- Security;
- Authentication;
- Service discovery.

The purpose of GAP is:

- To introduce definitions, recommendations, and common requirements related to modes and access procedures that are used by transport and application profiles.
- To describe how the devices behave in various states like standby and connecting. Special focus is put on discovery, connection establishment and security procedures.
- To state requirements on user interface aspects, names of procedures, and parameters, etc. This ensures a uniform user experience across all devices.

GAP defines the procedures for both BR/EDR and LE device types. It defines three device types:

- BR/EDR: Devices that support Basic Rate and Enhanced Data Rate.
- LE Only: Devices that support Low Energy configuration.
- BR/EDR/LE: Dual mode devices that support both BR/EDR and LE.

The BR/EDR procedures will be covered in this section and LE procedures will be covered in Chapter 14.

4.10.1 Bluetooth Parameters Representation

GAP states requirements about the generic terms that should be used on the user interface (UI) level. These are useful not only when designing user interfaces but also in user manuals, documentation, advertisements, etc. This helps to ensure a uniform user experience irrespective of the vendor who makes the device or the application.

GAP defines the requirements for:

- Bluetooth Device Address (BD_ADDR)
  - On the user interface level, the address should be referred to as "Bluetooth Device Address" and represented as 12 hex characters possibly separated by ":" symbol. An example of representation of the BD Address is 00:AB:CD:EF:12:34.
- Bluetooth Device Name
  - This is the user friendly name that can be used to refer to a device. This is in the form of a character string which can be retrieved by remote devices using the remote name request. The maximum length of the character string is 248 bytes.
- Bluetooth Passkey (Bluetooth PIN)

  - The pairing process was explained in the previous chapter. The Bluetooth passkey may be used to authenticate two devices via the pairing procedure. The PIN may either be entered on the UI level (For example on the user interface on mobile phone or laptop) or stored in the device (For example the predefined PIN key stored in a headset).
- Class of Device (CoD)
  - Class of device provides information on the type of device and the type of services it supports. The Class of Device is retrieved during the inquiry procedure. GAP defines that the Class of Device parameters should be referred to as "Bluetooth Device Class" and "Bluetooth Service Type" on the UI level. These are obtained from various fields of the CoD.

4.10.2 Modes

GAP defines different modes in which the device can be in with respect to inquiry and connection. These modes are explained in this section.

4.10.2.1 Discoverability Modes

Inquiry is the procedure to discover devices in the Bluetooth vicinity. With respect to inquiry, a device can be in one of the following two discoverability modes:

- Non-Discoverable mode: In this mode a device does not respond to inquiry. So it cannot be discovered by other devices.
- Discoverable mode: In this mode the device is set to discoverable mode and it may respond to inquiry from remote devices. There are two discoverable modes:
  - Limited Discoverable mode: The limited discoverable mode is used by devices that are discoverable only for a limited amount of time, during temporary conditions, or for a specific event. The device responds to a device that makes a limited inquiry.
  - General Discoverable mode: This mode is used by devices that need to be discoverable continuously or for no specific condition. The device responds to any device that make a general inquiry.

4.10.2.2 Connectability Modes

Paging is the procedure used to connect to remote devices. With respect to paging, a device can be in one of the following two connectable modes:

- Non-connectable mode: In this mode, the device does not enter the PAGE_SCAN state. So it's not possible to connect to this device. (PAGE_SCAN state was explained in the previous chapter).
- Connectable mode: In this mode, the device periodically enters the PAGE_SCAN state to check for incoming connection requests. So other devices can connect to this device.

4.10.2.3 Bondable Modes (for Pairing)

Bonding is the process of pairing when the passkey is entered on the device for the purpose of creating a "bond" between two Bluetooth devices. It may or may not be followed later on by creation of a connection.

With respect to bonding, a device can be in one of the following two bondable modes:

- Non-bondable mode: In this mode, the device does not accept a pairing request from other devices. It may still accept an incoming connection from devices that do not require bonding.
- Bondable mode: In this mode, the device accepts a pairing request from other devices. Pairing can be either in the form of legacy pairing or secure simple pairing (SSP).

4.10.3 Idle Mode Procedures

The idle mode procedures include procedures for inquiry, name discovery, bonding etc. These are called idle mode procedures because generally these are initiated when the device is in the idle mode though these can also be done when the device is already connected (for example, to create a scatternet).

4.10.3.1 Inquiry

Inquiry is the procedure to discover information about remote devices. Devices can dynamically enter and move out of the Bluetooth vicinity. This procedure is useful to find out the list of devices that are currently in the vicinity along with some basic information about those devices. Using this information, a decision may be taken to further move on to stages like discovering name, creation of connection, etc.

Inquiry provides the following information about the remote device:

- BD_ADDR;
- Clock information;
- Class of Device;
- Information about Page scan mode;
- Extended Inquiry Response Information if it is supported by the device.

Bluetooth specification defines two types of inquiry:

- General Inquiry: This is used to discover devices which are set to discoverable mode and are set to do an inquiry scan with General Inquiry Access Code. (GIAC). This is used for devices that are made discoverable continuously or for no specific condition.
- Limited Inquiry: This is used to discover devices which are scanning with Limited Inquiry Access Code (LIAC). This is used for devices which are made discoverable only for a limited amount of time, during temporary conditions for a specific event.

Devices that are set to Limited discoverable mode are also discovered in General Inquiry. The term used on User Interface level is "Bluetooth Device Inquiry."

4.10.3.2 Name Discovery

This procedure is used to get the Bluetooth name of the remote device. The term used on User Interface level is "Bluetooth Device Name Discovery."

4.10.3.3 Device Discovery

This procedure is similar to the Inquiry procedure. It is used to find some additional information besides the one provided in inquiry.

Discovery provides the following information about the remote device:

- BD_ADDR;
- Clock information;
- Class of Device;
- Information about Page scan mode;
- Extended Inquiry Response Information;
- Bluetooth Device Name—This is the additional information that is not reported during inquiry.

The term used on User Interface level is "Bluetooth Device Discovery."

4.10.3.4 Bonding

The bonding procedure is used to create a mutual trust relationship between two Bluetooth devices based on a common link key. The link key is created and exchanged during this procedure and is expected to be stored by both the Bluetooth devices. This link key is used for authentication when a connection is created.

The term used on the User Interface level is "Bluetooth Bonding."

4.10.4 Establishment Procedures

These procedures refer to the link establishment, channel establishment and connection establishment. A Device discovery or inquiry procedure is done before establishment procedures to get information about the device to which the connection is to be established. These are shown in Figure 4.15.

4.10.4.1 Link Establishment

This procedure is used to create an ACL link between two devices. The term used on the User Interface level is "Bluetooth link establishment."

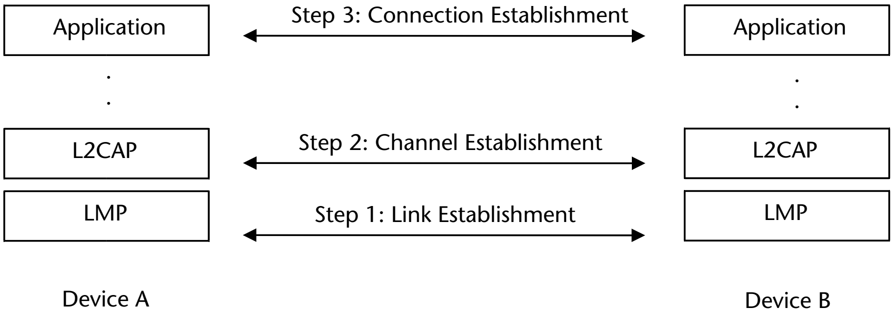
Application
Step 3: Connection Establishment
Application
L2CAP
Step 2: Channel Establishment
L2CAP
LMP
Step 1: Link Establishment
LMP
Device A
Device B

Figure 4.15 Establishment procedures.

4.10.4.2 Channel Establishment

This procedure is used to create an L2CAP channel between two devices. The term used on the User Interface level is "Bluetooth channel establishment."

4.10.4.3 Connection Establishment

This procedure is used to establish a connection between applications on two Bluetooth devices. The term used on the User Interface level is "Bluetooth connection establishment."

4.10.5 Authentication

Authentication was explained in the previous chapter. It is the process of verifying "who" is at the other end of the link. The authentication process starts when the two devices initiate a connection establishment. If the link is not already authenticated and the link key is not available, then the pairing procedure is started.

4.10.6 Security

There are two broad types of security modes: Legacy security mode and Simple Secure Pairing mode. Legacy mode is used by devices that do not support SSP.

Within legacy security mode, there are three types of security modes:

- Security mode 1 (non-secured): This mode means that no security is desired.
- Security mode 2 (service level enforced security): In this mode, the security is initiated after the connection establishment if the channel or service requires security. For example if security is required for a particular profile, then this security mode may be used.
- Security mode 3 (link level enforced security): In this mode, the security is initiated during the connection establishment.

Version 2.1 + EDR of the Bluetooth specification added Simple Secure Pairing and Security mode 4:

- Security mode 4 (service level enforced security): The security can be specified with the following attributes:
    - Authenticated link key required: This is the link key generated using numeric comparison, out-of-band, or passkey entry. It has protection against MITM attacks as explained in Chapter 3.
    - Unauthenticated link key required: This is the link key generated using just works method. It does not provide protection against MITM attacks.
    - Security optional: This is only used for SDP transactions.

It is possible for a device to support two security modes at the same time. This could be the case when it wants to connect to legacy devices using Security Mode 2 and devices that support Simple Secure Pairing with Security mode 4.

4.11 Serial Port Profile (SPP)

The serial port profile (SPP) defines the requirements for setting up emulated serial connections between Bluetooth devices. This provides similar user experience as compared to an RS232 cable connection between the two devices with the only difference that a physical wire is replaced by the Bluetooth connection between the two devices.

Bluetooth was originally designed as a cable replacement technology and this was amongst the first profiles that were used since it supports the cable replacement use case.

One of the strongest points about this profile is that it allows legacy applications (which were designed for RS232 serial ports) to use Bluetooth wireless connection instead of a wired connection. Generally another application (for example, a Bluetooth connection management application) is used to initially discover devices in the vicinity and create an SPP connection with the remote device. Once this is done, the legacy application can transparently use the Bluetooth connection as if it were a wired RS232 connection. As shown in Figure 4.14, SPP depends on GAP and re-uses the terms and procedures defined in the GAP profile. SPP is in turn used by other profiles like Hands-Free and GOEP.

SPP defines two roles:

- Dev A: This is the device that initiates a Bluetooth connection.
- Dev B: This is the device that waits for a device to make a connection and then accepts the incoming connection.

Figure 4.16 shows a typical usage scenario of Serial Port Profile. It uses the RFCOMM, L2CAP and lower layers of the Bluetooth protocol stack. A port emulation layer is used to emulate the serial port. In general this layer is dependent on the operating system. For example on Linux, this layer may expose a virtual serial

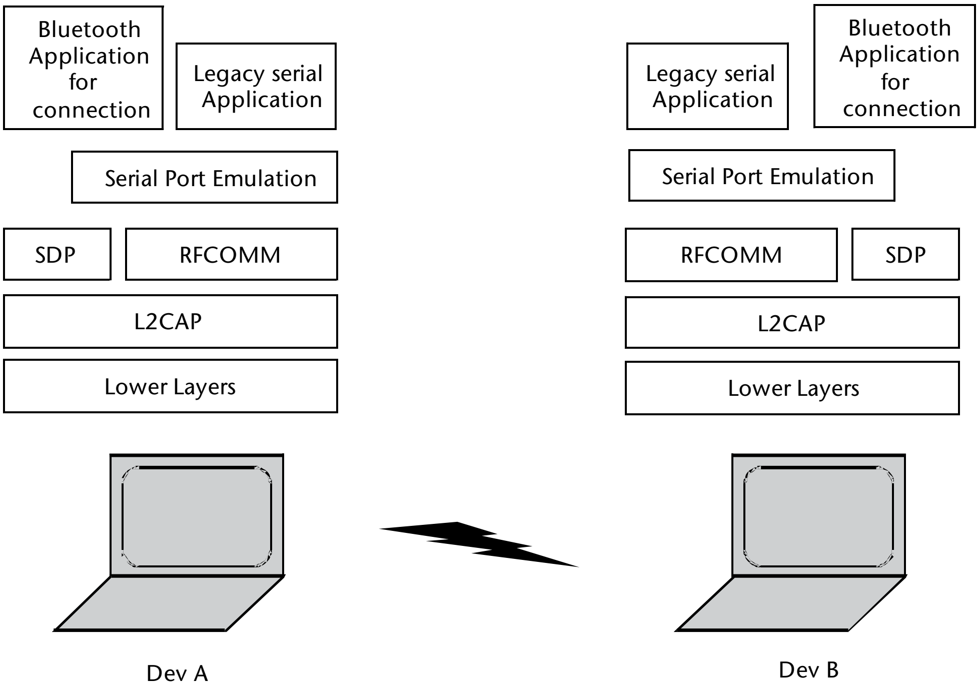
Bluetooth Application for connection
Legacy serial Application
Serial Port Emulation
SDP
RFCOMM
L2CAP
Lower Layers
Legacy serial Application
Bluetooth Application for connection
Serial Port Emulation
RFCOMM
SDP
L2CAP
Lower Layers
Dev A
Dev B

Figure 4.16 Typical usage scenario of serial port profile.

port device driver. A Legacy serial port application could be used on both devices to connect to the emulated serial port.

Besides this a separate Bluetooth application may be used for initial device discovery, services discovery, connection establishment, etc. Once a Bluetooth connection is established, the legacy application can start using the emulated serial port just like any other RS232 serial port.

4.12 Headset Profile, Hands-Free Profile

The Headset and Hands-Free profiles define the set of functions to be used for a Bluetooth connection between a mobile phone and a handsfree device, for example a Bluetooth mono headset, Carkit installed in a car, etc. The Headset profile was one of the first profiles defined by the Bluetooth specification. Later on this profile was superseded by the Hands-Free profile. The Hands-Free profile provides a superset of the Headset profile functionality.

Some of the functionality defined by this profile includes:

- Connection related functionality.
  - Connection to a Hands-Free device so that the audio can be routed from the mobile phone to the Hands-Free device.
  - Accept an incoming call.
  - Reject an incoming call.
  - Terminate a call.

- Display of phone status like signal strength, roaming, battery level, etc, on the Hands-Free device.
- Transfer an audio connection from phone to Hands-Free or vice versa.
- Different options for placing a call from the Hands-Free device.
  - From a number supplied by the Hands-Free device.
  - By memory dialing.
  - Redial last number.
- Activation of voice recognition.
- Call waiting notification.
- Three-way calling.
- Caller line identification (CLI).
- Echo cancellation and Noise reduction.
- Remote audio volume control.

Hands-Free profile defines the following two roles. These are shown in Figure 4.17.

- Audio Gateway (AG): This is the device that acts as a gateway for audio. Typically this is the mobile phone which acts as a gateway of the audio from the cellular network to the Hands-Free device.
- Hands-Free unit (HF): This is the device that acts as the audio input and output mechanism. This may also provide some means to control some of the functionality of the AG. Typically this is a Carkit or a handset.

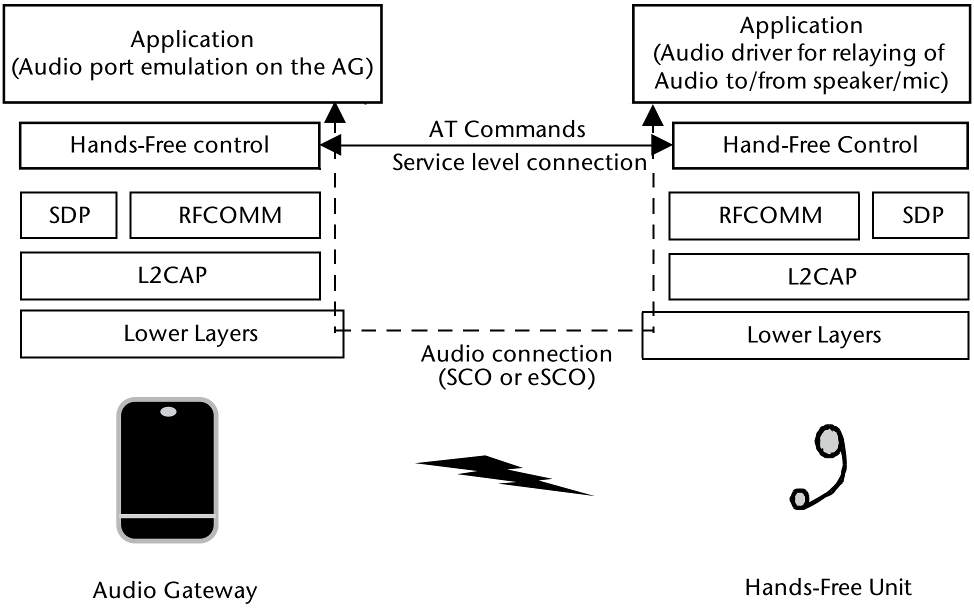

Figure 4.17 Hands-Free profile.

The Hands-Free profile defines two types of connections:

- Service level connection: This is the RFCOMM connection between the AG and HF which is used for transfer of control information.
- Audio Connection: This is the SCO or eSCO connection along with the complete audio path to route the audio (voice data) from the cellular network to the Hands-Free unit.

The Hands-Free profile uses AT commands extensively on the Service level connection to perform various tasks. The AT commands used by this profile are a subset of the 3GPP 27.0.0.2 specification.

4.13 Generic Object Exchange Profile (GOEP)

GOEP defines the requirements for devices like laptops, PDAs and smartphones to support capabilities to exchange objects. As shown in Figure 4.14, this profile is dependent on GAP and SPP. In turn it provides features to support profiles like FTP and OPP.

GOEP defines the following two roles. These are shown in Figure 4.18.

- Server: This is the device that acts as the object exchange server to and from which objects can be pushed and pulled.
- Client: This is the device that can push or pull objects to and from the server.

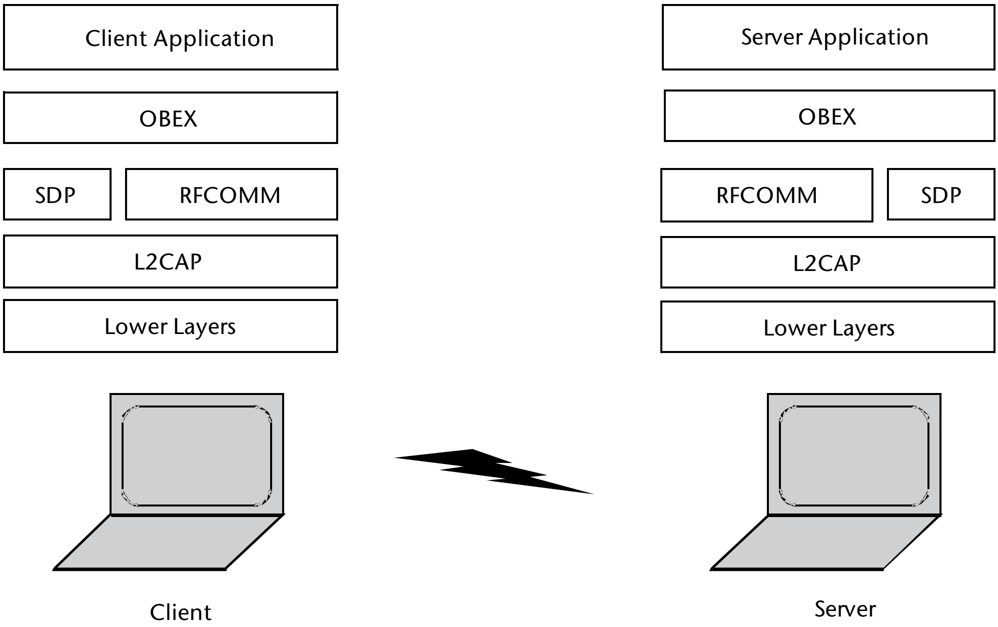

Figure 4.18 Generic object exchange profile.

As shown in Figure 4.18, this profile makes use of the OBEX, SDP, RFCOMM, L2CAP and lower layers of the protocol stack. It provides the following major features:

- Establishment of an object exchange session: This feature is used in the beginning to establish a connection between the client and the server. The remaining features can only be used once this procedure is successfully completed.
- Pushing a data object: This feature is used to transfer an object from the client to the server.
- Pulling a data object: This feature is used to retrieve an object from the server to the client.

4.14 Object Push Profile (OPP)

This profile defines the requirements needed to support the object push usage model between two Bluetooth devices. It is dependent on GOEP, SPP and GAP.

OPP defines the following two roles. These are shown in Figure 4.19.

- Push Server: This is the device that acts as the object exchange server to and from which objects can be pushed and pulled.
- Push Client: This is the device that can push or pull objects to and from the push server.

As shown in Figure 4.19, this profile makes use of the OBEX, SDP, RFCOMM, L2CAP and lower layers of the protocol stack. It provides the following major features:

- Object Push: This feature is used to push an object to the inbox of another device. For example to push a business card or an appointment to a mobile phone.
- Business Card Pull: This feature is used to pull an object from the server. For example to pull a business card from a mobile phone.
- Business Card Exchange: This feature is used to exchange objects. For example to push a business card followed by a pull of the business card.

The different objects that can be pushed by this profile are as follows:

- vCard: This is a format used for transferring contacts.
- vCalendar: This is a format used for transferring appointments.
- vMessage: This is the format used for messaging applications.
- vNote: This is the format used by notes applications.

References to the details of these formats are provided in the Bibliography section.

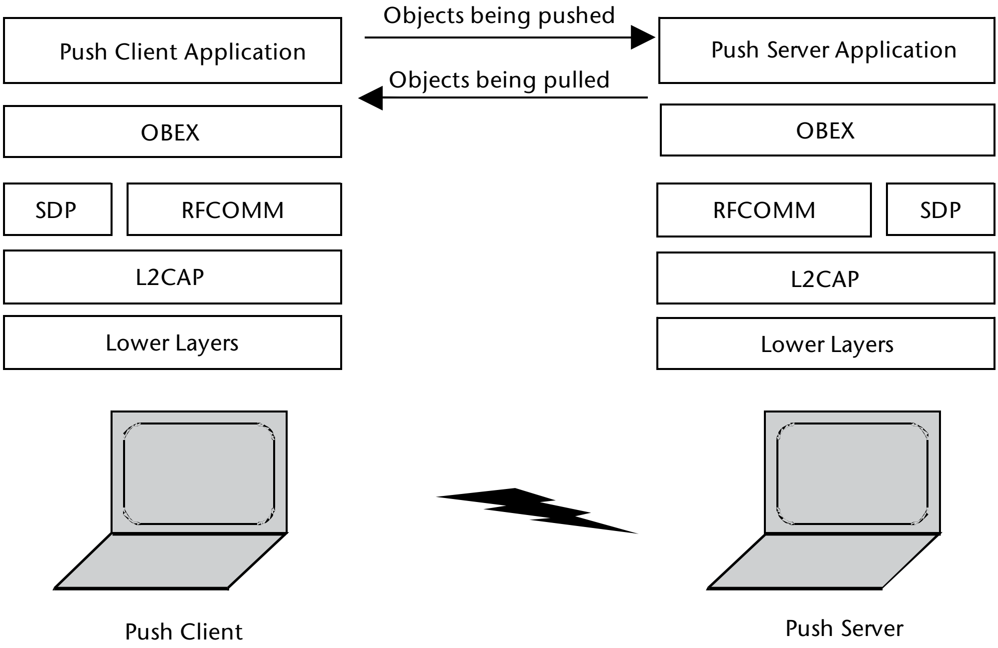

Figure 4.19 Object push profile.

4.15 File Transfer Profile (FTP)

This profile defines the requirements needed to support the file transfer usage model between two Bluetooth devices. It is dependent on GOEP, SPP and GAP.

FTP defines the following two roles. These are shown in Figure 4.20.

- Server: This is the device that acts as the file transfer server to and from which files can be pushed and pulled. It also provides the folder browsing capabilities.
- Client: This is the device that can push or pull files to and from the server.

As shown in Figure 4.20, this profile makes use of the OBEX, SDP, RFCOMM, L2CAP, and lower layers of the protocol stack. It provides the following major features:

- Browse the file system: This feature allows support for browsing the file system of the server. This includes viewing the files and folders and navigating the folder hierarchy of the other device.
- File Transfer: This feature provides support for transferring files and folders from one device to another.
- Object manipulation: This feature allows manipulating the objects on the other device. This may include deleting files, creating folders, deleting folders, etc.

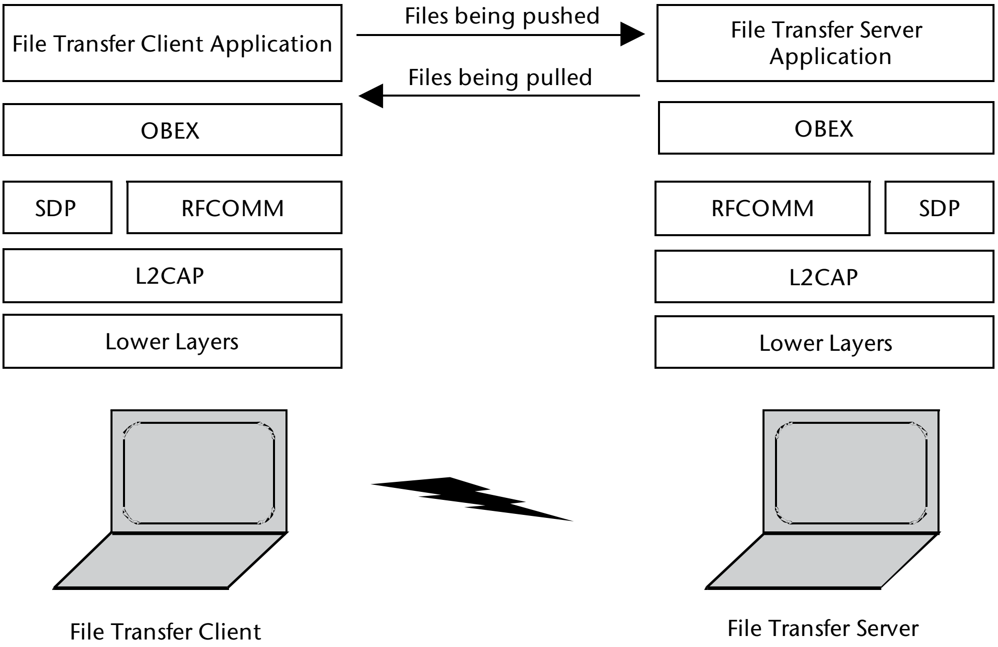

Figure 4.20 File transfer profile.

4.16 Generic Audio/Video Distribution Profile (GAVDP)

This profile defines the requirements for Bluetooth devices to support streaming channels for supporting audio/video distribution on ACL channels. It is dependent on GAP and uses AVDTP, L2CAP and lower layers.

GAVDP defines the following two roles. These are shown in Figure 4.21:

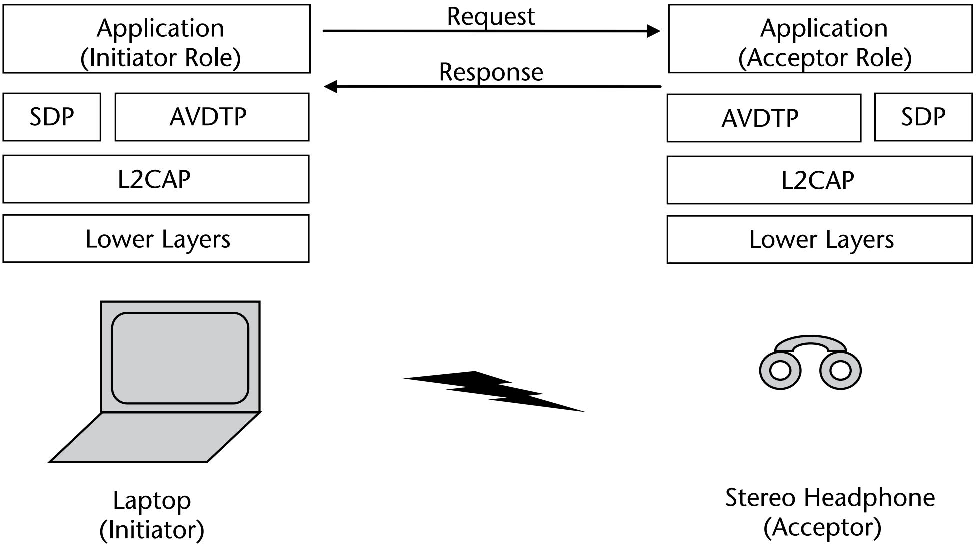

Figure 4.21 Generic audio video distribution profile.

- Initiator (INT): This is the device that initiates the GAVDP signaling procedures.
- Acceptor (ACP): This is the device that responds to the incoming requests from the INT.

A typical use case of the profile is the streaming of audio between a laptop and headphones. In this case, the laptop may act as the INT. It would send request to establish a streaming channel, negotiate parameters, control the stream etc. The headphones would act as the ACP and respond to the requests made by the INT.

GAVDP provides support for the following two scenarios:

- Setup the two devices for A/V data streaming and then create a connection between these two devices.
- Control the established streaming connection.

The detailed features and procedures supported by this profile are described in Table 4.7.

4.17 Advanced Audio Distribution Profile (A2DP)

This profile defines the requirements for Bluetooth devices to support high quality audio distribution. It uses the ACL channels for distribution of high quality audio. This is in contrast to the HF profile in which the SCO channels are used to transfer voice.

ACL channels are used because the bandwidth that is provided by SCO and eSCO channels is not sufficient to transfer the high quality audio data. In fact if raw audio data were to be streamed, then even the ACL channels don't have sufficient bandwidth. So, a codec is used to encode the data before sending and then decoding the data after it is received on the remote side.

Table 4.7 GAVDP Features and Procedures

Feature	*Procedure*	*Purpose*
Connection	Connection Establishment	This procedure is used when a device needs to create a connection with another device. It includes AVDTP procedures for finding the stream end points and getting the capabilities.
	Start Streaming	This procedure is used when both the devices are ready to start streaming and is used to start or resume streaming.
	Connection Release	This procedure is used to release the stream.
Transfer Control	Suspend	This procedure is used to suspend the A/V stream.
	Change Parameters	This procedure is used to change the service parameters of the stream.
Signaling Control	Abort	This procedure may be used to recover from a loss of signaling message.
Security Control	Security Control	This procedure is used to exchange security control messages between the two devices.