

TECHNICAL REPORT

DSL Forum TR-111

Applying TR-069 to Remote Management of Home Networking Devices

December 2005

**Produced by:
DSLHome-Technical Working Group**

**Editors:
Jeff Bernstein, 2Wire
Tim Spets, Westell
Christele Bouchat, Alcatel**

**Working Group Co-Chairs:
Greg Bathrick, Texas Instruments
Heather Kirksey, Motive
Wayne Daniel, Siemens**

Abstract:

This specification extends the mechanism defined in TR-069 for remote management of customer premises equipment to allow a management system to more easily access and manage devices connected via LAN through an Internet gateway.

Notice:

The DSL Forum is a non-profit corporation organized to create guidelines for DSL network system development and deployment. This Technical Report has been approved by members of the Forum. This document is not binding on the DSL Forum, any of its members, or any developer or service provider involved in DSL. The document is subject to change, but only with approval of members of the Forum.

©2005 Digital Subscriber Line Forum. All Rights Reserved.

DSL Forum technical reports may be copied, downloaded, stored on a server or otherwise re-distributed in their entirety only.

Notwithstanding anything to the contrary, the DSL Forum makes no representation or warranty, expressed or implied, concerning this publication, its contents or the completeness, accuracy, or applicability of any information contained in this publication. No liability of any kind shall be assumed by the DSL Forum as a result of reliance upon any information contained in this publication. The DSL Forum does not assume any responsibility to update or correct any information in this publication.

The receipt or any use of this document or its contents does not in any way create by implication or otherwise any express or implied license or right to or under any patent, copyright, trademark or trade secret rights which are or may be associated with the ideas, techniques, concepts or expressions contained herein.

Contents

Part 1	Device-Gateway Association	5
1.1	Introduction	5
1.1.1	Terminology	5
1.1.2	Document Conventions	6
1.2	Procedures	6
1.2.1	Gateway Requirements	6
1.2.2	Device Requirements	7
1.2.3	ACS Requirements	7
1.2.4	Device-Gateway Association Flows	8
1.2.5	DHCP Vendor Options	9
1.2.6	InternetGatewayDevice Data-Model Extension	10
1.3	Security Considerations	12
1.4	Normative References	13
Part 2	Connection Request via NAT Gateway	14
2.1	Introduction	14
2.1.1	Terminology	14
2.1.2	Document Conventions	14
2.2	Procedures	14
2.2.1	CPE Requirements	15
2.2.2	ACS Requirements	20
2.2.3	Message Flows	23
2.2.4	Data-Model Extension	25
2.3	Security Considerations	28
2.4	Normative References	29

Document Overview

This document specifies two mechanisms that extend the CPE WAN Management Protocol defined in TR-069 to enhance the ability to remotely manage devices that are connected via a LAN through an Internet gateway. Examples of such devices include VoIP phones, media set-top boxes, and gaming systems.

These two mechanisms are specified in two separate parts of this document. The two parts are briefly summarized as follows:

Part 1:**Device-Gateway Association**

Allows an ACS managing a device to identify the associated gateway through which that device is connected.

Part 2:**Connection Request via NAT Gateway**

Allows an ACS to initiate a TR-069 Session with a device that is operating behind a NAT gateway.

The two parts of this document are completely independent from each other. That is, a device, gateway, or ACS that supports the mechanism defined in one part of this document need not support the mechanism defined in the other part.

As part of this document, extensions to the Device and InternetGatewayDevice data models are defined. By extension to the previous data model versions, this document defines the following data model versions:

Device:1.1

InternetGatewayDevice:1.2

Part 1

Device-Gateway Association

1.1 Introduction

The CPE WAN Management Protocol defined in TR-069 [1-2] may be used to remotely manage CPE Devices that are connected via a LAN through a Gateway. When an ACS manages both a Device and the Gateway through which the Device is connected, it can be useful for the ACS to be able to determine the identity of that particular Gateway. TR-069 as currently defined does not specify a means by which the ACS could make this determination.

Part 1 of this document defines an extension to TR-069 that allows an ACS to determine the identity of the Gateway through which a given Device is connected.

As an example of when this capability might be needed, an ACS establishing QoS for a particular service may need to provision both the Device as well as the Gateway through which that Device is connected. To do the latter, the ACS would need to determine the identity of that particular Gateway.

The specific scenario that the defined mechanism is intended to accommodate is where both the Gateway and Device are TR-069 managed, and both are managed by the same ACS (or by distinct ACSs that are appropriately coupled). Where a Device and Gateway are managed by independent ACSs, it is assumed that there is no requirement for either ACS to be made aware of the Device-Gateway association.

The defined mechanism relies on the Device's use of DHCP [1-6]. It is expected that the vast majority of remotely manageable Devices will use DHCP, though not necessarily all such Devices. While the mechanism defined here for Device-Gateway association requires the use of DHCP, a Device using this mechanism need not use DHCP for address allocation. This mechanism makes no assumptions about the address allocated to the Device. That is, the Device may have a private or public IP address.

1.1.1 Terminology

The following terminology is used throughout this document.

ACS	Auto-Configuration Server. This is a component in the broadband network responsible for auto-configuration of the CPE for advanced services.
CPE	Customer Premises Equipment.
Device	CPE connected via local area network through a Gateway, bridge, or router.
Device Identity	A three-tuple that uniquely identifies a Device, which includes the manufacturer OUI, serial number, and (optionally) product class.
Gateway	Internet Gateway Device as defined in TR-069 [1-2].
Gateway Identity	<u>A three-tuple that uniquely identifies a Gateway, which includes the manufacturer OUI, serial number, and (optionally) product class.</u>
Parameter	A name-value pair representing a manageable CPE parameter made accessible to an ACS for reading and/or writing.

1.1.2 Document Conventions

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [1-1].

1.2 Procedures

The procedures for Device-Gateway association are summarized as follows:

- A Device following this specification will pass its Device Identity to the Gateway via a vendor-specific DHCP option. When the Gateway receives this information, it populates a table containing identity information for each Device on its LAN. This information is made available to the ACS via an extension to the Gateway's data model, defined in section 1.2.6 of this specification.
- In the DHCP responses, the Gateway provides the Device with its Gateway Identity, which the Device makes available to the ACS via the GatewayInfo data object defined in [1-4]. The Device notifies the ACS of changes to the contents of this object. Thus a Device connecting to a previously unknown Gateway will result in the ACS being notified of the Gateway Identity.
- To ensure the validity of this information, which is carried over an inherently insecure DHCP exchange, the ACS should validate the Gateway Identity provided by the Device by crosschecking against the Device Identity provided by the Gateway.

1.2.1 Gateway Requirements

A Gateway conforming to this specification MUST inspect all DHCP requests received on a LAN interface and determine if the requesting Device has included its Device Identity in the request. A DHCP request is determined to include the Device Identity if it contains a V-I Vendor-Specific Information DHCP Option (option number 125, as defined in [1-7]) that includes the Device Identity information, as defined in section 1.2.5. The DHCP requests for which this requirement applies are DHCPDISCOVER, DHCPREQUEST, and DHCPINFORM.

If the DHCP request is determined to include the Device Identity, then the Gateway MUST do the following:

- The Gateway MUST include its Gateway Identity in all subsequent DHCP responses. The Gateway Identity is carried in the V-I Vendor-Specific Information DHCP Option (option number 125, as defined in [1-7]), as defined in section 1.2.5. The DHCP responses for which this requirement applies are DHCPOFFER and DHCPACK.
- On successful completion of the DHCP exchange (following the DHCPACK), if an entry with a matching Device Identity is not currently listed in the ManagementDevice table, then the Gateway MUST add a new entry in its ManageableDevice table (see section 1.2.6) that includes the Device Identity for this Device.

The Gateway MUST adhere to the following additional requirements:

- The Gateway MUST retain a Device's entry in the ManagementDevice table as long as the Device remains actively connected to the Gateway's LAN.
- The Gateway MUST remove a Device's entry when either:
 - The DHCP lease expires or is released.
 - The Gateway determines that the Device is no longer actively connected to the Gateway's LAN using a locally defined means of connectivity detection.
- The Gateway MUST allow the ACS to request active notification on additions or deletions to the ManageableDevice table. If the ACS has set the Notification Attribute for the parameter Internet-GatewayDevice.ManagementServer.ManageableDeviceNumberOfEntries to Active Notification, then the Gateway MUST notify it each time a Device entry is added or removed using the

Notification mechanism defined in TR-069. If Active Notification is enabled for this parameter, the Gateway **MUST** limit the frequency of Active Notification resulting from changes to the number of entries in the ManageableDevice table as specified by the value of the ManageableDeviceNotificationLimit parameter in the same object.

1.2.2 Device Requirements

A Device conforming to this specification **MUST** do the following:

- In DHCP requests, the Device **MUST** include a V-I Vendor-Specific Information DHCP Option (option number 125, as defined in [1-7]) that includes its Device Identity information, as defined in section 1.2.5. The DHCP requests for which this requirement applies are DHCPDISCOVER, DHCPREQUEST, and DHCPINFORM.
- If the DHCPACK message includes the Gateway Identity carried in the V-I Vendor-Specific Information DHCP Option (option number 125, as defined in [1-7]), as defined in section 1.2.5, the Device **MUST** record the received value in the GatewayInfo data object defined in [1-4]. All of the following values must be recorded:

Device.GatewayInfo.ManufacturerOUI

Device.GatewayInfo.SerialNumber

Device.GatewayInfo.ProductClass

The DHCP responses for which this requirement applies are DHCPOFFER and DHCPACK.

- If any of the elements of the Gateway Identity are not present in the V-I Vendor-Specific Information DHCP Option, the Device **MUST** record an empty string for each such item (replacing the previous value, if any).
- For all of the parameters in the Device.GatewayInfo object, the Device **MUST** by default set the Notification attribute as defined in [1-2] to Active Notification. The Device **MUST** apply this default whenever the URL of the ACS is set or subsequently modified. Whenever Active Notification is enabled for these parameters, the device **MUST** actively notify the ACS as defined in [1-2] if the value of any of these parameters changes.
- If the DHCP lease is released or expires without renewal, all entries in the GatewayInfo object **MUST** be discarded (set to the empty string).

1.2.3 ACS Requirements

Whenever a Device is associated with a Gateway, the Device will notify the ACS, providing the new Gateway Identity information. When this occurs, the ACS **SHOULD** do the following:

- If the ACS has previously associated the Device with a Gateway, the ACS **SHOULD** examine the Gateway Identity from the Device from the GatewayInfo object) and compare it to the Gateway Identity of the prior association. If the association is unchanged, the ACS need not take any further action.
- If the Gateway Identity from the Device is different from the identity of the Gateway previously associated with the Device, or if there was no previous Gateway association for the Device, then the ACS **SHOULD** first validate the information provided by the Device, and if validated, update the Device-Gateway association to indicate the new Gateway Identity.

The ACS **SHOULD** consider the association valid *only* if all elements of the Device Identity match the Device Identity elements in at least one entry in the ManageableDevice table of the indicated Gateway (see section 1.2.6). The ACS would determine the current contents of the ManageableDevice table either by contacting the Gateway using a Connection Request to read the table, or receiving Active Notifications on additions and deletions to this table (by the ACS having previously requested Active Notifications on the ManageableDeviceNumberOfEntries parameter).

1.2.4 Device-Gateway Association Flows

Figure 1 shows the flow associated with the procedures for Device-Gateway association, where the Device uses a DHCP Discover message to initiate the association as part of DHCP address allocation.

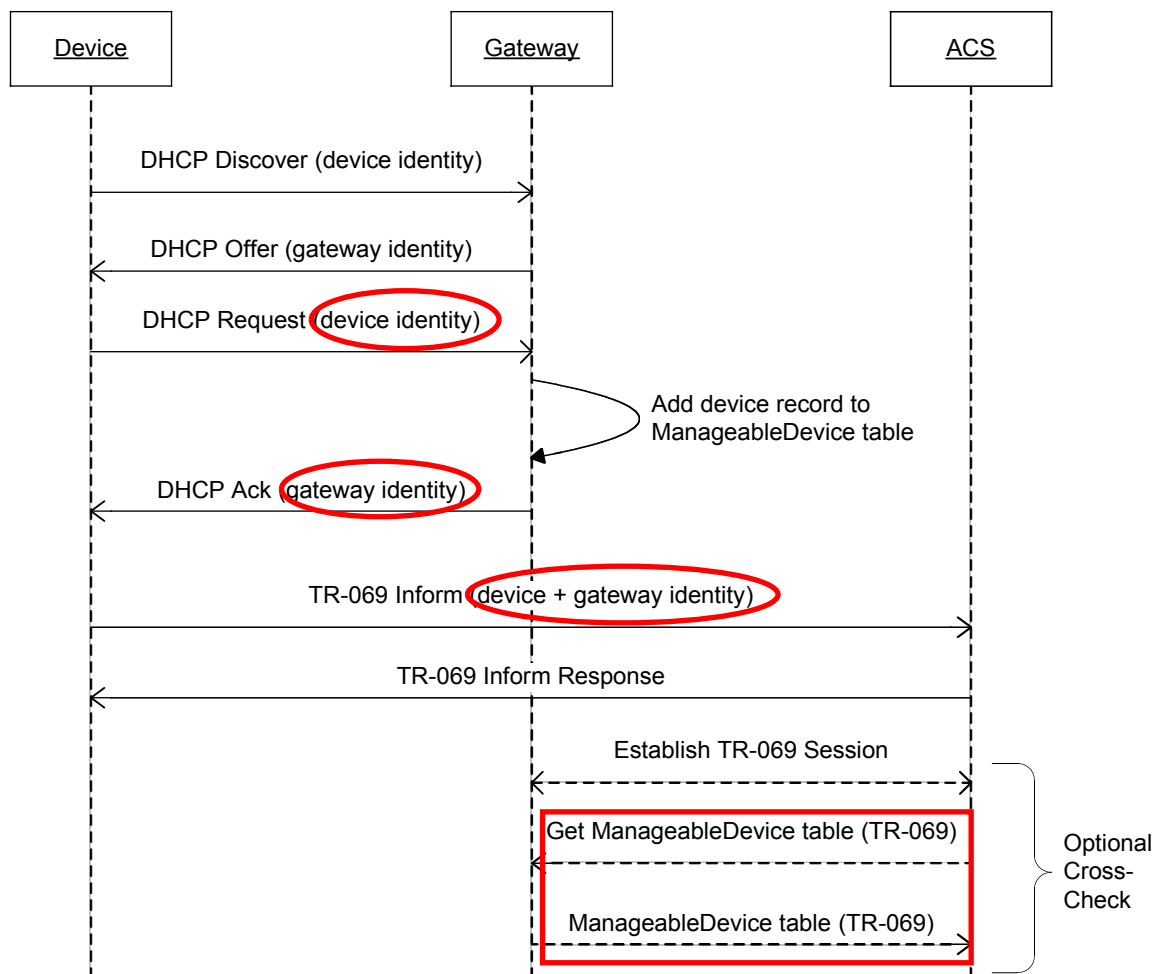


Figure 1 – Device-Gateway Association using **DHCP Discover**

The use of DHCP does not dictate that the device use DHCP for address allocation. If the Device obtains IP addressing parameters using other means, the device would use a DHCP Inform for the exchange of information with the Gateway. The flow for this case is show in Figure 2.

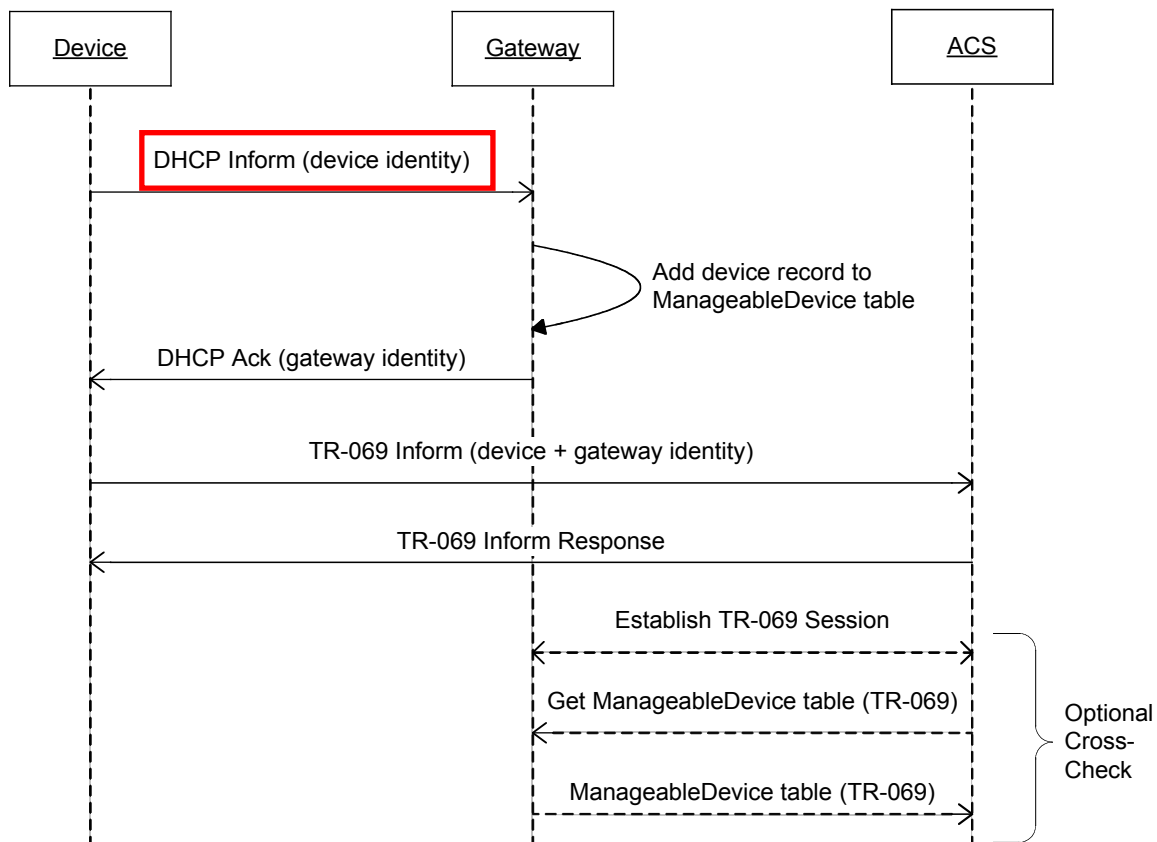


Figure 2 – Device-Gateway Association Using DHCP Inform

1.2.5 DHCP Vendor Options

The Device Identity and Gateway Identity information exchanged via DHCP MUST be contained within the V-I Vendor-Specific Information DHCP Option, which is option number 125, as defined in [1-7]. This DHCP option is defined to allow vendor-specific information from multiple distinct organizations, where the specific organization is explicitly identified via an IANA Enterprise Number.

For DHCP messages that contain Device Identity or Gateway Identity information, the V-I Vendor-Specific Information DHCP Option MUST include an element identified with the IANA Enterprise Number for the DSL Forum that follows the format defined below. The IANA Enterprise Number for the DSL Forum is **3561** in decimal (the “ADSL Forum” entry in the IANA Private Enterprise Numbers registry [1-8]).

Each vendor-specific element within this DHCP Option is defined to contain a series of one or more Encapsulated Vendor-Specific Option-Data fields, encoded as specified in [1-7]. Each such field includes a Sub-Option Code, a Sub-Option Length, and Sub-Option Data. The values for these elements defined in this specification are listed in Table 1.

Table 1 – Encapsulated Vendor-Specific Option-Data fields

Encapsulated Option	Sub-Option Code	Source Entity	Source Parameter ¹
DeviceManufacturerOUI	1	Device	Device.Info.ManufacturerOUI ²
DeviceSerialNumber	2	Device	Device.Info.SerialNumber ²
DeviceProductClass	3	Device	Device.Info.ProductClass ²
GatewayManufacturerOUI	4	Gateway	InternetGatewayDevice.DeviceInfo.ManufacturerOUI
GatewaySerialNumber	5	Gateway	InternetGatewayDevice.DeviceInfo.SerialNumber
GatewayProductClass	6	Gateway	InternetGatewayDevice.DeviceInfo.ProductClass

In encoding the source parameter value in the corresponding Sub-Option Data element, the resulting string MUST NOT be null terminated.

For a DHCP request from the Device that contains the Device Identity, the DHCP Option MUST contain the following Encapsulated Vendor-Specific Option-Data fields:

- DeviceManufacturerOUI
- DeviceSerialNumber
- DeviceProductClass (this may be left out if the corresponding source parameter is not present)

For a DHCP response from the Gateway that contains the Gateway Identity, the DHCP Option MUST contain the following Encapsulated Vendor-Specific Option-Data fields:

- GatewayManufacturerOUI
- GatewaySerialNumber
- GatewayProductClass (this may be left out if the corresponding source parameter is not present)

1.2.6 InternetGatewayDevice Data-Model Extension

To support the Device-Gateway association, an extension to the InternetGatewayDevice data model is specified in Table 2. This extension is considered part of **InternetGatewayDevice:1.2** (version 1.2 of the InternetGatewayDevice data model), which extends version 1.1 of the data model defined in TR-098 [1-3].

Table 2 – InternetGatewayDevice extension to support Device-Gateway Association

Name ³	Type	Write ⁴	Description	Default ⁵
InternetGatewayDevice.ManagementServer.	object	-	This object contains parameters relating to the CPE's association with an ACS.	-
...
ManageableDeviceNumberOfEntries	unsignedInt	-	Number of entries in the ManageableDevice table.	-
ManageableDeviceNotificationLimit	unsignedInt	W	The minimum time, in seconds, between Active Notifications resulting from changes to the	-





¹ The value of the corresponding Sub-Option Data element is obtained from the specified parameter value.

² As defined in [1-4].

³ The full name of a Parameter is the concatenation of the name of the object in which the Parameter is directly contained and the Parameter name listed. An object directly contains all of the Parameters listed below that object name in the table prior to the next object listed.

⁴ “W” indicates the parameter MAY be writable (if “W” is not present, the parameter is defined as read-only). For an object, “W” indicates object instances can be Added or Deleted.

⁵ The default value of the parameter on creation of an object instance via TR-069. If the default value is an empty string, this is represented by the symbol <Empty>.

Name ³	Type	Write ⁴	Description	Default ⁵
			ManageableDeviceNumberOfEntries (if Active Notification is enabled).	
InternetGatewayDevice.Management-Server.ManageableDevice.{i}. 	object	-	Each entry in this table corresponds to a distinct LAN Device that supports Device-Gateway Association according to this specification as indicated by the presence of the DHCP option specified in section 1.2.2.	-
ManufacturerOUI 	string(6)	-	Organizationally unique identifier of the <u>Device manufacturer</u> as provided to the Gateway by the Device. Represented as a six hexadecimal-digit value using all upper-case letters and including any leading zeros. The value MUST be a valid OUI as defined in [1-5].	-
SerialNumber 	string(64)	-	Serial number of the Device as provided to the Gateway by the Device.	-
ProductClass 	string(64)	-	Identifier of the class of product for which the Device's serial number applies as provided to the Gateway by the Device. If the Device does not provide a Product Class, then this parameter MUST be left empty.	-

1.2.6.1 Notification Requirements

An Internet Gateway Device MUST support Active Notification (see [1-2]) for all parameters defined in Table 2 with the exception of those parameters listed in Table 3. For only those parameters listed Table 3, the CPE MAY reject a request by an ACS to enable Active Notification via the SetParameterAttributes RPC by responding with fault code 9009 as defined in [1-2] (Notification request rejected).

An Internet Gateway Device MUST support Passive Notification (see [1-2]) for all parameters defined in Table 2, with no exceptions.

Table 3 – Parameters for which Active Notification MAY be denied by the CPE

Parameter ⁶
InternetGatewayDevice.ManagementServer.ManageableDevice.{i}.
ManufacturerOUI
SerialNumber
ProductClass

1.2.6.2 Profile Definitions

1.2.6.2.1 Notation

The following abbreviations are used to specify profile requirements:

Abbreviation	Description
R	Read support is REQUIRED.
W	Both Read and Write support is REQUIRED.
P	The object is REQUIRED to be present.
C	Creation and deletion of the object via AddObject and DeleteObject is REQUIRED.

⁶ The name of a Parameter referenced in this table is the concatenation of the name of the object in which the Parameter is directly contained and the Parameter name listed. An object directly contains all of the Parameters listed below that object name in the table prior to the next object listed.

1.2.6.2.2 DeviceAssociation Profile

The DeviceAssociation:1 profile implies support for all of the Gateway requirements defined in section 1.2.1 including the support for the data model parameters as shown in Table 4. The minimum required version for this profile is InternetGatewayDevice:1.2.

Table 4 – DeviceAssociation:1 Profile definition for InternetGatewayDevice:1

Name	Requirement
InternetGatewayDevice.ManagementServer.	-
ManageableDeviceNumberOfEntries	R
InternetGatewayDevice.ManagementServer.ManageableDevice.{i}.	P
ManufacturerOUI	R
SerialNumber	R
ProductClass	R

1.3 Security Considerations

While this specification was designed to provide a high degree of security, some known vulnerabilities remain:

- While the mechanism to allow the ACS to validate the identity information provided to it by the Device is optional, it is strongly encouraged that this validation be implemented. The use of this validation is the only means within the context of this specification to overcome the lack of an inherent integrity checking mechanism in the DHCP exchange between the Device and Gateway. By using this validation, attempts to tamper with the identity information of either the Device or Gateway can be detected by the ACS.
- The condition for validation of the Device-Gateway association is that the Device can communicate over the LAN to the Gateway and that the Device and Gateway can authenticate themselves via TR-069 to the ACS. The possibility exists that a valid Device not present on a Gateway's LAN could falsify its association with a Gateway by providing a communication path between the Device and the Gateway's LAN. For example, a Device could establish a communication path to a server, which in turn communicates with a Trojan horse application on the target LAN, which acts as a proxy for the Device. Providing such a path could make the Device indistinguishable from one physically connected to the LAN. To mitigate this possibility, the Gateway may optionally provide mechanisms to allow the user to monitor and regulate what devices are present on the LAN.

1.4 Normative References

Part 1 of this specification references the following documents:

- [1-1] RFC 2119, *Key words for use in RFCs to Indicate Requirement Levels*, <http://www.ietf.org/rfc/rfc2119.txt>
- [1-2] TR-069, *CPE WAN Management Protocol*, DSL Forum Technical Report
- [1-3] TR-098, *Internet Gateway Device Version 1.1 Data Model for TR-069*, DSL Forum Technical Report
- [1-4] TR-106, *Data Model Template for TR-069-Enabled Devices*, DSL Forum Technical Report
- [1-5] *Organizationally Unique Identifiers (OUIs)*, <http://standards.ieee.org/faqs/OUI.html>
- [1-6] RFC 2131, *Dynamic Host Configuration Protocol*, <http://www.ietf.org/rfc/rfc2131.txt>
- [1-7] RFC 3925, *Vendor-Identifying Vendor Options for Dynamic Host Configuration Protocol version 4 (DHCPv4)*, <http://www.ietf.org/rfc/rfc3925.txt>
- [1-8] *IANA Private Enterprise Numbers registry*, <http://www.iana.org/assignments/enterprise-numbers>

Part 2

Connection Request via NAT Gateway

2.1 Introduction

The CPE WAN Management Protocol defined in TR-069 [2-2] may be used to remotely manage CPE Devices that are connected via a LAN through a Gateway. When an ACS manages a Device connected via a NAT Gateway (where the Device has been allocated a private IP address), TR-069 can still be used for management of the Device, but with the limitation that the Connection Request mechanism defined in TR-069 that allows the ACS to initiate a Session cannot be used.

Part 2 of this document defines an extension to TR-069 that allows an ACS to initiate a Session with a device that is operating behind a NAT Gateway. This provides the equivalent functionality of the TR-069 Connection Request, but makes use of a different mechanism to accommodate this scenario.

The mechanism defined in this document does *not* assume that the Gateway through which the Device is connected supports TR-069. This mechanism requires support only in the Device and the associated ACS.

2.1.1 Terminology

The following terminology is used throughout this document.

ACS	Auto-Configuration Server. This is a component in the broadband network responsible for auto-configuration of the CPE for advanced services.
CPE	Customer Premises Equipment.
Device	CPE connected via local area network through a Gateway, bridge, or router.
Gateway	Internet Gateway Device as defined in TR-069 [2-2].
Parameter	A name-value pair representing a manageable CPE parameter made accessible to an ACS for reading and/or writing.

2.1.2 Document Conventions

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [2-1].

2.2 Procedures

To accommodate the ability for an ACS to issue the equivalent of a TR-069 Connection Request to CPE allocated a private address through a NAT Gateway that might not be TR-069 capable, the following is required:

- The CPE must be able to discover that its connection to the ACS is via a NAT Gateway that has allocated a private IP address to the CPE.
- The CPE must be able to maintain an open NAT binding through which the ACS can send unsolicited packets.

- The CPE must be able to determine the public IP address and port associated with the open NAT binding, and communicate this information to the ACS.

To accomplish the above items, this specification defines a particular use of the STUN mechanism, defined in RFC 3489 [2-5].

The use of STUN for this purpose requires that a new UDP-based Connection Request mechanism be defined to augment the existing TCP-based Connection Request mechanism defined in TR-069.

The procedures for making use of STUN to allow the use of UDP Connection Requests to a CPE are summarized as follows:

- The ACS enables the use of STUN in the CPE (if it is not already enabled by factory default) and designates the STUN server for the CPE to use.
- The CPE uses STUN to determine whether or not the CPE is behind a NAT Gateway with a private allocated address.
- If the CPE is behind a NAT Gateway with a private allocated address, the CPE uses the procedures defined in STUN to discover the binding timeout.
- The CPE sends periodic STUN Binding Requests at a sufficient frequency to keep alive the NAT binding on which it listens for UDP Connection Requests.
- When the CPE determines the public IP address and port for the NAT binding on which it is listening for UDP Connection Requests, and whenever it subsequently changes, the CPE communicates this information to the ACS. Two means are provided by which the ACS, at its discretion, can obtain this information—either from information provided in the STUN Binding Request messages themselves, or via Notification on changes to the UDPConnectionRequestAddress parameter, which the CPE must update to include the public Connection Request address and port.
- Whenever the ACS wishes to establish a connection to the CPE, it may send a UDP Connection Request to the CPE. To accommodate the broadest class of NAT Gateways, this must be sent from the same source address and port as the STUN server.

2.2.1 CPE Requirements

Whenever the STUNEnable parameter in the ManagementServer object is set to true, CPE following the requirements of this specification MUST make use of the procedures defined in STUN [2-5] to determine whether or not address and/or port translation is taking place between the CPE and the STUN server. If address and/or port translation is taking place, the CPE MUST:

- Determine the public IP address and port for the NAT binding on which it is listening for UDP Connection Request messages.
- Discover the NAT binding timeout, and send STUN Binding Request messages at a rate necessary to keep alive this binding.
- Indicate via STUN optional attributes on which binding it is listening for UDP Connection Requests, and if the binding has recently changed. Also, update the UDPConnectionRequestAddress parameter to indicate the current public IP address and port associated with the binding.
- Listen for UDP Connection Request messages, and act on these messages when they arrive.

The details of each of these functions are defined in the following sections.

Note – While the CPE requirements defined here certainly apply to a Device connected via LAN to a Gateway, the same procedures may be followed by a Gateway, which might be operating behind a network-based NAT gateway. Thus the requirements are defined generically for CPE, which may be either a Device or Gateway.

2.2.1.1 Binding Discovery

When STUN is enabled via the STUNEnable parameter in the ManagementServer object, the CPE MUST send Binding Request messages to the STUN server designated in the STUNServerAddress and STUNServerPort parameters, as defined in [2-5]. If no STUNServerAddress is given, the address of the ACS determined from the host portion of the ACS URL MUST be used as the STUN server address.

For the purpose of binding discovery, Binding Requests MUST be sent from the source address and port on which the CPE will be listening for UDP Connection Requests if it determines that address and/or port translation is in use (Binding Requests for binding timeout discovery, will be sent from a different port as described in section 2.2.1.2).

The basic Binding Request message allows the CPE to determine if address and/or port translation is in use between the CPE and the STUN server. This is determined by comparing the source address and port on which the request was sent to the MAPPED-ADDRESS attribute received in a response from the STUN server. If either the address or port is different, then translation is in use.

If it is determined that address and/or port translation is in use, the CPE MUST record the value of the MAPPED-ADDRESS attribute in the most recently received Binding Response. This represents the public IP address and port to which UDP Connection Requests would be sent.

Each time the CPE subsequently sends a Binding Request for the purpose of maintaining the binding (see 2.2.1.2), the CPE MUST again determine if address and/or port translation is in use, and if so, obtain the public IP address and port information from the MAPPED-ADDRESS attribute in a successful Binding Response. The actions the CPE must take when this information changes are defined in section 2.2.1.3.

If the CPE has been provisioned with a STUNUsername and STUNPassword in the ManagementServer object, then if the CPE receives a Binding Error Response from the STUN server with a fault code of 401 (Unauthorized), then the CPE MUST resend the Binding Request with the USERNAME and MESSAGE-INTEGRITY attributes as defined in [2-5]. Whenever a Binding Request is sent that includes the MESSAGE-INTEGRITY attribute, the CPE MUST discard a corresponding Binding Response if the MESSAGE-INTEGRITY attribute in the Binding Response is either invalid, as defined in [2-5], or is not present.

If the local IP address allocated to the CPE changes, the CPE MUST re-discover the binding using the procedures described above. The minimum limit on the Binding Request period defined by STUN-MinimumKeepAlivePeriod does *not* apply in this case.

Other than Binding Request messages sent explicitly in response to a Binding Error Response from the STUN server with a fault code of 401 (Unauthorized), the CPE MUST NOT include the MESSAGE-INTEGRITY attributes in any Binding Request.⁷

The STUN client in the CPE need not support the CHANGE-REQUEST attribute of STUN Binding Requests, nor need it understand the CHANGED-ADDRESS, SOURCE-ADDRESS, and REFLECTED-FROM attributes present in a Binding Response.⁸

The STUN client in the CPE need not support the STUN messages for exchanging a Shared Secret. None of these messages are used in the application defined in this specification.

2.2.1.2 Maintaining the Binding

To keep alive the NAT binding, the CPE MUST periodically retransmit Binding Request messages from the source address and port on which the CPE will be listening for UDP Connection Requests.

⁷ Because the STUN specification requires the STUN server to use message integrity in its response if message integrity was used in the request, the CPE cannot use message integrity for Binding Requests on its own, but only when so directed by the STUN server. This is to ensure that the server has total discretion as to when and whether message integrity is to be used.

⁸ These attributes are primarily intended to allow discovery of the type of NAT in use, which is not required for this specification.

The CPE MUST NOT send these Binding Requests more frequently than is specified by the STUN-MinimumKeepAlivePeriod parameter in the ManagementServer object.

The CPE MUST send these Binding Requests at least as frequently as is specified by the STUNMaximumKeepAlivePeriod parameter in the ManagementServer object, if a value is specified.

If the value of STUNMinimumKeepAlivePeriod and STUNMaximumKeepAlivePeriod are not equal, then the CPE MUST actively discover the longest keep-alive period for which the NAT binding is maintained. To do this, the CPE MUST use the procedures described generally in [2-5] to learn the binding timeout. Specifically, the CPE MUST be able to test whether the binding has timed out by sending Binding Requests from a secondary source port distinct from the primary source port, and use the RESPONSE-ADDRESS attribute in the Binding Request to indicate that the STUN Binding Response be sent to the primary source port (the port on which the CPE is listening for UDP Connection Request messages).

The specific procedures by which the CPE uses Binding Requests from the secondary source port to determine the binding timeout is left to the discretion of the CPE vendor. In general, the procedure would consist of two phases: a discovery phase, and a monitoring phase. During the discovery phase, the CPE is attempting to learn the value of the binding timeout, and would test different timeout values to determine the actual timeout value (for example, using a binary search). During the monitoring phase, the CPE would periodically test the binding prior to refreshing it to determine if the binding is still in place. If not, the CPE could then revert to the discovery phase to determine a new value for the binding.

The minimum limit on the Binding Request period defined by STUNMinimumKeepAlivePeriod does *not* apply to Binding Requests sent from a secondary source port.

2.2.1.3 Communication of the Binding Information to the ACS

Two means are defined by which the ACS can be informed of the binding information. The CPE MUST support both methods.⁹ The first method involves the use of optional STUN attributes sent in the Binding Requests. The second method involves the CPE updating the value of the UDPConnectionRequestAddress parameter as the binding information changes.

Table 5 specifies a set of STUN attributes defined for this application. These use Attribute Type values that are greater than 0x7FFF, which the STUN specification defines as “optional.” STUN servers that do not understand optional attributes, are required to ignore them.

⁹ Defining two methods allows flexibility by the ACS in making the tradeoffs between these two approaches. Specifically, the STUN-based approach may require a tighter coupling between the ACS itself and the associated STUN server, while the Notification-based approach may result in greater communication overhead.

Table 5 – Optional STUN attributes used in Binding Request messages

Attribute Type	Name	Description
0xC001	CONNECTION-REQUEST-BINDING	<p>Indicates the binding on which the CPE is listening for UDP Connection Requests.</p> <p>The content of the Value element of this attribute MUST be the following byte string:</p> <pre>0x64 0x73 0x6C 0x66 0x6F 0x72 0x75 0x6D 0x2E 0x6F 0x72 0x67 0x2F 0x54 0x52 0x2D 0x31 0x31 0x31 0x20</pre> <p>This corresponds to the following text string:¹⁰</p> <p>"dslforum.org/TR-111 "</p> <p>A space character is the last character of this string so that its length is a multiple of four characters.</p> <p>The Length element of this attribute MUST equal:</p> <pre>0x0014 (20 decimal)</pre>
0xC002	BINDING-CHANGE	<p>Indicates that the binding has changed.</p> <p>This attribute contains no value. Its Length element MUST be equal to zero.</p> <p>This attribute MUST only be used where the CONNECTION-REQUEST-BINDING is also included.</p>

A CPE **MUST** include the CONNECTION-REQUEST-BINDING attribute in every Binding Request message whose source address and port are the address and port on which it is listening for UDP Connection Request messages. In all other Binding Request messages, the CPE **MUST NOT** include this attribute.

In every Binding Request message sent in which the CPE includes the CONNECTION-REQUEST-BINDING attribute, if the value of the STUNUsername parameter in the ManagementServer object is non-empty, the CPE **MUST** include the USERNAME attribute set to the value of the STUNUsername parameter.

Whenever the CPE detects a change to the NAT binding (as well as the first time the CPE determines the binding), it **MUST** immediately send a Binding Request message from the primary source port (the port on which the CPE is listening for UDP Connection Request messages) that includes the BINDING-CHANGE attribute. This Binding Request **MUST NOT** include the RESPONSE-ADDRESS or CHANGE-REQUEST attributes. In all other Binding Request messages, the CPE **MUST NOT** include the BINDING-CHANGE attribute. The minimum limit on Binding Request period defined by STUNMinimumKeep-AlivePeriod does *not* apply to Binding Requests that include the BINDING-CHANGE attribute.

For Binding Requests that include the BINDING-CHANGE attribute, the CPE **MUST** follow the retransmission procedures define in [2-5] to attempt to ensure the successful reception. If, following these retransmission procedures, the CPE determines that the Binding Request has failed, it **MUST NOT** make further attempts to send Binding Requests that include the BINDING-CHANGE attribute (until the binding subsequently changes again).

When the CPE determines that address and/or port mapping is in use, and whenever the CPE determines that the binding has changed (as well as the first time the CPE determines the binding), the CPE **MUST**

¹⁰ This text string is used to allow an observer, including the NAT Gateway itself, to identify that these STUN messages represent UDP Connection Request bindings associated with this specification. A Gateway might use this knowledge to optimize the associated performance. For example, a Gateway could lengthen the UDP timeout associated with this binding to reduce the frequency of binding updates.

update the value of the `UDPConnectionRequestAddress` parameter in the `ManagementServer` object. Specifically:

- The Host portion of the `UDPConnectionRequestAddress` MUST be set to the current public IP address for the binding associated with the UDP Connection Request as determined from the most recent binding information.
- The Port portion of the `UDPConnectionRequestAddress` MUST be set to the current public port for the binding associated with the UDP Connection Request as determined from the most recent binding information.

When the CPE determines that address and/or port mapping is in use, the CPE MUST also set the `NATDetected` parameter in the `ManagementServer` object to true.

If the ACS has set the Notification attribute on the `UDPConnectionRequestAddress` parameter to Active Notification, then whenever the binding information has changed, the CPE MUST establish a connection to the ACS and include the `UDPConnectionRequestAddress` in the Inform message, as defined in [2-2].

When the `UDPConnectionRequestAddress` is changed, if the time since the most recent Notification on a change to the `UDPConnectionRequestAddress` is less than the value of `UDPConnectionRequestAddress-NotificationLimit`, the Notification MUST be delayed until the specified minimum time period is met.

Note – In addition to the specified minimum notification period, the CPE MAY use its discretion to delay notifying the ACS of updated binding information in order to avoid excessive notifications. Such a delay should only be used if the CPE is confident that the binding is likely to change again within a brief period. For example, during active discovery of the binding timeout it is reasonable to expect frequent binding changes. Similarly, a CPE may be able to detect that a security attack is causing frequent binding changes, and limit the number of notifications until the attack ceases.

If the CPE determines that neither address nor port mapping are in use, then the CPE MUST indicate this to the ACS by setting the `NATDetected` parameter to false, and setting the `UDPConnectionRequestAddress` such that the Host and Port are the local IP address and port on which the CPE is listening for UDP Connection Request messages.

2.2.1.4 UDP Connection Requests

A CPE conforming to this specification MUST listen for UDP Connection Request messages on the port that it has designated for this purpose. This MUST be true whether or not the CPE has detected address or port translation in use, and whether or not the use of STUN is enabled.

Note – a CPE MUST also continue to listen for TCP-based Connection Requests as defined in [2-2].

The format of the UDP Connection Request message is defined in section 2.2.2.3. When the CPE receives a UDP Connection Request message, it MUST both authenticate and validate the message.

A UDP Connection Request message is valid if and only if the following requirements are met:

- It MUST not violate any requirements specified in [2-6] for an HTTP 1.1 request message.
- The Method given in the Request Line MUST be “GET”.
- The Timestamp given by the value of the “ts” query string argument MUST be strictly greater than the Timestamp value for the UDP Connection Request message that had been most recently received, validated, and authenticated.

To allow the above comparison to be made, the CPE MUST maintain a persistent record of Timestamp value of the most recent UDP Connection Request that was successfully validated and authenticated (except across CPE reboots). The Timestamp value for any UDP Connection Request message that fails to be validated or authenticated MUST NOT be recorded. The CPE MAY maintain a record of this most recent Timestamp across CPE reboots. If the CPE does not maintain this value across reboots, then immediately following the reboot the value zero MUST be used.

The CPE MAY place stricter requirements on the Timestamp than stated above. The CPE MAY, for example, additionally verify that the Timestamp is within a time window relative to its understanding of the current time. If a CPE chooses to do this, it SHOULD avoid making the time window too narrow, in order to allow for a reasonable margin of error in both the CPE and ACS.

- The Message ID given by the value of the “id” query string argument MUST be distinct from that of the UDP Connection Request message that had been most recently received, validated, and authenticated.
- The Username given by the value of the “un” query string argument MUST match the value of the parameter Device.ManagementServer.ConnectionRequestUsername.

A UDP Connection Request message is authenticated if and only if the following requirements are met:

- The Signature given by the value of the “sig” query string argument MUST match the value of the signature locally computed by the CPE following the procedure specified in section 2.2.2.3 using the local value of the parameter Device.ManagementServer.ConnectionRequestPassword.

Whenever a CPE receives and successfully authenticates and validates a UDP Connection Request, it MUST follow the same requirements as for a TCP-based Connection Request that are defined in [2-2].

The CPE MUST ignore a UDP Connection Request that is not successfully authenticated or validated.

The CPE MUST ignore the content of any non-empty Message Body that might be present in the UDP Connection Request (this allows the possibility of the use of a non-empty message body in a future version of this protocol).

Because STUN responses and UDP Connection Requests will be received on the same UDP port, the CPE MUST appropriately distinguish STUN messages from UDP Connection Requests using the content of the messages themselves. As the first byte of all STUN messages defined in [2-5] is either 0 or 1, and the first byte of the UDP Connection Request is always an ASCII encoded alphabetic letter, the CPE MAY use this distinction to distinguish between these messages.

2.2.2 ACS Requirements

An ACS following the requirements of this specification MUST be associated with a STUN server that follows the requirements defined in this section.

2.2.2.1 STUN Server Requirements

The STUN server MUST conform to all of the requirements defined in [2-5], with the following exceptions, which the STUN server MAY choose not to implement.

- The STUN server need not support the Shared Secret exchange mechanism defined in [2-5]. If message integrity is used, the shared secrets MUST be statically provisioned, and correspond to the STUNUsername and STUNPassword parameters in the ManagementServer object in the CPE.
- The STUN server need not support a secondary source IP address or port for sending Binding Responses (A2/P2). If it does not, the CHANGED-ADDRESS attribute SHOULD be filled in with the primary address and port (A1/P1), and the STUN server MAY ignore the CHANGE-REQUEST attribute if received in a Binding Request.

The STUN server MAY require message integrity for any received Binding Requests of its choosing by responding to the request with a Binding Error Response with fault code 401 (Unauthorized).

2.2.2.2 Determination of the Binding Information

The ACS may choose either of the two defined mechanisms to determine the current binding information from a CPE.

2.2.2.2.1 *STUN-based Approach*

If the ACS chooses to use the attributes received by the STUN server, it **SHOULD** set a non-empty STUNUsername and STUNPassword in the ManagementServer object of each CPE. The STUNUsername **MUST** be unique among all CPE managed by the corresponding ACS to ensure that the CPE can be distinguished. The STUNPassword **SHOULD** be unique among all CPE managed by the corresponding ACS, and **SHOULD** follow the password strength guidelines specified in [2-5].

Whenever the STUN server receives a Binding Request that includes both the BINDING-CHANGE and CONNECTION-REQUEST-BINDING attributes:

- The STUN server **SHOULD** respond with a Binding Error Response with fault code 401 (Unauthorized) in order to force the CPE to retransmit the Binding Request with message integrity included.
- When the STUN server receives the retransmitted request with message integrity, it **SHOULD** authenticate the requester. This would likely involve communication between the STUN server and ACS if they were not implemented as a single entity.
- If the authentication fails, the STUN server **MUST** respond with a Binding Request Error as defined in [2-5] and take no further action.
- If the authentication is successful, the STUN server **SHOULD** extract the source IP address and port from the Binding Request message, and record these as the new IP address and port to be used for UDP Connection Requests. Depending on the implementation, this may involve the STUN server informing the ACS of the IP address and port along with the corresponding STUNUsername, from which the ACS would then record this information for the CPE corresponding to that STUNUsername.
- The STUN server should perform the above only once for a given Transaction ID in the Binding Request. Redundant copies of the Binding Request with the same Transaction ID **SHOULD** be ignored.

Using this approach, the STUN server **MAY** choose not to require message integrity or authenticate any Binding Requests other than those for which it follows the above procedures to determine the binding information.

The ACS **MAY** determine the current binding at any time even if no change was notified by following the above procedure on any received Binding Request for which the CONNECTION-REQUEST-BINDING attribute is present. The required presence of the USERNAME attribute in these Binding Requests allows the ACS to tentatively determine the CPE's identity prior to subsequent authentication. This allows an ACS to periodically verify the binding information to ensure that it is up-to-date in case explicit indications of a binding change had failed to reach the ACS.

If the ACS determines that the CPE is no longer behind a NAT that is doing address or port mapping, the ACS **MAY** use TCP-based Connection Requests as defined in [2-2].

2.2.2.2.2 *Notification-based Approach*

If the ACS chooses to use Active Notification on the UDPConnectionRequestAddress parameter, it **SHOULD** do the following:

- Set the Notification attribute for the UDPConnectionRequestAddress parameter to Active Notification.
- Record changes to the UDPConnectionRequestAddress parameter whenever this parameter is included in the Inform message, and use the most recently recorded value to determine the destination of UDP Connection Request messages. Specifically, the destination IP address for UDP Connection Request messages is determined from the "host" portion of this parameter, and the destination port is determined from the "port" portion of this parameter. If the host is given as a domain name, the ACS **MUST** use DNS to determine the associated IP address. If the port is not

explicitly given in the `UDPConnectionRequestAddress` parameter, port 80 MUST be used as the default value.

- Observe the value of the `NATDetected` parameter (either by reading it when `UDPConnectionRequestAddress` changes, or by enabling Active Notification on this parameter as well). Whenever this parameter is false, the ACS MAY use TCP-based Connection Requests as defined in [2-2].

Using this approach, the ACS MAY choose not to require message integrity or authenticate any STUN Binding Requests, since these requests are not used to convey information to the ACS. In this case, the ACS need not set a `STUNUsername` or `STUNPassword` in the CPE.

2.2.2.3 UDP Connection Requests

The ACS MUST send UDP Connection Request messages from the same source IP address and port as the STUN server.

A UDP Connection Request message MUST be transmitted within a single UDP packet sent to the IP address and port determined by the ACS as described in section 2.2.2.2.

The ACS SHOULD send multiple copies of the same UDP Connection Request message in order to reduce the likelihood that the message is lost due to packet loss. When an ACS sends multiple copies of the same UDP Connection Request, the content of the message (including the message ID, timestamp, and nonce, as defined below) MUST be identical for each successive copy.

There is no response message associated with a UDP Connection Request message.

The format of the UDP Connection Request message is derived from the format of an HTTP 1.1 [2-6] GET message, though the HTTP 1.1 protocol itself is not used. Specifically, the UDP Connection Request message MUST conform to the following requirements:

- It MUST be a valid HTTP 1.1 GET message as defined in [2-6].
- It MUST contain no Message Body.
- If a Content-Length header is present, its value MUST be zero.
- The Method given in the Request Line MUST be “GET”.
- The Request-URI given in the Request Line MUST be an Absolute-URI according to the rules defined in [2-7]. The URI MUST be formed as follows:
 - The Scheme portion of the URI MUST be “http” or “HTTP”.
 - The Authority portion of the URI MUST be as specified in [2-7]. The ACS MAY set this to the value of `Device.ManagementServer.UDPConnectionRequestAddress`, if it is known. Otherwise, the ACS MUST derive this string from the actual destination IP address and port to which the UDP Connection Request message will be sent. The “port” portion of this string MUST be present unless the destination port number is “80”.
 - The Path portion of the URI MUST be empty.
 - The Query portion of the URI MUST contain a query string encoded as defined by the “application/x-www-form-urlencoded” content type defined in [2-8]. The query string MUST contain the following name-value pairs:

Name	Value
ts	Timestamp. The number of seconds since the Unix epoch until the time the message is created (the standard Unix timestamp).
id	Message ID. An unsigned integer value that MUST be set to the same value for all retransmitted copies of the same UDP Connection Request. The value MUST change between successive distinct UDP Connection Requests.
un	Username. The value of the parameter <code>Device.ManagementServer.ConnectionRequestUsername</code> as read from the CPE.

Name	Value
cn	Cnonce. A random string chosen by the ACS.
sig	<p>Signature. Formed from the 40-character hexadecimal representation (case insensitive) of HMAC-SHA1 (Key, Text) [2-9], where:</p> <ul style="list-style-type: none"> Key is the value of the parameter Device.ManagementServer.Connection-RequestPassword as read from the CPE. Text is a string formed by concatenating the following elements (in the order listed, with no spaces between items): <ul style="list-style-type: none"> The value of the ts (Timestamp) element The value of the id (Message ID) element The value of the un (Username) element The value of the cn (Cnonce) element

Below is an example Request-URI:

```
http://10.1.1.1:8080?ts=1120673700&id=1234&un=CPE57689
&cn=XTGRWIPC6D3IPXS3&sig=3545F7B5820D76A3DF45A3A509DA8D8C38F13512
```

2.2.3 Message Flows

The following figures show example message flows associated with the procedures defined in sections 2.2.1 and 2.2.2 to support Connection Requests to devices behind a NAT gateway.

In all of the examples, the address/port pairs use the notation (A, P) , where A is the IP address and P is the port. In the examples, the CPE uses $(A1, P1)$ as its primary port (the port on which the CPE is listening for UDP Connection Request messages) and $(A1, P2)$ is its secondary port (used for binding timeout discovery). When passing through a NAT Gateway, these addresses are translated to $(A1', P1')$ and $(A1', P2')$, respectively. In all of the examples it is assumed that the STUN Server does not have a secondary address/port and thus the CHANGED-ADDRESS attribute in the Binding Response (which need not be used by the CPE) contains its primary address/port, $(A3, P3)$.

Figure 3 shows the periodic binding discovery and binding maintenance flows where the CPE sends the Binding Request from the primary source port and includes the CONNECTION-REQUEST-BINDING and (if a Username had been set) USERNAME attributes. In this example it is assumed that the STUN Server has not chosen to authenticate the request.

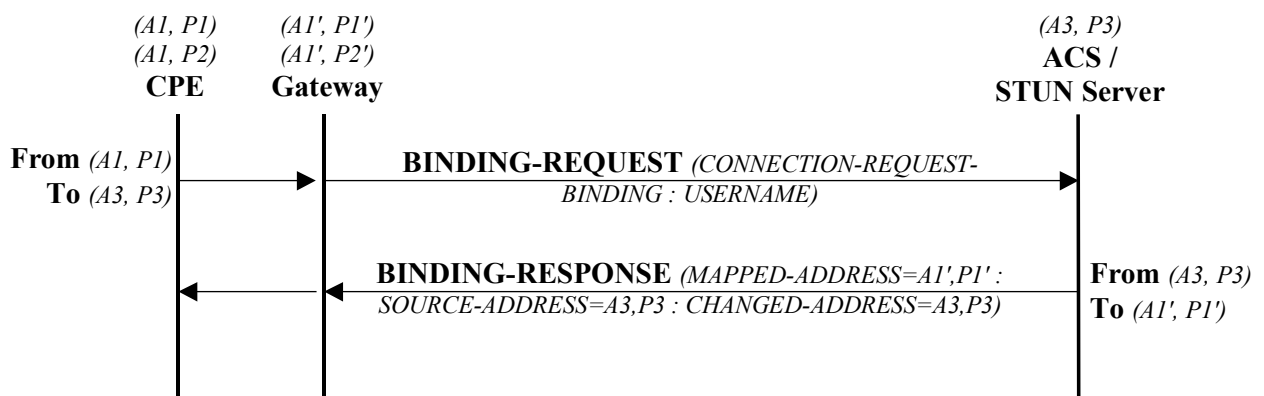


Figure 3 – Binding discovery / maintenance from the primary source port

Figure 4 shows a Binding Request sent by the CPE from its secondary source port for the purpose of discovering whether or not the primary binding has timed out in the NAT gateway. In this case the Binding Request does not include the CONNECTION-REQUEST-BINDING attribute since it is not sent from the

primary source port. The last leg of the exchange (shown in grey) will not occur if the primary binding has timed out.

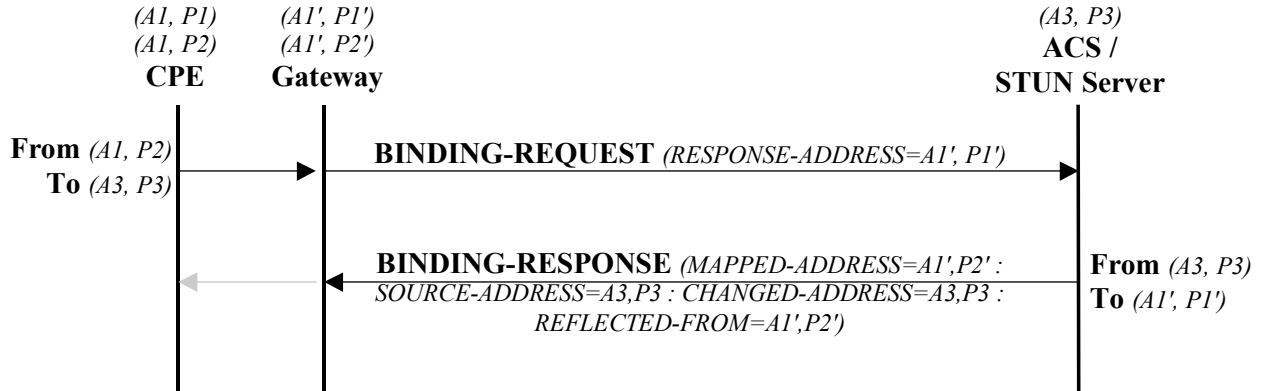


Figure 4 – Binding Request from secondary source port for binding timeout discovery

Figure 5 shows a Binding Change notification where the STUN Server has chosen to make use of the STUN-based approach (see section 2.2.2.2.1), and therefore authenticates the Binding Request prior to storing the information associating the Username with the current binding address and port.

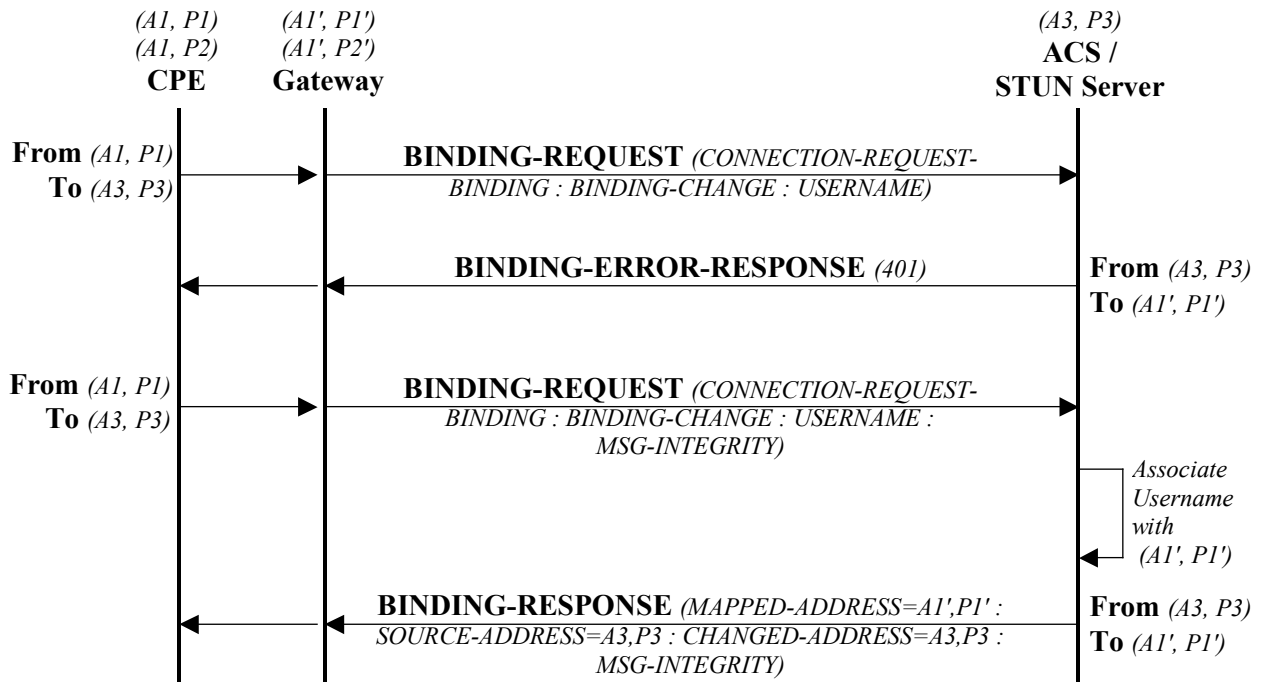


Figure 5 – Binding change notification authenticated by the ACS

Figure 6 shows a Binding Change notification where the STUN Server has chosen to make use of the Notification-based approach (see section 2.2.2.2.2), and therefore does not need to authenticate the Binding Request since the ACS instead uses TR-069 Notification to update the binding information.

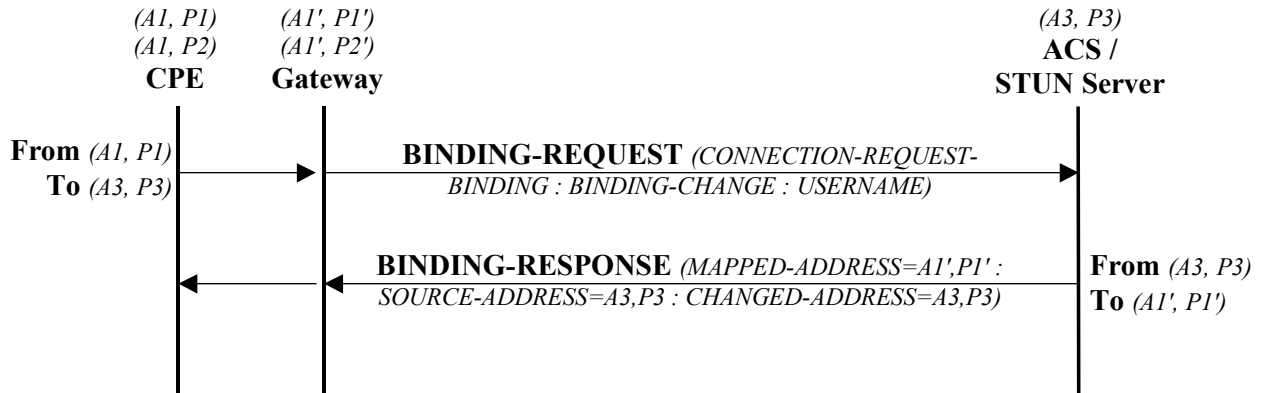


Figure 6 – Binding change notification *not* authenticated by the ACS

Figure 7 shows a UDP Connection Request message sent to the CPE to initiate a TR-069 session. In this example, the STUN Server sends the identical UDP Connection Request multiple times to improve the likelihood of successful reception by the CPE.

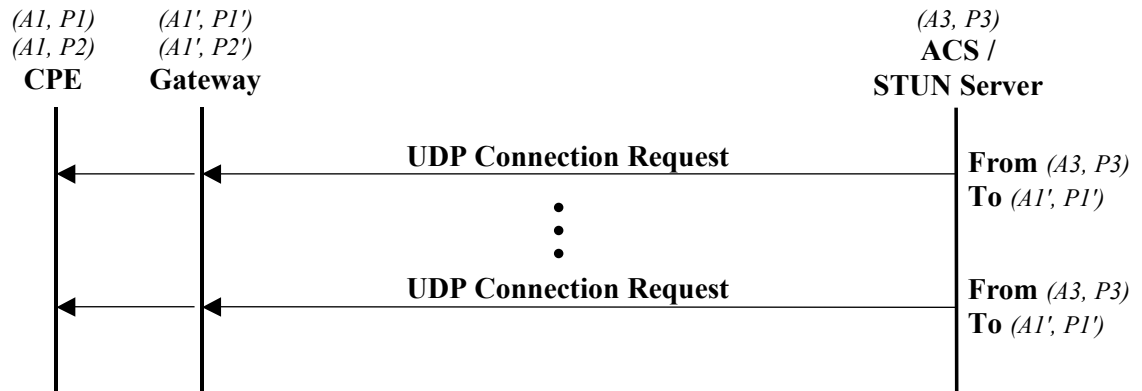


Figure 7 – UDP Connection Request

2.2.4 Data-Model Extension

To support the functionality defined in this specification to allow Connection Request via NAT Gateway, an extension to the Device data model is specified in Table 6. This extension is considered part of **Device:1.1** (version 1.1 of the Device data model), which extends version 1.0 of the Device data model defined in TR-106 [2-4].

If the CPE is an Internet Gateway, the same extension applies to the InternetGatewayDevice data model. This extension is considered part of **InternetGatewayDevice:1.2** (version 1.2 of the InternetGatewayDevice data model), which extends version 1.1 of the data model defined in TR-098 [2-3].

In both cases, the new parameters specified here are defined within the existing `ManagementServer` object.

Table 6 – Data-model extension to support UDP Connection Requests

Name ¹¹	Type	Write ¹²	Description	Default ¹³
.ManagementServer.	object	-	This object contains parameters relating to the CPE's association with an ACS.	-
...
UDPConnectionRequestAddress	string(256)	-	<p>Address and port to which an ACS MAY send a UDP Connection Request to the CPE.</p> <p>This parameter is represented in the form of an Authority element as defined in [2-7]. The value MUST be in one of the following two forms:</p> <p style="padding-left: 40px;">host:port</p> <p style="padding-left: 40px;">host</p> <p>When STUNEnable is true, the "host" and "port" portions of this parameter MUST represent the public address and port corresponding to the NAT binding through which the ACS can send UDP Connection Request messages (once this information is learned by the CPE through the use of STUN).</p> <p>When STUNEnable is false, the "host" and "port" portions of the URL MUST represent the local IP address and port on which the CPE is listening for UDP Connection Request messages.</p> <p>The second form of this parameter MAY be used only if the port value is equal to "80".</p>	-
UDPConnectionRequestAddressNotification-Limit	unsignedInt	W	The minimum time, in seconds, between Active Notifications resulting from changes to the UDP-ConnectionRequestAddress (if Active Notification is enabled).	-
STUNEnable	boolean	W	Enables or disables the use of STUN by the CPE. This applies only to the use of STUN in association with the ACS to allow UDP Connection Requests.	-
STUNServerAddress	string	W	<p>Host name or IP address of the STUN server for the CPE to send Binding Requests if STUN is enabled via STUNEnable.</p> <p>If empty and STUNEnable is true, the CPE MUST use the address of the ACS extracted from the host portion of the ACS URL.</p>	-
STUNServerPort	unsignedInt [0:65535]	W	<p>Port number of the STUN server for the CPE to send Binding Requests if STUN is enabled via STUNEnable.</p> <p>By default, this SHOULD be the equal to the default STUN port, 3478.</p>	-
STUNUsername	string(256)	W	<p>If non-empty, the value of the STUN USERNAME attribute to be used in Binding Requests (only if message integrity has been requested by the STUN server).</p> <p>If empty, the CPE MUST NOT send STUN Binding Requests with message integrity.</p>	-

¹¹ The full name of a Parameter is the concatenation of the name of the object in which the Parameter is directly contained and the Parameter name listed. An object directly contains all of the Parameters listed below that object name in the table prior to the next object listed.

¹² "W" indicates the parameter MAY be writable (if "W" is not present, the parameter is defined as read-only). For an object, "W" indicates object instances can be Added or Deleted.

¹³ The default value of the parameter on creation of an object instance via TR-069. If the default value is an empty string, this is represented by the symbol <Empty>.

Name ¹¹	Type	Write ¹²	Description	Default ¹³
STUNPassword	string(256)	W	The value of the STUN Password to be used in computing the MESSAGE-INTEGRITY attribute to be used in Binding Requests (only if message integrity has been requested by the STUN server). When read, this parameter returns an empty string, regardless of the actual value.	-
STUNMaximumKeepAlivePeriod	int[-1:]	W	If STUN Is enabled, the maximum period, in seconds, that STUN Binding Requests MUST be sent by the CPE for the purpose of maintaining the binding in the Gateway. This applies specifically to Binding Requests sent from the UDP Connection Request address and port. A value of -1 indicates that no maximum period is specified.	-
STUNMinimumKeepAlivePeriod	unsignedInt	W	If STUN Is enabled, the minimum period, in seconds, that STUN Binding Requests may be sent by the CPE for the purpose of maintaining the binding in the Gateway. This limit applies only to Binding Requests sent from the UDP Connection Request address and port, and only those that do not contain the BINDING-CHANGE attribute. This limit does not apply to retransmissions following the procedures defined in [2-5].	-
NATDetected	boolean	-	When STUN is enabled, this parameter indicates whether or not the CPE has detected address and/or port mapping in use. A true value indicates that the received MAPPED-ADDRESS in the most recent Binding Response differs from the CPE's source address and port. When STUNEnable is false, this value MUST be false.	-

2.2.4.1 Notification Requirements

CPE MUST support both Active and Passive Notification (see [2-2]) for all parameters defined in Table 6, with no exceptions.

2.2.4.2 Profile Definitions

2.2.4.2.1 Notation

The following abbreviations are used to specify profile requirements:

Abbreviation	Description
R	Read support is REQUIRED.
W	Both Read and Write support is REQUIRED.
P	The object is REQUIRED to be present.
C	Creation and deletion of the object via AddObject and DeleteObject is REQUIRED.

2.2.4.2.2 UDPConnReq Profile for Device:1

The UDPConnReq:1 profile for a Device implies support for all of the CPE requirements defined in section 2.2.1, including support for the data model parameters as shown in Table 7. The minimum required version for this profile is Device:1.1.

Table 7 – UDPConnReq :1 Profile definition for Device:1

Name	Requirement
Device.ManagementServer.	-
UDPConnectionRequestAddress	R
UDPConnectionRequestAddressNotificationLimit	W
STUNEnable	W
STUNServerAddress	W
STUNServerPort	W
STUNUsername	W
STUNPassword	W
STUNMaximumKeepAlivePeriod	W
STUNMinimumKeepAlivePeriod	W
NATDetected	R

2.2.4.2.3 UDPConnReq Profile for InternetGatewayDevice:1

The UDPConnReq:1 profile for an Internet Gateway Device implies support for all of the CPE requirements defined in section 2.2.1, including support for the data model parameters as shown in Table 8. The minimum required version for this profile is InternetGatewayDevice:1.2.

Table 8 – UDPConnReq :1 Profile definition for InternetGatewayDevice:1

Name	Requirement
InternetGatewayDevice.ManagementServer.	-
UDPConnectionRequestAddress	R
UDPConnectionRequestAddressNotificationLimit	W
STUNEnable	W
STUNServerAddress	W
STUNServerPort	W
STUNUsername	W
STUNPassword	W
STUNMaximumKeepAlivePeriod	W
STUNMinimumKeepAlivePeriod	W
NATDetected	R

2.3 Security Considerations

The following security considerations are associated with the procedures defined in this section are identified:

- The STUN specification describes several potential attacks using the STUN mechanism. The reader is referred to section 12 of RFC 3489 [2-5] for a detailed description of these potential attacks and the associated risk.
- Because binding changes will result in actions required by the ACS—authentication of a CPE, and subsequent database update, and potentially establishment of a TR-069 session over which to

receive an Inform—attacks that can cause frequent changes to the NAT binding could result in an increased burden on the ACS. The ACS may set a minimum limit on the rate of Notifications on binding changes if Active Notification is used. However, there is a tradeoff between the maximum Notification rate and the length of time for which the ACS may not be able to send Connection Requests to the CPE due to out-of-date information.

2.4 Normative References

Part 2 of this specification references the following documents:

- [2-1] RFC 2119, *Key words for use in RFCs to Indicate Requirement Levels*, <http://www.ietf.org/rfc/rfc2119.txt>
- [2-2] TR-069, *CPE WAN Management Protocol*, DSL Forum Technical Report
- [2-3] TR-098, *Internet Gateway Device Version 1.1 Data Model for TR-069*, DSL Forum Technical Report
- [2-4] TR-106, *Data Model Template for TR-069-Enabled Devices*, DSL Forum Technical Report
- [2-5] RFC 3489, *STUN - Simple Traversal of User Datagram Protocol (UDP) Through Network Address Translators (NATs)*, <http://www.ietf.org/rfc/rfc3489.txt>
- [2-6] RFC 2616, *Hypertext Transfer Protocol -- HTTP/1.1*, <http://www.ietf.org/rfc/rfc2616.txt>
- [2-7] RFC 3986, *Uniform Resource Identifier (URI): Generic Syntax*, <http://www.ietf.org/rfc/rfc3986.txt>
- [2-8] *HTML 4.01 Specification*, <http://www.w3.org/TR/html4>
- [2-9] RFC 2104, *HMAC: Keyed-Hashing for Message Authentication*, <http://www.ietf.org/rfc/rfc2104.txt>