DHC Working Group
Internet Draft
Intended status: Standards Track
Expires: March 2011

Rajiv Asati
Cisco Systems

Ralph Droms
Cisco Systems

September 29, 2010

DHCP Relay Agent Configuration Option
draft-asati-dhc-relay-agent-config-00.txt

Abstract

This document defines a Dynamic Host Configuration Protocol (DHCP) Relay Agent Configuration option and associated machinery to configure the Relay Agent.

Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at
http://www.ietf.org/ietf/1id-abstracts.txt

The list of Internet-Draft Shadow Directories can be accessed at
http://www.ietf.org/shadow.html

This Internet-Draft will expire on March 29, 2011.

Table of Contents

1. Introduction

There are scenarios in which a network operator (Service Provider or Enterprise) may desire the relay agent to be dynamically provisioned while facilitating the server-client communication to ultimately facilitate the service activation in a zero-touch manner.

One example is the provisioning of the Provider Edge (PE) router, acting as the relay agent for the Customer Edge (CE) router, acting as the (DHCP) client, during IP/VPN [RFC4364] service activation.

DHCP [RFC2131][RFC3315] is the predominant signaling protocol to dynamically assign IP addresses and other TCP/IP configuration parameters to routers and hosts. DHCP Relay Agent functionality [RFC3046] is specified to facilitate the forwarding of DHCP messages between the client and server through the relay agent.

DHCP server may use one or more sub-options within the "Relay Agent Information" option [RFC3046] appended by Relay Agent, for IP address and other parameter assignment policies to the Client. The "Relay Agent Information" option [RFC3046] is limited to providing the additional information from Relay Agent to the DHCP server to aid the server.

This document proposes a new DHCP option (and sub-options) that the Relay Agent can use to request and receive the desired Relay Agent configuration information and that the DHCP server can use to deliver the requested configuration information. The document also describes the associated procedures and operations for the Relay Agent and Server to achieve the auto-provisioning of VPN information at the PE router acting as the relay agent.

2. Key Words to Reflect Requirements

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 BCP 14, RFC 2119 [RFC2119]. RFC 2119 defines the use of these key words to help make the intent of standards track documents as clear as possible. While this document uses these keywords, this document is not a standards track document.

Additionally, this document freely uses the terms that are defined in [RFC2131][RFC2132][RFC3046].

3. Problem / Requirement

There are other methods to activate the VPN service by auto-provisioning the CE router after it establishes the layer2 connectivity. However, this assumes and requires the adjacent PE router to be provisioned in advance to ensure that the CE gets the IP reachability through the PE router, and is able to participate in the any-to-any VPN such as BGP IP/VPN [RFC4364]. This is one of the key challenges that serve as one of the requirements for the solution prescribed in this document. Another requirement is to make use of the existing signaling protocol(s) and not impose multiple protocols to achieve this.

4. Relay Agent Configuration Option

This document defines a new DHCP Option called the Relay Agent Configuration Option. It is a "container" option for specific sub-options. The format of the Relay Agent Configuration option is:

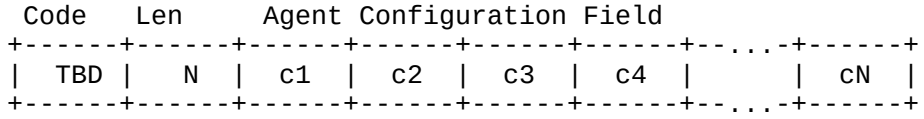


Figure 1 Relay Agent Configuration Option

Code = DHCP Option for Relay Agent Configuration (to be allocated by IANA)

Len = Total number of octets (N) in the Agent Configuration Field (inclusive of all sub-options)

Agent Configuration Field = One or more Sub-options, each encoded as a SubOpt/Length/Value tuple, as shown below:

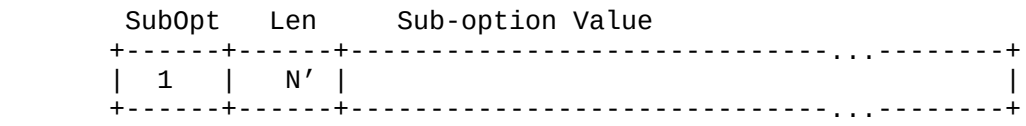


Figure 2 Relay Agent Configuration Sub-Option

SubOpt = DHCP Sub-Option for Relay Agent Configuration (to be allocated by IANA)

Len = Total number of octets (N') in a Sub-option

Sub-option Value = zero or more octets to encode the value.

The sub-options need not appear in any particular order.

5. Operation

5.1. DHCP Relay Agent Procedures

The relay agent adds the DHCP relay agent configuration option (& needed sub-options) in the relayed message to request the relay agent side configuration information from the server.

The addition of this option SHOULD be configurable, and SHOULD be disabled by default. Relay agents SHOULD have separate configurables for each sub-option to control whether it is added to client-to-server packets.

A relay agent adding a Relay Agent Configuration Information Option MUST add it as the last option (but before 'End Option' 255, if present) or the second last option, if option 82 is present, in the DHCP options field of any recognized BOOTP or DHCP packet forwarded from a client to a server.

If the configuration information, provided by the DHCP server in its response, is already present at the relay agent, then relay agent SHOULD compare the existing configuration with the new one, and in case of a mismatch, logs an error/event.

The relay agent MUST remove the relay agent configuration option from the DHCP response and forward the remaining response to the client.

The operation of relay agent for specific sub-options should be specified with that sub-option.

5.1.1. DHCP Relay Agent Chaining

Relay agents receiving a DHCP packet from an untrusted circuit with giaddr set to zero (indicating that they are the first-hop router) but with a Relay Agent Configuration option already present in the packet SHALL discard the packet and increment an error count.

A trusted circuit may contain a trusted downstream (closer to client) network element (bridge) between the relay agent and the client that MAY add a relay agent option but not set the giaddr field. In this case, the relay agent does NOT add a "second" relay agent option, but forwards the DHCP packet per normal DHCP relay agent operations, setting the giaddr field as it deems appropriate.

The mechanisms for distinguishing between "trusted" and "untrusted" circuits are specific to the type of circuit termination equipment, and may involve local administration.

5.2. DHCP Server Procedures

The DHCP server examines the DHCP options in the incoming request, and constructs the response. The DHCP server may poll any other servers present in the OSS/BSS to construct the requested configuration information, and ultimately include it in the relay agent configuration option/sub-options of DHCP response.

5.3. DHCP Client Procedures

This document doesn’t specify any changes to the client functioning.

The new option defined in this document is never passed to the client.

6. Security Considerations

There are no specific security considerations within the scope of this document.

7. IANA Considerations

TBD.

8. Acknowledgments

Thanks to Shwetha Bhandari for providing feedback.

This document was prepared using 2-Word-v2.0.template.dot.

9. References

9.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC2544] Bradner, S. and McQuaid, J., "Benchmarking Methodology for Network Interconnect Devices", RFC 2544, March 1999.

[RFC2131] Droms, R., "Dynamic Host Configuration Protocol", RFC 2131, March 1997.

[RFC5695] Droms, R. and Alexanderand S., "DHCP Options and BOOTP Vendor Extensions", RFC 5695, March 1997.

[RFC3315] Droms, et. al., "Dynamic Host Configuration Protocol for IPv6 (DHCPv6)", RFC 3315, July 2003.

[RFC3046] Patrick, M., "DHCP Relay Agent Information Option", RFC 3046, January 2001.

9.2. Informative References

[RFC4364] Rosen, E. and Rekther, Y., "BGP/MPLS IP Virtual Private Networks (VPNs)", RFC 4364, February 2006.

APPENDIX A: Applicability

A.1. Applicability to MPLS IP/VPN

Figure 3 below illustrates a sample MPLS/VPN network topology in which CE1, CE2 and CE3 are part of the same Virtual Private Network (VPN), which is represented by VRF VPN1, say, in the MPLS/VPN network.

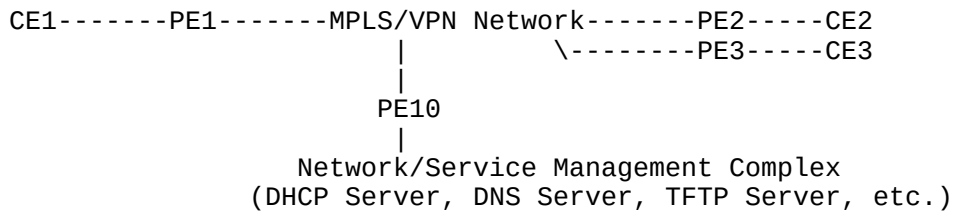


Figure 3 A Sample Network Topology

The "Network/Service Management Complex" is where the DHCP Server, DNS server, TFTP server etc. may reside.

The PE router is assumed to have the DHCP relay agent functionality as suggested in this document. The relay agent functionality may be included globally for all PE-CE interfaces or selectively on individual PE-CE interfaces.

Optionally, the unused PE-CE interfaces at the PE router may be assigned to a default VRF prior to the successful DHCP processing and auto-configuration. This helps to avoid having the CE get the global reachability by accident prior to the DHCP operation completion.

Assuming that the PE-CE interface is ready for the layer1/layer2 connectivity, CE would (be programmed to) broadcast the DHCP request when the layer2 connectivity is established on either all or designated port(s).

- This ensures that the DHCP request reaches the PE router.
- The DHCP request may include CE's unique identifier (such as MAC address or S/N or Unique Device Identifier (UDI) etc.) that is already known to the Servers in the Network/Service Management Complex.

PE router upon receiving the DHCP request on a layer2 interface that isn't configured with any IP address, relays it to the DHCP server that may be pre-provisioned.

PE adds the DHCP relay agent configuration option (& needed sub-options) in the relayed message to request the PE side configuration information.

The DHCP server examines the DHCP options in the incoming request, and constructs the response. The DHCP server may poll any other servers present in the OSS/BSS for the PE configuration information, so as to include it in the options/sub-options of DHCP response.

The PE configuration information, in RFC4364 environment, may contain one or more of the following -

- IP address and subnet for PE-CE interface
- VRF Configuration (RD, RT etc.)
- Other PE-CE Interface configuration (description, vrf mapping etc.)
- Selected Routing Protocol instance (for the CE)
- Neighbor and ASN information in case of BGP or EIGRP
- Security, QoS information etc. (for the CE)

If the VRF configuration, provided by the DHCP server in its response, is already present at the PE router, then PE router must compare the existing config with the new one, and logs an error/event that could be sent to the DHCP server or to the OSS/BSS or both, in case of a mismatch.

PE should accept the new config. The error/event log will help to get the operator attention for further validation. New DHCP sub-option is defined for this purpose.

The PE router removes the PE specific information (the new DHCP relay agent configuration option) from the DHCP response and forward the remaining response to the CE router, which will process it as usual (not impacted by this document).

Authors' Addresses

Rajiv Asati
Cisco Systems,
7025 Kit Creek Rd, RTP, NC, 27709
Email: rajiva@cisco.com

Ralph Droms
Cisco Systems,
200 Beaver Brook Road, Boxborough, MA, 01719
Email: rdroms@cisco.com

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: September 15, 2011

Y. Cui
J. Wu
P. Wu
Tsinghua University
C. Metz
Cisco Systems, Inc.
O. Vautrin
Juniper Networks
Y. Lee
Comcast

March 14, 2011

Public IPv4 over Access IPv6 Network
draft-cui-softwire-host-4over6-04

Abstract

This draft proposes a mechanism for bidirectional IPv4 communication between IPv4 Internet and end hosts or IPv4 networks sited in IPv6 access network. This mechanism follows the softwire hub & spoke model and uses IPv4-over-IPv6 tunnel as basic method to traverse IPv6 network. By allocating public IPv4 addresses to end hosts/networks in IPv6, it can achieve IPv4 end-to-end bidirectional communication between these hosts/networks and IPv4 Internet. This mechanism is an IPv4 access method for hosts and IPv4 networks sited in IPv6.

Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on September 15, 2011.

Copyright Notice

Table of Contents

1. Introduction

Global IPv4 addresses are running out fast. Meanwhile, the demand for IP address is still growing and may even burst in potential circumstances like "Internet of Things". To satisfy the end users, operators have to push IPv6 to the front, by building IPv6 networks and providing IPv6 services.

When IPv6-only network are widely deployed, users of those networks will probably still need IPv4 connectivity. This is because part of Internet will stay IPv4-only for a long time, and network users in IPv6-only network will communicate with network users sited in the IPv4-only part of Internet. This need could eventually decrease with the general IPv6 adoption.

Network operators should provide IPv4 services to IPv6 users to satisfy their needs, usually through tunnels. This type of IPv4 services differ in provisioned IPv4 addresses. If the users can't get public IPv4 addresses (e.g., new network users join an ISP which don't have enough unused IPv4 addresses), they have to use private IPv4 addresses on the client side, and IPv4-private-to-public translation is required on the carrier side, as is described in Dual-stack Lite[I-D.ietf-softwire-dual-stack-lite]. Otherwise the users can get public IPv4 addresses, and use them for IPv4 communication. In this case, translation on the carrier side won't be necessary. The network users and operators can avoid all the issues raised by translation, such as ALG, NAT traversal, state maintenance, etc. Note that this "public IPv4" situation is actually quite common. There're approximatively 2^32 network users who are using or can potentially get public IPv4 addresses. Most of them will switch to IPv6 sooner or later, and will require IPv4 services for a significant period after the switching. This draft focuses on this situation, i.e., to provide IPv4 access for users in IPv6 networks, where public IPv4 addresses are still available for allocation.

2. Requirements language

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 [RFC2119].

3. Terminology

Public 4over6: Public 4over6 is the mechanism proposed by this draft. Generally, Public 4over6 supports bidirectional communication between IPv4 Internet and IPv4 hosts or local networks in IPv6 access network, by leveraging IPv4-in-IPv6 tunnel and public IPv4 address allocation.

4over6 initiator: in Public 4over6 mechanism, 4over6 initiator is the IPv4-in-IPv6 tunnel initiator located on the user side of IPv6 network. The 4over6 initiator can be either a dual-stack capable host or a dual-stack CPE device. In the former case, the host has both IPv4 and IPv6 stack but is provisioned with IPv6 access only. In the latter case, the CPE has both IPv6 interface for access to ISP network and IPv4 interface for local network connection; hosts in the local network can be IPv4-only.

4over6 concentrator: in Public 4over6 mechanism, 4over6 concentrator is the IPv4-in-IPv6 tunnel concentrator located in IPv6 ISP network. It’s a dual-stack router which connects to both the IPv6 network and IPv4 Internet.

4. Deployment scenario

4.1. Scenario description

The general scenario of Public 4over6 is shown in Figure 1. Users in an IPv6 network take IPv6 as their native service. Some users are end hosts which face the ISP network directly, while others are local networks behind CPEs, such as a home LAN, an enterprise network, etc. The ISP network is IPv6-only rather than dual-stack, which means that ISP can't provide native IPv4 access to its users; however, it's acceptable that one or more routers on the carrier side become dual-stack and get connected to IPv4 Internet. So if network users want to connect to IPv4, these dual-stack routers will be their "entrances".

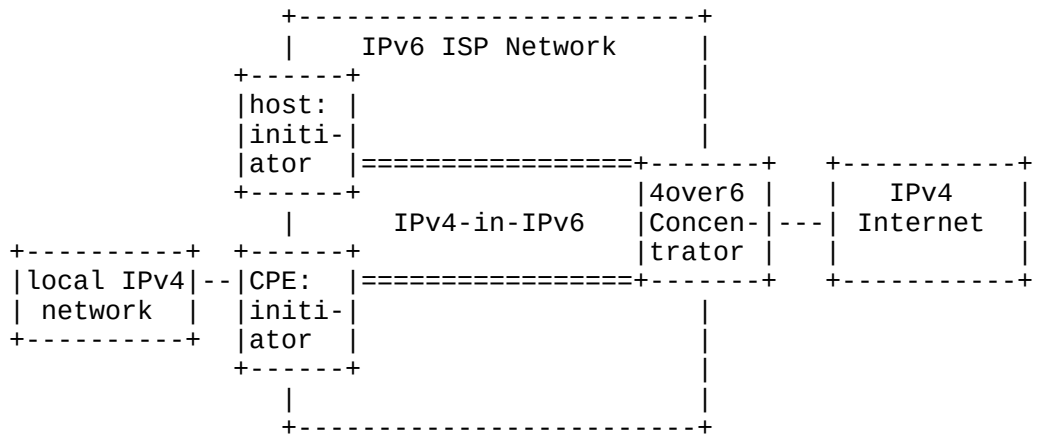


Figure 1 Public 4over6 scenario

4.2. Communication requirements

Before getting into any technical details, the communication requirements should be stated. The first one is that, 4over6 users require IPv4-to-IPv4 communication with the IPv4 Internet. An IPv4 access service is needed rather than an IPv6-to-IPv4 translation service. (IPv6-to-IPv4 communication is out of the scope of this draft.)

Second, 4over6 users require public IPv4 addresses rather than private addresses. Public IPv4 address means there's no IPv4 CGN along the path, so the acquired IPv4 service is better. In particular, some hosts may be application servers, public address

works better for reasons like straightforward access, direct DNS registration, no stateful mapping maintenance on CGN, etc. For the direct-connected host case, each host should get one public IPv4 address. For the local IPv4 network case, there're actually two subcases: one is that every CPE gets one public IPv4 address while local networks remains private IPv4, the other is that end hosts in local networks get public IPv4 addresses. In the first subcase, though the CPE has to run an IPv4 NAT, it's still much better than the situation that involves a CGN, since this NAT is in local network and can be configured and managed by the users.

Third, translation is not preferred in this scenario. If this IPv4-to-IPv4 communication is achieved by IPv4-IPv6 translation, it'll needs double translation along the path, one from IPv4 to IPv6 and the other from IPv6 back to IPv4. It's quite complicated. Contrarily a tunnel can achieve the IPv4-over-IPv6 traversing easily. That's the reason this draft follows the hub & spoke softwire model.

5. Public 4over6 Mechanism

5.1. Address allocation

Public 4over6 can be generally considered as IPv4-over-IPv6 hub & spoke tunnel using public IPv4 address. Each 4over6 initiator will use public IPv4 address for IPv4-over-IPv6 communication. As is described above, in the host initiator case, every host will get one IPv4 address; in the NATed CPE case, every CPE will get one IPv4 address, which will be shared by hosts behind the CPE; in the non-NAT CPE case, every host behind the CPE will get one IPv4 address.

The key problem here is IPv4 address allocation over IPv6 network, from ISP device(s) to separated 4over6 initiators. Native IPv4 address allocation is done either in a dynamic way throug DHCPv4, or in a static way through manual configuration. Public 4over6 should support both. DHCPv4 over IPv6 can be achieved upon IPv4-in-IPv6 tunnel between ISP device and 4over6 initiators. As to manual configuration, 4over6 users and the ISP operators should negotiate beforehand to authorize the IPv4 address. In addition, in the non-NAT CPE case, the address allocation should pass through the CPE initiator and reach IPv4 hosts. This will require a DHCP relay function on the CPE.

Along with this address allocation, the concentrator needs to maintain the address mappings between the allocated IPv4 address and IPv6 address of 4over6 initiators. This is required to provide correct destination address for encapsulation. There are several ways to maintain this mapping: DHCPv4-driven updating, traffic snooping and manual configuration. This draft recommends the first way since it naturally supports bidirectional communication. The next two subsection adopts the first method and describes it in detail. A comparison with traffic snooping is given in section 5.4.

5.2. 4over6 concentrator behavior

4over6 concentrator represents the IPv4-IPv6 border router working as the remote tunnel endpoint for 4over6 initiators, with its IPv6 interface connected to the IPv6 network, IPv4 interface connected to the IPv4 Internet, and a tunnel interface supporting IPv4-in-IPv6 encapsulation and decapsulation. There’s no CGN on the 4over6 concentrator, it won’t perform any translation function; instead, 4over6 concentrator maintains an IPv4-IPv6 address mapping table for IPv4 data encapsulation.

4over6 concentrator is responsible for IPv4 address allocation to 4over6 initiators. For static allocation, the concentrator just install the IPv4-IPv6 address mapping into the mapping table after

negotiating with a 4over6 user, and delete the mapping when the user doesn't need 4over6 anymore. As to dynamic allocation, the concentrator should either run a DHCPv4 server on the tunnel interface to dynamically allocate public addresses to 4over6 initiators, or perform the DHCPv4 relay functions and leave the actual address allocation job to a dedicated DHCPv4 server located in IPv4. In both cases, when allocating an address, the concentrator should install an entry of the allocated IPv4 address and the initiator's IPv6 address into the address mapping table. This entry should be deleted when receiving a DHCP release or reaching a lease expiration of that IPv4 address. All these mapping updates are triggered by the DHCP process(see Figure 2). Note that in the DHCP relay case, the relay should be extended to maintain the lifetime of address leases.

The concentrator sends and receives DHCP packets using IPv4-in-IPv6 tunnel. The difficulty here is that before DHCP address allocation is done, the initiator may not have an IPv4 address, and the concentrator doesn't have an IPv4-IPv6 mapping for IPv4-in-IPv6 encapsulation of the DHCP packets. So when the concentrator receives an encapsulated DHCP packet from an initiator, it should temporarily store the mapping between its IPv6 source address and the MAC address in DHCP payload. This mapping will be used for encapsulation of outgoing DHCP packets. The concentrator should use the MAC address in the payload of an outing DHCP packet to match the correct IPv6 encapsulation destination address.

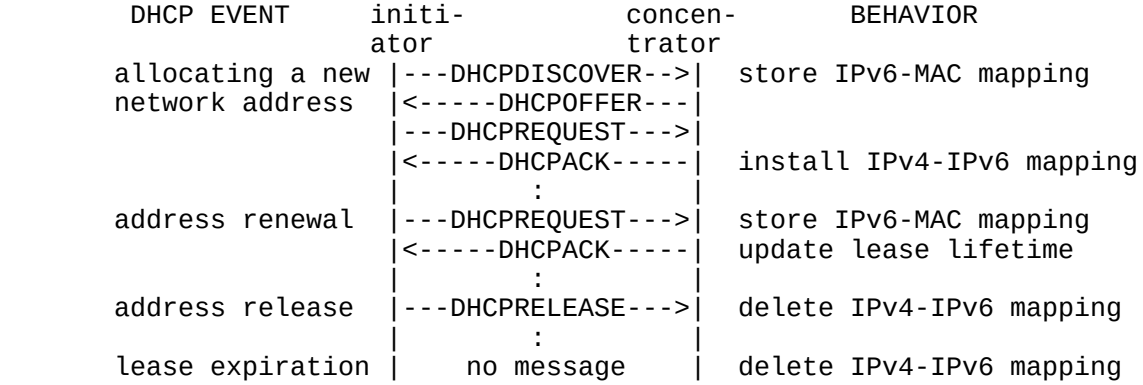


Figure 2 4over6 concentrator: DHCP behavior

On the IPv6 side, 4over6 concentrator decapsulates IPv4-in-IPv6 packets coming from 4over6 initiators. It removes the IPv6 header of every IPv4-in-IPv6 packet and forwards it to the IPv4 Internet. On the IPv4 side, the concentrator encapsulates the IPv4 packets

destined to 4over6 initiators. When performing the IPv4-in-IPv6 encapsulation, the concentrator uses its own IPv6 address as the IPv6 source address. As to the IPv6 destination address, the concentrator will look up the IPv4-IPv6 address mapping table, use the IPv4 destination address of the packet to find the correct IPv6 address. After the encapsulation, the concentrator sends the IPv6 packet on its IPv6 interface to reach an initiator.

The 4over6 concentrator, or its upstream router should advertise the IPv4 prefix which contains the IPv4 addresses of 4over6 users to the IPv4 side, in order to make these initiators reachable on IPv4 Internet.

Since the concentrator has to maintain the IPv4-IPv6 address mapping table, the concentrator is stateful in IP level. Note that this table will be much smaller than a CGN table, as there is no port information involved.

5.3. 4over6 initiator behavior

4over6 initiator has an IPv6 interface connected to the IPv6 ISP network, and a tunnel interface to support IPv4-in-IPv6 encapsulation. In CPE case, it has at least one IPv4 interface connected to IPv4 local network.

4over6 initiator should learn the 4over6 concentrator’s IPv6 address beforehand. For example, if the initiator gets its IPv6 address by DHCPv6, it can get the 4over6 concentrator’s IPv6 address through a DHCPv6 option[I-D.ietf-softwire-ds-lite-tunnel-option].

5.3.1. Host initiator

When the initiator is a direct-connected host, it’ll assign the allocated public IPv4 address to its tunnel interface. If the address allocation is static, the host should negotiate with the ISP operator beforehand. The host should learn the IPv4 address provisioned by the operator, and inform the operator its IPv6 address, to install the address mapping on the concentrator.

Usually, a host gets the public IPv4 address by DHCPv4 over an IPv4-in-IPv6 tunnel. A standard DHCPv4 client on the host will run on the tunnel interface to acquire IPv4 address. All the DHCPv4 packets generated by the client will be encapsulated and forwarded to the 4over6 concentrator, and all the DHCPv4 replies from the concentrator encapsulating in IPv6 will be decapsulated by the tunnel interface and handed to the DHCP client. This way the DHCP client can get a dynamic public IPv4 address from the concentrator, and assign it to the tunnel interface.

For IPv4 data traffic, the host performs the IPv4-in-IPv6 encapsulation and decapsulation on the tunnel interface, which has its IPv4 address already assigned. When sending out an IPv4 packet, it performs the encapsulation, using the IPv6 address of the 4over6 concentrator as the IPv6 destination address, and its own IPv6 address as the IPv6 source address. The encapsulated packet will be forwarded to the IPv6 network. The decapsulation on 4over6 initiator is simple. When receiving an IPv4-in-IPv6 packet, the initiator just drops the IPv6 header, and hands it to upper layer.

5.3.2. NATed CPE as initiator

The NATed CPE case is quite like the host initiator case. The IPv4 address allocation process between the CPE and the concentrator is the same with the corresponding process in host initiator case, and the allocated IPv4 address will be assigned to the tunnel interface of the CPE. The local IPv4 network won't take part in the public IPv4 allocation; instead end hosts will use private IPv4 addresses, possibly allocated by the CPE.

On data plan, the NATed CPE can be viewed as a regular IPv4 NAT(using tunnel interface as the NAT outside interface) cascaded with a tunnel initiator. For IPv4 data packets received from the local network, the CPE translates these packets, using the tunnel interface address as the source address, and then encapsulates the translated packet into IPv6, using the 4over6 concentrator IPv6 address as the destination address, the CPE's IPv6 address as source address. For IPv6 data packet received from the IPv6 network, the CPE performs decapsulation and IPv4 public-to-private translation. As to the CPE itself, it can use the public, tunnel interface address to communicate with the IPv4 Internet, and the private, IPv4 interface address to communicate with the local network.

5.3.3. non-NAT CPE as initiator

When the CPE doesn't perform a NAT function and end hosts in the local network get public IPv4 addresses allocated from the concentrator, the situation becomes a little complicated. To support dynamic address allocation in this situation, the CPE should act as an IPv4 DHCP relay, relaying the DHCP requests and replies between the host and the concentrator. Here the CPE's tunnel interface acts as the "upper" interface of the relay, i.e., the CPE uses an IPv4-in-IPv6 tunnel to forward DHCP messages to and receive DHCP messages from the concentrator. The static allocation method is similar to the former two case.

The remaining problem is what kind of IPv4 address does the CPE use. The address of tunnel interface is only used by the CPE itself, so it

could be a well-known IPv4 address, just like B4's configuration in DS-lite; Or the CPE could get a public IPv4 address from the concentrator and assigned it to the tunnel interface, in case that the CPE has its own IPv4 communication demand. As to the IPv4 interface connected to the local network, its address should be reachable in the local network, i.e., in the same range of the hosts' addresses. So the CPE should get a public IPv4 address from the concentrator and assigned it to the IPv4 interface, or the ISP could claim a specific IPv4 address from its 4over6 DHCPv4 pool and assign this unified address to every non-NAT CPE's IPv4 interface. In either case, the CPE should have its tunnel interface address and IPv4 interface address separated in different address ranges to avoid confusion. Here different strategies achieve different effects and consume IPv4 address to varying degrees. The authors would like to remain this topic as an open issue in this version of draft.

On data plan, for IPv4 data packets received from the local network, the CPE encapsulates them and forward them to IPv6 network. For IPv6 data packet received from the IPv6 network, the CPE performs decapsulation and forward them to IPv4 local network. No translation is requires since the end hosts use public addresses.

5.4. IPv4-IPv6 mapping maintaining methods

section 5.3 describes the address mapping maintaining with DHCP-driven updating, in which DHCP process on the concentrator triggers installation and deletion of IPv4-IPv6 address mappings. Another way to maintain the mapping is traffic snooping, i.e., record the address mapping when decapsulating IPv4-in-IPv6 data packets coming from 4over6 initiators. In this way, the mappings are installed based on the actual traffic rather than DHCP. However, the shortage of this method is that extra procedure is required to support inbound access. This happens when there's no mapping exists on the concentrator for an allocated IPv4 address, either because there's no outbound traffic from this IP yet or because the earlier-installed mapping has expired, while packets from the IPv4 Internet have already arrived on the concentrator and tried to reach the corresponding IP. To solve this problem, the 4over6 initiator need to send keepalive "pinhole" packets to the concentrator, or uses a protocol similar to PCP[I-D.ietf-pcp-base]. This draft recommends the DHCP-driven updating method since it's more accurate and controllable, and requires no extra procedure on the initiator.

If an operator chooses the DHCP-driven updating method, the concentrator need manual mapping configuration as well for static configured 4over6 initiators. The traffic snooping method works for both static and dynamic 4over6 initiators, though.

6. Technical advantages

Public 4over6 provides a method for users in IPv6 network to communicate with IPv4. In many scenarios, this can be viewed as an alternative to IPv6-IPv4 translation mechanisms which have well-known limitations described in [RFC4966] .

Since a 4over6 initiator uses a public IPv4 address, Public 4over6 supports full bidirectional communication between IPv4 Internet and hosts/IPv4 networks in IPv6 access network. In particular, it supports the servers in IPv6 network to provide IPv4 application service transparently.

Public 4over6 supports dynamic reuse of a single IPv4 address between multiple subscribers based on their dynamic requirement of communicating with IPv4 Internet. A subscriber will request a public IPv4 address for a period of time only when it need to communicate with IPv4 Internet. Besides, in the NATed CPE case, one public IPv4 address will be shared by the local network. So Public 4over6 can improve the reuse rate of IPv4 addresses.

Public 4over6 is suited for network users/ISPs which can still get/ provide public IPv4 addresses. Dual-stack lite is suited for network users/ISPs which can no longer get/provide public IPv4 addresses. By combining Public 4over6 and Dual-stack lite, the IPv4-over-IPv6 Hub & spoke problem can be well solved.

7. Acknowledgement

The authors would like to thank Alain Durand and Dan Wing for their valuable comments on this draft.

8. References

8.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC4925] Li, X., Dawkins, S., Ward, D., and A. Durand, "Softwire Problem Statement", RFC 4925, July 2007.

[RFC4966] Aoun, C. and E. Davies, "Reasons to Move the Network Address Translator - Protocol Translator (NAT-PT) to Historic Status", RFC 4966, July 2007.

8.2. Informative References

[I-D.ietf-pcp-base]
Wing, D., Cheshire, S., Boucadair, M., Penno, R., and F. Dupont, "Port Control Protocol (PCP)", draft-ietf-pcp-base-06 (work in progress), February 2011.

[I-D.ietf-softwire-ds-lite-tunnel-option]
Hankins, D. and T. Mrugalski, "Dynamic Host Configuration Protocol for IPv6 (DHCPv6) Option for Dual- Stack Lite", draft-ietf-softwire-ds-lite-tunnel-option-10 (work in progress), March 2011.

[I-D.ietf-softwire-dual-stack-lite]
Durand, A., Droms, R., Woodyatt, J., and Y. Lee, "Dual-Stack Lite Broadband Deployments Following IPv4 Exhaustion", draft-ietf-softwire-dual-stack-lite-07 (work in progress), March 2011.

Authors' Addresses

Yong Cui
Tsinghua University
Department of Computer Science, Tsinghua University
Beijing 100084
P.R.China

Phone: +86-10-6260-3059
Email: yong@csnet1.cs.tsinghua.edu.cn

Jianping Wu
Tsinghua University
Department of Computer Science, Tsinghua University
Beijing 100084
P.R.China

Phone: +86-10-6278-5983
Email: jianping@cernet.edu.cn

Peng Wu
Tsinghua University
Department of Computer Science, Tsinghua University
Beijing 100084
P.R.China

Phone: +86-10-6278-5822
Email: weapon@csnet1.cs.tsinghua.edu.cn

Chris Metz
Cisco Systems, Inc.
3700 Cisco Way
San Jose, CA 95134
USA

Email: chmetz@cisco.com

Olivier Vautrin
Juniper Networks
1194 N Mathilda Avenue
Sunnyvale, CA 94089
USA

Email: Olivier@juniper.net

Yiu L. Lee
Comcast

Email: yiu_lee@cable.comcast.com

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: January 9, 2012

Y. Cui
J. Wu
P. Wu
Tsinghua University
C. Metz
Cisco Systems, Inc.
O. Vautrin
Juniper Networks
Y. Lee
Comcast

July 8, 2011

Public IPv4 over Access IPv6 Network
draft-cui-softwire-host-4over6-06

Abstract

This draft proposes a mechanism for bidirectional IPv4 communication between IPv4 Internet and end hosts or IPv4 networks sited in IPv6 access network. This mechanism follows the softwire hub and spoke model and uses IPv4-over-IPv6 tunnel as basic method to traverse IPv6 network. By allocating public IPv4 addresses to end hosts/networks in IPv6, it can achieve IPv4 end-to-end bidirectional communication between these hosts/networks and IPv4 Internet. This mechanism is an IPv4 access method for hosts and IPv4 networks sited in IPv6.

Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on January 9, 2012.

Copyright Notice

Table of Contents

1. Introduction

Global IPv4 addresses are running out fast. Meanwhile, the demand for IP address is still growing and may even burst in potential circumstances like "Internet of Things". To satisfy the end users, operators have to push IPv6 to the front, by building IPv6 networks and providing IPv6 services.

When IPv6-only networks are widely deployed, users of those networks will probably still need IPv4 connectivity. This is because part of Internet will stay IPv4-only for a long time, and network users in IPv6-only networks will communicate with network users sited in the IPv4-only part of Internet. This demand could eventually decrease with the general IPv6 adoption.

Network operators should provide IPv4 services to IPv6 users to satisfy their demand, usually through tunnels. This type of IPv4 services differ in provisioned IPv4 addresses. If the users can't get public IPv4 addresses (e.g., new network users join an ISP which don't have enough unused IPv4 addresses), they have to use private IPv4 addresses on the client side, and IPv4-private-to-public translation is required on the carrier side, as is described in Dual-stack Lite[I-D.ietf-softwire-dual-stack-lite]. Otherwise the users can get public IPv4 addresses, and use them for IPv4 communication. In this case, translation on the carrier side won't be necessary. The network users and operators can avoid all the issues raised by translation, such as ALG, NAT traversal, state maintenance, etc. Note that this "public IPv4" situation is actually quite common. There're approximatively 2^32 network users who are using or can potentially get public IPv4 addresses. Most of them will switch to IPv6 sooner or later, and will require IPv4 services for a significant period after the switching. This draft focuses on this situation, i.e., to provide IPv4 access for users in IPv6 networks, where public IPv4 addresses are still available for allocation.

2. Requirements language

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 [RFC2119].

3. Terminology

Public 4over6: Public 4over6 is the mechanism proposed by this draft. Generally, Public 4over6 supports bidirectional communication between IPv4 Internet and IPv4 hosts or local networks in IPv6 access network, by leveraging IPv4-in-IPv6 tunnel and public IPv4 address allocation.

4over6 initiator: in Public 4over6 mechanism, 4over6 initiator is the IPv4-in-IPv6 tunnel initiator located on the user side of IPv6 network. The 4over6 initiator can be either a dual-stack capable host or a dual-stack CPE device. In the former case, the host has both IPv4 and IPv6 stack but is provisioned with IPv6 access only. In the latter case, the CPE has both IPv6 interface for access to ISP network and IPv4 interface for local network connection; hosts in the local network can be IPv4-only.

4over6 concentrator: in Public 4over6 mechanism, 4over6 concentrator is the IPv4-in-IPv6 tunnel concentrator located in IPv6 ISP network. It’s a dual-stack router which connects to both the IPv6 network and IPv4 Internet.

4. Deployment scenario

4.1. Scenario and requirements

The general scenario of Public 4over6 is shown in Figure 1. Users in an IPv6 network take IPv6 as their native service. Some users are end hosts which face the ISP network directly, while others are local networks behind CPEs, such as a home LAN, an enterprise network, etc. The ISP network is IPv6-only rather than dual-stack, which means that ISP can't provide native IPv4 access to its users; however, it's acceptable that one or more routers on the carrier side become dual-stack and get connected to IPv4 Internet. So if network users want to connect to IPv4, these dual-stack routers will be their "entrances".

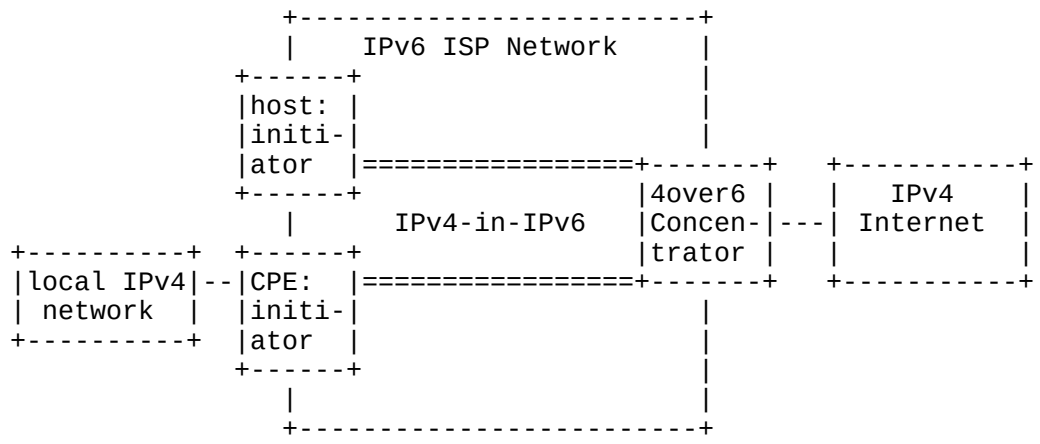


Figure 1 Public 4over6 scenario

Before getting into any technical details, the communication requirements should be stated. The first one is that, 4over6 users require IPv4-to-IPv4 communication with the IPv4 Internet. An IPv4 access service is needed rather than an IPv6-to-IPv4 translation service. (IPv6-to-IPv4 communication is out of the scope of this draft.)

Second, 4over6 users require public IPv4 addresses rather than private addresses. Public IPv4 address means there's no IPv4 CGN along the path, so the acquired IPv4 service is better. In particular, some hosts may be application servers, public address works better for reasons like straightforward access, direct DNS registration, no stateful mapping maintenance on CGN, etc. For the

direct-connected host case, each host should get one public IPv4 address. For the local IPv4 network case, the CPE can get a public IPv4 address and runs an IPv4 NAT for the local network. Here a local NAT is still much better than the situation that involves a CGN, since this NAT is in local network and can be configured and managed by the users.

Third, translation is not preferred in this scenario. If this IPv4-to-IPv4 communication is achieved by IPv4-IPv6 translation, it'll need double translation along the path, one from IPv4 to IPv6 and the other from IPv6 back to IPv4. This would be quite complicated, especially in addressing. Contrarily a tunnel can achieve the IPv4-over-IPv6 traversing easily. That's the reason this draft follows the hub and spoke softwire model.

Moreover, the ISP probably would like to keep their IPv4 and IPv6 addressing and routing separated when provisioning IPv4 over IPv6. Then the ISP can manage the native IPv6 network more easily and independently, and also provision IPv4 in a flexible, on-demand way. The cost is that the concentrator needs to maintain per-user address mapping state, which would be described in detail.

4.2. Use cases

Public 4over6 can be applicable in several practical cases. The first one is that ISPs which still own enough IPv4 addresses switch to IPv6. The ISPs can deploy public 4over6 to preserve IPv4 service for the customers. This case is actually quite common. The majority of the wired end users today get Internet access with public IPv4 address. When their ISPs switch to IPv6, these users can still use the same amount of IPv4 addresses for IPv4 access. Public 4over6 can leveraging these addresses and offer tunneled IPv4 access.

The second case is ISPs which don't have enough IPv4 addresses any more switch to IPv6. For these ISPs, dual-stack lite is so far the most mature solution to provision IPv4 over IPv6. In dual-stack lite, end users use private IPv4 addresses, experience a 44CGN and hence some service degradation. As long as the end users use public IPv4 addresses, all CGN issues can be avoided and the IPv4 service can be full bi-directional. In other words, Public 4over6 can be deployed along with DS-lite, to provide a value-added service. Common users adopt DS-lite to communicate with IPv4 while high-end users adopt Public 4over6. The two mechanisms can actually be coupled easily.

There is also a special situation in the second case that the end users are IPv4 application servers. In this situation, public address brings significant convenience. The DNS registration can be

direct using dedicated address; the access of application clients can be straightforward with no translation; there's no need to reserve and maintain address mapping on the CGN, and no well-known port collision will come up. So it's better to have servers adopt Public 4over6 for IPv4 access when they're located in IPv6 network.

Following the principle of Public 4over6, it's also possible to achieve address multiplexing and save IPv4 addresses. There're already efforts on this subject, see [I-D.cui-softwire-b4-translated-ds-lite] and [I-D.sun-v6ops-laft6]. The basic idea is that instead of allocating a full IPv4 address to every end user, the ISP can allocate an IPv4 address with restricted port range to every end user.

Besides, the draft would like to be explicit about the scope of direct-connected host case and CPE case. The host case is clear: the host is directly connected to IPv6 network, but the protocol stack on the host support IPv4 too. As to the CPE case, this draft would like to only focus on the case that the local network behind the CPE is private IPv4. If the users want to run public IPv4 into the local network, then they can either run dual-stack in the local network and turn into host case(likely home LAN situation), or they can acquire address blocks from the ISP and build configured tunnel or softwire mesh[RFC5565] with the ISP network(likely enterprise network situation). TC can be implemented to be compatible with the latter case too, though.

5. Public 4over6 Mechanism

5.1. Address allocation and mapping maintenance

Public 4over6 can be generally considered as IPv4-over-IPv6 hub and spoke tunnel using public IPv4 address. Each 4over6 initiator will use public IPv4 address for IPv4-over-IPv6 communication. As is described above, in the host initiator case, every host will get one IPv4 address; in the CPE case, every CPE will get one IPv4 address, which will be shared by hosts behind the CPE. The key problem here is IPv4 address allocation over IPv6 network, from ISP device(s) to separated 4over6 initiators.

There’re two possibilities here. One is DHCPv4 over IPv6, and the other is static configuration. DHCPv4 over IPv6 is achieved by performing DHCPv4 on IPv4-in-IPv6 tunnel between ISP device and 4over6 initiators. There do exist the DHCP encapsulation issue on server side, see details and solutions in [I-D.cui-softwire-dhcp-over-tunnel]. As to static configuration, 4over6 users and the ISP operators should negotiate beforehand to authorize the IPv4 address. Application servers usually falls into this case. Public 4over6 supports both address allocation manners. Actually, it is transparent to address allocation methods.

Along with IPv4 address allocation, Public 4over6 should maintain the IPv4-IPv6 address mappings on the concentrator. In this type of address mapping, the IPv4 address is the public IPv4 address allocated to a 4over6 initiator, and the IPv6 addresses is the initiator’s IPv6 address. This mapping is used to provide correct encapsulation destination address for the concentrator.

The initiator sends "pinhole" packets to the concentrator periodically, to install and renew the address mapping. A pinhole packet is an IPv4-in-IPv6 packet, which uses the concentrator’s IPv6 address as destination IPv6 address, the initiator’s IPv6 address as source IPv6 address, and the initiator’s IPv4 address as source IPv4 address. When the concentrator receives such a packet, it’ll resolve the IPv4 and IPv6 address information from the packet and trigger the mapping. Since any IPv4-in-IPv6 data packet from the initiator contains these exact informations, it can also serve as pinhole packet. Then dedicated pinhole packets are sent out when there’s no data packets. Another possible way to maintain the address mapping is to run PCP[I-D.ietf-pcp-base] while extending the protocol to support applying for a full address. The following sections describe the mechanism with the pinhole method.

5.2. 4over6 initiator behavior

4over6 initiator has an IPv6 interface connected to the IPv6 ISP network, and a tunnel interface to support IPv4-in-IPv6 encapsulation. In CPE case, it has at least one IPv4 interface connected to IPv4 local network.

4over6 initiator should learn the 4over6 concentrator's IPv6 address beforehand. For example, if the initiator gets its IPv6 address by DHCPv6, it can get the 4over6 concentrator's IPv6 address through a DHCPv6 option[I-D.ietf-softwire-ds-lite-tunnel-option].

5.2.1. Host initiator

When the initiator is a direct-connected host, it assigns the allocated public IPv4 address to its tunnel interface. The host uses this address for IPv4 communication. If this address is allocated through DHCP, the host should support DHCPv4 over tunnel. After the allocation, the host periodically sends pinhole packet to the concentrator to install the address mapping and keep it alive.

For IPv4 data traffic, the host performs the IPv4-in-IPv6 encapsulation and decapsulation on the tunnel interface. When sending out an IPv4 packet, it performs the encapsulation, using the IPv6 address of the 4over6 concentrator as the IPv6 destination address, and its own IPv6 address as the IPv6 source address. The encapsulated packet will be forwarded to the IPv6 network. The decapsulation on 4over6 initiator is simple. When receiving an IPv4-in-IPv6 packet, the initiator just drops the IPv6 header, and hands it to upper layer.

5.2.2. CPE initiator

The CPE case is quite similar to the host initiator case. The CPE assign the allocated IPv4 address to its tunnel interface. The local IPv4 network won't take part in the public IPv4 allocation; instead, end hosts will use private IPv4 addresses, possibly allocated by the CPE. After the allocation, the CPE periodically sends pinhole packet to the concentrator to install the address mapping and keep it alive.

On data plan, the CPE can be viewed as a regular IPv4 NAT(using tunnel interface as the NAT outside interface) cascaded with a tunnel initiator. For IPv4 data packets received from the local network, the CPE translates these packets, using the tunnel interface address as the source address, and then encapsulates the translated packet into IPv6, using the concentrator's IPv6 address as the destination address, the CPE's IPv6 address as source address. For IPv6 data packet received from the IPv6 network, the CPE performs decapsulation and IPv4 public-to-private translation. As to the CPE itself, it uses the public, tunnel interface address to communicate with the

IPv4 Internet, and the private, IPv4 interface address to communicate with the local network.

5.3. 4over6 concentrator behavior

4over6 concentrator represents the IPv4-IPv6 border router working as the remote tunnel endpoint for 4over6 initiators, with its IPv6 interface connected to the IPv6 network, IPv4 interface connected to the IPv4 Internet, and a tunnel interface supporting IPv4-in-IPv6 encapsulation and decapsulation. There's no CGN on the 4over6 concentrator, it won't perform any translation function; instead, 4over6 concentrator maintains an IPv4-IPv6 address mapping table for IPv4 data encapsulation.

4over6 concentrator maintains the address mapping according to the initiators' demand. When receiving a pinhole packet from an initiator, the concentrator reads the IPv4 and IPv6 source addresses from the packet, install the mapping entry into the mapping table or renew it if it already exists. When the lifetime of a mapping entry expires, the concentrator deletes it from the table. So the initiator should send pinhole packet with an interval shorter than the lifetime of the mapping entry. The mapping entry is used to provide correct encapsulation destination address for concentrator encapsulation. As long as the entry exists in the table, the concentrator can encapsulate inbound IPv4 packets destined to the initiator, with the initiator's IPv6 address as IPv6 destination.

On the IPv6 side, 4over6 concentrator decapsulates IPv4-in-IPv6 packets coming from 4over6 initiators. It removes the IPv6 header of every IPv4-in-IPv6 packet and forwards it to the IPv4 Internet. On the IPv4 side, the concentrator encapsulates the IPv4 packets destined to 4over6 initiators. When performing the IPv4-in-IPv6 encapsulation, the concentrator uses its own IPv6 address as the IPv6 source address, uses the IPv4 destination address in the packet to look up IPv6 destination address in the address mapping table. After the encapsulation, the concentrator sends the IPv6 packet on its IPv6 interface to reach an initiator.

The 4over6 concentrator, or its upstream router should advertise the IPv4 prefix which contains the IPv4 addresses of 4over6 users to the IPv4 side, in order to make these initiators reachable on IPv4 Internet.

Since the concentrator has to maintain the IPv4-IPv6 address mapping table, the concentrator is stateful in IP level. Note that this table will be much smaller than a CGN table, as there is no port information involved.

6. Technical advantages

Public 4over6 provides a method for users in IPv6 network to communicate with IPv4. In many scenarios, this can be viewed as an alternative to IPv6-IPv4 translation mechanisms which have well-known limitations described in [RFC4966] .

Since a 4over6 initiator uses a public IPv4 address, Public 4over6 supports full bidirectional communication between IPv4 Internet and hosts/IPv4 networks in IPv6 access network. In particular, it supports the servers in IPv6 network to provide IPv4 application service transparently.

Public 4over6 provides IPv4 access over IPv6 network while keeps IPv4-IPv6 addressing and routing separated. Therefore the ISP can manage the native IPv6 network independently without the influence of IPv4-over-IPv6 requirements, and also provision IPv4 in a flexible, on-demand way.

Public 4over6 supports dynamic reuse of a single IPv4 address between multiple subscribers based on their dynamic requirement of communicating with IPv4 Internet. A subscriber will request a public IPv4 address for a period of time only when it need to communicate with IPv4 Internet. Besides, in the CPE case, one public IPv4 address will be shared by the local network. So Public 4over6 can improve the reuse rate of IPv4 addresses.

Public 4over6 is suited for network users/ISPs which can still get/ provide public IPv4 addresses. Dual-stack lite is suited for network users/ISPs which can no longer get/provide public IPv4 addresses. By combining Public 4over6 and Dual-stack lite, the IPv4-over-IPv6 Hub and spoke problem can be well solved.

7. Acknowledgement

The authors would like to thank Alain Durand and Dan Wing for their valuable comments on this draft.

8. References

8.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC4925] Li, X., Dawkins, S., Ward, D., and A. Durand, "Softwire Problem Statement", RFC 4925, July 2007.

[RFC4966] Aoun, C. and E. Davies, "Reasons to Move the Network Address Translator - Protocol Translator (NAT-PT) to Historic Status", RFC 4966, July 2007.

[RFC5549] Le Faucheur, F. and E. Rosen, "Advertising IPv4 Network Layer Reachability Information with an IPv6 Next Hop", RFC 5549, May 2009.

[RFC5565] Wu, J., Cui, Y., Metz, C., and E. Rosen, "Softwire Mesh Framework", RFC 5565, June 2009.

8.2. Informative References

[I-D.cui-softwire-b4-translated-ds-lite]
Cui, Y., Wu, J., and D. Wu, "B4 translated DS-lite enable AFTR to serve more B4s", draft-cui-softwire-b4-translated-ds-lite-00 (work in progress), October 2010.

[I-D.cui-softwire-dhcp-over-tunnel]
Cui, Y., Wu, P., and J. Wu, "DHCPv4 Behavior over IP-IP tunnel", draft-cui-softwire-dhcp-over-tunnel-00 (work in progress), June 2011.

[I-D.ietf-pcp-base]
Wing, D., Cheshire, S., Boucadair, M., Penno, R., and P. Selkirk, "Port Control Protocol (PCP)", draft-ietf-pcp-base-13 (work in progress), July 2011.

[I-D.ietf-softwire-ds-lite-tunnel-option]
Hankins, D. and T. Mrugalski, "Dynamic Host Configuration Protocol for IPv6 (DHCPv6) Option for Dual- Stack Lite", draft-ietf-softwire-ds-lite-tunnel-option-10 (work in progress), March 2011.

[I-D.ietf-softwire-dual-stack-lite]
Durand, A., Droms, R., Woodyatt, J., and Y. Lee, "Dual-Stack Lite Broadband Deployments Following IPv4

Exhaustion", draft-ietf-softwire-dual-stack-lite-11 (work in progress), May 2011.

[I-D.sun-v6ops-laft6]
Sun, Q. and C. Xie, "LAFT6: Lightweight address family transition for IPv6", draft-sun-v6ops-laft6-01 (work in progress), March 2011.

Authors' Addresses

Yong Cui
Tsinghua University
Department of Computer Science, Tsinghua University
Beijing 100084
P.R.China

Phone: +86-10-6260-3059
Email: yong@csnet1.cs.tsinghua.edu.cn

Jianping Wu
Tsinghua University
Department of Computer Science, Tsinghua University
Beijing 100084
P.R.China

Phone: +86-10-6278-5983
Email: jianping@cernet.edu.cn

Peng Wu
Tsinghua University
Department of Computer Science, Tsinghua University
Beijing 100084
P.R.China

Phone: +86-10-6278-5822
Email: weapon@csnet1.cs.tsinghua.edu.cn

Chris Metz
Cisco Systems, Inc.
3700 Cisco Way
San Jose, CA 95134
USA

Email: chmetz@cisco.com

Olivier Vautrin
Juniper Networks
1194 N Mathilda Avenue
Sunnyvale, CA 94089
USA

Email: Olivier@juniper.net

Yiu L. Lee
Comcast
One Comcast Center
Philadelphia, PA 19103
USA

Email: yiu_lee@cable.comcast.com

Internet Engineering Task Force
Internet-Draft
Intended status: Standards Track
Expires: September 15, 2011

R. Despres, Ed.
RD-IPtech
S. Matsushima
SoftBank
T. Murakami
IP Infusion
O. Troan
Cisco

March 14, 2011

IPv4 Residual Deployment across IPv6-Service networks (4rd)
ISP-NAT's made optional
draft-despres-intarea-4rd-01

Abstract

This document specifies an automatic tunneling mechanism for providing IPv4 connectivity service to end users over a service provider's IPv6 network. During the long transition period from IPv4 to IPv6-only, a service provider's network will have to support IPv6, but will also have to maintain some IPv4 connectivity for a number of customers, for both outgoing and incoming connections, and for both exclusive and shared IPv4 addresses. The 4rd solution (IPv4 Residual Deployment) is designed as a lightweight solution for this.

In some scenarios, 4rd can dispense ISPs from supporting any NAT in their networks. In some others it can be used in parallel with NAT-based solutions such as DS-lite and/or NAT64/DNS4.

Table of Contents

1. Introduction

During the transition period from IPv4 to IPv6 Internet Service Providers (ISP's), will deploy networks that are IPv6 only. Some of them will do so while they still have to offer IPv4 connectivity. The IPv4 service can be one or multiple IPv4 addresses per end-user, or it can be an IPv4 address shared among multiple end-users.

In this document, Internet Service Provider is used as a generic term. It includes DSL or Broadband service providers, mobile operators, and private operators of networks of any sizes.

4rd (IPv4 Residual Deployment) is a generic lightweight solution for providing IPv4 connectivity across an IPv6 only infrastructure. As such, it is the reverse of 6rd (IPv6 Rapid Deployment) whose purpose is to rapidly introduce native IPv6 connectivity across an IPv4 network. It applies the same principles of automatic tunneling, an stateless address mappings between IPv4 and IPv6.

On the tradeoff scale between efficiency of address sharing ratios and simplicity, 4rd is on the side of design and operational simplicity.

The 4rd mechanism tunnels IPv4 over IPv6 using an algorithmic mapping from IPv4 addresses or IPv4 addresses and ports to the IPv6 addresses used as tunnel endpoints. Depending on ISP constraints and policies, 4rd can be used either standalone, with NAT44's in CE's but no NAT in ISP networks, or can co-exist with other mechanisms in the network on NAT's like DS-lite [I-D.ietf-softwire-dual-stack-lite] or NAT64/DNS64 [I-D.ietf-behave-v6v4-xlate-stateful] [I-D.ietf-behave-dns64].

2. Requirements Language

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 [RFC2119].

3. Terminology

4rd domain (Domain): an IPv6 routing network operated by an ISP and comprising one or several 4rd BR's having the same set of parameters. It offers to its 4rd-capable CE's global IPv4 connectivity, both outgoing and incoming, and with exclusive or shared IPv4 addresses.

4rd Border Relay (BR): A 4rd-capable router managed by the service provider at the edge of a 4rd domain. A BR has an IPv6-enabled interface connected to the ISP network, and an IPv4 virtual interface acting as an endpoint for the automatic 4rd tunnel. This tunnel (IPv4 in IPv6) is between the BR and all CE's of the Domain.

4rd Customer Edge (CE): A node at the border between a customer network and the 4rd domain. This node has an IPv6 interface connected to the ISP network, and a virtual IPv4 interface acting as the end-point of the automatic 4rd tunnel. This tunnel (IPv4 in IPv6) is between the CE and all other CE's and all BR's of the Domain. It may be a host, a router, or both.

CE IPv6 prefix: The IPv6 prefix assigned to a CE by other means than 4rd itself, and used by 4rd to derive a CE 4rd prefix.

CE IPv6 address: In the context of 4rd, the IPv6 address used to reach a CE from other CE's and from BR's. A CE typically has another IPv6 address, assigned to it at its IPv6 interface without relationship with 6rd.

CE 4rd prefix: The 4rd prefix of the CE. It is derived from the CE IPv6 prefix by a mapping rule according to Section 4.3. Depending on its length, it is an IPv4 prefix, an IPv4 address, or a shared IPv4 address followed by a Port-set ID (Section 4.3.2).

Port-set ID: In a CE 4rd prefix longer than 32 bits, bits that follow the first 32. It algorithmically identifies a set of ports exclusively assigned to the CE. As specified in Section 4.3.3, the set can comprise up to 4 disjoint port ranges.

Domain IPv6 prefix: An IPv6 prefix assigned by an ISP to a 4rd domain.

Domain 4rd prefix: A 4rd prefix assigned by an ISP to the 4rd domain. In typical operator applications, it is an IPv4 prefix. In a residential site in which an already shared IPv4 address has to be shared even more among several hosts, it may have more than 32 bits.

CE index: For a CE, the field that is common to its CE IPv6 prefix and its CE 4rd prefix. In the former, it follows the Domain IPv6 prefix. In the latter, it follows the Domain 4rd prefix.

4. Protocol Specification

4.1. General Principles

The principle of the 4rd protocol is that IPv4 packets, or in case of shared IPv4 addresses IPv4 datagrams, traverse a 4rd domain by means of automatic IPv4 in IPv6 tunnels. IPv6 addresses of destination tunnel endpoints are statelessly derived from IPv4 destinations, based on some mapping rule parameters, in such a way that tunnels between CE's follow direct IPv6 paths (i.e. without having to go via BR's). IPv4 destinations used for these mappings are either IPv4 addresses alone or IPv4 addresses + ports depending on whether global addresses assigned to CE's are exclusive or shared.

BR's and CE's MAY have the detailed behaviors specified in the following sections. Different behaviors are however permitted, but they MUST be equivalent as far as exchanged packets are concerned.

4.2. Mapping-Rule Parameters

Both CE's and BR's have to know the BR IPv6 address of their domain as well as, for each mapping rule, the following parameters:

- Domain IPv6 prefix
- Domain 4rd prefix
- IPv6-prefix length
- Domain IPv6 suffix (optional - default ::/0)

4.3. Mapping Rules

4.3.1. From a CE IPv6 Prefix to a CE 4rd Prefix

A 4rd mapping rule establishes a 1:1 mapping between CE IPv6 prefixes and CE 4rd prefixes.

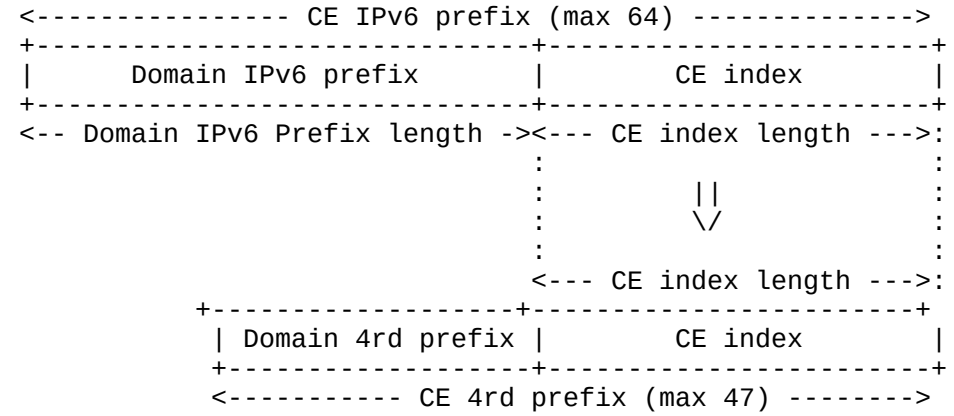


Figure 1: From a CE IPv6 Prefix to a CE 4rd Prefix

A CE derives its CE 4rd prefix from the IPv6 prefix it has been delegated on the IPv6 network, using for this parameters of the applicable mapping rule. If the domain has several mapping rules, that which applies is that whose Domain IPv6 prefix is at the beginning of the CE IPv6 prefix. As shown in Figure 1, the CE 4rd prefix is made of the Domain 4rd prefix followed by the CE index, where the CE index is the remainder of the CE IPv6 prefix after the Domain IPv6 prefix (the length of the Domain IPv6 prefix is defined by the mapping rule).

4.3.2. From a CE 4rd Prefix to a Port-set ID

Depending on its length, a CE 4rd prefix is either an IPv4 prefix, a full IPv4 address, or a shared IPv4 address followed by a Port-set ID (Figure 2). If it includes a port set ID, this ID specifies which ports are assigned to the the CE for its exclusive use (Section 4.3.3).

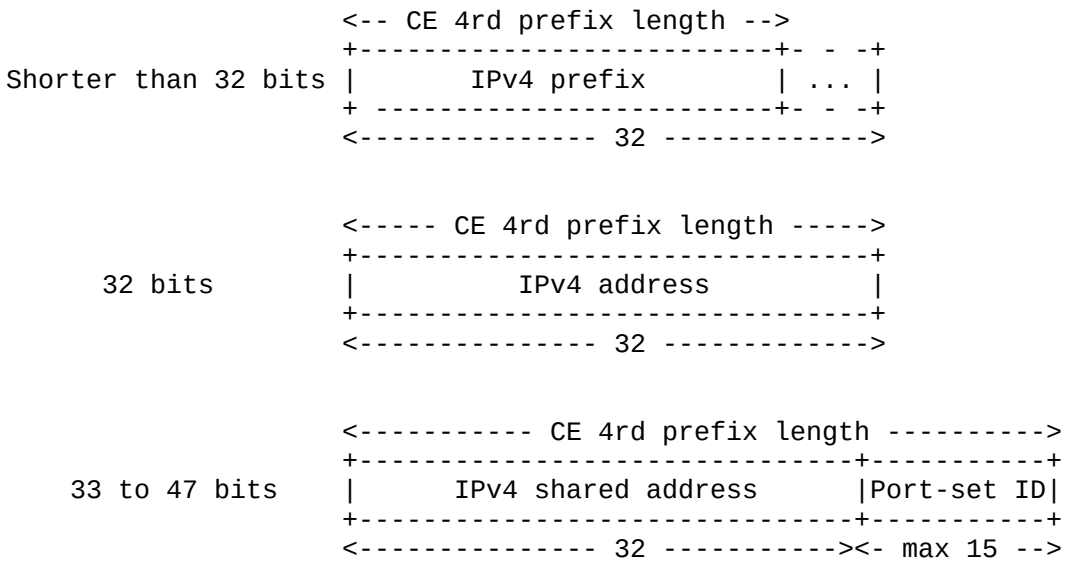


Figure 2: Variants of CE 4rd prefixes

4.3.3. From a Port-Set ID to a Port Set

Each value of a Port-set ID specifies which ports can be used by any protocol whose header format starts with source and destination ports (UDP, TCP, SCTP, etc.). Design constraint of the algorithm are the following:

"Fairness with respect to special-value ports"
No port-set must contain any port from 0 to 4095. (These ports, which have more value than others in OS's, are normally not used in dynamic port assignments to applications).

"Fairness with respect to the number of ports"
For a Port-set-ID's having the same length, all sets must have the same number of ports.

"Exhaustiveness"
For a any Port-set-ID length, the aggregate of port sets assigned for all values must include all ordinary-value ports (from 4,096 to 16,384).

If the Port-set ID has 1 to 12 bits, the set comprises 4 port ranges. As shown in Figure 3, each port range is defined by its port prefix, made of a range-specific "head" followed by the Port-set ID. Head values are in binary 1, 01, 001, and 0001. They are chosen to exclude ports 0-4095 and only them.

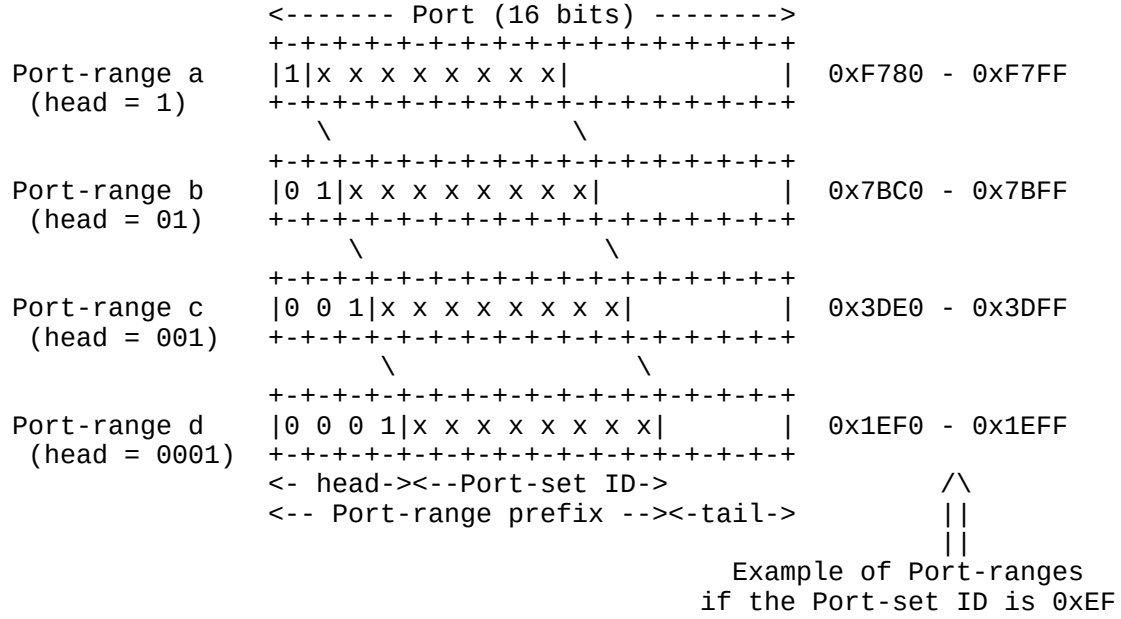


Figure 3: From Port-set ID to Port ranges

In the Port-set ID has 13 bits, only the 3 port ranges are assigned, having heads 1, 01, and 001. If it has 14 bits, only the 2 port ranges having heads 1 and 01 are assigned. If it has 15 bits, only the port range having head 1 is assigned. (In these three cases, the smallest port range has only one element).

NOTE: The port set assigned to a CE may be further subdivided by the CE among several functions such as the following: (1) an IPv4 NAPT (possibly configurable to do port forwarding, and possibly doing dynamic port assignments to hosts with UPnP and/or NAT-PMP); (2) an API for applications in the CE that need dynamic port assignments; (3) a new 4rd BR which assigns to its CE’s subsets of its own port

set. How to chose among these functions and/or combine them is beyond the scope of this specification. Readers are referred to documents dealing with operational applicability in diverse environments, e.g. [draft-sun-intarea-4rd-applicability] prepared in parallel of this one.

4.3.4. From an IPv4 Address or IPv4 address + Port to a CE IPv6 address

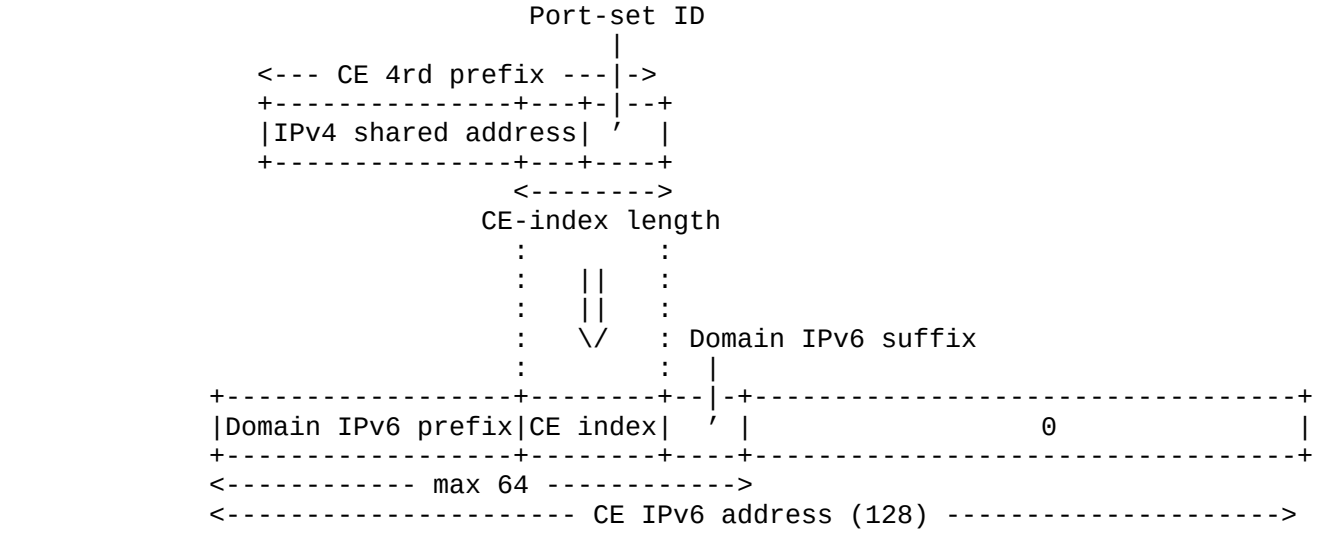


Figure 4: From 4rd Prefix to IPv6 address (shared IPv4 address case)

In order to find whether a CE IPv6 address can be derived from an IPv4 address, or an IPv6 address + a port, a mapping rule has to be found that matches the IPv4 information:

- If a mapping rule has a length L of CE IPv4 prefixes which does not exceed 32 bits, there is a match if the IPv4 address starts with the Domain 4rd prefix. The CE 4rd prefix is then the first L bits of the IPv4 address.

- If a mapping rule has a length L of CE IPv4 prefixes which exceeds 32 bits, the match can only be found with the IPv4 address and the port. For this, the port is examined to determine which port-range head it starts with: 1, 01,001, or 0001. The N bits that follow this head are taken as Port-set ID, where N is the length of Port set ID of the mapping rule. The CE 4rd prefix is then made of the IPv4 address followed by the Port-set ID.

  If a match has been found, the CE IPv6 prefix is then made of the

Domain IPv6 prefix followed by bits of the CE 4rd prefix that follow the Domain 4rd prefix, followed by the Domain IPv6 prefix of the mapping rule if there is one, and followed by 0's up to 128 bits to make a complete IPv6 address [RFC4291]. Figure 4 illustrates this process in the case of a shared IPv4 address.

4.4. Encapsulation and Fragmentation Considerations

For 4rd domain traversal, IPv4 packets are encapsulated in IPv6 packets whose Next header is set to 4 (i.e. IPv4). If fragmentation of IPv6 packets is needed, it is performed according to [RFC2460], and as illustrated in Figure 5. Absent more specific information, the path MTU of a 4rd Domain has to be set to 1280 [RFC2460].

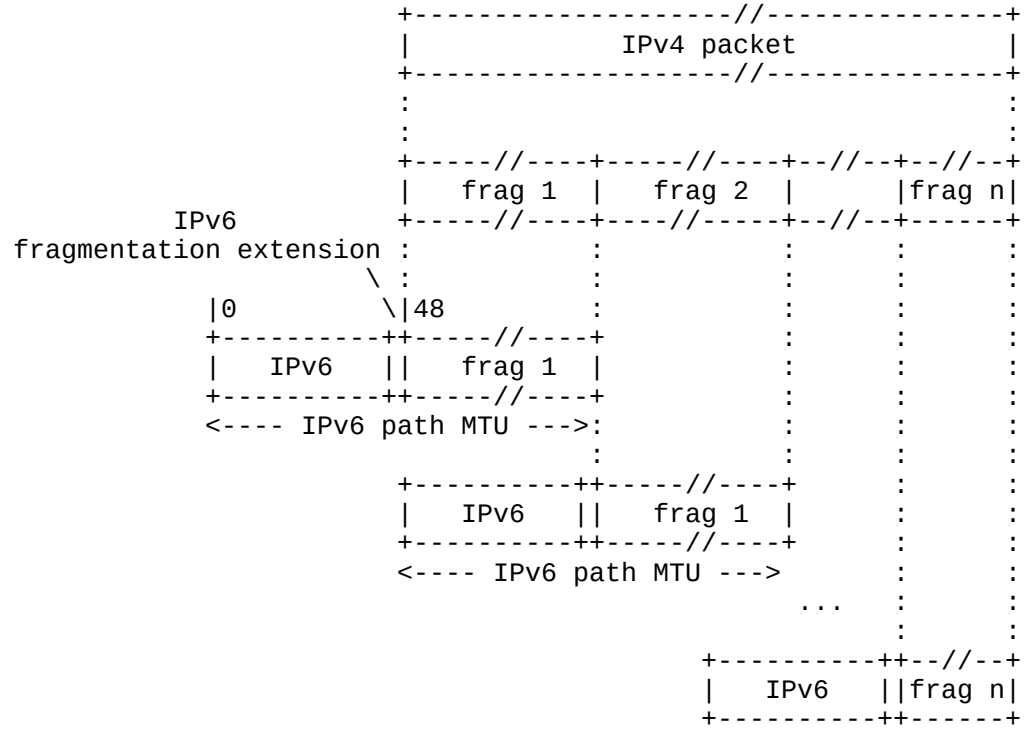


Figure 5: Fragmentation of long IPv4 packets for Domain Traversal

In domains where IPv4 addresses are not shared, IPv6 destinations are derived from IPv4 addresses alone. Thus, each IPv4 packet can be encapsulated and decapsulated independently of each other. 4rd processing is completely stateless.

On the other hand, in domains where IPv4 addresses are shared, BR's and CE's can have to encapsulate IPv4 packets whose IPv6 destinations depend on destination ports. Precautions are needed, due to the fact that the destination port of a fragmented datagram is available only in its first fragment. A sufficient precaution consists in reassembling each datagram received in multiple packets, and to treat it as though it would have been received in single packet. This function is such that 4rd is in this case stateful at the IP layer. (This is common with DS-lite and NAT64/DNS64 which, in addition, are stateful at the transport layer.) At Domain entrance, this ensures that all pieces of all received IPv4 datagrams go to the right IPv6 destinations.

Another peculiarity of shared IPv4 addresses is that, without precaution, a destination could simultaneously receive from different sources fragmented datagrams that have the same Datagram ID (the Identification field of [RFC0791]). This would disturb the reassembly process. To eliminate this risk, BR's and CE's SHOULD, in datagrams they receive from shared-IPv4-address CE's, replace received Datagram ID's by new ones. New values SHOULD be generated as though these datagrams would have been created locally (and with due respect of [RFC0791]). Note that replacing a Datagram ID in an IPv4 header implies an update of its Header-checksum field, by adding to it the one's complement difference between the old and the new values.

4.5. BR and CE behaviors

4.5.1. Domains having only One Mapping rule

(a) BR reception of an IPv4 packet

Step 1 If the length of CE 4rd prefixes does not exceed 32 bits, the BR proceeds to step 2. Otherwise, and unless the packet contains a complete IPv4 datagram, IPv4 datagram reassembly is performed. If a complete datagram is available, the BR proceeds to step 2 as though the datagram had been received in a single packet.

Step 2 The BR checks that the IPv4 source doesn't start with the Domain 4rd prefix, and that a CE IPv6 address is successfully derived from the IPv4 destination. In case of success, the packet is encapsulated and forwarded to this CE IPv6 address via the IPv6 interface.

(b) BR reception of an IPv6 packet

The BR checks that a CE IPv6 address is successfully derived from the source of the IPv4 encapsulated packet, and that the source address of the encapsulating packet is equal to it. In case of success: (1) if the length of CE 4rd prefixes exceeds 32 bits, the Datagram ID of the packet is replaced by a locally generated one; (2) the IPv4 packet is forwarded via the IPv4 interface.

(c) CE reception of an IPv4 packet

Step 1 If the CE 4rd prefix of the CE does not exceed 32 bits and the IPv4 destination address starts with the Domain 4rd prefix, the CE proceeds to step 2. Otherwise, and unless the packet contains a complete IPv4 datagram, IPv4 datagram reassembly is performed. If a complete datagram is available, the BR proceeds to step 2 as though the datagram had been received in a single packet.

Step 2 The CE tries tries to derive a CE IPv6 address from the IPv4 destination. It then encapsulates the IPv4 packet into an IPv6 packet whose destination is this CE IPv6 address, if one is obtained, or the BR IPv6 address otherwise.

(d) CE reception of an IPv6 packet (reassembled if applicable)

The CE checks that a CE IPv6 address is successfully derived from the source of the IPv4 encapsulated packet, AND that it is equal to the source address of the encapsulating packet. In case of success: (1) if the length of CE 4rd prefixes exceeds 32 bits, the Datagram ID of the packet is replaced by a locally generated one; (2) the IPv4 packet is forwarded via the IPv4 interface.

4.5.2. Domains having Multiple Mapping Rules

Some ISP will want to use 4rd in networks having several Domain 4rd prefixes, an/or several Domain IPv6 prefixes, and/or assigning CE 4rd prefixes of different lengths. For this several mapping rules are needed.

A first possibility consists in establishing several 4rd domains, each on having a single mapping rule. In this case, paths between CE's belonging to different 4rd domains go from one domain to the other in IPv4, and cross two BR's.

A second possibility permits direct IPv6 paths between CE's by supporting several mapping rules in a single domain, as described in this section. At time of writing, whether this will be in the 4rd specification a MAY, a SHOULD, or a MUST, remains an open question.

(a) BR reception of an IPv4 packet

Step 1 If a mapping rule whose length of CE 4rd prefixes does not exceed 32 bits applies to the IPv4 destination, the BR proceeds to step 2. Otherwise, and unless the packet contains a complete IPv4 datagram, IPv4 datagram reassembly is performed. If a complete datagram is available, the BR then proceeds to step 2 as though the datagram had been received in a single packet.

Step 2 The BR checks that the IPv4 source doesn't start with the Domain 4rd prefix of any rule. In case of success, the packet is encapsulated and forwarded to this CE IPv6 address via the IPv6 interface.

(b) BR reception of an IPv6 packet (reassembled if applicable)

The BR checks that a CE IPv6 address is successfully derived from the source of the IPv4 encapsulated packet, and that the source address of the encapsulating packet is equal to it. In case of success, the BR tries to derive a CE IPv6 address from the destination of the encapsulated packet. In case of success: (1) if the source CE 4rd prefix exceeds 32 bits, the Datagram ID of the packet is replaced by a locally generated one; (2) the encapsulating packet is retransmitted via the IPv6 interface with this CE IPv6 address as destination (and the BR IPv6 address as source address); in case of failure, the IPv4 packet is decapsulated and forwarded via the IPv4 interface.

(c) CE reception of an IPv4 packet

Step 1 If the CE 4rd prefix of the CE does not exceed 32 bits, and a mapping rule whose length of CE 4rd prefixes does not exceed 32 bits applies to the IPv4 destination, the CE proceeds to step 2. Otherwise, and unless the packet contains a complete IPv4 datagram, IPv4 datagram reassembly is performed. If a complete datagram is

available, the BR then proceeds to step 2 as though the datagram had been received in a single packet.

Step 2 The CE tries to derive a CE IPv6 address from the IPv4 destination. It then encapsulates the IPv4 packet into an IPv6 packet whose destination is this CE IPv6 address, if one is obtained, or the BR IPv6 address otherwise.

(d) CE reception of an IPv6 packet (reassembled if applicable)

The CE checks that a CE IPv6 address is successfully derived from the source of the IPv4 encapsulated packet, and that it is equal to the source address of the encapsulating packet. In case of success: (1) if the source CE 4rd prefix exceeds 32 bits, the Datagram ID of the packet is replaced by a locally generated one; (2) the IPv4 packet is decapsulated and forwarded via the IPv4 interface.

NOTE: With consistency check made between encapsulated and encapsulating sources in BR's and CE's when they received tunneled packets, no CE can forward an invalid IPv4 source address, or address plus port, and have it forwarded at by the egress BR or CE. Yet, if before tunneling a packet, a CE makes an additional check that the IPv4 source is consistent with the CE IPv6 address, it can discard invalid packets earlier than by leaving it to the egress BR or CE. At time of writing, whether this test can remain a MAY, or might require a SHOULD or a MUST remains an open question.

5. 4rd Configuration

A CE can acquire 4rd parameters of its 4rd domain in various ways: manual configuration by an administrator, software download by the ISP, a new DHCPv6 option, etc. This document describes how to configure the necessary parameters via a single DHCPv6 option. A CE that allows IPv6 configuration by DHCPv6 SHOULD implement this option. Other configuration and management methods, MAY use the format described by this option for consistency and convenience of implementation on CEs that support multiple configuration methods.

The format of Figure 6 is proposed for the DCHPv6 option. It is chosen to permit multiple mapping rules:

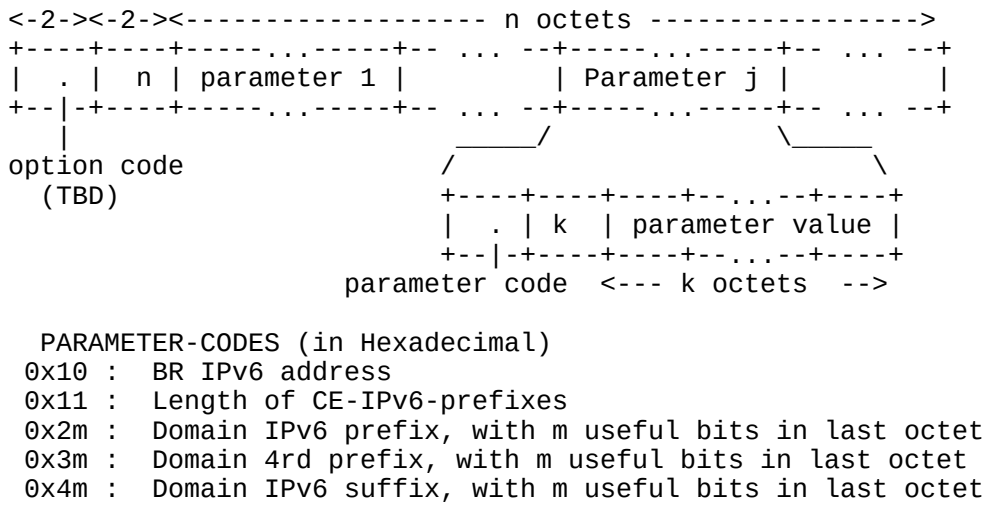


Figure 6: 4rd DHCPv6 option

In the parameter list the BR IPv6 address is first, followed by parameters of each rule. For each rule, the order is <Domain IPv6 prefix, Domain IPv4 prefix, Length of CE IPv6 prefixes, Domain IPv6 suffix (optional)>.

6. Security considerations

Spoofing attacks

With consistency checks between IPv4 and IPv6 sources that are performed on IPv4/IPv6 packets received by BR's and CE's (Section 4.5), 4rd does not introduce any opportunity for spoofing attack that would not pre-exist in IPv6.

Denial-of-service attacks

In 4rd domains where IPv4 addresses are shared, the fact that IPv4 datagram reassembly may be necessary introduces an opportunity for DOS attacks (Section 4.4). This is inherent to address sharing, and is common with other address sharing approaches such as DS-lite and NAT64/DNS64.

The best protection against such attacks is to accelerate IPv6 enablement in both clients and servers so that, where 4rd is supported, it is less and less used.

Routing-loop attacks

Routing-loop attacks that may exist in some automatic-tunneling scenarios are documented in [I-D.ietf-v6ops-tunnel-loops]. They cannot exist with 4rd because each BRs checks that the IPv6 source address of a received IPv6 packet is a CE address (Section 4.5.1 (b) and Section 4.5.2 (b) />).

Attacks facilitated by restricted port sets

From hosts that are not subject to ingress filtering of [RFC2827], some attacks are possible by intervening with faked packets during ongoing transport connections ([RFC4953], [RFC5961], [RFC6056]. These attacks, that have mitigations of their own are easier with hosts that only use restricted port sets (they depend on guessing which ports are currently used by target hosts). To avoid using restricted port sets, the easiest approach consists in increasing the proportion of connections that are IPv6, i.e. using unrestricted port sets.

7. IANA Considerations

IANA is requested to assign a DHCPv6 option number for 4rd (Section 5).

8. Acknowledgments

The authors wish to thank Mark Townsley for his active encouragements to pursue the 4rd approach since it was first introduced in [I-D.despres-softwire-sam]. Questions raised by Wojciech Dec have been useful to clarify explanations. Olivier Vautrin, who independently proposed a similar approach with the same acronym deserves special recognition. Particular gratitude is due to decision makers of the Japan ISP’s that have announced actual 4rd deployment projects (www.ietf.org/mail-archive/web/v6ops/current/msg05247).

9. References

9.1. Normative References

[RFC0791] Postel, J., "Internet Protocol", STD 5, RFC 791, September 1981.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate

Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC2460] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", RFC 2460, December 1998.

[RFC3315] Droms, R., Bound, J., Volz, B., Lemon, T., Perkins, C., and M. Carney, "Dynamic Host Configuration Protocol for IPv6 (DHCPv6)", RFC 3315, July 2003.

[RFC4291] Hinden, R. and S. Deering, "IP Version 6 Addressing Architecture", RFC 4291, February 2006.

9.2. Informative References

[I-D.despres-softwire-sam]
Despres, R., "Stateless Address Mapping (SAM) - a Simplified Mesh-Softwire Model", draft-despres-softwire-sam-01 (work in progress), July 2010.

[I-D.ietf-behave-dns64]
Bagnulo, M., Sullivan, A., Matthews, P., and I. Beijnum, "DNS64: DNS extensions for Network Address Translation from IPv6 Clients to IPv4 Servers", draft-ietf-behave-dns64-11 (work in progress), October 2010.

[I-D.ietf-behave-v6v4-xlate-stateful]
Bagnulo, M., Matthews, P., and I. Beijnum, "Stateful NAT64: Network Address and Protocol Translation from IPv6 Clients to IPv4 Servers", draft-ietf-behave-v6v4-xlate-stateful-12 (work in progress), July 2010.

[I-D.ietf-softwire-dual-stack-lite]
Durand, A., Droms, R., Woodyatt, J., and Y. Lee, "Dual-Stack Lite Broadband Deployments Following IPv4 Exhaustion", draft-ietf-softwire-dual-stack-lite-07 (work in progress), March 2011.

[I-D.ietf-v6ops-tunnel-loops]
Nakibly, G. and F. Templin, "Routing Loop Attack using IPv6 Automatic Tunnels: Problem Statement and Proposed Mitigations", draft-ietf-v6ops-tunnel-loops-03 (work in progress), February 2011.

[RFC2827] Ferguson, P. and D. Senie, "Network Ingress Filtering: Defeating Denial of Service Attacks which employ IP Source

Address Spoofing", BCP 38, RFC 2827, May 2000.

[RFC4953] Touch, J., "Defending TCP Against Spoofing Attacks", RFC 4953, July 2007.

[RFC5569] Despres, R., "IPv6 Rapid Deployment on IPv4 Infrastructures (6rd)", RFC 5569, January 2010.

[RFC5961] Ramaiah, A., Stewart, R., and M. Dalal, "Improving TCP's Robustness to Blind In-Window Attacks", RFC 5961, August 2010.

[RFC5969] Townsley, W. and O. Troan, "IPv6 Rapid Deployment on IPv4 Infrastructures (6rd) -- Protocol Specification", RFC 5969, August 2010.

[RFC6056] Larsen, M. and F. Gont, "Recommendations for Transport-Protocol Port Randomization", BCP 156, RFC 6056, January 2011.

Authors' Addresses

Remi Despres (editor)
RD-IPtech
3 rue du President Wilson
Levallois,
France

Email: remi.despres@free.fr

Satoru Matsushima
SoftBank
1-9-1 Higashi-Shinbashi, Munato-ku
Tokyo
Japan

Email: satoru.matsushima@tm.softbank.co.jp

Tetsuya Murakami
IP Infusion
1188 East Arques Avenue
Sunnyvale
USA

Email: tetsuya@ipinfusion.com

Ole Troan
Cisco
Bergen, Norway
France

Email: ot@cisco.com

softwires WG
Internet-Draft
Intended status: Informational
Expires: September 15, 2011

X. Xu
D. Guo
Huawei Technologies
O. Troan
W. Townsley
Cisco
March 14, 2011

IPv6 Host Configuration in 6rd
draft-guo-softwire-6rd-ipv6-config-02.txt

Abstract

The 6rd [RFC5969] linktype does not support IPv6 link-local addressing, multicast and 6rd nodes are off-link from each other. The host configuration protocol DHCPv6 [RFC3315] relies on link-local addressing and multicast to function. This document specifies how DHCPv6 can be used across a 6rd link.

Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on September 15, 2011.

include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

1. Introduction

IPv6 rapid deployment on IPv4 infrastructures (6rd) [RFC5969] enables a service provider to rapidly deploy IPv6 service to residential sites via stateless tunneling across its existing IPv4 network.

With 6rd, a 6rd CE can provide address assignments to hosts on the LAN side, but there is no provision for providing other configuration information to hosts on the LAN.

If only DNS configuration is required on IPv6-only hosts, DNS Proxy [RFC5625] mechanism implemented on the 6rd CE would be enough. Otherwise, stateless DHCPv6 [RFC3736] SHOULD be supported in 6rd for IPv6 hosts to obtain other configuration information besides DNS.

As specified in the DHCPv6 specification [RFC3315], "...The client MUST use a link-local address assigned to the interface for which it is requesting configuration information as the source address in the header of the IP datagram." A DHCPv6 client uses the All_DHCP_Servers_or_Relays IPv6 multicast address as the destination address of requests it sends. Link-local addresses are not supported on 6rd links. 6rd as described in [RFC5969] does not support multicast.

This document describes how DHCPv6 service can be provided across a 6rd link.

2. 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 RFC 2119 [RFC2119].

3. DHCPv6 over 6rd links

There are two problems to be solved with regards to providing DHCPv6 service over a 6rd link:

- A DHCPv6 client uses an IPv6 link-local address as the source address when requesting configuration information [RFC3315]. Link-local addressing is not supported on an 6rd link.

- o A DHCPv6 client sends a request to the All_DHCP_Relay_Agent_and_Servers multicast address. 6rd as specified in [RFC5969] does not support IPv6 multicast.

The first problem can be solved by changing the DHCPv6 protocol to allow for a global address to be used as the source address in requests. Another solution that does not require protocol changes, is to send DHCPv6 requests via a local DHCPv6 relay on the 6rd CE.

The 6rd CE MUST support a local DHCPv6 client and relay. The DHCPv6 client running on the 6rd CE's virtual tunnel interface MUST send DHCPv6 messages through a local DHCPv6 relay that encapsulates the client message and forwards it to a DHCPv6 server or relay using one of the 6rd CE's global unicast addresses as the source address.

The 6rd CE DHCPv6 relay agent SHOULD use the 6rd BR IPv6 anycast address as the destination address, section 20 of [RFC3315]. If the 6rd link supports multicast [I-D.ietf-mboned-auto-multicast] the 6rd CE DHCPv6 relay MAY use the All_DHCP_Servers [RFC3315] as the destination address of Relay-forward messages.

The 6rd BRs in the 6rd domain must be configured as DHCPv6 relays or servers on their 6rd virtual interfaces.

The 6rd CE SHOULD behave according to [I-D.ietf-v6ops-ipv6-cpe-router]. In particular it operates a DHCPv6 client on the WAN side (6rd virtual) interface and as a DHCPv6 server on the LAN-side interface(s).

4. IANA Considerations

This specification does not require any IANA actions.

5. Security Considerations

There are no new security considerations pertaining to this document.

6. Acknowledgements

7. References

7.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC3315] Droms, R., Bound, J., Volz, B., Lemon, T., Perkins, C., and M. Carney, "Dynamic Host Configuration Protocol for IPv6 (DHCPv6)", RFC 3315, July 2003.

[RFC3736] Droms, R., "Stateless Dynamic Host Configuration Protocol (DHCP) Service for IPv6", RFC 3736, April 2004.

[RFC5969] Townsley, W. and O. Troan, "IPv6 Rapid Deployment on IPv4 Infrastructures (6rd) -- Protocol Specification", RFC 5969, August 2010.

7.2. Informative References

[I-D.ietf-mboned-auto-multicast] Thaler, D., Talwar, M., Aggarwal, A., Vicisano, L., and T. Pusateri, "Automatic IP Multicast Without Explicit Tunnels (AMT)", draft-ietf-mboned-auto-multicast-10 (work in progress), March 2010.

[I-D.ietf-v6ops-ipv6-cpe-router] Singh, H., Beebee, W., Donley, C., Stark, B., and O. Troan, "Basic Requirements for IPv6 Customer Edge Routers", draft-ietf-v6ops-ipv6-cpe-router-09 (work in progress), December 2010.

[RFC5625] Bellis, R., "DNS Proxy Implementation Guidelines", BCP 152, RFC 5625, August 2009.

Authors' Addresses

Xiaohu Xu
Huawei Technologies
No.3 Xinxi Rd., Shang-Di Information Industry Base
Beijing, Hai-Dian District 100085
P.R. China

Phone: +86 10 82882573
Email: xuxh@huawei.com

Dayong Guo
Huawei Technologies
No.3 Xinxi Rd., Shang-Di Information Industry Base
Beijing, Hai-Dian District 100085
P.R. China

Phone: +86-10-82882578
Email: guoseu@huawei.com

Ole Troan
Cisco
Oslo,
Norway

Email: ot@cisco.com

Mark Townsley
Cisco
Paris,
France

Email: mark@townsley.net

softwire
Internet-Draft
Intended status: Standards Track
Expires: September 11, 2011

D. Miles
Alcatel-Lucent
W. Dec
Cisco Systems
J. Bristow
Swisscom Schweiz AG
R. Maglione
Telecom Italia
March 10, 2011

Forcerenew Nonce Authentication
draft-ietf-dhc-forcerenew-nonce-01

Abstract

DHCP Forcerenew allows for the reconfiguration of a single host by forcing the DHCP client into a Renew state on a trigger from the DHCP server. In Forcerenew Nonce Authentication the server exchanges a nonce with the client on the initial DHCP ACK that is used for subsequent validation of a Forcerenew message.

Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on September 11, 2011.

Copyright Notice

Table of Contents

1. Introduction

The DHCP Reconfigure Extension defined in [RFC3203] is a useful mechanism allowing dynamic reconfiguration of a single host triggered by the DHCP server. Its application is currently limited by a requirement that FORCERENEW message is always authenticated using procedures as described in [RFC3118]. Authentication for DHCP [RFC3118] is mandatory for Forcerenew, however as it is currently defined [RFC3118] requires distribution of constant token or shared-secret out-of-band to DHCP clients. The mandatory authentication was originally motivated by a legitimate security concern whereby in some network environments a FORCERENEW message can be spoofed. However, in some networks native security mechanisms already provide sufficient protection against spoofing of DHCP traffic. An example of such network is a DSL Forum TR-101 [TR-101] compliant access network. In such environments the mandatory coupling between FORCERENEW and DHCP Authentication [RFC3118] can be relaxed. This document defines extensions to Authentication for DHCP(v4) Messages [RFC3118] to create a new authentication protocol for DHCPv4 Forcerenew [RFC3203] messages; this method does not require out-of-band key distribution to DHCP clients. The Forcerenew Nonce is exchanged between server and client on initial DHCP ACK and is used for verification of any subsequent Forcerenew message.

2. Requirements Language

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 [RFC2119].

3. Message authentication

The FORCERENEW message must be authenticated using either [RFC3118] or the proposed Forcerenew Nonce Authentication protocol.

3.1. Forcerenew Nonce Authentication

The Forcerenew nonce authentication protocol provides protection against misconfiguration of a client caused by a Forcerenew message sent by a malicious DHCP server. In this protocol, a DHCP server sends a Forcerenew nonce to the client in the initial exchange of DHCP messages. The client records the Forcerenew nonce for use in authenticating subsequent Forcerenew messages from that server. The server then includes an HMAC computed from the Forcerenew nonce in subsequent Forcerenew messages.

Both the Forcerenew nonce sent from the server to the client and the HMAC in subsequent Forcerenew messages are carried as the Authentication information in a DHCP Authentication option. The format of the Authentication information is defined in the following section.

The Forcerenew nonce protocol is used (initiated by the server) only if the client and server are not using any other authentication protocol and the client and server have negotiated to use the Forcerenew Nonce Authentication protocol.

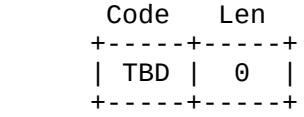
3.1.1. Forcerenew Nonce Protocol Capability Option

A DHCP client indicates DHCP Forcerenew Nonce Protocol capability by including a FORCERENEW_NONCE_CAPABLE(<TBD>) option in DHCP Discover and Request messages sent to the server.

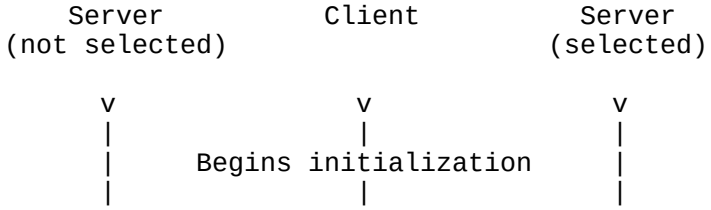
A DHCP server that does not support Forcerenew Nonce Protocol authentication should ignore the FORCERENEW_NONCE_CAPABLE(<TBD>) option. A DHCP server indicates DHCP Forcerenew Nonce Protocol preference by including a FORCERENEW_NONCE_CAPABLE(<TBD>) option in any DHCP Offer messages sent to the client.

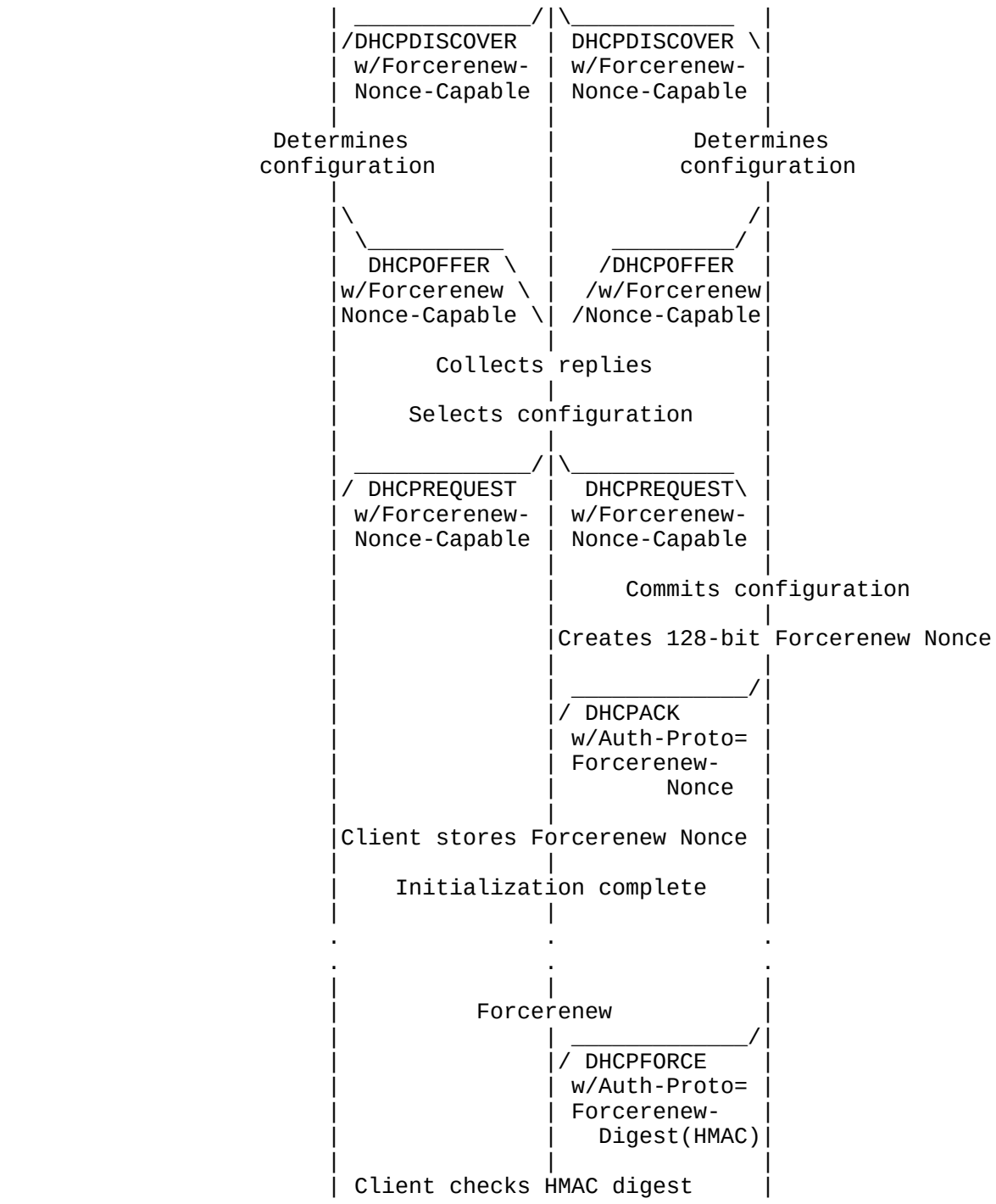
A DHCP client MUST NOT send DHCP messages with authentication options where the protocol value is Forcerenew Nonce Authentication(<TBD>).

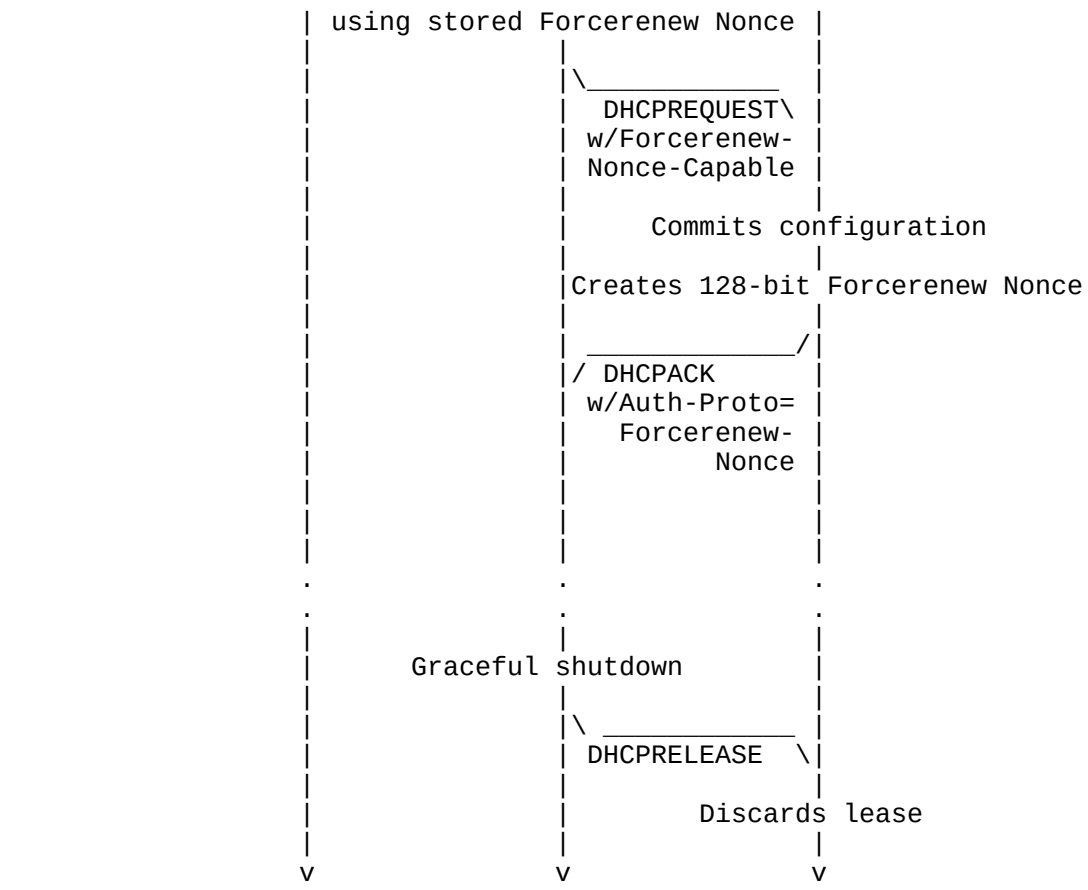
The FORCERENEW_NONCE_CAPABLE option is a zero length option with code of <TDB> and format as follows:



The client would indicate that it supports the functionality by inserting the FORCERENEW_NONCE_CAPABLE option in the DHCP Discover and Request messages. If the server supports Forcerenew nonce authentication and is configured to require Forcerenew nonce authentication, it will insert the FORCERENEW_NONCE_CAPABLE option in the DHCP Offer message.




DHCPDISCOVER w/Forcerenew-Nonce-Capable
DHCPDISCOVER w/Forcerenew-Nonce-Capable
Determines configuration
Determines configuration
DHCPOFFER w/Forcerenew Nonce-Capable
DHCPOFFER w/Forcerenew Nonce-Capable
Collects replies
Selects configuration
DHCPREQUEST w/Forcerenew-Nonce-Capable
DHCPREQUEST w/Forcerenew-Nonce-Capable
Commits configuration
Creates 128-bit Forcerenew Nonce
DHCPACK w/Auth-Proto= Forcerenew-Nonce
Client stores Forcerenew Nonce
Initialization complete
Forcerenew
DHCPFORCE w/Auth-Proto= Forcerenew-Digest(HMAC)
Client checks HMAC digest



3.1.2. Forcerenew Nonce Protocol

[RFC3118] defined an extensible DHCPv4 authentication option which supports multiple protocols. The Forcerenew Nonce Protocol makes use of the DHCP authentication option defined in [RFC3118] re-using the option format.

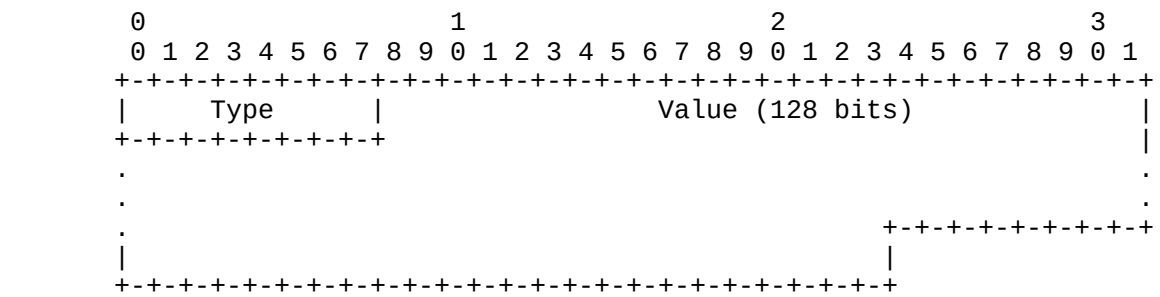
The following fields are set in an DHCP authentication option for the Forcerenew Nonce Authentication Protocol:

protocol <TBD (IANA)>

algorithm 1

RDM 0

The format of the Authentication information for the Forcerenew Nonce Authentication Protocol is:



Type Type of data in Value field carried in this option:

1 Forcerenew Nonce value (used in ACK message)

2 HMAC-MD5 digest of the message (FORCERENEW message)

Value Data as defined by field

3.1.3. Server considerations for Forcerenew Nonce Authentication

The use of Forcerenew Nonce Protocol is dependent on the client indicating its capability through the FORCERENEW_NONCE_CAPABLE(<TBD>) DHCP option in any DHCP Discover or Request messages. The DHCP Discovery or Request message from the client MUST contain the FORCERENEW_NONCE_CAPABLE(<TBD>) option if the Forcerenew Nonce Protocol is to be used by the server. The absence of the FORCERENEW_NONCE_CAPABLE(<TBD>) option indicates to the server that the Forcerenew Nonce Authentication protocol is not supported and thus the server MUST NOT include a Forcerenew Nonce Protocol Authentication option in the DHCP Ack.

The server indicates its support of the Forcerenew Nonce Protocol authentication by including the DHCP FORCERENEW_NONCE_CAPABLE(<TBDP>) option in the DHCP Offer message. The server SHOULD NOT include this option unless the client has indicated its capability in a DHCP Discovery message . The presence of the FORCERENEW_NONCE_CAPABLE(<TDB>) option in the DHCP offer may be used by clients to prefer Forcerenew nonce Protocol authentication-capable DHCP servers over those servers which do not support such capability.

The server selects a Forcerenew nonce for a client only during Request/Ack message exchange. The server records the Forcerenew nonce and transmits that nonce to the client in an Authentication option in the DHCP Ack message.

The Forcerenew nonce is 128 bits long, and MUST be a cryptographically strong random or pseudo-random number that cannot easily be predicted. The nonce is imbedded as a 128-bit value of the Authentication information where type is set to 1 (Forcerenew nonce Value).

To provide authentication for a Forcerenew message, the server selects a replay detection value according to the RDM selected by the server, and computes an HMAC-MD5 of the Forcerenew message using the Forcerenew nonce for the client. The server computes the HMAC-MD5 over the entire DHCP Forcerenew message, including the Authentication option; the HMAC-MD5 field in the Authentication option is set to zero for the HMAC-MD5 computation. The server includes the HMAC-MD5 in the authentication information field in an Authentication option included in the Forcerenew message sent to the client with type set to 2 (HMAC-MD5 digest).

3.1.4. Client considerations for Forcerenew Nonce Authentication

The client MUST indicate Forcerenew nonce Capability by including the FORCERENEW_NONCE_CAPABLE(<TBD>) DHCP option (Section 2.1.1) in all DHCP Discover and Request messages. DHCP servers that support Forcerenew nonce Protocol authentication MUST include the DHCP Forcerenew Nonce protocol authentication option in DHCP Offers with type set to zero(0), allowing the client to use this capability in selecting DHCP servers should multiple Offers arrive.

A DHCP server has indicates its support through the inclusion of the FORCERENEW_NONCE_CAPABLE(<TBD>) option in the DHCP Offer. The client MUST validate the DHCP Ack message contains a Forcerenew Nonce in a DHCP authentication option. If the server has indicated capability for Forcerenew Nonce Protocol authentication in the DHCP Offer and a subsequent Ack omits a valid DHCP authentication option for the Forcerenew Nonce Protocol, the client MUST send a DHCP Decline message and return to the DHCP Init state.

The client will receive a Forcerenew Nonce from the server in the initial DHCP Ack message from the server. The client records the Forcerenew Nonce for use in authenticating subsequent Forcerenew messages.

To authenticate a Forcerenew message, the client computes an HMAC-MD5 over the DHCP Forcerenew message, using the Forcerenew Nonce received

from the server. If this computed HMAC-MD5 matches the value in the Authentication option, the client accepts the Forcerenew message.

4. Acknowledgements

Comments are solicited and should be addressed to the DHC WG mailing list (dhcwg@ietf.org) and/or the authors. This contribution is based on work by Vitali Vinokour. Major sections of this draft use modified text from [RFC3315]. The authors wish to thank Ted Lemon and Bernie Volz for their support.

5. IANA Considerations

This document requests IANA to allocate an option code for the newly defined DHCP option FORCERENEW_NONCE_CAPABALE as described in the text.

This document requests IANA to allocate a DHCP Authentication Option(90) protocol number be assigned for Forcerenew Nonce Authentication, per [RFC3118].

This document requests IANA to create a new namespace associated with the Forcerenew Nonce Authentication protocol: algorithm, per [RFC3118].

6. Security Considerations

As in some network environments FORCERENEW can be used to snoop and spoof traffic, the FORCERENEW message MUST be authenticated using the procedures as described in [RFC3118] or this proposal. In this proposal any party able intercept the nonce exchange could impersonate a server and thus offers no protection from man-in-the-middle attacks. FORCERENEW messages failing the authentication should be silently discarded by the client.

6.1. Protocol vulnerabilities

The mechanism described in this document is vulnerable to a denial of service attack through flooding a client with bogus FORCERENEW messages. The calculations involved in authenticating the bogus FORECERENEW messages may overwhelm the device on which the client is running.

The mechanism described provides protection against the use of a FORCERENEW message by a malicious DHCP server to mount a denial of

service or man-in-the-middle attack on a client. This protocol can be compromised by an attacker that can intercept the initial message in which the DHCP server sends the nonce to the client.

7. References

7.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC3118] Droms, R. and W. Arbaugh, "Authentication for DHCP Messages", RFC 3118, June 2001.

[RFC3203] T'Joens, Y., Hublet, C., and P. De Schrijver, "DHCP reconfigure extension", RFC 3203, December 2001.

7.2. Informative References

[RFC3315] Droms, R., Bound, J., Volz, B., Lemon, T., Perkins, C., and M. Carney, "Dynamic Host Configuration Protocol for IPv6 (DHCPv6)", RFC 3315, July 2003.

[TR-101] Cohen et al, "Architecture & Transport: "Migration to Ethernet Based DSL Aggregation, DSL Forum TR-101", 2005.

Authors' Addresses

David Miles
Alcatel-Lucent
L3 / 215 Spring St
Melbourne, Victoria 3000,
Australia

Phone: +61 3 9664 3308
Fax:
Email: david.miles@alcatel-lucent.com
URI:

Wojciech Dec
Cisco Systems
Haarlerbergpark Haarlerbergweg 13-19
Amsterdam, NOORD-HOLLAND 1101 CH
Netherlands

Phone:
Fax:
Email: wdec@cisco.com
URI:

James Bristow
Swisscom Schweiz AG
Zentweg 9
Bern, 3050,
Switzerland

Phone:
Fax:
Email: James.Bristow@swisscom.com
URI:

Roberta Maglione
Telecom Italia
Via Reiss Romoli 274
Torino 10148
Italy

Phone:
Email: roberta.maglione@telecomitalia.it

Network Working Group
Internet Draft
Intended status: Standards Track
Expires: September 7, 2011

S. Jiang
Huawei Technologies Co., Ltd
G. Chen
China Mobile
March 4, 2011

Requirements for Addresses Registration
draft-jiang-6man-addr-registration-req-02.txt

Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on September 7, 2011.

Copyright Notice

Abstract

In the IPv6 address allocation scenarios, node self-generated addresses are notionally conflicted with the network managed address architecture. These addresses need to be registered in the networking management plate for the purposes of central address administration. This document discusses the requirements of address registration and analyzes the possible solutions.

Table of Contents

1. Introduction & Requirements

In the IPv6 address allocation scenarios, node self-generated addresses, such as addresses in IPv6 Stateless Address Configuration [RFC4862, RFC4941] scenario and Cryptographically Generated Addresses (CGA, [RFC3972]), is notionally conflicted with the network managed address architecture, such as DHCPv6-managed network or network with Access Control List, in which addresses are assigned and managed by the network management plate.

The current IPv4 address allocation mode in DHCPv4-managed network is that the DHCPv4 server assigns addresses. Many operators of enterprise networks and similarly tightly administered networks have expressed the desire to hold on to this model when moving to IPv6, because they don’t want to have hosts end up with essentially random IPv6 addresses. However, the notion that a server assigns an address is for the most part incompatible with IPv6 stateless configuration.

A useful way to give network administrators most of what they want, while at the same time retaining compatibility with normal stateless configuration would be: if the self-generated IPv6 addresses are used, they may need to be registered in and granted by the networking management plate. The node may be required to perform this registration since only granted IPv6 addresses are allowed to be used to access the network.

This document discusses the requirements of address registration and analyzes the possible solutions. Dynamic Host Configuration Protocol for IPv6 (DHCPv6) and Router Advertisement may be extended to propagate the address registration request from network management to nodes. A DHCPv6 server may play the address registration server with newly defined DHCPv6 options. However, this may conflict with the original DHCP notion. A new set of protocol may have to be defined for the address registration purpose.

2. Terminology

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 RFC2119 [RFC2119].

3. Potential Solutions

3.1. Generic Address Registration Procedure

By current default, the nodes with self-generated addresses do not register their addresses to any network devices. However, this may result that the network may reject the access request from these devices.

As showed in below Figure 1, in the generic address registration procedure, the network management plate should firstly propagate the request of registering self-generated addresses. By received such requests, a node using the self-generated address should send an address registration message to the network management. The network management should check whether the requested address is accepted, for example, performing a Duplicated Address Detect or checking the address does not use the Reserved IPv6 Interface Identifiers [RFC5453]. If the requested address is accepted by the network management, it is registered in the address manage database, which may be used by other network functions, such as DNS or ACL. An acknowledgement is sent to the node, granting the usage of this address. If the requested address is not accepted by the network management, a rejected acknowledgement is sent to the node to indicate that it must generate a new address.

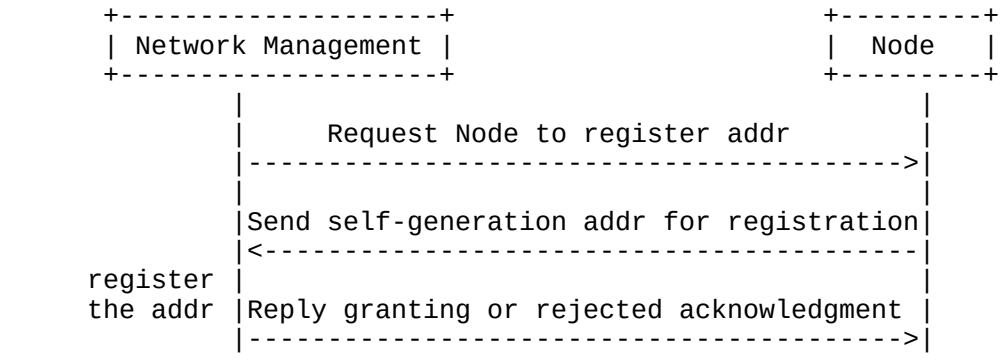


Figure 1: address registration procedure

3.2. Propagating the Registration Request

In order to indicate or force the nodes with self-generated addresses to register their addresses and the appointed address registration server, a new request option needs to be defined.

There are more than one mechanisms in which configuration parameters could be pushed to the end hosts. The address registration request option can be carried in Router Advertisement. In the DHCPv6 managed network, it can also be carried in DHCPv6 messages.

By receiving attendant of the address registration request option, a node MUST register its self-generated addresses, if there are any, to the appointed registration server. The option may be defined to include the default/enforced address registration server.

3.3. Address Registration Server and Protocol

In order to manage the address, an address registration server is needed with the support a set of address registration protocol.

The server should hold all registered addresses. It also needs to check whether the addresses meet the network address management policy, also performing a Duplicated Address Detect or checking the address does not use the Reserved IPv6 Interface Identifiers [RFC5453], etc. Its address data may be used by other network functions, such as DNS or ACL.

A set of address registration protocol need to at least support a basic information exchange: the node sends its address to the server and an acknowledgement is sent to the node.

3.3.1. Using DHCPv6 and DHCPv6 server

The current DHCPv6 protocol can be reused as the address registration protocol while a DHCPv6 serve plays as address registration server.

The current DHCPv6 specification allows for a host to communicate a set of "preferred" addresses to the server by listing these addresses in IA options [RFC3315]. In order to response to registration requests, an acknowledgement DHCPv6 option should be defined. It is used to indicate whether the registration of an IPv6 address is accepted.

3.3.2. Defining a new address Registration Protocol

However, the address registration procedure using DHC protocol may conflict with the initial notional of DHC protocol. The DHC protocol was originally designed to push configuration information from the network management side to the hosts while the address registration procedure is collecting information from hosts to the network management side.

A new set of address registration protocol may be defined.

[Author notes for IETF discussion:] Any other existing protocol may be used for address registration purposes?

4. Security Considerations

An attacker may use a faked address registration request option to indicate hosts reports their address to a malicious server and collect the user information. These attacks may be prevented by using secure protocols, in Neighbor Discovery protocol case, Secure Neighbor Discovery (SEND, [RFC3971]); in DHCP case, Secure DHCP [I-D.ietf-dhc-secure-dhcpv6]; or other additional security mechanisms.

An attacker could generate IPv6 address registration requests in order to exhaust the server resources (or to impact on any other operation that depend on the registration of the address).

In the use case of DHCPv6, the address registration procedure is as vulnerable as all other mechanisms based on DHCPv6 to DOS attacks to the server. Proper use of DHCPv6 autoconfiguration facilities [RFC3315], such as AUTH option or Secure DHCP [I-D.jiang-dhc-secure-dhcpv6] can prevent these threats.

5. IANA Considerations

There is no IANA considerations.

6. Change Log [RFC Editor please remove]

draft-jiang-6man-addr-registration-req-00, original version, 2010-03-01

draft-jiang-6man-addr-registration-req-01, minor update, 2010-08-27

draft-jiang-6man-addr-registration-req-02, minor update, 2010-03-04

7. Acknowledgments

The authors would like to thank Cao Wei, Huawei for been involved in the early requirement identification and early discussion.

8. References

8.1. Normative References

[RFC2119] S. Bradner, "Key words for use in RFCs to Indicate Requirement Levels", RFC2119, March 1997.

[RFC3315] R. Droms, Ed., J. Bound, B. Volz, T. Lemon, C. Perkins and M. Carne, "Dynamic Host Configure Protocol for IPv6", RFC3315, July 2003.

[RFC3971] J. Arkko, J. Kempf, B. Zill, P. Nikander, "SEcure Neighbor Discovery (SEND) ", RFC 3971, March 2005.

[RFC3972] T. Aura, "Cryptographically Generated Address", RFC3972, March 2005.

[RFC4862] S. Thomson, T. Narten and T. Jinmei, "IPv6 Stateless Address Autoconfiguration", RFC4862, September 2007.

[RFC4941] T. Narten, R. Draves and S. Krishnan, "Privacy Extensions for Stateless Address Autoconfiguration in IPv6", RFC 4941, September 2007.

[RFC5453] S. Krishnan, "Reserved IPv6 Interface Identifiers", RFC 4543, February 2009.

8.2. Informative References

[I-D.ietf-dhc-secure-dhcpv6] S. Jiang and S. Shen "Secure DHCPv6 Using CGAs", draft-ietf-dhc-secure-dhcpv6-02 (work in progress), December, 2010.

Author's Addresses

Sheng Jiang
Huawei Technologies Co., Ltd
Huawei Building, No.3 Xinxi Rd.,
Shang-Di Information Industry Base, Hai-Dian District, Beijing 100085
P.R. China
Email: jiangsheng@huawei.com

Gang Chen
China Mobile
53A,Xibianmennei Ave.,
Xuanwu District,
Beijing 100053
China
Email: phdgang@gmail.com

Network Working Group   Sheng Jiang
Internet Draft   Sam(Zhongqi) Xia
Intended status: Standards Track   Huawei Technologies Co., Ltd
Expires: May 30, 2011   November 19, 2010

Configuring Cryptographically Generated Addresses (CGA) using DHCPv6
draft-jiang-dhc-cga-config-dhcpv6-02.txt

Status of this Memo

Copyright Notice

Abstract

A Cryptographically Generated Address is an IPv6 addresses binding with a public/private key pair. However, the current CGA specifications are lack of procedures to enable proper management of the usage of CGAs. This document defines the process using DHCPv6 to manage CGAs in detail. A new DHCPv6 option is defined accordingly. This document also analyses the configuration of the parameters, which are used to generate CGAs, using DHCPv6. Although the document does not define new DHCPv6 option to carry these parameters for various reasons, the configuration procedure is described.

Table of Contents

1. Introduction

Cryptographically Generated Addresses (CGA, [RFC3972]) provide means to verify the ownership of IPv6 addresses without requiring any security infrastructure such as a certification authority.

CGAs were originally designed for SeND [RFC3971] and SeND is generally not used in the same environment as a Dynamic Host Configure Protocol for IPv6 (DHCPv6) [RFC3315] server. However, after CGA has been defined, as an independent security property, many other CGA usages have been proposed and defined, such as Site Multihoming by IPv6 Intermediation (SHIM6) [RFC5533], Enhanced Route Optimization for Mobile IPv6 [RFC4866], also using the CGA for DHCP security purpose [I-D.ietf-dhc-secure-dhcpv6], etc. The use of CGAs allows identity verification in different protocols. In these scenarios, CGAs may be used in DHCPv6-managed networks.

As [I-D.ietf-csi-dhcpv6-cga-ps] analyses, in the current specifications, there is a lack of procedures to enable proper management of the usage of CGAs. Particularly, in a DHCPv6-managed network, a new DHCPv6 option is missed, therefore, the DHCPv6 server can NOT grant the use of host-generated CGA addresses on request from the client, or reject the CGA on the basis of a too-low sec value. In order to fill this gap, a new DHCPv6 option, CGA Grant Option, is defined in this document.

This document also analyses the configuration of the parameters, which are used to generate CGAs, using DHCPv6. Although the document does not define new DHCPv6 option to carry these parameters for various reasons, the configuration procedure is described. The procedure works with existing options or future define options.

2. Terminology

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 RFC2119 [RFC2119].

3. CGA Configure Process Using DHCPv6

The CGA specifications [RFC3972] define the procedure to generate a CGA. However, it assumes that hosts decide by itself or have been preconfigured all CGA relevant parameters. In reality, the network management MAY want to assign/enforcement some parameters to hosts; the network management MAY also manage the use of CGAs.

Among the mechanisms in which configuration parameters could be pushed to the end hosts and/or CGA related information sent back to a central administration, we discuss the stateful configuration mechanism based on DCHPv6 in this document. Other mechanisms may also provide similar functions, but out of scope.

In this section, configuration CGA parameters and that a DHCPv6 server grants the CGA usage are described in details.

3.1. Configuration of the parameters required for the generation of CGA

Each CGA is associated with a CGA Parameters data structure, which is formed by all input parameters [RFC3972] except for Sec value that is embedded in the CGA. The CGA associated Parameters used to generate a CGA includes:

- a Public Key,

- a Subnet Prefix,

- a 3-bit security parameter, Sec. Additionally, it should be noted that the hash algorithm to be used in the generation of the CGA is also defined by the Sec value [RFC4982],

- any Extension Fields that could be used.

- Note: the modifier and the Collision Count value in the CGA Parameter data structure are generated during the CGA generation process. They do NOT need to be configured.

In a DHCPv6 managed network, a host may initiate a request for the relevant CGA configuration information needed to the DHCPv6 server. The server responds with the configuration information for the host. The Option Request Option, defined in Section 22.7 in [RFC3315], can be used for host to indicate which options the client requests from the server. For response, the requested Option should be included. The server MAY also initiatively push these parameters by attaching these option in the response messages which are initiated for other purposes.

- The Public/Private key pair is generated by hosts themselves and considered not suitable for network transmission for security reasons. The configuration of the client key pair or certificate is out of scope.

- Currently, there are convenient mechanisms for allowing an administrator to configure the subnet prefix for a host, by Router

Advertisement [RFC4861, RFC4862]. However, this does not suitable for the DHCP-managed network. To propagate the prefix through DHCP interactions, DHCPv6 Prefix Delegation Option [RFC3633] MAY be used. However, this option was designed to assign prefix block for routers. A new Prefix Assignment Option MAY need to be defined. Since alternative approach is existing and there are debates whether a new Prefix Assignment Option MAY is necessary, this document does not define it.

- Although the network management MAY want to enforce or configure a Sec value to the hosts, it is considered as a very dangerous action. A malicious fake server may send out a high Sec value to attack clients giving the fact that generation a CGA with a high Sec value is very computational intensive [I-D.ietf-csi-dhcpv6-cga-ps]. Another risk is that a malicious server could propagate a Sec value providing less protection than intended by the network administrator, facilitating a brute force attack against the hash, or the selection of the weakest hash algorithm available for CGA definition. A recommendation Sec value is considered as confusion information. The receiving host is lack for information to make choose whether generates a CGA according to the recommendation or not. Therefore, the document does not define a DHCPv6 option to propagate the Sec value.

- Although there is an optional Extension Fields in CGA Parameter data structure, there is NO any defined extension fields. If in the future, new Extension Fields in CGA Parameter data structure are defined, future specification may define correspondent DHCPv6 options to carry these parameters.

Upon reception of the CGA relevant parameters from DHCPv6 server, the end hosts SHOULD generate addresses compliant with the received parameters. If the parameters change, the end hosts SHOULD generate new addresses compliant with the parameters propagated.

3.2. Host requests CGA Approved to the DHCPv6 server

A CGA address is generated by the associated key pair owner, normally an end host. However, in a DHCPv6-managed network, hosts should use IPv6 global addresses only from a DHCPv6 server. The process described below allows a host, also DHCPv6 client, uses self-generated CGAs in a DHCPv6-managed environment, by requesting the granting from a DHCPv6 server.

The client sends a CGA, which is generated by itself, to a DHCPv6 server, and requests the DHCP server to determine whether the generated CGA satisfies the requirements of the network

configuration, wherein the network configuration comprises a CGA security level set by the DHCP; and generates a new CGA if the generated CGA does not satisfy the requirements of the network configuration.

Client initiation behavior

In details, a DHCPv6 client SHOULD send a DHCPv6 Request message to initiate the CGA granting process.

This DHCPv6 Request message MUST include an Option Request option, which requests the CGA Grant Option, defined in Section 4 in this document, to indicate the DHCPv6 server responses with the address granting decision. The CGA_Grant field in the embedded CGA Grant Option should be set all 1 (FFx).

The client MUST include one or more IA Options, either IA_NA or IA _TA, in the Request message. Each IA Option MUST include one or more IA Address Options. CGAs are carried in the IA Address Options.

Server behavior

Upon reception of the Request message, the DHCPv6 server SHOULD verify whether the client's CGAs satisfy the CGA-related configuration parameters of the network. The DHCPv6 server SHOULD NOT handle the Request which the CGA Grant field is not all 1(FFx). The DHCPv6 server then send an acknowledgement, a Reply message, to the client to either grant the use of the CGA or decline the requested CGA. The CGA_Grant field SHOULD be set following the rule, defined in Section 4 in this document. When the requested CGA is declined, the DHCPv6 server MAY also recommend a Sec value to the client a using the CGA Grant option.

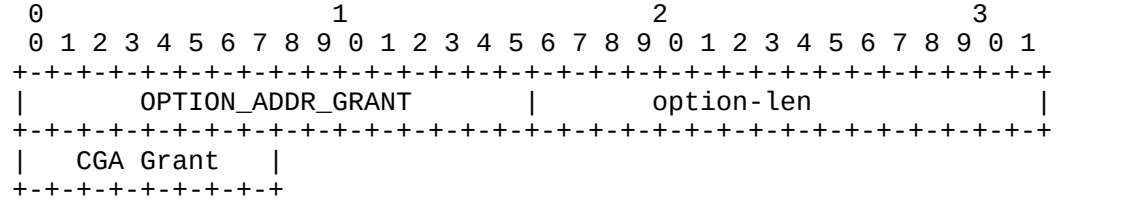
In the meantime, the DHCPv6 server MAY log the requested CGA addresses. This information MAY later be used by other network functions, such as ACL.

Client receiving behavior

Upon reception of the acknowledgement from server, the client can legally use the granted CGAs. The client SHOULD silently drop any message that has the CGA Grant field set any other value, but F0x, 00x˜07x. If the server declines the requested CGA, the client MUST generate a new CGA. If the server replies with CGA-relevant parameters, the client MAY generate a new CGA accordingly.

4. CGA Grant Option

DHCPv6 CGA Grant Option is used to indicate the DHCPv6 client whether the requested address is granted or not. In the decline case, a recommended Sec value MAY be sent, too.



option-code

OPTION_ADDR_GRANT (TBA1).

option-len

1.

CGA Grant

The CGA_Grant field sets all 1 (FFx) when a client requests granting from server. It sets F0x to indicate that the requested CGA is granted; it sets 00x to indicate that the requested Address is declined without any recommended Sec value. It sets 01x~07x to indicate that requested Address is declined and the recommended Sec value (value from 1~7).

Note: On receiving the CGA Grant Option with reject information and recommended Sec value, the client MAY generate a new CGA with the recommended Sec value. If choosing not use the recommended Sec value, the client MAY take the risk that it is not able to use full network capabilities.

5. Security Considerations

The mechanisms based on DHCPv6 are all vulnerable to attacks to the DHCP client. Proper use of DHCPv6 autoconfiguration facilities [RFC3315], such as AUTH option or Secure DHCP [I-D.ietf-dhc-secure-dhcpv6] can prevent these threats, provided that a configuration token is known to both the client and the server.

Note that, as expected, it is not possible to provide secure configuration of CGA without a previous configuration of security

information at the client (either a trust anchor, a DHCPv6 configuration token...). However, considering that the values of these elements could be shared by the hosts in the network segment, these security elements can be configured more easily in the end hosts than its addresses.

6. IANA Considerations

This document defines two new DHCPv6 [RFC3315] options, which must be assigned Option Type values within the option numbering space for DHCPv6 messages:

The DHCPv6 CGA Grant Option (TBA1), described in Section 4.

7. Acknowledgments

The authors would like to thank Marcelo Bagnulo Braun and Alberto Garcia-Martinez from Universidad Carlos III de Madrid for been involved in the early requirement identification. Valuable comments from Bernie Volz, Ted Lemon, John Jason Brzozowski and Dujuan Gu, Huawei are appreciated.

8. Change Log [RFC Editor please remove]

draft-jiang-dhc-cga-config-dhcpv6-02, remove Sec option according to IETF 79 discussion, 2010-11-19.

draft-jiang-dhc-cga-config-dhcpv6-01, remove CGA generation delegation according to IETF 77 and mail list discussion, 2010-08-24.

draft-jiang-dhc-cga-config-dhcpv6-00, original version, 2010-02-03.

9. References

9.1. Normative References

[RFC2119] S. Bradner, "Key words for use in RFCs to Indicate Requirement Levels", RFC2119, March 1997.

[RFC3315] R. Droms, Ed., "Dynamic Host Configure Protocol for IPv6", RFC3315, July 2003.

[RFC3633] O. Troan and R. Droms, "IPv6 Prefix Options for Dynamic Host Configuration Protocol (DHCP) version 6", RFC 3633, December 2003.

[RFC3971] J. Arkko, J. Kempf, B. Zill and P. Nikander, "SEcure Neighbor Discovery (SEND) ", RFC 3971, March 2005.

[RFC3972] T. Aura, "Cryptographically Generated Address", RFC3972, March 2005.

[RFC4861] T. Narten, et al., "Neighbor Discovery for IP version 6 (IPv6)", RFC 4861, September 2007.

[RFC4862] S. Thomson, T. Narten and T. Jinmei, "IPv6 Stateless Address Autoconfiguration", RFC4862, September 2007.

[RFC4866] J. Arkko, C. Vogt and W. Haddad, "Enhanced Route Optimization for Mobile IPv6", RFC4866, May 2007.

[RFC4982] M. Bagnulo, "Support for Multiple Hash Algorithms in Cryptographically Generated Addresses (CGAs) ", RFC4982, July 2007.

[RFC5533] E. Nordmark and M. Bagnulo, "Shim6: Level 3 Multihoming Shim Protocol for IPv6" FRC 5533, June 2009.

9.2. Informative References

[I-D.ietf-csi-dhcpv6-cga-ps]
S. Jiang, S. Shen and T. Chown, "DHCPv6 and CGA Interaction: Problem Statement", draft-ietf-csi-dhcpv6-cga-ps (work in progress), October, 2010.

[I-D.ietf-dhc-secure-dhcpv6]
S. Jiang and S. Shen, "Secure DHCPv6 Using CGAs", draft-ietf-dhc-secure-dhcpv6 (work in progress), June 2010.

Author's Addresses

Sheng Jiang
Huawei Technologies Co., Ltd
Huawei Building, No.3 Xinxi Rd.,
Shang-Di Information Industry Base, Hai-Dian District, Beijing 100085
P.R. China
Email: shengjiang@huawei.com

Sam(Zhongqi) Xia
Huawei Technologies Co., Ltd
Huawei Building, No.3 Xinxi Rd.,
Shang-Di Information Industry Base, Hai-Dian District, Beijing 100085
P.R. China
Email: xiazhongqi@huawei.com

DHC
Internet-Draft
Intended status: Standards Track
Expires: September 8, 2011

S. Joshi
Alcatel Lucent

March 7, 2011

Aggregate Route Option for Dynamic Host Control Protocol version 6 (DHCPv6)
draft-joshi-dhc-dhcpv6-aggr-route-opt-00

Abstract

The Aggregate Route option provides a mechanism for a requestor to retrieve an aggregate route(s) from a DHCPv6 server using the information-request message. The aggregate route is a single route representing multiple prefixes delegated by a DHCP server using Prefix Delegation, and maybe advertised using routing protocols instead of individual routes learnt from DHCPv6 Prefix Delegation. This document specifies the data contained in aggregate route option as well as the behavior of Requestor and DHCPv6 Server in requesting and processing of this option.

Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on September 8, 2011.

Copyright Notice

Table of Contents

1. Introduction

In service provider networks intermediate routers between DHCPv6 Server and Consumer Premise Equipment (CPE) equipment implement a DHCPv6 Relay function to learn Prefixes Delegated [RFC3633] by the DHCPv6 server to CPE equipment. The intermediate routers may use routing protocols to advertise themselves as routers for these individual delegated prefixes. With each intermediate router being connected to a large number of CPE equipment the number of routes the intermediate router needs to advertise is substantial, increasing the size of route tables in peer routers.

If the prefixes delegated by the DHCPv6 server are contiguous then a single aggregate route can represent multiple delegated prefixes. While it is possible to configure such an aggregate route either manually or through Simple Network Management Protocol, it would be operationally efficient if the intermediate router can query the DHCPv6 server for aggregate route be be advertised.

The Aggregate Route option provides such a mechanism to the intermediate router to query the DHCPv6 server for aggregate routes to advertise through routing protocols. Even though the mechanism proposed makes it easy to advertise and withdraw aggregate routes, it is expected that aggregate routes will have a long lifespan.

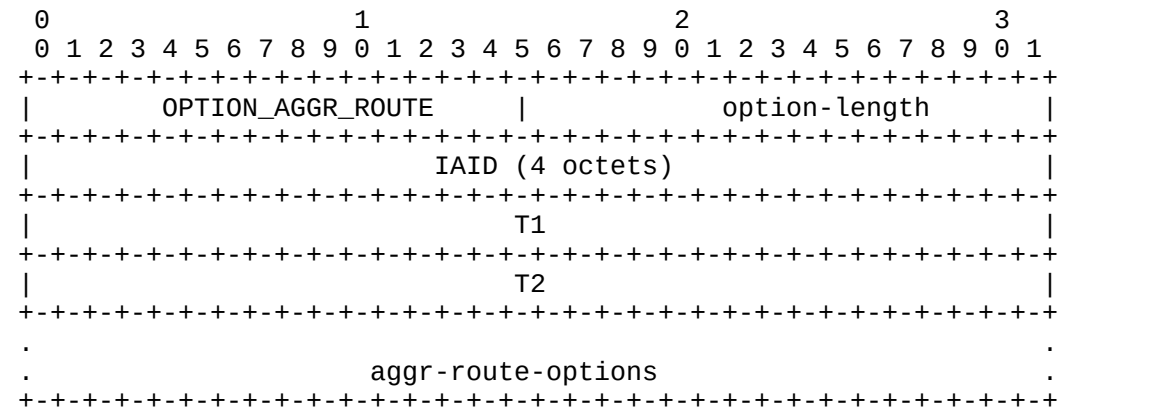
2. Terminology and Language

This document describes new DHCPv6 option for aggregate route. This document should be read in conjunction with the DHCPv6 specification, RFC 3315 and RFC 3633, for a complete mechanism. Definitions for terms and acronyms not specifically defined in this document are defined in RFC 3315, RFC 3633 and RFC 3769 [RFC3769].

The keywords MUST, MUST NOT, REQUIRED, SHALL, SHALL NOT, SHOULD, SHOULD NOT, RECOMMENDED, MAY, and OPTIONAL, when they appear in this document, are to be interpreted as described in BCP 14, RFC 2119 [RFC2119].

3. Aggregate Route option

The format of the Aggregate Route option is:



option-code: OPTION_AGGR_ROUTE (TBD)

option-length: 12 + length of aggr-route-options field

IAID: The unique identifier for this OPTION_AGGR_ROUTE; the IAID must be unique among the identifiers for all of this requesting router's OPTION_AGGR_ROUTES.

T1: The time at which the requestor should contact the delegating router from which the prefixes were obtained to extend the lifetimes of the aggregated route. T1 is a time duration relative to the current time expressed in units of seconds.

T2: The time at which the requestor should contact any available delegating router to extend the lifetimes of the prefixes assigned to the requestor; T2 is a time duration relative to the current time expressed in units of seconds.

aggr-route-options: Options associated with this aggregated route.

The aggr-route-options field encapsulates those options that are specific to this aggregate route request.

In a message sent by the requestor the values in these fields can be used to indicate requestors preference for those values. The requestor shall include one or more options e.g. OPTION_INTERFACE_ID necessary for server to select a unique set of prefixes to be selected for this aggregate route request.

A DHCP server includes the OPTION_IAPREFIX to indicate the prefixes associated with this aggregate route request. More than one prefixes can be associated with a OPTION_AGGR_ROUTE. The status of this OPTION_IAPREFIX is indicated in a Status Code option in the aggr-route-options field.

A OPTION_AGGR_ROUTE may only appear in the options area of a DHCP message. A DHCP message may contain multiple Aggregate Route options.

Note that the Aggregate Route option is an container option and does not have a valid lifetime of its own. When the lifetime of all the associated prefixes have expired, the Aggregate Route option can be considered as expired. T1 and T2 are included to give the DHCP server control over when the Requestor should contact the server for a specific prefix.

In a message sent by a Requestor to a Server, values in the T1 and T2 fields indicate the Requestors preference for those parameters. The Requestor sets T1 and T2 to zero if it has no preference for those values. In a message sent by a Server to a Requestor, the Requestor MUST use the values in the T1 and T2 fields for the T1 and T2 parameters. The values in the T1 and T2 fields are the number of seconds until T1 and T2.

The Server selects the T1 and T2 times to allow the Requestor to extend the lifetimes of any prefixes in the OPTION_AGGR_ROUTE before the lifetimes expire, even if the Server is unavailable for some short period of time. Recommended values for T1 and T2 are .5 and .8 times the shortest preferred lifetime of the prefixes in the OPTION_AGGR_ROUTE that the Server is willing to extend, respectively. If the time at which the prefixes in an OPTION_AGGR_ROUTE are to be renewed is to be left to the discretion of the requesting router, the Server sets T1 and T2 to 0.

If a Server receives an OPTION_AGGR_ROUTE with T1 greater than T2, and both T1 and T2 are greater than 0, the Server ignores the invalid values of T1 and T2 and processes the OPTION_AGGR_ROUTE as though the Server had set T1 and T2 to 0.

If a Requestor receives an OPTION_AGGR_ROUTE with T1 greater than T2, and both T1 and T2 are greater than 0, the client discards the OPTION_AGGR_ROUTE option and processes the remainder of the message as though the Server had not included the OPTION_AGGR_ROUTE option.

4. Requestor Behavior

4.1. Requesting Aggregate Route Information

The Requestor requests aggregate route information from the DHCP server by sending an information-request message containing one or more OPTION_AGGR_ROUTE (RFC 3315 Section 18.1.5).

The Requestor MUST include its DUID in the information-request message (for a client this is client ID and for a relay this is relay ID).

The Requestor MUST generate and include a transaction-id in the information-request message.

The Requestor within the aggr-route-options of each OPTION_AGGR_ROUTE includes information necessary for the server to associate a unique set of prefixes. The additional information may include options such as INTERFACE_ID.

The Requestor with multiple interface MAY include individual OPTION_AGGR_ROUTE in a single information-request message, with each OPTION_AGGR_ROUTE containing and INTERFACE_ID in its aggr-route-options.

The requestor MAY be configured to use a list of known DHCP server as destination addresses. The requestor SHOULD unicast the information-request to one or more known DHCPv6 servers. In case no such list is configured the requestor MAY send multicast request to All_DHCP_Servers address.

Requestor transmits the information-request according to Section 18.1.5 of RFC 3315.

4.2. Processing Server Reply

Upon receipt of a valid Reply message for each prefix in the OPTION_AGGR_ROUTE the Requestor MAY based on its local configuration add an aggregate route entry into its routing table. The Requestor MAY also advertise itself as a router for the valid prefixes through routing protocols such as OSPF and BGP. Before expiry of valid lifetime of each prefix, the Requestor sends a Renew message to DHCP Server with OPTION_AGGR_ROUTE containing the prefix.

4.3. Validation of aggregate route bindings

The Requestor may request validation of aggregate route binding from the server through the Rebind/Reply exchange. Events which can

trigger the validation MAY include.

- Requestor Reboots.

- Requestor detects connectivity loss towards the server.

- Physical disconnection from network.

5. Server Behavior

Upon receipt of a valid information-request containing OPTION_AGGR_ROUTE, Server uses the information contained in the aggr-route-options to identify the associated Prefixes and populates the OPTION_IAPREFIX in aggr-route-options for each of OPTION_AGGR_ROUTE of the REPLY message.

The Server SHALL copy into the REPLY message all the aggr-route-options received from the Requestor.

When the status of aggregate route is reset by manual configuration, the Server shall initiate the message of RECONFIGURE (10) with the Requestor.

6. Acknowledgements

This document offers an alternate mechanism to solution specified in draft-yeh-dhcp-dhcpv6-prefix-pool-opt, the author would like to thank the authors of draft-yeh-.. for discussion of the problem and solution which has served as an input to this draft.

7. Security Considerations

Security issues related DHCPv6 are described in section 23 of RFC 3315.

8. IANA Considerations

IANA is requested to assign an option code to OPTION_AGGR_ROUTE from the "DHCPv6 and DHCPv6 options" registry (http://www.iana.org/assignments/dhcpv6-parameters/dhcpv6-parameters.xml).

9. References

9.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC3315] Droms, R., Bound, J., Volz, B., Lemon, T., Perkins, C., and M. Carney, "Dynamic Host Configuration Protocol for IPv6 (DHCPv6)", RFC 3315, July 2003.

[RFC3633] Troan, O. and R. Droms, "IPv6 Prefix Options for Dynamic Host Configuration Protocol (DHCP) version 6", RFC 3633, December 2003.

[RFC3769] Miyakawa, S. and R. Droms, "Requirements for IPv6 Prefix Delegation", RFC 3769, June 2004.

9.2. Informative References

[BBF TR-177] Broadband Forum, "IPv6 in the context of TR-101 Issue: 1", November 2010.

Author’s Address

Shrinivas Joshi
Alcatel Lucent

Email: shrinivas_ashok.joshi@alcatel-lucent.com

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: September 5, 2011

F. Xia
B. Sarikaya
S. Jiang
Huawei Technologies
March 04, 2011

Usage of Host Generating Interface Identifier in DHCPv6
draft-xia-dhc-host-gen-id-03.txt

Abstract

This document describes a procedure for configuring a host's IPv6 address which prefix is allocated from a DHCPv6 server while it's interface identifier is independently generated by the host. The method is applicable to Cryptographically Generated Addresses (CGA).

Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on September 5, 2011.

Table of Contents

1. Introduction

[RFC3315] describes the operation of address assignment by a DHCP server. A client uses a Solicit message to discover DHCP servers configured to assign addresses. A server sends an Advertise message in response to announce the availability of the server to the client. The client then uses a Request message to request addresses. The server then returns addresses in a Reply message. The operation assumes that the server is responsible for the assignment of an integral address which include prefix and interface identifier parts as described in [RFC4291].

[RFC3633] defines Prefix Delegation options providing a mechanism for automated delegation of IPv6 prefixes using the DHCPv6. This mechanism is intended for delegating a long-lived prefix from a delegating router to a requesting router. The practice of separating prefix assignment from interface identifier assignment is only used for routers not hosts.

[RFC3972] describes a method for binding a public signature key to an IPv6 address in the Secure Neighbor Discovery (SEND) protocol [RFC3971]. The basic idea is to generate the interface identifier (i.e., the rightmost 64 bits) of the IPv6 address by computing a cryptographic hash of the public key. That is, the host decides its interface identifier. As for the prefix part of the CGA, it is probably got through Router Advertisement message defined in [RFC4861], or through DHCPv6 operations defined in this document.

[I-D.ietf-csi-dhcpv6-cga-ps] describes potential issues in the interaction between DHCPv6 and CGA. A usage of DHCPv6 for generating CGA is proposed in the document to facilitate separation of prefix and interface identifier assignment. A host’s IPv6 address prefix is allocated from a DHCPv6 server while interface identifier is independently generated by the host.

2. Terminology

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 [RFC2119].

The terminology in this document is based on the definitions in [RFC3315], in addition to the ones specified in this section

derivative prefix: A prefix is derived from another prefix. For example, a /64 prefix is derived from a /48 prefix, that is, the /64 prefix has the same leftmost 48 bits with the /48 prefix.

authorized prefix: A specific router is given a specific set of subnet prefixes to advertise; other routers have an authorization to advertise other subnet prefixes. In [RFC3971],Certification Path Advertisement message is used to convey authorized prefixes.

3. Address Auto-configuration in SEND

Router Advertisements in [RFC4861] allow routers to inform hosts how to perform Address Auto-configuration. For example, routers can specify whether hosts should use DHCPv6 and/or stateless address configuration. In Router Advertisement message, M and O bits are used for indication of address auto-configuration mode.

Whatever address auto-configuration mode a host uses, the following two parts are necessary for the host to formulate it's IPv6 address:

- A prefix part. In [RFC3971], Certification Path Solicitation and Certification Path Advertisement messages are designed for verifying routers being authorized to act as routers. Certification Path Advertisement message can also be used to verify that routers are authorized to advertise a certain set of subnet prefixes. In stateless auto-configuration mode, the prefixes in Router Advertisement message should be a subset of authorized prefixes, or derivative prefixes from authorized prefixes. In the stateful auto-configuration mode, Section 4 illustrates a procedure for prefix allocation from a DHCPv6 server.
- An interface identifier. The basic idea of [RFC3972] is to generate the interface identifier (i.e., the rightmost 64 bits) of the IPv6 address by computing a cryptographic hash of a public key of a host. The host is responsible for interface identifier generation.

4. DHCPv6 Operation

Figure 1 shows the operation of separating prefix assignment and interface identifier generation.

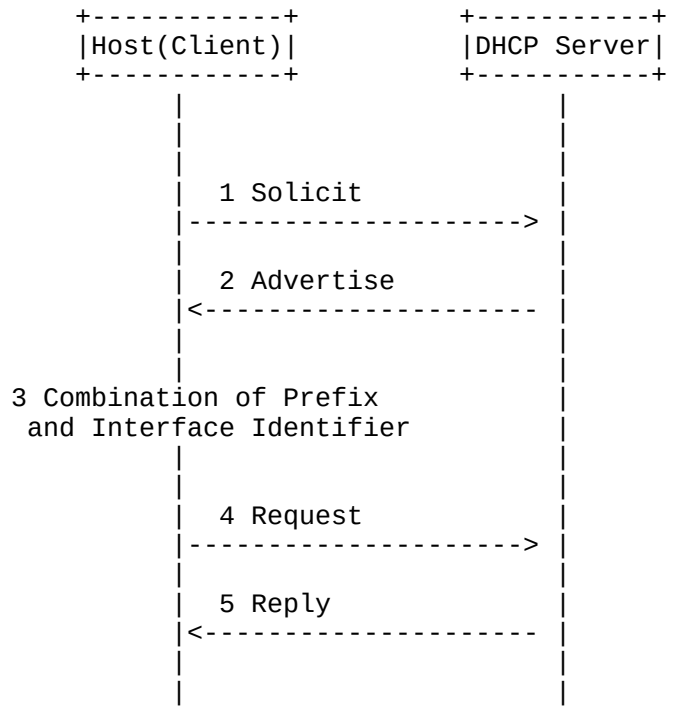


Figure 1: DHCPv6 Operation

1. A host uses a Solicit message to discover DHCP servers configured to assign prefixes for the host. Identity Association for Prefix Delegation Option (IA_PD) is defined in [RFC3633] for prefix delegation between a requesting router and delegating router. Referring to the definition, we design Identity Association for Prefix Assignment Option (IA-PA) in Section 5.1 for prefix assignment from a DHCPv6 server to a host. The host uses hints for prefix assignment preference. The hints are authorized prefixes advertised by an authorized router through Certification Path Advertisement defined in [RFC3971].
2. Based on the hints, the DHCP server assigns one or more prefixes to the host. The assigned prefixes SHOULD be a subset of the authorized prefixes or derivative prefixes of the authorized prefixes. Identity Association for Prefix Assignment Option in Section 5.1 is used for conveying the assigned prefixes. If there is not a proper prefix available, a status-code is returned to the host and the procedure is terminated.
3. The host generates an interface identifier and formulates a combined IPv6 address by concatenating the assigned prefix and the self-generated interface identifier. There are many ways to

generate interface identifier. [RFC3972] defines a method to generate the interface identifier by computing a cryptographic hash of a public key of the host.

4. The host sends a Request message for confirming usage of the combined address. An IA Address option described in Section 5.3 SHOULD be included to convey the combined address.
5. The DHCP server SHOULD verify the uniqueness of the combined IP address, and send Reply with IA Address option to grant the usage of the combined address. Otherwise, a status code is included to deny the usage of the combined address.

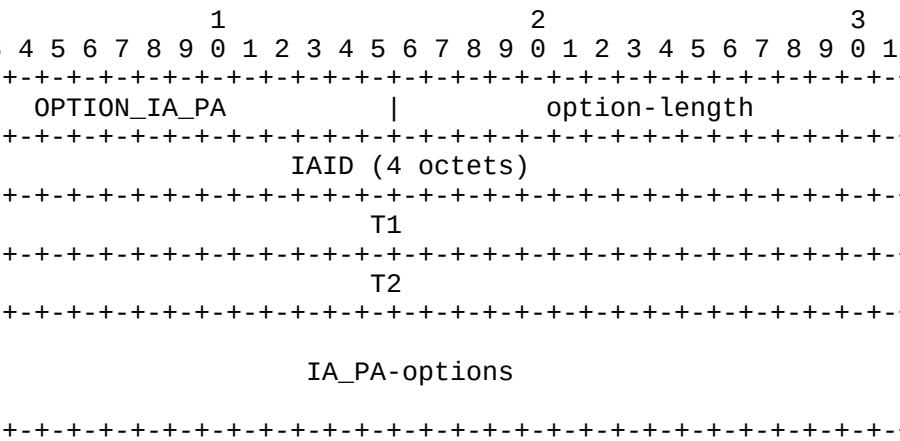
5. DHCPv6 Options

In this section, one new option is defined, Identity Association for Prefix Assignment Option . At the same time, we extend the usage of existing options, IA_PD Prefix and IA Address option.

5.1. Identity Association for Prefix Assignment Option

The IA_PA option is used to carry a prefix assignment identity association, the parameters associated with the IA_PA and the prefixes associated with it.

The format of the IA_PA option is:



option-code: OPTION_IA_PA (TBA1)

option-length: 12 + length of IA_PA-options field.

IAID: The unique identifier for this IA_PA; the IAID must be unique among the identifiers for all of this host's IA_PAs.

T1: The time at which the host should contact the DHCPv6 server from which the prefixes in the IA_PA were obtained to extend the lifetimes of the prefixes assigned to the IA_PA; T1 is a time duration relative to the current time expressed in units of seconds.

T2: The time at which the host should contact any available DHCPv6 server to extend the lifetimes of the prefixes assigned to the IA_PA; T2 is a time duration relative to the current time expressed in units of seconds.

IA_PA-options: Options associated with this IA_PA.

The details of the fields are similar to the IA_PD option description in [RFC3633]. The difference is here a DHCP server and a host involved, while a delegating router and requesting router involved in [RFC3633].

5.2. IA_PD Prefix option

IA_PD Prefix option in [RFC3633] is reused here. Originally the option is used for conveying prefix information between a delegating router and a requesting router. Here the IA_PD Prefix option is used to specify IPv6 address prefixes associated with an IA_PA in Section 5.1. The IA_PD Prefix option must be encapsulated in the IA_PA-options field of an IA_PA option.

5.3. IA Address Option

The IA Address option in [RFC3315] is reused here. It must be encapsulated in the Options field of an IA_NA or IA_TA option. IA_NA and IA_TA are also described in [RFC3315].

A host sends a DHCPv6 message with an IA Address option to a DHCPv6 server for validating the usage of an address in the option.

6. Applicability

In point-to-point link model, DHCPv6 operation with host generating interface identifier described in this document may be used. [RFC4968] provides different IPv6 link models that are suitable for 802.16 based networks and a point-to-point link model is recommended. Also, 3GPP and 3GPP2 have earlier adopted the point-to-point link model based on the recommendations in [RFC3314]. In this model, one prefix can only be assigned to one interface of a host (mobile station) and different hosts (mobile stations) can’t share a prefix. The unique prefix can be used to identify the host. It is not necessary for a DHCP server to generate an interface identifier for the host. The host may generates it’s interface identifier as described in [RFC4941]. An interface identifier could even be generated via random number generation.

7. IANA consideration

This document defines a new DHCPv6 [RFC3315] option, which must be assigned Option Type values within the option numbering space for DHCPv6 messages:

The OPTION_IA_PA Option (TBA1), described in Section 5.1.

8. Security Considerations

Security considerations in DHCPv6 are described in [RFC3315].

To guard against attacks through prefix assignment and address confirmation, a host and a DHCPv6 server SHOULD use DHCP authentication as described in section "Authentication of DHCP messages" of [RFC3315].

9. Acknowledgements

10. References

10.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC4861] Narten, T., Nordmark, E., Simpson, W., and H. Soliman, "Neighbor Discovery for IP version 6 (IPv6)", RFC 4861, September 2007.

[RFC3971] Arkko, J., Kempf, J., Zill, B., and P. Nikander, "SEcure Neighbor Discovery (SEND)", RFC 3971, March 2005.

[RFC3972] Aura, T., "Cryptographically Generated Addresses (CGA)", RFC 3972, March 2005.

[RFC3315] Droms, R., Bound, J., Volz, B., Lemon, T., Perkins, C., and M. Carney, "Dynamic Host Configuration Protocol for IPv6 (DHCPv6)", RFC 3315, July 2003.

[RFC3633] Troan, O. and R. Droms, "IPv6 Prefix Options for Dynamic Host Configuration Protocol (DHCP) version 6", RFC 3633, December 2003.

[RFC4291] Hinden, R. and S. Deering, "IP Version 6 Addressing Architecture", RFC 4291, February 2006.

[RFC4941] Narten, T., Draves, R., and S. Krishnan, "Privacy Extensions for Stateless Address Autoconfiguration in IPv6", RFC 4941, September 2007.

10.2. Informative references

[RFC4968] Madanapalli, S., "Analysis of IPv6 Link Models for 802.16 Based Networks", RFC 4968, August 2007.

[RFC3314] Wasserman, M., "Recommendations for IPv6 in Third Generation Partnership Project (3GPP) Standards",

RFC 3314, September 2002.

[I-D.ietf-csi-dhcpv6-cga-ps]
Jiang, S., "DHCPv6 and CGA Interaction: Problem Statement", draft-ietf-csi-dhcpv6-cga-ps-06 (work in progress), October 2010.

Authors' Addresses

Frank Xia
Huawei Technologies
1700 Alma Dr. Suite 500
Plano, TX 75075

Phone: +1 972-509-5599
Email: xiayangsong@huawei.com

Behcet Sarikaya
Huawei Technologies
1700 Alma Dr. Suite 500
Plano, TX 75075

Phone: +1 972-509-5599
Email: sarikaya@ieee.org

Sheng Jiang
Huawei Technologies
KuiKe Building, No.9 Xinxi Rd.,
Shang-Di Information Industry Base, Hai-Dian District, Beijing 100085
P.R. China

Phone: +86 10-82836774
Email: shengjiang@huawei.com

DHC Working Group
Internet-Draft
Intended status: Standards Track
Expires: January 29, 2013

L. Yeh, Ed.
Huawei Technologies
M. Boucadair
France Telecom
T. Lemon
Nominum, Inc
J. Hu
China Telecom
July 28, 2012

Prefix Pool Option for DHCPv6 Relay Agents on Provider Edge Routers
draft-yeh-dhc-dhcpv6-prefix-pool-opt-08

Abstract

The DHCPv6 Prefix Pool option provides a mechanism for DHCPv6 Prefix Delegation (DHCPv6-PD), allowing the DHCPv6 server to notify a DHCPv6 relay agent implemented on a Provider Edge (PE) router about active prefix pools allocated by the DHCPv6 server to the PE router. The information of active prefix pools can be used to enforce IPv6 route aggregation on the PE router by adding or removing aggregated routes according to the status of the prefix pools. The advertising of the aggregated routes in the routing protocol enabled on the network-facing interface of PE routers will dramatically decreases the number of the routing table entries in the ISP network.

Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on January 29, 2013.

Table of Contents

1. Introduction

The DHCPv6 protocol [RFC3315] specifies a mechanism for the assignment of IPv6 address and configuration information to IPv6 nodes. The DHCPv6 Prefix Delegation (DHCPv6-PD) [RFC3633] specifies a mechanism for the delegation of IPv6 prefixes from the Delegating Router (DR) acting as the DHCPv6 server to the Requesting Routers (RR) acting as the DHCPv6 Clients. DHCPv6 servers always maintain authoritative information associated to their operations including, but not limited to, the binding data of the delegated IPv6 prefixes, the lease data of the delegated IPv6 prefixes, and the status of their prefix pools. A prefix pool configured and maintained on the server can usually be a short prefix (e.g., a /40 prefix) out of which the longer prefixes (e.g., /56 prefixes) are delegated to customer networks.

In the scenario of a centralized DHCPv6 server, the Provider Edge (PE) routers act as DHCPv6 relay agents when the DHCPv6 server and the Customer Edge (CE) router (a.k.a. Routed-RG or Routed-CPE) acting as RRs and DHCPv6 clients are not on the same link. For ensuring reachability, the PE routers always need to add or withdraw the route entries directing to each customer network in their routing table to reflect the status of IPv6 prefixes delegated by the DHCPv6 server to CE routers (see Section 6.2, [BBF TR-177]).

When a routing protocol is enabled on the network-facing interface of the PE router, all the routes directing to the customer networks are advertised in the ISP network. This will make the number of route entries in the routing table on the ISP router be unacceptable large. Hence, it is desirable to aggregate the routes directing to the customer networks on the PE router.

Because the prefixes of the customer networks can not be guaranteed to be active and continuous, the routing protocol enabled on the PE router in general can not create one aggregated route automatically to cover all the prefixes delegated within the prefix pool. One way to make the aggregated routes (e.g., black-hole routes) pointing to each of the prefix pools is to configure them manually and permanently, but the PE router is not really aware about the status of the prefix pools, especially when it acts as the relay agent.

This document proposes a new Prefix Pool option for the DHCPv6 relay agent implemented on PE routers, allowing the DHCPv6 server to notify the DHCPv6 relay agent about the prefix of pools. After the PE router received information about the prefix pools, the aggregated route entries per the provision status of the prefix pools can be added or withdrawn in the routing table of the PE router. The aggregated routes will then be advertised into the ISP network

through the routing protocol enabled on the PE's network-facing interface.

DHCPv6 Bulk Leasequery [RFC5460] specifies a mechanism for bulk transfer of the binding data of each delegated prefix from the server to the requestor (i.e., a DHCPv6 relay agent), to support the replacement or reboot event of a relay agent. In this document, the capability of DHCPv6 Bulk Leasequery will be extended to support the bulk transfer of the status of the prefix pools for route aggregation.

The automatic mechanisms described in this document depends on the existing DHCPv6 protocols and implementations without requiring a new DHCPv6 message or a new interface for the configuration of the aggregated route. The administrator of the ISP network can decide whether to inject the aggregated route or not based on the policies defined on the DHCPv6 server.

2. Terminology and Conventions

This document defines a new DHCPv6 option to communicate the prefix of an IPv6 prefix pool. This document should be read in conjunction with the DHCPv6 specifications, [RFC3315], [RFC3633], [RFC5007] and [RFC5460], for understanding the complete mechanism. Definitions for terms and acronyms not specified in this document are defined in [RFC3315], [RFC3633], [RFC3769], [RFC5007] and [RFC5460].

The following terms can be found in this document:

- Requesting Router (RR): A router defined in [RFC3633] that acts as a DHCPv6 client, and is requesting prefix(es) to be assigned.
- Delegating Router (DR): A router defined in [RFC3633] that acts as a DHCPv6 server, and is responding to the prefix request.
- Prefix Pool: An IPv6 address space allocated with a common prefix out of which the longer prefixes are delegated via prefix delegation.
- Aggregated Route: A route entry created on an edge router, is based on the knowledge of a delegated prefix pool.
- Requestor: A router defined in [RFC5007] that acts as a DHCPv6 relay agent, is leasequery client.

The keywords MUST, MUST NOT, REQUIRED, SHALL, SHALL NOT, SHOULD, SHOULD NOT, RECOMMENDED, MAY, and OPTIONAL, when they appear in this

document, are to be interpreted as described in BCP 14, [RFC2119].

3. Scenario and Network Architecture

Figure 1 and Figure 2 illustrate two typical cases of the targeted network architectures.

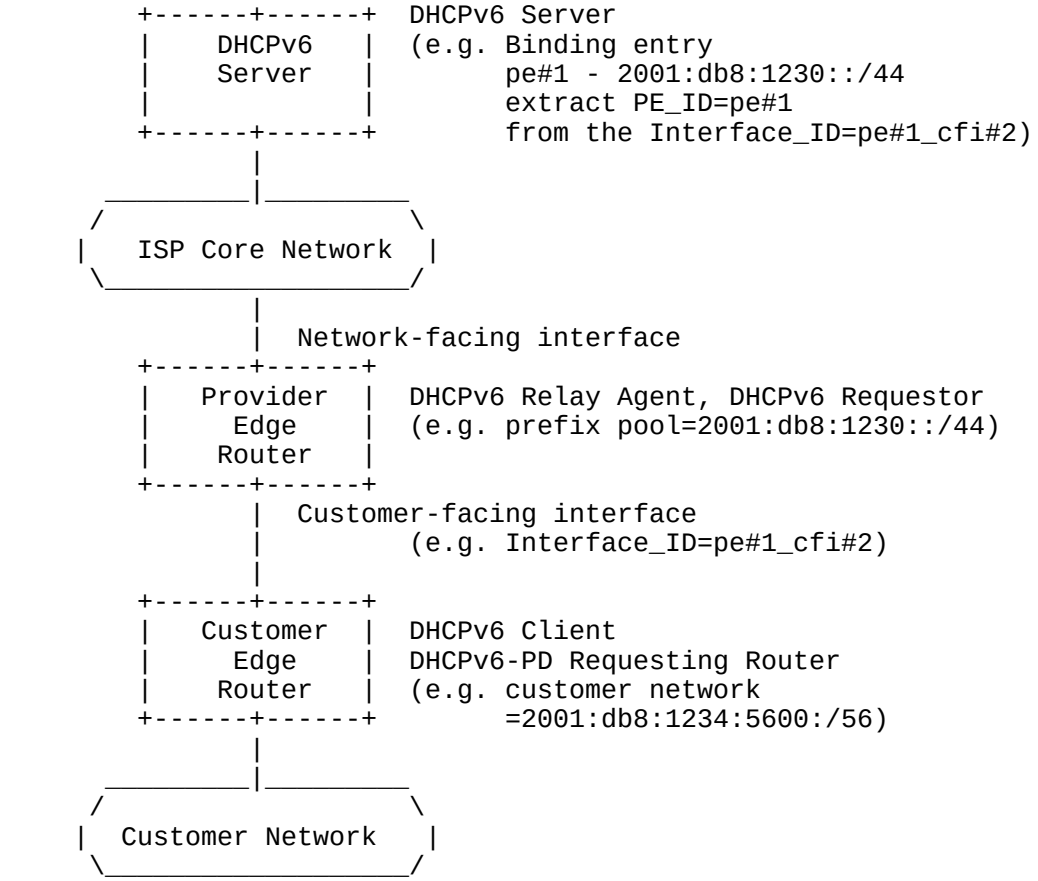


Figure 1: Use case of ISP-Customer network where CPE is directly connected to PE

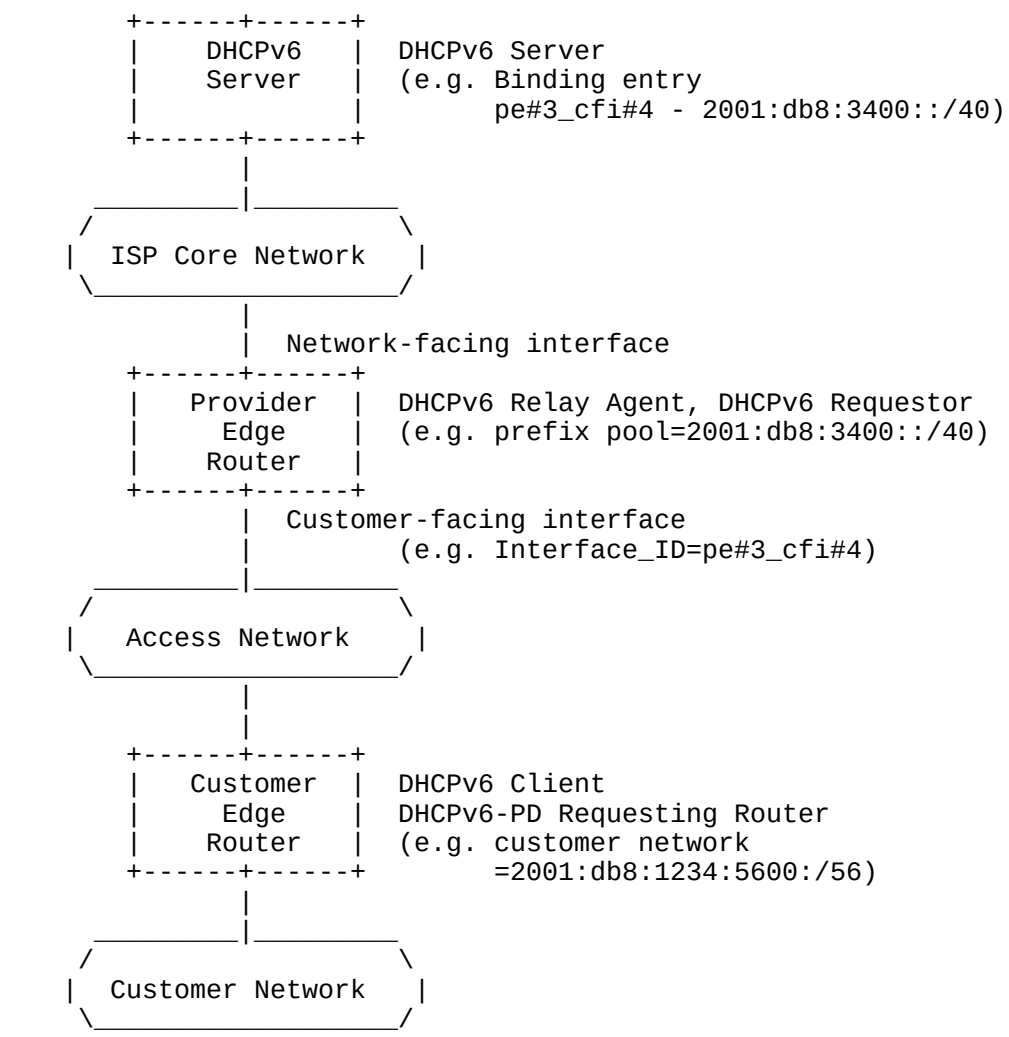
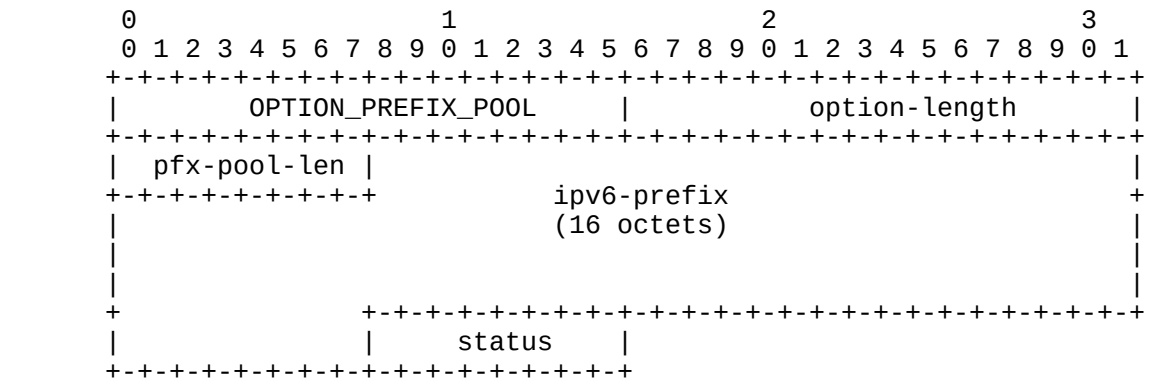


Figure 2: Use case of ISP-Customer network where CPE is connected to PE through access network

4. Prefix Pool Option

The format of the Prefix Pool option is shown in Figure 3.



option-code: OPTION_PREFIX_POOL (TBD)
option-length: 18
pfx-pool-len: Length for the prefix pool in bits
ipv6-prefix: IPv6 prefix of the prefix pool
status: Status of the prefix pool, indicating the availability of the prefix pool maintained on the server.

The codes of the status are defined in the following table.

Name	Code
Active	0
Released	1
Reserved	2~255

The 'Active' status of the prefix pool indicated in this option can be used to add the prefix pool and its associated aggregated route on the relay agent; while the 'Released' status of prefix pool indicated in this option can be used to withdraw the prefix pool and its associated aggregated route on the relay agent.

If the administrative policy on the DHCPv6 server permits to support route aggregation on the relay agent, the status of prefix pool can be determined by the delegated prefixes within the associated prefix pool. If there is one delegated prefix within the pool that has a valid lease, the status of the prefix pool will be 'Active'. Otherwise, the status of the prefix pool is 'Released'. If the administrative policy on the server does not permit to support route aggregation on the DHCPv6 relay agent, the status of the prefix pool will always be 'Released'.

Discussion:

The alternative option might include the lease information in the prefix pool, then populate it to relay agent, make the state machine on the relay agent keep synchronizing the lease and status of the associated prefix pool with the server. But the solution proposed in this draft is to let relay agent confirm the received status of the prefix pool by itself as per the leases of delegated customer prefixes within it, and build its own lease for the prefix pool.

5. Relay Agent Behavior

The relay agent who needs the information of prefix pools, must include the associated requested-option-code in Option Request option (OPTION_ORO, 6) to request the Prefix Pool option (OPTION_PREFIX_POOL, [TBD]) from the DHCPv6 server, who maintains the status of the prefix pools associated to the relay agent itself (Figure 1) or its particular customer-facing interface (Figure 2), when receiving the DHCPv6-PD message from clients. The DHCPv6 relay agent can include this Option Request option for the Prefix Pool option in the relay-forward (12) message of SOLICIT (1), REQUEST (3), RENEW(5), REBIND (6) and RELEASE (8). The relay agent may also include the Prefix Pool option with the values of pfx-pool-len and IPv6-prefix to indicate its preference, which the prefix pool the relay agent would like the server to return.

The relay agent should include the Interface ID option (OPTION_INTERFACE_ID, 18) so that the DHCPv6 server can identify the relay agent itself or its particular customer-facing interface to which the prefix pool is associated, if the server would not like to use the link-address field specified in the encapsulation of the DHCPv6 relay-forward message to identify the interface of the link on which the clients are located.

The relay agent may set up a table for the leases or status of the prefix pools on it as per the delegated customer prefixes within it. The lease of the prefix pools must dynamically set to be the maximum lease of the delegated customer prefixes. If there is no route entry directing to the customer network within the aggregated route associated with the prefix pool, the relay agent shall automatically withdraw the aggregated route.

After receiving the Prefix Pool option for the relay agent itself or its particular customer-facing interface in the relay-reply message (13) of REPLY (7) from the DHCPv6 server, the relay agent acting as the PE router shall confirm the status of the prefix pool as per the leases of delegated customer prefixes within it, then add or withdraw the aggregated route entry per the status of the prefix pool. If the

status of the prefix pool received and confirmed is 'Active', the relay agent shall add an aggregated route entry in its routing table, if the same entry has not been added in before. If the status of the prefix pool received is 'Released', the relay agent shall withdraw the associated aggregated route entry in its routing table, if the same entry has not been withdrawn before.

The relay agent advertises its routing table including the entries of the aggregated routes based on the information of prefix pools when the routing protocol is enabled on its network-facing interface.

The Relay Agent (i.e., Requestor) can use the DHCPv6 Bulk Leasequery [RFC5460] to query the binding data of prefix pools in the 'Active' status from the server. After established a TCP connection with the DHCPv6 server, the relay agent must include Query option (OPTION_LQ_QUERY, 44) and set the proper query-type (QUERY_BY_RELAY_ID, QUERY_BY_LINK_ADDRESS, QUERY_BY_REMOTE_ID), link-address and query-options in the LEASEQUERY (14) message. The query options must include Option Request option (OPTION_ORO, 6) to request the Prefix Pool option (OPTION_PREFIX_POOL, [TBD]) from the server.

6. Server Behavior

Per DHCPv6-PD [RFC3633], if the prefix of the customer network requested in relay-forward (12) message of SOLICIT, REQUEST, RENEW, REBIND from the DHCPv6 client (i.e., the RR) has a valid lease, the DHCPv6 server (i.e., the DR) will delegate the prefix with the relevant parameters in the relay-reply (13) message of REPLY. In order to give a meaningful reply, the server has to be able to maintain the binding data of the delegated IPv6 prefixes with the identification of the client. The Interface ID option (OPTION_INTERFACE_ID, 18) nested in the relay-forward message is usually used to identify the access line of the client.

After receiving the Option Request option (OPTION_ORO, 6) requesting the Prefix Pool option (OPTION_PREFIX_POOL, [TBD]) in the relay-forward messages of the PD, the server must include the Prefix Pool option with the status indicated for the associated relay agent itself (Figure 1) or its customer-facing interface (Figure 2) in the relay-reply messages if the relay-forward messages received are valid.

The server may use the link-address specified in relay-forward message to identify the relay agent itself or its particular customer-facing interface where the prefix pool is associated, but the server has to maintain the binding data of prefix pools in association with these link-addresses. To be more readable, the

server can alternatively use the Interface ID option (OPTION_INTERFACE_ID, 18) included in the relay-forward message by the relay agent to identify the relay agent itself (Figure 1) or its particular customer-facing interface (Figure 2) where the prefix pool is associated. In order to give a meaningful reply, the server has to maintain the binding data of prefix pools in association with the information derived from the Interface ID option.

Per DHCPv6 [RFC3315], the server shall copy the same Interface ID option received via the relay-forward message into the relay-reply message.

If the administrative policy on the DHCPv6 server permits to support route aggregation on the relay agent for some particular prefix, the status of prefix pool can be determined by the delegated prefixes within the associated prefix pool. If there is at least one delegated prefix within the pool that has a valid lease, the server shall set the status of the associated prefix pool to be 'Active'. After the last prefix releasing in the associated prefix pool, the server shall set the status of the associated prefix pool to be 'Released'. If the administrative policy on the server does not permit to support route aggregation on the DHCPv6 relay agent, the server shall set the status of the prefix pools always to be 'Released'.

When the administrator of the server changes the setting to support route aggregation on the relay agent for the particular prefix pool, the status of the prefix pool may change from 'Released' to be 'Active' if at least one delegated prefix within the prefix pool has the valid lease. When the administrator of the server changes the setting not to support route aggregation on the relay agent for the particular prefix pool, the status of the prefix pool may change from 'Active' to be 'Released' if at least one delegated prefix within the prefix pool has the valid lease. Then the server may send a relay-reply message of RECONFIGURE (10) to initiate immediately a Renew (5) / Reply (7) PD message exchange with Prefix Pool option between one active client and the server.

Multiple prefix pools may be associated with the same PE router implementing a DHCPv6 relay agent (Figure 1) or its customer-facing interface (Figure 2) in the binding table on the server. Note that the delegated prefix is only from one prefix pool.

After receiving the LEASEQUERY (14) message from the relay agent with the Query option (OPTION_LQ_QUERY, 44) including the Option Request option (OPTION_ORO, 6) to request the Prefix Pool option (OPTION_PREFIX_POOL, [TBD]), the server must include the Client Data options (OPTION_CLIENT_DATA, 45) in the LEASEQUERY-REPLY (15) and

LEASEQUERY-DATA (16) message to convey the binding data of the associated prefix pools with the 'Active' status through the established TCP connection per [RFC5460]. Each Client Data option shall contain a Prefix Pool option, and may contain the Interface ID option (OPTION_INTERFACE_ID, 18). In order to be able to provide meaningful replies to different query types, the server has to be able to maintain the relevant association of prefix pools with the RELAY_ID, link addresses or Remote_ID of the relay agent in its binding database.

7. Security Considerations

Security issues related DHCPv6 are described in section 23 of [RFC3315].

8. IANA Considerations

IANA is requested to assign an option code to Option_Prefix_Pool from the "DHCPv6 and DHCPv6 options" registry (http://www.iana.org/assignments/dhcpv6-parameters/dhcpv6-parameters.xml).

9. Contributors List

Juergen Schoenwaelder
Jacobs University Bremen
Bremen
Germany

Email: j.schoenwaelder@jacobs-university.de

Tina Tsou
Huawei Technologies
Santa Clara, CA
USA

Email: tina.tsou.zouting@huawei.com

10. Acknowledgements

Thanks to Ralph Droms for the inspiration from his expired draft (RAAN option), to the DHC working group members, Bernie Volz, Ole Troan and Roberta Maglione for the discussion in the mailing list, to

Christian Jacquenet for pointing out the draft should cover one more use case of ISP-Customer network where CPE is directly connected to PE, to Sven Ooghe for some revisions in the email review, to Shrinivas Ashok Joshi for pointing out the draft should cover the robust mechanism against the case of reboot, to Adrian Farrel for the orientation guide on this draft in IETF80 at Prague.

11. References

11.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC3315] Droms, R., Bound, J., Volz, B., Lemon, T., Perkins, C., and M. Carney, "Dynamic Host Configuration Protocol for IPv6 (DHCPv6)", RFC 3315, July 2003.

[RFC3633] Troan, O. and R. Droms, "IPv6 Prefix Options for Dynamic Host Configuration Protocol (DHCP) version 6", RFC 3633, December 2003.

[RFC5007] Brzozowski, J., Kinnear, K., Volz, B., and S. Zeng, "DHCPv6 Leasequery", RFC 5007, September 2007.

[RFC5460] Stapp, M., "DHCPv6 Bulk Leasequery", RFC 5460, February 2009.

11.2. Informative References

[BBF TR-177] Broadband Forum, "IPv6 in the context of TR-101, Issue 1", November 2010.

[RFC3769] Miyakawa, S. and R. Droms, "Requirements for IPv6 Prefix Delegation", RFC 3769, June 2004.

Authors' Addresses

Leaf Y. Yeh (editor)
Huawei Technologies
Shenzhen,
P. R. China

Email: leaf.y.yeh@huawei.com

Mohamed Boucadair
France Telecom
Rennes,
France

Email: mohamed.boucadair@orange.com

Ted Lemon
Nominum, Inc
USA

Email: Ted.Lemon@nominum.com

Jie Hu
China Telecom
Beijing,
P. R. China

Email: huj@ctbri.com.cn