discoverable device scans for the LIAC, the initiating device may choose any inquiry procedure (general or limited). Even if the remote device that is to be discovered is expected to be made limited discoverable (e.g. when a dedicated bonding is to be performed), the limited inquiry should be done in sequence with a general inquiry in such a way that both inquiries are completed within the time the remote device is limited discoverable, i.e. at least $T_{GAP}(103)$.

6.2.2 Term on UI level

'Bluetooth Device Inquiry'.

6.2.3 Description

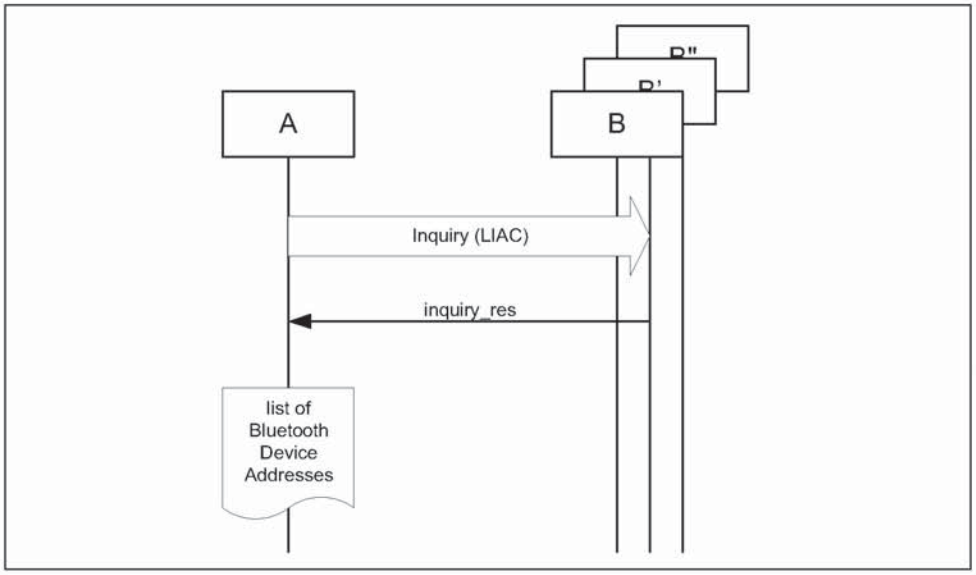


Figure 6.2: Limited inquiry where B is a device in non-discoverable mode, B' is a device in limited discoverable mode and B" is a device in general discoverable mode. (Note that only limited discoverable devices can be discovered using limited inquiry.)

6.2.4 Conditions

When limited inquiry is initiated by a Bluetooth device, it shall be in the INQUIRY state for at least $T_{GAP}(100)$ and perform inquiry using the LIAC.

In order to receive inquiry response, the remote devices in range has to be made limited discoverable.

6.3 NAME DISCOVERY

6.3.1 Purpose

The purpose of name discovery is to provide the initiator with the Bluetooth Device Name of connectable devices (i.e. devices in range that will respond to paging).

6.3.2 Term on UI level

'Bluetooth Device Name Discovery'.

6.3.3 Description

6.3.3.1 Name request

Name request is the procedure for retrieving the Bluetooth Device Name from a connectable Bluetooth device. It is not necessary to perform the full link establishment procedure (see Section 7.1) in order to just to get the name of another device.

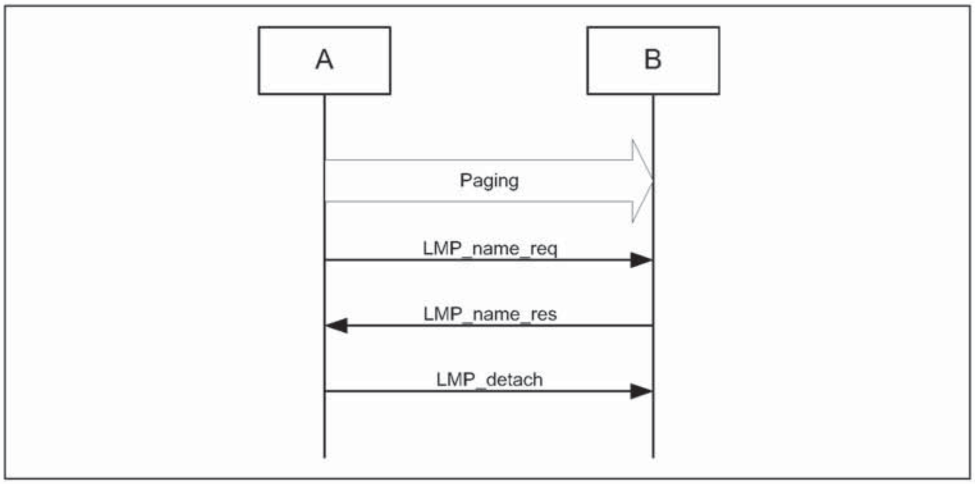


Figure 6.3: Name request procedure.

6.3.3.2 Name discovery

Name discovery is the procedure for retrieving the Bluetooth Device Name from connectable Bluetooth devices by performing name request towards known devices (i.e. Bluetooth devices for which the Bluetooth Device Addresses are available).

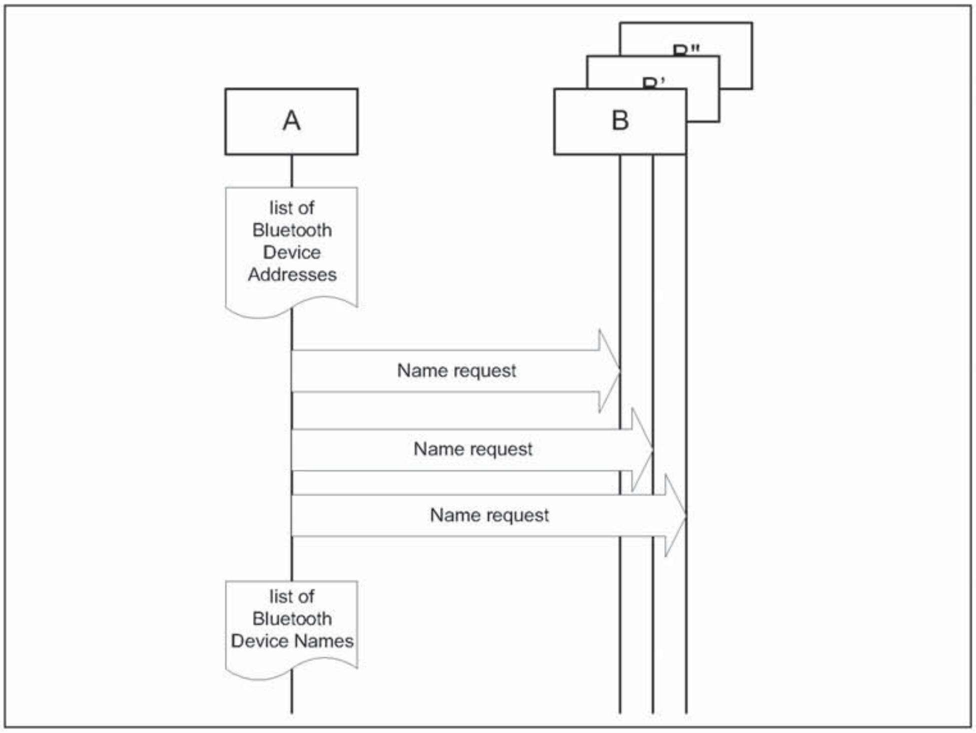

B"
B'
A
B
list of Bluetooth Device Addresses
Name request
Name request
Name request
list of Bluetooth Device Names

Figure 6.4: Name discovery procedure.

6.3.4 Conditions

In the name request procedure, the initiator will use the Device Access Code of the remote device as retrieved immediately beforehand – normally through an inquiry procedure.

6.4 DEVICE DISCOVERY

6.4.1 Purpose

The purpose of device discovery is to provide the initiator with the Bluetooth Address, clock, Class of Device, used page scan mode and Bluetooth device name of discoverable devices.

6.4.2 Term on UI level

'Bluetooth Device Discovery'.

6.4.3 Description

During the device discovery procedure, first an inquiry (either general or limited) is performed, and then name discovery is done towards some or all of the devices that responded to the inquiry.

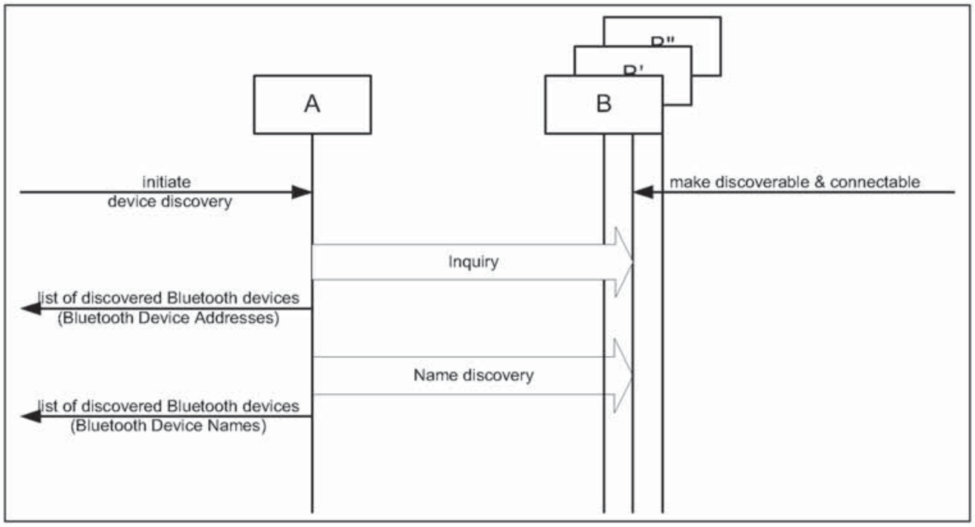

B"
B'
A
B
initiate
device discovery
make discoverable & connectable
Inquiry
list of discovered Bluetooth devices
(Bluetooth Device Addresses)
Name discovery
list of discovered Bluetooth devices
(Bluetooth Device Names)

Figure 6.5: Device discovery procedure.

6.4.4 Conditions

Conditions for both inquiry (general or limited) and name discovery must be fulfilled (i.e. devices discovered during device discovery must be both discoverable and connectable).

6.5 BONDING

6.5.1 Purpose

The purpose of bonding is to create a relation between two Bluetooth devices based on a common link key (a bond). The link key is created and exchanged (pairing) during the bonding procedure and is expected to be stored by both Bluetooth devices, to be used for future authentication.

In addition to pairing, the bonding procedure can involve higher layer initialization procedures.

6.5.2 Term on UI level

’Bluetooth Bonding’

6.5.3 Description

Two aspects of the bonding procedure are described here. Dedicated bonding is what is done when the two devices are explicitly set to perform only a creation and exchange of a common link key.

General bonding is included to indicate that the framework for the dedicated bonding procedure is the same as found in the normal channel and connection establishment procedures. This means that pairing may be performed successfully if A has initiated bonding while B is in its normal connectable and security modes.

The main difference with bonding, as compared to a pairing done during link or channel establishment, is that for bonding it is the paging device (A) that must initiate the authentication.

6.5.3.1 General bonding

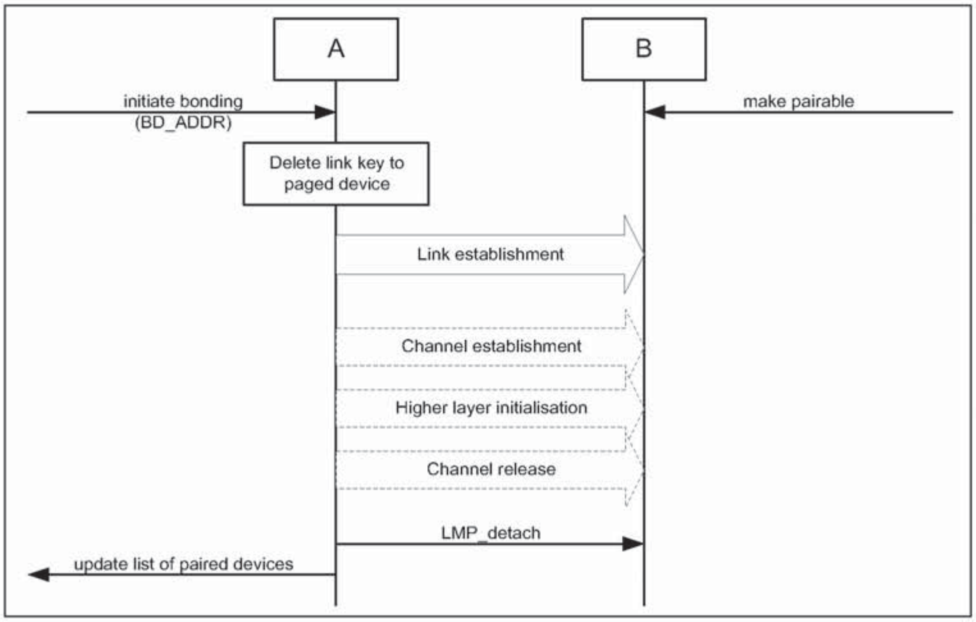


Figure 6.6: General description of bonding as being the link establishment procedure executed under specific conditions on both devices, followed by an optional higher layer initalization process.

6.5.3.2 Dedicated bonding

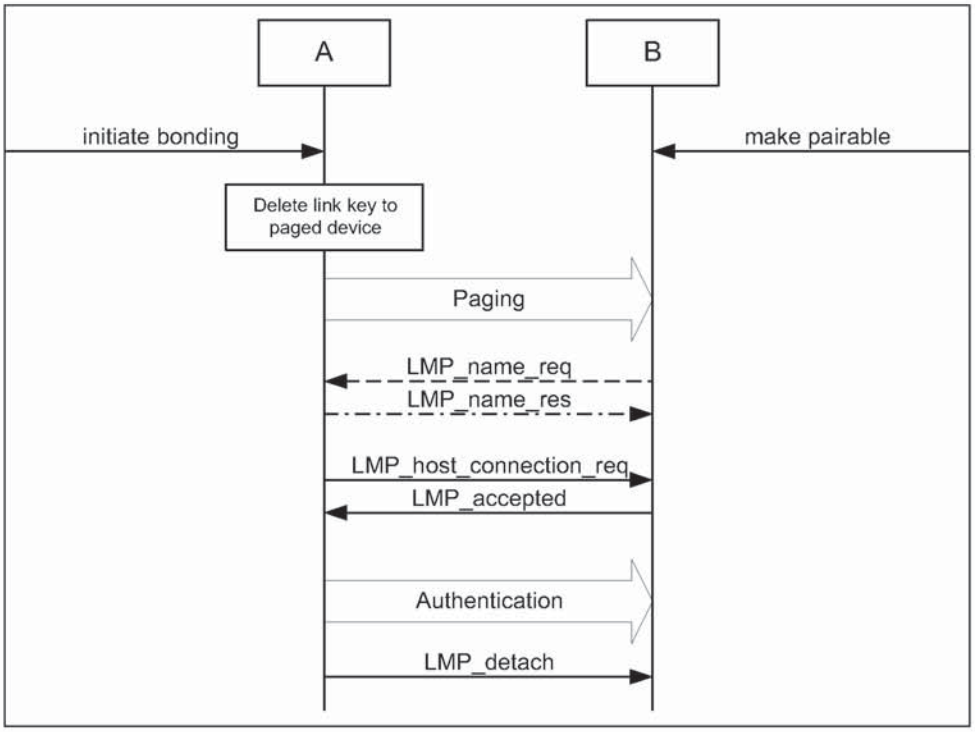


Figure 6.7: Bonding as performed when the purpose of the procedure is only to create and exchange a link key between two Bluetooth devices.

6.5.4 Conditions

Before bonding can be initiated, the initiating device (A) must know the Device Access Code of the device to pair with. This is normally done by first performing device discovery. A Bluetooth Device that can initiate bonding (A) should use limited inquiry, and a Bluetooth Device that accepts bonding (B) should support the limited discoverable mode.

Bonding is in principle the same as link establishment with the conditions:

- The paged device (B) shall be set into pairable mode. The paging device (A) is assumed to allow pairing since it has initiated the bonding procedure.
- The paging device (the initiator of the bonding procedure, A) shall initiate authentication.
- Before initiating the authentication part of the bonding procedure, the paging device should delete any link key corresponding to a previous bonding with the paged device.
- If the paging device does not intend to initiate any higher layer initialization during bonding, it need not send LMP_host_request before initiating authentication.

7 ESTABLISHMENT PROCEDURES

	Procedure	Ref.	Support in A	Support in B
1	Link establishment	7.1	M	M
2	Channel establishment	7.2	O	M
3	Connection establishment	7.3	O	O

Table 7.1: Establishment procedures

The establishment procedures defined here do not include any discovery part. Before establishment procedures are initiated, the information provided during device discovery (in the FHS packet of the inquiry response or in the response to a name request) has to be available in the initiating device. This information is:

- The Bluetooth Device Address (BD_ADDR) from which the Device Access Code is generated;
- The system clock of the remote device;
- The page scan mode used by the remote device.

Additional information provided during device discovery that is useful for making the decision to initiate an establishment procedure is:

- The Class of device;
- The Device name.

7.1 LINK ESTABLISHMENT

7.1.1 Purpose

The purpose of the link establishment procedure is to establish a physical link (of ACL type) between two Bluetooth devices using procedures from [1] and [2].

7.1.2 Term on UI level

'Bluetooth link establishment'

7.1.3 Description

In this sub-section, the paging device (A) is in security mode 3. The paging device cannot during link establishment distinguish if the paged device (B) is in security mode 1 or 2.

7.1.3.1 B in security mode 1 or 2

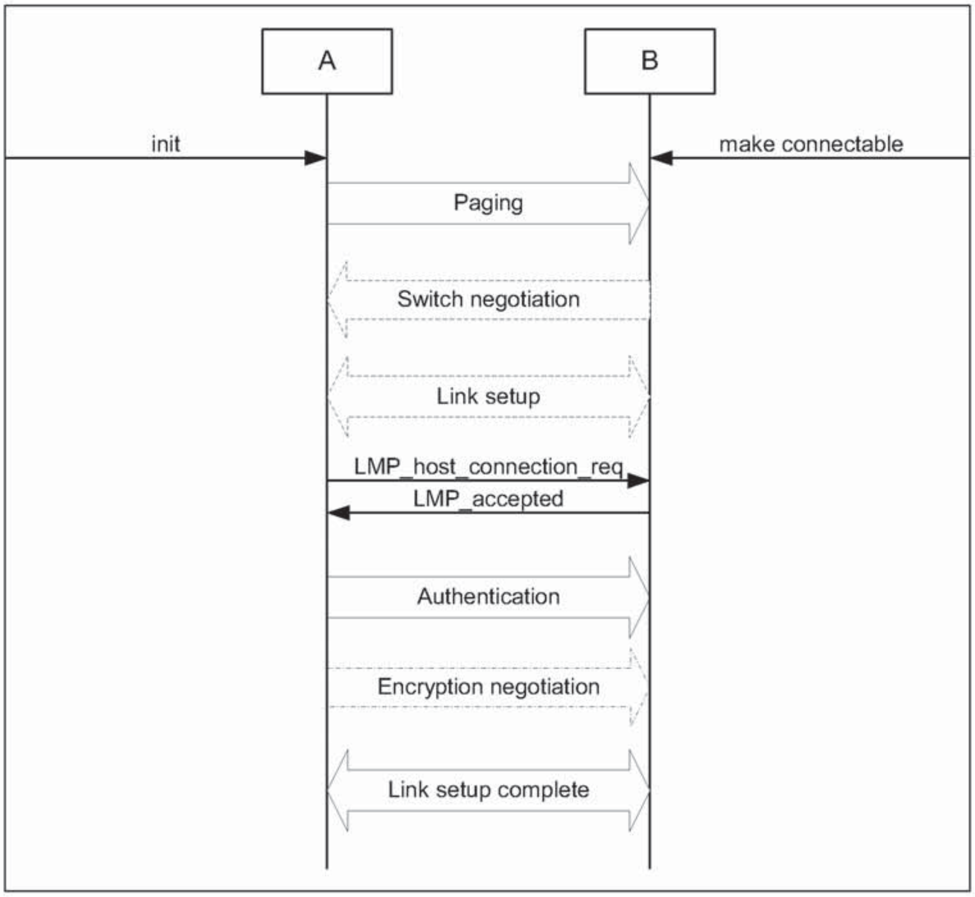


Figure 7.1: Link establishment procedure when the paging device (A) is in security mode 3 and the paged device (B) is in security mode 1 or 2.

7.1.3.2 B in security mode 3

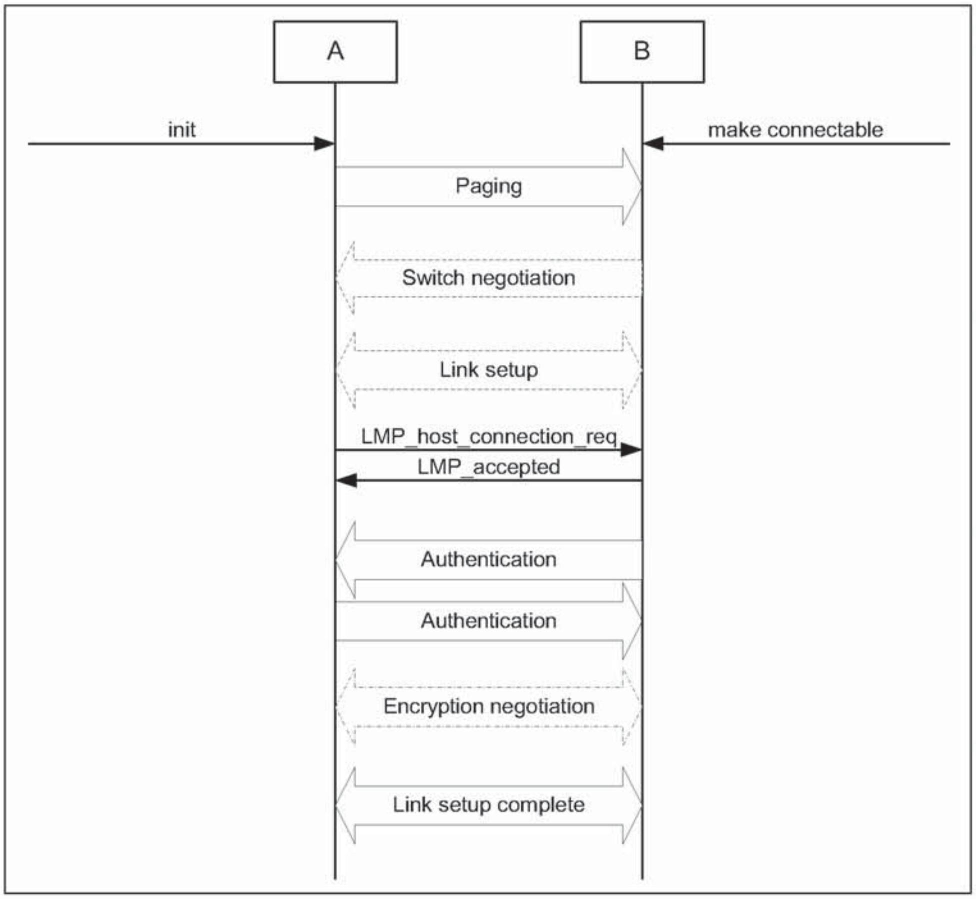


Figure 7.2: Link establishment procedure when both the paging device (A) and the paged device (B) are in security mode 3.

7.1.4 Conditions

The paging procedure shall be according to [1] and the paging device should use the Device access code and page mode received through a previous inquiry. When paging is completed, a physical link between the two Bluetooth devices is established.

If role switching is needed (normally it is the paged device that has an interest in changing the master/slave roles) it should be done as early as possible after the physical link is established. If the paging device does not accept the switch, the paged device has to consider whether to keep the physical link or not.

Both devices may perform link setup (using LMP procedures that require no interaction with the host on the remote side). Optional LMP features can be used after having confirmed (using LMP_feature_req) that the other device supports the feature.

When the paging device needs to go beyond the link setup phase, it issues a request to be connected to the host of the remote device. If the paged device is in security mode 3, this is the trigger for initiating authentication.

The paging device shall send LMP_host_connection_req during link establishment (i.e. before channel establishment) and may initiate authentication only after having sent LMP_host_connection_request.

After an authentication has been performed, any of the devices can initiate encryption.

Further link configuration may take place after the LMP_host_connection_req. When both devices are satisfied, they send LMP_setup_complete.

Link establishment is completed when both devices have sent LMP_setup_complete.

7.2 CHANNEL ESTABLISHMENT

7.2.1 Purpose

The purpose of the channel establishment procedure is to establish a Bluetooth channel (a logical link) between two Bluetooth devices using [3].

7.2.2 Term on UI level

'Bluetooth channel establishment'.

7.2.3 Description

In this sub-section, the initiator (A) is in security mode 3. During channel establishment, the initiator cannot distinguish if the acceptor (B) is in security mode 1 or 3.

7.2.3.1 B in security mode 2

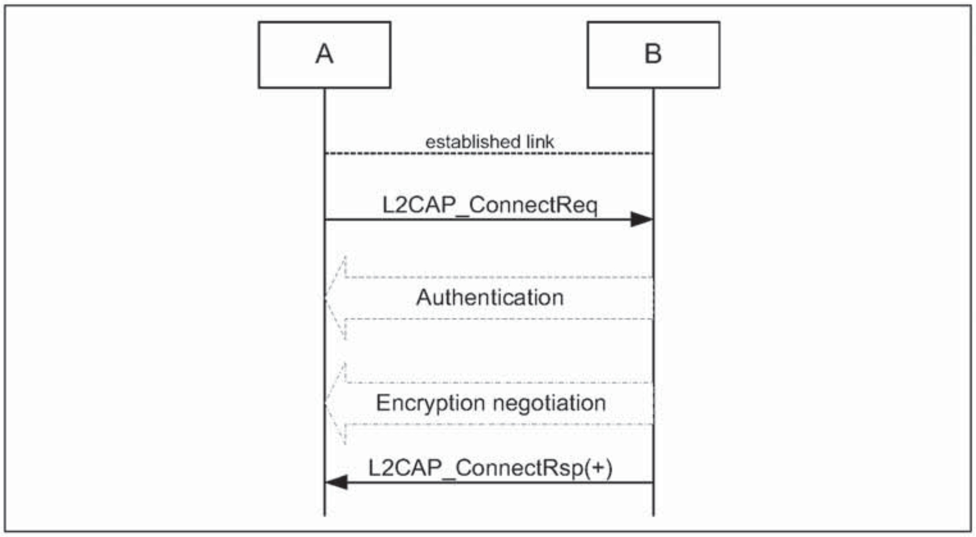


Figure 7.3: Channel establishment procedure when the initiator (A) is in security mode 3 and the acceptor (B) is in security mode 2.

7.2.3.2 B in security mode 1 or 3

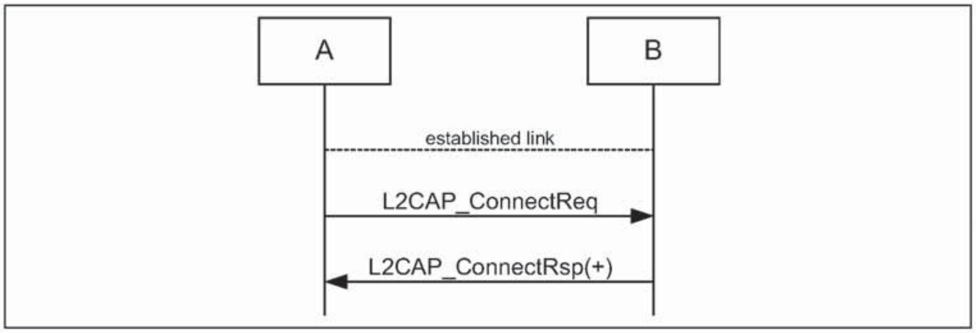


Figure 7.4: Channel establishment procedure when the initiator (A) is in security mode 3 and the acceptor (B) is in security mode 1 or 3.

7.2.4 Conditions

Channel establishment starts after link establishment is completed when the initiator sends a channel establishment request (L2CAP_ConnectReq).

Depending on security mode, security procedures may take place after the channel establishment has been initiated.

Channel establishment is completed when the acceptor responds to the channel establishment request (with a positive L2CAP_ConnectRsp).

7.3 CONNECTION ESTABLISHMENT

7.3.1 Purpose

The purpose of the connection establishment procedure is to establish a connection between applications on two Bluetooth devices.

7.3.2 Term on UI level

'Bluetooth connection establishment'

7.3.3 Description

In this sub-section, the initiator (A) is in security mode 3. During connection establishment, the initiator cannot distinguish if the acceptor (B) is in security mode 1 or 3.

7.3.3.1 B in security mode 2

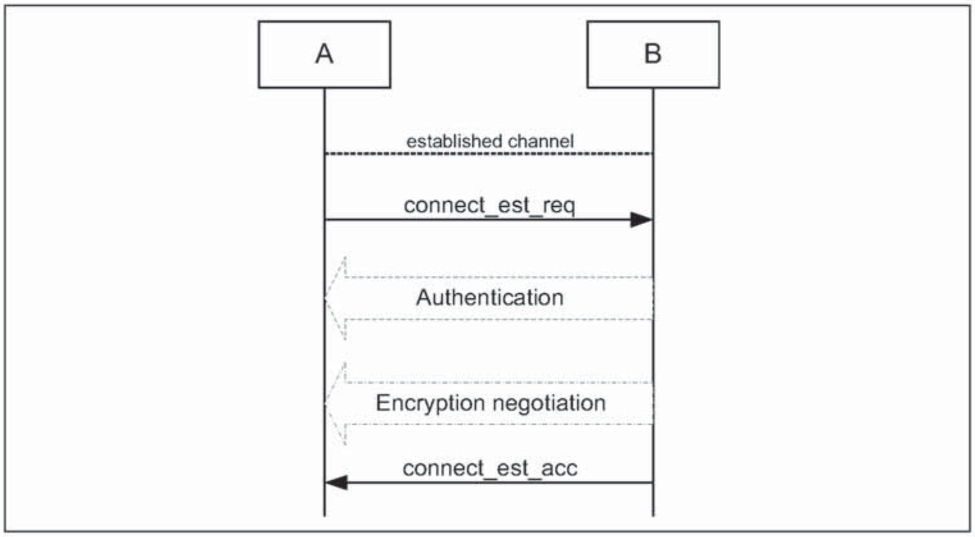


Figure 7.5: Connection establishment procedure when the initiator (A) is in security mode 3 and the acceptor (B) is in security mode 2.

7.3.3.2 B in security mode 1 or 3

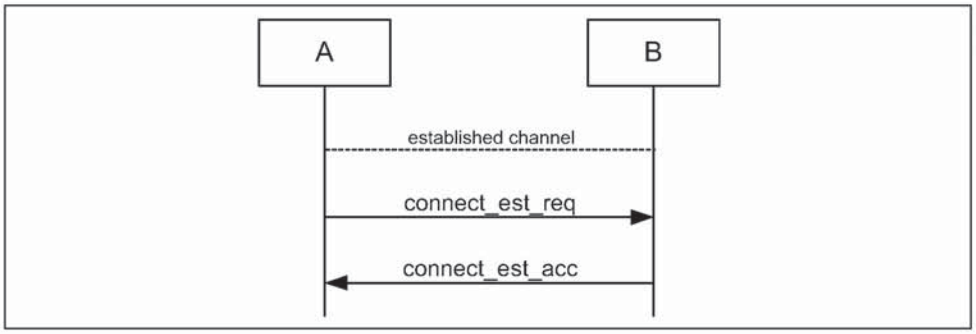


Figure 7.6: Connection establishment procedure when the initiator (A) is in security mode 3 and the acceptor (B) is in security mode 1 or 3.

7.3.4 Conditions

Connection establishment starts after channel establishment is completed, when the initiator sends a connection establishment request ('connect_est_req' is application protocol-dependent). This request may be a TCS SETUP message [5] in the case of a Bluetooth telephony application Cordless Telephony Profile, or initialization of RFCOMM and establishment of DLC [4] in the case of a serial port-based application Serial Port Profile (although neither TCS or RFCOMM use the term 'connection' for this).

Connection establishment is completed when the acceptor accepts the connection establishment request ('connect_est_acc' is application protocol dependent).

7.4 ESTABLISHMENT OF ADDITIONAL CONNECTION

When a Bluetooth device has established one connection with another Bluetooth device, it may be available for establishment of:

- A second connection on the same channel, and/or
- A second channel on the same link, and/or
- A second physical link.

If the new establishment procedure is to be towards the same device, the security part of the establishment depends on the security modes used. If the new establishment procedure is to be towards a new remote device, the device should behave according to active modes independent of the fact that it already has another physical link established (unless allowed co-incident radio and baseband events have to be handled).

8 DEFINITIONS

In the following, terms written with capital letters refer to states.

8.1 GENERAL DEFINITIONS

Mode A set of directives that defines how a device will respond to certain events.

Idle As seen from a remote device, a Bluetooth device is idle, or is in idle mode, when there is no link established between them.

Bond A relation between two Bluetooth devices defined by creating, exchanging and storing a common link key. The bond is created through the bonding or LMP-pairing procedures.

8.2 CONNECTION-RELATED DEFINITIONS

Physical channel A synchronized Bluetooth baseband-compliant RF hopping sequence.

Piconet A set of Bluetooth devices sharing the same physical channel defined by the master parameters (clock and BD_ADDR).

Physical link A Baseband-level connection[1] between two devices established using paging. A physical link comprises a sequence of transmission slots on a physical channel alternating between master and slave transmission slots.

ACL link An asynchronous (packet-switched) connection[1] between two devices created on LMP level. Traffic on an ACL link uses ACL packets to be transmitted.

SCO link A synchronous (circuit-switched) connection[1] for reserved bandwidth communications; e.g. voice between two devices, created on the LMP level by reserving slots periodically on a physical channel. Traffic on an SCO link uses SCO packets to be transmitted. SCO links can be established only after an ACL link has first been established.

Link Shorthand for an ACL link.

PAGE A baseband state where a device transmits page trains, and processes any eventual responses to the page trains.

PAGE_SCAN A baseband state where a device listens for page trains.

[1] The term 'connection' used here is not identical to the definition below. It is used in the absence of a more concise term.

Page The transmission by a device of page trains containing the Device Access Code of the device to which the physical link is requested.

Page scan The listening by a device for page trains containing its own Device Access Code.

Channel A logical connection on L2CAP level between two devices serving a single application or higher layer protocol.

Connection A connection between two peer applications or higher layer protocols mapped onto a channel.

Connecting A phase in the communication between devices when a connection between them is being established. (Connecting phase follows after the link establishment phase is completed.)

Connect (to service) The establishment of a connection to a service. If not already done, this includes establishment of a physical link, link and channel as well.

8.3 DEVICE-RELATED DEFINITIONS

Discoverable device A Bluetooth device in range that will respond to an inquiry (normally in addition to responding to page).

Silent device A Bluetooth device appears as silent to a remote device if it does not respond to inquiries made by the remote device. A device may be silent due to being non-discoverable or due to baseband congestion while being discoverable.

Connectable device A Bluetooth device in range that will respond to a page.

Trusted device A paired device that is explicitly marked as trusted.

Paired device A Bluetooth device with which a link key has been exchanged (either before connection establishment was requested or during connecting phase).

Pre-paired device A Bluetooth device with which a link key was exchanged, and the link key is stored, before link establishment.

Un-paired device A Bluetooth device for which there was no exchanged link key available before connection establishment was request.

Known device A Bluetooth device for which at least the BD_ADDR is stored.

Un-known device A Bluetooth device for which no information (BD_ADDR, link key or other) is stored.

Authenticated device A Bluetooth device whose identity has been verified during the lifetime of the current link, based on the authentication procedure.

8.4 PROCEDURE-RELATED DEFINITIONS

Paging A procedure for establishing a physical link of ACL type on baseband level, consisting of a page action of the initiator and a page scan action of the responding device.

Link establishment A procedure for establishing a link on LMP level. A link is established when both devices have agreed that LMP setup is completed.

Channel establishment A procedure for establishing a channel on L2CAP level.

Connection establishment A procedure for creating a connection mapped onto a channel.

Creation of a trusted relationship A procedure where the remote device is marked as a trusted device. This includes storing a common link key for future authentication and pairing (if the link key is not available).

Creation of a secure connection. A procedure of establishing a connection, including authentication and encryption.

Device discovery A procedure for retrieving the Bluetooth device address, clock, class-of-device field and used page scan mode from discoverable devices.

Name discovery A procedure for retrieving the user-friendly name (the Bluetooth device name) of a connectable device.

Service discovery Procedures for querying and browsing for services offered by or through another Bluetooth device.

8.5 SECURITY-RELATED DEFINITIONS

Authentication A generic procedure based on LMP-authentication if a link key exists or on LMP-pairing if no link key exists.

LMP-authentication An LMP level procedure for verifying the identity of a remote device. The procedure is based on a challenge-response mechanism using a random number, a secret key and the BD_ADDR of the non-initiating device. The secret key used can be a previously exchanged link key or an initialization key created based on a PIN (as used when pairing).

Authorization A procedure where a user of a Bluetooth device grants a specific (remote) Bluetooth device access to a specific service. Authorization implies that the identity of the remote device can be verified through authentication.

Authorize The act of granting a specific Bluetooth device access to a specific service. It may be based upon user confirmation, or given the existence of a trusted relationship.

LMP-pairing A procedure that authenticates two devices, based on a PIN, and subsequently creates a common link key that can be used as a basis for a trusted relationship or a (single) secure connection. The procedure consists of the steps: creation of an initialization key (based on a random number and a PIN), LMP-authentication based on the initialization key and creation of a common link key.

Bonding A dedicated procedure for performing the first authentication, where a common link key is created and stored for future use.

Trusting The marking of a paired device as trusted. Trust marking can be done by the user, or automatically by the device (e.g. when in pairable mode) after a successful pairing.

9 ANNEX A (NORMATIVE): TIMERS AND CONSTANTS

The following timers are required by this profile.

Timer name	Recommended value	Description	Comment
$T_{GAP}(100)$	10.24 s	Normal time span that a Bluetooth device performs inquiry.	Used during inquiry and device discovery.
$T_{GAP}(101)$	10.625 ms	Minimum time in INQUIRY_SCAN.	A discoverable Bluetooth device enters INQUIRY_SCAN for at least $T_{GAP}(101)$ every $T_{GAP}(102)$.
$T_{GAP}(102)$	2.56 s	Maximum time between repeated INQUIRY_SCAN enterings.	Maximum value of the inquiry scan interval, $T_{inquiry\ scan}$.
$T_{GAP}(103)$	30.72 s	A Bluetooth device shall not be in a discoverable mode less than $T_{GAP}(103)$.	Minimum time to be discoverable.
$T_{GAP}(104)$	1 min	A Bluetooth device should not be in limited discoverable mode more than $T_{GAP}(104)$.	Recommended upper limit.

Table 9.1: Defined GAP timers

10 ANNEX B (INFORMATIVE): INFORMATION FLOWS OF RELATED PROCEDURES

10.1 LMP-AUTHENTICATION

The specification of authentication on link level is found in [2].

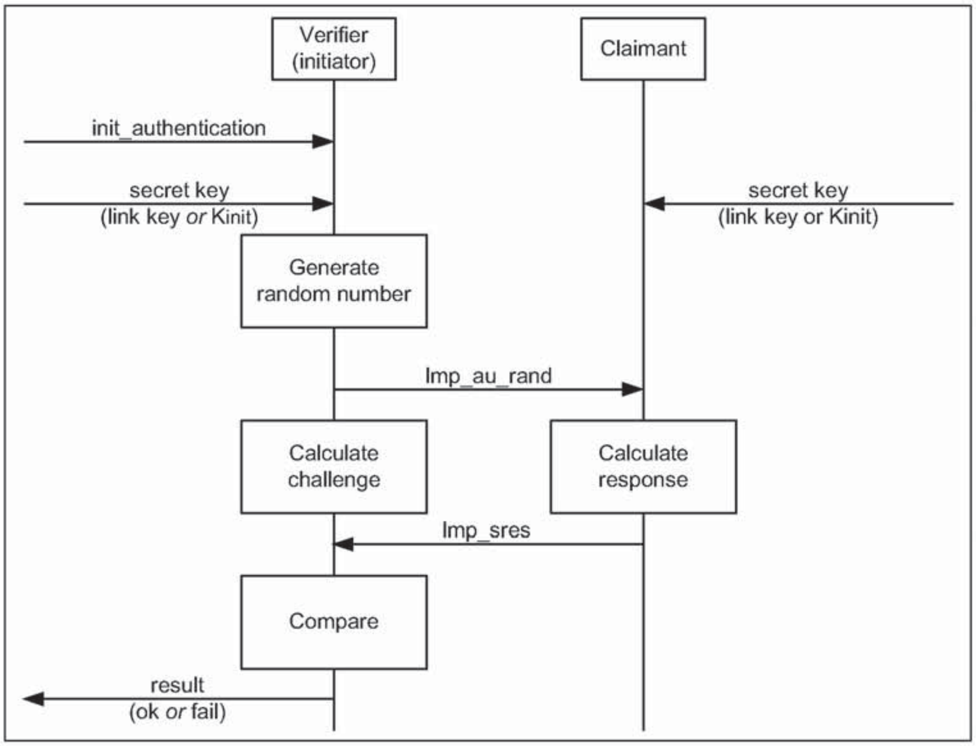


Figure 10.1: LMP-authentication as defined by [2].

The secret key used here may be either an already exchanged link key or an initialization key created in the LMP-pairing procedure.

10.2 LMP-PAIRING

The specification of pairing on link level is found in [2].

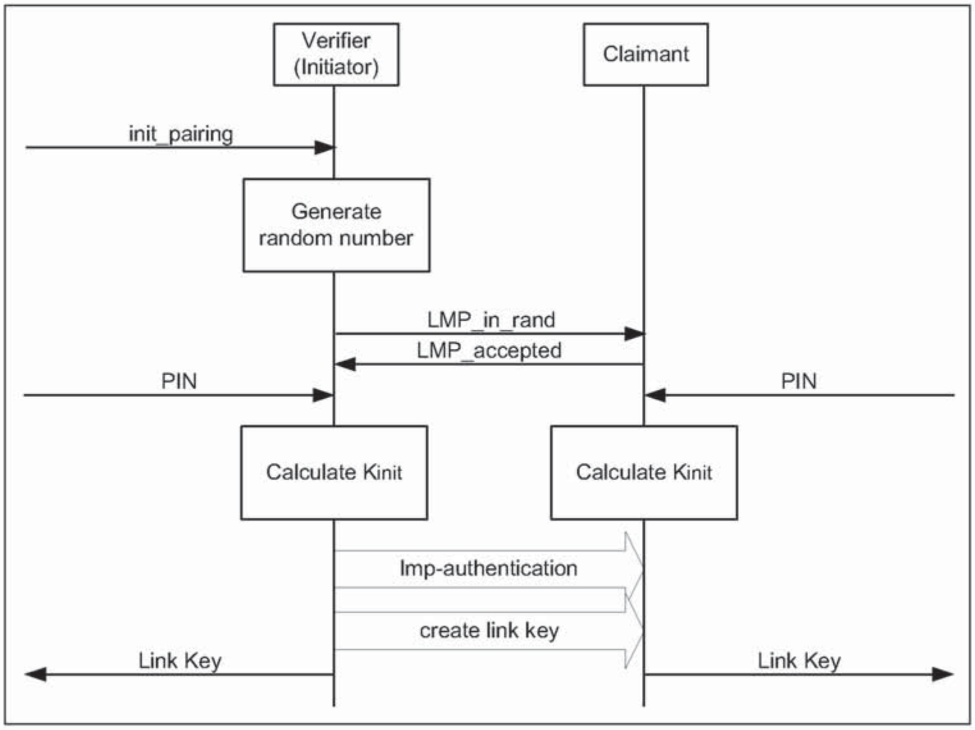


Figure 10.2: LMP-pairing as defined in [2].

The PIN used here is PN_{BB}.

The create link key procedure is described in section 3.3.4 of [2] and section 14.2.2 of [1]. In case the link key is based on a combination key, a mutual authentication takes place and shall be performed irrespective of current security mode.

10.3 SERVICE DISCOVERY

The Service Discovery Protocol [6] specifies what PDUs are used over-the-air to inquire about services and service attributes. The procedures for discovery of supported services and capabilities using the Service Discovery Protocol are described in the Service Discovery Application Profile. This is just an example.

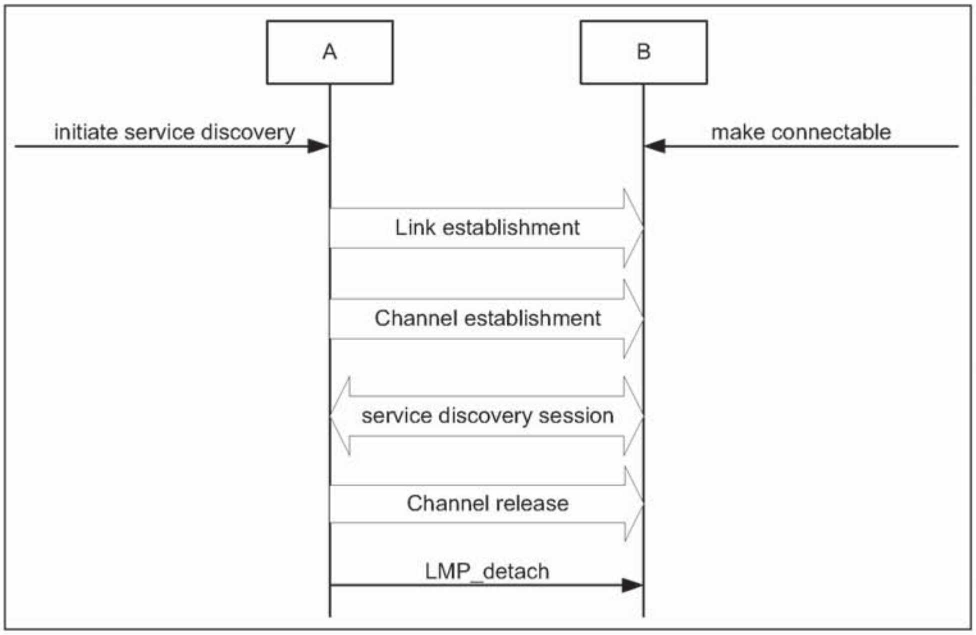


Figure 10.3: Service discovery procedure.

11 REFERENCES

[1] Bluetooth Baseband Specification

[2] Bluetooth Link Manager Protocol

[3] Bluetooth Logical Link Control and Adaptation Protocol

[4] Bluetooth RFCOMM

[5] Bluetooth Telephony Control Specification

[6] Bluetooth Service Discovery Protocol

[7] Bluetooth Service Discovery Application Profile

[8] Bluetooth Cordless Telephony Profile

[9] Bluetooth Serial Port Profile

[10] Bluetooth Security Architecture (white paper)

[11] Bluetooth Assigned Numbers

Part K:2

SERVICE DISCOVERY APPLICATION PROFILE

This document defines the features and procedures for an application in a Bluetooth device to discover services registered in other Bluetooth devices and retrieve any desired available information pertinent to these services.

CONTENTS

FOREWORD

Interoperability between devices from different manufacturers is provided for a specific service and use case, if the devices conform to a Bluetooth SIG-defined profile specification. A profile defines a selection of messages and procedures (generally termed *capabilities*) from the Bluetooth SIG specifications, and gives an unambiguous description of the air interface for specified service(s) and use case(s).

All defined features are process-mandatory. This means that, if a feature is used, it is used in a specified manner. Whether the provision of a feature is mandatory or optional is stated separately for both sides of the Bluetooth air interface.

1 INTRODUCTION

1.1 SCOPE

It is expected that the number of services that can be provided over Bluetooth links will increase in an undetermined (and possibly uncontrolled) manner. Therefore, procedures need to be established to aid a user of a Bluetooth-enabled device to sort the ever-increasing variety of services that will become available to him/her. While many of the Bluetooth-enabled services that may be encountered are currently unknown, a standardized procedure can still be put into place on how to locate and identify them.

The Bluetooth protocol stack contains a Service Discovery Protocol (SDP) BT_SDP_spec:[7] that is used to locate services that are available on or via devices in the vicinity of a Bluetooth enabled device. Having located what services are available in a device, a user may then select to use one or more of them. Selecting, accessing, and using a service is outside the scope of this document. Yet, even though SDP is not directly involved in accessing services, information retrieved via SDP facilitates service access by using it to properly condition the local Bluetooth stack to access the desired service.

The service discovery profile defines the protocols and procedures that shall be used by a service discovery application on a device to locate services in other Bluetooth-enabled devices using the Bluetooth Service Discovery Protocol (SDP). With regard to this profile, the service discovery application is a specific user-initiated application. In this aspect, this profile is in contrast to other profiles where service discovery interactions between two SDP entities in two Bluetooth-enabled devices result from the need to enable a particular transport service (e.g. RFCOMM, etc.), or a particular usage scenario (e.g. file transfer, cordless telephony, LAN AP, etc.) over these two devices. Service discovery interactions of the latter kind can be found within the appropriate Bluetooth usage scenario profile documents.

The service discovery in the other profile documents has a very narrow scope; e.g. learning about the protocols and related protocol parameters needed for accessing a particular service. Nevertheless, the fundamentals of the service discovery procedures covered in this profile document, and the use of the Bluetooth protocols in support of these procedures can be replicated in other profile documents as well. The only difference is that for the other profiles these procedures are initiated by application-level actions within the applications described by the corresponding profiles, as opposed to user-level actions for this profile.

SDP provides direct support for the following set of service inquiries:

- Search for services by service class;
- Search for services by service attributes; and
- Service browsing.

The generic service discovery application considered for this profile also covers the above service inquiry scenarios.

The former two cases represent searching for known and specific services. They provide answers to user questions like: “Is service A, or is service A with characteristics B and C, available?” The latter case represents a general service search and provides answers to questions like: “What services are available?” or “What services of type A are available?”

The above service inquiry scenarios can be realized two-fold:

- By performing the service searches on a particular device that a user ‘consciously’ has already connected to, and/or
- By performing the service searches by ‘unconsciously’ connecting to devices discovered in a device's vicinity.

Both of the above approaches require that devices need first to be discovered, then linked with, and then inquired about the services they support.

1.2 SYMBOLS AND CONVENTIONS

This profile uses the symbols and conventions specified in Section 1.2 of the Generic Access Profile [3].

2 PROFILE OVERVIEW

2.1 PROFILE STACK

Figure 2.1 shows the Bluetooth protocols and supporting entities involved in this profile.

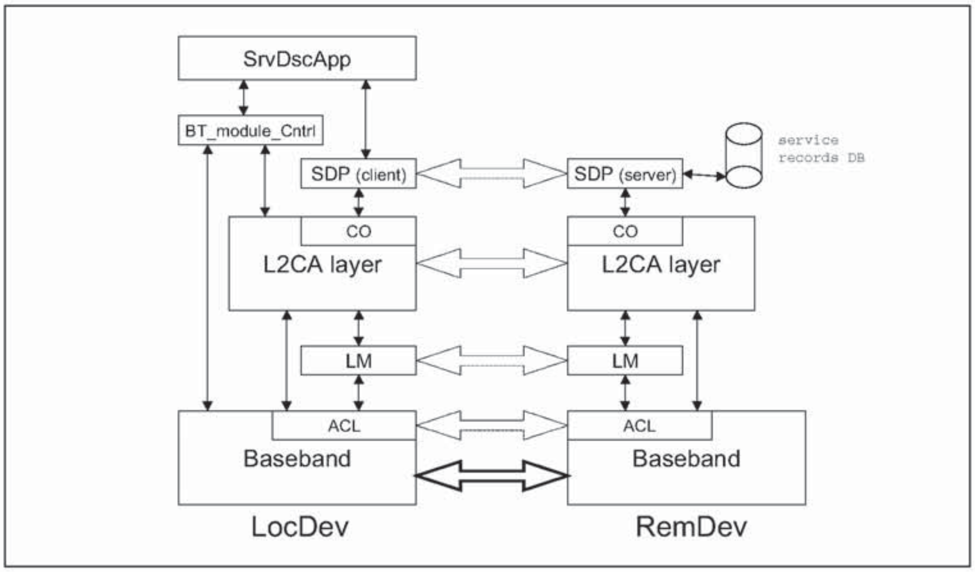


Figure 2.1: The Bluetooth protocol stack for the service discovery profile

The service discovery user application (SrvDscApp) in a local device (LocDev) interfaces with the Bluetooth SDP client to send service inquiries and receive service inquiry responses from the SDP servers of remote devices (RemDevs) BT_SDP_spec:[7]. SDP uses the connection-oriented (CO) transport service in L2CAP, which in turn uses the baseband asynchronous connectionless (ACL) links to ultimately carry the SDP PDUs over the air.

Service discovery is tightly related to discovering devices, and discovering devices is tightly related to performing inquiries and pages. Thus, the SrvDscApp interfaces with the baseband via the BT_module_Cntrl entity that instructs the Bluetooth module when to enter various search modes of operation.[1]

[1] The BT_module_Cntrl may be part of a Bluetooth stack implementation (and thus be shared by many Bluetooth-aware applications) or a 'lower part' of the SrvDscApp. Since, no assumptions about any particular stack or SrvDscApp implementations are made, the BT_module_Cntrl entity represents a logical entity separate from the SrvDscApp, which may or may not be part of the SrvDscApp itself, a stack component, or any other appropriate piece of code.

The service records database (DB) shown in Figure 2.1 next to an SDP server is a logical entity that serves as a repository of service discovery-related information. The 'physical form' of this database is an implementation issue outside the scope of this profile.

2.2 CONFIGURATIONS AND ROLES

The following roles are defined in this profile:

- **Local device (LocDev):** A LocDev is the device that initiates the service discovery procedure. A LocDev must contain at least the *client* portion of the Bluetooth SDP architecture BT_SDP_spec:[7]. A LocDev contains the service discovery application (SrvDscApp) used by a user to initiate discoveries and display the results of these discoveries.
- **Remote Device(s) (RemDev(s)):** A RemDev is any device that participates in the service discovery process by responding to the service inquiries generated by a LocDev. A RemDev must contain at least the *server* portion of the Bluetooth SDP architecture BT_SDP_spec:[7]. A RemDev contains a service records database, which the server portion of SDP consults to create responses to service discovery requests.

The LocDev or RemDev role assigned to a device is neither permanent nor exclusive. A RemDev may also have a SrvDscApp installed into it as well as an SDP client, and a LocDev may also have an SDP server. In conjuction with which device has an SrvDscApp installed, an SDP-client installed, and an SDP-server installed, the assignment of devices to the above roles is relative to each individual SDP (and related) transaction and which device initiates the transaction. Thus, a device could be a LocDev for a particular SDP transaction, while at the very same time be a RemDev for another SDP transaction.

With respect to this profile, a device without a UI (directly or indirectly available) for entering user input and returning the results of service searches is not considered as a candidate for a LocDev. Nevertheless, even if such a device is not considered as a candidate for a LocDev, the procedures presented in the following sections can still apply if applications running in such a device need to execute a service discovery transaction.

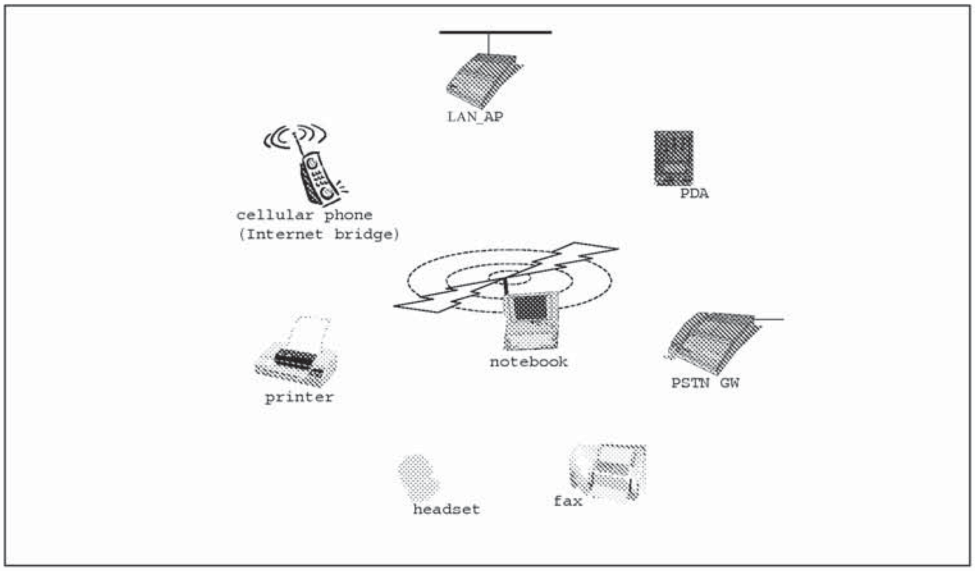


Figure 2.2: A typical service discovery scenario

The figure above shows a local device (the notebook) inquiring for services among a plethora of remote devices.

2.3 USER REQUIREMENTS AND SCENARIOS

The scenarios covered by this profile are the following:

- Search for services by service class,
- Search for services by service attributes, and
- Service browsing.

The first two cases represent searching for known and specific services, as part of the user question “Is service A, or is service A with characteristics B and C, available?” The latter case represents a general service search that is a response to the user question “What services are available?”

This profile implies the presence of a Bluetooth-aware, user-level application, the SrvDscApp, in a LocDev that interfaces with the SDP protocol for locating services. In this aspect, this profile is unique as compared to other profiles. It is a profile that describes an application that interfaces to a specific Bluetooth protocol to take full advantage of it for the direct benefit of an end-user.

2.4 PROFILE FUNDAMENTALS

Before any two Bluetooth-equipped devices can communicate with each other the following may be needed:

- The devices need to be powered-on and initialized. Initialization may require providing a PIN for the creation of a link key, for device authorization and data encryption.
- A Bluetooth link has to be created, which may require the discovery of the other device's BD_ADDR via an inquiry process, and the paging of the other device.

While it may be seem natural to consider a LocDev serving as a Bluetooth master and the RemDev(s) serving as Bluetooth slave(s), there is no such requirement imposed on the devices participating in this profile. Service discovery as presented in this document can be initiated by either a master or a slave device at any point for which these devices are members of the same piconet. Also, a slave in a piconet may possibly initiate service discovery in a new piconet, provided that it notifies the master of the original piconet that it will be unavailable (possibly entering the hold operational mode) for a given amount of time.[2]

The profile does not require the use of authentication and/or encryption. If any of these procedures are used by any of the devices involved, service discovery will be performed only on the subset of devices that pass the authentication and encryption security ‘roadblocks’ that may impose to each other. In other words, any security restrictions for SDP transactions are dictated by the security restrictions already in place (if any) on the Bluetooth link.

2.5 CONFORMANCE

If conformance to this profile is claimed, all capabilities indicated mandatory for this profile shall be supported in the specified manner (process-mandatory). This also applies to all optional and conditional capabilities for which support is indicated. All mandatory capabilities, and optional and conditional capabilities for which support is indicated, are subject to verification as part of the Bluetooth certification program.

[2] Recall that a master of a piconet cannot initiate a new piconet. Since a piconet is ultimately identified by the BD_ADDR and the Bluetooth clock of its master, the latter piconet will be identical to and indistinguishable from the former.

3 USER INTERFACE ASPECTS

3.1 PAIRING

No particular requirements regarding pairing are imposed by this profile. Pairing may or may not be performed. Whenever a LocDev performs service discovery against as yet 'unconnected' RemDev(s), it shall be the responsibility of the SrvDscApp to allow pairing prior to connection, or to by-pass any devices that may require pairing first. This profile is focused on only performing service discovery whenever the LocDev can establish a legitimate and useful baseband link[3] with RemDev(s).

3.2 MODE SELECTION

This profile assumes that, under the guidance of the SrvDscApp, the LocDev shall be able to enter the inquiry and/or page states. It is also assumed that a RemDev with services that it wants to make available to other devices (e.g. printer, a LAN DAP, a PSTN gateway, etc.) shall be able to enter the inquiry scan and/or page scan states. For more information about the inquiry and page related states see Section 8.

Since the SrvDscApp may also perform service inquiries against already connected RemDevs, it is not mandatory according to the profile that a LocDev always be the master of a connection with a RemDev. Similarly, a RemDev may not always be the slave of a connection with a LocDev.

[3] A legitimate and useful baseband link is a Bluetooth baseband link that is properly authenticated and encrypted (if so desired), whenever any of these options are activated by any of the devices participating in this profile.

4 APPLICATION LAYER

4.1 THE SERVICE DISCOVERY APPLICATION

In this subsection, the operational framework of the SrvDscApp is presented.[4] Figure 4.1 shows alternative possibilities for a SrvDscApp.

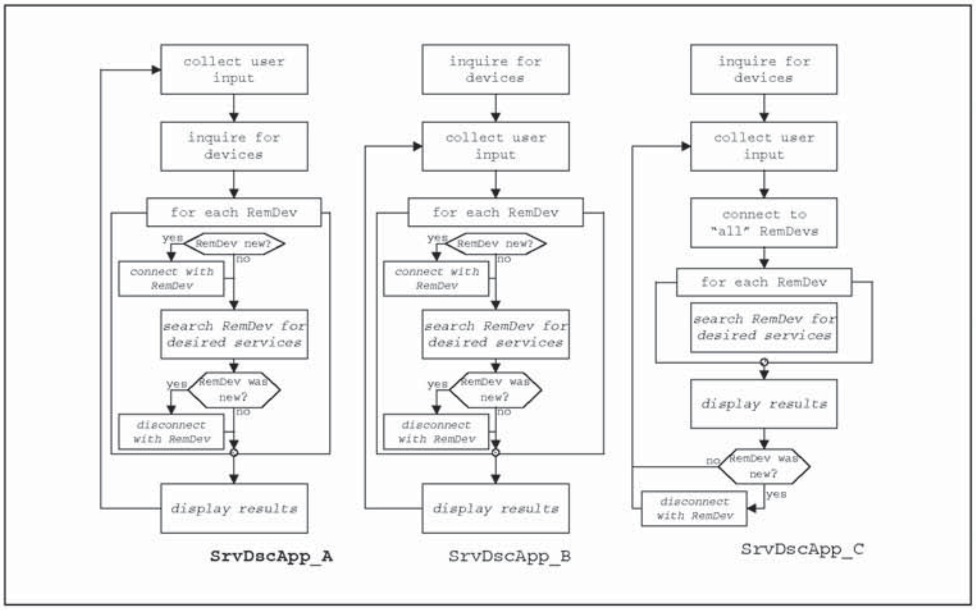


Figure 4.1: Three possible SrvDscApps

The SrvDscApp alternatives shown in Figure 4.1, which are not exhaustive by any means, achieve the same objectives but they follow different paths for achieving them. In the first alternative (SrvDscApp_A), the SrvDscApp on a LocDev inquires its user to provide information for the desired service search. Following this, the SrvDscApp searches for devices, via the Bluetooth inquiry procedure. For each device found, the LocDev will connect to it, perform any necessary link set-up, see related procedures in Generic Access Profile [3], and then inquire it for the desired services. In the second alternative (SrvDscApp_ B), the inquiry of devices is done prior to collecting user input for the service search.[5]

[4] This profile does not dictate any particular implementation for a SevDisApp. It only presents the procedures needed to achieve its objectives.

[5] Device inquiries may even occur by means outside the scope of a particular SrvDscApp implementation. But, since such other means are not guaranteed to exist, it is recommended that the SrvDscApp activates device inquiries too.

In the first two alternatives, page, link creation, and service discovery are done sequentially on a per RemDev basis; i.e., the LocDev does not page any new RemDev prior to completing the service search with a previous RemDev and (if necessary) disconnecting from it. In the last alternative (SrvDscApp_C), the LocDev, under the control of the SrvDscApp, will first page all RemDevs, then will create links with all of these devices (up to a maximum of 7 at a time), and then inquire all the connected devices for the desired services.

Just as an example, we focus on a SrvDscApp similar to the one represented by the SrvDscApp_A in Figure 4.1. In summary, SrvDscApp (for ease of notation, the suffix '_A' has been dropped) has the following features:

- The SrvDscApp activates Bluetooth inquiries following a user request for a service search,
- For any new RemDev found following an inquiry, the SrvDscApp will finish service discovery and terminate its link against this device prior to attempting to connect to the next RemDev,
- For any RemDev already connected, the LocDev does not disconnect following service discovery, and
- The user of the SrvDscApp has the option of a trusted and untrusted mode of operation, whereby the SrvDscApp permits connections –
    a) only with trusted RemDev, or
    b) with any of the devices above plus any newly discovered RemDevs that require nothing more beyond possibly pairing with the default all-zero PIN, or
    c) with any of the devices above, plus any additional RemDev for which the user explicitly enters a non-zero PIN.

The above options have to do with the degree of user involvement in configuring and interacting with the SrvDscApp and setting the security levels that the user is willing to accept for the service searches. When selecting options (a) or (b), then for the devices with which no legitimate connections can be established, it is assumed that the SrvDscApp ignores them without any cue to its user (however, this too is an implementation issue).

When a LocDev performs a service discovery search, it does so against three different types of RemDevs:

1. *trusted devices:* These are devices that are currently not connected with the LocDev but the LocDev device has already an established trusted relation with.
2. *unknown (new) devices:* These are untrusted devices that are currently not connected with the LocDev.
3. *connected devices:* These are devices that are already connected to the LocDev.

To discover type 1 or 2 RemDevs, the SrvDscApp needs to activate the Bluetooth inquiry and/or page processes. For type 3 RemDevs, the latter processes are needed. To perform its task, SrvDscApp needs to have access to the BD_ADDR of the devices in the vicinity of a LocDev, no matter whether these devices have been located via a Bluetooth inquiry process or are already connected to the LocDev. Thus, BT_module_Cntr in a LocDev shall maintain the list of devices in the vicinity of the LocDev and shall avail this list to the SrvDscApp.

4.2 SERVICE PRIMITIVES ABSTRACTIONS

This section briefly describes the functionality of a SrvDscApp. This functionality is presented in the form of service primitive abstractions that provide a formal framework for describing the user expectations from a SrvDscApp. It is assumed that the underlying Bluetooth stack can meet the objectives of these service primitive abstractions directly or indirectly.[6] The exact syntax and semantics of the service primitive abstractions (or simply "service primitives") may be platform-dependent (e.g. an operating system, a hardware platform, like a PDA, a notebook computer, a cellular phone, etc.) and are beyond the scope of this profile. However, the functionality of these primitives is expected to be available to the SrvDscApp to accomplish its task.

Table 4.1 contains a minimum set of enabling service primitives to support a SrvDscApp. Low-level primitives like **openSearch**(.) or **closeSearch**(.) are not shown and are assumed to be part of the implementation of the primitives shown whenever necessary. Different implementations of the Bluetooth stack shall (at a minimum) enable the functions that these service primitives provide. For example, the **serviceSearch**(.) service primitive permits multiple identical operations to be handled at once. A stack implementation that requires an application to accomplish this function by iterating through the multiple identical operations one-at-a-time will be considered as enabling the function of this service primitive.[7] The service primitives shown next relate only to service primitives whose invocation result or relate to an over-the-air data exchange using the Bluetooth protocols. Additional service primitives can be envisioned relating to purely local operations like *service registration*, but these primitives are outside the scope of this profile.

[6] These service primitive abstractions do *not* represent programming interfaces, even though they may be related to them. The word 'directly' is used to describe the possibility that the described function is the result of a single appropriate call of the underlying Bluetooth stack implementation. The word 'indirectly' is used to describe the possibility that the described function can be achieved by combining the results from multiple appropriate calls of the underlying Bluetooth stack implementation.

[7] Even though the service primitives presented in this profile are assumed to act upon a local device for accessing *physically* remote devices, they are general enough to apply in cases where the 'remote device' characterization is only a logical concept; i.e. inquired service records and service providers are located within the same device that invokes these primitives. This general situation is outside the scope of this profile.

service primitive abstraction	resulted action
serviceBrowse (LIST(*RemDev*) LIST(*RemDevRelation*) LIST(*browseGroup*) *getRemDevName* *stopRule*)	a search for services (service browsing) that belong to the list of *browseGroup* services in the devices in the list of RemDevs; the search may be further qualified with a list of *RemDevRelation* parameters, whereby a user specifies the trust and connection relation of the devices to be searched; e.g. search only the devices that are in the *RemDev* list for which there is a trust relation already established; when the *getRemDevName* parameter is set to "yes," the names of the devices supporting the requested services are also returned; the search continues until the stopping rule *stopRule* is satisfied
serviceSearch (LIST(*RemDev*) LIST(*RemDevRelation*) LIST(*searchPattern*, *attributeList*) *getRemDevName* *stopRule*)	a search whether the devices listed in the list of RemDevs support services in the requested list of services; each service in the list must have a service search pattern that is a superset of the *searchPattern*; for each such service the values of the attributes contained in the corresponding *attributeList* are also retrieved; the search may be further qualified with a list of *RemDevRelation* parameters, whereby a user specifies the trust and connection relation of the devices to be searched (e.g. search only the devices that are in the *RemDev* list for which there is a trust relation already established); when the *getRemDevName* parameter is set to "yes," the names of the devices supporting the requested services are also returned; the search continues until the stopping rule *stopRule* is satisfied
enumerateRemDev (LIST(*classOfDevice*) *stopRule*)	a search for RemDev in the vicinity of a LocDev; RemDev searches may optionally be filtered using the list of *classOfDevice (*e.g. LAN APs); the search continues until the stopping rule *stopRule* is satisfied
terminatePrimitive (*primitiveHandle* *returnResults*)	a termination the actions executed as a result of invoking the services primitive identified by the *primitiveHandle*;[*] optionally, this service primitive may return any partially accumulated results related to the terminated service primitive

Table 4.1: Service primitives in support of SrvDscApp

[*] It is assumed that each invocation of a service primitive can be identified by a *primitiveHandle,* the realization of which is implementation-dependent.

The *stopRule* parameter is used to guarantee a graceful termination of a service search. It could represent the number of search items found, or the duration of search, or both. A Bluetooth stack implementation may not expose this parameter, in which case it should provide guarantees that all searches terminate within a reasonable amount of time, for example, say, 120sec.

The **enumerateRemDev**(.) service primitive is directly related to the inquiry mode of operation for the baseband. It also relates to the collection of RemDev that a LocDev is currently connected with. This service is exported to the SrvDscApp via the BT_module_Cntr, see Figure 2.1. The interface between BT_module_Cntr and baseband is for activating Bluetooth inquiries and collecting the results of these inquiries. The interface between the BT_module_Cntrl and (an) L2CAP (implementation) is for keeping track of the RemDev that currently are connected to the LocDev.

The result of the **enumerateRemDev**(.) service primitive can be used with the **serviceSearch**(.) to search for desired services in the devices found. Once again, based on the implementation of the Bluetooth stack, this service primitive may not be provided explicitly, but its service may be provided within other service primitives; e.g. the **serviceSearch**(.).

Missing primitive parameters shall be interpreted (whenever appropriate) as a general service search on the remaining parameters. For example, if the LIST(*RemDev*) parameter is missing from the **serviceSearch**(.), it means that the search shall be performed against any device found in the vicinity of a LocDev. In this case, the first two service primitives may be combined to a single one.

The above service primitives return the requested information, whenever found. Based on the way that these service primitives are supported by a Bluetooth stack implementation, the results of a search may directly return by the corresponding calling function, or a pointer to a data structure may be returned that contains all the relevant information. Alternatively, a Bluetooth stack implementation may have altogether different means for providing the results of a search.

4.3 MESSAGE SEQUENCE CHARTS (MSCS)

This profile is concerned with three distinct Bluetooth procedures. Device discovery, device name discovery, service discovery. Note that each one of these procedures does not preclude any other; e.g. to connect to a RemDev, a LocDev may have to first discover it, and it may also ask for its name. The MSCs relating to the first two procedures (i.e., device and name discovery) are provided in section 2 of LM/HCI_MSCs:[6]. Sections 3, 4.1 and 4.2 of LM/HCI_MSCs:[6] provide the MSCs relating to the third procedure (i.e., service discovery). See also section 4 of BT_LM_spec:[4]. The first two procedures do not require host intervention, while the third does.

Figure 4.2 summarizes the key message exchange 'phases' encountered during the execution of this profile. Not all procedures are present at all times, and not all devices need to go through these procedures all the time. For example, if authentication is not required, the authentication phase in the figure will not be executed. If the SrvDsvApp needs to inquire for services on a specific RemDev with which the LocDev is currently connected, inquiries and pages may not be executed. In the figure, the conditions under which particular phases are executed or not are also provided.

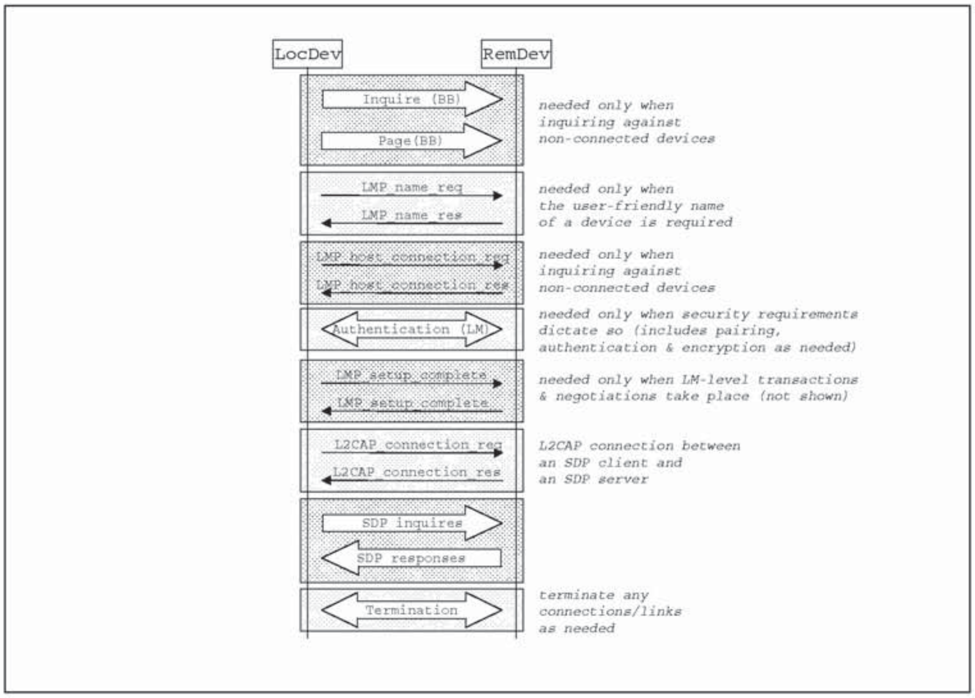

LocDev
RemDev
Inquire (BB)
Page(BB)
needed only when inquiring against non-connected devices
LMP name req
LMP name res
needed only when the user-friendly name of a device is required
LMP host connection req
LMP host connection res
needed only when inquiring against non-connected devices
Authentication (LM)
needed only when security requirements dictate so (includes pairing, authentication & encryption as needed)
LMP setup complete
LMP setup complete
needed only when LM-level transactions & negotiations take place (not shown)
L2CAP connection req
L2CAP connection res
L2CAP connection between an SDP client and an SDP server
SDP inquires
SDP responses
Termination
terminate any connections/links as needed

Figure 4.2: Bluetooth processes in support of this profile

In addition to the MSC in Figure 4.2, Annex A shows what Bluetooth procedures and PDUs are needed to support the service primitives presented in Section 4.2.

5 SERVICE DISCOVERY

The service discovery application does not make use of SDP as a means of accessing a service, but rather as a means of informing the user of a LocDev about the services that are available to his/her device by (and possibly via) RemDev(s). BT-aware applications running in a local device can also use the procedures described in this and the following sections to retrieve any pertinent information that will facilitate the application in accessing a desired service in a remote device.

Table 5.1 shows the SDP feature requirements in a LocDev and in a RemDev.

	SDP feature	Support in LocDev	Support in RemDev
1.	SDP client	M	O
2.	SDP server	O	M

Table 5.1: SDP feature requirements

Table 5.2 shows the SDP PDUs can be exchanged between devices following this profile.

SDP PDUs	Ability to Send		Ability to Receive	
	LocDev	RemDev	LocDev	RemDev
SDP_ErrorResponce	C1	M	M	C1
SDP_ServiceSearchRequest	M	C1	C1	M
SDP_ServiceSearchResponse	C1	M	M	C1
SDP_ServiceAttributeRequest	M	C1	C1	M
SDP_ServiceAttributeResponse	C1	M	M	C1
SDP_ServiceSearchAttributeRequest	M	C1	C1	M
SDP_ServiceSearchAttributeResponse	C1	M	M	C1
Comments:				
[C1]: With regard to this current profile, these PDU transmissions will not occur. Nevertheless, since a device could act as a LocDev on some occasions and as a RemDev on others, these PDU transmission may still take place between these devices.				

Table 5.2: Allowed SDP PDUs

5.1 AN SDP PDU EXCHANGE EXAMPLE

Figure 5.1 shows two examples of SDP PDU exchanges. In particular, it shows PDU exchange sequences for the inquiry and retrieval of any information pertinent to a particular Bluetooth profile.

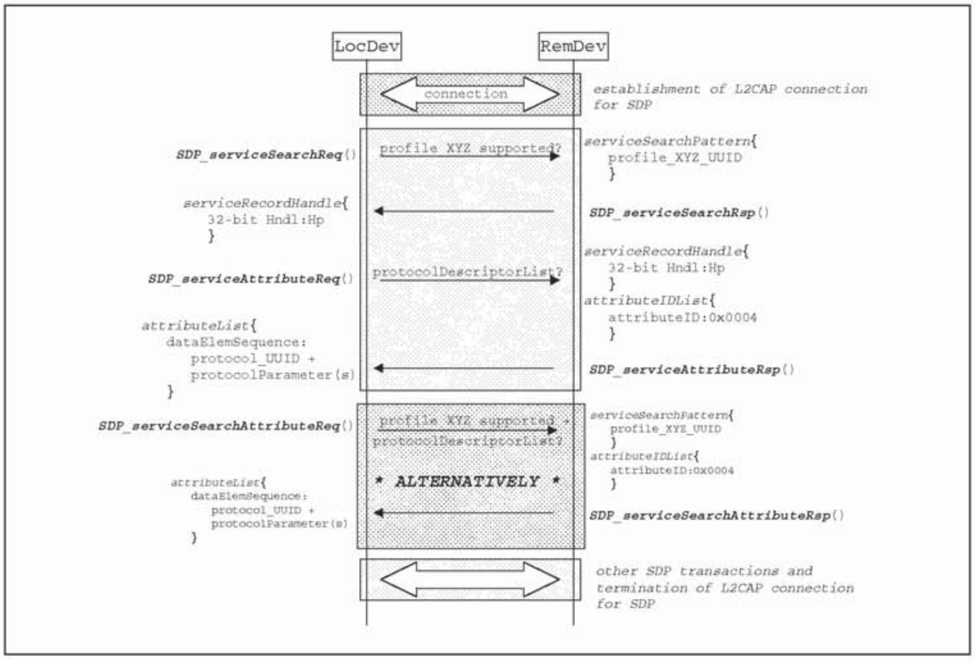

LocDev
RemDev
connection
establishment of L2CAP connection for SDP
SDP_serviceSearchReq()
profile XYZ supported?
serviceSearchPattern{ profile_XYZ_UUID }
serviceRecordHandle{ 32-bit Hndl:Hp }
SDP_serviceSearchRsp()
serviceRecordHandle{ 32-bit Hndl:Hp }
SDP_serviceAttributeReq()
protocolDescriptorList?
attributeIDList{ attributeID:0x0004 }
attributeList{ dataElemSequence: protocol_UUID + protocolParameter(s) }
SDP_serviceAttributeRsp()
SDP_serviceSearchAttributeReq()
profile XYZ supported + protocolDescriptorList?
serviceSearchPattern{ profile_XYZ_UUID }
attributeIDList{ attributeID:0x0004 }
* ALTERNATIVELY *
attributeList{ dataElemSequence: protocol_UUID + protocolParameter(s) }
SDP_serviceSearchAttributeRsp()
other SDP transactions and termination of L2CAP connection for SDP

Figure 5.1: SDP PDU exchange examples for retrieving protocolDescriptorLists

For each PDU sent, the figure shows which device sends it (shown on the starting side of an arrow) and any relative information that this PDU carries (shown on the ending side of an arrow). Note that the LocDev sends request PDUs, while the RemDev sends back response PDUs.

Two alternatives are shown utilizing different SDP PDUs to ultimately retrieve the same information – the *protocolDescriptorList* attribute from devices that support a specific Bluetooth profile. With the first alternative, the desired information is derived in two steps.

- The LocDev sends an *SDP_serviceSearchReq* PDU which contains a service search pattern composed of the UUID associated with the desired profile; see section 4.3 of BT_ASN:[2]. The desired profile (profile 'XYZ') is identified by its UUID, denoted in the figure as 'profile_XYZ_UUID.' In its response PDU, the SDP server returns one or more 32-bit service record handles whose corresponding service records contain the 'profile_XYZ_UUID' UUID. In the figure, only one such handle is shown, denoted as 'prHndl'.
- The LocDev then enters prHndl in an *SDP*_serviceAttribute PDU together with one or more attribute IDs. In this example, the attribute of interest is the

protocolDescriptorList, whose attribute ID is 0x0004. The SDP server then, in its response, returns the requested protocol list.

In the event that no service record containing the desired service search pattern is found in the SDP server, the *SDP_serviceSearchResp* PDU will contain an empty *serviceRecordHandleList* and a *totalServiceRecordCount* parameter set to its minimum value; see section 4.5.2 of BT_SDP_spec:[7].

If the desired attributes do not exist in the SDP server, the *SDP_serviceAttributeResp* PDU will contain an empty *attributeList* and an *attributeListByteCount* parameter set to its minimum value, see section 4.6.2 of BT_SDP_spec:[7].

With the second alternative, the desired attributes are retrieved in one step:

- The LocDev sends an *SDP_serviceSearchAttributeReq* PDU where both the desired profile is included (service search pattern: profile_XYZ_UUID) and the desired attribute(s) is provided (attribute ID: 0x0004). In its response the SDP server will provide the requested attribute(s) from the service record(s) that matches the service search pattern.

In case no service record containing the desired service search pattern and/or the desired attribute(s) is found in the SDP server, the *SDP_serviceSearchAttributeResp* PDU will contain an empty *attributeLists* and an *attributeListsByteCount* parameter set to its minimum value, see section 4.7.2 of BT_SDP_spec:[7].

While, in the example in Figure 5.1, only very few service attributes are shown retrieved by the SDP client, additional information could and should be requested. Particularly in cases where service information is to be cached for future use, an SDP client should also request any pertinent information that can aid in assessing whether cached information has become stale.
The service attributes *serviceDatabaseState*, *serviceRecordState*, and *serviceInfoTimeToLive* have been defined for this purpose in BT_SDP_spec:[7]; see sections 5.2.4, 5.1.3 and 5.1.8 respectively.

6 L2CAP

The following text, together with the associated subclauses, defines the mandatory requirements with regard to this profile.

	L2CAP procedure	Support in LocDev	Support in RemDev
1.	Channel types		
	Connection-oriented channel	M	M
	Connectionless channel	X1	X1
2.	Signalling		
	Connection Establishment	M	C1
	Configuration	M	M
	Connection Termination	M	C2
	Echo	M	M
	Command Rejection	M	M
3.	Configuration Parameter Options		
	Maximum Transmission Unit	M	M
	Flush Time-out	M	M
	Quality of Service	O	O
Comments: [X1]: This feature is not used in this profile, but its use by other applications running simultaneously with this profile is not excluded. [C1]: An SDP server shall not (and cannot) initiate an L2CAP connection for SDP transactions. Nevertheless, the device that the SDP server resides in may also have an SDP client that may initiate an L2CAP connection for SDP transactions. Such action does not contradict the execution of this profile. In any case, a RemDev shall be able to process incoming requests for connection establishment. [C2] Under normal operation, an SDP server shall not initiate the process of terminating an L2CAP connection for SDP. However, exceptional cases, such as when a RemDev shuts down during the execution on an SDP transaction, cannot be excluded. In such a case, prior to the final power-off, the RemDev may gracefully (or not!) terminate all its active L2CAP connections by sending connection termination PDUs. In any case, a RemDev shall always be able to process incoming requests for connection termination.			

Table 6.1: L2CAP procedures

6.1 CHANNEL TYPES

In this profile, only connection-oriented channels shall be used. In particular, no L2CAP broadcasts are to be used for this profile.

6.2 SIGNALLING

For the purpose of retrieving SDP-related information, only a LocDev can initiate an L2CAP connection request and issue an L2CAP connection request PDU; for exceptions, see comments C1 and C2 on Table 6.1. Likewise with the corresponding L2CAP connection terminations, and the same exceptional comments C1 and C2 on Table 6.1 apply. Other than that, SDAP does not impose any additional restrictions or requirements on L2CAP signalling.

In the PSM field of the Connection Request packet, the value 0x0001 (see section 5.2 of BT_L2CAP_spec:[5]) shall be used to indicate the request for creation of an L2CAP connection for accessing the SDP layer.

6.3 CONFIGURATION OPTIONS

This section describes the usage of configuration options in the service discovery profile.

6.3.1 Maximum Transmission Unit (MTU)

This profile does not impose any additional restrictions to MTU beyond the ones stated in section 6.1 of BT_L2CAP_spec:[5]. If no MTU negotiation takes place, the default MTU value in section 6.1 of BT_L2CAP_spec:[5] shall be used.

For efficient use of the communication resources, the MTU shall be selected as large as possible, while respecting any physical constraints imposed by the devices involved, and the need that these devices continue honoring any already agreed upon QoS contracts with other devices and/or applications. It is expected that during the lifetime of an L2CAP connection for SDP transactions (also referred to as the ‘SDP session’, see Section 6.4) between two devices, any one of these devices may become engaged in an L2CAP connection with another device and/or application. If this new connection has ‘non-default’ QoS requirements, the MTU for the aforementioned SDP session is allowed to be re-negotiated during the lifetime of this SDP session, to accommodate the QoS constraints of the new L2CAP connection.

6.3.2 Flush Time-out

The SDP transactions are carried over an L2CAP reliable channel. The flush time-out value (see section 6.2 of BT_L2CAP_spec:[5]) shall be set to its default value 0xFFFF.

6.3.3 Quality of Service

The use of Quality of Service (QoS) and QoS negotiation is optional. If QoS is to be negotiated, the default settings in section 6.4 of BT_L2CAP_spec:[5] shall be used. In particular, SDP traffic shall be treated as a best-effort service type traffic.

6.4 SDP TRANSACTIONS AND L2CAP CONNECTION LIFETIME

While, in general, SDP transactions comprise a sequence of service request-and-response PDU exchanges, SDP itself constitutes a connectionless datagram service in that no SDP-level connections are formed prior to any SDP PDU exchange. SDP delegates the creation of connections on its behalf to the L2CAP layer. It is thus the responsibility of SDP – or, more correctly, of the SDP layer – to request the L2CAP layer to 'tear down' these connections on its behalf as well.

Since SDP servers are considered stateless, 'tearing down' an L2CAP connection after a service request PDU is sent (as a true connectionless service may imply) will be detrimental to the SDP transaction. Moreover, significant performance penalty will have to be paid if, for each SDP PDU transmission, a new L2CAP connection is to be created. Thus, L2CAP connections for SDP transactions shall last more than the transmission of a single SDP PDU.

An SDP *session* between an SDP client and an SDP server represents the time interval that the client and the server have the same L2CAP connection continuously present. A *minimal* SDP transaction will represent a single exchange of an SDP request PDU transmission from an SDP client to an SDP server, and the transmission of a corresponding SDP response PDU from the SDP server back to the SDP client. With respect to this profile, under normal operational conditions, the minimum duration of an SDP session shall be the duration of a minimal SDP transaction.

An SDP session may last less than the minimum required in the event of unrecoverable (processing or link) errors in layers below SDP in the LocDev and RemDev, or in the SDP layer and the service records database in the RemDev. An SDP session may also be interrupted by user intervention that may terminate the SDP session prior to the completion of an SDP transaction.

The above minimum duration of an SDP session guarantees smooth execution of the SDP transactions. For improved performance, implementers may allow SDP sessions to last longer than the minimum duration of an SDP session. As a general implementation guideline, an SDP session shall be maintained for as long as there is a need to interact with a specific device. Since the latter time is in general unpredictable, SDP implementations may maintain timers used to time periods of SDP transaction inactivity over a specific SDP session.

SDP implementations may also rely on explicit input received from a higher layer (probably initiated from the SrvDscApp itself) to open and close an SDP session with a particular device using low level primitives; e.g. **openSearch**(.) and **closeSearch**(.). Finally, an implementation may permit users to interrupt an SDP session at any time, see the **terminatePrimitive**(.) service primitive in Section 4.2.

Normally, an SDP session shall not terminate by a RemDev. Yet, such an event can indeed occur, either having the RemDev gracefully terminating the SDP session, using the L2CAP connection termination PDU, or abnormally terminating the SDP by stopping responding to SDP requests or L2CAP signalling commands. Such an event may be an indication of an exceptional condition that SDP client/server implementers should consider addressing for the smooth execution of this profile. If a termination event initiates from a RemDev, an SDP client may want to consider clearing any information obtained by this RemDev. Such an exceptional event may imply that the SDP server has (or is about to) shut-down, in which case any service information retrieved from this server should automatically become stale.

7 LINK MANAGER

7.1 CAPABILITY OVERVIEW

In this section, the LMP layer is discussed. In the table below, all LMP features are listed. The table shows which LMP features are mandatory to support with respect to this service discovery profile, which are optional and which are excluded. The reason for excluding features is that they may degrade operation of devices in this use case. Therefore, these features shall never be activated by a unit active in this use case.

If any of the rules stated below are violated, the units shall behave as defined in Section 7.2.

Traffic generated during service discovery interactions has no particular QoS requirements. As such, no particular provision of the Bluetooth link is required to support this profile.

	LM Procedure	Support in LMP	Support in LocDev	Support in RemDev
1.	Authentication	M	C1	C1
2.	Pairing	M		
3.	Change link key	M		
4.	Change the current link key	M		
4.	Encryption	O	C1	C1
5.	Clock offset request	M		
6.	Timing accuracy information request	O		
7.	LMP version	M		
8.	Supported features	M		
9.	Switch of master slave role	O		
10.	Name request	M		
11.	Detach	M		
12.	Hold mode	M		
13.	Sniff mode	O		
14.	Park mode	O		
15.	Power control	O		

Table 7.1: LMP procedures

	LM Procedure	Support in LMP	Support in LocDev	Support in RemDev
16.	Channel quality driven DM/DH	O		
17.	Quality of service	M		
18.	SCO links	O	X1	X1
19.	Control of multi-slot packets	M		
20.	Concluding parameter negotiation	M		
21.	Host connection	M		
Comments:				
[C1] No authentication or encryption is required specifically by this profile. This profile will, however, not attempt to change the existing operational settings for these procedures. Nevertheless, when this profile is executed all by itself, the default operational settings are: - authentication: no active - encryption: no active In the latter case, a LocDev will always comply with the security requirements imposed by a RemDev. If it cannot comply, it will bypass the RemDev. [X1]: This feature is not used in this profile, but its use by other applications running simultaneously with this profile is not excluded.				

Table 7.1: LMP procedures

7.2 ERROR BEHAVIOR

If a unit tries to use a mandatory feature, and the other unit replies that it is not supported, the initiating unit shall send an LMP_detach PDU with detach reason "unsupported LMP feature."

A unit shall always be able to handle the rejection of the request for an optional feature.

7.3 LINK POLICY

There are no fixed master-slave roles for the execution of this profile.

This profile does not state any requirements on which low-power modes to use, or when to use them. It is up to the Link Manager of each device to decide and request special link features as seen appropriate.

8 LINK CONTROL

8.1 CAPABILITY OVERVIEW

The following table lists all features on the LC level

	Procedure	Support in baseband	Support in LocDev	Support in RemDev
1.	Inquiry	M	C1	
2.	Inquiry scan	M		C2
3.	Paging	M	C1	
4.	Page scan			
A	Type R0	M		C3
B	Type R1	M		C3
C	Type R2	M		C3
5.	Packet types			
A	ID packet	M		
B	NULL packet	M		
C	POLL packet	M		
D	FHS packet	M		
E	DM1 packet	M		
F	DH1 packet	M		
G	DM3 packet	O		
H	DH3 packet	O		
I	DM5 packet	O		
J	DH5 packet	O		
K	AUX packet	M	X1	X1
L	HV1 packet	M	X1	X1
M	HV2 packet	O	X1	X1
N	HV3 packet	O	X1	X1
O	DV packet	M	X1	X1
6.	Inter-piconet capabilities	O		
7.	Voice codec			

Table 8.1: LC features

	Procedure	Support in baseband	Support in LocDev	Support in RemDev
A	A-law	O	X1	X1
B	μ-law	O	X1	X1
C	CVSD	O	X1	X1
Comments:				
[C1]: This mandatory LC feature will be activated under the control of the SrvDscApp.				
[C2]: This mandatory LC feature is a settable device policy (outside the scope of this profile) that is activated whenever a device is to operate in a discoverable (public) mode.				
[C3] This mandatory LC feature is a settable device policy (outside the scope of this profile) that is activated whenever a device is to operate in a discoverable or connectable (private) mode.				
[X1]: These features are not used in this profile, but their use by other applications running simultaneously with this profile is not excluded.				

Table 8.1: LC features

For the next four subsections, it is assumed that a LocDev is to perform service searches with originally unconnected RemDevs. It thus needs to inquire for and page (or only page) these RemDevs. None of the following four subsections apply whenever a LocDev performs service searches with RemDevs to which it is already connected.

8.2 INQUIRY

Whenever instructed by the SrvDscApp, the LocDev shall advise its baseband to enter the inquiry state. Entry into this state may or may not be immediate, however, depending on QoS requirements of any already existing and ongoing connections.

The user of the SrvDscApp shall be able to set the criteria for the duration of an inquiry, see *stopRule* service primitive parameter in Section 4.2. Nevertheless, the actual residence time in the inquiry state must comply with the recommendation given in section 10.7.3 of Bluetooth Baseband Specification [1].

When inquiry is invoked in a LocDev, the general inquiry procedure shall be used using a GIAC as described in Section 6.1 of Bluetooth GAP_profile:[3].

Instead of a GIAC, an appropriate DIAC can be used to narrow down the scope of the inquiry. Since the only defined DIAC (referred to as the LIAC) does not reflect any specific device or service categories, the use of DIACs is of limited (but non-zero) benefit in this profile. In particular, the profile does not exclude (but neither does it encourage) performing inquiries according to the limited inquiry procedure described in Section 6.2 of GAP_profile:[3].The information contained in the Class of Device field in the FHS packet returned by the 'inquired devices' can be used as a filter to limit the number of devices to page and connect to for subsequent SDP transactions.

8.3 INQUIRY SCAN

Inquiry scans are device-dependent policies outside the scope of this profile. Devices that operate in a discoverable mode of operation, see Section 4.1 of GAP_profile:[3], could be discovered by inquiries sent by other devices.

To be discovered by an inquiry resulting from a SrvDscApp action, a RemDev must enter inquiry scans using the GIAC; see general discoverable mode in Section 4.1.3 of GAP_profile:[3]. A DIAC can be used instead of a GIAC. As previously mentioned, the use of DIACs are of limited (but non-zero) benefit in this profile. In particular, performing inquiry scans according to the limited discoverable procedure described in Section 6.2 of GAP_profile:[3] is not excluded, but is not encouraged either.

8.4 PAGING

Whenever the SrvDscApp needs to connect to a specific RemDev for inquiring about its service records, the LocDev will advise its baseband to enter the page state. Entry into this state may or may not be immediate, however, depending on QoS requirements of any already existing and ongoing connections.

Depending on the paging class (R0, R1, or R2) indicated by a RemDev device, the LocDev shall page accordingly. The total residence time in the page state must comply with the recommendation given in section 10.6.3 of BT_BB_spec:[1]. For the pages, the 48-bit BD_ADDR of the RemDev must be used.

8.5 PAGE SCAN

Just like inquiry scans, page scans are device-dependent policies outside the scope of this profile. Devices that operate in a connectable mode of operation, see Section 4.2.2 of GAP_profile:[3], could establish Bluetooth links with other devices from pages sent by these other devices. To establish a link with a RemDev, a LocDev must send a page that results from a SrvDscApp action using the RemDev's 48-bit BD_ADDR.

8.6 ERROR BEHAVIOR

Since most features on the LC level have to be activated by LMP procedures, errors will usually be caught at that layer. However, there are some LC procedures that are independent of the LMP layer, such as inquiry or paging. Misuse of such features is difficult or sometimes impossible to detect. There is no mechanism defined to detect or prevent such improper use.

9 REFERENCES

9.1 NORMATIVE REFERENCES

[1] Baseband specification (see Volume 1, Part B)

[2] Bluetooth Assigned Numbers (see Volume 1, Appendix VIII)

[3] Generic Access Profile (see Volume 2, Part K1)

[4] Link Manager Protocol (see Volume 1, Part C)

[5] Logical Link Control and Adaptation Protocol Specification (see Volume 1, Part D)

[6] Message Sequence Charts between Host–Host Controller/Link Manager (see Volume 1, Appendix IX)

[7] Service Discovery Protocol (see Volume 1, Part E)

10 DEFINITIONS

Term	Definition
conscious	(usually referred to) a process that requires the explicit intervention of a user to be accomplished
known	(with respect to a specific device) opposite to *unknown*; a known devices is not necessarily a *paired* device
new (RemDev)	(with regard to this profile) an additional remote device (RemDev) that is discovered during a Bluetooth inquiry, and that is not already connected to local device (LocDev)
private	a mode of operation whereby a device can only be found via Bluetooth baseband pages; i.e. it only enters page scans
public	a mode of operation whereby a device can be found via Bluetooth baseband inquiries; i.e. it enters into inquiry scans. A public device also enters into page scans (contrast this with *private*)
unconscious	opposite to *conscious*
unknown	(with respect to a specific device) any other device that a specific device has no record of

11 APPENDIX A (INFORMATIVE): SERVICE PRIMITIVES AND THE BLUETOOTH PDUS

In this Annex, we relate the service primitives shown in section 4.2 with the various Bluetooth PDUs which support these primitives. The table below only shows the actions taken at the higher involved Bluetooth layer. Thus, unless specifically stated, the low-level inquiries and pages needed to discover and connect to Bluetooth devices are not discussed in detail.

service primitive	(highest layer) Bluetooth PDUs involved
serviceBrowse (LIST(*RemDev*) LIST(*RemDevRelation*) LIST(*browseGroup*) *getRemDevName* *stopRule*)	For the subset of *RemDev* that satisfy the *RemDevRelation*, this service primitive will cause the LocDev to send: an *SDP_ServiceSearchRequest* PDU and receives a corresponding response PDU, see section 4.5 in BT_SDP_spec:[7]; an *SDP_ServiceAttributeRequest* PDU and receives a corresponding response PDU, see section 4.6 in BT_SDP_spec:[7]. The first transaction above identifies the SDP servers that contain pertinent service records, while the second transaction retrieves the desired information; Alternatively, the two transactions above are combined to one: LocDev sends an *SDP_ServiceSearchAttributeRequest* PDU and receives a corresponding response PDU, see section 4.7 in BT_SDP_spec:[7] In either of the above cases, the corresponding SDP transaction may last a number of request and response PDU exchanges, due to the L2CAP MTU limitation. If the *getRemDevName* parameter is set to 'yes', then for each RemDev involved in the execution of this service primitive, the service primitive will cause a sequence of *LMP_name_request*() LM level PDUs to be sent by the LocDev.* The corresponding RemDev responds with a *LMP_name_response*() LM level PDU containing the requested user-friendly device name.
serviceSearch (LIST(*RemDev*) LIST(*RemDevRelation*) LIST(*searchPattern*, *attributeList*) *getRemDevName* *stopRule*)	same as above

Table 11.1: Bluetooth PDUs related to the service primitives in Section 4.2

service primitive	(highest layer) Bluetooth PDUs involved
enumerateRemDev (LIST(*classOfDevice*) *stopRule*)	This service primitive will cause a Bluetooth baseband *inquiry* process. The inquiry will 'indiscriminately[†]' find devices residing in the vicinity of the LocDev. Prior to returning the results of this inquiry the LocDev may filter them using the classOfDevice qualifier.
terminatePrimitive (*primitiveHandle* *returnResults*)	This service primitive will cause the termination of any outstanding operation caused by the invocation of the service primitive identified by the *primitiveHandle* parameter. This may cause an L2CAP connection termination request PDU to be sent from the LocDev to the RemDev, and the subsequent transmission of an L2CAP termination response PDU. It the LocDev is connecting to the RemDev only for the purposes of an SDP transaction, the baseband link will also be severed by the transmission of an LMP_detach LM level PDU.

Table 11.1: Bluetooth PDUs related to the service primitives in Section 4.2

[*] If the information requested is already stored (cached) in the LocDev, this service primitive may not have to cause the described LM level PDU transaction.

[†] The inquiries considered here use the GIAC. No CoD-specific DIACs have been defined. Nevertheless, the use of appropriate DIACs whenever possible is not excluded and is not outside the scope of this profile.

Part K:3

CORDLESS TELEPHONY PROFILE

This profile defines the features and procedures that are required for interoperability between different units active in the '3-in-1 phone' use case. The scope of this profile includes the following layers/protocols/profiles: Bluetooth Baseband, Link Manager Protocol, L2CAP, Service Discovery Protocol, Telephony Control Protocol Specification (TCS-Binary) and the General Access Profile.

CONTENTS

1 INTRODUCTION

1.1 SCOPE

The Cordless Telephony profile defines the protocols and procedures that shall be used by devices implementing the use case called '3-in-1 phone'.

The '3-in-1 phone' is a solution for providing an extra mode of operation to cellular phones, using Bluetooth as a short-range bearer for accessing fixed network telephony services via a base station. However, the 3-in-1 phone use case can also be applied generally for wireless telephony in a residential or small office environment, for example for cordless-only telephony or cordless telephony services in a PC – hence the profile name 'Cordless Telephony'.

This use case includes making calls via the base station, making direct intercom calls between two terminals, and accessing supplementary services provided by the external network.

1.2 PROFILE DEPENDENCIES

In Figure 1.1, the Bluetooth profile structure and the dependencies of the profiles are depicted. A profile is dependent upon another profile if it re-uses parts of that profile, by implicitly or explicitly referencing it. Dependency is illustrated in the figure. A profile has dependencies on the profile(s) in which it is contained – directly and indirectly. As indicated in the figure, the Cordless Telephony profile is dependent only upon the Generic access profile. The terminology, user interface and security aspects, modes and procedures as defined in the Generic access profile are applicable to this profile, unless explicitly stated otherwise.

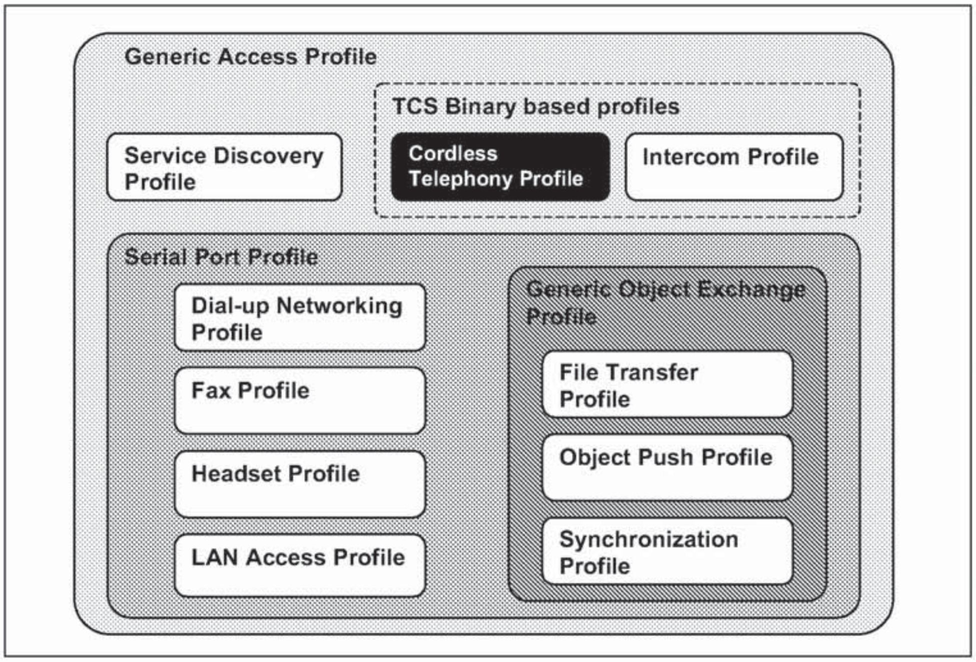


Figure 1.1: Bluetooth Profiles

1.3 SYMBOLS AND CONVENTIONS

1.3.1 Requirement status symbols

In this document, the following symbols are used:

'M' for mandatory to support (used for capabilities that shall be used in the profile);

'O' for optional to support (used for capabilities that can be used in the profile);

'C' for conditional support (used for capabilities that shall be used in case a certain other capability is supported);

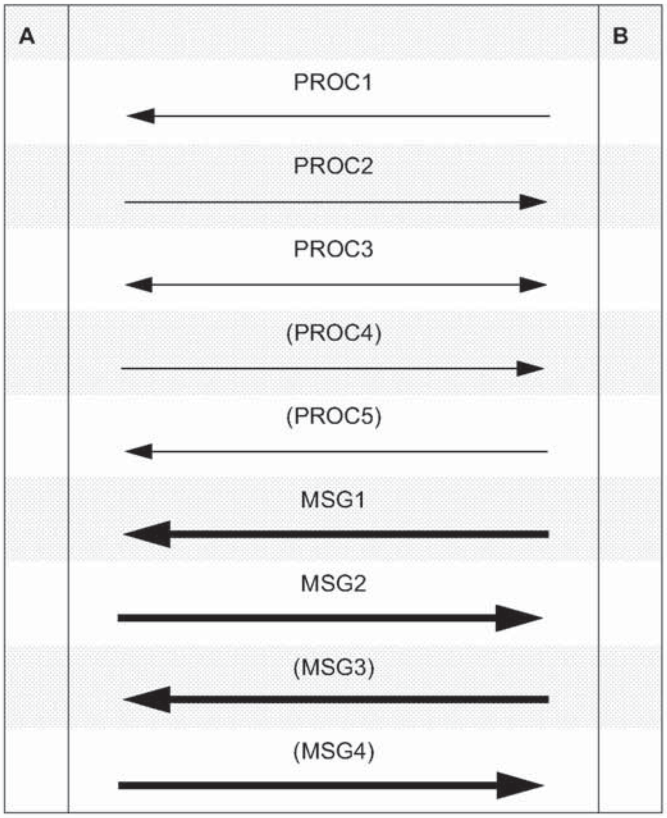
'X' for excluded (used for capabilities that may be supported by the unit, but which shall never be used in the profile);

'N/A' for not applicable (in the given context it is impossible to use this capability).

Some excluded capabilities are capabilities that, according to the relevant Bluetooth specification, are mandatory. These are features that may degrade operation of devices following this profile. Therefore, these features shall never be activated while a unit is operating as a unit within this profile.

1.3.2 Signalling diagram conventions

The following arrows are used in diagrams describing procedures:



In the table above, the following cases are shown: PROC1 is a sub-procedure initiated by B. PROC2 is a sub-procedure initiated by A. PROC3 is a sub-procedure where the initiating side is undefined (may be both A and B). PROC4 indicates an optional sub-procedure initiated by A, and PROC5 indicates an optional sub-procedure initiated by B.

MSG1 is a message sent from B to A. MSG2 is a message sent from A to B. MSG3 indicates an optional message from A to B, and MSG4 indicates an optional message from B to A.

1.3.3 Notation for timers and counters

Timers and counters may be introduced specific to this profile. To distinguish them from timers (counters) used in the Bluetooth protocol specifications and other profiles, these timers (counters) are named in the following format: ‘T_{CTP}*nnn*’ (’N_{CTP}*nnn*’).

2 PROFILE OVERVIEW

2.1 PROFILE STACK

Figure 2.1 below shows the protocols as used within this profile:

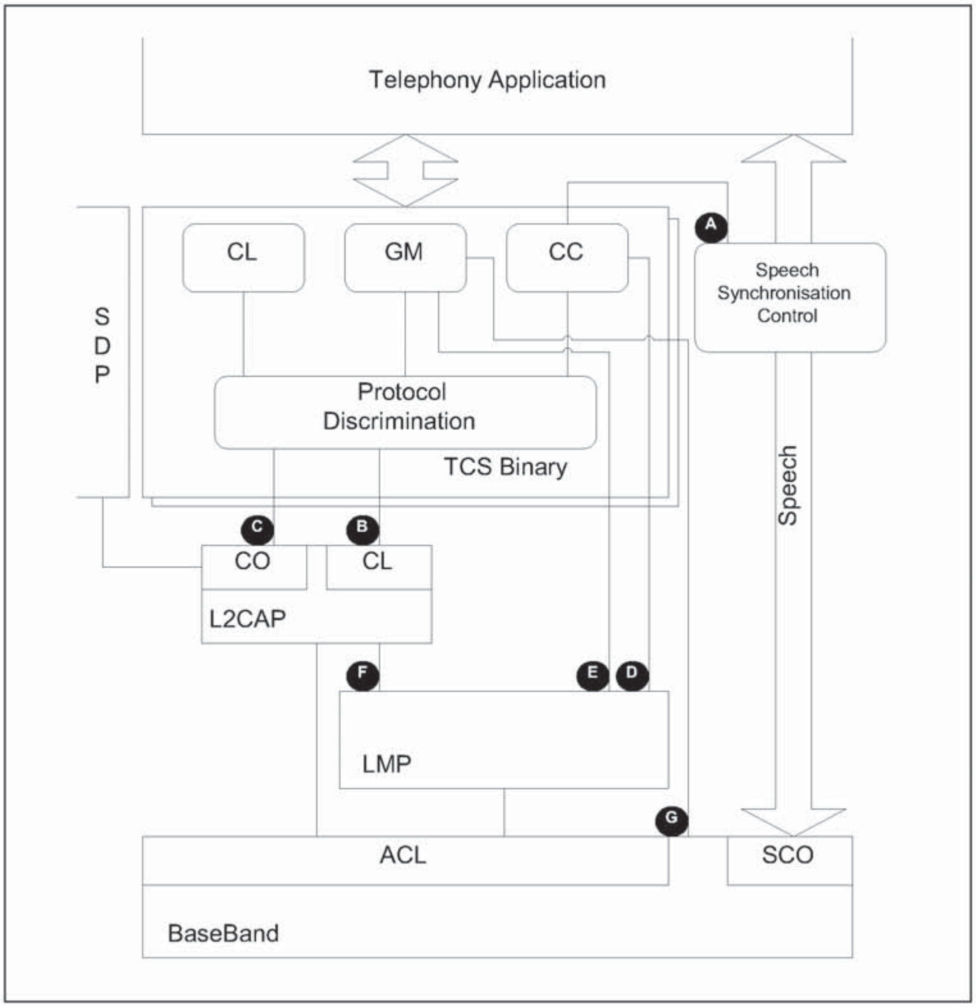


Figure 2.1: Protocol model

This profile will define the requirements for each of the layers in the model above for the Cordless Telephony profile.

In the profile, the interfaces in Figure 2.1 above are used for the following purposes:

A) The Call Control entity uses this interface to the speech synchronization control to connect and disconnect the internal speech paths.

B) This interface is used by the GW to send and by the TL to receive broadcast TCS-Binary messages.

C) This interface is used to deliver all TCS messages that are sent on a connection-oriented (point-to-point) L2CAP channel.

D) This interface is used by the Call Control entity to control the Link Manager directly for the purpose of establishing and releasing SCO links.

E) This interface is used by the Group Management to control Link Manager functions when initializing and for key handling purposes.

F) This interface is not within the scope of this profile.

G) This interface is used by the Group Management entity to control the LC/ Baseband directly to enable inquiry, paging, inquiry scan and page scan.

2.2 CONFIGURATIONS AND ROLES

The following two roles are defined for this profile:

Gateway (GW) – The GW acts as a terminal endpoint from the external network point of view and handles all interworking towards that network. The GW is the central point with respect to external calls, which means that it handles all call set-up requests to/from the external network. Examples of devices that can act as a gateway include a PSTN home base station, an ISDN home base station, a GSM gateway, a satellite gateway and an H.323 gateway.

With respect to this profile, the gateway may have the functionality to support multiple terminals being active at once, or be of a simple kind where only one terminal may be active. The simple gateway will not support multiple ringing terminals, multiple active calls or services involving more than one terminal simultaneously.

Terminal (TL) – The TL is the wireless user terminal, which may for example be a cordless telephone, a dual-mode cellular/cordless phone or a PC. Note that the scope of this profile with respect to a dual-mode cellular/cordless phone acting as TL is only the cordless mode.

The Cordless Telephony profile supports a topology of one gateway (GW) and a small number (≤7) of terminals (TLs)[1]. Figure 2.2 below shows an example of the considered architecture:

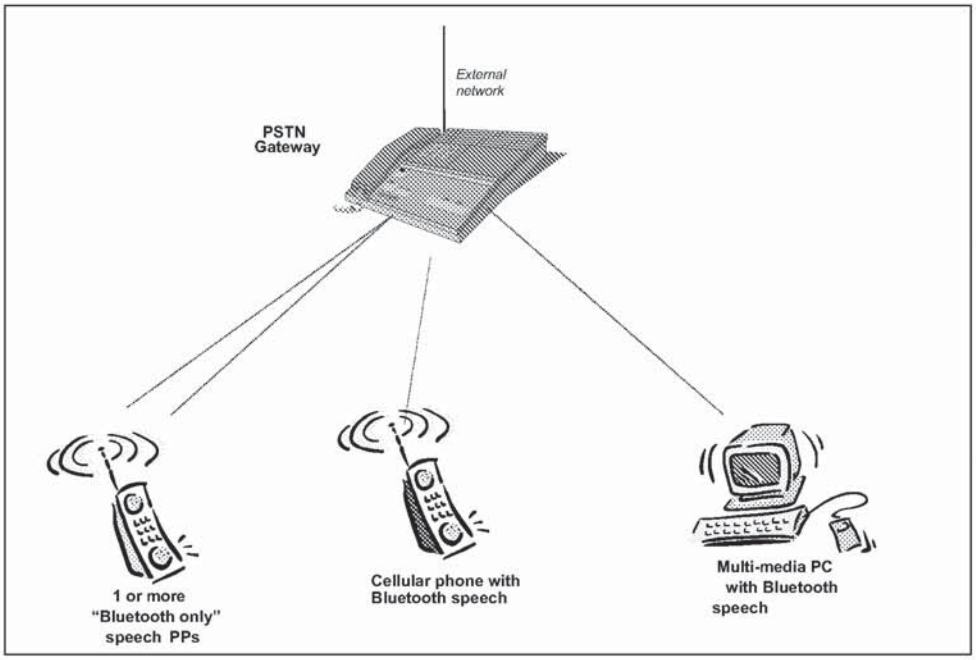

External network
PSTN Gateway
1 or more "Bluetooth only" speech PPs
Cellular phone with Bluetooth speech
Multi-media PC with Bluetooth speech

Figure 2.2: System configuration example

2.3 USER REQUIREMENTS AND SCENARIOS

The following scenarios are covered by this profile:

1. Connecting to the gateway so that incoming calls can be routed to the TL and outgoing calls can be originated.
2. Making a call from a TL to a user on the network that the gateway is connected to.
3. Receiving a call from the network that the gateway is connected to.
4. Making direct calls between two terminals.
5. Using supplementary services provided by the external network by means of DTMF signalling and register recall (hook flash).

[1] Optionally, more terminals may be supported.

2.4 PROFILE FUNDAMENTALS

The GW is normally the master of the piconet in the Cordless Telephony profile. As master, the GW will control the power mode of the TLs and may broadcast information to the TLs.

A TL that is out of range of a GW searches for it by periodically trying to page it. A GW shall devote as much of its free capacity as possible (considering power limitations and ongoing signalling) to page scanning in order to allow roaming TLs that enter the range of the GW to find it as quickly as possible. This scheme minimizes ‘air pollution’ and gives reasonable access time when coming into range of the GW. When a TL has successfully paged a GW, a master-slave switch shall be performed since the GW shall be the master. A connection-oriented L2CAP channel and, possibly, a L2CAP connectionless channel are established to be used for all TCS signalling during that Cordless Telephony session.

A TL that is within range of a GW shall normally be in park mode when it is not engaged in calls. This mode is power-efficient, allows for reasonable call set-up times and allows broadcasting to the attached TLs.

Upon arrival of an incoming call, or when a TL wants to make an outgoing call, the GW shall be put in active mode. The L2CAP channels (see above) are used for all TCS control signalling. Voice is transported using SCO links.

For security purposes, authentication of TLs and GW is used, and all user data is encrypted. To facilitate secure communication between cordless units, the WUG concept (see TCS Binary, Section 3) is used. The GW always acts as WUG master.

2.5 FEATURE DEFINITIONS

Calling line identification presentation (CLIP) – The ability to provide the calling party number to the called party before accepting the call.

Call information – The ability to provide additional information during the active phase of a call.

Connection Management – The ability to accept and (TLs only) request connections for the purposes of TCS-Bin procedures.

DTMF signalling – The ability, in external calls, to send a DTMF signal over the external network to the other party.

Incoming external call – A call originating from the external network connected to the GW.

Initialization – The infrequent process whereby a TL receives access rights to a certain GW.

Intercom call – A call originating from a TL towards another TL.

Multi-terminal support –

1. In the GW, the ability to handle multiple active terminals being registered at the same time[2]
2. In the TL, the support for a Wireless User Group (WUG)

On hook – The ability to indicate the action of going on-hook (e.g. to terminate a call), and release of all radio resources related to that call.

Outgoing external call – A call originated by a TL towards the external network connected to the GW.

Post-dialling – The ability to send dialling information after the outgoing call request set-up message is sent.

Register recall – The ability of the TL to request 'register recall', and of the GW to transmit the request to the local network. Register recall means to seize a register (with dial tone) to permit input of further digits or other actions. In some markets, this is referred to as 'hook flash'.

2.6 CONFORMANCE

If conformance to this profile is claimed, all capabilities indicated as mandatory for this profile shall be supported in the specified manner (process-mandatory). This also applies to all optional and conditional capabilities for which support is indicated. All mandatory capabilities, and optional and conditional capabilities for which support is indicated, are subject to verification as part of the Bluetooth certification program.

Note that the Intercom Profile is used for intercom calls. This means that a TL claiming conformance to the Cordless Telephony profile must conform to Intercom Profile.

[2] Note that a GW may support multiple active terminals but not a Wireless User Group (WUG).

3 APPLICATION LAYER

The following text, together with the associated sub-clauses, defines the feature requirements with regard to this profile.

Table 3.1 shows the feature requirements made by this profile.

Item no.	Feature	Support in TL	Support in GW
1.	Connection Management	M	M
2.	Outgoing external call	M	M
3.	Incoming external call	M	M
4.	Intercom call	M	N/A
5.	On hook	M	M
6.	Post-dialling	O	O
7.	Multi-terminal support	O	O
8.	Call information	O	O
9.	Calling line identification presentation (CLIP)	M	O
10.	DTMF signalling	M	M
11.	Register recall	M	M

Table 3.1: Application layer features

Table 3.2 maps each feature to the procedures used for that feature, and shows if the procedure is optional, mandatory or conditional for that feature. The procedures are described in the referenced section.

Feature	Procedure	Ref.	Support in TL	Support in GW
1. Connection Management	Connecting to a GW	4.1.1	M	M
	Connecting to a TL	4.1.2	M	N/A
2. Outgoing external call	Call request	4.2.3	M	M
	Overlap sending	4.2.4	C2	C2
	Call proceeding	4.2.5	C2	C2
	Call confirmation	4.2.6	M	O
	Call connection	4.2.7	M	M
	In-band tones and announcements	4.2.9	M	O

Table 3.2: Application layer feature to procedure mapping

Feature	Procedure	Ref.	Support in TL	Support in GW
3. Incoming external call	Call request	4.2.3	M	M
	Call confirmation	4.2.6	M	M
	Call connection	4.2.7	M	M
	Non-selected user clearing	4.2.8	M	M
	In-band tones and announcements	4.2.9	M	O
4. Intercom call	NOTE 1			
5. On hook	Call clearing	4.2.11	M	M
6. Post-dialling	Overlap sending	4.2.4	M	M
	Call proceeding	4.2.5	M	M
7. Multi-terminal support	Obtain access rights	4.4.1	M	O
	Configuration distribution	4.4.1	M	O
	Fast inter-member access	4.4.4	M	O
	Periodic key update	4.4.3	M	O
8. Call information	Call information	4.2.12	M	M
9. Calling line identification presentation (CLIP)	Calling line identity	4.3.2	M	M
10. DTMF signalling	DTMF signalling	4.3.1	M	M
11. Register recall	Register recall	4.3.3	M	M
C2: IF feature 6 THEN M else N/A				

Table 3.2: Application layer feature to procedure mapping

Note 1: For intercom calls, the intercom profile is used. Before initiating the intercom call, the TL which is initiating the call may optionally use the fast inter-member access procedure to speed up the call set-up.

4 TCS-BIN PROCEDURES

The following text together with the associated sub-clauses defines mandatory requirements with regard to this profile.

When describing TCS-BIN procedures, this section provides additional information concerning lower layer handling. The normative reference for TCS-BIN procedures is TCS Binary.

Annex A contains signalling flows that illustrate the procedures in this section.

4.1 CONNECTION MANAGEMENT

4.1.1 Connecting to a GW

When a TL connects to the GW, the link is configured and the L2CAP connection that is used for further signalling during that TCS-BIN session is set up and configured. The TL which is connecting is responsible for setting up the connection-oriented L2CAP channel.

Only trusted TLs are allowed to connect to the GW.

Note that, in order to avoid the paging delay at call set-up and to enable broadcasted messages, the TL establishes a L2CAP connection to the GW when it comes into range, and not before every call. This L2CAP connection remains until the radio link is lost or the TL stops being active in this profile. This means that the L2CAP connections used may be idle (i.e. not used to transfer data) for long periods of time.

A GW supporting feature 7, 'Multi-terminal support', uses a connectionless L2CAP channel for TCS-BIN broadcasted messages. A TL is added to the connectionless group when it connects to the GW.

4.1.2 Connecting to another TL

In the case of an intercom call, the TL which initiates the call establishes a direct link to the other TL. See the Intercom Profile for a description of these procedures.

If the TL has the capability to participate in two piconets at the same time, the TL may remain a member of the GW piconet and participate in signalling towards the GW during the intercom call.

If the TL does not have the capability to participate in two piconets at the same time, it must detach from the GW while the intercom call is active. After the intercom call is finished, the TL must re-establish the connection to the GW.

4.2 CALL CONTROL PROCEDURES

4.2.1 Sides

This section describes which sides shall be assumed for the purpose of reading TCS Binary.

In an outgoing external call, the TL is the outgoing side and the GW is the incoming side. In an incoming external call, the TL which terminates the call is the incoming side and the GW is the outgoing side.

Refer to the Intercom Profile for the sides assumed in intercom calls.

4.2.2 Call class

This section describes the usage of call classes in the Cordless Telephony profile.

An *external call* is a call between a TL and a third party connected via an external network (PSTN, ISDN, GSM or other). The call class used in SETUP messages for external calls (outgoing and incoming) is 'external call'.

An *intercom call* is a call between two TLs, which may be setup with GW support if the two TLs are members of the same WUG. Refer to Intercom Profile for call class usage in intercom calls.

4.2.3 Call request

This procedure shall be performed as defined in TCS Binary.

4.2.4 Overlap sending

This procedure shall be performed as defined in TCS Binary.

4.2.5 Call proceeding

This procedure shall be performed as defined in TCS Binary.

4.2.6 Call confirmation

This procedure shall be performed as defined in TCS Binary.

If the call is an incoming external call, and the SETUP message was delivered on a connection-oriented channel, the incoming side must acknowledge the SETUP message by performing the call confirmation procedure.

4.2.7 Call connection

This procedure shall be performed as defined in TCS Binary. The following text defines the mandatory requirements with regard to this profile.

If the bearer capability for this call is 'Synchronous Connection-Oriented', the SCO link establishment sub-procedure (see LMP, Section 3.21) shall be initiated before sending a CONNECT.

If the bearer capability for this call is 'Synchronous Connection-Oriented', the audio path shall be connected to by a unit when it receives a CONNECT or CONNECT ACKNOWLEDGE.

4.2.8 Non-selected user clearing

This procedure shall be performed as defined in TCS Binary. Additionally, the text in 4.2.11 defines the mandatory requirements with regard to this profile concerning call clearing.

4.2.9 In-band tones and announcements

This procedure shall be performed as defined in TCS Binary. The following text defines the mandatory requirements with regard to this profile.

Only the GW may provide in-band tones and announcements. The SCO link establishment sub-procedure (see Link Manager Protocol, Section 3.21) is initiated before sending a Progress Indicator information element #8, "In-band information or appropriate pattern is now available".

The audio path shall be connected to by a TL when it receives a Progress Indicator information element #8, "In-band information or appropriate pattern is now available".

4.2.10 Failure of call establishment

This procedure shall be performed as defined in TCS Binary. Additionally, the text in 4.2.11 defines the mandatory requirements with regard to this profile concerning call clearing.

4.2.11 Call clearing

All call clearing and call collision procedures as defined in TCS Binary shall be supported by both GW and TL. For a specification of the complete behavior, see TCS Binary. This section describes how the lower layers are used to release circuit switched (SCO) connections.

A unit shall release the SCO link by invoking the appropriate LMP sub-procedure (see Link Manager Protocol, Section 3.21) when a unit has received a RELEASE message.

A unit shall release the SCO link (if not already released) by invoking the appropriate LMP sub-procedure (see Link Manager Protocol, Section 3.21) when it has received a RELEASE COMPLETE message.

4.2.12 Call information

This procedure shall be performed as defined in TCS Binary.

4.3 SUPPLEMENTARY SERVICES

Supplementary services can be either internal services within the WUG, or external services provided by the network the GW is connected to.

The exact set of external supplementary services is not defined in this profile and is dependent on the network the GW is connected to. This profile provides the means for accessing them; for example through the use of DTMF signalling and register recall.

The required support for internal services and DTMF signalling is defined in the following sub-clauses.

4.3.1 DTMF signalling

The capability to request DTMF signalling towards the external network is mandatory for the TL. The capability to accept DTMF signalling requests is mandatory for the GW.

Depending on the network the GW is connected to, it shall translate the DTMF messages to the appropriate in-band or out-of-band signalling. If the network has no DTMF signalling capability, or if the GW for some reason is unable to perform DTMF signalling towards the external network, the GW shall reject the request for DTMF signalling as described below. In the START DTMF REJECT message, the GW shall use Cause #29, “Facilities rejected”.

4.3.2 Calling line identity

This procedure shall be performed as defined in TCS Binary.

It is recommended that all GWs that are connected to networks that provide calling line identity have the capability to provide this information to the user.

4.3.3 Register recall

This procedure shall be performed as defined in TCS Binary.

4.4 GROUP MANAGEMENT PROCEDURES

4.4.1 Obtain Access Rights

This procedure shall be performed as defined in TCS Binary.

A TL which wants to become member of a WUG may initiate this procedure towards a GW. The GW may accept or reject the request depending, for example, on configuration, or if the user has physical access to the base.

A GW which accepts the access rights request shall add the TL to the WUG and initiate the Configuration distribution procedure.

4.4.2 Configuration distribution

This procedure shall be performed as defined in TCS Binary.

Because of the security implications of this procedure, a TL is not forced to store the key information received during this procedure. In addition, GW may always reject the ACCESS RIGHTS REQUEST from a TL because of implementation-dependent reasons. For example, the user may be required to press a button on the GW before being granted access to the group.

Note that for intercom calls, two TLs that are members of the WUG do not need to perform the initialization procedure described in the Intercom profile (see Intercom Profile) if they use the keys distributed in the Configuration distribution procedure.

A TL which stores link keys during the Configuration Distribution procedure shall never overwrite existing link keys to other WUG members. Only if there was previously no link key to a specific device shall the key obtained during the Configuration Distribution procedure be used.

In addition to the link-loss handling described in Section 4.8, Section 4.4.2.1 applies for this procedure.

4.4.2.1 Link loss detection by GW

If the GW detects loss of link before receiving the INFO ACCEPT message, it shall consider the WUG update to be terminated unsuccessfully and consider the TL detached. If the GW detects loss of link after receiving the INFO ACCEPT message, it shall consider the WUG update to be terminated successfully.

4.4.3 Periodic key update

The K_{master} to be used during a GW-TL connection is issued to the TL when connecting to a GW. The K_{master} is intended to be a key valid for a single session only, but since the GW piconet is operational all the time, this would mean that the same K_{master} would always be used. In order to increase the security level, the K_{master} is changed periodically.

Timer T_{CTP}400 determines the interval between key changes. When T_{CTP}400 expires, the GW tries to do a periodic key update on all TLs. However, some TLs may be out of range or powered off, or the procedure may fail for some other reason. The new key in these cases is given to the TL when it attaches the next time. After there has been an attempt to update all TLs, T_{CTP}400 is reset.

The periodic key update for one TL is performed as follows. First, if the TL was parked, it is unparked. Then, the new link key is issued. After this, the new link key is activated by turning encryption off and back on. Finally, the TL may be parked.

If any of the sub-procedures fails, further sub-procedures will not be performed on that TL. The GW shall proceed with updating the next TL.

4.4.4 Fast inter-member access

The Fast inter-member access procedure is used when two TLs that are members of the same WUG need to establish a piconet of their own. This may be needed when an intercom call shall be established. Refer to TCS Binary for a definition of the procedure.

The TL_T may detach from the GW after having sent the LISTEN ACCEPT message by terminating the L2CAP channel to the GW and sending a LMP_detach.

The TL_I may detach from the GW after having received the LISTEN ACCEPT message by terminating the L2CAP channel to the GW and sending a LMP_detach.

4.5 CONNECTIONLESS PROCEDURES

TCS-BIN Connectionless (CL) messaging is not within the scope of the Cordless Telephony profile.

4.6 TCS-BIN MESSAGE OVERVIEW

This section defines the allowed TCS-BIN messages in the Cordless Telephony profile.

Message	Ability to Send		Ability to Receive	
	TL	GW	TL	GW
Access rights accept	N/A	O	C1	N/A
Access rights reject	N/A	O	C1	N/A
Access rights request	C1	N/A	N/A	O
Alerting	M	O	M	M
Call Proceeding	C2	C2	M	M
Connect	M	M	M	M
Connect Acknowledge	M	M	M	M
Disconnect	M	M	M	M
Info suggest	N/A	O	C1	N/A
Info accept	C1	N/A	N/A	O
Information	M	O	O	M
Listen request	C1	N/A	N/A	O
Listen suggest	N/A	O	C1	N/A
Listen accept	C1	O	C1	O
Listen reject	C1	O	C1	O
Progress	N/A	O	M	N/A
Release	M	M	M	M
Release Complete	M	M	M	M
Setup	M	M	M	M
Setup Acknowledge	N/A	O	O	N/A
Start DTMF	M	N/A	N/A	M
Start DTMF Acknowledge	N/A	M	M	N/A
Start DTMF Reject	N/A	M	M	N/A
Stop DTMF	M	N/A	N/A	M
Stop DTMF Acknowledge	N/A	M	M	N/A
C1: IF feature 7 THEN M else N/A C2: IF feature 6 THEN M else N/A				

Table 4.1: TCS-BIN messages

4.7 INFORMATION ELEMENT OVERVIEW

This section together with the associated sub-clauses defines the allowed information elements used in TCS-BIN messages in the Cordless Telephony profile.

Information Element	Ability to Send		Ability to Receive	
	TL	**GW**	**TL**	**GW**
Message type	M	M	M	M
Audio control	N/A	N/A	N/A	N/A
Bearer capability	M	M	M	M
Call class	M	M	M	M
Called party number	M	O	O	M
Calling party number	O	C2	M	O
Cause	M	M	M	M
Clock offset	C1	O	C1	O
Company-specific	O	O	O	O
Configuration data	N/A	O	C1	N/A
Destination CID	N/A	N/A	N/A	N/A
Keypad facility	M	N/A	N/A	M
Progress indicator	N/A	O	M	N/A
SCO handle	M	M	M	M
Sending complete	M	N/A	N/A	M
Signal	N/A	M	M	N/A
C1: IF feature 7 THEN M else N/A C2: IF feature 9 THEN M else N/A				

Table 4.2: TCS-BIN information elements

The following subsections define restrictions that apply to the contents of the TCS-BIN information elements in the Cordless Telephony profile. Note that in the tables, only fields where restrictions apply are shown. If a field is not shown in a table, it means that all values defined in TCS Binary for that field are allowed.

For those information elements not listed below, no restrictions apply.

4.7.1 Bearer capability

The following restrictions apply to the contents of the Bearer capability information element:

Field	Values allowed
Link type	SCO, None

Table 4.3: Restrictions to contents of Bearer capability information element

4.7.2 Called party number

Maximum information element length is 27 octets, thus allowing a maximum of 24 number digits.

4.7.3 Calling party number

Maximum information element length is 28 octets, thus allowing a maximum of 24 number digits.

4.7.4 Cause

The following restrictions apply to the contents of the Cause information element:

Field	Values allowed
Cause value	#1 – “Unassigned (unallocated number)” #3 – “No route to destination” #16 – “Normal call clearing” #17 – “User busy” #18 – “No user responding” #19 – “No answer from user (user alerted)” #21 – “Call rejected by user” #22 – “Number changed” #26 – “Non-selected user clearing” #28 – “Invalid number format (incomplete number” #29 – “Facilities rejected” #34 – “No circuit/channel available” #41 – “Temporary failure” #44 – “Requested circuit/channel not available” #58 – “Bearer capability not presently available” #65 – “Bearer capability not implemented” #69 – “Requested facility not implemented” #102 – “Recovery on timer expiry”

Table 4.4: Restrictions to contents of Cause information element

4.8 LINK LOSS

If a unit in a CC state other than *Null* detects loss of link, it shall immediately go to the *Null* state. Release procedures shall in this case not be performed.

A unit in any GM state which detects loss of link shall consider itself to be in the null state. Any ongoing GM procedure shall immediately be aborted and considered to be terminated unsuccessfully.

5 SERVICE DISCOVERY PROCEDURES

Table 5.1 below lists all entries in the SDP database of the GW defined by this profile. The 'Status' column indicates whether the presence of this field is mandatory or optional.

The codes assigned to the mnemonic's used in the 'Value' column, and the codes assigned to the attribute identifiers, can be found in the Bluetooth Assigned Numbers.

Item	Definition:	Type:	Value:	Status	Default
Service Class ID List				M	
Service Class #0		UUID	Generic Telephony	O	
Service Class #1		UUID	Cordless Telephony	M	
Protocol Descriptor List				M	
Protocol #0		UUID	L2CAP	M	
Protocol #1		UUID	TCS-BIN-CORDLESS	M	
Service Name	Displayable Text name	String	Service-provider defined	O	'Cordless Telephony'
External Network		UInt8	0x01=PSTN 0x02=ISDN 0x03=GSM 0x04=CDMA 0x05=Analogue cellular 0x06=Packet-switched 0x07=Other	O	
BluetoothProfileDescriptorList				M	
Profile #0		UUID	Cordless Telephony	M	
Parameter for Profile #0	Version	UInt16	0x0100*	O	0x100

Table 5.1: SDP entry for GW service

*. Indicating version 1.0

6 L2CAP PROCEDURES

The following text, together with the associated sub-clauses, define the mandatory requirements with regard to this profile.

6.1 CHANNEL TYPES

In this profile, both connection-oriented channels and connectionless channels are used.

Connectionless channels are used to broadcast information from the GW to the TLs. Only the GW shall use connectionless channels for sending. Refer to the Bluetooth Security Architecture White paper for information on the security implications of using L2CAP connectionless traffic.

In this profile, only the TL may initiate the establishment of connection-oriented channels. When connecting to the GW, the TL shall use the value 0x0007 (TCS-BIN-CORDLESS) in the PSM field of the Connection Request packet. For PSM usage in intercom calls, see Intercom Profile.

6.2 CONFIGURATION OPTIONS

This section describes the usage of configuration options in the Cordless Telephony Profile.

6.2.1 Maximum Transmission unit

The minimum MTU that a L2CAP implementation used for this profile should support is 171 octets. This means that the maximum number of TLs supported by this profile is 7.

6.2.2 Flush timeout option

The flush timeout value used for both the GW and the TL shall be the default value of 0xFFFF.

6.2.3 Quality of Service

Negotiation of Quality of Service is optional.

7 LMP PROCEDURES OVERVIEW

In this section the LMP layer is discussed. In the table below, all LMP features are listed. In the table it is shown what LMP features are mandatory to support with respect to the Cordless Telephony profile, which are optional and which are excluded. The reason for excluding features is that they may degrade operation of devices in this profile. Therefore, these features shall never be activated by a unit active in this profile.

	Procedure	Support in LMP	Support in TL	Support in GW
1.	Authentication	M		
2.	Pairing	M		
3.	Change link key	M		
4.	Change the current link key	M		
5.	Encryption	O	M	M
6.	Clock offset request	M		
7.	Slot offset information	O		
8.	Timing accuracy information request	O		
9.	LMP version	M		
10.	Supported features	M		
11.	Switch of master slave role	O	M	C1
12.	Name request	M		
13.	Detach	M		
14.	Hold mode	O		
15.	Sniff mode	O		
16.	Park mode	O	M	M
17.	Power control	O		
18.	Channel-quality driven DM/DH	O		
19.	Quality of service	M		
20.	SCO links	O	M	M
21.	Control of multi-slot packets	O		
22.	Paging scheme	O		
23.	Link supervision	M		
24.	Connection establishment	M		
C1: IF feature 7 THEN M else N/A				

Table 7.1: LMP procedures

7.1 MASTER-SLAVE SWITCH

A GW supporting feature 7, 'Multi-terminal support', must always be the master of the piconet. Such a GW will request a master-slave switch when a TL connects. If the TL rejects the request, the GW may detach it. Thus, a TL which does not accept master-slave switch requests can not be guaranteed service by all GWs.

7.2 LINK POLICY

The GW shall be as conservative as possible when deciding what power mode to put the TLs in. This means that when a TL is not engaged in signalling, the GW shall put it in a low-power mode. The recommended low-power mode to use is the park mode.

The low-power mode parameters shall be chosen such that the TL can always return to the active state within 300 ms.

If the GW can save power during a call, it may use the sniff mode. A TL may request to be put in the sniff mode.

8 LC FEATURES

The following table lists all features on the LC level.

	Procedure	Support in TL	Support in GW
1.	Inquiry		X
2.	Inquiry scan	X	
3.	Paging		X
4.	Page scan		
A	Type R0		
B	Type R1		
C	Type R2		
5.	Packet types		
A	ID packet		
B	NULL packet		
C	POLL packet		
D	FHS packet		
E	DM1 packet		
F	DH1 packet		
G	DM3 packet		
H	DH3 packet		
I	DM5 packet		
J	DH5 packet		
K	AUX packet	X	X
L	HV1 packet		
M	HV2 packet		
N	HV3 packet	M	M
O	DV packet	X	X
6.	Inter-piconet capabilities	O	C1
7.	Voice codec		
A	A-law		
B	μ-law		
C	CVSD	M	M
C1: IF feature 7 THEN M else O			

Table 8.1: LC features

8.1 INQUIRY SCAN

A device which is active in the GW role of the Cordless Telephony profile shall, in the Class of Device field:

1. Set the 'Telephony' bit in the Service Class field
2. Indicate 'Phone' as Major Device class

This may be used by an inquiring device to filter the inquiry responses.

8.2 INTER-PICONET CAPABILITIES

Inter-piconet capability is the capability, as master, to keep the synchronization of a piconet while page scanning in free slots and allowing for new members to join the piconet. While a new unit is joining the piconet (until the master-slave switch has been performed), operation may temporarily be degraded for the other members.

A GW which supports feature 7, 'Multiple terminal support', shall have inter-piconet capabilities. The TL may have inter-piconet capabilities.

9 GENERAL ACCESS PROFILE INTEROPERABILITY REQUIREMENTS

This profile requires compliance to the Generic Access Profile.

This section defines the support requirements with regards to procedures and capabilities defined in Generic Access Profile.

9.1 MODES

The table shows the support status for Modes within this profile.

	Procedure	Support in TL	Support in GW
1	Discoverability modes		
	Non-discoverable mode	N/A	M
	Limited discoverable mode	N/A	O
	General discoverable mode	N/A	M
2	Connectability modes		
	Non-connectable mode	N/A	X
	Connectable mode	N/A	M
3	Pairing modes		
	Non-pairable mode	M	M
	Pairable mode	O	M

Table 9.1: Modes

9.2 SECURITY ASPECTS

The table shows the support status for Security aspects within this profile.

	Procedure	Support in TL	Support in GW
1	Authentication	M	M
2	Security modes		
	Security mode 1	X	X
	Security mode 2	C1	C1
	Security mode 3	C1	C1
C1: Support for at least one of the security modes 2 and 3 is mandatory.			

Table 9.2: Security aspects

9.3 IDLE MODE PROCEDURES

The table shows the support status for Idle mode procedures within this profile.

	Procedure	Support in TL	Support in GW
1	General inquiry	M	N/A
2	Limited inquiry	O	N/A
3	Name discovery	O	N/A
4	Device discovery	O	N/A
5	Bonding	M	M

Table 9.3: Idle mode procedures

9.3.1 Bonding

It is mandatory for the TL to support initiation of bonding, and for the GW to accept bonding.

10 ANNEX A (INFORMATIVE): SIGNALLING FLOWS

This annex contains signalling diagrams that are used to clarify the interworking between units. This annex is informative only. The diagrams do not represent all possible signalling flows as defined by this profile.

10.1 OUTGOING EXTERNAL CALL WITHOUT POST-DIALLING

The following sequence shows the successful case when the TL does not use overlap sending:

```
TL                                                    GW
                      SETUP
          ========================>
                (CALL PROCEEDING)
          <========================
                   (ALERTING)
          <========================
             SCO LINK ESTABLISHMENT
           <--------------------------------------
                     CONNECT
          <========================
             CONNECT ACKNOWLEDGE
          ========================>
```

Figure 10.1: TL-originated call when overlap sending is not used

10.2 OUTGOING EXTERNAL CALL WITH POST-DIALLING

The following sequence shows the successful case when post-dialling is used.

TL		GW
	SETUP ========================>	
	SETUP ACKNOWLEDGE <========================	
This message is repeated until the GW has enough dialling information	INFORMATION ========================>	
	(CALL PROCEEDING) <========================	When the GW has sufficient information to complete the call, CALL PROCEEDING, ALERTING or CONNECT is sent.
	(ALERTING) <========================	
	SCO LINK ESTABLISHMENT <--	
	CONNECT <========================	
	CONNECT ACKNOWLEDGE ========================>	

Figure 10.2: Outgoing external call with post-dialling

10.3 INCOMING EXTERNAL CALL, SETUP DELIVERED ON CONNECTIONLESS CHANNEL

The figure below shows the allowed signalling flow in the successful case:

```
TL                                                 GW
                      SETUP
          <========================
                     (UNPARK)
           ---------------------------------------->
              SCO LINK ESTABLISHMENT
           ---------------------------------------->
                     CONNECT
          ========================>
              CONNECT ACKNOWLEDGE
          <========================
```

Figure 10.3: Incoming external call, SETUP delivered on connectionless channel

10.4 INCOMING EXTERNAL CALL, SETUP DELIVERED ON CONNECTION-ORIENTED CHANNEL

The figure below shows the allowed signalling flow in the successful case:

```
TL                                                 GW
                      SETUP
          <========================
                     (UNPARK)
           ---------------------------------------->
                     ALERTING
          ========================>
              SCO LINK ESTABLISHMENT
           ---------------------------------------->
                     CONNECT
          ========================>
              CONNECT ACKNOWLEDGE
          <========================
```

Figure 10.4: Incoming external call, SETUP delivered on connection-oriented channel

10.5 CALL CLEARING

The figure below shows the allowed signalling flow in the successful case when the TL initiates call clearing:

```
TL                                                       GW
                      DISCONNECT
          ========================>
                       RELEASE
          <========================
                   SCO LINK RELEASE
           <----------------------------------------
                   RELEASE COMPLETE
          ========================>
```

Figure 10.5: Call Clearing signalling flow, successful case

10.6 DTMF SIGNALLING

The figure below shows the allowed signalling flow in the successful case:

```
TL                                                       GW
                      START DTMF
          ========================>
               START DTMF ACKNOWLEDGE
          <========================
                      STOP DTMF
          ========================>
               STOP DTMF ACKNOWLEDGE
          <========================
```

Figure 10.6: DTMF signalling, successful case

10.7 DTMF SIGNALLING FAILURE

The figure below shows the allowed signalling flow in the unsuccessful case:

```
TL                                                       GW
                      START DTMF
          ========================>
                  START DTMF REJECT
          <========================
```

Figure 10.7: DTMF signalling, unsuccessful case

10.8 ACCESS RIGHTS REQUEST

The figure below shows the allowed signalling flow in the successful case:

TL		GW
	ACCESS RIGHTS REQUEST ========================>	
	ACCESS RIGHTS ACCEPT <========================	

Figure 10.8: Signalling diagram for Access Rights Request

10.9 CONFIGURATION DISTRIBUTION

The figure below shows the allowed signalling flow in the successful case:

TL		GW
	(UNPARK) <--	
	LMP_USE_SEMI_PERMANENT_KEY <---------------------------------------	For additional security, GW uses a point-to-point key to distribute WUG info.
	START ENCRYPTION <---------------------------------------	
	INFO SUGGEST <=======================	
	INFO ACCEPT ========================>	
	CHANGE CURRENT LINK KEY <---------------------------------------	GW switches back to a temporary key
	START ENCRYPTION <---------------------------------------	
	(PARK) <---------------------------------------	

Figure 10.9: Signalling diagram for Configuration distribution

10.10 PERIODIC KEY UPDATE

The figure below shows the allowed signalling flow in the successful case:

TL		**GW**
	(UNPARK)	
	<---------------------------------------	
	CHANGE THE CURRENT LINK KEY	
	<--------------------------------------	
	TURN OFF ENCRYPTION	
	<--------------------------------------	
	TURN ON ENCRYPTION	
	<--------------------------------------	
	(PARK)	
	<--------------------------------------	

Figure 10.10: Signalling diagram for periodic key update

10.11 FAST INTER-MEMBER ACCESS

The figure below shows the allowed signalling flow in the successful case:

TL$_O$		**GW**
	(UNPARK)	
	-------------------------------------->	
	LISTEN REQUEST	
	=========================>	
	LISTEN ACCEPT	
	<========================	
	(L2CAP CHANNEL RELEASE)	
	------------------------------------->	
	(DETACH)	
	------------------------------------->	

Figure 10.11: Signalling diagram for Fast inter-member access, originating side

The figure below shows the valid sub-procedure sequence between the TL$_T$ and GW:

TL$_T$		**GW**
	(UNPARK) <--	
	LISTEN SUGGEST <=======================	
(as soon as possible after sending the LISTEN ACCEPT, the TL$_T$ starts page scanning)	LISTEN ACCEPT =======================>	
	(L2CAP CHANNEL RELEASE) ------------------------------------>	
	(DETACH) ------------------------------------>	

Figure 10.12: Signalling diagram for Fast inter-member access, terminating side

11 TIMERS AND COUNTERS

Timer name	Proposed value	Description	Comment
$T_{CTP}400$	1 week	Time between periodic key updates, depending on the required security level	

Table 11.1: Defined timers

12 REFERENCES

[1] Bluetooth Baseband Specification

[2] Bluetooth Link Manager Protocol

[3] Bluetooth Logical Link Control and Adaptation Protocol Specification

[4] Bluetooth Telephony Control Protocol Specification

[5] Bluetooth Service Discovery Protocol

[6] Bluetooth Intercom Profile

[7] Bluetooth Assigned Numbers

[8] Thomas Müller, Security Architecture Whitepaper, version 0.5

13 LIST OF FIGURES

14 LIST OF TABLES

Part K:4

INTERCOM PROFILE

This profile defines the requirements for Bluetooth devices necessary for the support of the intercom functionality within the 3-in-1 phone use case. The requirements are expressed in terms of end-user services, and by defining the features and procedures that are required for interoperability between Bluetooth devices in the 3-in-1 phone use case.

CONTENTS

1 INTRODUCTION

1.1 SCOPE

This Intercom profile defines the protocols and procedures that shall be used by devices implementing the intercom part of the usage model called '3-in-1 phone'. More popularly, this is often referred to as the 'walkie-talkie' usage of Bluetooth.

1.2 PROFILE DEPENDENCIES

In Figure 1.1, the Bluetooth profile structure and the dependencies of the profiles are depicted. A profile is dependent upon another profile if it re-uses parts of that profile, by implicitly or explicitly referencing it. Dependency is illustrated in the figure: a profile has dependencies on the profile(s) in which it is contained – directly and indirectly. As indicated in the figure, the Intercom profile is dependent only upon the Generic Access Profile – details are provided in Section 9.

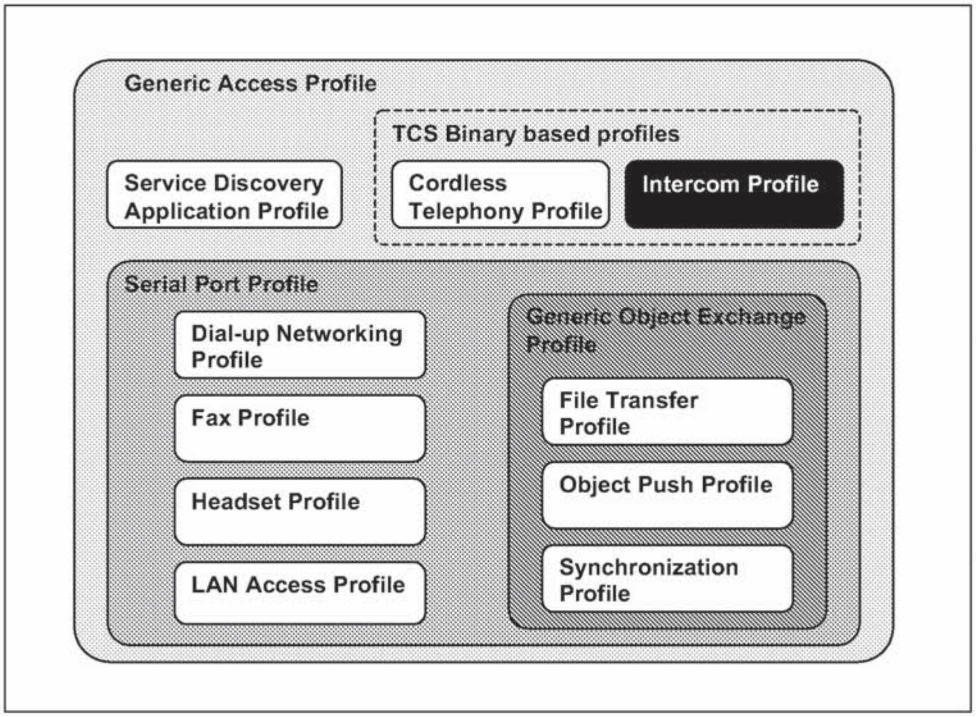


Figure 1.1: Bluetooth Profiles

1.3 SYMBOLS AND CONVENTIONS

1.3.1 Requirement status symbols

In this document, the following symbols are used:

- 'M' for mandatory to support
- 'O' for optional to support
- 'X' for excluded (used for capabilities that may be supported by the unit but shall never be used in the use case)
- 'C' for conditional to support
- 'N/A' for not applicable (in the given context it is impossible to use this capability)

Some excluded capabilities are capabilities that, according to the relevant Bluetooth specification, are mandatory. These are features that may degrade operation of devices in this use case. Therefore, these features shall never be activated while a unit is operating as a unit within this use case.

1.3.2 Signalling diagram conventions

The following arrows are used in diagrams describing procedures:

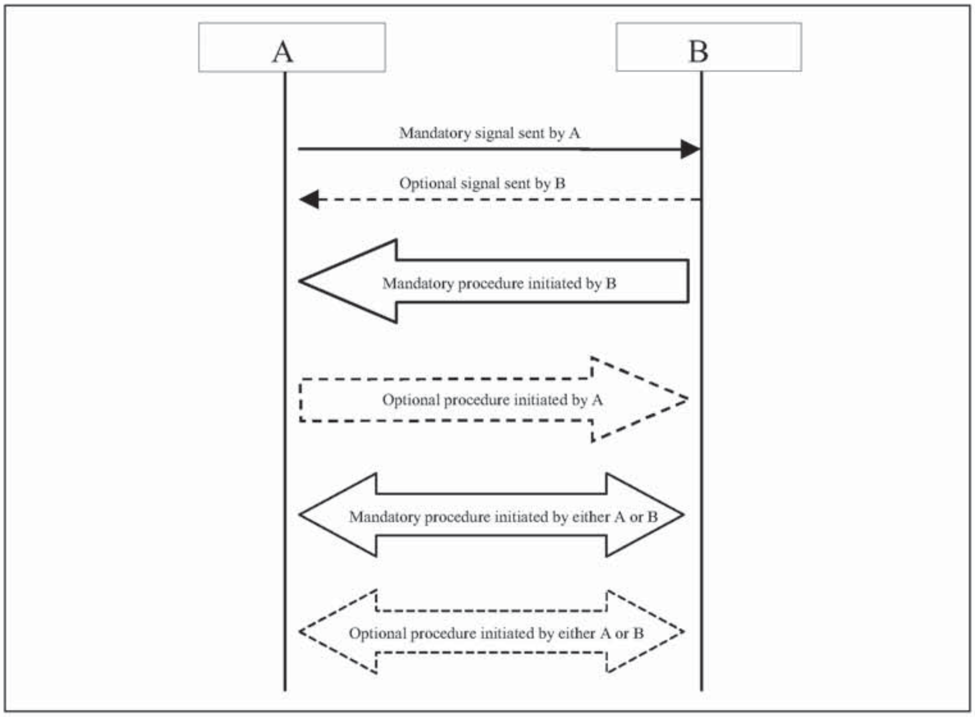

A
B
Mandatory signal sent by A
Optional signal sent by B
Mandatory procedure initiated by B
Optional procedure initiated by A
Mandatory procedure initiated by either A or B
Optional procedure initiated by either A or B

Figure 1.2: Arrows used in signalling diagrams

2 PROFILE OVERVIEW

2.1 PROFILE STACK

Figure 2.1 below shows the protocols as used within this profile:

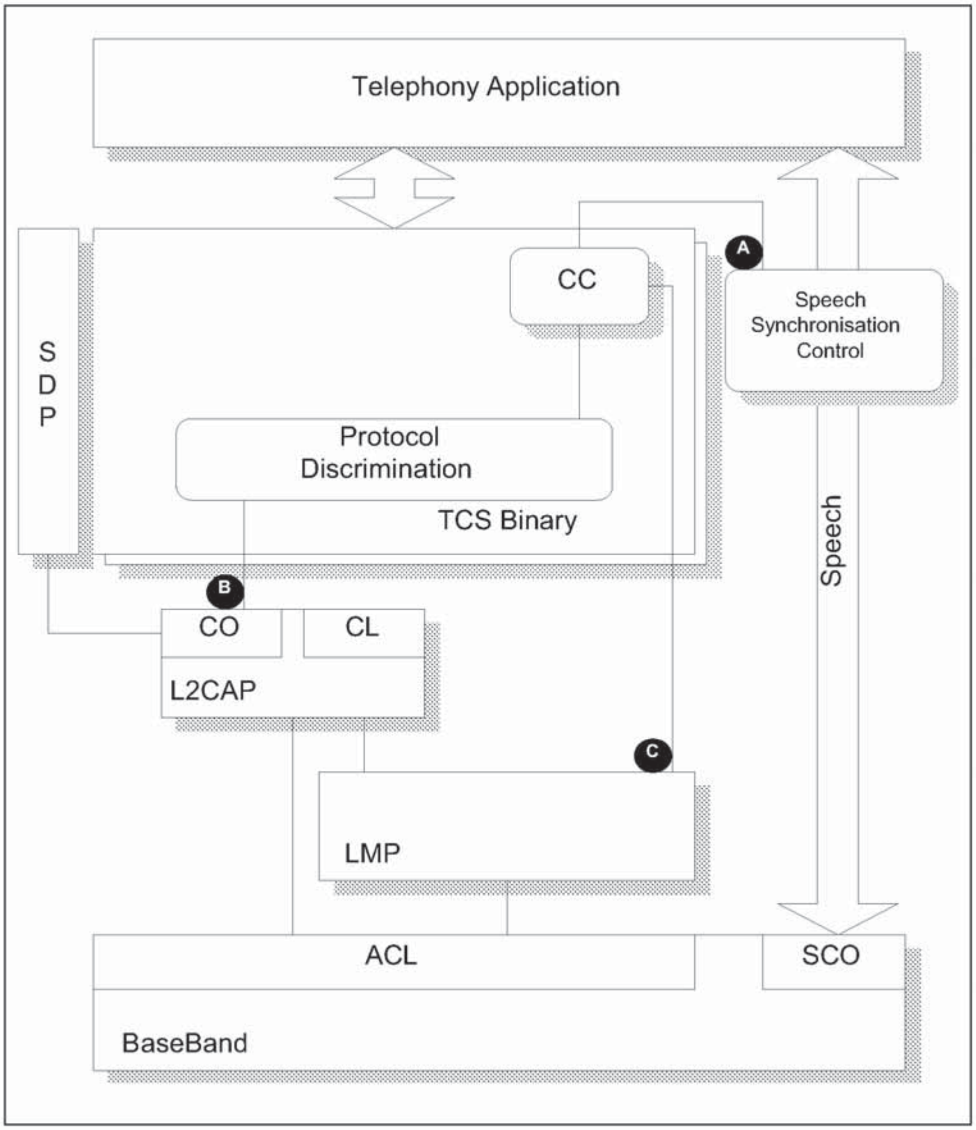


Figure 2.1: Intercom Profile Stack

This profile will define the requirements for each of the layers in the model above.

In the profile, the interfaces in Figure 2.1 above are used for the following purposes:

A) The Call Control entity uses this interface to the speech synchronization control to connect and disconnect the internal speech paths;

B) Used to deliver TCS messages on the connection-oriented (point-to-point) L2CAP channel;

C) Used by the Call Control entity to control the Link Manager directly for the purpose of establishing and releasing SCO links;

Note that, for initialization purposes, it is additionally required to control the LC/Baseband directly, to enable inquiry, paging, inquiry scan, page scan.

2.2 CONFIGURATION AND ROLES

The figure below shows a typical configuration of devices for which the Intercom profile is applicable:

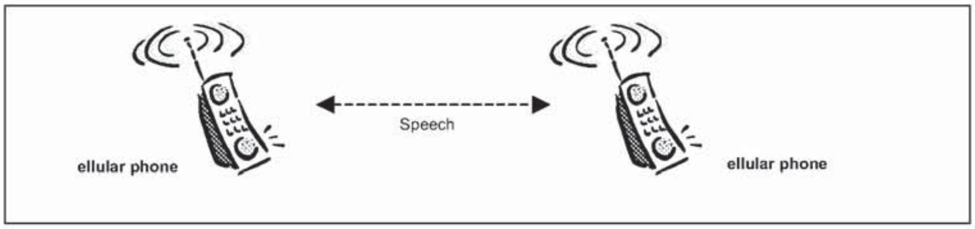


Figure 2.2: Intercom profile, example

As the intercom usage is completely symmetrical, there are no specific roles defined. A device supporting the Intercom profile will generally be denoted as Terminal (TL).

2.3 USER REQUIREMENTS AND SCENARIOS

The Intercom profile defines the protocols and procedures that shall be used by devices implementing the intercom part of the use case called '3-in-1 phone'.

The scenarios targeted by this use case are typically those where a direct speech link is required between two devices (phone, computer, ...), established using telephony-based signalling.

A typical scenario is the following:

- Two (cellular) phone users engaged in a speech call, on a direct phone-to-phone connection using Bluetooth only.

2.4 PROFILE FUNDAMENTALS

Here is a brief summary of the interactions that take place when a terminal wants to establish an intercom call towards another terminal. In the description below, the term initiator (A-party) and acceptor (B-party) will be used to designate the direction of the call.

1. If the initiator of the intercom call does not have the Bluetooth Address of the acceptor, it has to obtain this; e.g. using the Device discovery procedure – see Section 6.4 of Generic Access profile.
2. The profile does not mandate a particular security mode. If users of either device (initiator/acceptor) want to enforce security in the execution of this profile, the authentication procedure (see Section 5.1 of Generic Access profile) has to be performed to create a secure connection.
3. The initiator establishes the link and channel as indicated in Section 7 of the Generic Access profile. Based on the security requirements enforced by users of either device, authentication may be performed and encryption may be enabled.
4. The intercom call is established.
5. After the intercom call has been cleared, the channel and link will be released as well.

2.5 FEATURE DEFINITIONS

Call information – The ability to provide additional information during the active phase of a call.

Intercom call – A speech call between two terminals.

On hook – The ability to indicate the action of going on-hook (e.g. to terminate a call) and release of all radio resources related to that call.

2.6 CONFORMANCE

When conformance to this profile is claimed, all capabilities indicated mandatory for this profile shall be supported in the specified manner (process-mandatory). This also applies for all optional and conditional capabilities for which support is indicated. All mandatory capabilities, and optional and conditional capabilities, for which support is indicated are subject to verification as part of the Bluetooth certification program.

3 APPLICATION LAYER

The following text together with the associated sub-clauses defines the feature requirements with regard to this profile.

Table 3.1 below shows the feature requirements made by this profile.

Item no.	Feature	Support
1.	Intercom call	M
2.	On hook	M
3.	Call information	O

Table 3.1: Application layer features

Table 3.2 below maps each feature to the TCS Binary procedures used for that feature and shows whether the procedure is optional, mandatory or conditional for that feature.

Item no.	Feature	Procedure	Ref.	Support
1.	Intercom call	Call request	4.1.1	M
		Call confirmation	4.1.2	M
		Call connection	4.1.3	M
2.	On hook	Call clearing	4.1.5	M
3.	Call information	Call information	4.1.6	M

Table 3.2: Application layer feature to procedure mapping

4 TCS BINARY

The following text together with the associated sub-clauses defines the mandatory requirements with regard to this profile.

When describing TCS Binary procedures, this chapter provides additional information concerning lower layer handling. The normative reference for TCS Binary procedures is TCS Binary.

Annex A contains signalling flows that illustrate the procedures in this chapter.

4.1 CALL CONTROL PROCEDURES

4.1.1 Call request

This procedure shall be performed as defined in Section 2.2.1 of TCS Binary. In addition, the following applies: before a call request can be made, a connection-oriented L2CAP channel needs to be established between the two devices, using the procedures as indicated in Section 6. When the L2CAP channel has been established, the terminating side will start timer $T_{ic}(100)$. When, at expiry of timer $T_{ic}(100)$, the terminating side has not received the SETUP message initiating the call request, it may terminate the L2CAP channel. Receiving the SETUP message before expiry of $T_{IC}(100)$ will cancel the timer.

4.1.2 Call confirmation

This procedure shall be performed as defined in Section 2.2.5 of TCS Binary.

4.1.3 Call connection

This procedure shall be performed as defined in Section 2.2.6 of TCS Binary. The following text defines the mandatory requirements with regard to this profile.

The SCO link establishment sub-procedure (see LMP, Section 3.21) shall be initiated before sending a CONNECT.

The speech path shall be connected by a unit when it receives a CONNECT or CONNECT ACKNOWLEDGE.

4.1.4 Failure of call establishment

This procedure shall be performed as defined in Section 2.2.10 of TCS Binary. Additionally, the text in Section 4.1.5 defines the mandatory requirements with regard to this profile concerning call clearing.

4.1.5 Call clearing

All call-clearing and call-collision procedures as defined in Section 2.3 of TCS Binary shall be supported by the TL.

In addition, the following applies: after the last call-clearing message has been sent, a unit shall:

- release the SCO link by invoking the appropriate LMP sub-procedure (see LMP, Section 3.21.5), if not already released.
- terminate the L2CAP channel used for TCS Call-control signalling (if not already terminated) and detach the other unit.

4.1.6 Call information

This procedure shall be performed as defined in Section 2.2.7 of TCS Binary.

4.2 TCS BINARY MESSAGE OVERVIEW

This section defines the allowed TCS Binary messages in the Intercom profile. Messages not mentioned are not applicable.

Message	Support
Alerting	M
Connect	M
Connect Acknowledge	M
Disconnect	M
Information	O
Release	M
Release Complete	M
Setup	M

Table 4.1: TCS Binary messages

4.3 INFORMATION ELEMENT OVERVIEW

This section together with the associated sub-clauses defines the allowed information elements used in TCS Binary messages in the Cordless Telephony profile.

Information element	Support
Message type	M
Audio control	O
Bearer capability	M
Call class	M
Called party number	O
Calling party number	O
Cause	M
Clock offset	N/A
Company-specific	O
Configuration data	N/A
Destination CID	N/A
Keypad facility	O
Progress indicator	N/A
SCO handle	M
Sending complete	O
Signal	O

Table 4.2: TCS Binary information elements

The following subsections define restrictions that apply to the contents of the TCS Binary information elements in the Intercom profile. Note that, in the tables, only fields where restrictions apply are shown. If a field is not shown in a table, it means that all values defined in Section 7 of TCS Binary for that field are allowed.

For those information elements not listed below, no restrictions apply.

4.3.1 Bearer capability

The following restrictions apply to the contents of the Bearer capability information element:

Field	Values allowed
Link type	SCO, None
User information layer 1	CVSD

Table 4.3: Restrictions to contents of Bearer capability information element

4.3.2 Call class

The following restrictions apply to the contents of the Call class information element:

Field	Values allowed
Call class	Intercom call

Table 4.4: Restrictions to contents of Call class information element

4.3.3 Cause

The following restrictions apply to the contents of the Cause information element:

Field	Values allowed
Cause value	#16 – “Normal call clearing” #17 – “User busy”, #18 – “No user responding”, #19 – “No answer from user (user alerted)”, #21 – “Call rejected by user” #34 – “No circuit/channel available”, #41 – “Temporary failure”, #44 – “Requested circuit/channel not available”, #58 – “Bearer capability not presently available”, #65 – “Bearer capability not implemented”, #69 – “Requested facility not implemented”, #102 – “Recovery on timer expiry”

Table 4.5: Restrictions to contents of Cause information element

4.4 LINK LOSS

If a unit in a CC state other than *Null* detects loss of link, it shall immediately go to the *Null* state. Call clearing procedures shall in this case not be performed.

5 SDP INTEROPERABILITY REQUIREMENTS

Table 5.1 lists all intercom-related entries in the SDP database. For each field, the Status column indicates whether the presence of this field is mandatory or optional.

The codes assigned to the mnemonic's used in the Value column as well as the codes assigned to the attribute identifiers (if not specifically mentioned in the AttrID column) can be found in the Bluetooth Assigned Numbers section.

Item	Definition	Type	Value	AttrID	Status	Default
ServiceClassIDList					M	
ServiceClass0		UUID	Generic Telephony		M	
ServiceClass1		UUID	Intercom		M	
Protocol Descriptor List					M	
Protocol0		UUID	L2CAP		M	
Protocol1		UUID	TCS-BIN		M	
BluetoothProfileDescriptorList					O	
Profile0	Supported Profiles	UUID	Intercom		M	Intercom
Param0	Profile Version	Uint16	0x0100*		M	0x0100
Service Name	Displayable Text name	String	Service-provider defined		O	"Intercom"

Table 5.1: Service Record

*. Indicating version 1.0

6 L2CAP INTEROPERABILITY REQUIREMENTS

The following text together with the associated sub-clauses define the mandatory requirements with regard to this profile.

6.1 CHANNEL TYPES

In this profile, only connection-oriented channels are used. In the PSM field of the Connection Request packet, the default value for TCS-BIN, 0x0005 (see Section 3.2 of Assigned Numbers) shall be used.

6.2 CONFIGURATION OPTIONS

This section describes the usage of configuration options.

6.2.1 Maximum Transmission unit

The minimum MTU that a L2CAP implementation used for this profile should support is 3 octets.

6.2.2 Flush timeout option

The flush timeout value used for both the GW and the TL shall be the default value of 0xFFFF.

6.2.3 Quality of Service

Negotiation of Quality of Service is optional.

7 LINK MANAGER (LM) INTEROPERABILITY REQUIREMENTS

7.1 CAPABILITY OVERVIEW

In the table below, all LM capabilities are listed. In the table it is shown what LMP features are mandatory to support with respect to this profile and which are optional.

	Procedure	Support in LMP	Support
1.	Authentication	M	
2.	Pairing	M	
3.	Change link key	M	
4.	Change the current link key	M	
5.	Encryption	O	
6.	Clock offset request	M	
7.	Slot offset information	O	
8.	Timing accuracy information request	O	
9.	LMP version	M	
10.	Supported features	M	
11.	Switch of master slave role	O	
12.	Name request	M	
13.	Detach	M	
14.	Hold mode	O	
15.	Sniff mode	O	
16.	Park mode	O	
17.	Power control	O	
18.	Channel quality driven DM/DH	O	
19.	Quality of service	M	
20.	SCO links	O	M
21.	Control of multi-slot packets	O	
22.	Paging scheme	O	
23.	Link supervision	M	
24.	Connection establishment	M	

Table 7.1: LMP procedures

8 LINK CONTROL (LC) INTEROPERABILITY REQUIREMENTS

8.1 CAPABILITY OVERVIEW

The following table lists all capabilities on the LC level.

	Capabilities	Support
1.	Inquiry	
2.	Inquiry scan	
3.	Paging	
4.	Page scan	
A	Type R0	
B	Type R1	
C	Type R2	
5.	Packet types	
A	ID packet	
B	NULL packet	
C	POLL packet	
D	FHS packet	
E	DM1 packet	
F	DH1 packet	
G	DM3 packet	
H	DH3 packet	
I	DM5 packet	
J	DH5 packet	
K	AUX packet	X
L	HV1 packet	
M	HV2 packet	
N	HV3 packet	
O	DV packet	
6.	Inter-piconet capabilities	

Table 8.1: Baseband/LC capabilities

	Capabilities	Support
7.	Voice codec	
A	A-law	
B	μ-law	
C	CVSD	M

Table 8.1: Baseband/LC capabilities

8.2 CLASS OF DEVICE

The Class of Device field shall be set to the following:

1. Set the ‘Generic Telephony’ bit in the Service Class field
2. Indicate ‘Phone’ as Major Device class

9 GENERIC ACCESS PROFILE

This section defines the support requirements for the capabilities as defined in Generic Access Profile.

9.1 MODES

The table shows the support status for Modes within this profile.

	Procedure	Support
1	Discoverability modes	
	Non-discoverable mode	M
	Limited discoverable mode	O
	General discoverable mode	M
2	Connectability modes	
	Non-connectable mode	N/A
	Connectable mode	M
3	Pairing modes	
	Non-pairable mode	O
	Pairable mode	C3
C3: If the bonding procedure is supported, support for pairable mode is mandatory, otherwise optional		

Table 9.1: Modes

9.2 SECURITY ASPECTS

No changes to the requirements as stated in the Generic Access Profile.

9.3 IDLE MODE PROCEDURES

The table shows the support status for Idle mode procedures within this profile.

	Procedure	Support
1	General inquiry	M
2	Limited inquiry	O
3	Name discovery	O
4	Device discovery	O
5	Bonding	O

Table 9.2: Idle mode procedures

10 ANNEX A (INFORMATIVE): SIGNALLING FLOWS

This annex contains signalling diagrams that are used to clarify the interworking between units. This annex is informative only. The diagrams do not represent all possible signalling flows as defined by this profile.

10.1 CALL ESTABLISHMENT

The figure below shows the allowed signalling flow in the successful case:

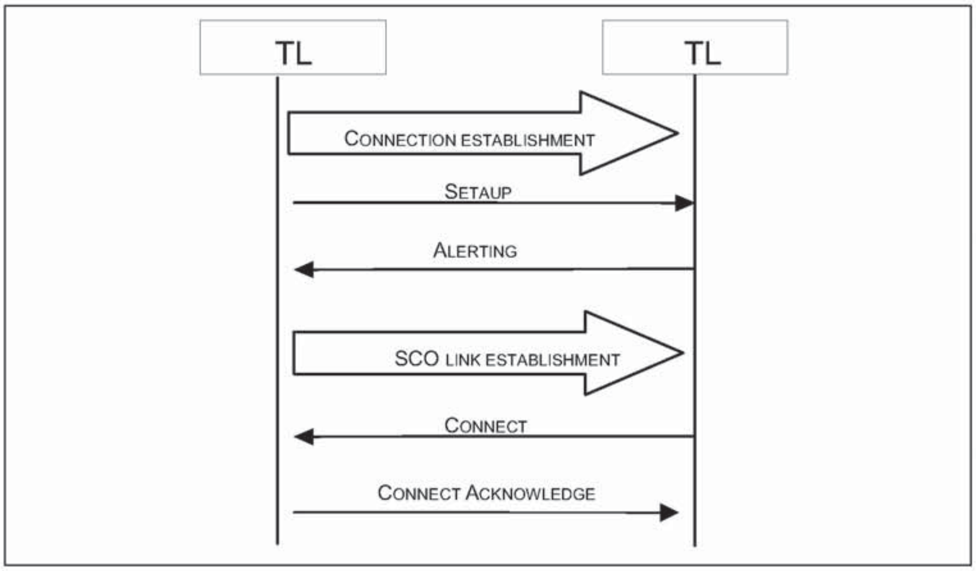


Figure 10.1: Call establishment

10.2 CALL CLEARING

The figure below shows the allowed signalling flow for the call clearing:

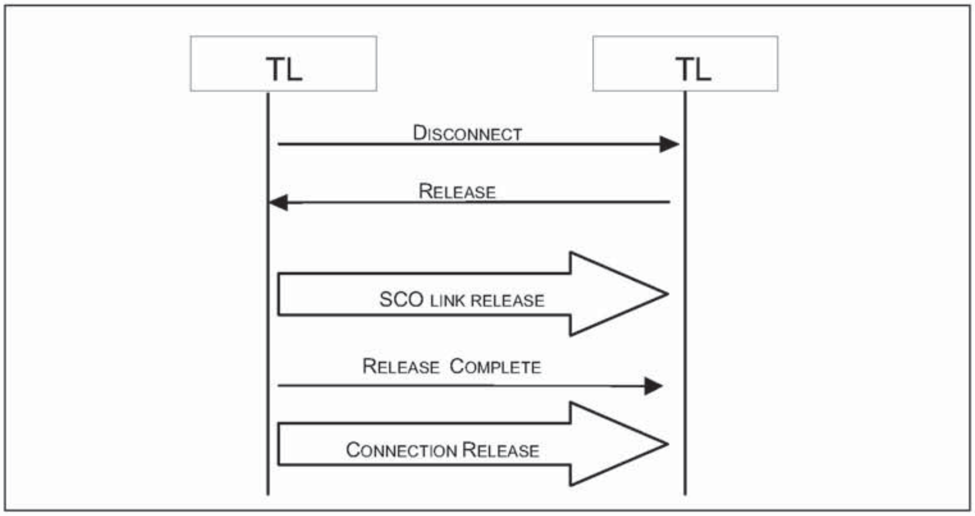


Figure 10.2: Call Clearing signalling flow, successful case

11 TIMERS AND COUNTERS

Timer name	Proposed value	Description	Comment
T_{IC}(100)	10s	Time between L2CAP connection establishment and call request initiation	

12 LIST OF FIGURES

13 LIST OF TABLES

Part K:5

SERIAL PORT PROFILE

This profile defines the requirements for Bluetooth devices necessary for setting up emulated serial cable connections using RFCOMM between two peer devices. The requirements are expressed in terms of services provided to applications, and by defining the features and procedures that are required for interoperability between Bluetooth devices.

CONTENTS

FOREWORD

Interoperability between devices from different manufacturers is provided for a specific service and use case, if the devices conform to a Bluetooth SIG-defined profile specification. A profile defines a selection of messages and procedures (generally termed *capabilities*) from the Bluetooth SIG specifications, and gives an unambiguous description of the air interface for specified service(s) and use case(s).

All defined features are process-mandatory. This means that if a feature is used, it is used in a specified manner. Whether the provision of a feature is mandatory or optional is stated separately for both sides of the Bluetooth air interface.

1 INTRODUCTION

1.1 SCOPE

The Serial Port Profile defines the protocols and procedures that shall be used by devices using Bluetooth for RS232 (or similar) serial cable emulation.

The scenario covered by this profile deals with legacy applications using Bluetooth as a cable replacement, through a virtual serial port abstraction (which in itself is operating system-dependent).

1.2 BLUETOOTH PROFILE STRUCTURE

In Figure 1.1, the Bluetooth profile structure and the dependencies of the profiles are depicted. A profile is dependent upon another profile if it re-uses parts of that profile, by implicitly or explicitly referencing it. Dependency is illustrated in the figure: a profile has dependencies on the profile(s) in which it is contained – directly and indirectly.

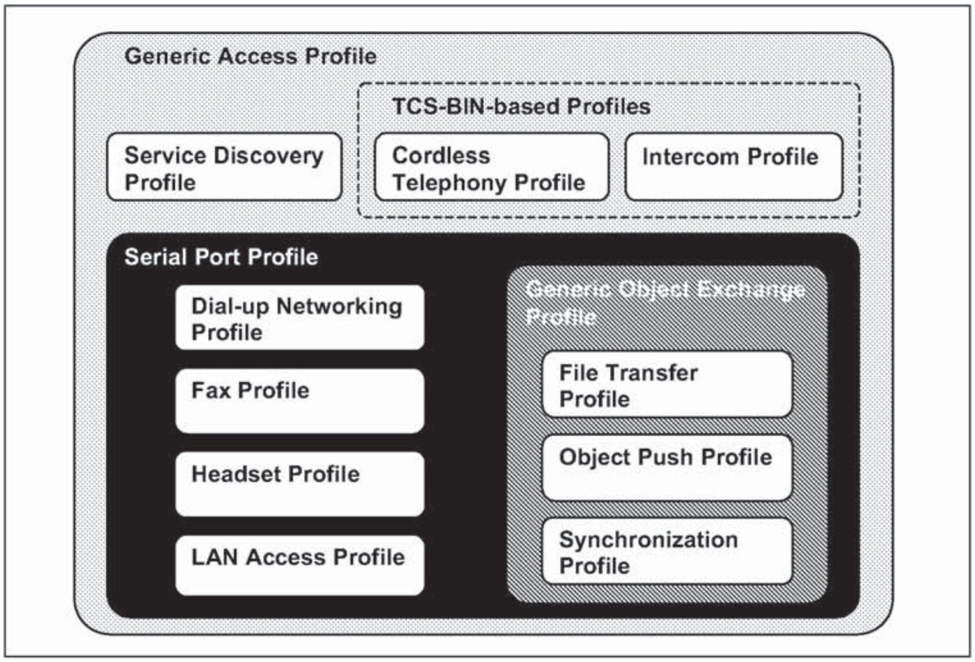


Figure 1.1: Bluetooth Profiles

1.3 SYMBOLS AND CONVENTIONS

This profile uses the symbols and conventions specified in Section 1.2 of the Generic Access Profile [9].

2 PROFILE OVERVIEW

2.1 PROFILE STACK

The figure below shows the protocols and entities used in this profile.

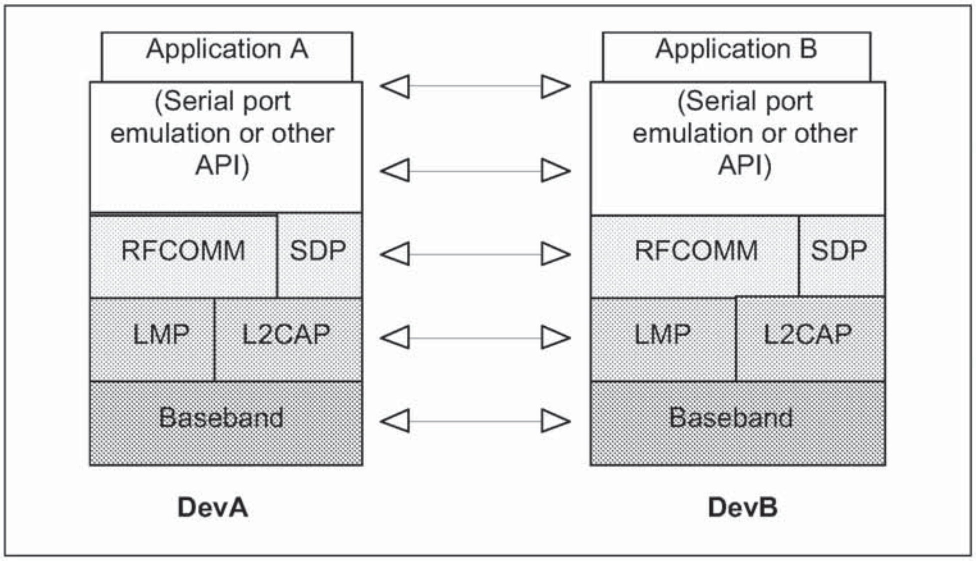


Figure 2.1: Protocol model

The Baseband [1], LMP [2] and L2CAP [3] are the OSI layer 1 and 2 Bluetooth protocols. RFCOMM [4] is the Bluetooth adaptation of GSM TS 07.10 [5], providing a transport protocol for serial port emulation. SDP is the Bluetooth Service Discovery Protocol [6].

The port emulation layer shown in Figure 2.1 is the entity emulating the serial port, or providing an API to applications.

The applications on both sides are typically legacy applications, able and wanting to communicate over a serial cable (which in this case is emulated). But legacy applications cannot know about Bluetooth procedures for setting up emulated serial cables, which is why they need help from some sort of Bluetooth-aware helper application on both sides. (These issues are not explicitly addressed in this profile; the major concern here is for Bluetooth interoperability.)

2.2 CONFIGURATIONS AND ROLES

The figure below shows one possible configuration of devices for this profile:

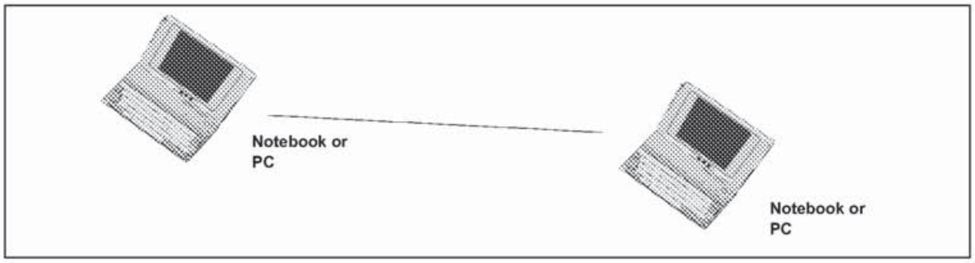


Figure 2.2: Serial Port profile, example with two notebooks.

The following roles are defined for this profile:

Device A (DevA) – This is the device that takes initiative to form a connection to another device (DevA is the *Initiator* according to Section 2.2 in GAP [9]).

Device B (DevB) – This is the device that waits for another device to take initiative to connect. (DevB is the *Acceptor* according to Section 2.2 in GAP [9]).

Note that the order of connection (from DevA to DevB) does not necessarily have to have anything to do with the order in which the legacy applications are started on each side respectively.

Informational note: For purposes of mapping the Serial Port profile to the conventional serial port architecture, both DevA and DevB can be either a Data Circuit Endpoint (DCE) or a Data Terminal Endpoint (DTE). (The RFCOMM protocol is designed to be independent of DTE-DCE or DTE-DTE relationships.)

2.3 USER REQUIREMENTS AND SCENARIOS

The scenario covered by this profile is the following:

- Setting up virtual serial ports (or equivalent) on two devices (e.g. PCs) and connecting these with Bluetooth, to emulate a serial cable between the two devices. Any legacy application may be run on either device, using the virtual serial port as if there were a real serial cable connecting the two devices (with RS232 control signalling).

This profile requires support for one-slot packets only. This means that this profile ensures that data rates up to 128 kbps can be used. Support for higher rates is optional.

Only one connection at a time is dealt with in this profile. Consequently, only point-to-point configurations are considered. However, this should not be construed as imposing any limitation on concurrence; multiple executions of this profile should be able to run concurrently in the same device. This also includes taking on the two different roles (as DevA and DevB) concurrently.

2.4 PROFILE FUNDAMENTALS

For the execution of this profile, use of security features such as authorization, authentication and encryption is optional. Support for authentication and encryption is mandatory, such that the device can take part in the corresponding procedures if requested from a peer device. If use of security features is desired, the two devices are paired during the connection establishment phase (if not earlier), see GAP, Section 7.

Bonding is not explicitly used in this profile, and thus, support for bonding is optional.

Link establishment is initiated by DevA. Service discovery procedures have to be performed to set up an emulated serial cable connection.

There are no fixed master slave roles.

RFCOMM is used to transport the user data, modem control signals and configuration commands.

2.5 CONFORMANCE

When conformance to this profile is claimed, all capabilities indicated mandatory for this profile shall be supported in the specified manner (process-mandatory). This also applies for all optional and conditional capabilities for which support is indicated. All mandatory capabilities and optional and conditional capabilities, for which support is indicated, are subject to verification as part of the Bluetooth certification program.

3 APPLICATION LAYER

This section describes the feature requirements on units complying with the Serial Port profile.

This profile is built upon the Generic Access Profile [9].

- When reading [9], the A-party (the connection initiator) is equivalent to DevA and the B-party is equivalent to the DevB.
- All the mandatory requirements defined in [9] are mandatory for this profile.
- Unless otherwise stated below, all the optional requirements defined in [9] are optional for this profile.

3.1 PROCEDURE OVERVIEW

Table 3.1 shows the required procedures:

	Procedure	Support in DevA	Support in DevB
1.	Establish link and set up virtual serial connection.	M	X
2.	Accept link and establish virtual serial connection.	X	M
3.	Register Service record for application in local SDP database.	X	M

Table 3.1: Application layer procedures

3.1.1 Establish link and set up virtual serial connection

This procedure refers to performing the steps necessary to establish a connection to an emulated serial port (or equivalent) in a remote device. The steps in this procedure are:

1. Submit a query using SDP to find out the RFCOMM Server channel number of the desired application in the remote device.
   This might include a browsing capability to let the user select among available ports (or services) in the peer device. Or, if it is known exactly which service to contact, it is sufficient look up the necessary parameters using the Service Class ID associated with the desired service.
2. Optionally, require authentication of the remote device to be performed. Also optionally, require encryption to be turned on.
3. Request a new L2CAP channel to the remote RFCOMM entity.
4. Initiate an RFCOMM session on the L2CAP channel.

5. Start a new data link connection on the RFCOMM session, using the aforementioned server channel number.

After step 5, the virtual serial cable connection is ready to be used for communication between applications on both sides.

Note: If there already exists an RFCOMM session between the devices when setting up a new data link connection, the new connection must be established on the existing RFCOMM session. (This is equivalent to skipping over steps 3 and 4 above.)

Note: The order between steps 1 and 2 is not critical (may be the other way round).

3.1.2 Accept link and establish virtual serial connection

This procedure refers to taking part in the following steps:

1. If requested by the remote device, take part in authentication procedure and, upon further request, turn on encryption.
2. Accept a new channel establishment indication from L2CAP.
3. Accept an RFCOMM session establishment on that channel.
4. Accept a new data link connection on the RFCOMM session. This may trigger a local request to authenticate the remote device and turn on encryption, if the user has required that for the emulated serial port being connected to (and authentication/encryption procedures have not already been carried out).

Note: steps 1 and 4 may be experienced as isolated events when there already exists an RFCOMM session to the remote device.

3.1.3 Register Service record in local SDP database

This procedure refers to registration of a service record for an emulated serial port (or equivalent) in the SDP database. This implies the existence of a Service Database, and the ability to respond to SDP queries.

All services/applications reachable through RFCOMM need to provide an SDP service record that includes the parameters necessary to reach the corresponding service/application, see Section 6.1. In order to support legacy applications running on virtual serial ports, the service registration must be done by some helper-application, which is aiding the user in setting up the port.

3.2 POWER MODE AND LINK LOSS HANDLING

Since the power requirements may be quite different for units active in the Serial Port profile, it is not required to use any of the power-saving modes. However, requests to use a low-power mode shall, if possible, not be denied.

If sniff, park, or hold mode is used, neither RFCOMM DLCs nor the L2CAP channel are released.

If a unit detects the loss of link, RFCOMM shall be considered having been shut down. The disconnect DLC and shutdown RFCOMM procedures referenced in Section 4 shall not be performed. Before communication on higher layers can be resumed, the Initialize RFCOMM session procedure has to be performed.

4 RFCOMM INTEROPERABILITY REQUIREMENTS

This section describes the requirements on RFCOMM in units complying with the Serial Port profile.

	Procedure	Ability to initiate		Ability to respond	
		DevA	DevB	DevA	DevB
1.	Initialize RFCOMM session	M1	X1	X1	M1
2.	Shutdown RFCOMM session	M	M	M	M
3.	Establish DLC	M	X1	X1	M
4.	Disconnect DLC	M	M	M	M
5.	RS232 control signals	C1	C1	M	M
6.	Transfer information	M	M	N/A1	N/A1
7.	Test command	X	X	M	M
8.	Aggregate flow control	C1	C1	M	M
9.	Remote Line Status indication	O	O	M	M
10.	DLC parameter negotiation	O	O	M	M
11.	Remote port negotiation	O	O	M	M

Table 4.1: RFCOMM capabilities

M1: The ability to have more than one RFCOMM session operational concurrently is optional in the RFCOMM protocol. Although support of concurrence is encouraged where it makes sense, this profile does not mandate support of concurrent RFCOMM sessions in either DevA or DevB.

X1: Within the execution of the roles defined in this profile, these abilities will not be used.

N/A1: Information transfer is unacknowledged in the RFCOMM protocol.

C1: Which flow control mechanism to use (per-DLC, aggregate, or both) is an implementation issue. But, if an implementation cannot guarantee that there will always be buffers available for data received, the ability to send either per-DLC flow control or aggregate flow control is mandatory.

Some of the procedures are further commented in subsections below.

4.1 RS232 CONTROL SIGNALS

According to TS 07.10 [5], section 5.4.6.3.7, all devices are required to send information on all changes in RS232 control signals with the Modem Status Command.

However, since RFCOMM can be used with an adaptation layer implementing any kind of API (not only virtual serial ports), it is optional to use all RS232 control signals except flow control (the RTR signal in TS 07.10 [5]). This signal can be mapped on RTS/CTS or XON/XOFF or other API mechanisms, which is an implementation issue.

Informative note: To provide interoperability between devices actually using all RS232 control signals, and devices not using them, the former type of implementation must set the states of the appropriate signals in APIs/connectors to suitable default values depending on RFCOMM DLC state. The implementation must not rely on receiving any RS232 control information from peer devices. The dependency on RFCOMM DLC state may mean that DSR/DTR as well as DCD are set to high level when an RFCOMM DLC has been established, and that the same signals are set to low level if the corresponding DLC is closed for any reason.

4.2 REMOTE LINE STATUS INDICATION

It is required to inform the other device of any changes in RS232 line status with the Remote Line Status indication command, see [5], section 5.4.6.3.10, if the local device relays information from a physical serial port (or equivalent) where overrun-, parity- or framing-errors may occur.

4.3 REMOTE PORT NEGOTIATION

DevA may inform DevB of RS232 port settings with the Remote Port Negotiation Command, directly before DLC establishment. See [5], section 5.4.6.3.9. There is a requirement to do so if the API to the RFCOMM adaptation layer exposes those settings (e.g. baud rate, parity).

DevB is allowed to send the Remote Port Negotiation command.

Informative note: the information conveyed in the remote port negotiation procedure is expected to be useful only in type II devices (with physical serial port) according to section 1.2 in RFCOMM [4], or if data pacing is done at an emulated serial port interface for any reason. RFCOMM as such will not artificially limit the throughput based on baud rate settings, see RFCOMM [4], chapter 2.

5 L2CAP INTEROPERABILITY REQUIREMENTS

The following text together with the associated sub-clauses defines the mandatory requirements with regard to this profile.

	Procedure	Support in DevA/DevB
1.	Channel types	
	Connection-oriented channel	M
	Connectionless channel	X1
2.	Signalling	
	Connection Establishment	M
	Configuration	M
	Connection Termination	M
	Echo	M
	Command Rejection	M
3.	Configuration Parameter Options	
	Maximum Transmission Unit	M
	Flush Timeout	M
	Quality of Service	O

Table 5.1: L2CAP procedures

X1: Connectionless channel is not used within the execution of this profile, but concurrent use by other profiles/applications is not excluded.

5.1 CHANNEL TYPES

In this profile, only connection-oriented channels shall be used. This implies that broadcasts will not be used in this profile.

In the PSM field of the Connection Request packet, the value for RFCOMM defined in the Assigned Numbers document [8], section 3.2 must be used.

5.2 SIGNALLING

Only DevA may issue an L2CAP Connection Request within the execution of this profile. Other than that, the Serial Port Profile does not impose any additional restrictions or requirements on L2CAP signalling.

5.3 CONFIGURATION OPTIONS

This section describes the usage of configuration options in the Serial Port Profile.

5.3.1 Maximum Transmission unit

This profile does not impose any restrictions on MTU sizes over the restrictions stated in L2CAP [3], section 6.1.

5.3.2 Flush Timeout

Serial Port data is carried over a reliable L2CAP channel. The flush timeout value shall be set to its default value 0xffff.

5.3.3 Quality of Service

Negotiation of Quality of Service is optional in this profile.

Recommendation: Implementations should try to keep an upper limit of 500 milliseconds on the latency incurred when going back from a low power mode to active mode.

6 SDP INTEROPERABILITY REQUIREMENTS

6.1 SDP SERVICE RECORDS FOR SERIAL PORT PROFILE

There are no SDP Service Records related to the Serial Port Profile in DevA.

The following table is a description of the Serial Port related entries in the SDP database of DevB. It is allowed to add further attributes to this service record.

Item	Definition	Type/Size	Value	AttributeID
ServiceClassIDList			Note1	0x0001
ServiceClass0	SerialPort / Note3	UUID/32-bit	Note1	
ProtocolDescriptorList				0x0004
Protocol0	L2CAP	UUID/32-bit	L2CAP /Note1	
Protocol1	RFCOMM	UUID/32-bit	RFCOMM /Note1	
ProtocolSpecificParameter0	Server Channel	Uint8	N = server channel #	
ServiceName	Displayable text name	DataElement/ String	"COM5" / Note4	Note2

Table 6.1: SDP Service Record

Notes:

1. Defined in the Assigned Numbers document [8].
2. For national language support for all "displayable" text string attributes, an offset has to be added to the LanguageBaseAttributeIDList value for the selected language (see the SDP Specification [6], section 5.1.14 for details).
3. The 'SerialPort' class of service is the most generic type of service. Addition of other, more specific services classes are not excluded by this profile.
4. The ServiceName attribute value suggested here is merely an example; a helper application setting up a serial port may give the port a more descriptive name.

6.2 SDP PROCEDURES

To retrieve the service records in support of this profile, the SDP client entity in DevA connects and interacts with the SDP server entity in DevB via the SDP and L2CAP procedures presented in sections 5 and 6 of SDAP [7]. In accordance to SDAP, DevA plays the role of the LocDev, while DevB plays the role of the RemDev.

7 LINK MANAGER (LM) INTEROPERABILITY REQUIREMENTS

7.1 CAPABILITY OVERVIEW

In addition to the requirements on supported procedures stated in the Link Manager specification itself (see Section 3 in the Link Manager Protocol), this profile also requires support for Encryption both in DevA and DevB.

7.2 ERROR BEHAVIOR

If a unit tries to use a mandatory feature, and the other unit replies that it is not supported, the initiating unit shall send an LMP_detach with detach reason "unsupported LMP feature."

A unit shall always be able to handle the rejection of the request for an optional feature.

7.3 LINK POLICY

There are no fixed master-slave roles for the execution of this profile.

This profile does not state any requirements on which low-power modes to use, or when to use them. That is up to the Link Manager of each device to decide and request as seen appropriate, within the limitations of the latency requirement stated in Section 5.3.3.

8 LINK CONTROL (LC) INTEROPERABILITY REQUIREMENTS

8.1 CAPABILITY OVERVIEW

The following table lists all capabilities on the LC level, and the extra requirements added to the ones in the baseband specification by this profile.

	Capabilities	Support in DevA	Support in DevB
1.	Inquiry		X1
2.	Inquiry scan	X1	
3.	Paging		X1
4.	Page scan		
A	Type R0	X1	
B	Type R1	X1	
C	Type R2	X1	
5.	Packet types		
A	ID packet		
B	NULL packet		
C	POLL packet		
D	FHS packet		
E	DM1 packet		
F	DH1 packet		
G	DM3 packet		
H	DH3 packet		
I	DM5 packet		
J	DH5 packet		
K	AUX packet	X1	X1
L	HV1 packet		
M	HV2 packet		
N	HV3 packet		
O	DV packet		
6.	Inter-piconet capabilities		

Table 8.1: Baseband/LC capabilities

	Capabilities	Support in DevA	Support in DevB
7.	Voice codec		
A	A-law		
B	μ-law		
C	CVSD		

Table 8.1: Baseband/LC capabilities

X1: These capabilities are not used within the execution of this profile, but concurrent use by other profiles/applications is not excluded.

8.2 INQUIRY

When inquiry is invoked in DevA, it shall use the General Inquiry procedure, see GAP [9], Section 6.1.

Only DevA may inquire within the execution of this profile.

8.3 INQUIRY SCAN

For inquiry scan, (at least) the GIAC shall be used, according to one of the discoverable modes defines in GAP [9], Section 4.1.2. and Section 4.1.3. That is, it is allowed to only use the Limited discoverable mode, if appropriate for the application(s) residing in DevB.

In the DevB INQUIRY RESPONSE messages, the Class of Device field will not contain any hint as to whether DevB is engaged in the execution of the Serial Port Profile or not. (This is due to the fact the generalized Serial Port service for legacy applications delivered by this profile does not fit within any of the major Service Class bits in the Class Of Device field definition.)

8.4 PAGING

Only DevA may page within the execution of this profile. The paging step will be skipped in DevA when execution of this profile begins when there already is a baseband connection between DevA and DevB. (In such a case the connection may have been set up as a result of previous paging by DevB.)

8.5 ERROR BEHAVIOR

Since most features on the LC level have to be activated by LMP procedures, errors will mostly be caught at that layer. However, there are some LC procedures that are independent of the LMP layer, e.g. inquiry or paging. Misuse of such features is difficult or sometimes impossible to detect. There is no mechanism defined to detect or prevent such improper use.

9 REFERENCES

[1] Bluetooth Special Interest Group, Bluetooth baseband specification

[2] Bluetooth Special Interest Group, Link Manager Protocol

[3] Bluetooth Special Interest Group, L2CAP Specification

[4] Bluetooth Special Interest Group, RFCOMM with TS 07.10

[5] ETSI, TS 101 369 (GSM 07.10) version 6.3.0

[6] Bluetooth Special Interest Group, Service Discovery Protocol (SDP)

[7] Bluetooth Special Interest Group, Service Discovery Application Profile

[8] Bluetooth Special Interest Group, Bluetooth Assigned Numbers

[9] Bluetooth Special Interest Group, Generic Access Profile

10 LIST OF FIGURES

11 LIST OF TABLES

Part K:6

HEADSET PROFILE

This profile defines the requirements for Bluetooth devices necessary to support the Headset use case. The requirements are expressed in terms of end-user services, and by defining the features and procedures that are required for interoperability between Bluetooth devices in the Headset use case.

CONTENTS

1 INTRODUCTION

1.1 SCOPE

This Headset profile defines the protocols and procedures that shall be used by devices implementing the usage model called 'Ultimate Headset'. The most common examples of such devices are headsets, personal computers, and cellular phones.

The headset can be wirelessly connected for the purposes of acting as the device's audio input and output mechanism, providing full duplex audio. The headset increases the user's mobility while maintaining call privacy.

1.2 PROFILE DEPENDENCIES

In Figure 1.1, the Bluetooth profile structure and the dependencies of the profiles are depicted. A profile is dependent upon another profile if it re-uses parts of that profile, by implicitly or explicitly referencing it. Dependency is illustrated in the figure: a profile has dependencies on the profile(s) in which it is contained – directly and indirectly.

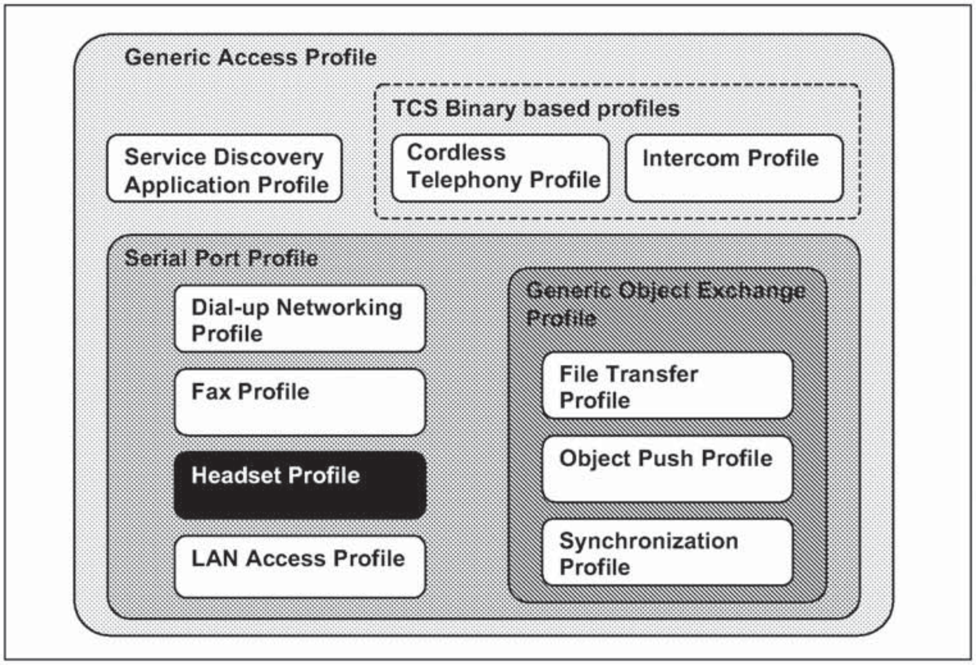


Figure 1.1: Bluetooth Profiles

As indicated in the figure, the Headset profile is dependent upon both the Serial Port Profile and the Generic access profile – details are provided in

Section 5, "Serial Port Profile," on page 210 and Section 6, "Generic Access Profile," on page 214.

1.3 SYMBOLS AND CONVENTIONS

1.3.1 Requirement status symbols

In this document, the following symbols are used:

- 'M' for mandatory to support
- 'O' for optional to support
- 'X' for excluded (used for capabilities that may be supported by the unit but shall never be used in this use case)
- 'C' for conditional to support
- 'N/A' for not applicable (in the given context it is impossible to use this capability)

Some excluded capabilities are capabilities that, according to the relevant Bluetooth specification, are mandatory. These are features that may degrade operation of devices in this use case. Therefore, these features shall never be activated while a unit is operating as a unit within this use case.

1.4 SIGNALLING DIAGRAM CONVENTIONS

The following arrows are used in diagrams describing procedures:

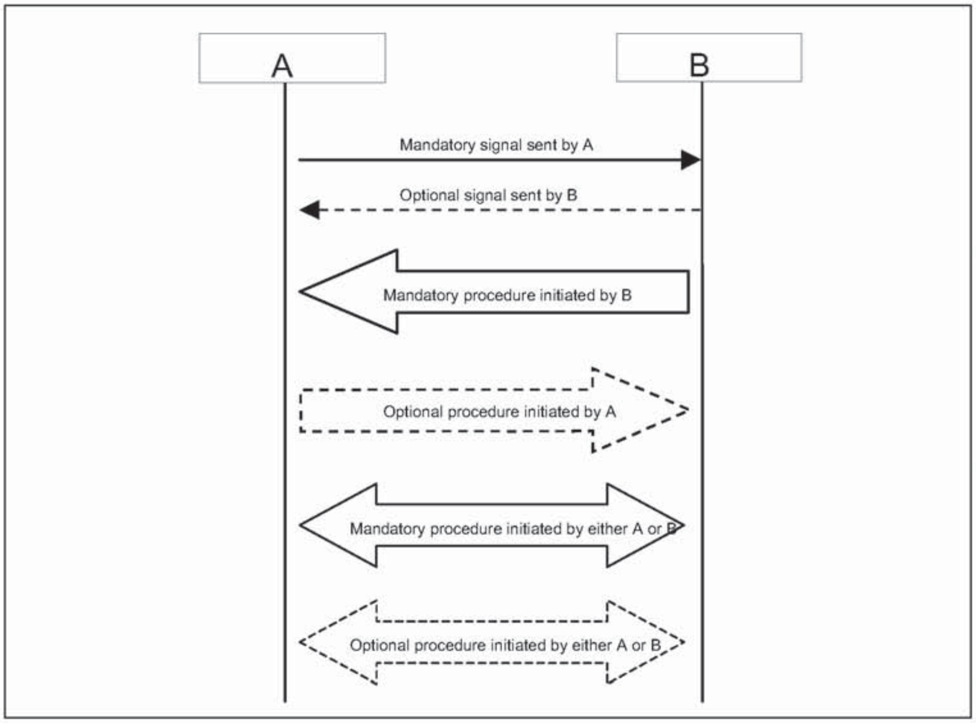

A
B
Mandatory signal sent by A
Optional signal sent by B
Mandatory procedure initiated by B
Optional procedure initiated by A
Mandatory procedure initiated by either A or B
Optional procedure initiated by either A or B

Figure 1.2: Arrows used in signalling diagrams

2 PROFILE OVERVIEW

2.1 PROFILE STACK

The figure below shows the protocols and entities used in this profile.

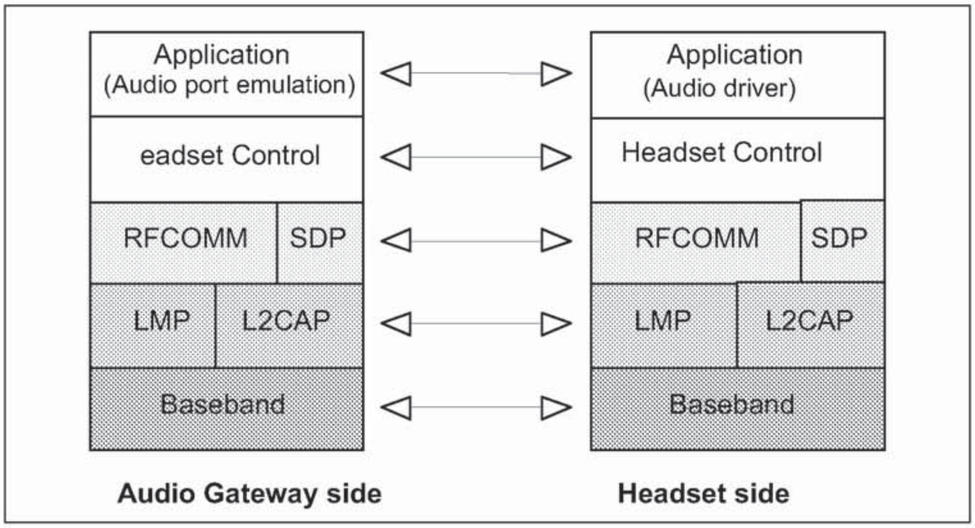


Figure 2.1: Protocol model

The Baseband, LMP and L2CAP are the OSI layer 1 and 2 Bluetooth protocols. RFCOMM is the Bluetooth adaptation of GSM TS 07.10 [5]. SDP is the Bluetooth Service Discovery Protocol. Headset Control is the entity responsible for headset specific control signalling; this signalling is AT command based.

Note: although not shown in the model above, it is assumed by this profile that Headset Control has access to some lower layer procedures (for example SCO link establishment).

The audio port emulation layer shown in Figure 2.1 is the entity emulating the audio port on the cellular phone or PC, and the audio driver is the driver software in the headset.

For the shaded protocols/entities in Figure 2.1, the Serial Port Profile is used as base standard. For these protocols, all requirements stated in the Serial Port Profile apply except in those cases where this profile explicitly states deviations.

2.2 CONFIGURATION AND ROLES

The figures below show two typical configurations of devices for which the Headset profile is applicable:

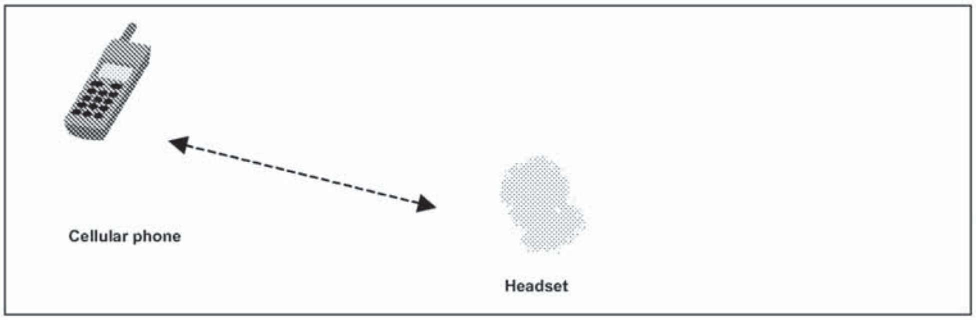


Figure 2.2: Headset profile, example with cellular phone

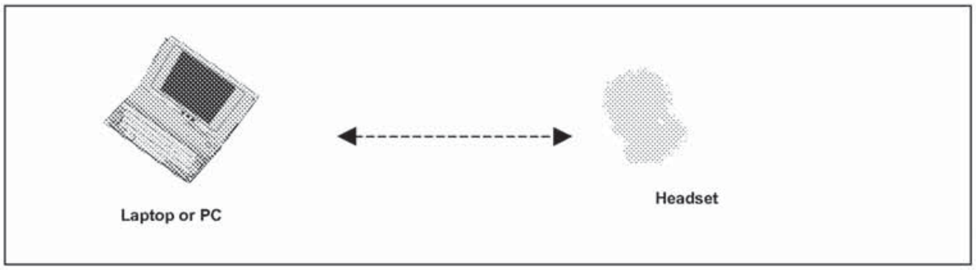


Figure 2.3: Headset profile, example with personal computer

The following roles are defined for this profile:

Audio Gateway **(AG)** – This is the device that is the gateway of the audio, both for input and output. Typical devices acting as Audio Gateways are cellular phones and personal computer.

Headset **(HS)** – This is the device acting as the Audio Gateway's remote audio input and output mechanism.

These terms are in the rest of this document only used to designate these roles.

2.3 USER REQUIREMENTS AND SCENARIOS

The Headset profile defines the protocols and procedures that shall be used by devices implementing the use case called 'Ultimate Headset'.

The following restrictions apply to this profile:

a) For this profile, it is assumed that the ultimate headset use case is the only use case active between the two devices;

b) The profile mandates the usage of CVSD for transmission of audio (for the Bluetooth part). The resulting audio is monophonic, with a quality that – under normal circumstances – will not have perceived audio degradation.

c) Between headset and audio gateway, only one audio connection at a time is supported;

d) The audio gateway controls the SCO link establishment and release. The headset directly connects (disconnects) the internal audio streams upon SCO link establishment (release). Valid speech exists on the SCO link in both directions, once established;

e) The profile offers only basic interoperability – for example, handling of multiple calls at the audio gateway is not included;

f) The only assumption on the headset's user interface is the possibility to detect a user initiated action (e.g. pressing a button).

2.4 PROFILE FUNDAMENTALS

A headset may be able to use the services of audio gateway without the creation of a secure connection. It is up to the user, if he/she wants to enforce security on devices that support authentication and/or encryption in the execution of this profile. If baseband authentication and/or encryption is desired, the two devices have to create a secure connection, using the GAP authentication procedure as described in Section 5.1 of the Generic Access profile. This procedure may then include entering a PIN code, and will include creation of link keys. In most cases, the headset will be a device with a limited user interface, so the (fixed) pin code of the headset will be used during the GAP authentication procedure.

A link has to be established when a call is initiated or received. Normally, this requires paging of the other device but, optionally, it may require unparking.

There are no fixed master/slave roles.

The audio gateway and headset provide serial port emulation. For the serial port emulation, RFCOMM is used. The serial port emulation is used to transport the user data including modem control signals and AT commands from the headset to the audio gateway. AT commands are parsed by the audio gateway and responses are sent to the headset.