RADASync:2 Service

For UPnP Version 1.0
Status: Standardized DCP
Date: April 30, 2010
Document Version: 1.0
Service Template Version: 2.00

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 PROPOSED SERVICES, IMPLEMENTATIONS OR IN ANY ASSOCIATED TEST SUITES. THE PROPOSED SERVICES, STANDARDIZED SERVICES, IMPLEMENTATIONS AND ANY ASSOCIATED TEST SUITES ARE PROVIDED "AS IS" AND "WITH ALL FAULTS". THE UPNP FORUM MAKES NO WARRANTIES, EXPRESS, IMPLIED, STATUTORY, OR OTHERWISE WITH RESPECT TO THE PROPOSED SERVICES, STANDARDIZED SERVICES, IMPLEMENTATIONS AND ASSOCIATED TEST SUITES 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
Bich Nguyen (Co-chair)	Cisco
Sridhar Ramaswamy	Cisco
Ayodele Damola	Ericsson
Bryan Roe	Intel
Vlad Stirbu	Nokia
Cathy Chan	Nokia
Jan Brands	NXP
Jeffrey Kang	Philips
Wouter van der Beek	Philips
Shrikant Kanaparti	Samsung
Se-Hee Han	Samsung
Mahfuzur Rahman (Co-chair)	Samsung
Sanjeev Verma	Samsung

UPnP Forum in no way guarantees the accuracy or completeness of this author list and in no way implies any rights for or support from those members listed. This list is not the specifications' contributor list that is kept on the UPnP Forum's website.

Contents

List of Tables

List of Figures

1 Overview and Scope

This service definition is compliant with the UPnP Device Architecture version 1.0. It defines a service type referred to herein as *RADASync* service.

This service type enables an in-band synchronization mechanism between Remote Access Clients and Remote Access Server.

1.1 Introduction

A Remote Access Discovery Agent (RADA) aggregates information about UPnP devices and services from two primary sources, depending if the devices are located in the local network or they are located in a

- Aggregation of the the local view of the UPnP network.
- Relaying discovery messages in the local network on behalf of remote devices
- Configuration of which devices are visible to and from a remote network.
- Transport protocol used to facilitate Remote Access or its configurtion.

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 this 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. If the 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. If the 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 over protocol and application features and behavior that affect the interoperability and security of implementations. When these words are not capitalized, they are meant in their natural-language sense.

- Strings that are to be taken literally are enclosed in "double quotes".
- Placeholder values that need to be replaced are enclosed in the curly brackets "{" and "}".
- Words that are emphasized are printed in *italic*.
- Keywords that are defined by the UPnP Working Committee are printed using the *forum* character style.
- Keywords that are defined by the UPnP Device Architecture are printed using the **arch** character style.
- A double colon delimiter, "::", signifies a hierarchical parent-child (parent::child) relationship between the two objects separated by the double colon. This delimiter is used in multiple contexts, for example: Service::Action(), Action()::Argument, parentProperty::childProperty.

1.3 Vendor-defined Extensions

Whenever vendors create additional vendor-defined state variables, actions or properties, their assigned names and XML representation MUST follow the naming conventions and XML rules as specified in [DEVICE], Section 2.5, "Description: Non-standard vendor extensions".

1.4 References

1.4.1 Normative References

This section lists the normative references used in this specification and includes the tag inside square brackets that is used for each such reference:

[DEVICE] – UPnP Device Architecture, version 1.0.
Available at: http://www.upnp.org/specs/arch/UPnP-arch-DeviceArchitecture-v1.0-20080424.pdf.
Latest version available at: http://www.upnp.org/specs/arch/UPnP-DeviceArchitecture-v1.0.pdf.

[DADS-XSD] – XML Schema for UPnP RA Discovery Agent XML Data Structures
Available at: http://www.upnp.org/schemas/ra/dads-v1-20090930.xsd.
Latest version available at: http://www.upnp.org/schemas/ra/dads-v1.xsd.

[RADiscoveryAgent] – RADiscoveryAgent:1, UPnP Forum,
Available at: http://www.upnp.org/specs/ra/UPnP-ra-RADiscoveryAgent-v1-Device-20090930.pdf.
Latest version available at: http://www.upnp.org/specs/ra/UPnP-ra-RADiscoveryAgent-v1-Device.pdf.

[RFC 2119] – IETF RFC 2119, Key words for use in RFCs to Indicate Requirement Levels, S. Bradner, March 1997.
Available at: http://www.ietf.org/rfcs/rfc2119.txt.

[RFC 3986] – IETF RFC 3986, Uniform Resource Identifier (URI): Generic Syntax, Tim Berners-Lee, et. Al, January 2005.
Available at: http://www.ietf.org/rfc/rfc3986.txt

[XML] – "Extensible Markup Language (XML) 1.0 (Third Edition)", François Yergeau, Tim Bray, Jean Paoli, C. M. Sperberg-McQueen, Eve Maler, eds., W3C Recommendation, February 4, 2004.
Available at: http://www.w3.org/TR/2004/REC-xml-20040204/.

[CIDR] – Classless Inter-domain Routing (CIDR): The Internet Address Assignment and Aggregation Plan, V. Fuller, T. Li, August 2006.
Available at: http://www.ietf.org/rfc/rfc4632.txt.

1.4.2 Informative References

This section lists the informative references that are provided as information in helping understand this specification:

[RAARCH] – RAArchitecture:1, UPnP Forum,
Available at: http://www.upnp.org/specs/ra/UPnP-ra-RAArchitecture-v1-20090930.pdf.
Latest version available at: http://www.upnp.org/specs/ra/UPnP-ra-RAArchitecture-v1.pdf.

[RADAConfig] – RADAConfig:1, UPnP Forum,
Available at: http://www.upnp.org/specs/ra/UPnP-ra-RADAConfig-v1-Service-20090930.pdf.
Latest version available at: http://www.upnp.org/specs/ra/UPnP-ra-RADAConfig-v1-Service.pdf.

[RATAConfig] – RATAConfig:1, UPnP Forum
Available at: http://www.upnp.org/specs/ra/UPnP-ra-RATAConfig-v1-Service-20090930.pdf.
Latest version available at: http://www.upnp.org/specs/ra/UPnP-ra-RATAConfig-v1-Service.pdf.

2 Service Modeling Definitions

2.1 Service Type

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

urn:schemas-upnp-org:service:*RADASync:2*

RADASync service is used herein to refer to this service type.

2.2 Terms and Abbreviations

2.2.1 Abbreviations

Table 2-1: Abbreviations

Definition	Description
DADS	Discovery Agent Data Structures
RA	Remote Access
RAC	Remote Access Client
RADA	Remote Access Discovery Agent
RAS	Remote Access Server
RATA	Remote Access Transport Agent
ALG	Application Layer Gateway

2.2.2 Terms

2.2.2.1 Local device

A local device is a UPnP device that is attached to the physical network where the RADA is located.

2.2.2.2 RADAListener

The RADAListener is a logical support function of RADA and incorporates control point and device functionality for facilitating the SSDP offloading:

1.) The RADAListener establishes the initial state of its local network by performing an M-SEARCH to detect all devices on the local network and notifies the RADA of those devices.

2.) The RADAListener monitors the local SSDP traffic and notifies the RADA when devices are joining and leaving the UPnP network as described in the UPnP Device Architecture.

2.2.2.3 RADARelay

The RADARelay is a logical support function of RADA and incorporates control point and device functionality for facilitating the SSDP offloading:

1.) For each device in the remote synchronization tree of the RADA, the RADARelay will send periodic SSDP announcements (i.e. ssdp:alive) onto the local network according the UPnP Device Architecture.

2.) Whenever a device is removed from the remote synchronization tree, the RADARelay will send an SSDP expiration (i.e. ssdp:byebye) onto the local network according the UPnP Device Architecture.

3.) Whenever a RADARelay receives an SSDP Search request (i.e. M-SEARCH) for a device or service that is contained in the remote synchronization tree, it will answer the search request on behalf of the device in the remote synchronization tree according the UPnP Device Architecture.

4.) When the remote connection is broken, the RADARelay will send an SSDP expiration (i.e. ssdp:byebye) on the local network for each remote device.

2.2.2.4 Remote Access Client

The Remote Access Client (RAC) is the peer physical device that is not part of the physical home network. The RAC is exposing only the UPnP devices and services that are embedded in the physical device.

2.2.2.5 Remote Access Server

The Remote Access Server (RAS) is the peer physical device located in the home network. RAS is exposing to the RAC the UPnP devices and services available in the physical home network as well as any embedded in the physical RAS device.

2.2.2.6 Remote device

A remote device is a UPnP device that is not attached to the physical network where the RADA is located.

2.3 *RADASync* Service Architecture

This service enables synchronization between two Remote Access endpoints. Each RADA endpoint exposes the device aggregation tree from the local network, that is to be remotely accessable.

Each RADA client builds an aggregation tree for the remote network, such that it can facilitate SSDP traffic on the local network on behalf of these remote devices, thereby alleviating the remote transport protocol from network timing dependencies inherent to UPnP Discovery.

2.4 State Variables

Reader Note: For a first-time reader, it may be more helpful to read the action definitions before reading the state variable definitions.

2.4.1 State Variable Overview

Table 2-2: State Variables

Variable Name	R/O[1]	Data Type	Allowed Values	Eng. Units
SystemInfo	*R*	**string**	See Section 2.4.2	
A_ARG_TYPE_DDDLocation	*R*	**string**	See Section 2.4.3	
A_ARG_TYPE_DeviceInfoList	*R*	**string**	See Section 2.4.4	
A_ARG_TYPE_UUID	*R*	**string**	See Section 2.4.5	
A_ARG_NetworkAddressInfo	*O*	**string**	See Section 2.4.6	

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

2.4.2 *SystemInfo*

This state variable contains the snapshot of all devices that are reachable/visible from the RADA, which are grouped according to the network in which they are present. The state variable contains also the identity of the remote networks and filters that determine how the synchronization process will be performed.

The structure of the *SystemInfo* argument is a DADS XML Document:

- <systemInfo> is the root element.
- See the DADS schema [DADS-XSD] for more details on the structure. The available properties and their names are described in Appendix A.1.

Note that since the value of SystemInfo is XML, it needs to be escaped (using the normal XML rules: [XML] Section 2.4 Character Data and Markup) before embedding in a SOAP response message.

Warning: This state variable is the exact same state variable that is defined in the RADAConfig service, which updates the filtering information. Furthermore, this is also the same state variable as defined in the RATAConfig service, which maintains information about Remote Access Transport profiles. This state variable MUST be updated by the device and propagated internally to those other services (if present on the same device). Each modification in *SystemInfo* MUST be signalled by the device through the *SystemInfoUpdateID* evented state variable of the RADAConfig service (see Section 2.4.3 of [RADAConfig]).

2.4.3 *A_ARG_TYPE_DDDLocation*

This state variable is defined to provide type information for action arguments used to contain the URL to the UPnP description of the root device that embeds the remote RADASync service and will be used by the co-located RADASync Control Point. Its format is a single absolute URL (see RFC 3986 [RFC 3986]).

2.4.4 *A_ARG_TYPE_DeviceInfoList*

This state variable is defined to provide type information for action arguments used to contain a list of DeviceInfo objects. Each DeviceInfo object represents a remote UPnP device.

The structure of the *A_ARG_TYPE_DeviceInfoList* argument is a DADS XML Document:

- <deviceInfoList> is the root element.
- See the DADS schema for more details on the structure. The available properties and their names are described in Appendix A.2.

Note that since the value of A_ARG_TYPE_DeviceInfoList is XML, it needs to be escaped (using the normal XML rules: [XML] Section 2.4 Character Data and Markup) before embedding in a SOAP response message.

2.4.5 *A_ARG_TYPE_UUID*

This state variable is introduced to provide type information for action arguments used for identifying the *RADiscoveryAgent* Devices that host the embedded RADASync Control Point. The type of the *A_ARG_TYPE_UUID* argument is string.

2.4.6 *A_ARG_TYPE_NetworkAddressInfo*

This state variable is defined to provide type information for the *LocalNetworkAddress* and *RemoteNetworkAddress* argument in the *GetLocalNetworkAddressInfo()* action. It takes the form of an address/prefix length [CIDR]. For example, 192.168.1.1/24 is interpreted as the subnet is the first leftmost 24 bits of the address of 192.168.1.1, i.e., 192.168.1.

2.5 Eventing and Moderation

Table 2-3: Eventing and Moderation

Variable Name	Evented	Moderated Event	Max Event Rate[1]	Logical Combination	Min Delta per Event[2]
SystemInfo	*NO*	*NO*			
A_ARG_TYPE_DDDLocation	*NO*	*NO*			

Variable Name	Evented	Moderated Event	Max Event Rate[1]	Logical Combination	Min Delta per Event[2]
A_ARG_TYPE_DeviceInfoList	*NO*	*NO*			
A_ARG_TYPE_UUID	*NO*	*NO*			
A_ARG_TYPE_NetworkAddressInfo	*NO*	*NO*			

[1] Determined by N, where Rate = (Event)/(N secs).

[2] (N) * (allowedValueRange Step).

2.6 Actions

Table 2-4: Actions

Name	R/O1
AddRemoteDevices()	*R*
RemoveRemoteDevices()	*R*
SetDDDLocation()	*R*
HeartbeatUpdate()	*R*
GetLocalNetworkAddressInfo()	*O*

[1] *R* = REQUIRED, *O* = OPTIONAL, *X* = Non-standard.

2.6.1 *AddRemoteDevices()*

This action specifies that a remote device or set of remote devices is to be added to the collection of locally accessible devices. This action specifies the devices that require SSDP offloading.

2.6.1.3 Effect on State

When this action is invoked, the *SystemInfo* state variable must be updated so that the devices indicated in the *DeviceList* argument are added to the *RemoteNetwork* element (identified by the ID argument) corresponding to the RADASync Control Point that invoked the action.

Additionally, the value of the *heartbeat* attribute associated with each *remoteNetwork* element in the *SystemInfo* state variable is reseted to the min *cache-control* value in that *remoteNetwork*.

Note:since the *SystemInfo* state variable is shared with the RADAConfig and RATAConfig services, a modification of its value must be propagated internally to the other services (if present on the same device).

2.6.1.4 Control Point Requirements

The Control Point must add to the *DeviceList* only the devices that are visible to the particular remote network after applying the appropriate filters according the policies of the *SystemInfo* state variable of the RADASync service that it is co-located with.

2.6.1.5 Errors

Table 2-6: Error Codes for *AddRemoteDevices()*

ErrorCode	errorDescription	Description
400-499	TBD	See UPnP Device Architecture section on Control.
500-599	TBD	See UPnP Device Architecture section on Control.
600-699	TBD	See UPnP Device Architecture section on Control.
701	Unknown ID	Remote RADA ID not known
702	Invalid DeviceList	The DeviceList provided is not properly formatted

2.6.2 *RemoveRemoteDevices()*

This action specifies that a remote device or set of remote devices is to be removed from the collection of locally accessible devices. This action specifies the devices that require SSDP offloading.

2.6.2.1 Arguments

Table 2-7: Arguments for *RemoveRemoteDevices()*

Argument	Direction	relatedStateVariable
DeviceList	*IN*	*A_ARG_TYPE_DeviceInfoList*
ID	*IN*	*A_ARG_TYPE_UUID*

2.6.2.1.1 *DeviceList*

The collection of device(s) that are to be removed from the remote branch.

2.6.2.1.2 *ID*

The UUID of the RADA device where the invoking RADASync control point is co-located with.

2.6.2.2 Dependency on State

None.

2.6.2.3 Effect on State

When this action is invoked, the *SystemInfo* state variable must be updated so that the devices indicated in the DeviceList argument are removed from the *RemoteNetwork* element corresponding to the RADASync Control Point that invoked the action.

Additionally, the value of the *heartbeat* attribute associated with each *remoteNetwork* element in the *SystemInfo* state variable is reseted to the min *cache-control* value in that *remoteNetwork*.

Note:since the *SystemInfo* state variable is shared with the RADAConfig and RATAConfig services, a modification of its value must be propagated internally to the other services (if present on the same device).

2.6.2.4 Control Point Requirements

The Control Point must add to the *DeviceList* only the devices that are no longer visible to the particular remote network after applying the appropriate filters according the policies of the *SystemInfo* state variable of the RADASync service that it is co-located with.

2.6.2.5 Errors

Table 2-8: Error Codes for *RemoveRemoteDevices()*

ErrorCode	errorDescription	Description
400-499	TBD	See UPnP Device Architecture section on Control.
500-599	TBD	See UPnP Device Architecture section on Control.
600-699	TBD	See UPnP Device Architecture section on Control.
701	Unknown ID	Remote RADA ID not known
702	Invalid DeviceList	The DeviceList provided is not properly formatted

2.6.3 *SetDDDLocation()*

This action is used by the RADASync Control Point embedded in the server RADA to indicate to the RADASync Control Point embedded in the client RADA from where to fetch the DDD of the server RADA.

There will be situations in which a RADA will not know ahead of time the location of its peer RADA. In such a case, it will not be able to carry-out synchronization until it can associate with the peer RADA.

For example, when a user is trying to connect to their home network which is protected by a NAT firewall, the remote user’s RADA may not know the private address of the home network RADA. However in such a case, when the Remote Access Client establishes the remote transport into the home network, the Home Network’s RADA will be aware of the establishment of the remote transport.

It must then call this action to supply the Remote RADA with the necessary information for the remote RADA to associate itself with the home RADA. Once this association is established, the two RADAs can proceed to carry out synchronization.

Upon receiving this action, the co-located RADASync control point MUST immediately retrieve the device description document from the indicated location. If no description document is found at the indicated location or the UUID in the description document is not the same as the one provided in the *ID* argument, the device must respond to this action with the appropriate error code.

2.6.3.1 Arguments

Table 2-9: Arguments for *SetDDDLocation()*

Argument	Direction	relatedStateVariable
DDDLocation	*IN*	*A_ARG_TYPE_DDDLocation*
ID	*IN*	*A_ARG_TYPE_UUID*

2.6.3.1.1 *DDDLocation*

The location URL that the RADA can use to fetch the description document for the RADA that wishes to receive synchronization updates.

2.6.3.1.2 *ID*

The UUID of the RADA device where the invoking RADASync control point is co-located with.

2.6.3.2 Dependency on State

None.

2.6.3.3 Effect on State

None.

2.6.3.4 Control Point Requirements

The *DDDLocation* parameter has as value the device description document URL of the RADiscoveryAgent device that embeds the control point.

2.6.3.5 Errors

Table 2-10: Error Codes for *SetDDDLocation()*

ErrorCode	errorDescription	Description
400-499	TBD	See UPnP Device Architecture section on Control.
500-599	TBD	See UPnP Device Architecture section on Control.
600-699	TBD	See UPnP Device Architecture section on Control.
701	Unknown ID	Remote RADA ID not known
703	DDD Not Found	No device description document found at provided location.

2.6.4 *HeartbeatUpdate()*

This action defines a heartbeat mechanism that allows the RADA to detect that the corresponding remote RADA is still connected.

2.6.4.1 Arguments

Table 2-11: Arguments for *HeartbeatUpdate()*

Argument	Direction	relatedStateVariable
ID	*IN*	*A_ARG_TYPE_UUID*

2.6.4.1.1 *ID*

The UUID of the RADA device where the invoking RADASync control point is co-located with.

2.6.4.2 Dependency on State

None.

2.6.4.3 Effect on State

The effect of this action is that the value of the *heartbeat* attribute associated with each *remoteNetwork* element in the *SystemInfo* state variable is reseted to the min *cache-control* value in that *remoteNetwork*.

Note:since the *SystemInfo* state variable is shared with the RADAConfig and RATAConfig services, a modification of its value must be communicated to the other services (if present on the same device).

2.6.4.4 Control Point Requirements

Control points are invoking this action on a particular remote RADASync service at regular intervals determined by the value of the *heartbeat* attribute associated with each remoteNetwork element in the *SystemInfo* state variable.

2.6.4.5 Errors

Table 2-12: Error Codes for *HeartbeatUpdate()*

ErrorCode	errorDescription	Description
400-499	TBD	See UPnP Device Architecture section on Control.
500-599	TBD	See UPnP Device Architecture section on Control.
600-699	TBD	See UPnP Device Architecture section on Control.
701	Unknown ID	Remote RADA ID not known

2.6.5 *GetLocalNetworkAddressInfo()*

This action is optional for the RADA which does not have an ALG that translates the IP addresses for the incoming and the outgoing messages through the secure channel. This action returns the local network address information of the RAS which is co-located with the RADA.

2.6.5.1 Arguments

Table 2-13: Arguments for *GetLocalNetworkAddressInfo()*

Argument	Direction	relatedStateVariable
LocalNetworkAddress	*OUT*	*A_ARG_TYPE_NetworkAddressInfo*

2.6.5.1.1 *LocalNetworkAddress*

The subnet information that is being used on the LAN interface of the network where the RADA device resides. The *GetLocalNetworkAddressInfo()* is invoked on the local RADASync service from the RADASync CP of the remote network.

2.6.5.2 Dependency on State

None.

2.6.5.3 Effect on State

None.

2.6.5.4 Control Point Requirements

The *LocalNetworkAddress* argument enables the RADASync Control Point to detect the collision by comparing its local network address information with the *LocalNetworkAddress* argument, and if a collision is detected then theRADASync Control Point configures the ALG to translate the local IP addresses in the RADA synchronization messages along with the other messages passing through the remote access secure channel.

2.6.5.5 Errors

Table 2-14: Error Codes for *GetLocalNetworkAddressInfo ()*

ErrorCode	errorDescription	Description
400-499	TBD	See UPnP Device Architecture section on Control.
500-599	TBD	See UPnP Device Architecture section on Control.
600-699	TBD	See UPnP Device Architecture section on Control.
704		
705		

2.6.6 Error Code Summary

The following table lists error codes common to actions for this service type. If an action results in multiple errors, the most specific error should be returned.

Table 2-15: Error Code Summary

ErrorCode	errorDescription	Description
400-499	TBD	See UPnP Device Architecture section on Control.
500-599	TBD	See UPnP Device Architecture section on Control.
600-699	TBD	See UPnP Device Architecture section on Control.
700		Reserved for future extensions.
701	Unknown ID	Remote RADA ID not known
702	Invalid DeviceList	The DeviceList provided is not properly formatted
703	DDD Not Found	No device description document found at provided location.

Note: 800-899 Error Codes are not permitted for standard actions. See UPnP Device Architecture section on Control for more details.

2.7 Description of Service Requirements

2.7.1 Interaction

Remote Access Discovery Agents can only expose access to those UPnP devices which were added to the local synchronization tree. The Remote Access Discovery Agent maintains an aggregate view of these devices that will be visible outside the home network by maintaining a control point which maintains the liveliness of devices that are to be visible outside the home. The RADAConfig service is used to filter devices in the local synchronization tree, such that a subset of the available devices can be exposed.

2.7.2 Initialization

The first step in initiating Remote Access into the home network is to setup and establish the remote access transport. (See RATA, RATAConfig, InboundConnectionConfig services).

All Remote Access Client RADAs must be bound to TCP port 1900 on the remote access transport interface.

Once the remote access transport is established, the RADA residing in the Remote Access Server is notified of the remote access connection by the RATA. The RADA then attempts to establish UPnP communications with the remote RADA by attempting to retrieve the UPnP Device Description Document from port 1900 of the IP address that facilitated the remote access transport. For example, if the IP address assigned to the facilitator of the connection was 192.168.15.1, then the URL for the description document is: http://192.168.15.1:1900/.

The Remote Access Server's RADA then invokes the action "SetDDDLocation" on the Remote Access Client's RADA. Once this is completed, the two RADAs will be able to facilitate synchronization.

2.7.3 Synchronization

The RADAListener maintained by the RADA is responsible to update and flush UPnP devices from the local synchronization tree whenever these UPnP devices join/leave the local UPnP network. Whenever the view of the network changes, the RADA must update the peer RADA with these changes. For example, when a UPnP device joins the network, the RADA updates the peer RADA of this newly joined UPnP Device.

To update the peer RADA, a RADA simply calls "AddRemoteDevices/RemoveRemoteDevices", to propagate changes in the remote network view. When a remote access session is initially established, the RADAs must call "AddRemoteDevices" of the peer RADA passing only those devices that are visible there, such that the RADAs can be initially synchronized. After they are initialized, the RADAs need only update each other whenever a device leaves or joins the network.

The RADAListener behavior is described as follows:

1.) The RADAListener must establish the initial state of its local network by performing an M-SEARCH to detect all devices on the local network and notify the RADA of those devices.

2.) The RADAListener must monitor the local SSDP traffic and notify the RADA when devices are joining and leaving the UPnP network as described in the UPnP Device Architecture.

2.7.4 SSDP Offloading

UPnP Devices and Control Points utilize SSDP to facilitate discovery. In remote access scenarios, SSDP is restricted to the local network, such that the UPnP Devices and Control Points offload SSDP to the RADA on the remote network. In other words, each RADA incorporates a RADARelay, to facilitate SSDP on behalf of remote UPnP Devices and Control Points.

There are a number of reasons for this type of interaction:

1.) UPnP Discovery is timing dependent. Whenever a UPnP Control Point issues a search request onto the network, the MX field in the search request specifies a maximum number of seconds that the device may wait before sending the response. In a remoted situation, the control point may not be aware of the propagation delays of the network transport, such that the response will be missed.

    For example, if a UPnP Camera device at a local hotspot is looking for a Media Server in the home, to upload its content, it may issue a SEARCH request with an MX value of 3 seconds. The control point will have no idea how internet access is being provided. It may be connected via WiFi, but internet access may be provided by GPRS, Satellite, Dial-Up, etc. As such, the Camera may wait 5 seconds for the response, but that may be shorter than the round-trip time to the home network.

2.) UPnP Discovery is often described as a chatty protocol. The frequency of the SSDP announcements from a device or control point is dependent on the vendor of that device or control point, and may vary wildly. However, there are many times where internet access may be metered. In these scenarios, it will be important to minimize traffic to and from the home network.

3.) Removes dependency of multicast propagation from the remote access transport. Some remote access transports do not support propagation of multicast UDP packets, which would prevent UPnP discovery from functioning correctly. Other transports may support multicast propagation, but only if all hops along the connection support it and propagate it. Further, other transports may support it, but the control point or device may specify a multicast-TTL value that is too small to make it to the destination network.

The RADARelay behavior is described as follows:

1.) For each device in the remote synchronization tree of the RADA, the RADARelay must send periodic SSDP announcements (i.e. ssdp:alive) onto the local network according the UPnP Device Architecture.

2.) Whenever a device is removed from the remote synchronization tree, the RADARelay must send an SSDP expiration (i.e. ssdp:byebye) onto the local network according the UPnP Device Architecture.

3.) Whenever a RADARelay receives an SSDP Search request (i.e. M-SEARCH) for a device or service that is contained in the remote synchronization tree, it must answer the search request on behalf of the device in the remote synchronization tree according the UPnP Device Architecture.

4.) When the remote connection is broken, the RADARelay must send an SSDP expiration (i.e. ssdp:byebye) on the local network for each remote device.

2.7.4.1 Avoiding Offloading Loops

The RADA must not add to the local tree devices that are detected in the local network as a result of the RADARelay.

2.7.5 Heartbeat Mechanism

The RADA Heartbeat provides a mechanism that allows a RADA to detect if a corresponding remote RADA is still available regardless of the keep alive capabilities of the underlaying transport protocol. The mechanism is needed in order to ensure the end devices have the same responsiveness to failures (e.g. lowest cache-control delays is ensured) as if they were all present in the same physical network.

The Heartbeat is performed according to the following algorithm:

```
1.) If remote sync tree does not contain any devices go to 1
2.) Set timer as min(remote CACHE-CONTROL)
3.) While timer not elapsed
      a. If remote Add/RemoveRemoteDevice go to 1
      b. If any other remote SOAP action then reset timer
      c. If SOAP action response received then reset timer
      d. Else flush remote network
4.) Else send HeartbeatUpdate
      a. If response reset and go to 2
      b. Flush remote network
```

Independent heartbeats should be maintained between the RAS and each RAC.

2.7.6 Administrative Shutdown

A first step in administratively shutting down the RADA is to remove the local devices from all remote RADA that are connected at that particular time; the local *RADiscoveryAgent* Control Point must perform *RemoveRemoteDevice()* on all connected RADiscoveryAgent Services. When the first step is completed, the RADA must notify the *RATAConfig* service for tearing down the remote access transport connection. The notification is done by changing the *connected* flag to “false” for the respective *remoteNetwork* in the *SystemInfo* state variable.

2.7.7 Signalling IPv6 Devices to Remote Networks

The RADAListener is able to detect which devices in the local network are IPv4 and which are IPv6 by looking at the HOST header of the SSDP announcements. Additionally, the IPv4 or IPv6 addresses appear in the LOCATION header. This information is updated in the *SystemInfo* state variable and is passed to the remote RADA using the established mechanisms. The RADASync protocol is independent on the addressing mechanisms used by devices in the local network.

The RADiscoveryAgent Device is an IPv4-only device.

3 XML Service Description

```
<?xml version="1.0"?>
<scpd xmlns="urn:schemas-upnp-org:service-1-0">

   <specVersion>
      <major>1</major>
      <minor>0</minor>
   </specVersion>

   <actionList>

      <action>
         <name>AddRemoteDevices</name>
         <argumentList>
            <argument>
               <name>DeviceList</name>
               <direction>in</direction>
               <relatedStateVariable>
                  A_ARG_TYPE_DeviceInfoList
               </relatedStateVariable>
            </argument>
            <argument>
               <name>ID</name>
               <direction>in</direction>
               <relatedStateVariable>
                  A_ARG_TYPE_UUID
               </relatedStateVariable>
            </argument>
            <argument>
         </argumentList>
      </action>

      <action>
         <name>RemoveRemoteDevices</name>
         <argumentList>
            <argument>
               <name>DeviceList</name>
               <direction>in</direction>
               <relatedStateVariable>
                  A_ARG_TYPE_DeviceInfoList
               </relatedStateVariable>
            </argument>
            <argument>
               <name>ID</name>
               <direction>in</direction>
               <relatedStateVariable>
                  A_ARG_TYPE_UUID
               </relatedStateVariable>
            </argument>
            <argument>
         </argumentList>
      </action>
      <action>
         <name>SetDDDLocation</name>
         <argumentList>
            <argument>
               <name>DDDLocation</name>
               <direction>in</direction>
               <relatedStateVariable>
                  A_ARG_TYPE_DDDLocation
               </relatedStateVariable>
            </argument>
            <argument>
               <name>ID</name>
               <direction>in</direction>
               <relatedStateVariable>
                  A_ARG_TYPE_UUID
               </relatedStateVariable>
            </argument>
            <argument>
         </argumentList>
      </action>

      <action>
         <name>HeartbeatUpdate</name>
         <argumentList>
            <argument>
               <name>ID</name>
               <direction>in</direction>
               <relatedStateVariable>
                  A_ARG_TYPE_UUID
               </relatedStateVariable>
            </argument>
            <argument>
         </argumentList>
      </action>

      <!-- Declarations for other actions defined by UPnP vendor
           (if any)go here. -->

   </actionList>

   <serviceStateTable>

      <stateVariable sendEvents="no">
         <name>SystemInfo</name>
         <dataType>string</dataType>
      </stateVariable>

      <stateVariable sendEvents="no">
         <name>A_ARG_TYPE_DDDLocation</name>
         <dataType>string</dataType>
      </stateVariable>

      <stateVariable sendEvents="no">
         <name>A_ARG_TYPE_DeviceInfoList</name>
         <dataType>string</dataType>
      </stateVariable>
      <stateVariable sendEvents="no">
         <name>A_ARG_TYPE_UUID</name>
         <dataType>string</dataType>
      </stateVariable>

      <!-- Declarations for other state variables defined by UPnP 
vendor
           (if any)go here. -->

   </serviceStateTable>
</scpd>
```

4 Test

No semantic tests have been specified for this service.

Appendix A. RADiscoveryAgent Data Structures (Normative)

A.1 SystemInfo Template

The following shows the generalized layout of a SystemInfo Template. More elements and/or attributes MAY be added in future versions of SystemInfo templates.

The *forum* character style is used to indicate names defined by the Remote Access Working Committee. Implementations need to fill out the parts that are printed in *vendor* character style.

```
<?xml version="1.0" encoding="UTF-8"?>
<dads xmlns="urn:schemas-upnp-org:ra:dads"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:schemaLocation="urn:schemas-upnp-org:ra:dads
   http://www.upnp.org/schemas/ra/dads-v1.xsd">
   <systemInfo
      updateID="">
      <localNetwork>
         <rada
            uuid=""
            exportControlMode="allowList">
            <deviceInfo
               uuid=""
               cache-control=""
               descriptionDocument="URL to description document"
               server="">
               <accessControl>
                  <access credentialID=""></access>
                  <!-- Any other access (if any) go here.-->
               </accessControl>
            </deviceInfo>
            <!-- Any other deviceInfo (if any) go here.-->
         </rada>
      </localNetwork>
      <remoteNetwork>
         <rata
            credentialID=""
            connected="true"></rata>
         <rada
            uuid=""
            dddLocation=""
            importControlMode="denyList"
            heartbeat="">
            <deviceInfo
               uuid=""
               cache-control=""
               descriptionDocument=""
               server="">
               <accessControl>
                  <!-- No access elements for remote networks.-->
               </accessControl>
            </deviceInfo>
            <!-- Any other deviceInfo (if any) go here.-->
         </rada>
      </remoteNetwork>
      <!-- Any other remoteNetwork (if any) go here.-->
   </systemInfo>
```

```
</dads>
```

xml

REQUIRED for all XML documents. Case sensitive.

dads

REQUIRED. Must have “urn:schemas-upnp-org:ra:dads” as the value for the xmlns attribute; this references the UPnP Remote Access Working Committee RADA Datastructure Template Schema. As long as the same xmlns is used, the data structure template MUST be backward compatible, i.e. usable by legacy implementations.

systemInfo

REQUIRED. Contains the network images visible for the RADA. Contains the following sub elements:

@updateID

REQUIRED. xs:int. Contains the unique identifier for the current systemInfo.

localNetwork

REQUIRED. Contains the list of devices that are currently available in the local network. Contains the following sub elements:

rada

REQUIRED. Contains the discovery specific information. Contains the following subelements:

@uuid

REQUIRED. Contains the UUID of the local RADA.

@exportControlMode

REQUIRED. Describes the access control mode for what devices you want to advertise in a remote network. Possible values are “allowList” or “denyList”.

deviceInfo

OPTIONAL. Contains information about a single device in the network. MUST be present for each device present in the network. Contains the following sub elements:

@descriptionDocument

REQUIRED. xs:anyURI. Contains the pointer to the device description document.

@uuid

REQUIRED. xs: string. Contains the UUID of the root device.

@cache-control

REQUIRED. xs:int. Must have max-age directive that specifies number of seconds the advertisement is valid. Value taken from the original SSDP message

@server

REQUIRED. xs:string. Concatenation of OS name, OS version, UPnP/1.x, product name, and product version. Value taken from the original SSDP message

- For any other SSDP headers specified by the UPnP Device Architecture or a UPnP vendor the attribute name MUST be in the format: header name followed by a domain name owned by the vendor. The value of the attribute is the exact value of the header.

  Example:

  @bootid.upnp.org

accessControl

OPTIONAL. Contains the list of remote RADAs that can or cannot see this local device. Contains the following sub elements and attributes:

access

OPTIONAL. Describes the remote RADA that can or cannot see the local device. Contains the following attribute:

@credentialID

REQUIRED. contains the CredentialID of the remote RADA that is affected by this entry. If rada@exportControlMode is “allowList”, the remote RADA has visibility rights on this local device. If rada@exportControlMode is “denyList”, the remote RADA does not have visibility rights on this local device.

remoteNetwork

OPTIONAL. Contains the list of devices that are currently available in the remote network. MUST exist for each connected remote RADA.

rata

REQUIRED. Contains the transport specific information. Contains the following subelements:

@credentialID

REQUIRED. Contains the credentialID used with the remote RADA.

@connected

REQUIRED. xs:boolean. Determines if the remote RADA is currently connected.

rada

REQUIRED. Contains the discovery specific information. Contains the following attributes:

@uuid

REQUIRED. xs:string. Contains the UUID of the remote RADA.

@dddLocation

REQUIRED. xs:anyURI. Contains the URL to the UPnP description of the root device that hosts the remote RADA.

@importControlMode

REQUIRED. xs:token. Describes the access control mode for remote devices that are advertised in the local network. Possible values are “allowList” or “denyList”. If the @importControlMode attribute is “denyList” then the remote devices are not advertised in the local network; if “allowList”, then the remote devices are advertised.

@heartbeat

REQUIRED. xs:int. Contains the lowest value of Cache Control of devices that are in this particular remote network.

deviceInfo

OPTIONAL. Contains information about a single device visible in the remote network. MUST be present for each device present in the network. Contains the following sub elements and attributes:

@descriptionDocument

REQUIRED. xs:anyURI. Contains the pointer to the device description document.

@uuid

REQUIRED. xs:string. Contains the UUID of the root device.

@cache-control

REQUIRED. xs:int. Must have the value of the max-age directive that specifies number of seconds the advertisement is valid. Value taken from the original SSDP message

@server

REQUIRED. xs:string. Concatenation of OS name, OS version, UPnP/1.0, product name, and product version. Value taken from the original SSDP message

- For any other SSDP headers specified by the UPnP Device Architecture or a UPnP vendor the attribute name MUST be in the format: header name followed by a domain name owned by the vendor. The value of the attribute is the exact value of the header.

Example:

@bootid.upnp.org

accessControl
OPTIONAL. MUST be empty.

A.2 DeviceInfoList Template

The following shows the generalized layout of a SystemInfo Template. More elements and/or attributes MAY be added in future versions of SystemInfo templates.

The *forum* character style is used to indicate names defined by the Remote Access Working Committee. Implementations need to fill out the parts that are printed in *vendor* character style.

```
<?xml version="1.0" encoding="UTF-8"?>
<dads xmlns="urn:schemas-upnp-org:ra:dads"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:schemaLocation="urn:schemas-upnp-org:ra:dads
   http://www.upnp.org/schemas/ra/dads-v1.xsd">
   <deviceInfoList>
      <deviceInfo
         uuid=""
         cache-control=""
         descriptionDocument=""
         server=""/>
      <!-- Any other device info (if any) go here.-->
   </deviceInfoList>
</dads>
```

xml
REQUIRED for all XML documents. Case sensitive.

dads
REQUIRED. Must have “urn:schemas-upnp-org:ra:dads” as the value for the xmlns attribute; this references the UPnP Remote Access Working Committee RADA Datastructure Template Schema. As long as the same xmlns is used, the data structure template MUST be backward compatible, i.e. usable by legacy implementations.

deviceInfoList
REQUIRED. Contains the list of devices visible for the remote RADA. Contains the following sub elements:

deviceInfo
REQUIRED. Contains the details of a device. MUST be present for each device visible in the remote network. Contains the following attributes:

@uuid
REQUIRED. xs:string. Contains the unique id of the device.

@descriptionDocument
REQUIRED. xs:anyURI. Contains URL to the description document of the device.

@cache-control
REQUIRED. xs:int. Must have the value of max-age directive that specifies number of seconds the advertisement is valid. Value taken from the original SSDP message

@server
REQUIRED. xs:string. Concatenation of OS name, OS version, UPnP/1.0, product name, and product version. Value taken from the original SSDP message

- For any other SSDP headers specified by the UPnP Device Architecture or a UPnP vendor the attribute name MUST be in the format: header name followed by a domain name owned by the vendor. The value of the attribute is the exact value of the header.

  Example:

  @bootid.upnp.org

Appendix B. ALG configuration for address translation

As discussed in the Section 2.6.5.4 of Control Point Requirements of *GetLocalNetworkAddressInfo()*, the RADASync Control Point can benefit from detecting the IP address collision between the local and the remote network. Once the collision is detected, the RADASync Control Point configures the ALG (Application Level Gateway) which is co-located with the RA Transport Agent to translate the IP addresses of the incoming and the outgoing traffic passing through the remote access secure channel provided by the RA Transport Agent. The ALG implementation needs to buffer the traffic and search the local IP address and replace those IP addresses with the translated IP addresses. The way to select the new network address to be used to change the local IP address could be different for different ALG implementations. The following figure illustrates an example of an ALG that selects the IP address from the address space that are being used for the RADA for the remote access channel. The port number is allocated randomly for the local device. This ALG approach is connection agnostic. This could be used for both over the top clear and VPN connections.

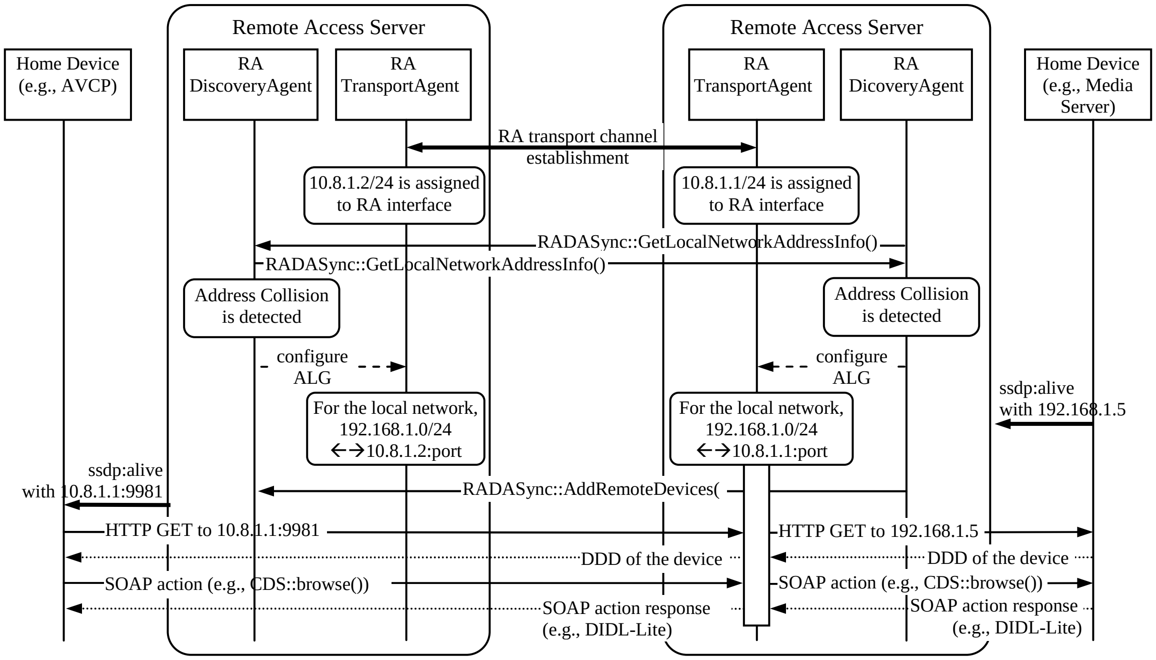
Remote Access Server
Remote Access Server
Home Device (e.g., AVCP)
RA DiscoveryAgent
RA TransportAgent
RA TransportAgent
RA DicoveryAgent
Home Device (e.g., Media Server)
RA transport channel establishment
10.8.1.2/24 is assigned to RA interface
10.8.1.1/24 is assigned to RA interface
RADASync::GetLocalNetworkAddressInfo()
RADASync::GetLocalNetworkAddressInfo()
Address Collision is detected
Address Collision is detected
configure ALG
configure ALG
For the local network, 192.168.1.0/24 ←→10.8.1.2:port
For the local network, 192.168.1.0/24 ←→10.8.1.1:port
ssdp:alive with 192.168.1.5
ssdp:alive with 10.8.1.1:9981
RADASync::AddRemoteDevices(
HTTP GET to 10.8.1.1:9981
HTTP GET to 192.168.1.5
DDD of the device
DDD of the device
SOAP action (e.g., CDS::browse())
SOAP action (e.g., CDS::browse())
SOAP action response (e.g., DIDL-Lite)
SOAP action response (e.g., DIDL-Lite)

Figure B-1 ALG Configuration for translation (full proxy architecture)

The ALG implementation should manage a mapping table with 3 tuples: local IP address, assocated port number, and target port number being used for translation. When the local IP address (and port) changes, the ALG should search the mapping table to find the target port number allocated to the IP address (and port).

It is possible that the ALG can reside only in one side of the network. In that case, the ALG implementation should translate the IP addresses of the incoming and the outgoing traffic of the devices on the local and the remote network.

The ALG implementation should have following the requirements:

- It should be activated when the IP address collision is detected.

- It should manage a mapping table to be used for the IP address translation of the network. The ALG should search the table and add a new pair of IP address and port number to the table when the IP address and the port does not exist in the table.
- The ALG should change the IP address (and port) in the payload of the packet. Options to accomplish this can range from full proxy architecture to a payload translation on the fly. The ALG should update the relevant header (e.g., HTTP Length header) if the change of the IP address changes the total length of the payload.
- Once the ALG is configured, all the traffic including HTTP and RTP should be relayed by the ALG. For example, if the ALG receives the CDS::browse() action from the remote network, it will translate the IP addresses of the action according to the mapping table maintained by the ALG.