Dynamic Host Configuration
Internet-Draft
Updates: 2563 (if approved)
Intended status: Standards Track
Expires: December 19, 2020

L. Colitti
J. Linkova
Google
M. Richardson
Sandelman
T. Mrugalski
ISC

June 17, 2020

IPv6-Only-Preferred Option for DHCPv4
draft-ietf-dhc-v6only-03

Abstract

This document specifies a DHCPv4 option to indicate that a host supports an IPv6-only mode and willing to forgo obtaining an IPv4 address if the network provides IPv6 connectivity. It also updates RFC2563 to specify the DHCPv4 server behavior when the server receives a DHCPDISCOVER not containing the Auto-Configure 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 https://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 December 19, 2020.

Copyright Notice

Table of Contents

1. Introduction

One of the biggest challenges of deploying IPv6-only LANs is that such networks might contain rather heterogeneous collection of hosts. While some hosts are capable of operating in IPv6-only mode (either because the OS and all applications are IPv6-only capable or because the host has some form of 464XLAT [RFC6877] deployed), others might still have IPv4 dependencies and need IPv4 addresses to operate properly. To incrementally rollout IPv6-only, network operators might need to provide IPv4 on demand whereby a host receives an IPv4

address if it needs it, while IPv6-only capable hosts (such as modern mobile devices) are not allocated IPv4 addresses. Traditionally that goal is achieved by placing IPv6-only capable devices into a dedicated IPv6-only network segment or WiFi SSID, while dual-stack devices reside in another network with IPv4 and DHCPv4 enabled. However such approach has a number of drawbacks, including but not limited to:

- Doubling the number of network segments leads to operational complexity and performance impact, for instance due to high memory utilization caused by an increased number of ACL entries.

- Placing a host into the correct network segment is problematic. For example, in the case of 802.11 Wi-Fi the user might select the wrong SSID. In the case of wired 802.1x authentication the authentication server might not have all the information required to make the correct decision and choose between an IPv6-only and a dual-stack VLAN.

It would be beneficial for IPv6 deployment if operators could implement IPv6-mostly (or IPv4-on-demand) segments where IPv6-only hosts co-exist with legacy dual-stack devices. The trivial solution of disabling IPv4 stack on IPv6-only capable hosts is not feasible as those clients must be able to operate on IPv4-only networks as well. While IPv6-only capable devices might use a heuristic approach to learning if the network provides IPv6-only functionality and stop using IPv4 if it does, such approach might be practically undesirable. One important reason is that when a host connects to a network, it does not know if the network is IPv4-only, dual-stack or IPv6-only. To ensure that the connectivity over whatever protocol is present becomes available as soon as possible the host usually starts configuring both IPv4 and IPv6 immediately. If hosts were to delay requesting IPv4 until IPv6 reachability is confirmed, that would penalize IPv4-only and dual-stack networks, which does not seem practical. Requesting IPv4 and then releasing it later, after IPv6 reachability is confirmed, might cause user-visible errors as it would be disruptive for applications which have started using the assigned IPv4 address already. Instead it would be useful to have a mechanism which would allow a host to indicate that its request for an IPv4 address is optional and a network to signal that IPv6-only functionality (such as NAT64, [RFC6146]) is available. The proposed solution is to introduce a new DHCPv4 option which a client uses to indicate that it does not need an IPv4 address if the network provides IPv6-only connectivity (as NAT64 and DNS64). If the particular network segment provides IPv4-on-demand such clients would not be supplied with IPv4 addresses, while on IPv4-only or dual-stack segments without NAT64 services IPv4 addresses will be provided.

[RFC2563] introduces the Auto-Configure DHCPv4 option and describes DHCPv4 servers behavior if no address is chosen for a host. This document updates [RFC2563] to modify the server behavior if the DHCPOFFER contains the IPv6-only Preferred option.

1.1. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

1.2. Terminology

IPv6-only capable host: a host which does not require an IPv4 address and can operate on IPv6-only networks. Strictly speaking IPv6-only capability is specific to a given interface of the host: if some application on a host require IPv4 and 464XLAT CLAT [RFC6877] is only enabled on one interface, the host is IPv6-only capable if connected to a NAT64 network via that interface.

IPv4-requiring host: a host which is not IPv6-only capable and can not operate in IPv6-only network providing NAT64 service.

IPv4-on-demand: a deployment scenario when end hosts are expected to operate in IPv6-only mode by default and IPv4 addresses can be assigned to some hosts if those hosts explicitly opt-in to receiving IPv4 addresses.

IPv6-mostly network: a network which provides NAT64 (possibly with DNS64) service as well as IPv4 connectivity and allows coexistence of IPv6-only, dual-stack and IPv4-only hosts on the same segment. Such deployment scenario allows operators to incrementally turn off IPv4 on end hosts, while still providing IPv4 to devices which require IPv4 to operate. But, IPv6-only capable devices need not be assigned IPv4 addresses.

IPv6-Only network: a network which does not provide routing functionality for IPv4 packets. Such networks may or may not allow intra-LAN IPv4 connectivity. IPv6-Only network usually provides access to IPv4-only resources via NAT64 [RFC6146].

NAT64: Network Address and Protocol Translation from IPv6 Clients to IPv4 Servers [RFC6146].

RA: Router Advertisement, a message used by IPv6 routers to advertise their presence together with various link and Internet parameters [RFC4861].

DNS64: a mechanism for synthesizing AAAA records from A records [RFC6147].

2. Reasons to Signal IPv6-Only Support in DHCPv4 Packets

For networks which contain both IPv6-only capable and IPv4-requiring hosts and utilize DHCPv4 for configuring the IPv4 network stack on hosts, it seems only natural to leverage the same protocol to signal that IPv4 is discretional on a given segment. An ability to remotely disable IPv4 on a host can be seen as a new denial-of-service attack vector. The proposed approach limits the attack surface to DHCPv4-related attacks without introducing new vulnerable elements.

Another benefit of using DHCPv4 for signaling is that IPv4 will be disabled only if both the client and the server indicate IPv6-only capability. It allows IPv6-only capable hosts to turn off IPv4 only upon receiving an explicit signal from the network and operate in dual-stack or IPv4-only mode otherwise. In addition, the proposed mechanism does not introduce any additional delays to the process of configuring IP stack on hosts. If the network does not support IPv6-only/IPv4-on-demand mode, an IPv6-only capable host would configure an IPv4 address as quickly as on any other host.

Being a client/server protocol, DHCPv4 allows IPv4 to be selectively disabled on a per-host basis on a given network segment. Coexistence of IPv6-only, dual-stack and even IPv4-only hosts on the same LAN would not only allow network administrators to preserve scarce IPv4 addresses but would also drastically simplify incremental deployment of IPv6-only networks, positively impacting IPv6 adoption.

3. IPv6-Only Preferred Option

3.1. Option format

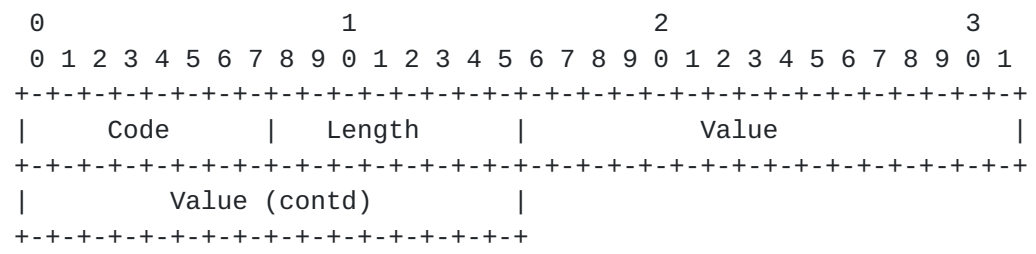

Figure 1: IPv6-Only Preferred Option Format

Fields:

Code: 8-bit identifier of the IPv6-Only Preferred option code as assigned by IANA: TBD

Length: 8-bit unsigned integer. The length of the option excluding the Code and Length Fields. The server MUST set the length field to 4. The client MUST ignore the IPv6-Only Preferred option if the length field value is not 4.

Value: 32-bit unsigned integer.
The number of seconds the client should disable DHCPv4 for (V6ONLY_WAIT configuration variable).
If the server pool is explicitly configured with a V6ONLY_WAIT timer the server MUST set the field to that configured value. Otherwise the server MUST set it to zero.
The client MUST process that field as described in Section 3.2.

3.2. DHCPv4 Client Behavior

A DHCPv4 client SHOULD allow a device administrator to configure IPv6-only preferred mode either for a specific interface (to indicate that the device is IPv6-only capable if connected to a NAT64 network via that interface) or for all interfaces. If only a specific interface is configured as IPv6-only capable the DHCPv4 client MUST NOT consider the host to be an IPv6-only capable for the purpose of sending/receiving DHCPv4 packets over any other interfaces.

The DHCPv4 client on an IPv4-requiring host MUST NOT include the IPv6-only Preferred option in the Parameter Request List of any DHCPv4 packets and MUST ignore that option in packets received from DHCPv4 servers.

DHCPv4 clients running on IPv6-only capable hosts SHOULD include the IPv6-only Preferred option code in the Parameter Request List in DHCPDISCOVER and DHCPREQUEST messages for interfaces so enabled and follow the processing as described below on a per interface enabled basis.

If the client did not include the IPv6-only Preferred option code in the Parameter Request List option in the DHCPDISCOVER or DHCPREQUEST message it MUST ignore the IPv6-only Preferred option in any messages received from the server.

If the client includes the IPv6-only Preferred option in the Parameter Request List and the DHCPOFFER message from the server contains a valid IPv6-only Preferred option, the client SHOULD NOT request the IPv4 address provided in the DHCPOFFER. If the IPv6-only Preferred option returned by the server contains a value greater or equal to MIN_V6ONLY_WAIT, the client SHOULD set the V6ONLY_WAIT timer to that value. Otherwise, the client SHOULD set the V6ONLY_WAIT timer to MIN_V6ONLY_WAIT. The client SHOULD stop the DHCPv4 configuration process for at least V6ONLY_WAIT seconds or until a network attachment event happens. The host MAY disable the IPv4 stack completely for V6ONLY_WAIT seconds or until the network disconnection event happens.

The IPv6-only Preferred option SHOULD be included in the Parameter Request List option in DHCPREQUEST messages (after receiving a DHCPOFFER without this option, for a INIT-REBOOT, or when renewing or rebinding a leased address). If the DHCPv4 server responds with a DHCPACK that includes the IPv6-only Preferred option, the client MAY send a DHCPRELEASE message and MAY either stop the DHCPv4 configuration process or disable IPv4 stack completely for V6ONLY_WAIT seconds or until the network disconnection event happens. Alternatively the client MAY continue to use the assigned IPv4 address until further DHCPv4 reconfiguration events.

If the client includes the IPv6-only Preferred option in the Parameter Request List and the server responds with DHCPOFFER message without a valid IPv6-only Preferred option, the client MUST proceed as normal with a DHCPREQUEST.

If the client waits for multiple DHCPOFFER responses and selects one of them, it MUST follow the processing for the IPv6-only Preferred option based on the selected response. A client MAY use the presence of the IPv6-only Preferred option as a selection criteria.

When an IPv6-only capable client receives the IPv6-Only Preferred option from the server, the client MAY configure IPv4 link-local address [RFC3927]. In that case IPv6-Only capable devices might still be able to communicate over IPv4 to other devices on the link. The Auto-Configure Option [RFC2563] can be used to control IPv4 link-local addresses autoconfiguration. Section 3.3.1 discusses the interaction between the IPv6-only Preferred and the Auto-Configure options.

3.3. DHCPv4 Server Behavior

The DHCPv4 server SHOULD be able to configure certain pools to include the IPv6-only preferred option in DHCPv4 responses if the client included the option code in the Parameter Request List option. The DHCPv4 server MAY have a configuration option to specify V6ONLY_WAIT timer for all or individual IPv6-mostly pools.

The server MUST NOT include the IPv6-only Preferred option in the DHCPOFFER or DHCPACK message if the YIADDR field in the message does not belong to a pool configured as IPv6-mostly. The server MUST NOT include the IPv6-only Preferred option in the DHCPOFFER or DHCPACK message if the option was not present in the Parameter Request List sent by the client.

If the IPv6-only Preferred option is present in the Parameter Request List received from the client and the corresponding DHCPv4 pool is explicitly configured as belonging to an IPv6-mostly network segment, the server MUST include the IPv6-only Preferred option when responding with the DHCPOFFER or DHCPACK message. If the server responds with the IPv6-only Preferred option and the V6ONLY_WAIT timer is configured for the pool, the server MUST copy the configured value to the IPv6-only Preferred option value field. Otherwise it MUST set the field to zero. The server SHOULD NOT assign an address for the pool. Instead it SHOULD return 0.0.0.0 as the offered address. Alternatively, the server MAY include an available IPv4 address from the pool into the DHCPOFFER as per recommendations in [RFC2131]. In this case, the offered address MUST be a valid address that is not committed to any other client. Because the client is not expected ever to request this address, the server SHOULD NOT reserve the address and SHOULD NOT verify its uniqueness. If the client then issues a DHCPREQUEST for the address, the server MUST process it per [RFC2131], including replying with a DHCPACK for the address if in the meantime it has not been committed to another client.

If a client includes both a Rapid-Commit option [RFC4039] and IPv6-Only Preferred option in the DHCPDISCOVER message the server SHOULD NOT honor the Rapid-Commit option if the response would contain the IPv6-only Preferred option to the client. It SHOULD instead respond with a DHCPOFFER as indicated above.

3.3.1. Interaction with RFC2563

[RFC2563] defines an Auto-Configure DHCPv4 option to disable IPv4 link-local address configuration for IPv4 clients. Clients can support both, neither or just one of IPv6-Only Preferred and Auto-Configure options. If a client sends both IPv6-Only Preferred and Auto-Configure options the network administrator can prevent the host

from configuring an IPv4 link-local address on IPv6-mostly network. To achieve this the server needs to send DHCPOFFER which contains a 'yiaddr' of 0x00000000, and the Auto-Configure flag saying "DoNotAutoConfigure".

However special care should be taken in a situation when a server supports both options and receives just IPv6-Only Preferred option from a client. Section 2.3 of [RFC2563] states that if no address is chosen for the host (which would be the case for IPv6-only capable clients on IPv6-mostly network) then: "If the DHCPDISCOVER does not contain the Auto-Configure option, it is not answered." Such behavior would be undesirable for clients supporting the IPv6-Only Preferred option w/o supporting the Auto-Configure option as they would not receive any response from the server and would keep asking, instead of disabling DHCPv4 for V6ONLY_WAIT second. Therefore the following update is proposed to Section 2.3 of [RFC2563]"

OLD TEXT:

However, if no address is chosen for the host, a few additional steps MUST be taken.

If the DHCPDISCOVER does not contain the Auto-Configure option, it is not answered.

NEW TEXT:

However, if no address is chosen for the host, a few additional steps MUST be taken.

If the DHCPDISCOVER does not contain the Auto-Configure option and the IPv6-Only Preferred option is not present, it is not answered. If the DHCPDISCOVER does not contain the Auto-Configure option but contains the IPv6-Only Preferred option, the processing rules for the IPv6-Only Preferred option apply.

3.4. Constants and Configuration Variables

V6ONLY_WAIT	The minimum time the client SHOULD stop the DHCPv4 configuration process for. MUST be no less than MIN_V6ONLY_WAIT seconds. Default: 1800 seconds
MIN_V6ONLY_WAIT	The lower boundary for V6ONLY_WAIT. Value: 300 seconds

4. IPv6-Only Transition Technologies Considerations

Until IPv6 adoption in the Internet reaches 100%, communication between an IPv6-only host and IPv4-only destination requires some form of transition mechanism deployed in the network. At the time of writing, the only such mechanism that is widely supported by end hosts is NAT64 [RFC6146] (either with or without 464XLAT). Therefore the IPv6-only Preferred option is only sent by hosts capable of operating on NAT64 networks. In a typical deployment scenario, a network administrator would not configure the DHCPv4 server to return the IPv6-only Preferred option unless the network provides NAT64 service.

Hypothetically it is possible for multiple transition technologies to coexist. In such scenario some form of negotiation would be required between a client and a server to ensure that the transition technology supported by the client is the one the network provides. However it seems unlikely that any new transition technology would arise and be widely adopted in any foreseeable future. Therefore adding support for non-existing technologies seems to be suboptimal and the proposed mechanism implies that NAT64 is used to facilitate connectivity between IPv6 and IPv4.

It should be also noted that declaring a host or (strictly speaking, a host interface) IPv6-only capable is a policy decision. For example,

- An operating system vendor may make such decision and configure their DHCPv4 clients to send the IPv6-Only Preferred option by default if the OS has 464XLAT CLAT [RFC6877] enabled.

- An enterprise network administrator may provision the corporate hosts as IPv6-only capable if all applications users are supposed to run have been tested in IPv6-only environment (or if 464XLAT CLAT is enabled on the devices).

- IoT devices may be shipped in IPv6-only capable mode if they are designed to connect to IPv6-enabled cloud destination only.

5. IANA Considerations

The IANA is requested to assign a new DHCPv4 Option code for the IPv6-Only Preferred option from the BOOTP Vendor Extensions and DHCPv4 Options registry, located at https://www.iana.org/assignments/bootp-dhcp-parameters/bootp-dhcp-parameters.xhtml#options . If possible, please assign option code 108.

Option Name	Code
IPv6-only Preferred option	(TBD)

Table 1

6. Security Considerations

The proposed mechanism is not introducing any new security implications. While clients using the IPv6-only Preferred option are vulnerable to attacks related to a rogue DHCPv4 server, enabling IPv6-only Preferred option does not provide an attacker with any additional mechanisms.

It should be noted that disabling IPv4 on a host upon receiving the IPv6-only Preferred option from the DHCPv4 server protects the host from IPv4-related attacks and therefore could be considered a security feature.

7. Acknowledgements

Thanks to the following people (in alphabetical order) for their review and feedback: Mohamed Boucadair, Ted Lemon, Roy Marples, Bjorn Mork, Peng Shuping, Bernie Volz, Eric Vyncke. Authors would like to thank Bob Hinden and Brian Carpenter for the initial idea of signaling IPv6-only capability to hosts. Special thanks to Erik Kline, Mark Townsley and Maciej Zenczykowski for the discussion which led to the idea of signalling IPv6-only capability over DHCPv4.

8. References

8.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

[RFC2131] Droms, R., "Dynamic Host Configuration Protocol", RFC 2131, DOI 10.17487/RFC2131, March 1997, <https://www.rfc-editor.org/info/rfc2131>.

[RFC2563] Troll, R., "DHCP Option to Disable Stateless Auto-Configuration in IPv4 Clients", RFC 2563, DOI 10.17487/RFC2563, May 1999, <https://www.rfc-editor.org/info/rfc2563>.

[RFC3927] Cheshire, S., Aboba, B., and E. Guttman, "Dynamic Configuration of IPv4 Link-Local Addresses", RFC 3927, DOI 10.17487/RFC3927, May 2005, <https://www.rfc-editor.org/info/rfc3927>.

[RFC4039] Park, S., Kim, P., and B. Volz, "Rapid Commit Option for the Dynamic Host Configuration Protocol version 4 (DHCPv4)", RFC 4039, DOI 10.17487/RFC4039, March 2005, <https://www.rfc-editor.org/info/rfc4039>.

[RFC4861] Narten, T., Nordmark, E., Simpson, W., and H. Soliman, "Neighbor Discovery for IP version 6 (IPv6)", RFC 4861, DOI 10.17487/RFC4861, September 2007, <https://www.rfc-editor.org/info/rfc4861>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

8.2. Informative References

[RFC6146] Bagnulo, M., Matthews, P., and I. van Beijnum, "Stateful NAT64: Network Address and Protocol Translation from IPv6 Clients to IPv4 Servers", RFC 6146, DOI 10.17487/RFC6146, April 2011, <https://www.rfc-editor.org/info/rfc6146>.

[RFC6147] Bagnulo, M., Sullivan, A., Matthews, P., and I. van Beijnum, "DNS64: DNS Extensions for Network Address Translation from IPv6 Clients to IPv4 Servers", RFC 6147, DOI 10.17487/RFC6147, April 2011, <https://www.rfc-editor.org/info/rfc6147>.

[RFC6877] Mawatari, M., Kawashima, M., and C. Byrne, "464XLAT: Combination of Stateful and Stateless Translation", RFC 6877, DOI 10.17487/RFC6877, April 2013, <https://www.rfc-editor.org/info/rfc6877>.

Authors' Addresses

Lorenzo Colitti
Google
Shibuya 3-21-3
Shibuya, Tokyo 150-0002
JP

Email: lorenzo@google.com

Jen Linkova
Google
1 Darling Island Rd
Pyrmont, NSW 2009
AU

Email: furry@google.com

Michael C. Richardson
Sandelman Software Works

Email: mcr+ietf@sandelman.ca
URI: http://www.sandelman.ca/

Tomek Mrugalski
Internet Systems Consortium, Inc.
950 Charter Street
Redwood City, CA 94063
USA

Email: tomasz.mrugalski@gmail.com