LowPowerDevice:1
Service Template Version 1.01

For UPnP™ Version 1.0
Status: Design Complete
Date: August 28, 2007
Document Version: 1.00

This Standardized DCP has been adopted as a Standardized DCP by the Steering Committee of the UPnP™ Forum, pursuant to Section 2.1(c)(ii) of the UPnP™ Forum Membership Agreement. UPnP™ Forum Members have rights and licenses defined by Section 3 of the UPnP™ Forum Membership Agreement to use and reproduce the Standardized DCP in UPnP™ Compliant Devices. All such use is subject to all of the provisions of the UPnP™ Forum Membership Agreement.

THE UPNP™ FORUM TAKES NO POSITION AS TO WHETHER ANY INTELLECTUAL PROPERTY RIGHTS EXIST IN THE STANDARDIZED DCPS. THE STANDARDIZED DCPS ARE PROVIDED "AS IS" AND "WITH ALL FAULTS". THE UPNP™ FORUM MAKES NO WARRANTIES, EXPRESS, IMPLIED, STATUTORY, OR OTHERWISE WITH RESPECT TO THE STANDARDIZED DCPS, INCLUDING BUT NOT LIMITED TO ALL IMPLIED WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT AND FITNESS FOR A PARTICULAR PURPOSE, OF REASONABLE CARE OR WORKMANLIKE EFFORT, OR RESULTS OR OF LACK OF NEGLIGENCE.

Authors	Company
Shailendra Sinha	Intel
Ujwal Paidipathi	Intel
Yin-Ling Liong	Nokia
Yinghua Ye	Nokia
Jose Costa-Requena	Nokia
Mika Saaranen (editor)	Nokia
Bruce Fairman	Sony
Sridhar Uravakonda	Motorola

Contents

List of Tables

1. Overview and Scope

This service definition is compliant with the UPnP Device Architecture version 1.0 and This service enables implementation of power saving functionality for UPnP devices.

Low power devices implement all or some of the sleep states defined in this document (e.g. Transparent Sleep, Deep Sleep Online and Deep Sleep Offline) and may be in any of those sleep states. Depending on the triggers that cause the device to transition from sleep states, the device can be modeled as a sleep/autonomous device that changes its sleep state autonomously, or a sleep/controlled device that changes its sleep state by actions from a control point. Sleeping devices (i.e., sleep capable devices in a state other than the active or disconnected) are discoverable if they respond to M-Search requests issued by Control Points. Sleeping devices that do not respond to M-Search requests are still discoverable if there is a Basic Power Management Proxy that keeps track of low power devices in the network by caching low power states announced in SSDP messages. A BPMPX is an optional component in the network that is used to cache the low power states of low power devices. A low power aware control point can find a sleeping device by querying it from a Basic Power Management Proxy.

The low power device must implement at least one of the following power states. Table 1 illustrates the relation between the internal UPnP low device power states and user visible power states [1]. Notice that when devices enter Deep Sleep Online or Offline states they will not respond to the UPnP M-Search discovery messages, but a low power aware control points can locate devices in this state by invoking UPnP control action on the optional Basic Power Management Proxy.

Table 1. Low power states.

Power State	UPnP State[2]	IP Connectivity	Bearer Status	Wake UP mechanism	Proxy	User Power State
Active	FULL	ON	802.3: LINK ON/ATTACHED[3] 802.11:ON BTH: ON/PAN ON	None	None	On
Transparent Sleep	FULL	ON	802.3: LINK ON/ATTACHED 802.11: ON/Power Save BTH: Sniff &Hold/PAN ON	Invoking the Wakeup Action on the device or autonomous wake up	Optional[4]	On/ Sleep[5]
Deep Sleep Online	PARTIAL [6]	ON	802.3: LINK ON/ATTACHED	Invoking the Wakeup Action on the device (e.g.	Optional[7]/	Sleep

[2] UPnP state consists of the UPnP stack information that is active in the device. The FULL state corresponds to the full maintenance of discovery, control and subscriptions state. The PARTIAL state corresponds to the maintenance of certain parts of the UPnP device state (e.g. the device maintains IP connectivity and, can be waken up by later defined wake up mechanisms).

[3] ATTACHED, DETACHED refers to the physical connection, both electrically and mechanically.

[4] The device has full UPnP state but it may provide lower quality of service due low power state.

[5] The UI on the control point device can represent the *Transparent* sleep as ON or SLEEP state depending on the implementation of the vendor.

[6] PARTIAL UPnP state means the discovery layer of the UPnP stack is ON. The device will only respond to the Wakeup action.

[7] The proxy is optional if the device wakes up autonomously.

[8] A proxy is required to allow a device to go into Deep Sleep Online and still be able to be part of the UPnP network.

			802.11: ON/ Power Save BTH: Sniff &Hold/PAN ON	Unicast Wakeup Action message)	Required [8]	
Deep Sleep Offline	OFF	OFF	802.3: LINK OFF/ATTACHED 802.11: OFF BTH: LINK ON/PAN OFF	Bearer specific wakeup mechanisms i.e., Wake-On-XXX mechanism (e.g. WoL[9]) Non-bearer specific wakeup mechanisms (e.g. infrared) Autonomous Wake up	Optional[10]/ Required[11]	Sleep
Disconnect	OFF	OFF	802.3: LINK OFF/DETACHED 803.11: OFF BTH: OFF	Vendor defined method (e.g. POWER ON BUTTON)	None	Off

[9] Wake On LAN (WoL) is a wake up mechanism defined for Ethernet networks

[10] A proxy is optional if the sleeping device is waking up autonomously based on e.g. timers.

[11] A proxy is required to allow a device to go into *Deep Sleep Offline* and still be able to be part of the UPnP network. The proxy handles the bearer dependent wake up mechanism.

1.1. Referenced Specifications

Unless explicitly stated otherwise herein, implementation of the mandatory provisions of any standard referenced by this specification shall be mandatory for compliance with this specification.

[1] IEEE 1621 Standard for User Interface Elements in Power Control of Electronic Devices Employed in Office/Consumer Environments” http://eetd.LBL.gov/Controls/1621.

[2] UPnP Device Architecture, version 1.1

[3] UPnP Device Architecture, version 1.0

[4] Extensible Markup Language (XML) 1.0 (Second Edition), T. Bray, J.Paoli, C. M. Sperberg-McQueen, E Maler, eds. W3C Recommendations, 6 October 2000.

[5] ISO 8601 Specification for representation of dates and times in information exchange

1.2. Notation

In this document, features are described as Required, Recommended, or Optional as follows: The key words “MUST,” “MUST NOT,” “REQUIRED,” “SHALL,” “SHALL NOT,”

“SHOULD,” “SHOULD NOT,” “RECOMMENDED,” “MAY,” and “OPTIONAL” in specification are to be interpreted as described in [RFC 2119].

In addition, the following keywords are used in this specification:

PROHIBITED – The definition or behavior is an absolute prohibition of this specification. Opposite of REQUIRED.

CONDITIONALLY REQUIRED – The definition or behavior depends on a condition. specified condition is met, then the definition or behavior is REQUIRED, otherwise it is PROHIBITED.

CONDITIONALLY OPTIONAL – The definition or behavior depends on a condition. specified condition is met, then the definition or behavior is OPTIONAL, otherwise it is PROHIBITED.

These keywords are thus capitalized when used to unambiguously specify requirements protocol and application features and behavior that affect the interoperability and security implementations. When these words are not capitalized, they are meant in their natural- sense.

- Strings that are to be taken literally are enclosed in “double quotes”.
- Words that are emphasized are printed in italic.
- Keywords that are defined by the UPnP AV Working Committee are printed using thecharacter style.
- Keywords that are defined by the UPnP Device Architecture are printed using the archstyle.

1.3. Abbreviations

Table 2: Abbreviations

Definition	Description
BPMPX	Basic Power Management Proxy
BTH	Bluetooth
LP	Low Power

1.4. Non-goals

This specification does not aim to define how power management is implemented internally, but to provide framework that allows a device to communicate its power states to other UPnP devices. Also, this specification do not specify out of band mechanisms or methods to implement out of band mechanisms.

2. Service Modeling Definitions

2.1. ServiceType

The following service type identifies a service that is compliant with this template:

urn:schemas-upnp-org:service:LowPowerDevice:1

2.2. State Variables

Reader Note: For first-time reader, it may be more insightful to read the action definition (follow link to Table 3***) before reading the state variable definitions.***

The “regular" state variables represent some actual, persistent state of the device's service.

The A_ARG_TYPE state variables are associated with temporary values only valid for the duration of the action. These state variables are used in arguments that do not directly relate to a specific state variable.

2.2.1. Derived data Types

This section defines some derived data types that are represented as UPnP string data types with special syntax.

2.2.1.1. XML documents as UPnP Arguments

The UPnP Low Power service uses XML documents as arguments in UPnP actions. The containing UPnP data type is a string. This places restrictions on a string’s content; it has to represent a well-formed XML document.

The XML schemas used in UPnP Low Power Device are defined in the respective files located on `http://www.upnp.org/schemas/`

In the XML documents, implementations may use an explicit reference to appropriate namespaces. Finally, an XML document, in adherence to the UPnP™ V1.0 architecture [3], needs to be escaped by using the normal XML rules[4] before embedding it in a SOAP request or response message. The XML escaping rules are summarized from the reference mentioned above:

- The (<) character is encoded as (&lt;)
- The (>) character is encoded as (&gt;)
- The (&) character is encoded as (&amp;)
- The (”) character is encoded as (&quot;)
- The (‘) character is encoded as (&apos;)

Table 3: State Variables

Variable Name	Req. or Opt.[1]	Data Type	Allowed Value [11]	Default Value [2]	Eng. Units
BatteryLow	O	boolean	0 for false; 1 true.	0	N/A
ExternalPowerSupplySource	O	ui1	1 for AC, 0 for internal power source	N/A	N/A
WakeupMethod	R	string (XML document)	See Section 2.2.4	N/A	N/A
PowerSupplyStatus	R	string (XML document)	See section 2.2.5	N/A	N/A
SleepPeriod	R	ui4	-1= infinite Integer value in seconds	-1	Seconds
PowerState	R	string	Four PowerStates: -Active=0 Transparent=1 -Deep Sleep Online=2 -Deep Sleep Offline=4 (range 0-4)	0	N/A

[1] R = Required, O = Optional, X = Non-standard.

[2] Values listed in this column are required. To specify standard optional values or to delegate assignment of values to the vendor, it must be referenced a specific instance of an appropriate table below.

2.2.2. BatteryLow

This variable is used to indicate when battery charge is low. The type of this variable is Boolean and vendors may select desired charge level to set this variable to true state or to false state, respectively..

This is an optional variable of Boolean type. The Default value is "0" indicating that battery charge is not low.

2.2.3. ExternalPowerSupplySource

This variable indicates whether external or internal power source is in use. This is intended for eventing changes on power supply status.

This is an optional variable of ui1 type. Values are 1 for AC power, and 0 for internal power source

2.2.4. WakeupMethod

This variable includes the wake up methods supported by the device for the specific bearer on which the action was received. This is required variable of type of XML document

This variable is described by schema identified by “urn: schema-upnp-org:lp:wakeupmethod” ; and it is located at : http://www.upnp.org/schemas/lp/WakeupMethod.xsd. The elements are:

- BearerWakeupMethod: Unique element per wake up method
    - IanaTechnologyType is an integer that indicates media interface type, such as 802.3 (value=6) or 802.11 (value=71). The allowed integer values for this parameter are specified in the IANA reference ifType-MIB <http://www.iana.org/assignments/ianaiftype-mib>.
    - WakeUpPattern has to be transmitted to wake up the device as specified by the specific Wakeup method. The type is string and it is dependent on the system used and exact format is not defined by UPnP forum.
    - AdditionalBearerInfo allows vendors to add MAC address of the device. This is a string format, but it should follow the address type defined by the bearer. For instance, 802 address types are used in 802 network. A vendor may also add additional information in string format, for instance, the name of the interface (e.g. MAC address, BTH Address).
- NonBearerWakeupMethod includes two string type elements (e.g. Infrared remote control):
    - BearerType is enumerated list that can be extended with new bearers. Currently two methods have been defined: “NFC”, and “Infrared”
    - VendorNonBearerInfo is of type string defining the vendor of this bearer

2.2.5. PowerSupplyStatus

This required variable is described by schema identified by “urn: schema-upnp-org:lp:powersupplystatus” ; and it is located at : http://www.upnp.org/schemas/lp/PowerSupplyStatus.xsd. It includes the following fields:

- ExternalPowerSupply
    - ExternalPowerSupplySource is of type string including vendor defined information for external power source, for instance, “ mains”
    - IsConnected is of type Boolean and values used are “1” indicating connected state, 0 for unconnected state
- InternalPowerSupply
    - The remaining power status in terms of percentage (value range 0-100)
    - The remaining time that battery lasts (The value should be in format of PxxDTyyHzzM, where xx is days (0-99) , yy is hours (0-24), and zz is minutes (0-60). See details from [5]

.

2.2.6. SleepPeriod

This required variable includes duration that the device is going to be in sleep state. The type of this variable is ui4.

2.2.7. PowerState

This required variable provides the power state information of the device. This variable is used in "GoToSleep" action. Low power aware Control Point can define the sleeping PowerState that the device should transition to. Since the sleeping device can have three sleeping power states, i.e., Transparent, Deep Sleep Online, Deep Sleep Offline, PowerState=1 is used to define Transparent state, PowerState=2 defines Deep Sleep Online, and PowerState=4 stands for Deep Sleep Offline. The type of this variable is string and service description document must have updated list of supported powerstates in allowedvaluelist attribute. The device can transition into only one of the power states, therefore a combination of power states is not allowed. See states description from Table 1.

2.2.8. Low power SSDP extensions

The low power device has to include a set of new SSDP headers to communicate the power state changes.

Table 4. SSDP power management extension headers.

Header	Value	Description
Powerstate	Power states in Table 1	Indicate the new power state of the device.
SleepPeriod	Number of seconds or infinite	Indicate the period that the device is expected to remain in certain power mode state (e.g. Sleep: number of seconds or infinite represented with an integer (no space) or -1 for infinite The Basic Power Management Proxy may include recommended time to sleep and the device may not set a sleep period bigger than the one recommended by the proxy.
RegistrationState	Binary state	This item is reserved for future versions.

2.3. Eventing and Moderation

Table 5: State variable eventing

Variable Name	Evented	Moderated event	Max event rate(N/min)	Logical combination	Min Delta per event
BatteryLow	Yes	Yes	1	N/A	N/A
ExternalPowerSupplySource	Yes	Yes	1	N/A	N/A
WakeupMethod	No	N/A	N/A	N/A	N/A
PowerSupplyStatus	No	N/A	N/A	N/A	N/A
SleepPeriod	No	N/A	N/A	N/A	N/A
PowerState	No	N/A	N/A	N/A	N/A

UPnP low power device provides two evented variables (Table 5) that can be used to advertise current power source status. BatteryLow is intended for battery powered device to send warning messages to control points stating that the device is operating on a low battery. The ExternalPowerSupplySource is intended to send events when the device's power is switched between main power (AC) and battery power (DC). Vendors may tune event rates from maximum to 1 event per minute if so desired.

2.4. Generic LP-aware control point recommendations

Control points may cache power management information of all low power devices in the network. Especially, if the LP-aware control point vendor does not want to rely on the presence of Basic power management proxies.

If Basic Power Management Proxy is present, LP-aware control point should fetch the list of sleeping devices from the proxy during the discovery process.

If a low power aware control point can perform out-of-band wake-up provided by the device, the control point should wake the low power device directly.

If no caching proxies are available in the network, LP-aware control point should stay active to maintain list of sleeping devices.

LP-aware Control point must support low power SSDP extension headers. .

If a low power device does not respond to M-Search or UPnP action, the control point should assume that device is in disconnected state. However, the device may be in deep sleep offline state and the LP-aware control point may try to wake up the device several times. After several unsuccessful wake up attempts, the LP-aware control point should assume that device is not available and the UPnP Low Power device should be considered to be in the disconnected state.

If a low power device is in deep sleep online or deep sleep offline state, the control point should not remove it from cache before the low power device's sleep period has expired. If wakeup command fails, then control point should consider device disconnected (some additional wait time should be added). This is necessary as a low power device in deep sleep online or deep sleep offline does not send SSDP keep alive messages.

2.5. Actions

Table 6 presents actions that are later presented in detail followed by detailed information about actions, including short descriptions of the actions, the effects of the actions on state variables, and error codes defined by the actions.

Table 6: Actions

Name	Req. or Opt. [1]
GetPowerManagementInfo()	R
Wakeup()	R
GoToSleep()	R

[1] R = Required, O = Optional, X = Non-standard.

2.5.1. GetPowerManagementInfo()

The device will provide power management related information of the interface on which the action was received (e.g. bearer specific wake up mechanism, network interface information, vendor defined information, non-bearer specific wakeup mechanism, etc).

2.5.1.1. Arguments

Table 7: Arguments for GetPowerManagementInfo()

Argument	Direction	relatedStateVariable
WakeupMethod	Out	WakeupMethod
PowerSupplyStatus	Out	PowerSupplyStatus

2.5.1.2. Dependency on State (if any)

None, these are static capabilities.

2.5.1.3. Effect on State (if any)

None.

2.5.1.4. Control point requirements

This action is used to retrieve information from the device. The control point may cache Wake-up information for later use. This information can be used to wake up device in deep sleep states in the case when Basic Power Management Proxy is not present. Naturally, devices in active or transparent states with standard keep alive messaging may be kept in cache.

2.5.1.5. Errors

Refer to UPnP Device architecture for common error codes.

Table 8: Error Codes for GetPowerManagementInfo()

errorCode	ErrorDescription	Description
701	Cannot return power management info	Device error

2.5.2. Wakeup()

The device will provide a control action to wake up the device when it is transparent or deep sleep online state.

2.5.2.1. Dependency on State (if any)

This action can be used only in online modes

2.5.2.2. Effect on State (if any)

The power state changes after the device responds to the wake up action. When wake up action is received, the device transitions to the Active state. The sleep period should be set to 0.

2.5.2.3. Control point requirements

Control point must verify new power state of a device after wake up action. The Device has the sole control of internal power states and may not change its power state. Verification may happen either from device advertisement or directly through GetPowerManagementInfo() action.

Control point should not engage new power management action until the new state is advertised.

When a device transitions to active state, the sleep period should not be used.

2.5.2.4. Errors

Device may refuse to change into active mode. In this case it returns error 707

Table 9: Error Codes for WakeUp()

errorCode	ErrorDescription	Description
707	Action refused	Device refused to change power state

2.5.3. GoToSleep()

The device will provide a control action to go into sleep mode when the action is received. This action provides a recommendation to the device to go into sleep mode (i.e. Transparent, deep sleep online or offline)

2.5.3.1. Arguments

Table 10: Arguments for GoToSleep()

Argument	Direction	RelatedStateVariable
RecommendedSleepPeriod	In	SleepPeriod
RecommendedPowerState	In	PowerState
SleepPeriod	Out	SleepPeriod
PowerState	Out	PowerState

2.5.3.2. Dependency on State (if any)

2.5.3.3. Effect on State (if any)

The power state changes when the device responds to the action with the recommended power state change.

2.5.3.4. Control point requirements

Figure 1 presents allowed state transitions for a low power device. A low power device retains in allowed state transitions and may decline to transition to some state either by violating transitions defined in figure one or due internal reasons

Control point should gracefully handle situation when certain power state or transition is not implemented.

2.5.3.5. Errors

Refer to UPnP Device architecture for common error codes.

Table 11: Error Codes for GoToSleep()

errorCode	ErrorDescription	Description
705	Device rejects to change power state	The device refuses to change the current power state
706	Powerstate not implemented	The recommended PowerState is not implemented by device
402	Invalid argument	The input arguments are not valid

2.6. Theory of Operation

The Low Power Device Control Protocol provides actions for control points to query and manage low-power devices. The device must implement the required SSDP low power extension headers and should conform to the low power state machine, described in Figure 3, when transitioning between power states.

2.6.1. Explanation of Actions and operations

A UPnP Low Power device will expose its static Power Management related information through the GetPowerManagementInfo() action. This information includes bearer specific wakeup method such as Bearer type, wakeup pattern, AdditionalBearerInfo (e.g. MAC address, BTH Address, etc), NonBearerWakeupMethod (e.g. Infrared, NFC), supported power states and PowerSupplyStatus.

The Low Power device will transition out of the sleeping state (i.e., transparent state or Deep Sleep Online state) to active state upon receiving a wakeup() action. The GoToSleep() action provides a mechanism for low power aware Control Point to request the device to enter one of the sleeping states for a certain period. The input sleeping state and sleeping period are suggested by the low power aware control point. The device may or may not implement the recommended values. It will return the power state that it is going to transition to and the sleeping period that it chooses.

Whenever the Low Power device changes its power state, it must announce this power state transition via the SSDP extension header inside SSDP messages.

- When the Low Power device enters active state, it sends SSDP alive messages with PowerState set to active. The low power device includes the PowerState optional header set to active in the keepalive messages.
- When the Low Power device transitions to Transparent state, it sends SSDP alive messages with PowerState set to transparent, and when the Low Power device is in Transparent state, it also sends periodic SSDP announcements with PowerState set to transparent.
- When the Low Power device transitions to Deep Sleep Online or Deep Sleep Offline state, it sends SSDP byebye messages with the corresponding power state of device (i.e., PowerState set to deep sleep online or deep sleep offline).
- When a device transitions from Disconnected, Deep Sleep online, or Deep Sleep offline to more active state, it must first transition to the active state. It is allowed to transition step-by-step to less active state(e.g. active to transparent sleep, transparent sleep to deep-sleep online,

deep sleep online to deep sleep offline, or deep sleep offline to disconnected). The device may also bypass states when going to less active state.

- When a device transitions to disconnected state from any state it should send a byebye message without Low power extension headers.

On receiving M-Search, the Low Power device in either Active state or Transparent state responds with the corresponding PowerState of the device. If a Low Power device is in Deep Sleep Online state, the Low Power device responds only to the wakeup() action.

When a device is in active or transparent states, it sends normal SSDP alive messages and responds to all searches as defined in the device architecture. If device is in deep sleep online state, it will not send keepalive messages before it transitions to active state. In deep sleep offline state the device cannot communicate and can be woken up only by out-of-band methods. The device should wake up before the sleep period it advertised in the low power SSDP extension headers expires and renew its UPnP state with a proxy or control points.

2.6.2. Low power state behavior

A low power device exhibits state behavior to UPnP network by advertising its low power states. Device vendors may implement internal behavior in various ways and decide how the low power states are actually implemented. Regardless of implementation specific variations, the implementations must follow responsiveness requirements given in this specification. The low power device state transitions is presented in Figure 1. Notice that in Figure 1 some of the transitions are caused by an internal action of the device and these transitions cannot be invoked by the control points or other external entities. The following state specific behavior is required:

- Active: In the *Active* state the UPnP Low Power device is on the network and is visible to other UPnP devices in the network. The device in *Active* state responds to M-search requests. The device in *Active* state sends out regular SSDP: Alive messages.

- Transparent: In the *Transparent* state the UPnP Low Power device is in online and visible to other UPnP Low Power devices. The device in *Transparent* state responds to M-search requests and can be woken up to the *Active* state by invoking the Wakeup() control action. The device in *Transparent* state sends out regular SSDP: Alive messages.

- Deep Sleep Online: A device in *Deep Sleep Online* is on the network but will not respond to any UPnP control actions except for the Wakeup() control action because the UPnP stack in *Deep Sleep Online* is partially online. A device in *Deep Sleep Online* will not respond to an M-search request and will not send any SSDP: Alive messages. A device in this state can be woken up by invoking the Wakeup() control action.

- Deep Sleep Offline: A device in *Deep Sleep Offline* will not be on the network and thus will not be visible to other devices on the network. A device in *Deep Sleep Offline* will not respond to an M-search request and will not send any SSDP: Alive messages. A device in this state can be woken up by out of band bearer dependent wake up mechanisms.

- Disconnect: A device in *Disconnect* state is shutdown or is in the OFF state. A device in this state can be woken using vendor defined methods.

Figure below shows a transition diagram for the various sleep states and the triggers that cause state transitions.

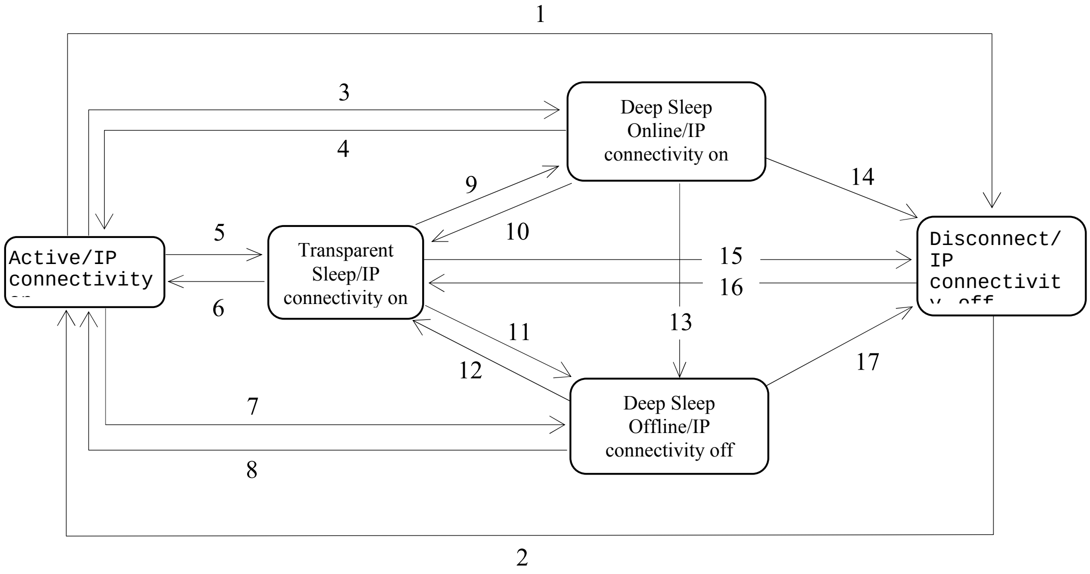

Figure 1: UPnP Low Power states

The transitions are explained in the following table. In the last column, *internal triggers* are the triggers generated by a sleeping device, such as a device's decision to switch to power save mode, to leave the network or to wake up after a timeout value. Conversely, *external triggers* are the triggers generated by Low Power Aware Control Points or Basic Power Management Proxies, such as a request for the device to enter a sleep mode or to wake up. In addition, the case of network failures is not included as a trigger because it is a generic problem for all the power states. In *Active* or *Transparent* sleep state, an unexpected disconnection can be detected by the absence of periodic advertisements. In deep sleep state, Control Points can infer that an unexpected disconnection occurred from an overdue sleep period timeout or when a control point fails to wake the Low Power device.

Table 12: State machine Transition Description

Transition Number	Transition Description	Examples of Triggers
1	Transition from *Active* to *Disconnect*: *byebye* messages are sent.	Internal
2	Transition from *Disconnect* to *Active*: *Alive* messages are sent.	Internal
.3	Transition from *Active* to *Deep Sleep Online*:*byebye* message and with PowerState equal to *Deep Sleep Online* (2) is sent.	Internal or external sleep request
4	Transition from *Deep Sleep Online* to *Active*: *Alive* messages are sent.	Internal or external Unicast control wake-up message
5	Transition from *Active* to *Transparent* Sleep: *Alive* messages with PowerState equal to *Transparent* (1) sleep state are sent.	Internal or external sleep request
6	Transition from *Transparent* Sleep to *Active*: *Alive* messages are sent.	Internal, external Unicast control messages
7	Transition from *Active* to *Deep Sleep Offline*: *byebye* message and with PowerState equal to Deep Sleep Offline (4) is sent.	Internal or external sleep request
8	Transition from *Deep Sleep Offline* to *Active*: *Alive* messages are sent.	Internal or external, bearer specific wake up mechanism (e.g. WoL)
9	Transition from *Transparent* Sleep to *Deep Sleep Online*: *byebye* message and with PowerState equal to *Deep Sleep Online* (2) is sent.	Internal or external sleep request
10	Transition from *Deep Sleep Online* to *Transparent* Sleep: *Alive* messages with PowerState equal to *Transparent* (1) sleep state are sent.	Internal
11	Transition from *Transparent* Sleep to *Deep Sleep Offline*: *byebye* message and with PowerState equal to Deep Sleep/ Offline (4) is sent.	Internal or external sleep request
12	Transition from *Deep Sleep Offline* to *Transparent* Sleep: *Alive* messages with PowerState equal to *Transparent* (1) sleep state are sent.	Internal or external bearer specific wake-up mechanism
13	Transition from *Deep Sleep Online* to *Deep Sleep Offline*: *byebye* message and with PowerState equal to Deep Sleep Offline (4) is sent.	Internal
14	Transition from *Deep Sleep Online* to *Disconnect*: *byebye* messages are sent.	Internal

Transition Number	Transition Description	Examples of Triggers
15	Transition from *Transparent* Sleep to *Disconnect*: *byebye* messages are sent.	Internal
16	Transition from *Disconnect* to *Transparent* Sleep: *Alive* messages with PowerState equal to *Transparent* (1) sleep state are sent.	Internal
17	Transition from *Deep Sleep Offline* to *Disconnect*: No messages can be sent by the sleeping device due to lack of IP connectivity.	Internal

3. XML Service Descriptions

The device vendors must updated state variable “powerstate” to include only those AllowedValue items that have implemented powerstate. Support for active state is mandatory

```
<?xml version="1.0" encoding="utf-8"?>
<scpd xmlns="urn:schemas-upnp-org:service-1-0">
   <specVersion>
      <major>1</major>
      <minor>0</minor>
   </specVersion>
  <actionList>
      <action>
         <name>GetPowerManagementInfo</name>
         <argumentList>
            <argument>
               <name>WakeupMethod</name>
               <direction>out</direction>
<relatedStateVariable>WakeupMethod</relatedStateVariable>
            </argument>
            <argument>
               <name>PowerSupplyStatus</name>
               <direction>out</direction>
     <relatedStateVariable>PowerSupplyStatus</relatedStateVariable>
            </argument>
         </argumentList>
      </action>
      <action>
         <name>Wakeup</name>
      </action>
      <action>
         <name>GoToSleep</name>
         <argumentList>
            <argument>
               <name>RecommendedSleepPeriod</name>
               <direction>in</direction>

<relatedStateVariable>SleepPeriod</relatedStateVariable>
            </argument>
            <argument>
               <name>RecommendedPowerState</name>
               <direction>in</direction>

<relatedStateVariable>PowerState</relatedStateVariable>
            </argument>
            <argument>
               <name>SleepPeriod</name>
               <direction>out</direction>

<relatedStateVariable>SleepPeriod</relatedStateVariable>
            </argument>
            <argument>
               <name>PowerState</name>
               <direction>out</direction>

<relatedStateVariable>PowerState</relatedStateVariable>
            </argument>
         </argumentList>
      </action>
   </actionList>
   <serviceStateTable>
```

```
      <stateVariable sendEvents="no">
         <name>WakeupMethod</name>
         <dataType>string</dataType>
      </stateVariable>
      <stateVariable sendEvents="no">
         <name>PowerSupplyStatus</name>
         <dataType>string</dataType>
      </stateVariable>
      <stateVariable sendEvents="no">
         <name>PowerState</name>
         <dataType>string</dataType>
         <AllowedValueList>
            <AllowedValue>0</AllowedValue>
            <AllowedValue>1</AllowedValue>
            <AllowedValue>2</AllowedValue>
            <AllowedValue>4</AllowedValue> </AllowedValueList>
</stateVariable>
      <stateVariable sendEvents="no">
         <name>SleepPeriod</name>
         <dataType>i4</dataType>
      </stateVariable>
      <stateVariable sendEvents="Yes">
            <name>BatteryLow</name>
            <dataType>Boolean</dataType></stateVariable>
      <stateVariable sendEvents="Yes">
            <name>ExternalPowerSupplySource</name>
            <dataType>i4</dataType>
      </stateVariable>
   </serviceStateTable>
</scpd>
```

4. Test

No semantic tests have been specified for this service.

5. Sample Argument XML documents

5.1. PowerSupplyStatus

```
<?xml version="1.0" encoding="UTF-8"?>
<p:PowerSupplyStatus xmlns:p="urn:schemas-upnp-
org:lp:PowerSupplyStatus"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" >
  <p:ExternalPowerSupply>
    <p:ExternalPowerSourceInfo>AC</p:ExternalPowerSourceInfo>
    <p:IsConnected>true</p:IsConnected>
  </p:ExternalPowerSupply>
  <p:InternalPowerSupply>
    <p:PowerRemaining>95</p:PowerRemaining>
    <p:TimeRemaining>P00DT10H30M</p:TimeRemaining>
  </p:InternalPowerSupply>
</p:PowerSupplyStatus>
```

5.2. WakeUpMethod

```
<?xml version="1.0" encoding="UTF-8"?>
<p:WakeupMethod xmlns:p="urn:schemas-upnp-org:lp:WakeupMethod"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <p:BearerWakeupMethod>
    <p:IanaTechnologyType>71</p:IanaTechnologyType>

<p:WakeupPattern>0008744C7F1D0008744C7F1D0008744C7F1D0008744C7F1D0008
744C7F1D0008744C7F1D0008744C7F1D0008744C7F1D0008744C7F1D0008744C7F1D0
008744C7F1D0008744C7F1D0008744C7F1D0008744C7F1D0008744C7F1D0008744C7F
1D</p:WakeupPattern>
    <p:AdditionalBearerInfo>
      <p:Address>0008744C7F1D</p:Address>
      <p:Bearer_Info>Adsilon</p:Bearer_Info>
    </p:AdditionalBearerInfo>
  </p:BearerWakeupMethod>
  <p:NonBearerWakeupMethod>
    <p:BearerType>Infrared</p:BearerType>
    <p:VendorNonBearerInfo>Kolhips</p:VendorNonBearerInfo>
  </p:NonBearerWakeupMethod>
</p:WakeupMethod>
```