Remote Access Architecture:2

For UPnP™ Version 1.0
Status: Standardized DCP
Date: April 30, 2011
Document Version: 1.0
Service Template Version: 2.00

Authors	Company
Bill Russell	Canon
Tom Lawrence	Canon
Bich Nguyen (Co-Chair)	Cisco
Mark Baugher	Cisco
Sridhar Ramaswamy	Cisco
Mark Enright	Cisco
Ayodele Damola	Ericsson
Rémi Bars	Orange
Yu Zhu	Huawei
Bryan Roe	Intel
Gunner Danneels	Intel
Ally (Yu-kyoung) Song	LGE
Alexander Kokhanyuk	Motorola

Authors	Company
Jim Morikuni	Motorola
Vlad Stirbu	Nokia
Cathy Chan	Nokia
Jan Brands	NXP
Marco Kuystermans	NXP
Daniel Meirsman	Philips
Suresh Gangadharan	Philips
Jeffrey Kang	Philips
Wouter van der Beek	Philips
Shrikant Kanaparti	Samsung
Mahfuzur Rahman (Co-Chair)	Samsung
Se-Hee Han	Samsung
Sander Smith	Sericon Technology
Bruce Fairman	Sony
Jack Manbeck	Texas Instruments
Rami Kohanim	Universal Devices

The 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 document describes an architecture that provides the infrastructure that allows generic UPnP devices, services and control points deployed in remote physical devices to interact with the corresponding UPnP devices, services and control points physically attached to the home network. The mechanisms defined in this architecture will allow to extend the home network so that it will logically include the remote devices so that all devices will be able to communicate among themselves using the UPnP Forum defined mechanisms, e.g. UDA. The desired behavior of the interactions between the remote device and home devices is envisioned to be similar with the one expected as if all devices are located in the same local area network.

In order to accommodate the above mentioned goals, the Remote Access Architecture will provide means to connect the two segments of the extended home network using established mechanisms. The architecture recognizes that there might be several possible alternative models to "bridge" the two segments and will provide an interface that will allow them to be plugged, while enforcing the same overall behavior of the whole system regardless of the model used.

The architecture does not describes any interfaces to "service" gateways that will enable non-UPnP entities to interact with the UPnP devices, services and control points physically attached to the home network.

1.1 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".
- 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.

1.2 References

1.2.1 Informative References

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

[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.

[DEVICE-IPv6] – UPnP Device Architecture, version 1.0., Annex A – IP Version 6 Support.
Available at: http://www.upnp.org/resources/documents/AnnexA-IPv6_000.pdf

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

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

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

[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.

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

[IGD] – InternetGatewayDevice:1, UPnP Forum, November, 2001
Available at: http://www.upnp.org/specs/gw/igd1?.

[RFC 1889] – IETF RFC 1889, RTP: A Transport Protocol for Real-Time Applications, H. Schulzrinne, S. Casner, R. Frederick, V. Jacobson, January 1996.
Available at: http://www.ietf.org/rfc/rfc1889.txt.

[RFC 1918] – IETF RFC 1918, *Address Allocation for Private Internets*, Y. Rekhter, et. Al, February 1996
Available at: http://www.ietf.org/rfc/rfc1918.txt

[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 2131] – IETF RFC 2131, *Dynamic Host Configuration Protocol*, R. Droms, March 1997
Available at: http://www.ietf.org/rfc/rfc2131.txt

[RFC 2516] – IETF RFC 2516, *A Method for Transmitting PPP Over Ethernet (PPPoE)*, L. Mamakos, et. Al, February 1999
Available at: http://www.ietf.org/rfc/rfc2516.txt

[RFC 3056] – IETF RFC 3056, *Connection of IPv6 Domains via IPv4 Clouds*, B. Carpenter, K. Moore, February 2001
Available at: http://www.ietf.org/rfc/rfc3056.txt

[RFC 3489] – IETF RFC 3489, *STUN - Simple Traversal of User Datagram Protocol (UDP) Through Network Address Translators (NATs)*, J. Rosenberg, et. Al, March 2003
Available at: http://www.ietf.org/rfc/rfc3489.txt

[RFC 3550] – IETF RFC 3550, *RTP: A Transport Protocol for Real-Time Applications*, H. Shulzrinne, et. Al, July 2003
Available at: http://www.ietf.org/rfc/rfc3550.txt

[RFC 4380] – IETF RFC 4380, *Teredo: Tunneling IPv6 over UDP through Network Address Translations (NATs)*, C. Huitema, February 2006
Available at: http://www.ietf.org/rfc/rfc4380.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

[ETSI ES 282 001 (2009)] – TISPAN: NGN Functional Architecture, Feb. 2009.
Available at: http://portal.etsi.org/docbox/TISPAN/Open/NGN_LATEST_DRAFTS/RELEASE3/02067-ngn-r3v330.pdf

[TISPANCust] – TISPAN Customer Premises Networks: Protocol Specification.
Available at: http://pda.etsi.org/exchangefolder/ts_185010v020101p.pdf
[RFC 4787] – IETF RFC 4787, *Network Address Translation (NAT) Behavioral Requirements for Unicast UDP*, C. Jennings, F. Audet, January 2007[RFC 2136] – IETF RFC 2136, *Dynamic Updates in the Domain Name System (DNS Update)*, P. Vixie, et. al, April 1997

[RFC 2782] – IETF RFC 2782 – *A DNS RR for specifying the location of services (DNS SRV)*, A. Gulbrandsen, et. al, February 2000

[RFC 5389] – IETF RFC 5389, *Session Traversal Utilities for NAT (STUN)*, J. Rosenberg, et. al,, October 2008

[BEHAVE TURN] – IETF Internet Draft, *Traversal Using Relays around NAT (TURN): Relay Extensions to Session Traversal Utilities for NAT (STUN), draft-ietf-behave-turn-16*, J. Rosenberg, July 2009

[RFC 5128] – IETF RFC 5128, *State of Peer-to-Peer (P2P) Communication accress Network Address Translators (NATs)*, P. Srisuresh, et. al., March 2008

[P2P Com] – Proceedings of the Usenix 2005 Annual Technical Conference, pp. 179-192, *Peer-to-Peer Coummunication Across Network Address Translators*, B. Ford, P. Srisuresh, D. Kegel,, March 2005]

[DLNA Design Guidelines] – DLNA Networked Device Interoperability Guidelines, Available at: http://www.dlna.org/industry/certification/guidelines/

1.3 Terms and Abbreviations

1.3.1 Abbreviations

Table 1-1: Abbreviations

Definition	Description
DHCP	Dynamic Host Configuration Protocol
DNS	Domain Name System
FQDN	Fully Qualified Domain Name
ICC	Inbound Connection Config
IGD	Internet Gateway Device
IPsec	IP Security
NAT	Network Address Translation
RAC	Remote Access Client
RADA	Remote Access Discovery Agent
RAS	Remote Access Server
RATA	Remote Access Transport Agent
STUN	Simple Traversal of UDP Through NATs
TLS	Transport Layer Security

1.3.2 Terms

1.3.2.1 Credentials

The term credentials refer to certificates, shared secrets or other means of authentication used in the RATA context.

1.3.2.2 IPv6 Support in Transport Agents

A Transport Agent is said to have IPv6 support if it allows the local and remote devices to interact according to the UPnP Device Architecture over IPv6 [DEVICE-IPv6].

1.3.2.3 Local Device

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

1.3.2.4 Management Console

The collection of control points that are used to setup, manage and monitor the operations related to Remote Access.

1.3.2.5 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.

1.3.2.6 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 (e.g. 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 (e.g. ssdp:byebye) onto the local network according the UPnP Device Architecture.

3.) Whenever a RADARelay receives an SSDP Search request (e.g. ssdp 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.

1.3.2.7 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.

1.3.2.8 Remote Access Network Interface

The RA network interface is the network interface that is created by the Remote Access Transport Agent. The settings for this interface are contained in a RATA profile.

1.3.2.9 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. The Remote Access Server can be the residential router, a personal computer or any 3rd party dedicated device.

1.3.2.10 Remote Access Transport Agent Profile

A RATA profile is a configured RATA connection ready to be used by either accepting connections on the RAS side or to initiate connections on the RAC side.

1.3.2.11 Remote Device

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

2 Introduction

UPnP technology was envisioned to be deployed in local area networks. This initial design goal leads to some decisions which will pose some challenges when trying to expand the original scope of the UPnP technology beyond the physical boundaries of local area networks such as those found at home. For example, the discovery step described in UPnP Device Architecture v1.0 involves multicast messages that will be difficult to forward them beyond the home network due to the fact that a typical internet router will discard such messages.

Remote Access to UPnP Networks enables a remote UPnP Device or UPnP Control Point to connect to the home network and interact with the UPnP entities physically attached to the home network. During this process it is expected that the remote user will experience the remote device behaving in a similar way as in the home network. In practice, the overall user experience will be degraded due to the limitations induced by external factors, such as network latencies and bandwidth, but nevertheless the remote device will have to go to the same steps (e.g. IP addressing, discovery, description, control, eventing) as any UPnP device present in a home network.

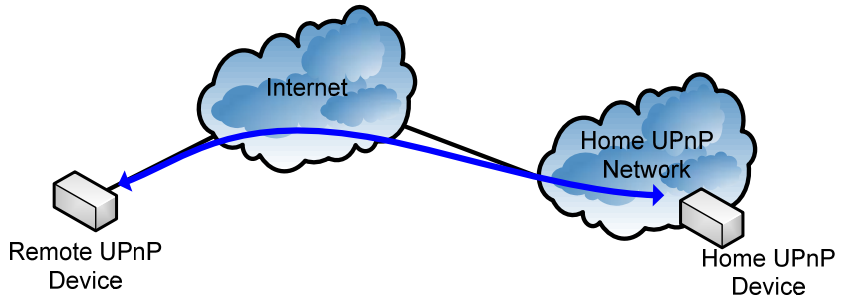


Figure 2-1: UPnP Remote Access.

Remote Access architecture undertakes the provisions needed in order to minimize the adverse effects of the internal and external factors and bring the remote user experience as close as possible to the one available in the local area network.

In a home to home scenario two networks are connected with each other via the remote access gateway devices. This allows services and devices of one home to be accessible to services and devices of another home and vice-versa.

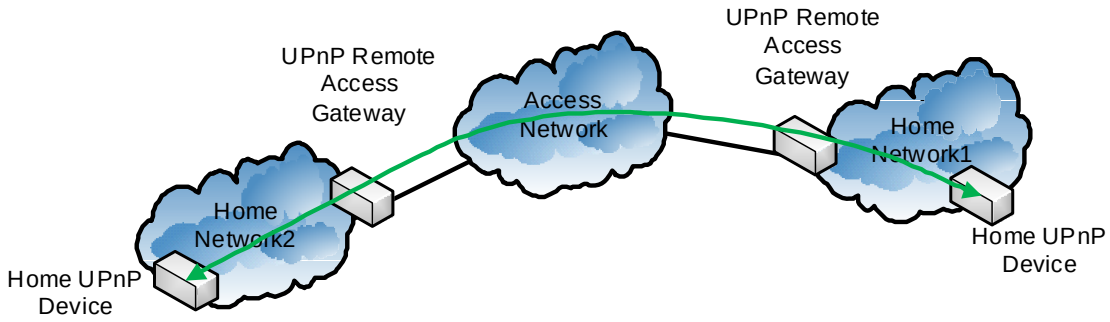


Figure 2-2: UPnP Home to home Remote Access.

3 Operational Considerations

3.1 Remote Access Environment

A typical Remote Access network scenario is depicted in Figure 3-1, where we have a remote UPnP entity that connects to the home UPnP network in order to interact with the home UPnP entities. The remote UPnP entity connects to the internet via an access network and establishes a remote access connection to the home network via public internet backbone to the Remote Access server that has access to the internet via the ISP network. There are cases when, due to various reasons, the remote UPnP entity cannot establish remote access connection by itself and it needs the help of some support services that can be hosted in the ISP network or can be hosted in the internet by a 3rd party service provider.

In this environment, it is highly probable that one or more of the intermediate network elements deploying NAT and firewall techniques will break the end-to-end connectivity between the remote device and home devices. Generally speaking, the problems introduced by NAT boxes are well understood and protocol designers have created built-in capabilities to transverse several NATs on the initiator side. However, on the receiving side, things are more complex and hosts behind NATs are not able to accept connections without support from some 3rd party support services located in the public internet. Due to technology availability, this version of Remote Access Architecture will provide solutions for Remote Access connectivity to a limited number of network scenarios. Supported network scenarios will increase in future versions of this document as required standards are developed.

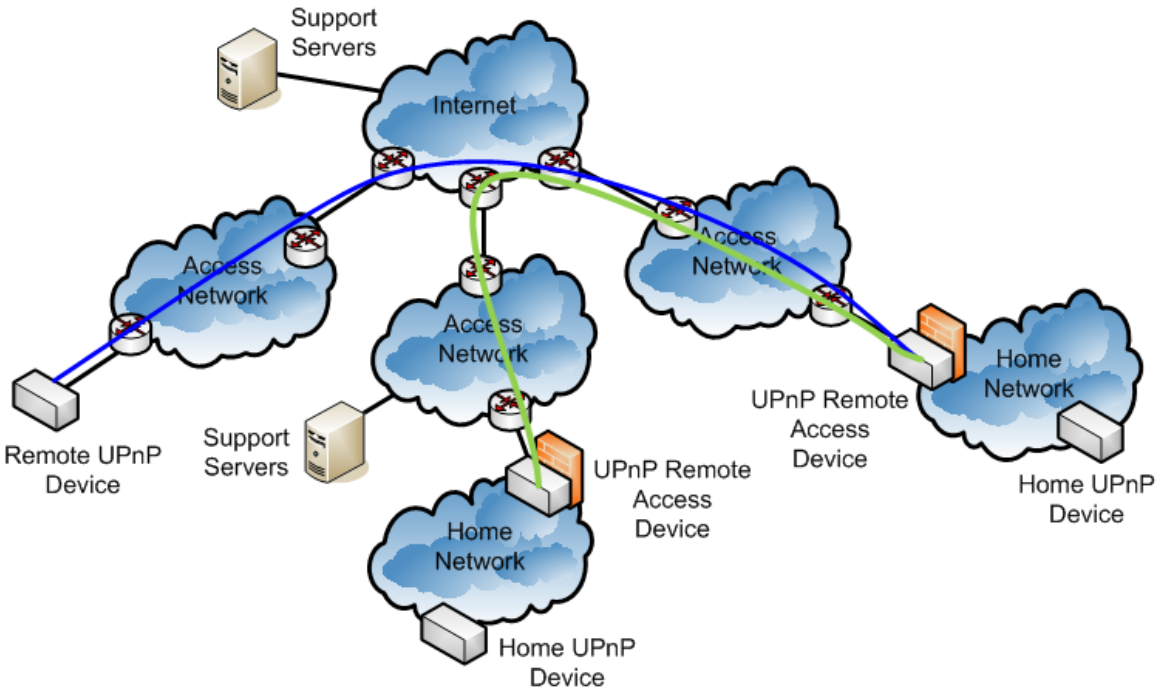


Figure 3-1: Remote Access Environment.

3.2 Access Networks

Remote Access can be initiated from different locations using multiple device categories, access technologies and having various network topologies between the remote device and the home network.

Remote Access Architecture v1.0 provides support for mainly three types of access networks: service provider networks (e.g. airport hotspots or hotel provided networks), mobile operator networks and visited home networks. Additionally, the current version of the Remote Access Architecture provides limited

support for using corporate networks as access networks in the cases where the corporate infrastructure (e.g. firewalls and proxy servers) are not actively enforcing a policy that forbids the use of “Remote Access to UPnP Networks”. This document will not provide guidelines for bypassing firewalls or proxy servers or guidelines on how to install software implementing UPnP Remote Access functionality on equipments where the user doesn’t have rights to do so.

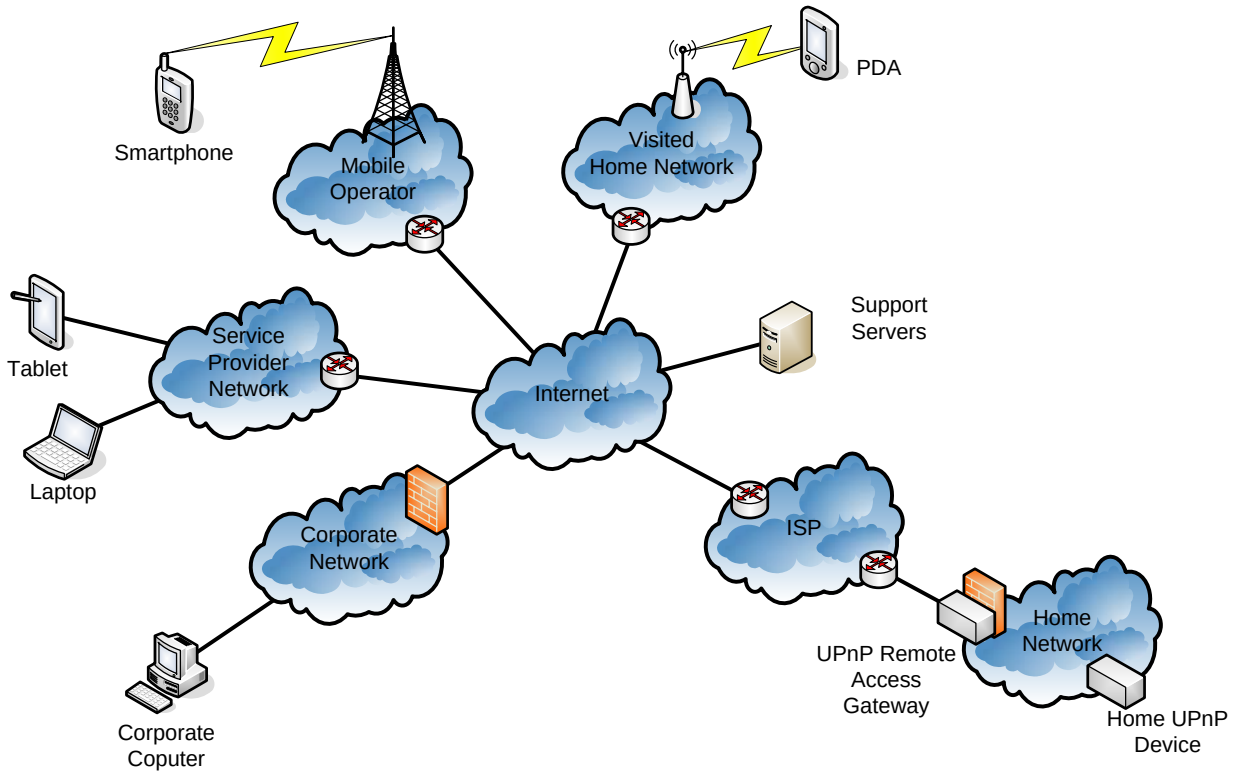


Figure 3-2: Access Networks.

3.2.1 IPv4 Addressing and NAT issues

Due to the shortages in the available IPv4 addresses, service providers are forced to deploy NAT devices to serve more devices behind a single public IP address by allocating them private IP addresses [RFC 1918].

The presence of NAT devices in the intermediate nodes between the remote device and the remote access server is generally transparent to end points and breaks the end-to-end connectivity. The effects of NAT devices are typically different depending on the role of the device in the communication channel, e.g. initiator or receiver.

3.2.1.1 Remote Device

In general, protocols that can be used in order to enable the remote access to home networks have the ability to traverse NAT devices on the initiator side. In our scenarios, the remote device is the initiator and it should be able to use both public and private addresses.

3.2.1.2 Remote Access Server

The Remote Access Server can be hosted by the residential gateway or it can be hosted by a separate device in the home network, e.g. stand-alone device or software component in a PC.

When the Remote Access Server is co-located with the residential gateway, the IP address of the external interface is allocated by the ISP using specific means, e.g. DHCP [RFC 2131], PPPoE [RFC 2516], etc. The ISP can allocate a public routable IP address, which can be either static of dynamic, or it can allocate a private IP address in the case it deploys a NAT device in its network. At his point in time, due to lack of an established standard mechanism to traverse the NAT boxes deployed in the ISP network, Remote Access Architecture v1.0 will support only the scenarios where the external interface of the residential gateway has a public routable IP address.

The other alternative is to have the Remote Access Server deployed in a stand alone device or in a PC. In this scenario the Remote Access Server acquires an IP address from the DHCP server located in the residential gateway. For this option to work, following requirements have been identified:

- Ability to add a UPnP device routing entry into the residential gateway is necessary (ideally using IGD from the RAS)
- RAS needs to support routing functionality

This version of the specification does not specify these mechanisms, which will be specified in the next version of the specifications.

Mechanisms to detect NAT presence in the ISP network and to inform the user about this situation are described in Section 4.4.1.2.

3.2.2 IPv6 Addressing

If an access network provides native IPv6 connectivity, the Remote Access Client may use it depending on IPv6 support available in the home network. The Remote Access client may use IPv6 even if the access network is providing only IPv4 connectivity, via some generic IPv4-IPv6 transition mechnanisms, e.g. 6to4 [RFC 3056] or Teredo [RFC 4380].

3.3 Home Network Environment

3.3.1 IPv4 Support in Home Routers

The Remote Access Architecture supports home networks having IPv4 connectivity. The home router may provide public or private IPv4 addresses to the home devices.

3.3.2 IPv6 Support in Home Routers

The Remote Access Architecture supports home networks having IPv6 connectivity. The home router should provide 6to4 or native IPv6 addresses to the home devices.

3.4 Support Services in the Public Network

3.4.1 Server Name Resolution

Dynamic DNS is a system for allowing an Internet domain name to be assigned to a varying IP address. This makes it possible for other sites on the Internet to establish connections to a machine having dynamic IP address (e.g. a cable or DSL service where the IP address of the modem is changed by the ISP occasionally), without needing to track the IP addresses themselves.

To implement dynamic DNS it is necessary to set a maximum caching time of the domain to an unusually short period (typically a few minutes). This prevents other sites on the Internet from retaining the old address in their cache, so that they will typically contact the name server of the domain for each new connection.

The use of the DNS is recommended, regardless if the IP address is static or dynamic, as IPv4/IPv6 addresses are too long to be typed by a user on a regular basis.

3.4.2 Detecting NAT and NAT Type on Server Side

A typical consumer is not aware if his ISP is deploying NAT boxes into its network or not. So, in order to determine if its network setup supports the Remote Access feature a mechanisms is needed to determine the presence of NAT boxes in the ISP network.

STUN [RFC 3489] is a network protocol allowing clients behind NAT (or multiple NATs) to find out its public address, the type of NAT it is behind and the internet side port associated by the NAT with a particular local port.

The STUN client embedded in the Remote Access Server sends a request to a STUN server. The server then reports back to the STUN client what the public IP address of the NAT router is, and what port was opened by the NAT to allow incoming traffic back in to the network. The response also allows the STUN client to determine what type of NAT is in use, as different types of NATs handle incoming UDP packets differently. It will work with three of four main types: full cone NAT, restricted cone NAT, and port restricted cone NAT. It will not work with symmetric NAT (also known as bi-directional NAT).

Remote Access Architecture v1.0 is using STUN to detect only the existence of the NAT in the ISP network and to inform the user that remote access connection cannot be established due to the ISP's NAT boxes.

Remote Access will add support for traversing NAT devices in the ISP network in future releases as standard mechanisms to traverse symmetric NAT mature.

4 Remote Access Reference Architecture

The remote access architecture caters for two RA scenarios. The first enables a remote device to access a UPnP network via a secure transport channel. The remote device becomes part of the home network and can access its devices and services. The second scenario allows two UPnP home networks to be connected together and make their respective devices part of a common joined network for the duration of the RA connection.

4.1 Remote Access Architecture Paradigm

The Remote Access Architecture envisions to recreate the UPnP experience for devices that are not physically attached to the home network. There are two concepts that make this vision possible: a transport channel, which provides the security for UPnP Device Architecture protocols and for any associated protocols that are used in the context of various DCPs (e.g. RTP [RFC 3550]), and a Discovery Agent, which enables a UPnP device or service to be visible in a remote location and controls the visibility of these devices according to some filters configured by the home owner.

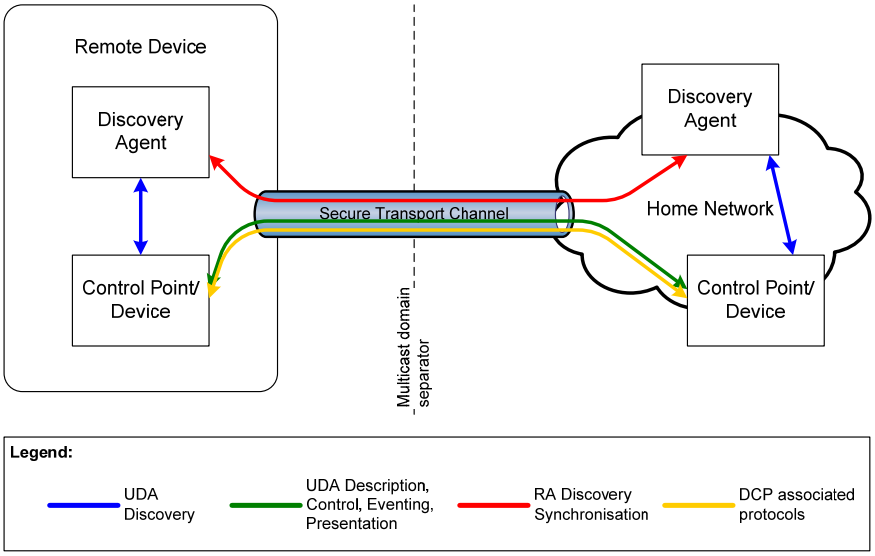


Figure 4-1: Remote Access Architecture Paradigm.

The experience provided by the Remote Access Architecture is similar to the one encountered in home, with certain limitations due to the available bandwidth on the path between the remote device and the home network.

4.2 Remote Access Components Overview

This section provides an overview of the Remote Access architecture components.

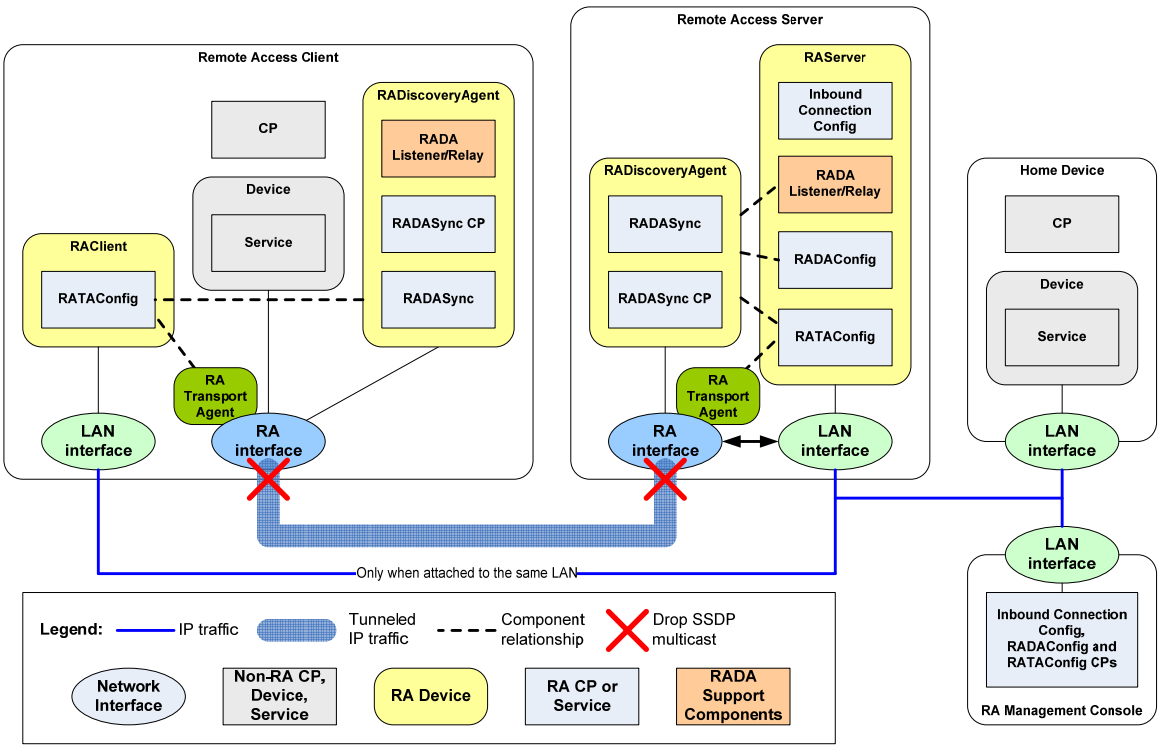


Figure 4-2: Remote Access Components Overview.

The Remote Access Secure Channel can be implemented using different mechanisms. In the context of this document the Remote Access Transport Agent (RATA) provides the secure communication channel between the remote device and the home network. The RATA parameters and options are configured by the remote access administrator via the RATAConfig service. A RATA may support multiple transport mechanisms, e.g. IPsec, TLS, etc.

Typically, the RATA connection is initiated by the remote device, thus the Remote Access Server (RAS) located in the home network needs to be discoverable and reachable from the Internet. The InboundConnectionConfig service allows the remote access administrator to verify if the RAS is reachable and to configure the settings that will allow the Remote Access Client to establish a RATA connection to the RAS.

The functionality of the Network Image Aggregator is provided by the Remote Access Discovery Agent (RADA) device together with the associated RADA Listener and RADA Relay functionalities. The RADA Listener is a control point that is constantly monitoring the SSDP messages in the local network allowing it to detect when devices are joining and leaving the network or when they are changing status. All changes detected by the RADA Listener are feed to the RADA.

The RADA has two components: the RADASync service and the RADASync Control Point. The role of the RADASync is to act as a synchronization sink allowing a RADASync Control Point, acting as a synchronization source, to push network image information about a remote network. This is a one-way sync process and in order to synchronise both network endpoints there is a need for two RADASync relationships, one form each direction. The synchronization process may be asymmetric and is determined by the filters that are configured by the remote access administrator via the RADAConfig service.

A local RADA is informed about the status changes in a UPnP remote network by the corresponding remote RADA. Those changes are notified to the RADA Relay that is reconstructing the original SSDP messages, which were sent by the remote devices, and distributes them into the local network. Additionally, the RADA Relay will respond on behalf of the remote devices to SSDP queries issued in the local network.

The multicast domain separation is done by the the routing module that prevents the UPnP multicast traffic to travel inside the remote access secure channel provided by RATA.

To make possible a home to home interaction extra functionality is required to the set define above . The figure below illustrates the set of components which enable two home networks connect to each other via the RAS devices.

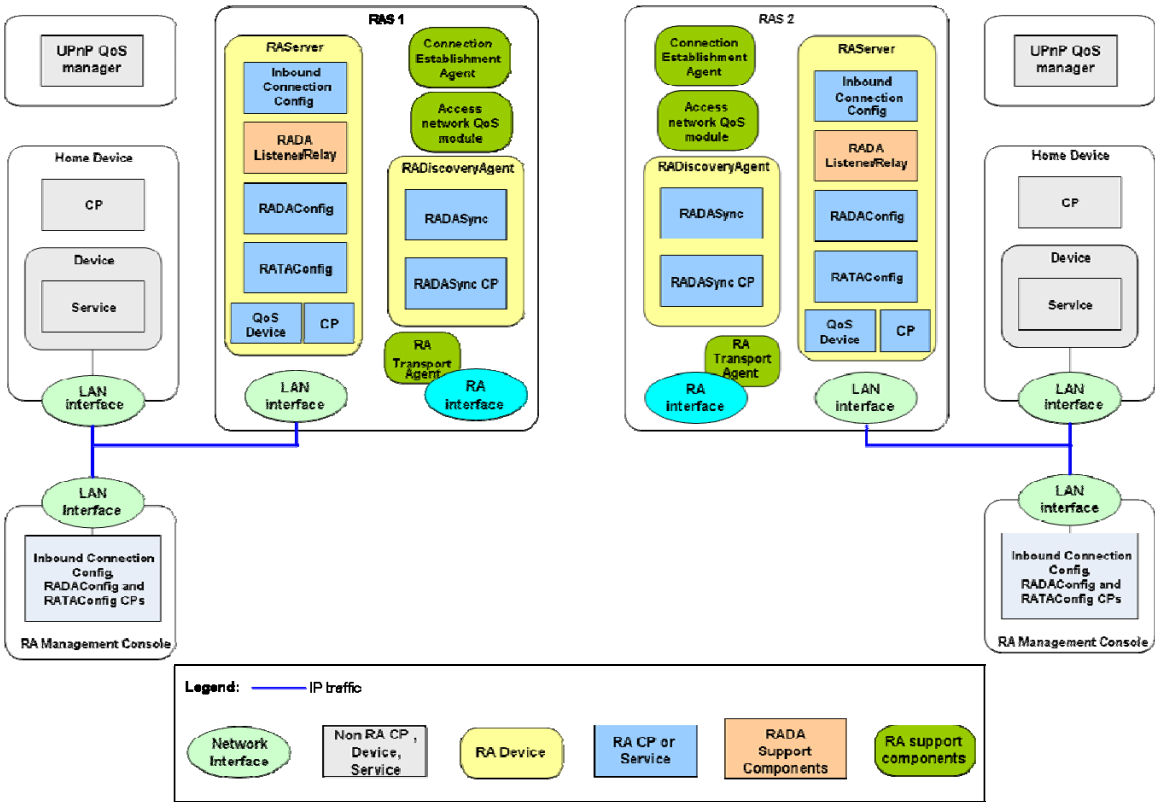


Figure 4-3: Home to Home Remote Access Components.

4.3 Remote Access Phases Overview

This section describes when and how the Remote Access components are used.

4.3.1 Setup Services

Before an UPnP device can be used remotely, the user has to configure the Remote Access Server and the Remote Access Client devices so that they can establish a Remote Access Transport channel between them. This step is considered to be the Remote Access Setup phase and usually takes place while both Remote Access Client and Remote Access Server are in the home network.

The Remote Access Architecture allows using multiple out-of-band mechanisms to enable the Remote Access Transport. In the context of this document, the functionality required for establishing the communication channel using the Remote Access Transport will be named the Remote Access Transport Agent (RATA). A different RATA is required for each RAT mechanism.

The Remote Access Architecture will provide a common configuration interface (e.g. RATAConfig) that will allow the configuration of each RATA. The interface will be the same for both RAC and RAS.

Additionally, in order to enable the reachability of the RAS from the internet the RAA will provide a configuration interface that will enable inbound connections, e.g. InboundConnectionConfig.

4.3.2 Operational Services

The Remote Access Usage phase takes place immediately after the remote device has successfully established a connectivity channel over the RAT. In this phase, the standard devices, services and control points embedded in the RAC can communicate using the mechanisms provided by the UPnP Device Architecture with the corresponding devices, services and control points physically attached to the home network.

In order to make this step possible, the RAA provides a Remote Access Discovery Agent (RADA), which has the role of "mirroring" the discovery messages from the home network to the remote device and vice-versa. Additionally, the RADA may apply some filters that will restrict the visibility of the home devices or service from the remote control points or vice-versa.

The RADAs located in the RAC and the RAS communicate with each other using a synchronization SCPD which allows them to push updates to each other.

RADA is kept is sync with the status of the local UPnP network by RADA Listener, a component that is constantly monitoring the advertisements in the network and is informing the RADA when devices are joining or leaving.

Another support function of the RADA is the RADA Relay that performs the task of responding to discovery queries on behalf of a remote device. Another function of the RADA is to notify local devices when remote devices are joining or leaving the remote network. In order to perform thise two tasks, the RADA Relay is constantly checking the information on remote devices maintained by the RADA.

4.3.3 Management Service

The RAA defines a management interfaces that allow the configuration of the Access Control Lists (ACL) that will restrict the visibility of the home devices from the remote control points (e.g. RADAConfig).

4.4 Remote Access Functionalities

4.4.1 Inbound Connection Configuration

This component provides the features that enable the end user to determine if a Remote Access Server can be deployed in the home network by checking if the Remote Access Server is reachable from the Internet.

4.4.1.1 Server Naming

The lack of available IPv4 addresses prevents ISPs to allocate an IP address to each WAN interface of the residential gateways. To overcome this problem, service providers are dynamically allocating public IP addresses only to those gateways that are connected to the internet. This behavior will make difficult for an end user to connect to his home network when the IP addresses are dynamically allocated. The DNS System provides a way to associate IP addresses with Fully Qualified Domain Names, which are easily remembered by humans comparing to number sequences. Dynamic DNS is a system for allowing an Internet domain name to be assigned to a varying IP address.

A dynamic DNS client is colocated with the RAS so that the DNS server is notified whenever the IP address of the RAS has changed in order to update the DNS records with the latest information. When the RAS is not collocated with the residential gateway it can find the public address of the gateway using Internet Gateway Device means.

4.4.1.2 NAT Detection

Shortages in the available IPv4 address have lead to the deployment of Network Address Translators. While providing many benefits, NATs also come with many drawbacks. The most troublesome of those drawbacks is the fact that it breaks the end-to-end connectivity which in turns breaks many IP applications.

Therefore, in a consumer environment such as the one encountered in the home UPnP network, it is of paramount importance to detect the elements what will hinder the establishment of the Remote Access connection.

Simple Traversal of UDP through NAT (STUN) is a lightweight protocol that allows applications to discover the presence and types of NATs and firewalls between them and the public Internet. It also provides the ability for applications to determine the public IP addresses allocated to them by the NAT. STUN works with many existing NATs, and does not require any special behavior from them. As a result, it allows a wide variety of applications to work through existing NAT infrastructure.

A typical NAT deployment configuration that can be found in home networks is described in Figure 5. Lack of available public IPv4 addresses may determine ISPs to deploy NAT devices in their network in order to serve increasing numbers of customers. Considering the NAT functionality that is found in residential gateways (e.g. DSL or Cable routers), it is fair to say that a typical consumer will have at most two NAT devices between the devices located in his home network and the Internet.

Usually the residential gateways have built-in UPnP Internet Gateway Device functionality enabling control points located in home devices can find what the WAN IP address of the gateway is and create port mappings so that incoming connections to certain port numbers to be forwarded to a machine located in the home network. This functionality allows us to consider that the residential gateway having UPnP functionality is equivalent with a full cone NAT device.

In order to detect if a NAT device is deployed in the ISP network, the Remote Access Server needs to have STUN client functionality. The client sends a request to a STUN server, and the server returns a response, by which the client is able to detect the NAT type, e.g. full cone, restricted cone, port restricted cone or symmetric.

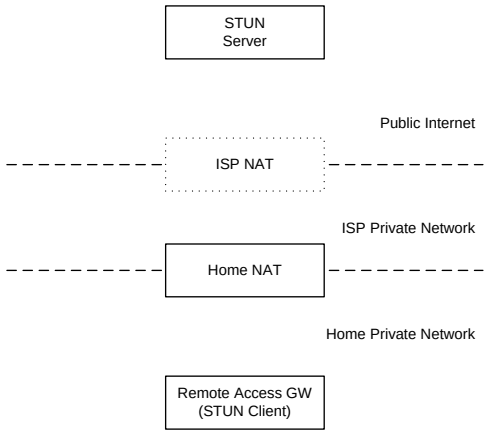


Figure 4-4: Typical STUN Configuration in Home Networks

A simplified variation of this scenario is when the Remote Access Server is collocated with the residential gateway and the only NAT device of concern is the one potentially deployed in the ISP network.

A more restrictive scenario is typically found in apartment buildings which share the same internet connection for all users. In this scenario there might be an additional NAT device between the residential gateway and NAT deployed in the ISP network.

In the event of multiple NATs between the client and the Internet, the type that is discovered will be the type of the most restrictive NAT between the client and the Internet. The types of NAT, in order of restrictiveness, from most to least, are symmetric, port restricted cone, restricted cone, and full cone.

The current version of the Remote Access Architecture supports only full cone NAT between the Remote Access Server and the public Internet. Future versions of the Remote Access architecture will use the information collected through STUN to add support for traversing NAT devices in the ISP network, as standard mechanisms to traverse symmetric NAT mature.

4.4.2 Remote Access Discovery Agent

4.4.2.1 Discovery Information Aggregation

A 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 remote device.

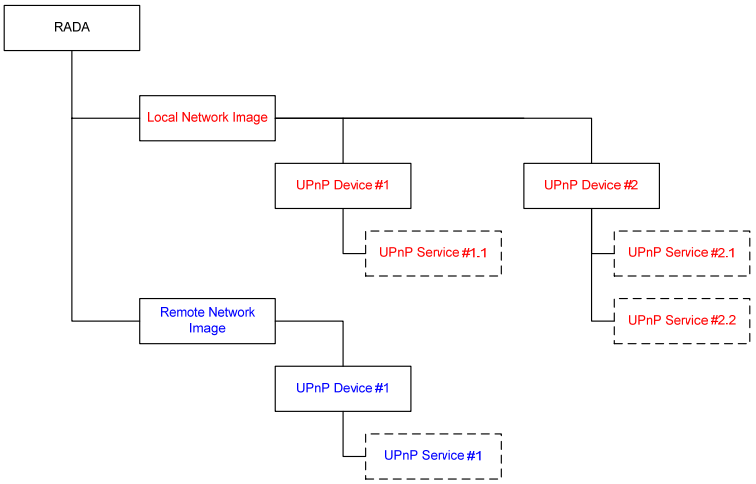


Figure 4-5: Discovery Information Aggregation

RADAListener, a support function of the RADA, is aggregating the devices and services available in the local network by constantly monitoring the SSDP announcements sent in the local network. The RADA Listener detects when devices are joining or leaving the network and notifies RADA about the changes, enabling the RADA to have an up-to-date image of the local UPnP network.

The RADA finds information about remote UPnP devices and services by synchronizing with remote RADA.

An RADA should keep the local and remote information separately. It might be possible that, in situations where multiple remote devices are connected to the same home network, the RADA keeps several branches of remote devices and services. The RADA should keep track of the identity of the remote entity for each remote branch.

4.4.2.2 Discovery Synchronization

Remote Access Discovery Agents expose a UPnP Service interface to facilitate in-band synchronization with other discovery agents. Each Remote Access Discovery Agent will register itself with the other Discovery Agents in the remote connection by providing information about itself such that it can be notified of changes in the device aggregation tree.

The Location URL for the UPnP Service exposed by the Discovery Agent is fixed and will always be bound to TCP port 1900 on the established network link between the two networks, negating any need to “discover” the other RADA. A Discovery Agent simply needs to download the description document from this URL in order to determine the Control and Event URLs.

Whenever a UPnP Device is added or removed from the aggregation tree, the Discovery Agent will notify other Discovery Agents by invoking the appropriate action on the UPnP Service exposed by the remote Discovery Agent.

Figure 4-6: Discovery Synchronization

4.4.2.3 Discovery Replication for Remote End

During the SSDP synchronization process, the information about the UPnP Devices and services that is maintained in local branch of one Discovery Agent is transferred to the remote branch of the corresponding remote Discovery Agent. Before transferring the local branch information, the Discovery Agent may apply some filters defined by the user in order to restrict the visibility of some of the local devices from remote entities.

The discovery replication for remote end is facilitated RADA Relay, which is a RADA support function. Whenever a change occurs in a remote branch of the RADA, the RADA Relay gets notified by the change and and recreates the corresponding original SSDP announcements, which are then multicasted in the local network.

Another function that is provided by the RADA Relay is to respond to discovery queries on behalf of the remote devices listed in the remote ranch of the RADA.

In the case of a home to home scenario, the remote information of the aggregated information is never shared between the RADAs of local and remote RAS devices, only local information is shared. This is done to enforce a pair-wise only view between a local and remote RAS devices even in the case where a particular RAS is connected to multiple homes.

4.4.3 Remote Access Transport Agent

The Remote Access Transport Agent (RATA) is responsible with providing a secure communication channel that enables a remote UPnP device to interact with the UPnP devices located in the home network. A RATA can provide the secure channel through several underlaying technologies, e.g. IPsec or TLS tunnels.

The two parties involved in the Remote Access agree on a common Remote Access transport mechanism with matching capabilities before a connection could be established. This is done by the management console. The Remote Access Architecture provides a configuration interface for the configuration of RATAs in the form of RATAConfig service.

4.4.4 Connection Establishment Agent

The role of the the Connection Establishment Agent is to enable the local RAS to locate the remote RAS across a WAN. This makes possible to pair the RAS devices not residing in the same UPnP network. The negotiation of the RAT profiles to be used for the RA connection is also performed by the agents in the respective devices. After the remote RAS has been located and the RAT profiles have been negotiated, the RATA and RADA take over and function as will be described in section 5.

Connection establishment of RA devices over the WAN enables the exchange of security parameters and credentials needed to successfully establish a remote access connection. This is valid for both RAC-to-RAS as well as RAS-to-RAS cases. The connection establishment interaction is a handshake in which the initiating RA device (RAS or RAC) makes an ‘offer’ which includes the UPnP RA security profiles supported by that device. These profiles are described in the Appendix section of the *RATAConfig* service specification. The profiles include: IPSec based on certificates profile, IPSec based on shared key null policy profile, IPSec based on shared key advanced policy profile and OpenVPN profile. The receiving RA device (RAS or RAC) makes a selection of one of the profiles which will be used to establish the RA tunnel, this interaction is shown in Figure 4-7.

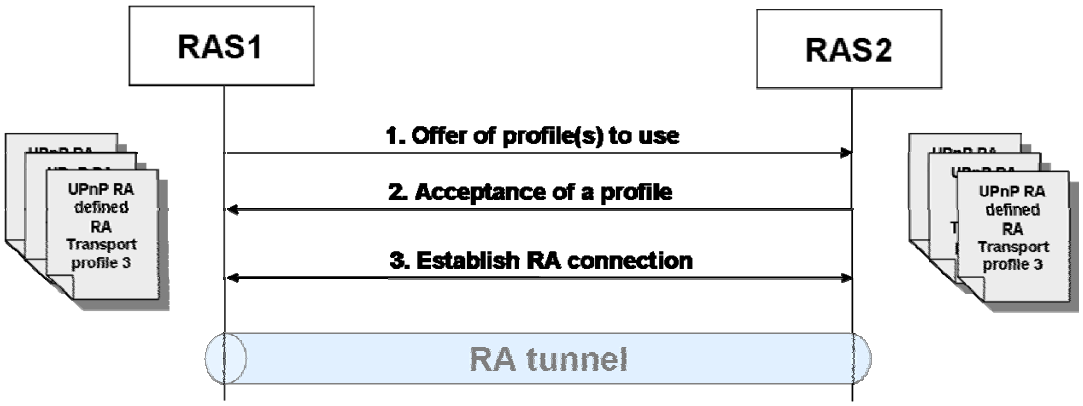


Figure 4-7: Handshake and Connection between RA Devices over the WAN.

The handshake interaction could also implement a model whereby the initiating RA device sends to the receiving RA device a set of parameters with which it would like to establish a RA connection. If the receiving side accepts these parameters the secure tunnel would then be set up based on these parameters.

For the RAS-to-RAS case, connection establishment can be originated independently from either side thus care should be taken to avoid the glare condition. A glare condition is defined as the condition in which two end-points simultaneously originate connection to each other, resuting in (1) neither side being able to establish the connection or (2) two independent connections are established. Glare case (1) is recoverable by building a re-try mechanism into the RATA. Glare case (2) can be recovered by requesting that one of the RATA (on either side) to terminate one of the established connection and should be handled by the Connection Establishment Agent.

4.5 Remote Access With QoS

This section of the document details different approaches on how the Quality of Service can be enabled between the two home networks using the two Remote Access Servers (RAS). The first approach describes a solution using the UPnP QoS where the QoS Control Point is in the RAS. The second approach describes a solution using the UPnP QoS where the QoS Manager is in the RAS, and the last approach describes how DLNA QoS can be used between the two RAS. These different solutions are dependent on the mechanism used in the operator network, indeed in some case QoS information will be lost.

4.5.1 Remote Access With UPnP QoS

The UPnP Remote Access architecture enables home to home remote access scenarios where devices in one home are able to connect to devices in another home spread across geographically in the wide area network. The establishment of QoS to connect devices within a single home is starigforward by use of currently defined UPnP QoS standardized interfaces that make use of different L2/L3 technologies in a heterogenous home network. In a typical QoS enabled home network there will be at least one QosManager service, optionally a QosPolicyHolder service, a number of QosDevice services (one for every device that is UPnP-QoS aware) and some QoS Control Points that manage the Quality of Service for their streams by using the services provided by a QosManager service. The basic idea for that approach is that a QoS request is sent by a QoS enabled control point to a QoSManager in the home that decomposes end-to-end QoS requirements among the devices in the path and subsequently instructs the QoSDevices in the path to reserve resources.

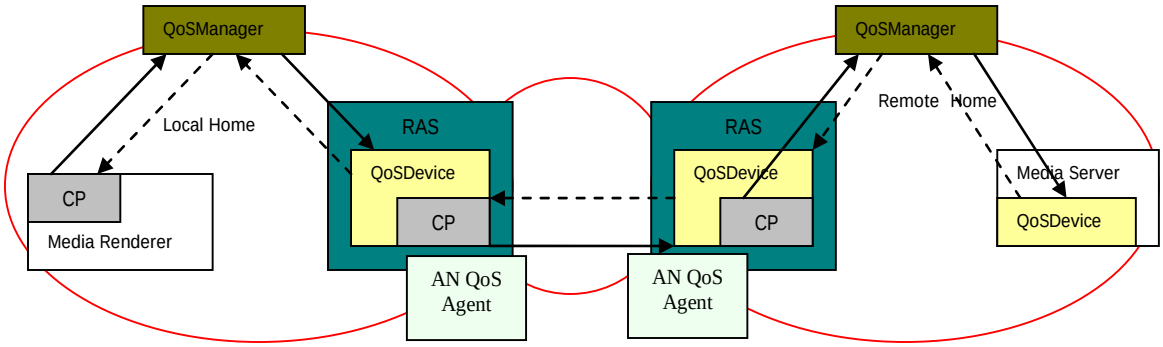


Figure 4-8: End-to-End QoS Setup Between Two Homes Conncected by Remote Access

The establishment of QoS to connect two devices that are located in separate homes connected through the remote access technologies is more complicated than the single home scenario and the end-to-end QoS setup requires establishment of QoS in the local home, QoS in the tunnel between the two homes, and finally the QoS in the remote home. To establish QoS in the remote home, the QoS parameters need to be passed on from the local Remote Access Server (RAS) to the remote Remote Access Server (RAS) which in turn will initiate the QoS request to a remote QoSManager. A support function called the Access Network QoS Agent takes care of signalling the QoS requirements to a QoS function in the access network and thereby providing QoS support for the tunnel between the two homes. This is done where a QoS function is available in the network.

To establish end-to-end QoS between two devices located in separate homes, the QoS enabled control point initiating the QoS setup in the local home will contact a QoS Manager in the local home to setup QoS in the local home. The initiating control point will specify the boundary of the request as the address of the local RAS and the destination as the remote device in the remote home with which the control point will setup QoS. The QoSDevice inside the RAS in the local home will get a request of QoS from the QoS Manager in the local home. The QoS and remote access enabled control point at the QoSDevice in the local RAS will invoke a SOAP action on the QoSDevice in the Remote RAS to initiate QoS setup in the remote home. The remote access and QoS enabled control point in the QoSDevice in the remote RAS will contact the QoS Manager of the Remote home to establish QoS in the remote home. This setup process will require QoSDevice in the RAS of the remote home to expose itself through the remote access mechanism to the RAS in the local home. No other QoS devices or QoS Managers in the Remote home need to be exposed. The QoS Manager in the remote home will setup QoS in the remote home as per the usual QoS mechanism.

The acknowledgement of the QoS setup will traverse back from the remote QoSManager to local control point in the QoSDevice of the local RAS and then from the QoSDevice in the local RAS to the QoSManager in the local home and then back to the initiating control point from the local QoSManager. Figure 4-8 shows the QoS setup process where all the requests are shown in solid lines and the responses are shown in dotted lines.

The Figure below shows the sequence diagram for home-to-home QoS setup.

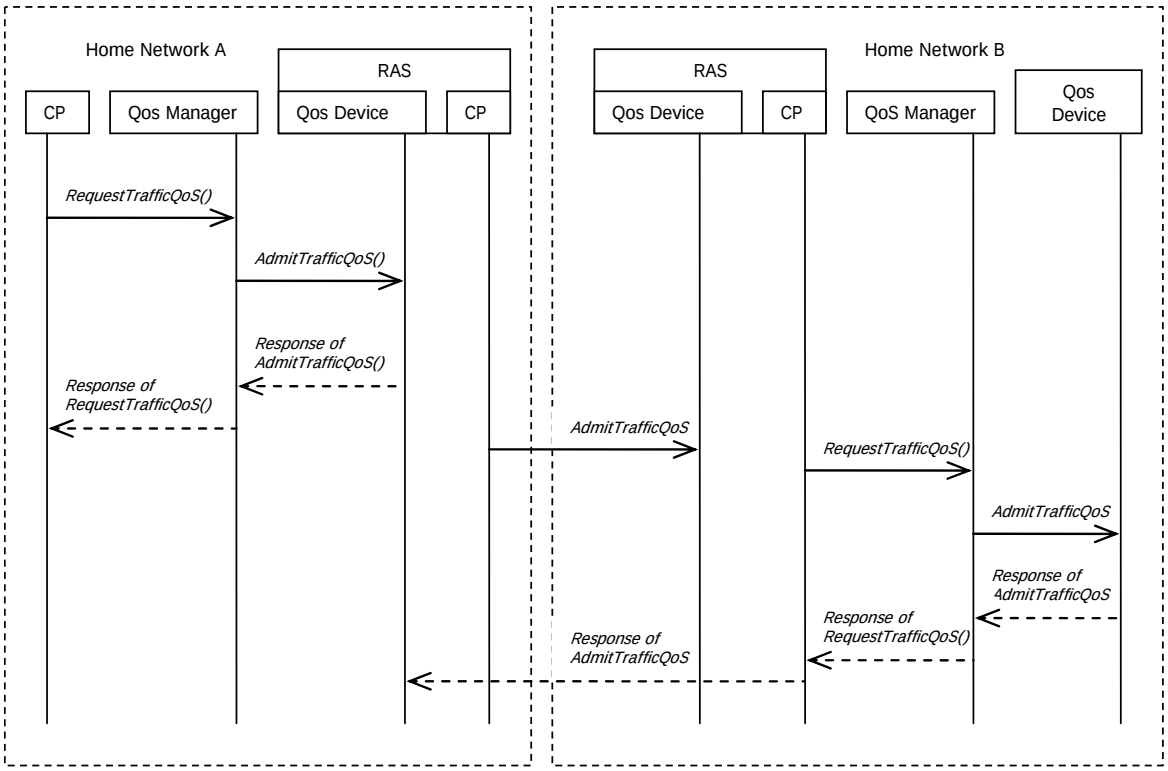


Figure 4-9: Home-to-Home QoS Setup Sequence Diagram

The CP in the RAS is a QoS Manager CP that requests *AdmitTrafficQoS()* to the QosDevice. The above sequence diagram is based on QoS3. However, the diagram equally applies to QoS2 if the *AdmitTrafficQoS()* action is replaced with *SetupTrafficQoS().*

For the case of a RAC assessing media from a remote RAS, the QoS setup is performed by means of a QoS Manager CP in the RAC device. The CP invokes an action requesting QoS in the remote QoS service of the RAS. The QoS request contains the QoS parameters needed for the media stream carried in the TSPEC. Upon receiving the QoS request, the remote RAS will pass the request to an internal QoS CP which will in turn trigger a standard QoS setup procedure as specified in the UPnP QoS specification. The result of the QoS setup action will be returned to the RAC and depending on the outcome, the media stream will be initiated.

Optionally the RAC may have a regular QoS control point for QoS setup when the device hosting the RAC joins a home LAN act as a media renderer device. This CP is denoted CP#1’ in the figure below.

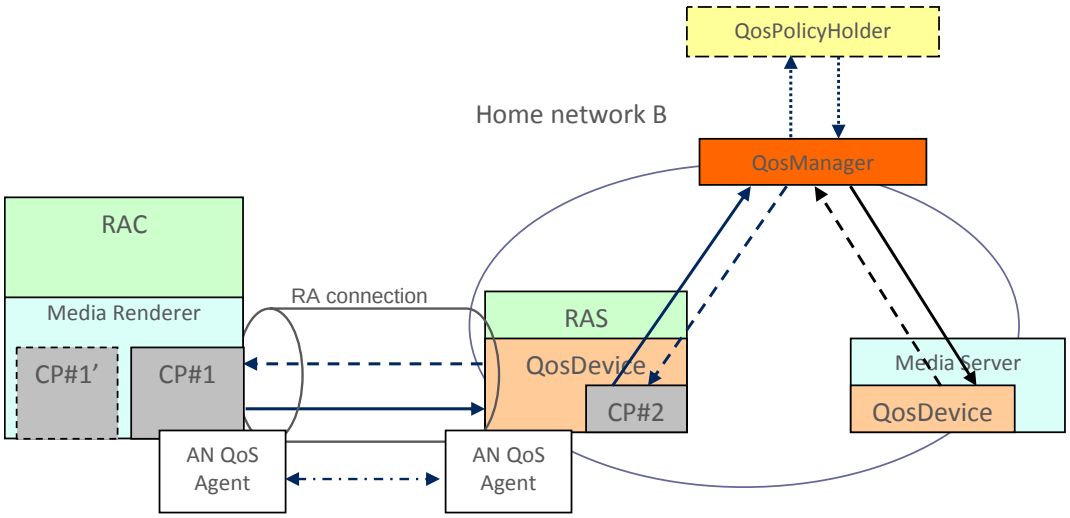


Figure 4-10: End-to-End QoS setup between RAC and RAS

Optionally the RAC may have a regular QoS control point for QoS setup when the device hosting the RAC joins a home LAN act as a media renderer device. This CP is denoted CP#1' in the figure above. The sequence diagram describing the interaction of the entities is given below.

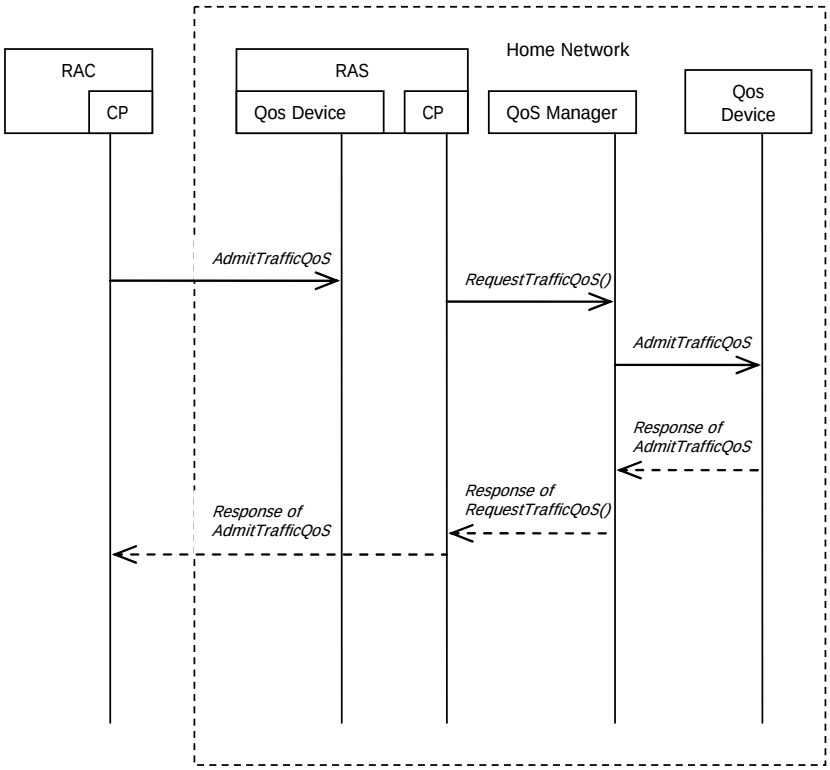


Figure 4-11: Client-to-Home QoS Setup Sequence Diagram

In parallel to the QoS setup procedure for the RAS-RAS case and the RAC-RAS case, the Access Network QoS Agent will try to perform a QoS setup in the access network between the remote access points. This option will be invoked if the access network has support for QoS setup. Appendix B.5 provides an example of the access network QoS setup procedure.

4.5.2 Remote Access With DLNA QoS

DLNA provides guidelines for prioritized QoS which is referreed to as DLNAQOS and intended to allow DLNA applications that wish to take advantage of User Priority (UP) which is referred to as DLNAQOS_UP. DLNA defines four different types of QOS priorities (DLNAQOS_3 (hightest, for RTCP messages generated by content receiver, DLNA Link Protection key exchange messages), DLNAQOS_2 (for Audio-only or A/V streaming transfer, UPnP AVTransport stream control, RTCP messages generated by Content Sources and RTSP messages), DLNAQOS_1 (default priority for any traffic defined by DLNA guidelines, interactive transfer, remote user interface messages), DLNAQOS_0 (for background transfer)). The DLNA traffic are tagged based on the types of traffic as mentioned previously and whch are then mapped to the priroty levels of the underlying technologies such as 802.1Q, WMM Access category, MoCA priority, and DSCP etc. The mapping of DLNAQOS pririties to the underlying technologies are defined in [DLNA Design Guidelines].

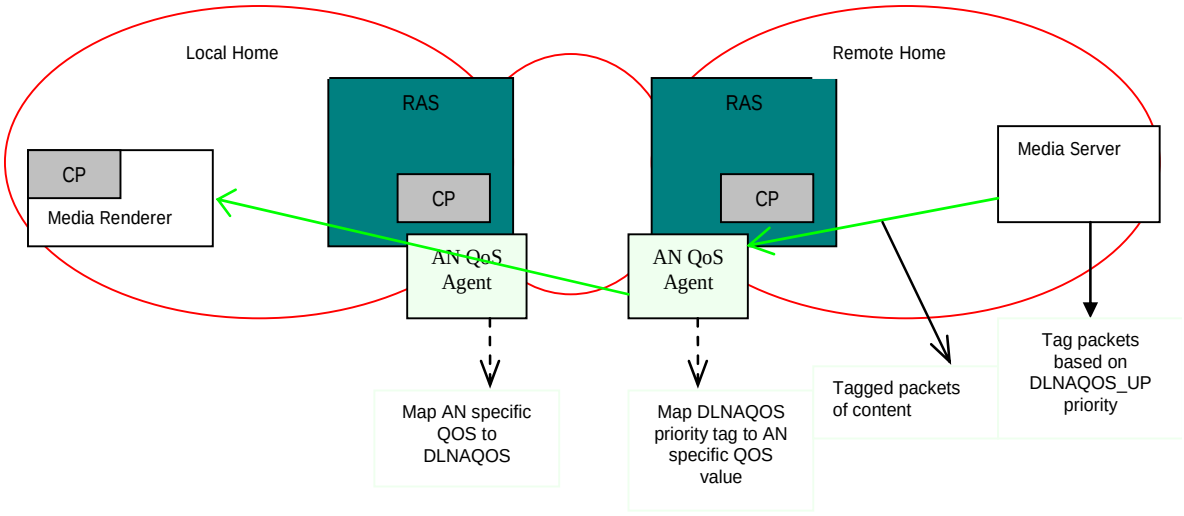


Figure 4-12: Remote Access With DLNA QoS Between Two Homes

The support of DLNAQOS in UPnP Remote Access can be provided by following the requirements outlined for DLNAQOS in [DLNA Design Guidelines]. An UPnP remote access application that wishes to use DLNAQOS, will label (tag) the packets with the DLNAQOS_UP priority value to indicate the User Priority that dictates how the packets are allowed to access the network resources. [DLNA Design Guidelines] defines the correlation of the DLNAQOS_UP priority to the DSCP (Differtiated Service Code Point) value and then to Wi-Fi WMM, MoCA, WMM, and 802.1Q etc. . DSCP is a QoS field defined by the Diffserv and can be found in layer 3 header of IP packets. When DLNAQOS is supported, the implementation apply the DSCP tag to the outgoing traffic in accordance with the DLNAQOS_UP value in addition to applying the tag value for the physical interface (i.e., WMM, 802.1Q etc.). There is no additional requirements needed in UPnP remote access to support DLNAQOS. The packets are transmitted with the appropriate tag value between the two homes through the remote access tunnel. The figure above shows this scenario where streams are flowing from the Media Server device to the Media Rendere device. The Media Server device tags the packets with appropriate DLNAQOS_UP priority value based on the

traffic types which then travels through the remote network. The AN QoS Agent at the remote RAS may support DLNAQOS in the access or operator network and gives prirotiy of the traffic based on the DSCP tag value that resides in the IP header. The packet then travels through the local RAS to the local home network and then to the local Media Renderer. An operator network that does not support Diffserv should not erase the DSCP code in the IP header but there is no gurantee for that. However, the AN QoS agent at the remote home can always try to tag the packets with the appropriate DLNAQOS_UP value for the traffic based on the traffic type wherever possible even when the DSCP code value in the IP header is lost or does not exist.

5 Interaction Model

5.1 UPnP RA Setup

This section describes the setup interactions for the RAC to RAS and RAS to RAS scenarios.

5.1.1 RAC-RAS configuration

5.1.1.1 Configure the RAS for Remote Access to Home Network

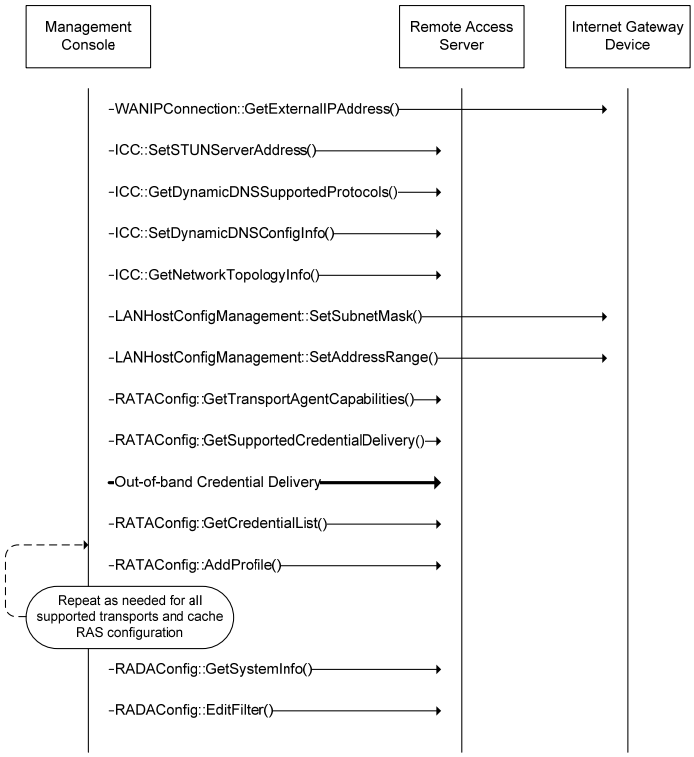


Figure 5-1: Remote Access Setup.

5.1.1.2 Configure the RAC for Remote Access to Home Network

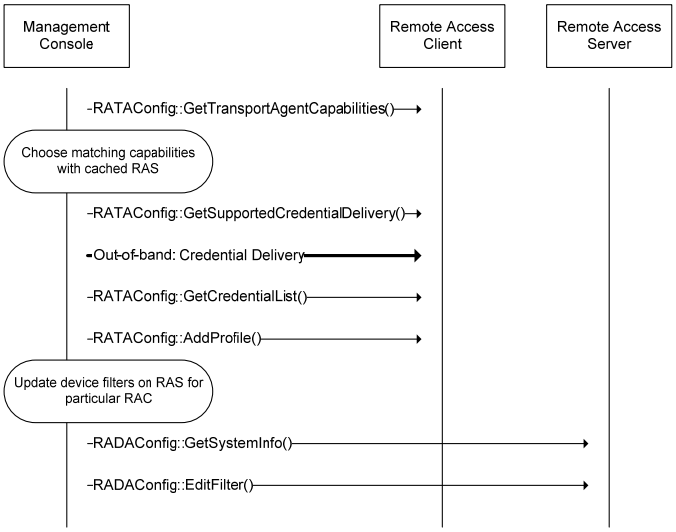


Figure 5-2: Configure the RAC for Remote Access to Home Network

5.1.2 RAS-RAS configuration

The RAS to RAS configuration is taken care of by the Connection Establishment Agent described in Section 4.4.4. the configuration procedure entails the negotiations of parameters used for the RA tunnel setup between the two devices. The negotiation is based on handshake model where the initiating RAS device proposes a set of parameters to the remote RAS. The remote RAS may either accept the paramters triggering a RA tunnel set up, or reject the connection establishment. The protocol supported by the connection Establishment Agent determines the sequence of messages exchanged between the RAS devices. An example of the interaction is made in the Appendix B.1.

5.2 Access Home Network Remotely from RAC over the Internet

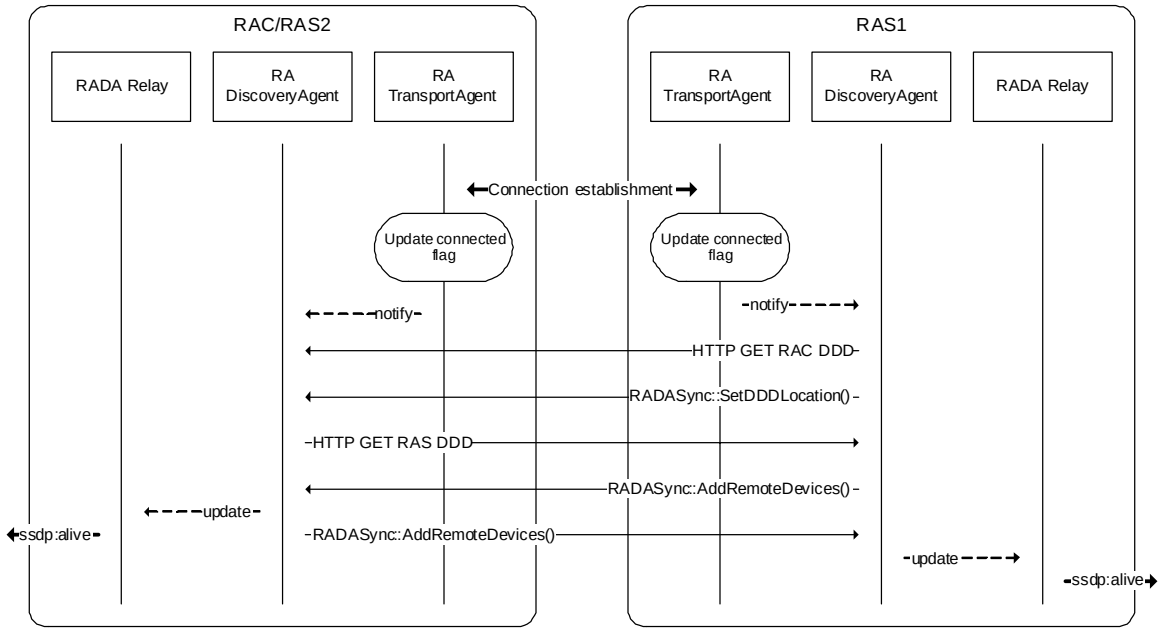


Figure 5-3: Access Home Network Remotely from RAC/RAS over the Internet

5.3 UPnP RA Connection Use

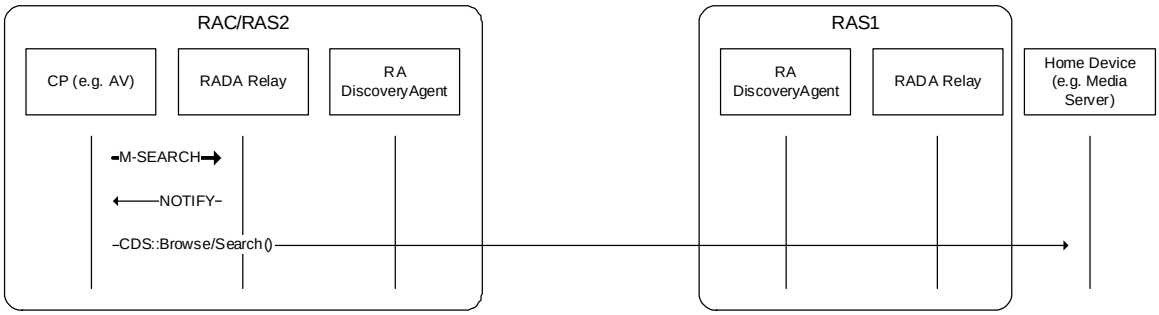


Figure 5-4: UPnP RA Connection Use

For the case of RAS to RAS, the AV CP will be external to the RAS.

5.4 RADA Synchronization Process

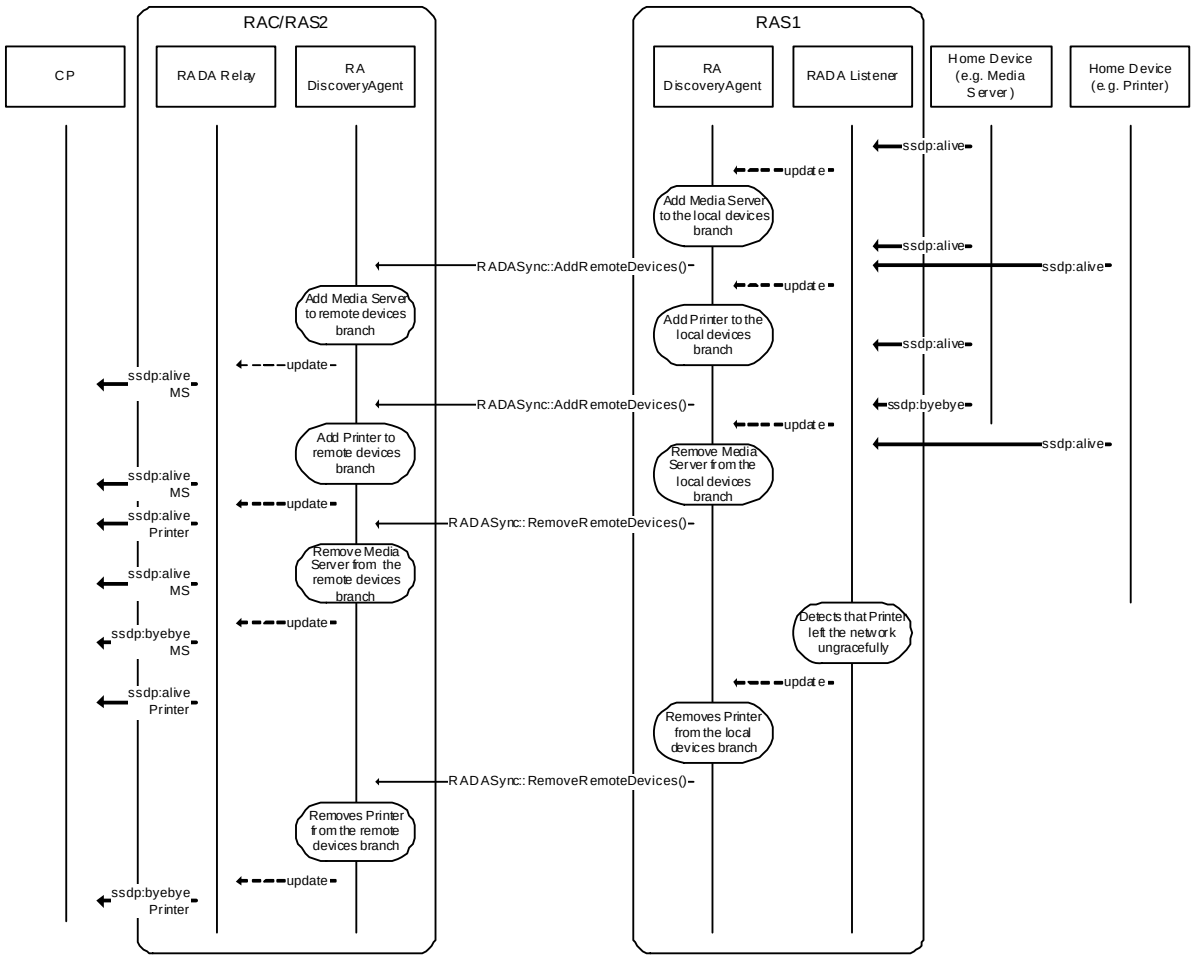


Figure 5-5: RADA Synchronization Process

5.5 RADA Heartbeat

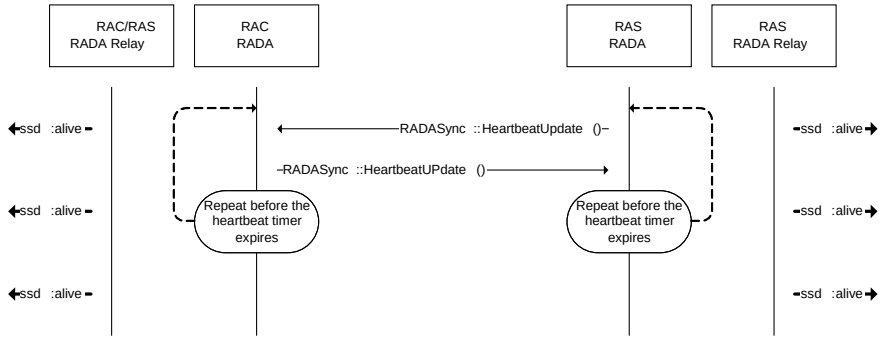


Figure 5-6: RADA Heartbeat

5.6 RADA Communication Time-out

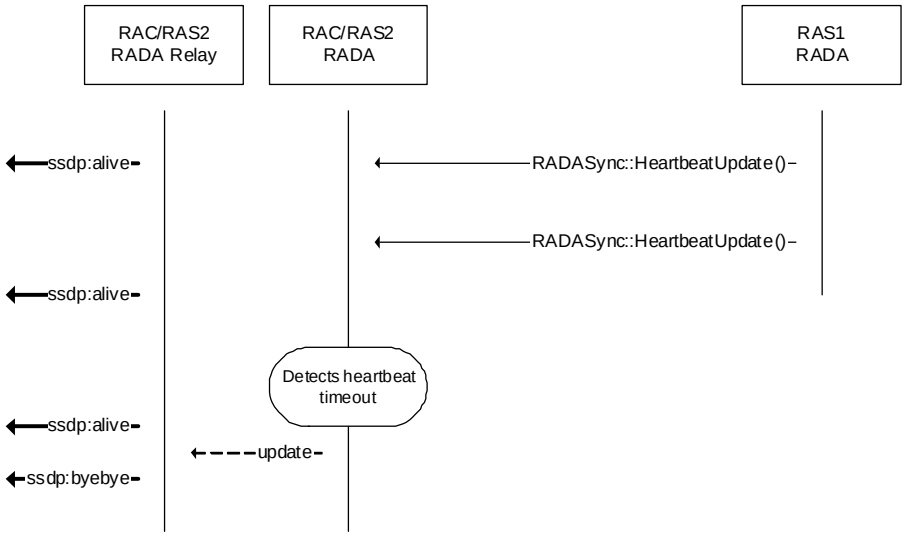


Figure 5-7: RADA Communication Time-out

5.7 RADA Administrative Shutdown

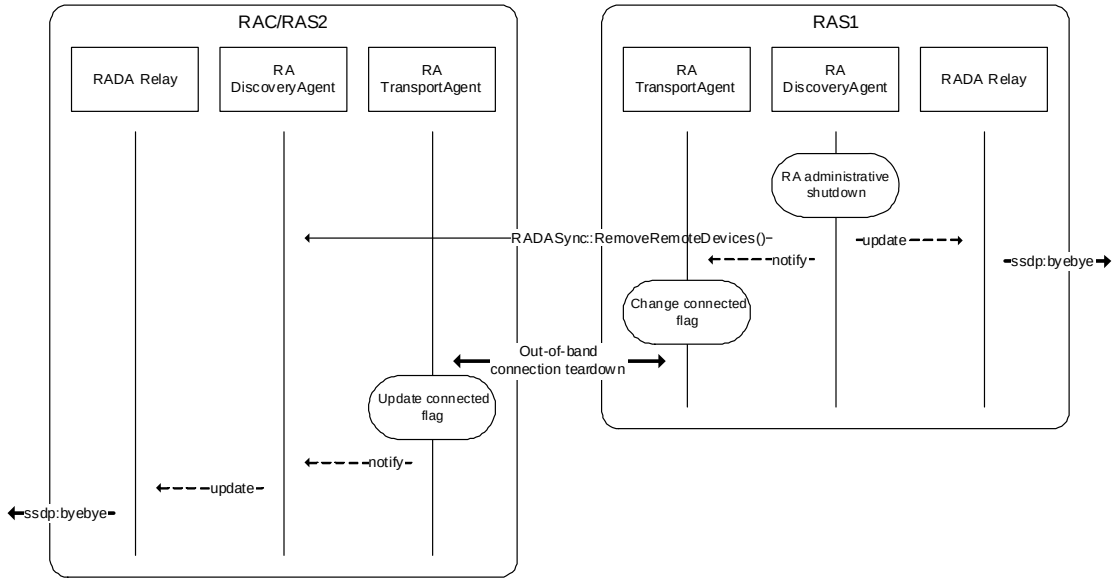


Figure 5-8: RADA Administrative Shutdown

Appendix A. Deployment Scenarios

A.1 Home Intended Deployment Scenarios

A.1.1 Remote Access Server in Residential Gateway

In this (defacto) deployment scenario, the Remote Access Server is located in the Residential Gateway together with the Internet Gateway Device. This setup is the simplest as it provides for the RAS direct access to the hardware WAN interface of the residential gateway.

A.1.2 Remote Access Server in a 3rd Party Device

In this (possible) deployment scenario, the Remote Access Server is located in a PC or in a standalone device other than the residential gateway. Comparing to the previous setup, the RAS has to take some actions to ensure that it is reachable from the internet and that the Remote Access transport can traverse the NAT deployed in the residential gateway. Additional considerations for this deployment scenario has been discussed in Section 3.2.1.2.

A.2 Internet deployement scenarios

A.2.1 Remote Access Server Hosted by a 3rd Party in the Internet

In this (potential) deployment scenario, the Remote Access Server is hosted in the ISP network or by a 3rd party provider in the internet. Typically, the RAS functionality is bundled with the other support services, e.g. DynDNS and STUN, and offered as a single service.

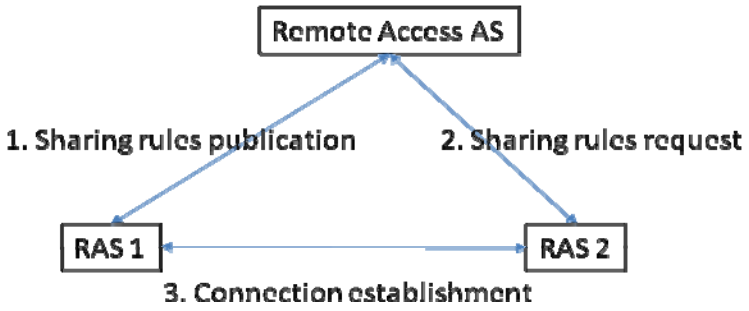
A.2.2 Remote Access Client Hosted by a 3rd Party in the Internet

Problem Definition: how to give access to a client wich do not have a UPnP device for the RAC.

This problem could be solved by deploying RAC into an AS hosted by a 3rd Party in the internet. When a remote user wants to connect to his home, he will connect to this AS, use some credential to be authenticated, and the the AS will play the role of a RAC to connect to the Home and display the information to the final user through a browser compliant GUI. In this (potential) solution the AS will have to be able to store association between a user (credential or id) and the remote home he wants to access to.

A.2.3 Identification for Session Establishment Between two RAS

Currently a user, who wants to access another home, should have the information on his RAS. Based on the TISPAN mechanism [TISPAN05(10)0017r2 TS185003_CR_On_Remote_Access_using UPnP] we can imagine another scenario where part of the sharing information are stored in the network.



1. When a User 1 wants to share his home with another User 2 , the RAS 1 will update sharing rules in the Remote Access Sharing Rules Server. The only needed information is User 1 share his home to user 2
2. On connection, RAS 2 retrieves information from the Remote Access AS to know if someone shares is home for User 2.
3. Based on the Sharing Rules User 2 establishes a Remote Access Session with User 1

Remote Access AS could be a presence server in SIP/IMS application or a web server based on HTTP mechanism

Appendix B. Best Practices

B.1 Connection Establishment

This section of the document details different approaches where the connection can be established between the two Remote Access Servers (RAS). The first approach uses SIP to establish the connection between the RAS using the IMS network [ETSI ES 282 001 (2009)] , the second approach uses the Session Initiation Protocol (SIP) by passing RATA parameters between two RAS, and the third approach describes solution where connection can be established over the top.

The deployment model should take care of these different approaches; a Remote Access Product could be only compliant with one, two or three connection establishment solutions. In addition to these scenarios, the solution will take care of the different profiles available between the different homes – Ipsec VPN, OpenVPN or without VPN.

To be able to manage a kind of interoperability between different solutions and service provider's roadmap (IMS and / or IPv6 deployment), a Remote Access Product should be compliant with the different connection establishment solutions.

B.1.1 Connection establishment using profile negotiation using IMS (IP Multimedia Subsystem)

The connection establishment model will be exemplified for the case of a managed network where two RAS devices establish a remote access connection via the IMS network [ETSI ES 282 001 (2009)] . With the aid of the session initiation protocol (SIP), two UPnP RAS devices are able to exchange information needed to enable a remote pairing procedure. As soon as the VPN tunnel is established, RADA synchronization proceeds as described in section 5.3.

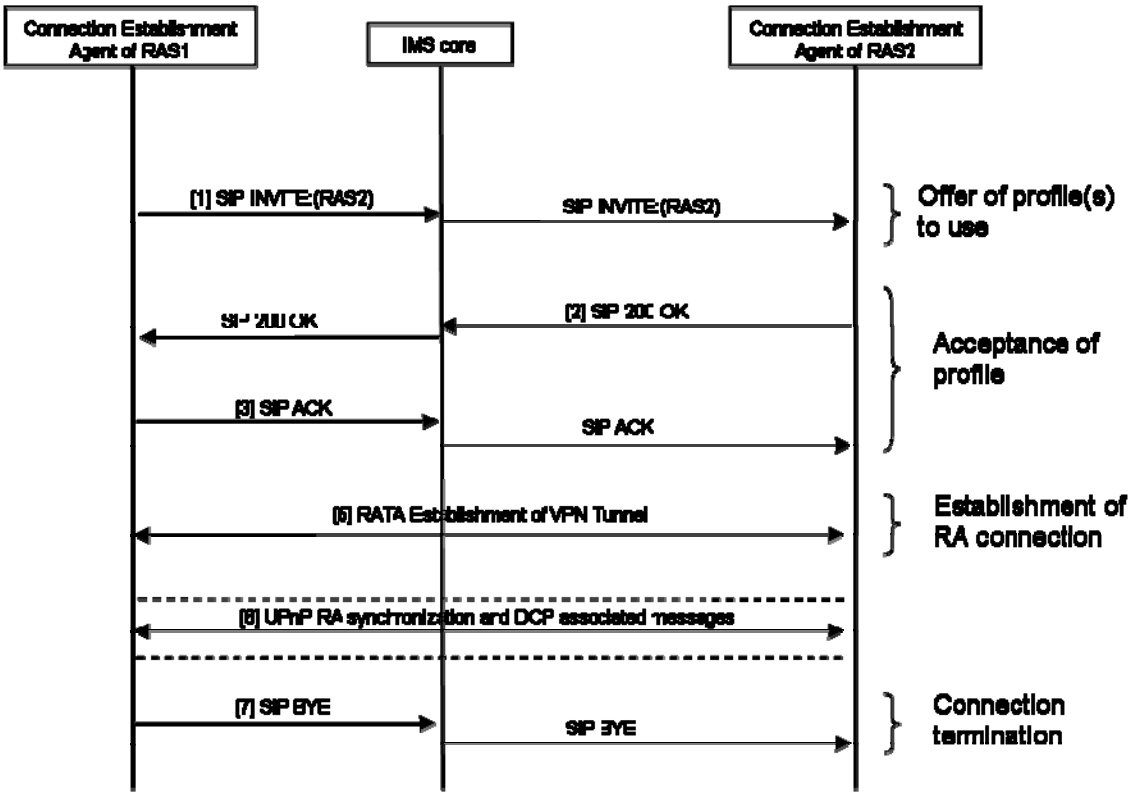


Figure B-5-9: Home to Home Remote Access establishment

Precondition: The Connection Establishment Agents of the RAS1 and RAS2 have registered themselves to the SIP registrar. Also each RAS has a set of RATA cofiguration profiles (these are specified in the appendix of the RATAConfig document). The configuration profiles contain a list of parameters attributed to either OpenVPN or IPSec.

1. The TransportAgentCapabilites of the RAS1 are extracted by the Connection Establishment Agent. It the initiates a SIP INVITE towards the RAS2. The request is sent through the SIP registrar.

   In order to initiate a Remote Access connection with RAS2 the session description protocol (SDP) in the SIP body of the INVITE message is used. The SDP contains a set of different parameters (see [TISPANCust]). The following are of special interest:

   The “c” – connection, contains the IP address of the RAS1.
   The “b” – bandwidth, contains the requested bandwidth, e.g. 384 kbps. (the actual requested bandwidth depends on the application.)
   The “a” – attribute, contains a reference to RATA cofiguration profile (OpenVPN and or the IPSec variants). For the case of IPSec, the “a” key attribute contains the corresponding IKEv2 keying material.

   Example of the SDP (RFC 2327) of the above:

   …
   c=IN IP4 172.21.0.1
   a=ra-client-private: ip=192.168.2.1 netmask=255.255.255.0
   b=AS:384
   a= ra-profile: 'null policy' key= LX1mY6iKA1n
   a= ra-profile: 'advanced policy' key= LX1mY6iKA1n

In access networks which support QoS e.g. mobile operator networks, the “b” attribute can carry the bandwith needed to be reserved for the media. If the QoS is needed in the access network after the RA connection has already been established, a RE-INVITE message is sent with the SDP carrying the “b” attribute which holds the new bandwidth value.

2. RAS2 validates if the request for RA connection from RAS1 shall be granted or rejected, this is done using the SIP identity (SIP URI). The assumption is the Connection Establishment Agent of RAS2 has a list, called SIP ACL, of SIP identities which are allowed to perform a RA request. If the SIP ID of RAS1 is not listed in the SIP ACL of RAS1, it shall respond with a SIP Response containing response code 403 (e.g. User Not Authorised).

   Regarding the offered setup profiles, if RAS2 does not support any of the RATA configuration profiles in the SDP offer of RAS1, RAS2 shall send a response with SIP Response containing response code 488 (Not Acceptable Here).

   The Connection Establishment Agent is able to perform IP address collision detection: if RAS2 detects that at least one network interface provided by the client affects its own network address range it will dismiss the INVITE received. Two networks collide when minimum one IP-address belongs to both networks. The Response given shall be a SIP Response containing response code 488 (Not Acceptable Here) together with a Warning header field value explaining why the offer was rejected.

   The Connection Establishment Agent allocates IP-addresses and ports and prepares for the remote access procedures by returning SIP 200 OK.

   In the response back the following is located in the SDP:
   The “c” – connection, contains the IP address of the RAS2
   The “a” – the RA profile for the supported transport setup described in the Appendix of the *RATAConfig* service specification.
   The “a=ra-virtualip” – The LAN IP address for the remote client.

   Examples in the SDP of the above specified are:

   …
   c=IN IP4 172.23.0.1
   b=AS:384
   a=ra-profile: 'advanced policy'
   a=ra-virtualip:IN IP4 192.168.1.49

   Connection Establishment Agent invokes the *RATAConfig::AddProfile()* action storing the selected RATA configuration profile and the accompanying key. Also the device filters of RAS2 for RAS1 are created using *RATAConfig::EditFilter()*.

3. SIP ACK is sent to acknowledge the SIP RA session setup. At this point RAS1 may detect a possible address collision. An address collision occurs when the offered IP is already in use for other purposes on the RAS2 side, therefore making routing impossible to perform. If the client cannot handle this an error message should be sent to the user and the tunnel setup should not proceed. Immediately after the SIP ACK is sent, a SIP BYE message should be sent to terminate the session.

   If there was not collision RAS1 creates a profile using *RATAConfig::AddProfile()* and creates a filter list for RAS2 using *RATAConfig::EditFilter()*.

4. The key management procedures commences generating IKEv2 keys in the case of IPSec.

5. The provisioning phase is complete and the RAS1 and RAS2 can establish a RA tunnel.

6. The RATA sends a notify to the RADA and operations as described in section ‘Access Home Network Remotely from RAC over the Internet’ continue from here.

7. After the RA session has been terminated based on the RADA administrative shutdown procedure, the Connection Establishment Agent close the SIP with SIP BYE message.

B.1.2 Connection establishment by dynamically exchanging parameters

This section of the document details an approach where the connection can be established using the Session Initiation Protocol (SIP) by passing RATA parameters between the two Remote Access Servers (RAS). The previous example uses SDP in the SIP INVITE message to convey connection profiles between two RASs. The approach in this section also uses the SIP INVITE message. However, rather than conveying connection profiles as part of SDP, this approach uses a separate MIME type body to carry connection establishment parameters in its entirety.

The diagram below shows how a SIP INVITE message with multipart body can be used to convey connection parameters. The multipart body of a SIP INVITE request message includes both an `application/sdp` MIME type and a MIME type (i.e. `application/rata-conncetion-para-request+xml`) for RATA connection parameters. The response of the SIP INVITE message will also contain accepted RATA connection parameters of the receiving end and in this way both sides can exchange their connection parameters. The content type of the SIP INVITE message is `multipart/mixed`, which indicates to the receiving end that there are multiple bodies in this INVITE message. The INVITE message can also contain an Accept header with `application/rata-conncetion-para-response+xml` content type to tell the receiving entity that the sender can accept “`application/rata-connection-para-response+xml`” MIME type in the response.

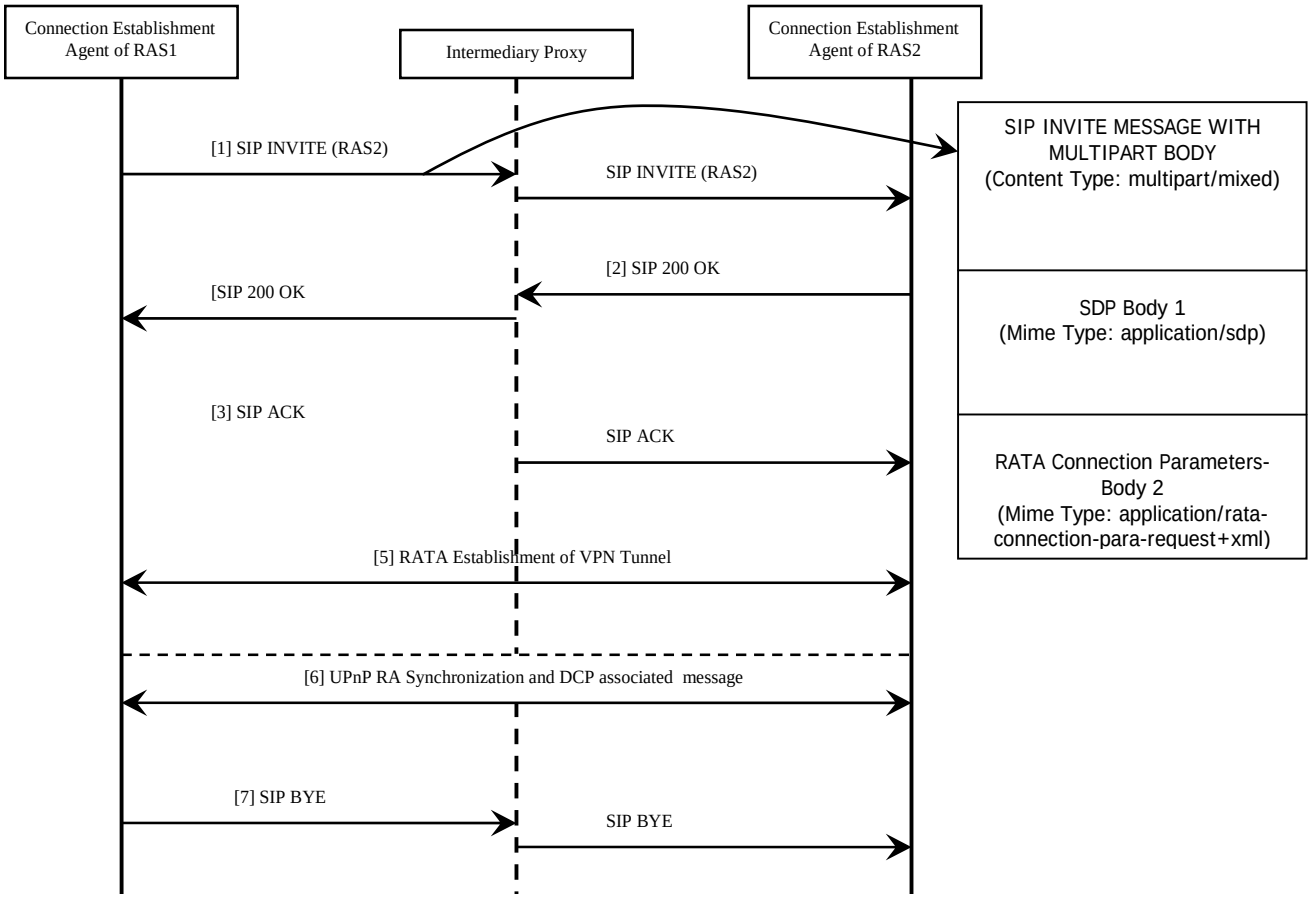


Figure B-5-10: Home to Home Remote Access establishment using parameter exchange

The example below shows a SIP INVITE message that includes RATA connection parameters information.

```
INVITE sip:addrss@example.com SIP/2.0
…other headers…
Accept: application/rata-connection-para-response+xml
Content-Type: multipart/mixed; boundary=unique-boundary
Content-Length:xx
--unique-boundary
      Content-Type: application/sdp
      Content-Length:xx
      v=0
      o=userA 28908442526 28908442526 IN IP4 example.com
      s=Session SDP
      c=IN IP4 pc33.example.com
      t=0 0
      m=audio 49172 RTP/AVP 0
      a=rtpmap:0 PCMU/8000

--unique-boundary
      Content-Type: application/rata-connection-para+xml
      Content-Length:xx

      <?xml version="1.0" encoding="UTF-8"?>
```

```
<tads
   xmlns="urn:schemas-upnp-org:ra:tads"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance"
    xmlns:opt="urn:schemas-upnp-org:ra:tacfg:openvpn"
   xsi:schemaLocation="urn:schemas-upnp-org:ra:tads
   http://www.upnp.org/schemas/ra/tads-v1.xsd
   urn:schemas-upnp-org:ra:tacfg:ipsec
   http://www.upnp.org/schemas/ra/tacfg-ipsec-v1.xsd">
   <transportAgentCapability
      transportAgentName="IPsec">
      <transportAgentOptions>
         <opt:ipsecOPT authenticationMethod="RSA
         Digital Signature"
            credentialEncoding="PKCS #7 wrapped X.509
         certificate"
            keyExchangeProtocol="IKEv2">
            <opt:encryptionAlgorithm>AES_CBC</opt:enc
         ryptionAlgorithm>
            <opt:authenticationAlgorithm>HMAC_SHA1_96
         </opt:authenticationAlgorithm>
            <opt:integrityAlgorithm>AES_XCBC_96</opt:
         integrityAlgorithm>
            <opt:pseudoRandomFunction>AES128_XCBC</op
         t:pseudoRandomFunction>
         </opt:ipsecOPT>
      </transportAgentOptions>
      <!-- Other transport agent options (if any) go
here. -->
   </transportAgentCapability>
   <transportAgentCapability
      transportAgentName="OpenVPN">
   </transportAgentCapability>
   <!-- Other transport agent capabilities (if any) go
here. -->
</tads>
--unique-boundary--
```

The response of the SIP INVITE message from RAS 2 will include only the accepted transport agent parameters in the SIP 200 OK message. For example, even though the SIP INVITE message includes both IPSec and OpenVPN options, the RAS2 may elect only IPSec to connect with RAS1. In that case, the 200 OK response message will have a `multipart/mixed` body that includes both `application/sdp` and `application/rata-connection-para+xml`. The following example illustrates the scenario:

```
SIP/2.0 200 OK
…other SIP headers…
Accept: application/rata-connection-para+xml
Content-Type: multipart/mixed; boundary=unique-boundary
Content-Length:xx

--unique-boundary
      Content-Type: application/sdp
      Content-Length:xx

      …accepted SDP parameters…
```

```
--unique-boundary
      Content-Type: application/rata-connection-para-response+xml
      Content-Length:xx

            <?xml version="1.0" encoding="UTF-8"?>
            <tads
               xmlns="urn:schemas-upnp-org:ra:tads"
               xmlns:cfg="urn:schemas-upnp-org:ra:tacfg:ipsec"
               xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
               xsi:schemaLocation="urn:schemas-upnp-org:ra:tads
               http://www.upnp.org/schemas/ra/tads-v1.xsd
               urn:schemas-upnp-org:ra:tacfg:ipsec
               http://www.upnp.org/schemas/ra/tacfg-ipsec-v1.xsd">
               <profileConfig dataStructureType="client">
                  <profileInfo id="12" transportAgentName="IPsec">
                     IPsec configuration
                  </profileInfo>
                  <profileData>
                     <cfg:ipsecCFG configurationType="client">
                        <cfg:policy>
                           <cfg:perfectForwardSecrecy>
                                 true
                           </cfg:perfectForwardSecrecy>

               <cfg:replayWindowLength>10</cfg:replayWindowLength>

               <cfg:remoteIdentity>alice@home.com</cfg:remoteIdentity
            >
                           <cfg:proposal protocol="ESP">
                                 <cfg:encryptionAlgorithm
            keyLength="256">
                                       AES_CBC
                                 </cfg:encryptionAlgorithm>
                                 <cfg:lifetime>

               <cfg:seconds>28800</cfg:seconds>
                                       <cfg:kBytes>5000</cfg:kBytes>
                                 </cfg:lifetime>
                           </cfg:proposal>
                        </cfg:policy>
                        <cfg:ike version="IKEv2">

               <cfg:remoteAddress>129.178.89.81</cfg:remoteAddress>

               <cfg:sendNotification>true</cfg:sendNotification>
                           <cfg:idType>ID_DER_ASN1_DN</cfg:idType>

               <cfg:useIPsecExpire>true</cfg:useIPsecExpire>

               <cfg:useReplayDetection>true</cfg:useReplayDetection>

               <cfg:useInternalAddress>true</cfg:useInternalAddress>
                           <cfg:dpdHeartbeat>600</cfg:dpdHeartbeat>
                           <cfg:natKeepalive>100</cfg:natKeepalive>

               <cfg:rekeyingThreshold>90</cfg:rekeyingThreshold>
                           <cfg:proposal protocol="IKE">
                                 <cfg:encryptionAlgorithm
            keyLength="256">
                                       AES_CBC
```

```
                        </cfg:encryptionAlgorithm>
                        <cfg:integrityAlgorithm>
                              AES_XCBC_96
                        </cfg:integrityAlgorithm>
                        <cfg:pseudoRandomFunction>
                              AES128_XCBC
                        </cfg:pseudoRandomFunction>

   <cfg:groupDescription>MODP_1536</cfg:groupDescription>
                        <cfg:groupType>MODP</cfg:groupType>
                        <cfg:lifetime>

   <cfg:seconds>28800</cfg:seconds>
                              <cfg:kBytes>5000</cfg:kBytes>
                        </cfg:lifetime>
                  </cfg:proposal>
                  <cfg:authenticationMethod>
                        RSA Digital Signature
                  </cfg:authenticationMethod>
                  <cfg:credentialID>100</cfg:credentialID>
               </cfg:ike>
            </cfg:ipsecCFG>
         </profileData>
   </profileConfig>
</tads>
--unique-boundary--
```

B.2 IP Address Collision

B.2.1 Problems

In the RAS/HG collocated RAS-to-RAS use case, two home networks are connected via a RATA connection and a routing problem arises whenever the two home networks that are being connected have the same LAN sub-net. This is because all clients in the local network would see that all remote servers that are advertised by RADASync as belonging to the local network. The same holds true for all clients in the remote network.

To resolve the address collision problem, this appendix proposes a best practice to be adopted by the HG (with a co-located RAS).

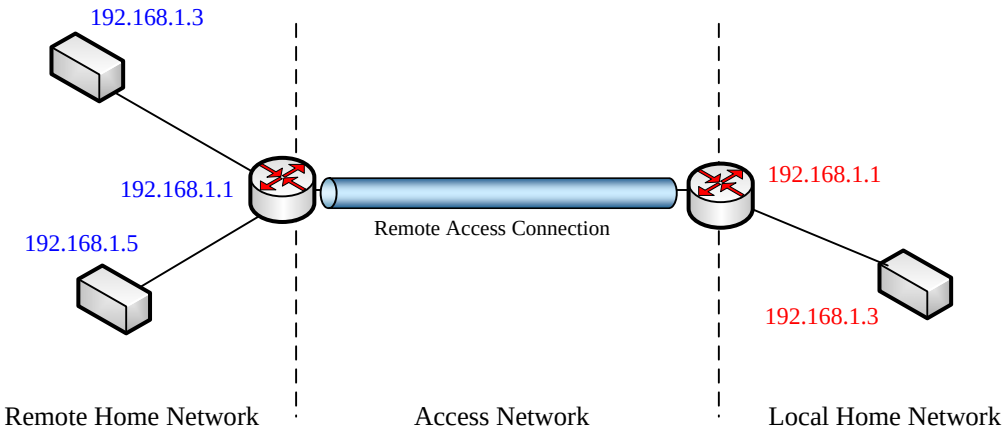


Figure B-1: Address Space Collision Problem

B.2.2 Practice 1: Subnet Randomization

The predominance of private address usage in the consumer home, along with the tendency to embed IP Addresses within the UPnP control messages causes addressing conflicts for Remote Access through VPNs. The problem is especially exacerbated by the current HG practice of uing static manufacturer supplied IP Address for LAN clients.

One method that could resolve the address collision problem is to focus on collision avoidance. Instead of HG's always using a manufacturer default for the LAN address, each HG should use a randomizing function to choose a private network at first out-of-the-box boot. Alternatively, UPnP IGD can be used to perform a one time randomize.

The following analysis presents how well this collision avoidance technique performs for the typical RAS-to-RAS use case.

Assuming that a home requires at most 30 host/devices and therefore 5 address bits are required for each home, such a randomizing function would use the following RFC 1918 address spaces:

19 bits in the 10/8 prefix (520200 address), and
15 bits in the 172.16/12 prefix (32768 addresses), and
11 bits in the 192.168/16 prefix (2048 addresses.

This gives a total of 555,016 distinct private network addresses each of which can support 30 hosts.

The birthday paradox is a useful way of examining the likelihood of address collision among randomized private networks.

One way of looking at the birthday paradox is to ask what is the maximum number of people in a room that would result in the probability n(p) of any 2 of them sharing a birthday is less than a given probability p.

This is analogous to asking what is the maximum number of networks that would make the probability of collision within that collection smaller than a desired percentage. Using a network address space of 555,016 and the approximation of: n(p) = sqrt(2 * 555,016 * ln(1/(1-p))).

The results are:

probability of collision	number of remote addresses visited
.1	342 networks
.01	106 networks
.001	33 networks
.0001	11 networks
.00001	3 networks

For example, if 10,000,000 networks are randomized, and each RA device is used in a pool of 33 other remote networks, then expect a total of 10,000 collision reports worldwide.

If each RA device is used in 11 remote networks, then 1000 collisions worldwide would be expected from these 10,000,000 units.

Another way to look at the probability of a conflict is to examine the birthday paradox for the question in a room of n other people, what is the probability q(n) that anyone has the same birthday as you.

This is analogous to asking given a random network address, what is the probability that as a device visits several remote networks it will encounter a conflict with its home network.

In this case, for the randomized network the probability is given by: q(n) = 1 - (((555,016 - 1)/555,016))^n)

The results are:

number of visited networks	probability of collision
342	.0006
106	.0002
33	.00006
11	.00002
3	.000006

This tells us that if 10,000,000 people each visit 33 different remote networks, then we should expect 600 conflicts worldwide.

The reason that the probability is significantly lower using this calculation is that we are measuring the probability of a conflict for a specific network as it joins other remote networks, and not the probability of a conflict between any two networks within a group of networks as earlier performed. The second case better models the way remote devices would be used.

B.2.3 Practice 2: IPv6

It must be noted that this procedure does not eliminate the possibility of address space collisions but will lead to a situation where, in practice, it will be highly unlikely that the access network and home network will be sharing the same address space. The transition to IPv6 will eliminate the problem of address space collision.

B.2.4 Practice 3: Address Translation based on Application Layer Gateway

To resolve address collision problem, the IP addresses used in one home network that collide with the IP addresses being used in another home network will be translated to a different address space. This translation will be done by the ALG (Application Layer Gateway) incorporated into the RATA and activated only when a collision is detected. If the RADA detects a collision of the local address space then the RADA chooses an address space that is different from the one being used in the other networks that are connected through the remote access channel. The ALG does the address translation for the RADA to the newly selected address space after detection of address collision. It is recommended that after selecting a new address space, the RAS should check again whether there is an address collision. This interaction is depicted in Figure B-2. It illustrates the case that both of the Remote Access Servers implement ALG. It is also possible that the ALG can reside only in one side of the network.

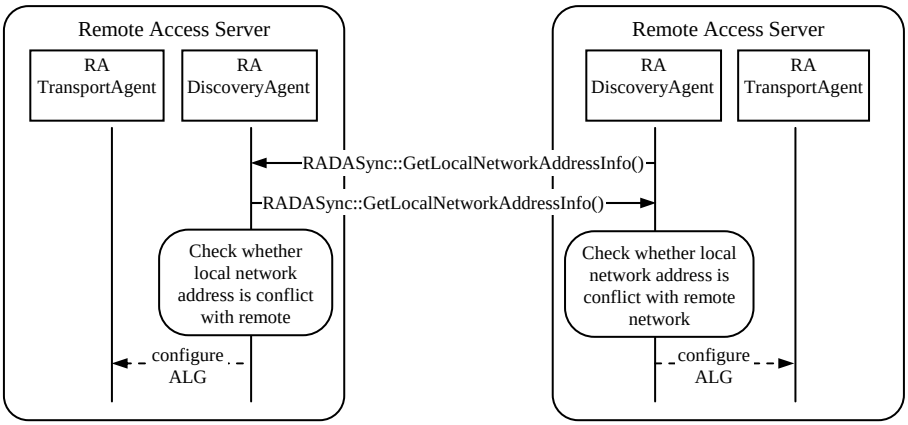


Figure B-2. ALG Configuration for address translation

Once the ALG is configured for address translation, it will translate the addresses for the traffic across remote access channel not only for the IP header but also for the body of the message. An implementation of the address translation of ALG can be vendor-dependent, but followings are the examples for a couple of viable approaches.

ALG can simply find the local IP addresses in the body of the message for the outgoing traffic, and replaces them with new the IP addresses that are in the address space selected by the RADA. Likewise, the ALG will convert them back to the local IP addresses for the incoming traffic. The ALG should have intelligence such that it can update the value of the message body if the replacement of the IP address in the body will result in changes to some other part in the body of the message. In case of HTTP messages, the ALG should update the value of Content-Length header if the length of the message body is changed due to the change of the IP address. This approach can simplify the complexity of the ALG but it can sometimes lead to some changes in the message body that should not be changed (i.e., the title of the content that includes IP address intended for the local network should not be translated)

In the second appoach the ALG parses the entire messages and replace the IP addresses based on its context of the protocol carried over the messages. The ALG needs to have an ability to understand the context of the IP addresses appeared in the body of messages. This would reduce the possiblity of erroneous replacement of the IP address but this would require a complex ALG algorithm.

B.2.4.1 Example ALG Implementation

ALG address translation occurs during UPnP description, control and eventing steps. In the discovery step, actual ALG configuration for each UPnP home device can be prepared. Usually, packet-by-packet ALG implementation during routing process is quite difficult because UPnP description, control and eventing steps use HTTP/TCP protocol and an IP address information that should be translated can be separated into two or more consecutive TCP/IP packets. Also message length changes caused by ALG processing make it worse because it can make some trouble in TCP sliding window flow control between end devices. The more practical and easiest way to implement ALG is to establish two individual TCP connections which are the one between remote CP and local RAS and the other between local RAS and UPnP home device. Local RAS can do ALG processing such as buffering entire messages and address translation, and it can bridge the two TCP connections. Figure B-3 shows such configuration.

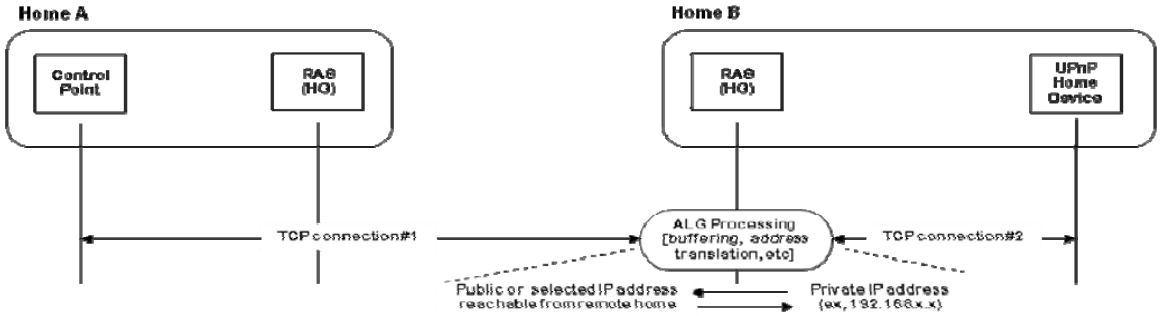


Figure B-3. A Practical TCP connections setup for ALG processing

In the discovery step, if the local RAS detects an UPnP home device, it can allocate a TCP socket at the HG that it is embedded in to prepare a TCP connection segment between remote CP and the local RAS. Then, It can rebuild LOCATION URL of the device using HG IP and the port number of the TCP socket. The modified LOCATION URL of the device can be transmitted to remote home. Figure B-4 shows such interactions.

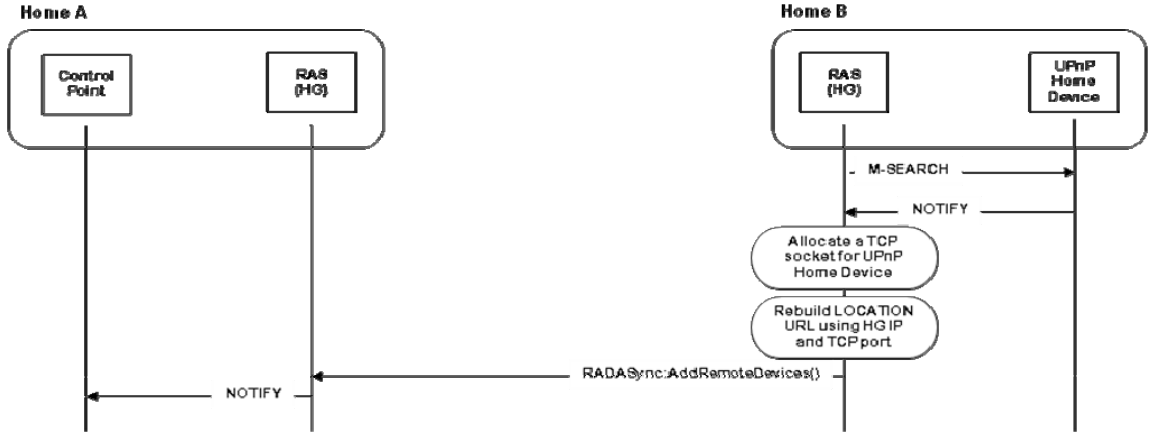


Figure B-4. A Practical ALG implementation: UPnP Discovery step

Using the received LOCATION URL of the home device, remote CP tries to establish a TCP connection. The TCP connction request from the remote CP arrives at the local RAS and the local RAS can find the target UPnP home device using the TCP socket port number. Then the local RAS establishes the second TCP connection between the local RAS and the home device. Figure B-5 shows such interactions.

Figure B-5. A Practical ALG implementation: TCP connection setup interaction.

In the description step, local RAS receives a HTTP GET message for the LOCATION URL of the home device and it can simply relay that request to the home device through the second TCP connection. If the local RAS receives the response from the home device, it can buffer the entire message and do the ALG processing which is the replacement of local address and UPnP port number of the home device with the HG IP and the TCP socket port number. Figure B-6 shows such interactions.

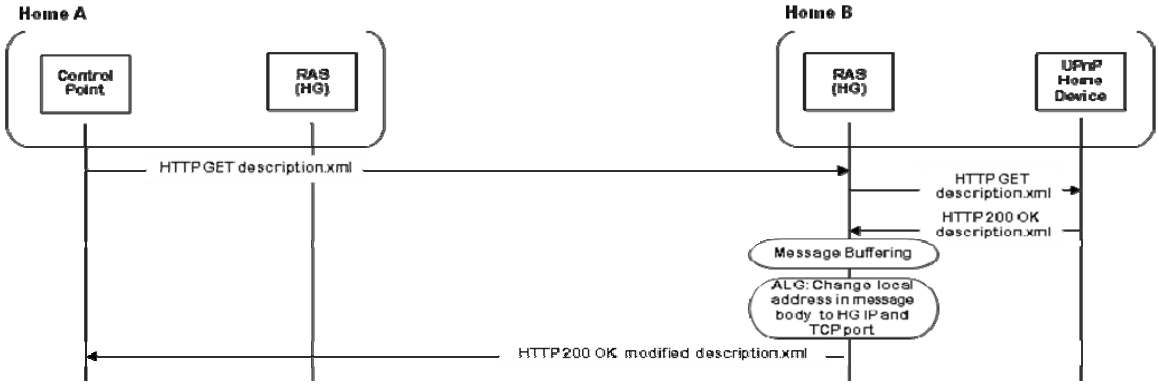


Figure B-6. A Practical ALG implementation: UPnP Description step

In the control step, the interaction is very similar to the description step except that the ALG processing is done with the SOAP response message. Figure B-7 shows such interactions.

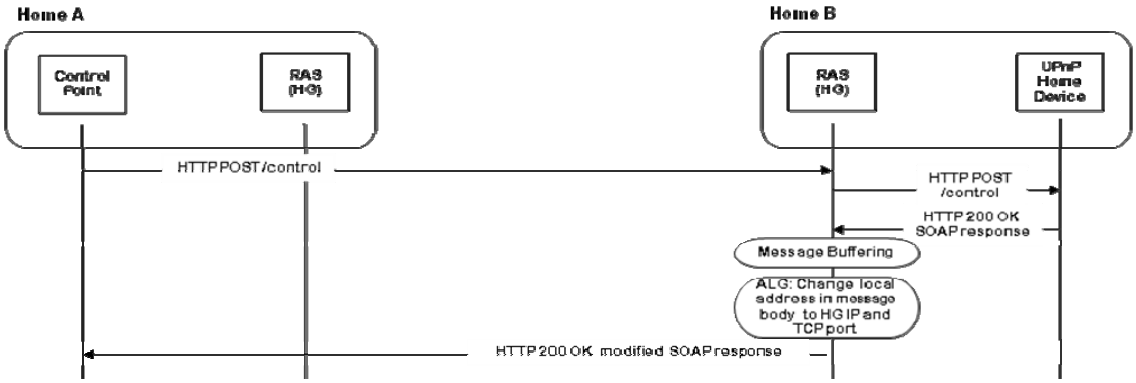


Figure B-7. A Practical ALG implementation: UPnP Control step

When we consider out-of-band transfer between remote UPnP devices just like as between UPnP Media Server and Media Renderer, there is no need of ALG processing because the contents are usually binary data. But if the Media Server provides AV contents through the same UPnP service port, the contents should be transmitted through the same TCP connections established for the ALG processing. But in this case, there is no need to buffer the entire contents. Only simple relay of the response packets are enough. Figure B-8 shows such case.

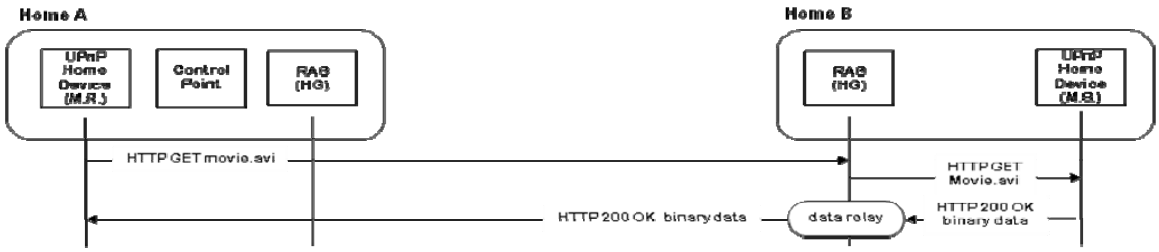


Figure B-8. A Practical ALG implementation: Out-of-band transfer through ALG

If the Media Server provides AV contents through different port, port forwarding is the best choice because it consumes less resources than ALG does. Normally, we can know whether port forwarding is possible or not in the UPnP control step during ALG processing. When we find an out-of-band content URL that uses different service port, we can configure port forwarding. And after the configuration, the out-of-band content URL can be modified using HG IP and the external port number of the port forwarding. Figure B-9 extends Figure B-7 to include port forwarding configuration process and Figure B-10 shows out-of-band transfer through port forwarding.

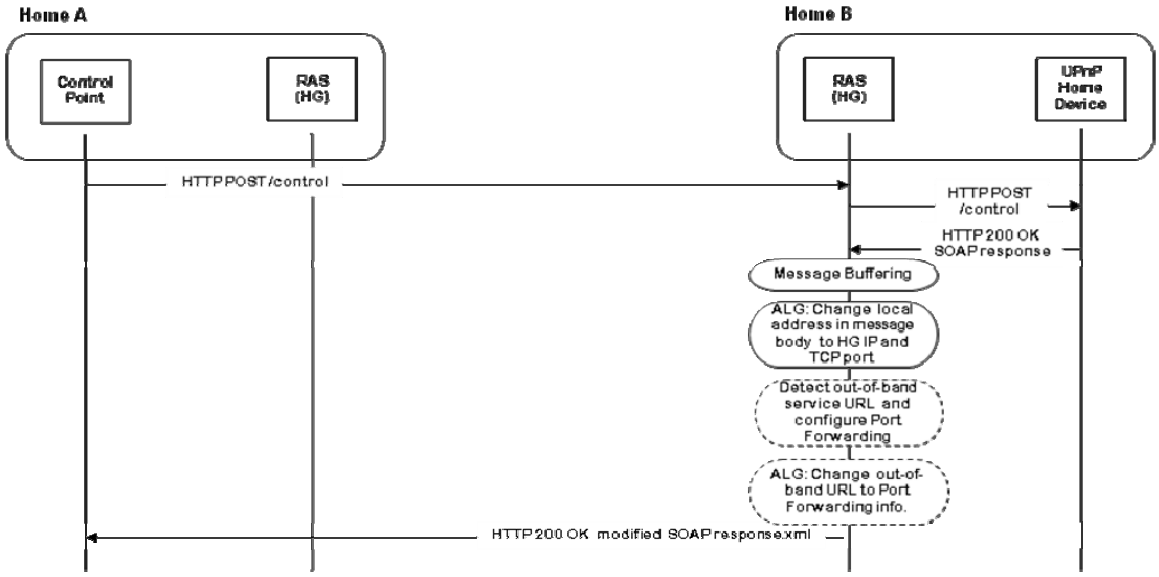


Figure B-9. A Practical ALG implementation: UPnP Control step with port forwarding setup

Figure B-10. A Practical ALG implementation: Out-of-band transfer through Port Forwarding

If we consider the UPnP AV 3-box model and the case that a local CP tries to send a content URL of a local Media Server to a remote Media Renderer, the address information in the content URL should be translated. If the local CP knows it has to translate the address, it can ask the local RAS about the target address information which has been constructed through the ALG or the port forwarding configuration and use that information when it calls AVTransport:SetAVTransportURI() action. If the local CP does not know that, the content URL in SetAVTransportURI() contains local address information and the remote Media Renderer can not access to this URL. In this case , the local RAS can do the address translation through packet filtering. When the remote RAS sends AddRemoteDevices() of the remote Media Renderer to the local RAS, it can configure a packet filter that captures packets going to the remote Media Renderer. The local RAS can translate the content URL of the captured packet. Figure B-11 shows an example interaction.

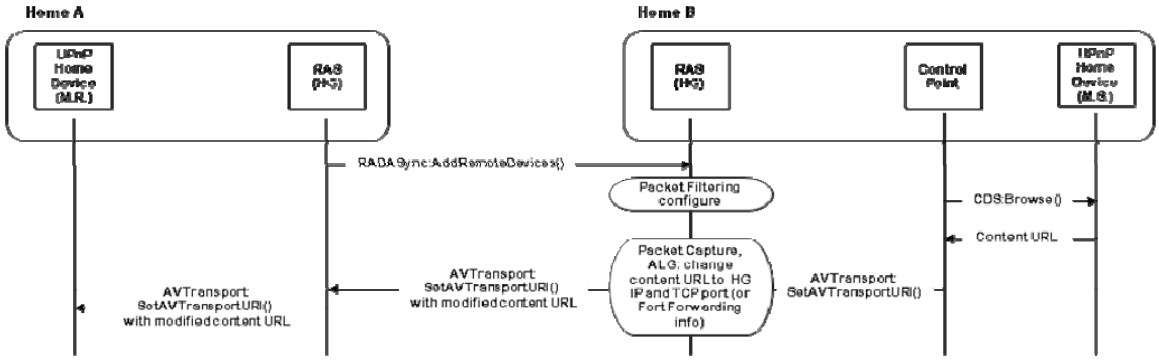


Figure B-11. A Practical ALG implementation: UPnP AV 3-box with remote Media Renderer

In the eventing step, when the local CP want to subscribe an event notification to a remote UPnP home device, it sends a SUBSCRIBE message with a local CALLBACK URL information. If the CALLBACK URL is not translated, the local CP can not receive event notification from the remote device. The local RAS can do the translation through packet filtering similarly to the above 3-box case. We can configure a packet filter when we receive AddRemoteDevices() of the remote device that captures packets going to the remote device. If the local RAS captures the SUBSCRIBE message and the port forwarding to the local CP is not configured yet, additional port forwarding configuration can be performed. Figure B-12 shows an example interaction.

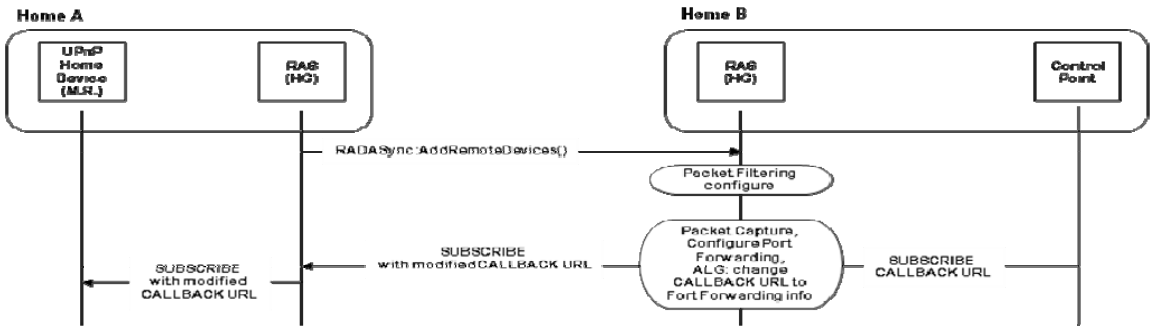


Figure B-12. A Practical ALG implementation: UPnP Eventing step

B.2.5 Practice 2: IPv6 over IPv4

It must be noted that this procedure does not eliminate the possibility of address space collisions but will lead to a situation where, in practice, it will be highly unlikely that the access network and home network will be sharing the same address space. The transition to IPv6 will eliminate the problem of address space collision.

B.3 NAT Traversal

As metioned in other parts of this document, Network Address Translation (NAT) has been deployed by some Service Providers to deal with the IPv4 addresse exhaustion issue, which is now predicted to occur sometime between 2011 and 2012. NAT are deployed in routers and help to reduce the IPv4 address usage by supporting multiple devices behind a single public IP address. It allocates private IP addresses [RFC 1918] to these devices and manages the dynamic translation/mapping between the internal and external IP addresses / ports. These dynamic address translations creates a problem for a remote access device that tries to establish a connection to a remote access server because the “external” IP addresses / ports to the remote access server may no longer be static. This section summarizes the approaches available to help solve this end-to-end connectivy problem for the different NAT scenarios.

There are a variety of NAT implementations available in the market today. Description of the NAT behaviors are outside the scope of this document but a detail description of NAT behavriors are available in [RFC 4787] . The following NAT behaviors are addressed in this document:

- Endpoint Independent Mapping (also referred to as Full Cone NAT)
- Address Dependent Mapping (also referred to as Restricted Cone)
- Address and Port Dependent Mapping (also referred to as Port Restricted NAT)

- (Unique) Address and Port Dependent Mapping (also referred to as Symmetic NAT)

The following techniques are considered:

- Dynamic Domain Name System (DNS) update [RFC 2136]
- Domain Name System Resource Record for Location of Services (DNS SRV) [RFC 2782]
- Session Traversal Utilities for NAT (STUN) [RFC 5389]
- Traversal using relays around NAT (TURN) [BEHAVE TURN]
- Connection Reversal mechanism for establishing a P2P connection [RFC 5128]
- Hole Punching mechanism for establishing a P2P connection [P2P Com]

The table below summarizes the use of the above techniques and combination of these techniques to help establish a connection between a client and a server in the presence of the various NAT permutations:

Client / Server	1. Routable IP Address	2. Full Cone	3. Restricted Cone	4. Port Restricted	5. Symmetric NAT
1. Routable IP Address	DDNS-SRV	Same as 1.1 (+)	Same as 1.1	Same as 1.1	Same as 1.1
2. Full Cone	STUN, DDNS-SRV	Same as 2.1	Same as 2.1	Same as 2.1	Same as 2.1
3. Restricted Cone	STUN/TURN, DDNS-SRV, Connection Reversal	Same as 3.1	STUN/TURN, DDNS-SRV, Hole Punching	Same as 3.3	Same as 3.3
4. Port Restricted	STUN/TURN, DDNS-SRV, Connection Reversal	Same as 4.1	Same as 4.1 + Hole Punching	STUN/TURN (with relay), DDNS-SRV (*)	Same as 4.4 (*)
5. Symmetric NAT	STUN/TURN, DDNS-SRV, Connection Reversal	Same as 5.1	Same as 5.1 + Hole Punching	STUN/TURN (with relay), DDNS-SRV (*)	Same as 5.4 (*)

(+) "Same as 1.1" means the solution is the same as the solution in row 1, column 1
(*) Data is relayed through the TURN Server

B.4 Access network QoS