice Hall 2001
ISBN: 0-13-090294-2

[FEP] Title: Performance Evaluation of Scheduling Algorithms for Bluetooth
Author: N.J. Johansson, U. Korner, P. Johansson
Doc. no.: Proceedings of IFIP TC6 Firth International Conference Broadband Communications '99
Date: Hong-Kong, November 10-12, 1999