BLUETOOTH DOC	Date / Year-Month-Day	Approved	Revision	Document No
	2001-01-31	Draft	**0.95a**	X.Y.000/0.0
Prepared	e-mail address			N.B.
Arun Ayyagari	aruna@microsoft.com			Confidential

Bluetooth ESDP for UPnP

Abstract:

This document is a Bluetooth Extended Service Discovery Profile (ESDP) for Universal Plug and Play™ (UPnP™). The profile defines how devices with Bluetooth wireless communications can use the Bluetooth Service Discovery Protocol (SDP) initially to discover other devices that support UPnP services and retrieve information about these services. This profile further defines how a device with Bluetooth wireless communications can support UPnP services over the Bluetooth protocol stack using the Logical Link Control and Adaptation Protocol (L2CAP) layer and/or an Internet Protocol (IP) stack using either the Personal Area Network (PAN) Profile or the Local Area Network (LAN) Access Profile.

Revision History

Revision	Date	Comments
0.40	June 02, 2000	Initial draft.
0.49	June 29, 2000	Incorporate comments from ESDP face-to-face meeting on 12th and 13th June 2000.
0.50	July 20, 2000	Incorporate review comments.
0.51	August 11, 2000	Incorporate review comments.
0.70	August 24, 2000	Incorporate comments from ESDP face-to-face meeting on 17th and 18th August 2000.
0.71	August 24, 2000	Include reference for LAN Access Profile test specification.
0.90	October 04, 2000	Incorporate comments from ESDP face-to-face meeting on 3rd and 4th October 2000.
0.91	November 07, 2000	Incorporate review comments.
0.92	November 21, 2000	Incorporate review comments.
0.93	December 18, 2000	Incorporate comments from ESDP face-to-face meeting on 4th December 2000.
0.95	January 12, 2001	Incorporate review comments.
0.95a	January 31, 2001	Incorporate review comments.

Contributors

Arun Ayyagari	Microsoft Corporation
Salim AbiEzzi	Microsoft Corporation
Ned Plasson	3Com
Brent Miller	IBM Corporation
Chatschik Bisdikian	IBM Corporation
Johannes Elg	Ericsson Mobile Communications AB
Dale Farnsworth	Motorola
Srikanth Kambhatla	Intel Corporation
Markus Schetelig	Nokia Mobile Phones
Thomas Mueller	Nokia Mobile Phones
Franklin Reynolds	Nokia Mobile Phones
Toru Homma	Toshiba
Brian Redding	Motorola
Gerrit Slot	Ericsson Mobile Communications AB
Thierry Walrant	Philips Consumer Electronics
Markku Tamski	Nokia Mobile Phones
Ted Hartzell	Axis Communications
Shigeo Kohno	Toshiba
Sailesh Rachabathuni	Philips Consumer Communications
Toshiki Kizu	Toshiba
Graham Hamilton	Sun Microsystems
Om Sharma	Microsoft Corporation
Philip Mooney	Lucent
Willy Sagefalk	Axis Communications
Ramesh Caushik	Intel Corporation

Disclaimer and copyright notice

Contents

1 Introduction

1.1 Scope

Bluetooth Service Discovery Protocol (SDP) enables the discovery of available services in a Bluetooth system. SDP provides the mechanism for a device with Bluetooth wireless communications to locate services offered by other such devices. Since Logical Link Control and Adaptation Protocol (L2CAP) does not provide robust networking functions, it limits the discovery of services to the active devices in a given Bluetooth piconet. In a networked computing environment, it is desirable for devices to be able to discover services beyond their current Bluetooth piconet to extend the functionality of the device. In addition, extended service discovery could enable devices with Bluetooth wireless technology to control remote resources on other devices (that may or may not employ Bluetooth wireless communications) within and outside their piconets.

Universal Plug & Play (UPnP) is a distributed, open networking architecture that is designed to enable simple, ad hoc communication among distributed devices and services. UPnP leverages TCP/IP and the web model to provide seamless, media independent, peer-to-peer device connectivity and control. UPnP is a computing, electronics, telephony, and networking industry initiative designed to enable connectivity among standalone devices and Personal Computers (PCs) from many different vendors. These UPnP characteristics make it ideally suitable as a Bluetooth ESDP that is intended to provide an enhanced mechanism for service discovery and control within Bluetooth environments.

This profile defines two approaches to implementing UPnP within a Bluetooth system:

- The first approach ("L2CAP-based solution") shall focus on layering UPnP services over the L2CAP layer of the Bluetooth protocol stack for use by devices that lack IP support.

- The second approach ("IP-based solution") shall focus on layering UPnP services over the Bluetooth Personal Area Networking (PAN) Profile and the Bluetooth Local Area Network (LAN) Access Profile; these profiles define IP support in the Bluetooth protocol stack.

While outside the scope of this profile specification, both approaches might also include bridge devices that could provide interconnectivity between devices with Bluetooth wireless communications, both with and without IP support, and devices that provide UPnP services across the bridge to

enhance service discovery and control functionality. Appendix A provides an overview of the bridge device operation.

1.2 Definitions

L2CAP-based solution: By an L2CAP-based solution we mean UPnP over the L2CAP layer of the Bluetooth protocol stack. This configuration does not include an IP stack and the UPnP messages are transmitted directly over a connection-oriented L2CAP channel between peer devices with Bluetooth wireless communications.

IP-based solution: By an IP-based solution we mean UPnP over an IP stack provided by either PAN or LAN Access profiles. In this configuration the UPnP messages are transmitted appropriately over TCP or UDP layers on top of the IP stack provided by the PAN or LAN Access profiles.

Control point: The control point consists of a set of software modules that enables communication with UPnP devices (defined below) [1]. A control point initiates discovery and communication with UPnP devices, and receives events from UPnP devices. Control points are typically implemented on devices that have a user interface. This user interface is used to interact with UPnP devices over the network.

UPnP device: The UPnP device consists of a set of software modules that enables communication with a UPnP control point [1]. UPnP devices respond to discovery requests, accept incoming communications from control points and may send events to control points.

Local Device (LocDev): A LocDev is the device that initiates the service discovery process. A LocDev must contain at least the client portion for Bluetooth SDP. 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 for Bluetooth SDP. A RemDev contains a service record database, which the server portion of SDP consults to create responses to service discovery requests.

1.3 Profile Dependencies

1.3.1 L2CAP-based Solution

The L2CAP-based solution is dependent upon the Generic Access Profile (as are all profiles). While there is no direct dependency upon the Service Discovery Application Profile (SDAP), that profile does provide guidance relevant to the SDP discovery facets of the ESDP for UPnP.

1.3.2 IP-based Solution

The IP-based solution is dependent upon at least one of the IP profiles (PAN or LAN Access), as described below.

1.3.2.1 IP-based Solution using PAN Profile

Figure 1.1 depicts the IP-based solution's dependencies on the PAN profile. The IP-based solution using PAN profile is dependent upon the PAN profile that in turn is dependent on other profiles.

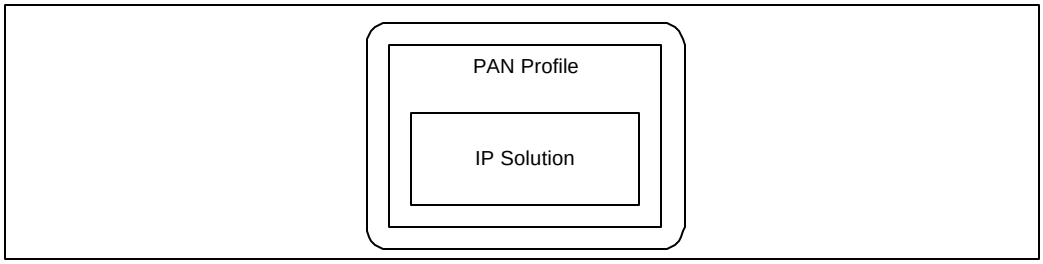

Figure 1.1: Bluetooth Profiles -- IP-based Solution using PAN

1.3.2.2 IP-based Solution using LAN Access Profile

Figure 1.2 depicts the IP-based solution's dependencies on the LAN Access profile. The IP-based solution using LAN Access profile is dependent upon the LAN Access profile, which in turn is dependent upon the Serial Port and Generic Access profiles; it also has the same relationship to SDAP noted above for the L2CAP-based solution.

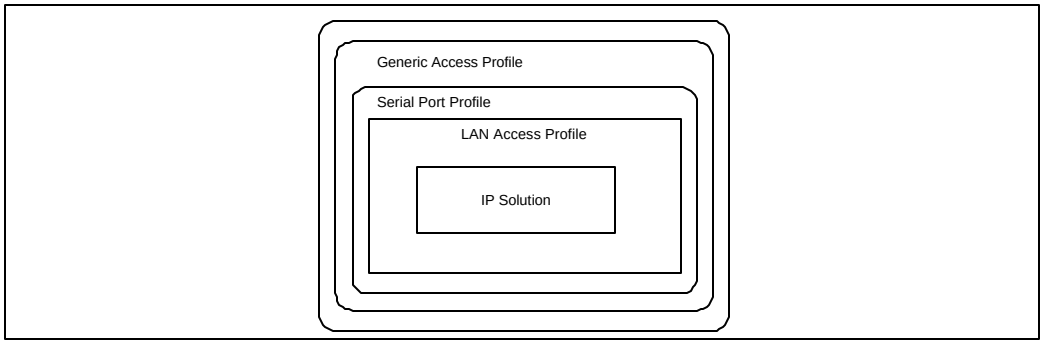

Figure 1.2: Bluetooth Profiles – IP-based Solution using LAN Access Profile

1.4 Symbols and Conventions

This profile uses the symbols and conventions specified in Section 1.2 of the Generic Access Profile [4] or [5].

2 Profile Overview

2.1 Profile Stack

The protocols for communication between UPnP control points and devices as defined by the UPnP Device Architecture [1] are shown in Figure 2.1[1]. However, by definition, the protocols referred to as UPnP in this profile are shown in Figure 2.2. This definition of the UPnP protocol stack does not include the UDP/TCP and IP layers described in [1].

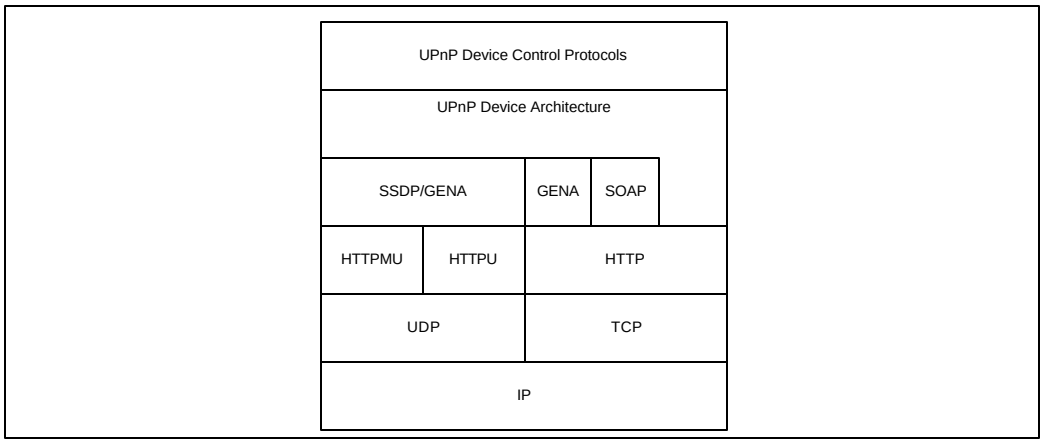

Figure 2.1: UPnP Device Architecture Protocol Stack

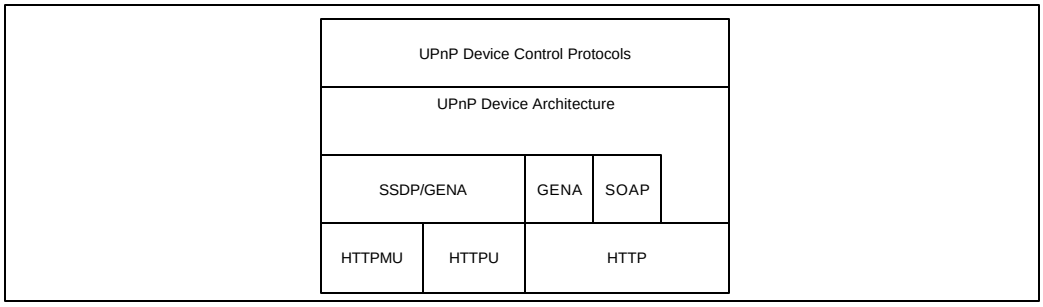

Figure 2.2: UPnP Protocol Stack

UPnP Device Control Protocols are contained at the highest layer of the UPnP device architecture protocol stack. Device control protocols transfer UPnP vendor-specific information about the device and information defined by the UPnP Forum working committees. Messages from the UPnP Device Control Protocols are hosted in UPnP-specific protocols at the UPnP Device Architecture layer. In turn, the messages from the UPnP Device Architecture layer are formatted using Simple Service Discovery Protocol (SSDP), General

[1] Note that the UPnP Device Architecture [1]is evolving and hence the protocol stack in Figure 2.1 and Figure 2.2 may differ from the representation in future versions of the UPnP Device Architecture.

Event Notification Architecture (GENA), and Simple Object Access Protocol (SOAP) and delivered via HTTP, either through multicast (HTTPMU) or unicast (HTTPU) variety or standard HTTP. A subset of the overall UPnP protocol stack depicted in Figure 2.1 and Figure 2.2 is used for UPnP networking functions consisting of discovery, description, control, eventing, and presentation. Details of the functions and the protocols used are described in the UPnP Device Architecture [1].

2.1.1 L2CAP-based Solution

Figure 2.3 shows the Bluetooth protocols and supporting entities for the L2CAP-based solution.

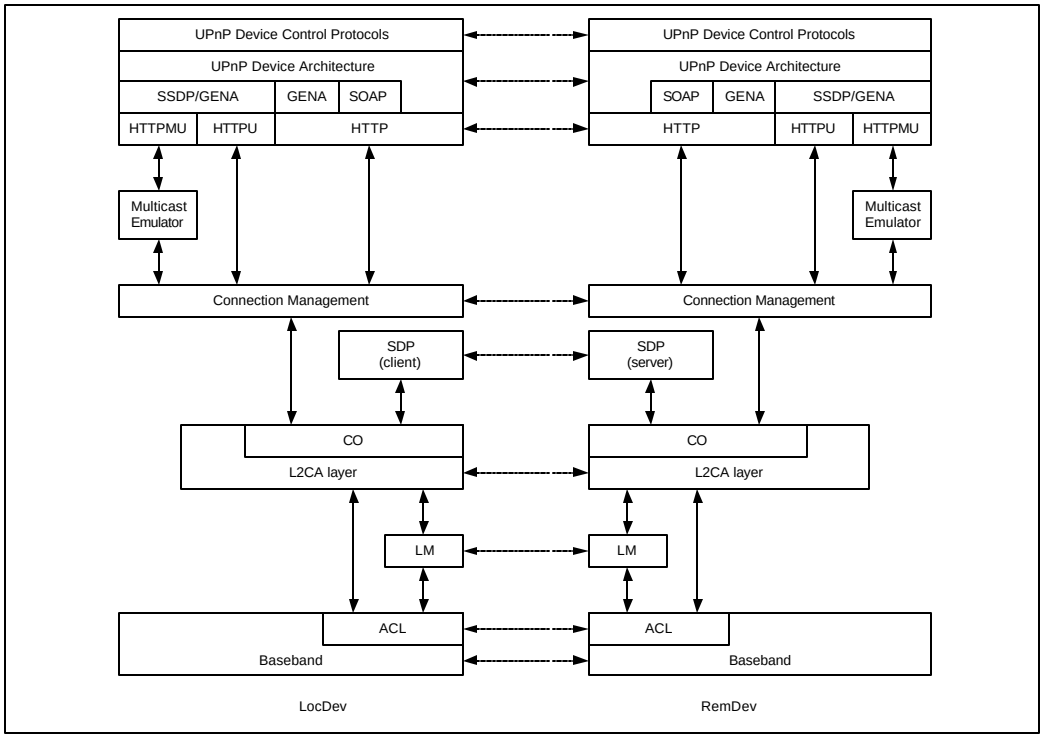

Figure 2.3: L2CAP-based Solution

The UPnP protocol stack in the LocDev or the RemDev may either be a UPnP control point or a UPnP device. UPnP uses the connection-oriented transport service in L2CAP, which in turn uses the baseband asynchronous connectionless (ACL) links to carry UPnP messages over the air-interface. Discovery of UPnP services is performed by SDP. Service discovery entails discovering Bluetooth services in proximity after establishing an L2CAP connection. One of the services that can be discovered using Bluetooth SDP is the capability of a device to enable or even support directly other discovery protocols, including UPnP.

2.1.2 IP-based Solution

2.1.2.1 IP-based Solution using PAN Profile

Figure 2.4 shows the Bluetooth protocols and supporting entities for the IP-based solution using the PAN profile.

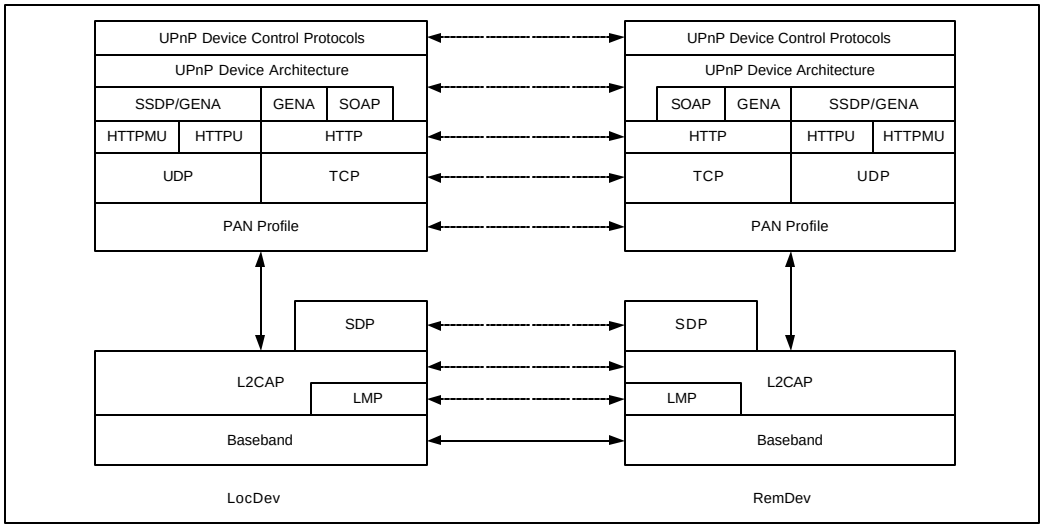

Figure 2.4: IP-based Solution using PAN Profile

UPnP uses the TCP and UDP transport services over the IP stack provided by the PAN profile. Initial discovery for the availability of UPnP services is performed using SDP. Following the discovery of UPnP services within LocDev and RemDev, further networking functions such as discovery, description, control, eventing, and presentation can be performed using the UPnP services.

In another scenario it may be possible that the peer device with IP support may not have UPnP services but may be able to reach another device that does provide UPnP services (perhaps in another piconet or across a bridge device). In this case, once IP connectivity between the devices is established the discovery of UPnP services can be performed using the UPnP service discovery mechanisms SSDP and GENA.

Note that the discovery of UPnP via SDP only implies the availability of UPnP services on the immediate peer device. Lack of UPnP on an immediate peer device that has IP support does not imply that the device cannot avail itself of UPnP services (since an IP stack can enable use of UPnP). In such cases the device can still perform UPnP service discovery using SSDP and GENA. Hence discovery of UPnP via SDP may assist in expediting the discovery of desired UPnP services.

2.1.2.2 IP-based Solution using LAN Access Profile

Figure 2.5 shows the Bluetooth protocols and supporting entities for IP-based solution using the LAN Access profile

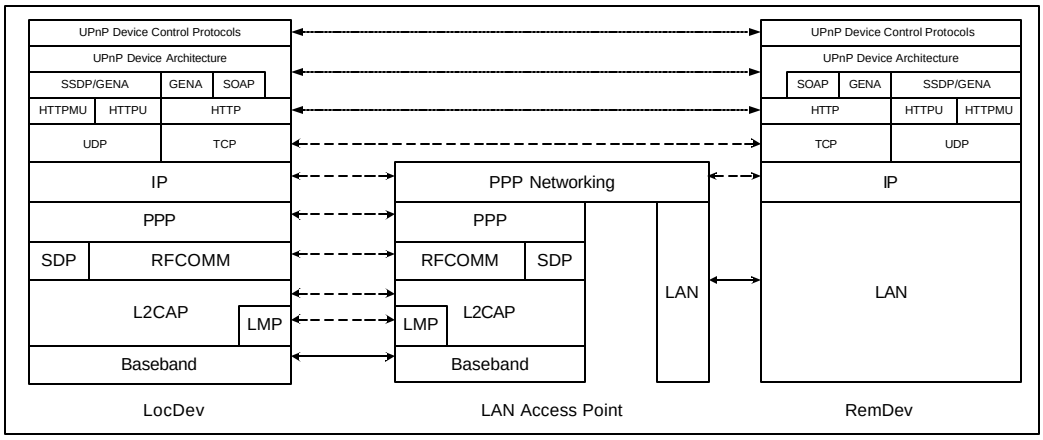

Figure 2.5: IP-based Solution using LAN Access Profile

UPnP uses the TCP and UDP transport services over the IP support provided by the LAN Access profile. In this scenario SDP may not be used to discover UPnP services (although it is likely to be used to discover support for the LAN Access profile). Discovery of UPnP services across the LAN Access Point can be performed using the UPnP service discovery mechanisms SSDP and GENA.

2.2 User Requirements and Scenarios

UPnP Device Architecture [1] defines the protocols and procedures that shall be used by devices to use UPnP networking functions. This profile implies the availability of UPnP networking functions using an L2CAP-based solution and/or an IP-based solution.

2.2.1 L2CAP-based Solution

The following scenarios are covered by this profile:

- Two devices with the L2CAP-based solution establish a peer-to-peer connection. Once connected, the devices can avail themselves of all the services provided by UPnP networking functions.

- A device with an L2CAP-based solution establishes a peer-to-peer connection with a device supporting UPnP services. This peer device may be the bridge device if it supports UPnP services or a device supporting UPnP services that can be reached via a bridge device. The bridge device may also proxy UPnP services available across the bridge. Once

connected, the devices can avail themselves of all the services provided by UPnP networking functions.

2.2.2 IP-based Solution

2.2.2.1 IP-based Solution using PAN Profile

The following scenarios are covered by this profile:

- Two devices with the IP-based solution using PAN profile establish a peer-to-peer IP connectivity. Once connected, the devices can avail themselves of all the services provided by UPnP networking functions.

- A device with an IP-based solution using PAN profile establishes peer-to-peer IP connectivity with a device that may not support UPnP but does provide IP access to other devices that provide UPnP services. Once connected, the device with the IP-based solution using PAN profile can avail itself of UPnP services so long as the other device remains connected to the devices offering UPnP services.

- A device with an IP-based solution using PAN profile establishes a peer-to-peer connection with a device supporting UPnP services on or across a bridge device. The bridge device shall proxy UPnP services available across the bridge. Once connected, the devices can avail themselves of all the services provided by UPnP networking functions.

2.2.2.2 IP-based Solution using LAN Access Profile

The following scenario is covered by this profile:

- A device with an IP-based solution using LAN Access profile establishes peer-to-peer IP connectivity with a LAN Access Point that may not support UPnP but does provide IP access to other devices that provide UPnP services. Once connected, the device with the IP-based solution using LAN Access profile can avail itself of UPnP services so long as the LAN Access Point remains connected to the devices offering UPnP services.

2.3 Profile Fundamentals

Before any two devices with Bluetooth wireless communications can communicate with each other the following steps may be required:

- The device needs to be powered-on and initialised. Initialisation may require providing a PIN for the creation of a link key, for device authentication and data encryption.

- Discovery of another device's BD_ADDR via the inquiry process, and the paging of (an)other device(s) to establish a connection.

This 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 and UPnP will be performed only on the subset of devices that pass authentication and security requirements imposed by each other and that have compatible ESDP for UPnP present in each device.

2.3.1 L2CAP-based Solution

A brief summary of the interactions in establishing peer-to-peer UPnP networking connectivity between devices supporting UPnP using L2CAP-based solution is given below. Subsequent sections in this profile provide more detail for each of the following steps.

1. Discover another device's BD_ADDR via inquiry process and paging of other device(s).

2. If necessary, establish Bluetooth links with these devices and using SDP find a device that provides UPnP services using the L2CAP-based solution. The LocDev may either query multiple RemDevs to determine whether or not they provide UPnP services using L2CAP-based solution, or it may stop after encountering a RemDev that provides the desired services. This is an implementation detail and hence is beyond the scope of this profile.

3. Select a RemDev that provides UPnP services using L2CAP-based solution and establish a baseband physical link to the selected device if one does not exist.

4. The devices establish a peer-to-peer L2CAP connection.

5. Peer-to-peer UPnP networking connectivity is established.

6. At any time either of the devices may terminate the L2CAP connection thereby also terminating the peer-to-peer UPnP networking connection. If however the peer-to-peer UPnP networking connection is terminated the underlying L2CAP connection may or may not also be terminated. The management of the L2CAP connection by the connection management layer is an implementation detail and hence is beyond the scope of this profile.

2.3.2 IP-based Solution

2.3.2.1 IP-based Solution using PAN Profile

A brief summary of the interactions in establishing peer-to-peer UPnP networking connectivity between a device supporting UPnP services using IP-based solutions and a device across an IP network that supports UPnP services is given below. Note that in this case the device(s) within radio range of the device supporting UPnP services using the IP-based solution shall provide IP support using PAN profile but may not provide UPnP services. Subsequent sections in this profile provide more detail for each of the following steps.

1. Discover another device's BD_ADDR via inquiry process and paging of other device(s).

2. If necessary, establish Bluetooth links with these devices and using SDP find a RemDev that provides IP support using the PAN profile; also using SDP determine whether or not that device also provides UPnP services. The LocDev may either query all RemDevs or it may stop after encountering a RemDev that provides the desired services. This is an implementation detail and hence is beyond the scope of this profile.

3. Select a RemDev that provides IP support using the PAN profile and preferably also provides UPnP services and establish a baseband physical link to the selected device if one does not exist.

4. The devices establish a peer-to-peer PAN/L2CAP connection that includes the negotiation of a suitable IP address.

5. Once the connection is established, peer-to-peer IP traffic can flow between the devices.

6. Peer-to-peer UPnP networking connectivity is established using the TCP and UDP transport stack over the IP stack provided by the PAN profile if the peer device also provides UPnP services. Once connected, the devices can avail themselves of all the services provided by UPnP networking functions.

   If the RemDev does not support UPnP services, the LocDev can still initiate UPnP services once the IP connectivity has been established. Following the establishment of IP connectivity, the devices can initiate the UPnP networking functions starting with the discovery of UPnP services using SSDP and GENA.

7. At any time either of the devices may terminate the PAN connection thereby also terminating the peer-to-peer UPnP networking connection. If

however the peer-to-peer UPnP networking connection is terminated the underlying PAN connection may or may not also be terminated. The management of the PAN connection is an implementation detail and hence is beyond the scope of this profile.

2.3.2.2 IP-based Solution using LAN Access Profile

A brief summary of the interactions in establishing peer-to-peer UPnP networking connectivity between a device supporting UPnP services using IP-based solution and a device across an IP network that supports UPnP services is given below. Note that in this case the device(s) within radio range of the device supporting UPnP services shall provide IP support but may not provide UPnP services. Subsequent sections in this profile provide more detail for each of the following steps.

1. Discover another device's BD_ADDR via inquiry process and paging of other device(s).

2. If necessary, establish Bluetooth links with these devices and using SDP find a RemDev that provides IP support using the LAN Access profile; also using SDP determine whether or not that device also provides UPnP services. The LocDev may either query multiple RemDevs or it may stop after encountering a RemDev that provides the desired services. This is an implementation detail and hence is beyond the scope of this profile.

3. Select a RemDev that provides IP support using the LAN Access profile. In addition to providing IP support, the RemDev (LAN Access Point) may also provide UPnP services, i.e., UPnP services are hosted on the LAN Access Point. The LocDev may preferably select a RemDev that provides UPnP services in addition to IP support over another RemDev that only provides IP support. Following the selection of the desired RemDev, the LocDev shall establish a baseband physical link to the selected RemDev if one does not exist.

4. The devices establish a peer-to-peer connection that includes the negotiation of a suitable IP address.

5. Once the connection is established, peer-to-peer IP traffic can flow between the devices.

6. Peer-to-peer UPnP networking connectivity is established using the TCP and UDP transport stack over the IP stack provided by the LAN Access profile if the device also provides UPnP services. Once connected, the devices can avail themselves of all the services provided by UPnP networking functions.

If the RemDev does not support UPnP services, the LocDev can still initiate UPnP services once the IP connectivity has been established. Following the establishment of IP connectivity, the devices can initiate the UPnP networking functions starting with the discovery of UPnP services using SSDP and GENA.

7. At any time either of the devices may terminate the LAN connection thereby also terminating the peer-to-peer UPnP networking connection. If however the peer-to-peer UPnP networking connection is terminated the underlying LAN connection may or may not also be terminated. The management of the LAN connection is an implementation detail and hence is beyond the scope of this profile.

3 Conformance

Conformance to this profile can be claimed if either L2CAP-based solution and/or IP-based solution using PAN profile and/or IP-based solution using LAN Access profile is/are implemented. Note that conformance to a particular solution stated above only ensures interoperability between peer devices using the same type of solution and does not provide interoperability between peer devices using different types of solutions. If conformance to this profile is claimed, all capabilities indicated as mandatory for the L2CAP-based solution and/or the IP-based solution using PAN profile and/or the IP-based solution using LAN Access profile shall be supported in the specified manner (process mandatory). This also applies for all optional and conditional capabilities for which support is indicated and shall be implemented as specified. Conformance is subject to verification as part of the Bluetooth qualification program.

4 Compatibility

Compatibility in this profile defines whether the ESDP for UPnP profile in peer devices are compatible at the application level. Note this assumes conformance between the peer devices, that is, they are interoperable using the same type of solution as stated in Section 3. A minimum requirement for application level compatibility to this profile is the capability for devices involved in executing the profile to retrieve and interpret the profile and profile version number service record and attribute from one another. This minimal requirement does not ensure compatibility for other required protocols and profiles specified by the service record.

4.1 VersionNumberList Attribute

Attribute Name	Attribute ID	Attribute Value Type
VersionNumberList	0x0200	Data Element Sequence

The VersionNumberList is a data element sequence in which each element of the sequence is a version number supported by the device advertising the UPnP service.

A version number is a 16-bit unsigned integer consisting of two fields. The higher-order 8 bits contain the major version number field and the low-order 8 bits contain the minor version number field. Version numbers start at 0 and the minor version number is reset to 0 after a major version number increment. The initial version of ESDP for UPnP profile shall specify the current version of the UPnP Device Architecture specification [1]. When upward compatible changes are made to the protocol, the minor version will be incremented. If incompatible changes are made to ESDP for UPnP profile, the major version number will be incremented. If two implementations of this profile with version numbers [major_A:minor_A] and [major_A:minor_B], then these implementation shall provide full support of all the related protocols and protocol features required by the minimal minor version number implementation of the same major version number, denoted as [major_A;0]. This guarantees that implementations of the profile with the same major version number will always interoperate.

5 Test Strategy

Key steps to verifying the ESDP for UPnP services functionality are stated in this section.

5.1 L2CAP-based Solution

Test strategy for L2CAP-based solution shall focus on service discovery, connection management and multicast emulator functionality, and operation of UPnP services. Testing for this profile shall be limited to evaluating the required Bluetooth functionality in support of the ESDP for UPnP profile. In particular the test strategy shall not address functionality specific to UPnP since it is assumed that test procedures for UPnP functionality will follow the guidelines established by the UPnP Forum. Additionally, this profile shall also rely on the test specifications defined for other required Bluetooth components of the profile stack such as the test specifications for SDP [6] and SDAP [7]. Test specification for SDP primarily focuses on conformance testing while the test specification for SDAP focuses on interoperability testing. The SDP and SDAP test specifications provide the test purposes and test cases for service discovery. Test specification for connection management, multicast emulator, and HTTP address mapping shall be based on interoperability testing. List of tested functionality is given below.

- Retrieve SDP service record for ESDP for UPnP services.
- Verify the functionality of connection management layer.
- Verify the functionality of the multicast emulator layer.
- Verify HTTP address mapping for point-to-point HTTP/HTTPU request and point-to-multipoint HTTPMU request.
- Select the RemDev and initiate operation of UPnP services.

5.2 IP-based Solution

5.2.1 IP-based Solution using PAN Profile

Test strategy for IP-based solution using PAN profile shall focus on service discovery, PAN profile functionality, and operation of UPnP services. Testing for this profile shall be limited to evaluating the required Bluetooth functionality in support of the ESDP for UPnP profile. In particular the test strategy shall not address functionality specific to UPnP since it is assumed that test procedures for UPnP functionality will follow the guidelines established by the UPnP Forum. Additionally, this profile shall also rely on the test specifications

defined for other required Bluetooth components of the profile stack such as the test specifications for SDP [6], SDAP [7] and PAN profile [8]. Test specification for SDP primarily focuses on conformance testing while the test specification for SDAP focuses on interoperability testing. The SDP and SDAP test specifications provide the test purposes and test cases for service discovery. Test specification for PAN profile functionality shall be based on test purposes defined by the PAN profile. List of tested functionality is given below.

- Retrieve SDP service record for IP connectivity.
- Optionally retrieve SDP service record for UPnP services.
- Verify the PAN profile functionality.
- Select the RemDev and initiate operation of UPnP services.

5.2.2 IP-based Solution using LAN Access Profile

Test strategy for IP-based solution using LAN Access profile shall focus on service discovery, LAN Access profile functionality, and operation of UPnP services. Testing for this profile shall be limited to evaluating the required Bluetooth functionality in support of the ESDP for UPnP profile. In particular the test strategy shall not address functionality specific to UPnP since it is assumed that test procedures for UPnP functionality will follow the guidelines established by the UPnP Forum. Additionally, this profile shall also rely on the test specifications defined for other required Bluetooth components of the profile stack such as the test specifications for SDP [6], SDAP [7] and LAN Access Profile [9]. Test specification for SDP primarily focuses on conformance testing while the test specification for SDAP focuses on interoperability testing. The SDP and SDAP test specifications provide the test purposes and test cases for service discovery. Test specification for LAN Access profile functionality shall be based on test purposes defined by the LAN Access profile. List of tested functionality is given below.

- Retrieve SDP service record for IP connectivity.
- Optionally retrieve SDP service record for UPnP services.
- Verify the LAN Access profile functionality.
- Select the RemDev and initiate operation of UPnP services.

6 List of Test Purposes (TP)

ESDP for UPnP profile is dependent on the successful validation of the required Bluetooth profile stack components and the UPnP application functionality. TPs are also intended to verify the device conformance to the ESDP for UPnP profile and interoperability with other compliant devices.

6.1 L2CAP-based Solution

TPs shall not address functionality specific to UPnP since it is assumed that test procedures for UPnP functionality will follow the guidelines established by the UPnP Forum. Additionally, this profile shall also rely on the test specifications defined for other required Bluetooth components of the profile stack such as the test specifications for SDP and SDAP.

6.1.1 Connection Management

The objective of this test set is to verify that the Implementation Under Test (IUT) can connect with a RemDev and that the established connection adheres to the defined flow control mechanism.

6.1.1.1 Establish Connection

1. Verify that IUT is able to establish a connection management session to a RemDev.

2. Verify that IUT is able to connect to a trusted RemDev when both IUT and RemDev simultaneously initiate the establishment of connection management session.

6.1.1.2 Terminate Connection

1. Verify that IUT is able to terminate an established connection management session to a RemDev.

6.1.1.3 Flow Control

1. Verify that IUT is able to advertise to a RemDev an increase or a decrease in the receive window size for an established connection management session.

2. Verify that IUT is able to change its transmit window size when it receives an increase or a decrease in the advertised receive window size from the RemDev for an established connection management session.

6.1.2 Multicast Emulator

The objective of this test subgroup is to verify that when the IUT receives an HTTPMU message from its higher layer, for e.g., an application, it translates the multicast request into multiple unicast connection management session requests, one per RemDev in its radio range. Following the establishment of the connection management sessions, IUT transmits the HTTPMU message to each RemDev over the established connection(s).

6.1.2.1 Establish Multicast Connections

1. Verify that when the IUT receives an HTTPMU message from its higher layer, it translates the multicast message into multiple unicast connection management session requests, one connection management session per RemDev.

6.1.2.2 Terminate Multicast Connections

1. Verify that IUT terminates the established multiple unicast connection management sessions to RemDevs.

6.2 IP-based Solution

6.2.1 IP-based Solution using PAN Profile

TPs shall not address functionality specific to UPnP since it is assumed that test procedures for UPnP functionality will follow the guidelines established by the UPnP Forum. Additionally, this profile shall also rely on the test specifications defined for other required Bluetooth components of the profile stack such as the test specifications for SDP, SDAP, and PAN profile.

6.2.2 IP-based Solution using LAN Access Profile

TPs shall not address functionality specific to UPnP since it is assumed that test procedures for UPnP functionality will follow the guidelines established by the UPnP Forum. Additionally, this profile shall also rely on the test specifications defined for other required Bluetooth components of the profile stack such as the test specifications for SDP, SDAP, and LAN Access profile.

7 Application Layer

This section describes the feature requirements for devices with Bluetooth wireless communications using ESDP for UPnP services. The first subsection focuses on the feature requirements for UPnP services using the L2CAP-based solution while the second subsection focuses on the feature requirements for UPnP services using the IP-based solution.

7.1 L2CAP-based Solution

Section	Feature	LocDev	RemDev
7.1.1	Discover UPnP services	M[2]	O
7.1.1	Advertise UPnP services	O	M[2]
7.1.2	Operation of UPnP services	M[2]	M[2]

7.1.1 Discover and Advertise UPnP Services

This procedure is initiated by the LocDev to discover UPnP services provided by RemDev(s) within radio range. The RemDev shall advertise the availability of UPnP services via SDP service records if these services are provided. The LocDev shall use Bluetooth SDP mechanisms to discover and retrieve service information.

1. The LocDev first searches for RemDevs, via Bluetooth inquiry procedure.

2. For each RemDev found, the LocDev may connect to it, perform any necessary link setup, and then query it for UPnP services. The LocDev may choose to terminate this step at any point following the first device queried.

The result of the selection process is RemDev(s) responding with UPnP services using the L2CAP-based solution.

7.1.2 Operation of UPnP services

Following the discovery of UPnP services the LocDev establishes a peer-to-peer UPnP networking connection to the selected RemDev(s) that provide

[2] This feature is mandatory for devices that support the UPnP L2CAP-based solution. However, using the L2CAP-based solution is not mandatory for a device with Bluetooth wireless communications. That is, this part of the profile is optional but when supported, this feature is mandatory within it.

UPnP services using the L2CAP-based solution This might be done for each selected RemDev.

1. The LocDev initiates UPnP services starting with UPnP service discovery using SSDP and GENA.

- UPnP protocol stack messages are communicated via HTTPMU, HTTPU, and standard HTTP messages.

- The application shall use connection-oriented services over L2CAP to establish peer-to-peer connectivity between the LocDev and RemDev. Since HTTP transactions comprise a sequence of service messages that constitutes a connectionless datagram service, prior to an actual HTTP message exchange to occur, an L2CAP connection to carry the HTTP traffic will have to be established, if one does not already exist. The UPnP service application delegates the creation of connections on its behalf to the connection management and multicast emulator layers that also have the responsibility to request the L2CAP layer to establish and terminate these connections on its behalf as well.

2. The UPnP protocol stack shall use BD_ADDR in place of source and destination IP address in its messages. BD_ADDR for each device with Bluetooth wireless communications is unique and hence is an ideal proxy for source and destination IP addresses in the UPnP messages using HTTP syntax. Addressing format based on BD_ADDR for the L2CAP-based solution is described in Section 9.1.5.1.

7.2 IP-based Solution

7.2.1 IP-based Solution using PAN Profile

Section	Feature	LocDev	RemDev
7.2.1.1	Discover IP support[3]	M[4]	M[4]
7.2.1.2	Discover UPnP services	O	O
7.2.1.2	Advertise UPnP services	O	O
7.2.1.3	Operation of UPnP services	M[4]	O

[3] This is intended to support IP bridge functionality.
[4] This feature is mandatory for devices that support the UPnP IP-based solution using PAN Profile. However, the use of the IP-based solution using PAN Profile is not mandatory for a device with Bluetooth wireless communications. That is, this part of the profile is optional but when supported, this feature is mandatory within it.

7.2.1.1 Discover IP Support

This procedure is initiated by the LocDev to discover RemDevs within radio range that support IP connectivity via the PAN profile. This may include discovery of a bridge device. The LocDev shall use Bluetooth SDP to discover and retrieve service information.

1. The LocDev first searches for RemDevs, via Bluetooth inquiry procedure.

2. For each RemDev found, the LocDev may connect to it, perform any necessary link setup, and then query it for IP support. Note that a LocDev with IP service using the PAN profile will query for the corresponding support in the RemDevs. The LocDev may choose to terminate this step at any point following the first device queried.

- The application will be presented with a list of RemDevs that provide IP support using PAN profile.

- If multiple RemDevs provide IP support using PAN profile, the application may select one or more suitable RemDevs that provide the desired service. This selection process might involve additional SDP queries to determine if the selected RemDev(s) also support(s) UPnP services as described below.

The result of the selection process is one or more RemDevs with IP support using PAN profile.

7.2.1.2 Discover and Advertise UPnP Services

This procedure is initiated by the LocDev to discover UPnP services provided by RemDev(s) within radio range or across a Bluetooth bridge device or in another piconet. The LocDev shall use Bluetooth SDP mechanisms to discover and retrieve service information. The RemDev may advertise the availability of UPnP services via SDP service records if these services are provided.

1. For each RemDev known to provide IP support using the PAN profile, the LocDev will establish a Bluetooth link with it (if one does not exist yet), query it for UPnP services using SDP. The LocDev may choose to terminate this step at any point following the first device queried.

- The application will be presented with a list of RemDev(s) that are within radio range and provide UPnP services via IP-based solution using the PAN profile.

- If UPnP services via IP-based solution using PAN profile are provided by multiple RemDev(s), the application may select a suitable RemDev that provides the desired service.

The result of the selection process is a RemDev within radio range with UPnP services using IP-based solution. It is possible that none of the RemDevs may provide UPnP services. However, the LocDev can still initiate UPnP services with device(s) that are reachable via the selected RemDev, which is known to support IP via the PAN profile (from the process in Section 7.2.1.1) even if it does not support UPnP services.

7.2.1.3 Operation of UPnP Services

Following the establishment of IP connectivity and possibly the discovery of UPnP services, the LocDev initiates the UPnP networking functions starting with the discovery of UPnP services.

1. The LocDev initiates UPnP services starting with UPnP service discovery using SSDP and GENA.

- UPnP protocol stack messages are communicated via HTTPMU, HTTPU, and standard HTTP messages [1].

- The application shall use TCP and UDP services over IP stack to establish peer-to-peer connectivity between the LocDev and RemDev.

7.2.2 IP-based Solution using LAN Access Profile

Section	Feature	LocDev	RemDev
7.2.2.1	Discover IP support[5]	M[6]	M[6]
7.2.2.2	Discover UPnP services	O	O
7.2.2.2	Advertise UPnP services	O	O
7.2.2.3	Operation of UPnP services	M[6]	O

7.2.2.1 Discover IP Support

This procedure is initiated by the LocDev to discover RemDevs within radio range that support IP connectivity via the LAN Access profile. The LocDev shall use Bluetooth SDP to discover and retrieve service information.

[5] This is intended to support IP bridge functionality.
[6] This feature is mandatory for devices that support the UPnP IP-based solution using LAN Access Profile. However, the use of the IP-based solution using LAN Access Profile is not mandatory for a device with Bluetooth wireless communications. That is, this part of the profile is optional but when supported, this feature is mandatory within it.

1. The LocDev first searches for RemDevs, via Bluetooth inquiry procedure.

2. For each RemDev found, the LocDev may connect to it, perform any necessary link setup, and then query it for IP support. Note that a LocDev with IP service using LAN Access profile will query for the corresponding support in RemDevs. The LocDev may choose to terminate this step at any point following the first device queried.

- The application will be presented with a list of RemDevs that provide IP support using LAN Access profile.

- If multiple RemDevs provide IP support using LAN Access profile, the application may select one or more suitable RemDevs that provide the desired service. This selection process might involve additional SDP queries to determine if the selected RemDev(s) also support(s) UPnP services as described below.

The result of the selection process is one or more RemDevs with IP support using LAN Access profile.

7.2.2.2 Discover and Advertise UPnP Services

This procedure is initiated by the LocDev to discover UPnP services provided by RemDev(s) within radio range or across a Bluetooth bridge device or in another piconet. The LocDev shall use Bluetooth SDP to discover and retrieve service information.

1. For each RemDev known to provide IP support using the LAN Access profile, the LocDev will establish a Bluetooth link with it (if one does not exist yet), query it for UPnP services using SDP. The LocDev may choose to terminate this step at any point following the first device queried.

- The application will be presented with a list of RemDev(s) that are within radio range and provide UPnP services via IP-based solution using the LAN Access profile.

- If UPnP services via IP-based solution using LAN Access profile are provided by multiple RemDevs, the application may select a suitable RemDev that provides the desired service.

The result of the selection process is a RemDev within radio range with UPnP services using IP -based solution. It is possible that none of the RemDevs may provide UPnP services. However, the LocDev can still initiate UPnP services with device(s) that are reachable via the selected RemDev, which is known to support IP via the LAN Access profile (from the process in Section 7.2.2.1) even if it does not support UPnP services.

7.2.2.3 Operation of UPnP Services

Following the establishment of IP connectivity and possibly the discovery of UPnP services, the LocDev initiates the UPnP networking functions starting with the discovery of UPnP services.

1. The LocDev initiates UPnP services starting with UPnP service discovery using SSDP and GENA.

- UPnP protocol stack messages are communicated via HTTPMU, HTTPU, and standard HTTP messages [1].

The application shall use TCP and UDP services over IP stack to establish peer-to-peer connectivity between the LocDev and RemDev.

8 Service Discovery

In this section, SDP transactions for ESDP for UPnP service discovery are presented. SDP shall be used as the initial discovery mechanism to determine which of the RemDevs support UPnP services and/or profile requirements to enable UPnP service discovery mechanism.

8.1 SDP Service Records

A device with Bluetooth wireless communications may provide UPnP services using an L2CAP-based solution and/or an IP -based solution. The following sections define SDP service records for these alternative methods.

8.1.1 L2CAP-based Solution

Item	Definition	Mand./ Opt.	Type/Size	Value
ServiceClassIDList		Mand.		See [2] or [3]
ServiceClass0	"UPnP L2CAP-based Solution"	Mand.	UUID/32-bit	See [2] or [3]
ProtocolDescriptorList		Mand.		
Protocol0	L2CAP	Mand.	UUID/32-bit	L2CAP See [2] or [3]
ProtocolSpecificParameter0	PSM	Mand.	UInt16	PSM for UPnP (ESDP/Connection Management) Layer See [2] or [3]
Protocol 1	UPnP	Mand.	UUID	See [2] or [3]
ProfileDescriptor List				
Profile0	"UPnP over L2CAP-based Solution"	Mand.	UUID/32-bit	See [2] or [3]
Parameter0	Version number list	Mand.	DataElement/ Sequence	
ServiceName	Displayable text name	Opt.	DataElement/ String	"UPnP"
ServiceDescription	Displayable information	Opt.	DataElement/ String	"UPnP using L2CAP-based Solution"

The actual values of universal attribute IDs are defined in the Assigned Numbers section in [2] or [3]. Values that are of the type UUID are defined in the Assigned Numbers section in [2] or [3].

- The ServiceName attribute is a short user-friendly name for the service; e.g. "UPnP", etc.

- The ServiceDescription attribute is a longer description of the service. For example, "UPnP using L2CAP-based Solution."

8.1.2 IP-based Solution

8.1.2.1 IP-based Solution using PAN Profile

Item	**Definition**	**Mand./ Opt.**	**Type/Size**	**Value**
ServiceClassIDList		Mand.		See [2] or [3]
ServiceClass0	"UPnP IP-based Solution using PAN Profile"	Mand.	UUID/32-bit	See [2] or [3]
ProtocolDescriptorList		Mand.		
Protocol0		Mand.	DataElement/ Alternative	
UDP	UDP	Mand.	UUID/16-bit	UDP See [2] or [3]
TCP	TCP	Mand.	UUID/16-bit	TCP See [2] or [3]
ProfileDescriptor List		Mand.		
Profile0		Mand.	DataElement/ Alternative	
NAP	NAP	Mand.	UUID	NAP See [2] or [3]
GN	GN	Mand.	UUID	GN See [2] or [3]
Profile1	"UPnP using IP-based Solution"	Opt.	UUID/32-bit	See [2] or [3]
Parameter0	Version number list	Mand.	DataElement/ Sequence	
ServiceName	Displayable text name	Opt.	DataElement/ String	"UPnP"
ServiceDescription	Displayable information	Opt.	DataElement/ String	"UPnP using IP-based Solution"
IpSubnet	Displayable information	Opt.	DataElement/ String	

The actual values of universal attribute IDs are defined in the Assigned Numbers section in [2] or [3]. Values that are of the type UUID are defined in the Assigned Numbers section in [2] or [3].

- The ServiceName attribute is a short user-friendly name for the service; e.g. “UPnP”, etc.

- The ServiceDescription attribute is a longer description of the service. For example, “UPnP using IP-based Solution.”

- The IpSubnet attributeID is 0x0200. This attribute is a displayable string containing subnet definition of the network, e.g., “192.34.12.0/24”. The first 4 numbers define the IP subnet is dotted-decimal notation. The fifth number, after the “/” character, is the number of subnet bits to use in the subnet mask; e.g., 24 means a subnet mask of 255.255.255.0.

8.1.2.2 IP-based Solution using LAN Access Profile

Item	Definition	Mand./ Opt.	Type/Size	Value
ServiceClassIDList		Mand.		See [2] or [3]
ServiceClass0	"UPnP IP-based Solution using LAN Access Profile"	Mand.	UUID/32-bit	See [2] or [3]
ProtocolDescriptorList		Mand.		
Protocol0	IP	Mand.	UUID/16-bit	IP See [2] or [3]
Protocol1		Mand.	DataElement/ Alternative	
UDP	UDP	Mand.	UUID/16-bit	UDP See [2] or [3]
TCP ProfileDescriptor List	TCP	Mand. Mand.	UUID/16-bit	TCP See [2] or [3]
Profile0	"LAN Access Profile"	Mand.	UUID/32-bit	See [2] or [3]
Parameter0	Version "1.00"	Mand.	UInt16	
Profile1	"UPnP using IP-based Solution"	Opt.	UUID/32-bit	See [2] or [3]
Parameter0	Version number list	Mand.	DataElement/ Sequence	
ServiceName	Displayable text name	Opt.	DataElement/ String	"UPnP"
ServiceDescription	Displayable information	Opt.	DataElement/ String	"UPnP using IP-based Solution"
IpSubnet	Displayable information	Opt.	DataElement/ String	

The actual values of universal attribute IDs are defined in the Assigned Numbers section in [2] or [3]. Values that are of the type UUID are defined in the Assigned Numbers section in [2] or [3].

- The ServiceName attribute is a short user-friendly name for the service; e.g. “UPnP”, etc.

- The ServiceDescription attribute is a longer description of the service. For example, “UPnP using IP-based Solution.”

- The IpSubnet attributeID is 0x0200. This attribute is a displayable string containing subnet definition of the network, e.g., “192.34.12.0/24”. The first 4 numbers define the IP subnet is dotted-decimal notation. The fifth number, after the “/” character, is the number of subnet bits to use in the subnet mask; e.g., 24 means a subnet mask of 255.255.255.0.

9 L2CAP

9.1 L2CAP-based Solution

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

	L2CAP Procedure	**LocDev**	**RemDev**
1.	Channel types		
	Connection-oriented channel	M	M
	Connectionless channel	X[1]	X[1]
2.	Signalling		
	Connection Establishment Configuration	M M	M M
	Connection Termination	M	M
	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
4.	Connection Management	M	M
5.	Mapping of HTTP Messages to L2CAP	M	M
Comments:			
[X[1]]: This feature is not used in this profile, but its use by other applications running simultaneously with this profile is not excluded.			

9.1.1 Channel Types

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

9.1.2 Signalling

In the PSM field on the Connection Request packet, the value used shall indicate the request for creation of an L2CAP connection for accessing the UPnP protocol stack.

9.1.3 Configuration Options

This section describes the usage of configuration options in ESDP for UPnP services.

9.1.3.1 Maximum Transmission Unit (MTU)

This profile does not impose any additional restrictions to MTU beyond the ones stated in L2CAP specification [2] or [3]. If no MTU negotiation takes place, the default MTU value in L2CAP specification 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 honouring any already agreed upon QoS contracts with other devices and/or applications. It is expected that during the lifetime of an L2CAP connection for UPnP service transactions 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 UPnP service session is allowed to be re-negotiated during the lifetime of the UPnP service session, to accommodate the QoS constraints of the new L2CAP connection.

9.1.3.2 Flush Time-out

The UPnP service transactions are carried over an L2CAP reliable channel. The flush time-out value shall be set to its default value 0xFFFF.

9.1.3.3 Quality of Service

The use of Quality of Service (QoS) and QoS negotiation is optional. If QoS is to be negotiated, the default setting shall be used. In particular, UPnP service traffic shall be treated as best-effort service type traffic.

9.1.4 Connection Management

L2CAP provides a reliable channel using the mechanisms available at the Baseband layer. However, currently L2CAP does not exercise flow control to prevent data buffer overflow nor does it enforce a reliable channel to ensure data integrity. The purpose of the connection management layer is to provide a reliable connection service between peer-to-peer applications using a connection-oriented L2CAP channel[7]. The connection management layer achieves a reliable transmission of data between peer devices using a go back *n* automatic repeat request (ARQ) mechanism. If the errors and retransmissions are ignored, with a window size of *n*, at most *n* PDUs for a given L2CAP connection can be outstanding in the piconet at a given time. A new PDU *j* can be sent only after an acknowledgement is received for PDU *j* – *n*. The effect of this end-to-end window flow control is that as receive buffers fill up and delay increases, acknowledgements are delayed and the source device is slowed down. Also, if the destination device was busy or its buffer is

[7] Connection management layer includes flow control because currently L2CAP does not provide the required flow control functionality. However, should L2CAP layer include flow control in the future, the connection management layer would not require flow control functionality.

full, it would advertise a zero receive window size to the peer device to stop further transmission of PDUs until the destination device advertises a non-zero receive window size. Additionally, the destination device may also decrease or increase the reception of PDUs from its peer device by appropriately advertising a smaller or larger receive window size respectively. The combined effect results in mitigating buffer overflow at both the source and destination devices.

Data transmissions are classified into 2 categories, information and control. Data transmissions classified as information are the actual information-carrying PDUs and the devices use a go back *n* ARQ protocol with a modulus of 256 (2^8 for a 8-bit Sequence Number) for peer-to-peer flow control and error recovery [10]. Data transmissions classified as control are used by a device advertise its receive window size to the peer transmitting device and they use a go back *1* ARQ protocol with a modulus of 256 (2^8 for a 8-bit Sequence Number) for peer-to-peer flow control and error recovery.

9.1.4.1 Number of Connections

The number of connections for UPnP services between a pair of peer devices must be limited to 1 during any particular period of time. In addition, different UPnP enabled services between the particular pair of peer devices shall use the already established connection or establish a connection if one does not already exist, in either of the two scenarios, the number of connections for UPnP services between the pair of peer devices is limited to 1. Limiting to 1 connection simplifies connection management layer implementation by avoiding issues such as criteria for opening/closing multiple connections and load balancing across multiple open connections.

It is possible that the connection management layer in peer devices may initiate the establishment of L2CAP connections to each other simultaneously. In this scenario the request for L2CAP connection establishment from the device with higher BD_ADDR will proceed to completion and the request for L2CAP connection establishment from the device with lower BD_ADDR is rejected or terminated by both devices.

9.1.4.2 Byte and Bit Order

The byte and bit ordering when defining a connection management PDU and its fields must use the Little Endian format, with less significant (lower-order) bytes and bits being transferred before more-significant (higher-order) bytes and bits.

9.1.4.3 Protocol Data Unit Format

The connection management PDU consists of a PDU header followed by optional data payload as shown in Figure 9.1. The PDU size shall be the minimum supported MTU for connection-oriented L2CAP packets (MTU_{cno}) negotiated during L2CAP channel configuration. Note that the negotiated MTU_{cno} for an established L2CAP connection must be greater than 4 bytes [8]; else, the connection management layer terminates the L2CAP connection and indicates a lack of L2CAP communication resources to the higher layer.

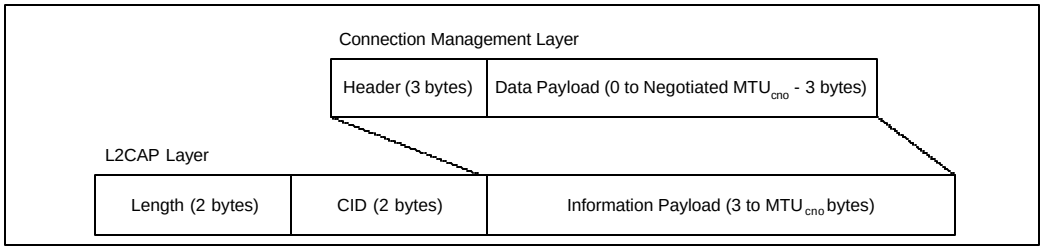

Figure 9.1: Connection Management Layer PDU Mapping to L2CAP Packet

There are two types of connection management PDU formats: Information and Control [10]. The PDU header contains a 4-bit Type field that specifies the type of connection management PDU format. The first bit of the Type field is set to 0 for Information format and is set to 1 for Control format. Information and Control formats are further classified into two categories based on whether the PDUs carry data payload and acknowledgements or only acknowledgements. The second bit of the Type field is set to 0 for PDUs that carry data payload and acknowledgements and is set to 1 for PDUs that carry only acknowledgements.

The information format is used for a Data PDU to perform information transfer between peer devices. A Data Acknowledgement PDU also based on the Information format is used to acknowledge the receipt of a Data PDU and to indicate the next sequence number of the Data PDU the sender device of the Data Acknowledgement PDU expects to receive. A Data Acknowledgement PDU does not contain the optional Data Payload field following the PDU header field.

Control format is used for Window Size Control PDU to control the flow of Data PDUs between peer devices by managing Data PDU transmit and receive window sizes. Window Size Control Acknowledgement PDU based on the Control format is used to acknowledge the receipt of Window Size Control PDU and to indicate the next sequence number of the Window Size Control PDU the sender device of the Window Size Control Acknowledgement PDU expects to receive. Window Size Control PDU contains a 1 byte Data Payload field called Window Size that indicates the Data PDU receive window size.

[8] L2CAP implementations must support a minimum MTU size of 48 bytes [2] or [3].

Window Size Control Acknowledgement PDU does not contain the optional Data Payload field following the PDU header field.

Type	PDU Format	PDU Name	Description
0x0	Information	Data	Data Payload and Acknowledgements
0x2	Information	Data Acknowledgement	Acknowledgement
0x1	Control	Window Size Control	Data Payload and Acknowledgements
0x3	Control	Window Size Control Acknowledgement	Acknowledgement

Figure 9.2 shows the layout of the Data PDU. The header includes Type (4 bits), Sequence Number (8 bits), and Request Number (8 bits) fields and the payload includes Data Payload (0 to MTU_{cno} – 3 bytes) field. Data integrity of all transmitted Data PDUs is ensured via an acknowledgement and retransmission mechanism.

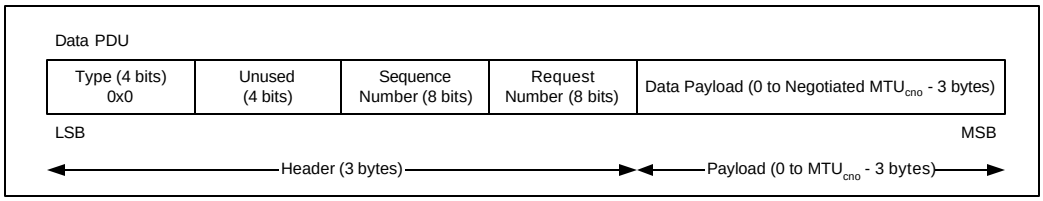

Figure 9.2: Data PDU

Sequence Number (8 bits): Successive connection management Data PDUs are numbered modulo 2^8 and this number is placed in the Sequence Number field of the outgoing Data PDU. The size of the Data Payload field may range from 0 to the negotiated MTU_{cno} - 3 bytes.

Request Number (8 bits): This field contains the value of the next Sequence Number the sender of the Data PDU is expecting to receive from its peer device. The Request Number serves an implicit acknowledgement for all Data PDUs with Sequence Numbers less than the number in the Request Number field. Piggybacking the Request Number on an outgoing Data PDU improves the throughput efficiency of the piconet.

Data Payload: This field contains data from higher layer peer-to-peer message exchanges. The data payload size can range from 0 to the negotiated MTU_{cno} - 3 bytes. Data PDU without a Data Payload implies that the particular Data PDU does not contain higher layer data. Data PDUs without the optional Data Payload may be used as keep-alive messages by peer devices to maintain the established L2CAP connectivity.

Figure 9.3 shows the layout of the Data Acknowledgement PDU. The header includes Type (4 bits) and Request Number (8 bits) fields. The Data Acknowledgement PDU is used to acknowledge the successful receipt of one or more successive Data PDUs when there does not exist an outgoing Data PDU to piggyback the Request Number to the peer device. The connection management layer does not ensure data integrity for Data Acknowledgement PDUs and therefore does not perform retransmission of lost Data Acknowledgement PDUs.

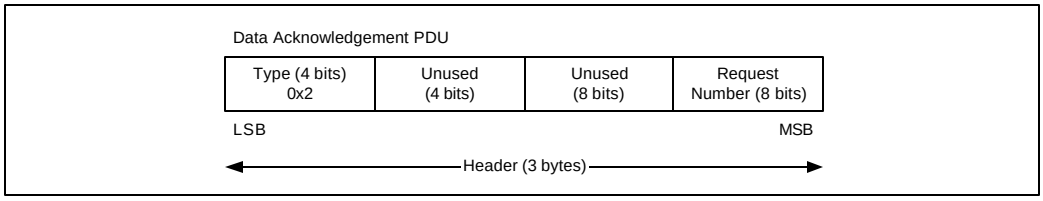

Figure 9.3: Data Acknowledgement PDU

Request Number (8 bits): This field contains the value of the next Sequence Number the sender of the Data Acknowledgement PDU is expecting to receive from its peer device. The Request Number serves an implicit acknowledgement for all Data PDUs with Sequence Numbers less than the number in the Request Number field.

Figure 9.4 shows the layout of the Window Size Control PDU. The header includes Type (4 bits), Sequence Number (8 bits) and Request Number (8 bits) fields and the payload includes Window Size (1 byte) field. Data integrity of all transmitted Window Size Control PDUs is ensured via an acknowledgement and retransmission mechanism.

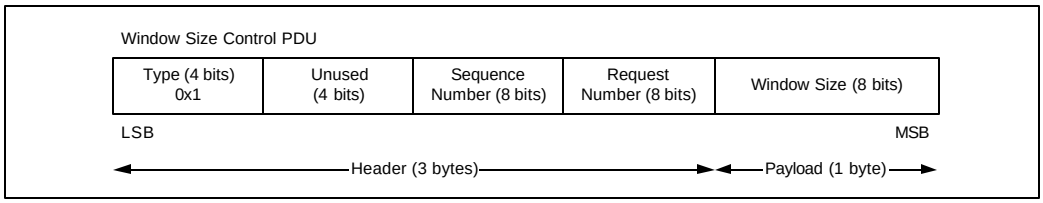

Figure 9.4: Window Size Control PDU

Sequence Number (8 bits): Successive connection management Window Size Control PDUs are numbered modulo 2^8 and this number is placed in the sequence number field of the outgoing Window Size Control PDU. The size of the Data Payload field is 1 byte.

Request Number (8 bits): This field contains the value of the next Sequence Number the sender of the Window Size Control PDU is expecting to receive from its peer device. The Request Number serves an implicit acknowledgement for all Window Size Control PDUs with Sequence Numbers less than the number in the Request Number field. Piggybacking the Request Number on an outgoing Window Size Control PDU improves the throughput efficiency of the piconet.

Window Size (8 bits): This field contains the value of the Data PDU receive window size for the sender of the Window Size Control PDU. Peer devices shall initialise the connection management layer to a default window size of 2 for both Data PDU transmit and receive window sizes following the establishment of peer-to-peer L2CAP connection. The transmitting device shall determine the transmit window size to be a function of the advertised Data PDU receive window size of the peer device and available buffer space at the transmitting device. Note that the transmit window size shall not exceed the smaller of the two values, the advertised receive window size of the peer device or the available buffer space at the transmitting device. The transmit window size shall be updated and effective immediately upon the receipt of an advertised receive window size from the peer device or a change in the available buffer space at the transmitting device. When the receive buffer for Data PDUs is full, a stop indication (Window Size = 0) is transmitted to the peer device to stop the transmission of Data PDUs. When the receive buffer is available, a go indication (Window Size not equal to 0) is returned. The device sending the Window Size Control PDU may also decrease or increase the reception of Data PDUs from its peer device by appropriately advertising a smaller or larger receive window size. In case of multiple receive window size updates that have not been transmitted, only the most current receive window size update is transmitted to the peer device while the rest of the pending updates are discarded. Transmitting the most recent receive window size update and discarding older updates ensures that current receive buffer state is reflected to the transmitting peer device while minimizing Window Size Control PDU transmissions.

Figure 9.5 shows the layout of the Window Size Control Acknowledgement PDU. The header includes Type (4 bits) and Request Number (8 bits) fields. The Window Size Control Acknowledgement PDU is used to acknowledge the successful receipt of one or more successive Window Size Control PDUs when there does not exist an outgoing Window Size Control PDU to the peer device to piggyback the Request Number. The connection management layer does not ensure data integrity for Window Size Control Acknowledgement PDUs and therefore does not perform retransmission of lost Window Size Control Acknowledgement PDUs.

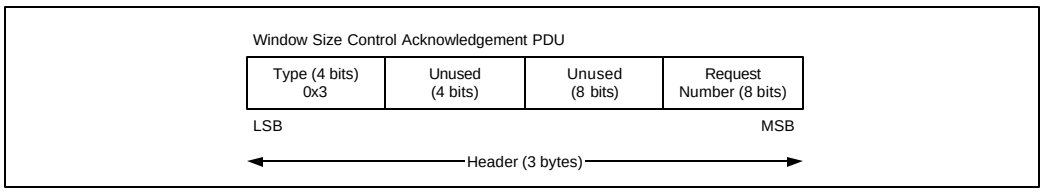

Figure 9.5: Window Size Control Acknowledgement PDU

Request Number (8 bits): This field contains the value of the next Sequence Number the sender of the Window Size Control Acknowledgement PDU is expecting to receive from its peer device. The Request Number serves an implicit acknowledgement for all Window Size Control PDUs with Sequence Numbers less than the number in the Request Number field.

9.1.4.4 Segmentation and Reassembly

Segmentation and reassembly (SAR) is used to support the peer-to-peer message exchanges over a connection management layer.

Segmentation: The connection management layer will segment the information packets into PDUs to limit the data size sent to the L2CAP layer below L2CAP layer's negotiated MTU_{cno} limit. This implies that the connection management PDU's Data Payload field has a limit of MTU_{cno} - 3 bytes, i.e., excluding the 3 bytes for connection management header.

Reassembly: The connection management layer delivers PDUs received in sequence to the higher layer of the peer device.

9.1.4.5 Flow Control and Error Recovery for Data PDUs

Data PDU transmissions must use a go back *n* ARQ protocol with a modulus of 256 (2^8 for an 8-bit Sequence Number) for peer-to-peer flow control and error recovery [10]. Successive Data PDUs are numbered modulo 256 and this Sequence Number is sent in the PDU header at the connection management layer. The destination device up on receiving the Data PDU acknowledges its receipt by returning the Request Number using either an outgoing Data PDU or a Data Acknowledgement PDU. Upon initialisation *n* is set to 2 since the default transmit and receive window size is equal to 2. The destination peer device may change the window size from its default setting by advertising its receive window size using the Window Size Control PDUs. If the errors and retransmissions are ignored, with a window size of *n*, at most *n* Data PDUs for a given L2CAP connection can be outstanding in the piconet at a given time. The effect of this end-to-end window flow control is that as receive buffers fill up and delay increases, acknowledgements are delayed and the source device is slowed down. Also, if the destination device was busy or its buffer is full, it would advertise a zero receive window size (Window Size = 0) by sending a Window Size Control PDU to the peer device to stop further transmission of Data PDUs until the destination device advertises a non-zero receive window size. Additionally, the destination device may also decrease or increase the reception of Data PDUs from its peer device by appropriately advertising a smaller or larger receive window size respectively. The combined effect results in mitigating buffer overflow at both the source and destination devices.

Upon the receipt of a Data PDU, the destination device may piggyback the next sequence number it expects to receive in the Request Number field of an outgoing Data PDU. However, when there are no pending outgoing Data PDUs, an outgoing Data Acknowledgement PDU is generated by the destination device to the source device with the sequence number it expects to receive in the Request Number field.

The transmitting device uses variables Sequence Number$_{min}$ and Sequence Number$_{max}$ to represent Data PDUs sent but not acknowledged. Sequence Number$_{min}$ denotes the smallest-numbered Data PDU that has not been acknowledged (i.e., the lower end of the transmit window *n*) and Sequence Number$_{max}$ denotes the number of the next Data PDU to be accepted from the higher layer. Data PDUs from the higher layer are accepted for transmission as long as Sequence Number$_{max}$ < Sequence Number$_{min}$ + transmit window size *n*. This ensures that at any instant there can be at the most *n* outstanding Data PDUs that have not been acknowledged and thereby ensuring flow control at the transmitting device. When a Request Number greater that Sequence Number$_{min}$ is received via a return Data PDU or Data Acknowledgement PDU, the Sequence Number$_{min}$ is set to the Request Number. This implicitly acknowledges all successive Data PDUs with Sequence Number less than the Request Number. The transmit window size *n* in the transmitting device is initialised to 2.

The receiving device uses a variable Request Number to represent the Sequence Number of the next expected Data PDU from the transmitting device. Request Number variable in the receiving device is initialised to 0. When a received Data PDU with Sequence Number equal to Request Number is transferred to the higher layer, the Request Number is incremented. The updated Request Number variable is sent to the transmitting device in the Request Number field of an outgoing Data PDU or a Data Acknowledgement PDU.

It is possible that either the transmitted Data PDU or the acknowledgement piggybacked over a return Data PDU or sent via a Data Acknowledgement PDU is lost. In order to handle such errors, the transmitting device starts a timer called Data PDU Acknowledgement Timeout eXpired (DPDUACKTX) timer whenever a Data PDU is transmitted. Note that since there can be multiple outstanding Data PDUs, the transmitting device logically needs multiple DPDUACKTX timers, one per outstanding Data PDU. Each DPDUACKTX timer times out independently of all other DPDUACKTX timers. If the DPDUACKTX timer for a particular Data PDU expires before the Data PDU is acknowledged, the go back *n* ARQ protocol initiates the retransmission of Data PDUs starting with the Sequence Number of the Data PDU whose DPDUACKTX timer expired up to the Sequence Number$_{max}$[9]. The time-out for the DPDUACKTX timer should be chosen long enough to include round-trip propagation and processing delay plus transmission time for L2CAP PDUs equal to the value of currently active transmit window size at the negotiated MTU$_{cno}$ limits in the direction of the data flow. The value of this timer is implementation-dependent but the minimum initial value is 1 second and the maximum initial value is 180 seconds. If retransmissions occur due to a failing Data Acknowledgements, the destination device receives multiple

[9] The number of retransmissions is implementation dependent. Multiple retransmissions of a Data PDU may point to an inoperative connection management layer, or even to a severed L2CAP channel or Bluetooth link. Implementors may terminate retransmissions of a Data PDU based on application and user requirements for their implementations.

Data PDUs with the same Sequence Number. By comparing the Sequence Number of consecutive Data PDUs, correctly received retransmissions can be discarded.

Note that Control format PDUs awaiting transmission should be given precedence over Information format PDUs. This ensures that when a device has queued Control format and Information format PDUs awaiting transmission, Control format PDUs are transmitted before Information format PDUs.

9.1.4.6 Flow Control and Error Recovery for Window Size Control PDUs

The flow control and error recovery mechanism for the Window Size Control PDU transmissions is based on the ARQ protocol similar to the mechanism described for Data PDUs in Section 9.1.4.5. However, there are differences between the ARQ protocol implementation described in Section 9.1.4.5 for Data PDU transmissions and ARQ protocol implementation for Window Size Control PDUs described in the following paragraphs.

Window Size Control PDU transmissions must use a go back *1* ARQ protocol with a modulus of 256 (2^8 for an 8-bit Sequence Number) for peer-to-peer flow control and error recovery [10]. Successive Window Size Control PDUs are numbered modulo 256 and this Sequence Number is sent in the PDU header at the connection management layer. The destination device up on receiving the Window Size Control PDU, acknowledges its receipt by returning the Request Number using either an outgoing Window Size Control PDU or a Window Size Control Acknowledgement PDU. Transmit and receive window sizes for peer devices must be set to 1 and cannot be changed. This ensures that at most 1 Window Size Control PDU for a given L2CAP connection can be outstanding, thereby enforcing end-to-end window flow control between peer devices.

Upon the receipt of a Window Size Control PDU the destination device may piggyback the next sequence number it expects to receive in the Request Number field of an outgoing Window Size Control PDU. However, when there is no pending outgoing Window Size Control PDU, an outgoing Window Size Control Acknowledgement PDU is generated by the destination device to the source device with the sequence number it next expects to receive in the Request Number field.

The receiving device uses a variable Request Number to represents the Sequence Number of the next expected Window Size Control PDU from the transmitting device. Request Number variable in the receiving device is initialised to zero. When a received Window Size Control PDU with Sequence Number equal to Request Number is transferred to the higher layer, the Request Number is incremented. The updated Request Number variable is sent to the transmitting device in the Request Number field of an outgoing Window Size Control PDU or a Window Size Control Acknowledgement PDU.

It is possible that either the transmitted Window Size Control PDU or the acknowledgement piggybacked over a return Window Size Control PDU or sent via a Window Size Control Acknowledgement PDU is lost. In order to handle such errors, the transmitting device starts a timer called Window Size Control PDU Acknowledgement Timeout eXpired (WSCPDUACKTX) timer whenever a Window Size Control PDU is transmitted. If the WSCPDUACKTX timer expires before the Window Size Control PDU is acknowledged, the go back *1* ARQ protocol initiates the retransmission of the outstanding Window Size Control PDU[10]. The time-out for the WSCPDUACKTX timer should be chosen long enough to include round-trip propagation and processing delay plus transmission time for one L2CAP PDU at the negotiated MTU_{cno} limits plus one Window Size Control PDU in the direction of the acknowledgement flow. The value of WSCPDUACKTX timer is implementation-dependent but the minimum initial value is 1 second and the maximum initial value is 60 seconds. If retransmissions occur due to a failing Window Size Control PDU Acknowledgements, the destination device receives multiple Window Size Control PDUs with the same Sequence Number. By comparing the Sequence Number of consecutive Window Size Control PDUs, correctly received retransmissions can be discarded.

9.1.5 Mapping of HTTP Messages to L2CAP

HTTP messages are mapped on to the L2CAP layer via the intermediate connection management and the multicast emulator layers. The connection management layer may directly receive HTTP messages with 2 types of schemes, standard HTTP and HTTPU. The connection management layer may also indirectly receive HTTP messages with HTTPMU scheme from the multicast emulator layer.

A connection management layer provides peer-to-peer connectivity over a connection-oriented L2CAP channel. The connection management layer transfers the higher layer unicast point-to-point message using either the standard HTTP or HTTPU schemes over a connection-oriented L2CAP channel between the device originating the message and the destination device with Bluetooth wireless communications. If the connection management layer has already established an L2CAP connection between the peer devices, then the higher layer message is transmitted over this connection. However, if an L2CAP connection between the connection management layers of peer devices does not exist, the connection management layer establishes a connection-oriented L2CAP channel between the peer devices. Subsequently all message exchanges between the peer higher layers are transacted over the established connection.

[10] The number of retransmissions is implementation dependent. Multiple retransmissions of a Window Size Control PDU may point to an inoperative connection management layer, or even to a severed L2CAP channel or Bluetooth link. Implementors may terminate retransmissions of a Window Size Control PDU based on application and user requirements for their implementations.

The connection management layer cannot directly map higher layer multicast point-to-multipoint message using HTTPMU scheme since the underlying connection-oriented L2CAP channel only provides point-to-point connectivity. The multicast emulator layer, shown in Figure 2.3, in between the HTTPMU layer and connection management layer maps the point-to-multipoint message into multiple unicast point-to-point messages, one per each destination device and transfers them to the connection management layer.

9.1.5.1 Addressing Format

The LocDev discovers other RemDev(s) that are within radio range using the inquiry and paging procedures. Using the inquiry and paging procedures the LocDev retrieves and caches the RemDev's BD_ADDR and class of device/service information. Using SDP, the LocDev determines the subset of RemDev(s) that provide UPnP services using L2CAP-based solution from the cached list of RemDev(s) that are within radio range. The LocDev caches the BD_ADDR and class of device/service information of the RemDev(s) that provide UPnP service using L2CAP-based solution.

Point-to-point standard HTTP and HTTPU requests have one sender device and one receiver device. The semantics of the HTTPU Universal Resource Locator (URL) are identical to HTTP. Given below is the address representation for the URL host field using the BD_ADDR of the destination device with Bluetooth wireless communications.

BD_ADDR_String.bt.local

Where BD_ADDR_String is the ASCII string of the 48-bit BD_ADDR in hexadecimal notation (leftmost symbol is most significant) and the "bt.local" suffix indicates to the HTTP message parser that the prefix is a BD_ADDR. In the L2CAP-based solution the scope of this addressing format is limited to peer devices with Bluetooth wireless communications within a single piconet. Examples of HTTP and HTTPU URLs are given below.

HTTP://123456789ABC.bt.local/

HTTPU://123456789ABC.bt.local/

Point-to-multipoint HTTPMU requests have one sender device and many receiver devices. The semantics of the HTTPMU Universal Resource Locator (URL) is identical to HTTP with the exception that the URL host field shall use the broadcast BD_ADDR for destination devices with Bluetooth wireless communications. Given below is the URL host field for the HTTPMU request.

FFFFFFFFFFFF.bt.local

Where FFFFFFFFFFFF is the ASCII string of the 48-bit broadcast BD_ADDR in hexadecimal notation (leftmost symbol is most significant) and the “bt.local” suffix indicates to the HTTP message parser that the prefix is a BD_ADDR. Similar to the addressing format for HTTP and HTTPU requests described above, HTTPMU addressing format is also limited to peer devices with Bluetooth wireless communications. An example of an HTTPMU URL is given below.

HTTPMU://FFFFFFFFFFFF.bt.local/

9.1.5.2 Multicast Emulator

The multicast emulator layer, shown in Figure 2.3, maps point-to-multipoint HTTPMU requests into multiple unicast point-to-point messages, one per each destination device and transfers them to the connection management layer since the underlying connection-oriented L2CAP channel only provides point-to-point connectivity.

Discovery of UPnP services using the L2CAP-based solution, described in Section 7.1.1, presents the application with a list of RemDev(s) that are within radio range and provide UPnP services using the L2CAP-based solution. The multicast emulator layer has access to this list of RemDev(s) and their BD_ADDRs. Upon receiving an HTTPMU request from the HTTPMU layer the multicast emulator layer performs the following steps:

1. Retrieve from the application the list of RemDev(s) and their BD_ADDRs that are within radio range and provide UPnP services using the L2CAP-based solution.

2. Select a device from the list of RemDev(s) (to which the HTTPMU message has not already been sent). If a connection to the particular device does not already exist, initiate the establishment of a connection-oriented L2CAP channel to the device via the connection management layer.

   Alternatively, if a connection to the particular devices do not already exist, initiate the establishment of one connection-oriented L2CAP channel per device in the list of RemDev(s) (to which the HTTPMU message has not already been sent) via the connection management layer (up to a maximum of 7 at a time).

3. Transmit the HTTPMU message to the RemDev(s) using the connection(s) established by the connection management layer in Step 2.

4. Indicate the completion of the HTTPMU message transfer to the connection management layer. This indication shall be used to determine

whether the connection management layer should terminate open L2CAP channel(s) between peer connection management layers.

5. HTTPMU message transfer is complete when the message has been sent to each device in the list of RemDev(s), else return to Step 2.

9.1.6 UPnP Service Transactions and L2CAP Connection Lifetime

Since HTTP transactions comprise a sequence of service messages that constitutes a connectionless datagram service, no connection is made prior to the peer-to-peer HTTP message exchange. The UPnP service application delegates the creation of connection on its behalf to the connection management layer and also has the responsibility to request the L2CAP layer to establish and terminate the connection on its behalf. Note that there is no signalling between the UPnP service application and the connection management layer. The receipt of data from the higher layer initiates establishment of connections by the connection management layer.

Since HTTP transactions are considered stateless, terminating an L2CAP connection after a UPnP service message is sent will be detrimental to UPnP service transaction. Moreover, a significant performance penalty will have to be paid if, for each UPnP service transmission, a new L2CAP connection is to be created. Therefore it is suggested that the L2CAP connection for UPnP service messages shall last more than the transmission of a single UPnP service message.

HTTP/1.1 standard allows for persistent connections for pipelining message transmissions and provides for connection keep-alive indication in messages. HTTP keep-alive indication along with performance improvement techniques by implementers such as timers to time periods of UPnP service transaction inactivity over a specific UPnP service connection established by the connection management layer may be used to decide how long to maintain an L2CAP connection.

9.2 IP-based Solution

9.2.1 IP-based Solution using PAN Profile

This profile scenario does not impose any additional requirements on L2CAP connections other than those required by PAN profile.

9.2.2 IP-based Solution using LAN Access Profile

This profile scenario does not impose any additional requirements on L2CAP connections other than those required by LAN Access profile.

10TCP/UDP/IP

10.1L2CAP-based Solution

This profile scenario does not require TCP/UDP/IP services.

10.2IP-based Solution

Transport protocol standards called Requests for Comments (RFCs) are defined by the Internet Engineering Task Force (IETF) and used for communication across the Internet [11].

10.2.1 IP-based Solution using PAN Profile

The set of IETF RFCs required for communication are classified into two categories depending on whether the network layer is based on IP Version 4 or IP Version 6.

10.2.1.1 TCP/UDP/IP Version 4

The mandatory set of IETF RFCs required for communication is listed in the following table. Inclusion of additional IETF RFCs is optional in this profile scenario.

Description	RFC Number	RFC Title
TCP: Transmission Control Protocol	0793	Transmission Control Protocol
	1323	TCP Extensions for High Performance
	2018	TCP Selective Acknowledgement Options
UDP: User Datagram Protocol	0768	User Datagram Protocol
IP: Internet Protocol	0791	Internet Protocol
	0792	Internet Control Message Protocol
	0826	An Ethernet Address Resolution Protocol
	0894	A Standard for the Transmission of IP Datagrams over Ethernet Networks
	0919	Broadcasting Internet Datagrams
	0922	Broadcasting Internet Datagrams in the Presence of Subnets
	0950	Internet Standard Subnetting Procedure
	1034	Domain Names – Concepts and Facilities
	1035	Domain Names – Implementation and Specification

Description	RFC Number	RFC Title
	1112	Host Extensions for IP Multicasting
	1256	ICMP Router Discovery Messages
	2131	Dynamic Host Configuration Protocol
	2132	DHCP Options and BOOTP Vendor Extensions

10.2.1.2 TCP/UDP/IP Version 6

The mandatory set of IETF RFCs required for communication is listed in the following table. Inclusion of additional IETF RFCs is optional in this profile scenario.

Description	RFC Number	RFC Title
TCP: Transmission Control Protocol	0793	Transmission Control Protocol
	1323	TCP Extensions for High Performance
	2018	TCP Selective Acknowledgement Options
UDP: User Datagram Protocol	0768	User Datagram Protocol
IP: Internet Protocol	1034	Domain Names – Concept and Facilities
	1035	Domain Names – Implementation and Specification
	1752	The Recommendation for the IP Next Generation Protocol
	1886	DNS Extensions to Support IP Version 6
	1981	Path MTU Discovery for IP Version 6
	2373	IP Version 6 Address Architecture
	2374	An IPv6 Aggregate Global Unicast Address Format
	2460	Internet Protocol, Version 6 (IPv6) Specification
	2461	Neighbor Discovery for IP Version 6 (IPv6)
	2462	IPv6 Stateless Address Autoconfiguration
	2463	Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification
	2464	Transmission of IPv6 Packets over Ethernet Networks
	2526	Reserved IPv6 Subnet Anycast Addresses
	2732	Format for Literal IPv6 Addresses in URL's

10.2.2 IP-based Solution using LAN Access Profile

The set of IETF RFCs required for communication are classified into two categories depending on whether the network layer is based on IP Version 4 or IP Version 6.

10.2.2.1 TCP/UDP/IP Version 4

The mandatory set of IETF RFCs required for communication is listed in the following table. Inclusion of additional IETF RFCs is optional in this profile scenario.

Description	RFC Number	RFC Title
TCP: Transmission Control Protocol	0793	Transmission Control Protocol
	1323	TCP Extensions for High Performance
	2018	TCP Selective Acknowledgement Options
UDP: User Datagram Protocol	0768	User Datagram Protocol
IP: Internet Protocol	0791	Internet Protocol
	0792	Internet Control Message Protocol
	0826	An Ethernet Address Resolution Protocol
	0894	A Standard for the Transmission of IP Datagrams over Ethernet Networks
	0919	Broadcasting Internet Datagrams
	0922	Broadcasting Internet Datagrams in the Presence of Subnets
	0950 1034	Internet Standard Subnetting Procedure Domain Names – Concepts and Facilities
	1035	Domain Names – Implementation and Specification
	1112	Host Extensions for IP Multicasting
	1256	ICMP Router Discovery Messages
	2131	Dynamic Host Configuration Protocol
	2132	DHCP Options and BOOTP Vendor Extensions

10.2.2.2 TCP/UDP/IP Version 6

The mandatory set of IETF RFCs required for communication is listed in the following table. Inclusion of additional IETF RFCs is optional in this profile scenario.

Description	RFC Number	RFC Title
TCP: Transmission Control Protocol	0793	Transmission Control Protocol
	1323	TCP Extensions for High Performance
	2018	TCP Selective Acknowledgement Options
UDP: User Datagram Protocol	0768	User Datagram Protocol
IP: Internet Protocol	1034	Domain Names – Concept and Facilities
	1035	Domain Names – Implementation and Specification
	1752	The Recommendation for the IP Next Generation Protocol
	1886	DNS Extensions to Support IP Version 6
	1981	Path MTU Discovery for IP Version 6
	2373	IP Version 6 Address Architecture
	2374 2460	An IPv6 Aggregate Global Unicast Address Format Internet Protocol, Version 6 (IPv6) Specification
	2461	Neighbor Discovery for IP Version 6 (IPv6)
	2462	IPv6 Stateless Address Autoconfiguration
	2463	Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification
	2472	IP Version 6 over PPP
	2526	Reserved IPv6 Subnet Anycast Addresses
	2732	Format for Literal IPv6 Addresses in URL's

11References

[1] Universal Plug and Play Device Architecture, Version 1.0, Jun. 2000 (http://www.upnp.org).

[2] Specification of the Bluetooth System: Core, Specification Vol. 1, Version 1.0B, Dec. 1999 (http://www.bluetooth.com).

[3] Specification of the Bluetooth System: Core, Specification Vol. 1, Version 1.1, Dec. 2000.

[4] Specification of the Bluetooth System: Profiles, Specification Vol. 2, Version 1.0B, Dec. 1999 (http://www.bluetooth.com).

[5] Specification of the Bluetooth System: Profiles, Specification Vol. 2, Version 1.1, Dec. 2000.

[6] Bluetooth Test Specification, Part E: Test Suite Structure (TSS) and Test Purpose (TP) For Service Discovery Protocol, Revision 0.4, Nov. 1999.

[7] Bluetooth Test Specification, Part K:2: Test Suite Structure (TSS) and Test Purposes (TP) For Service Discovery Application Profile, Revision 0.1, Nov. 1999.

[8] Bluetooth Personal Area Networking (PAN) Profile, Version 0.70, Nov. 2000.

[9] Bluetooth Test Specification, Part K:9: Test Suite Structure (TSS) and Test Purposes (TP) For LAN Access Profile, Revision 1.0, Dec. 1999.

[10] D. Bertsekas and R Gallager, *Data Networks*, Prentice-Hall, Inc., 1992.

[11] Internet Engineering Task Force, IETF Directory List of RFCs (http://www.ietf.org).

Appendix A - Informational

Overview of Bridge Device Operation

As noted above, bridge devices per se are outside the scope of this profile. However, because bridge devices are relevant to a discussion of the ESDP for UPnP, this informational appendix is supplied.

The Bluetooth bridge device shall interact with other devices with Bluetooth wireless communications as yet another piconet member. Since devices may provide UPnP services using an L2CAP-based solution and/or an IP-based solution, they may communicate with other devices within their piconet irrespective of whether or not a bridge device exists within the piconet. The bridge device is primarily intended to enhance the functionality of devices within a piconet by extending their reach in service discovery and device control beyond their piconet. Additionally, the bridge device may also manage UPnP services based on device/user access lists. Management of UPnP services based on device/user access lists enables the delivery of customized services to the users and devices with Bluetooth wireless communications while potentially improving the Bluetooth wireless bandwidth utilization.

Figure A.1 shows a bridge device configuration where a device with Bluetooth wireless communications may establish peer-to-peer UPnP networking connectivity with another device with Bluetooth wireless communications and/or a bridge device.

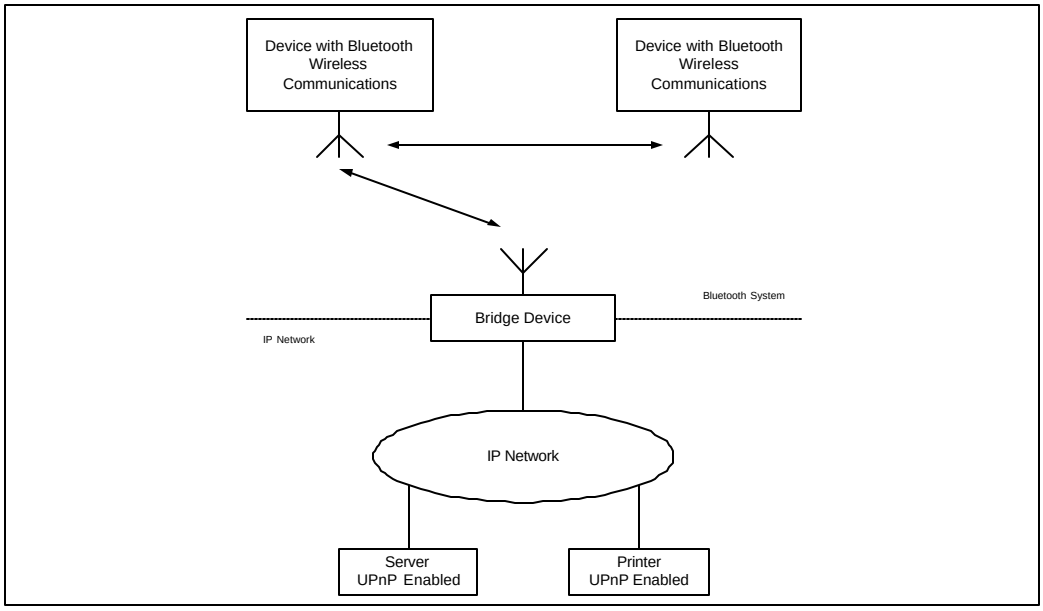

Figure A.1: Bridge Device Configuration

The bridge device provides proxy functionality of the services available to devices on the IP network and the Bluetooth system. The bridge device is expected to proxy the services available on the IP Network to the Bluetooth system. A bridge device is envisioned to have a pool of public IP addresses associated with its subnet that it would dynamically assign to a registered device with Bluetooth wireless communications using the L2CAP-based solution or the IP-based solution with a private IP address when they initiate UPnP networking functionality across the IP network shown in Figure A.1. This binding between the public IP address and the device in the Bluetooth system is local to the bridge and the devices within the Bluetooth system have no knowledge of this association. When a device in the Bluetooth system intends to avail itself of UPnP services, the bridge would proxy the transactions across the IP network using the dynamically associated public IP address with the registered device with Bluetooth wireless communications. This is analogous to the IP Network Address Translation (NAT) functionality that dynamically binds a private IP address to a valid public IP address from a pool of available addresses thus enabling a device on the private IP network to communicate with devices on the public IP network. Similarly, bridge devices enable UPnP networking connectivity between a device in the Bluetooth system and a device across the IP network by binding the device with Bluetooth wireless communications to a public IP address and proxy transactions across the IP network. Peer-to-peer interaction from a device on the IP network to a device in the Bluetooth system is expected to follow a similar approach in the reverse direction. In this scenario the bridge device would proxy the announcement of services available from registered devices in the Bluetooth system to the devices across the IP network using the messaging and control sequences defined in UPnP Device Architecture [1].