

IPv6 operations  
Internet-Draft  
Intended status: Informational  
Expires: 23 April 2026

N. Buraglio  
Energy Sciences Network  
O. Caletka  
RIPE NCC  
J. Linkova  
Google  
20 October 2025

IPv6-Mostly Networks: Deployment and Operations Considerations  
draft-ietf-v6ops-6mops-04

## Abstract

This document discusses a deployment scenario called "an IPv6-Mostly network", when IPv6-only and IPv4-enabled endpoints coexist on the same network (network segment, VLAN, SSID etc).

## 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 23 April 2026.

## Copyright Notice

Copyright (c) 2025 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Revised BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Revised BSD License.

## Table of Contents

1. Introduction . . . . .	3
2. Requirements Language . . . . .	4
3. Terminology . . . . .	4
4. Solution Overview . . . . .	5
4.1. IPv6-only capable Endpoints . . . . .	6
4.2. IPv6-Only and IPv4-enabled Endpoints Coexistence . . . . .	6
4.3. Access to IPv4-only Destinations . . . . .	7
4.3.1. NAT64 . . . . .	7
4.3.2. 464XLAT . . . . .	7
4.3.3. Signalling NAT64 Prefix to Hosts . . . . .	8
4.3.4. Signalling DNS and DNS64 to Hosts . . . . .	8
5. Solution Analysis . . . . .	9
5.1. IPv6-Only Compared to Dual-Stack . . . . .	9
5.2. IPv6-Mostly Compared to a Dedicated IPv6-Only Network . . . . .	11
6. Incremental Rollout Considerations . . . . .	12
6.1. Opt-In and Opt-Out Modes . . . . .	12
6.2. Per-Device and Per-Subnet Incremental Rollout . . . . .	13
6.3. Rollback Approach . . . . .	13
7. Operational Considerations . . . . .	14
7.1. Address Assignment Policy . . . . .	14
7.2. Extension Headers . . . . .	14
7.3. Onlink Communication and Service Discovery . . . . .	14
7.4. NAT64 IPv4 Pools . . . . .	15
7.5. Typical Issues . . . . .	15
7.5.1. Hosts with Disabled or Disfunctional IPv6 . . . . .	15
7.5.2. Network Extension . . . . .	16
7.5.3. Multiple Addresses per Device . . . . .	16
7.5.4. Host Mobility and Renumbering . . . . .	17
7.5.5. Fragmentation . . . . .	18
7.5.6. IPv4 UDP Packets with Zero Checksum . . . . .	19
7.5.7. Representing IPv6 Addresses by CLAT . . . . .	19
7.5.8. IPv4-Dependencies in Network Admission Control . . . . .	20
7.5.9. Custom DNS Configuration on Endpoints . . . . .	20
8. Security Considerations . . . . .	20
9. Privacy Considerations . . . . .	21
10. IANA Considerations . . . . .	21
11. References . . . . .	21
11.1. Normative References . . . . .	21
11.2. Informative References . . . . .	23
Acknowledgements . . . . .	25
Authors' Addresses . . . . .	25

## 1. Introduction

Most network operators initially deploy IPv6 alongside their existing IPv4 infrastructure, as the dual-stack approach is seen as a necessary transition phase, allowing operators to gain experience with IPv6 while minimizing disruption. Pure IPv6-only networks, where endpoints are not assigned IPv4 addresses and access to IPv4 destination is provided via some form of address family translation (such as NAT64 [RFC6146]) remain uncommon outside of the mobile carrier space.

However, dual-stack networks do not address the core problem driving IPv6 adoption: IPv4 address exhaustion. They still require the same amount of IPv4 resources as IPv4-only networks. Even worse, this dual-stack approach has been demonstrated to be a long-term crutch, frequently masking problems that would otherwise be exposed and remedied as normal operational workflows.

The solution is to stop assigning IPv4 addresses to endpoints, or, in other words, to start deploying IPv6-only networks. To ensure that IPv6-only endpoints can communicate with IPv4-only ones, IPv6-only networks provides translation services, such as NAT64 [RFC6146] and SIIT [RFC7755]. However, even with those translation mechanisms in place, many systems and applications retain deep IPv4 dependencies, requiring both IPv4 address assignment to the host and the network to provide IPv4 routing functionality.

The less control a network operator has over devices and applications, the more difficult it is to eliminate IPv4 dependencies and move to IPv6-only. This is particularly challenging in enterprise networks with legacy IPv4-dependent applications and public Wi-Fi networks where operators cannot guarantee device compatibility. As a result, a chicken-and-egg problem arises: IPv6-only networks seem impractical with so many incompatible applications, yet applications continue to rely on IPv4 because IPv6-only networks are rare.

To enable a gradual transition from dual-stack to IPv6-only, operators need to identify which devices can function in IPv6-only mode and which cannot. This still leaves the challenge of how to provide IPv4 addresses only to devices which can not operate without them. Creating separate network segments for each type of devices introduces complexity and scalability issues a major hurdle to IPv6-only adoption.

A more desirable approach is to deploy an "IPv6-mostly" network that provides IPv4 on demand. This allows IPv6-capable devices to remain IPv6-only while the network is seamlessly supplying IPv4 to those

that require it. An IPv6-mostly network allows endpoints to operate at the highest level of their network stack evolution, on demand, while still allowing for legacy compatibility.

This document explores the requirements, recommendations, and challenges associated with deploying IPv6-mostly networks in enterprise and public Wi-Fi environments. While the principles discussed may be applicable to other network types, this document's focus remains on these specific use cases.

## 2. 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.

## 3. Terminology

This document reuses most of Terminology section from [RFC8925] .

CLAT: a customer-side translator (CLAT) as specified in [RFC6877] and [I-D.ietf-v6ops-claton].

Endpoint: A device connected to a network and considered a host From the operator's perspective. However, some endpoint can also extend the network to other physical or logical systems, thereby assuming routing functions. Examples include:

- \* Personal computers: While primarily a host, such devices might run virtual systems/containers, and route traffic to them, extending the network and acting as routers.
- \* Mobile phone with tethering enabled: Acts as a host on the Wi-Fi network, but also as a router for tethered devices, potentially without the operator's knowledge or consent.

NAT44: Network Address Translation from IPv4 to IPv4.

Native IPv4 connectivity: IPv4 connectivity or default gateway provided by the network without using any form of translation mechanisms (such as 464XLAT, [RFC6877]).

Network segment: a link (VLAN, a broadcast domain etc) where hosts share the same IP subnet.

PREF64: (or NAT64 prefix): An IPv6 prefix used for IPv6 address synthesis and for network addresses and protocols translation from IPv6 clients to IPv4 servers, [RFC6146].

#### 4. Solution Overview

In a nutshell, an IPv6-mostly network is very similar to a dual-stack one with two additional key elements:

- \* The network provides NAT64 ([RFC6146] ) functionality, enabling IPv6-only clients to communicate with IPv4-only destinations.
- The network also provides the information about the NAT64 prefix (PREF64), for example via RAs ([RFC8781]), via DNS64 ([RFC6147], [RFC7050]), or both. This is to ensure that clients and the network's NAT64 use the same PREF64 to translate between IPv6 and IPv4. Section 4.3.3 and Section 4.3.4 discuss those mechanisms in more detail.
- \* The DHCPv4 server infrastructure offers DHCPv4 Option 108 as per [RFC8925]. This is to ensure that IPv6-only capable devices are not consuming IPv4 addresses. Section 4.2 discusses other approaches to provide IPv4 addresses on demand.

Upon connecting to an IPv6-mostly network segment, an endpoint configures its IP stack based on its capabilities:

- \* IPv4-Only Endpoint: Acquires an IPv4 address through DHCPv4.
- \* Dual-Stack Endpoint (Not IPv6-only capable): Configures IPv6 addresses using any supported protocol. Additionally, it obtains an IPv4 address via DHCPv4.
- \* IPv6-only capable endpoint configures its IPv6 addresses and, while performing DHCPv4, includes option 108 ([RFC8925] ) into the Parameter Request List. The DHCP server returns the option and, as per [RFC8925] , the endpoint forgoes requesting an IPv4 address, remaining in IPv6-only mode.

An IPv6-mostly network segment can support a mix of IPv4-only, dual-stack, and IPv6-only devices. IPv6-only endpoints utilize the network-provided NAT64 to reach IPv4-only destinations.

The following sections discussed those solution elements in more details.

#### 4.1. IPv6-only capable Endpoints

The term "IPv6-only capable endpoint" lacks a strict technical definition. It broadly describes a device that can function without native IPv4 connectivity/IPv4 addresses, providing the same user experience as if native IPv4 connectivity were present. Examples include but are not limited to:

- \* An endpoint capable of communicating only with a limited set of endpoints, all of which are available over IPv6.
- \* An endpoint capable of utilizing only IPv6 sockets, employing NAT64 for accessing IPv4 resources.
- \* An endpoint implementing CLAT to provide IPv4 connectivity for the legacy applications.

#### 4.2. IPv6-Only and IPv4-enabled Endpoints Coexistence

One effective way to restrict IPv4 addresses solely to devices that require them is to enable support for the IPv6-Only Preferred DHCPv4 option (Option 108, [RFC8925]) on the network's DHCP infrastructure. As most IPv6-only capable devices support Option 108 in their DHCP clients, enabling Option 108 on the server side would allow those devices to cease requesting IPv4 addresses.

Certain devices, such as resource-constrained embedded systems, may operate in IPv6-only mode without specific adjustments if their communication is limited to IPv6-enabled destinations. Since these systems often lack Option 108 support, administrators may need alternative methods to prevent IPv4 address assignment. One approach is to block IPv4 traffic at the network attachment level (such as a switchport or wireless access point). This could involve methods like:

- \* Static ACL: Applying a static filter with a "deny ip any any" rule.
- \* Dynamic ACL via RADIUS: If 802.1x authentication is in use, RADIUS can provide an ACL blocking all IPv4 traffic.
- \* Filtering the IPv4 ethertype (0x0800): Some hardware platforms may allow for low level filtering of ethertype 0x0800 preventing IPv4 from passing into or out of a given switchport.

The ACL-based approach has some scalability implications and increases operational complexity, therefore it could only be recommended as a stopgap solution.

### 4.3. Access to IPv4-only Destinations

#### 4.3.1. NAT64

IPv6-only endpoints require NAT64 to access IPv4-only destinations. Quite often operators choose to combine NAT44 and NAT64 functions, deploying NAT64 at the Internet edge. However, if not all internal services are IPv6-enabled, then NAT64 is needed to access internal destinations as well as external ones. In that case it might be beneficial to deploy NAT64 closer to the users, instead of the network Internet edge, in order to simplify routing/firewall rules and reduce latency.

If IPv4-only internal destinations are using [RFC1918] address space and IPv6-only nodes are expected to communicate with those IPv4-only destinations, then the operator **MUST NOT** use the well-known prefix 64:ff9b::/96 for NAT64 (see section 3.1 of [RFC6052]). In that scenario the operator is required to allocate a network-specific NAT64 prefix from the operator's address space. ).

Discussion: reports from the field indicate that it's very common for NAT64 and CLAT implementations to ignore the requirement from section 3.1 of [RFC6052]. Such implementations successfully translate [RFC1918] IPv4 addresses using the well-known NAT64 prefix.

#### 4.3.2. 464XLAT

Unless the usage of IPv4-only network sockets is prohibited or mitigated (e.g. by implementing translation mechanisms such as BIH [RFC6535]) in the operating system, enabling CLAT (Customer-side translator) on endpoints is essential for seamless operation of IPv4-only applications in IPv6-only environments. CLAT provides an IPv4 address and IPv4 default route, ensuring functionality even without a native IPv4 address from the network. Without CLAT or similar compatibility layer inside the operating system, IPv4-only applications would fail, negatively impacting user experience and increasing support overhead.

Recommendations for Network Administrators controlling the endpoints:

- \* CLAT + DHCPv4 Option 108: If the network administrator can control endpoint configuration, CLAT **SHOULD** be enabled on endpoints sending DHCPv4 Option 108. This streamlines the transition.
- \* Option 108 Without CLAT **MAY** be enabled if the administrator aims to identify IPv4-only systems/applications, or if all applications are confirmed to work in IPv6-only mode.

#### 4.3.3. Signalling NAT64 Prefix to Hosts

Hosts running 464XLAT, translating IPv4 literal to an IPv6 literals (Section 7.1 of [RFC8305]) or performing local DNS64 functions need to discover the NAT64 prefix. The network administrator SHOULD configure the first-hop routers to include PREF64 information in Router Advertisements as per ([RFC8781] ) even if the network provides DNS64 (so hosts can use DNS64-based prefix discovery, [RFC7050]). [RFC9872] discusses this recommendation in more details.

#### 4.3.4. Signalling DNS and DNS64 to Hosts

To provide IPv6-only endpoints with DNS recursive server addresses, the administrator MUST configure the network to include the DNS servers IPv6 addresses into RA option, as per [RFC8106].

Traditionally, DNS64 (with NAT64) is used to enable IPv6-only endpoints to access IPv4-only destinations. However, using DNS64 has a number of drawbacks, such as:

- \* DNSSEC Incompatibility: DNS64 responses fail DNSSEC validation.
- \* Custom Resolvers: Endpoints or applications configured with custom resolvers or running recursive resolvers locally can not benefit from the DNS64 provided by the network. Such systems would need to rely on CLAT or local synthesis instead (Section 7.5.9 discusses this scenario in more details).
- \* Additional requirements for application: to benefit from DNS64, applications need to be IPv6-enabled, use DNS (do not use IPv4 literals). Many applications do not satisfy those requirements and therefore fail if the endpoint does not have an IPv4 address/native IPv4 connectivity. Existence of such applications is the main reason why transition to IPv6-only must be incremental (via IPv6-mostly phase) and requires the vast majority of endpoints to support CLAT.

Those concerns make DNS64 is suboptimal and undesirtable solution long-term. To eliminate the needs for DNS64, the network shall signal PREF64 to endpoints (see Section 4.3.3), and the endpoints need to use the obtained PREF64 for performing local synthesis and for CLAT. It should be noted that not every application can benefit from local synthesis performed by the operating system, as it would require the application to use DNS (not IP literals). Therefore DNS64 is only required to support the following cases:

- \* Systems and applications which perform local synthesis but do not support [RFC8781] for prefix discovery, and can only discover the NAT64 prefix using via DNS64 [RFC7050].
- \* IPv6-only devices without CLAT (unless those endpoints are guaranteed never to need IPv4-only destinations, e.g. in case of a specialized network segment communicating solely with IPv6-capable destinations).

On the other hand, using DNS64 bypasses CLAT for IPv6-capable applications communicating with IPv4-only destinations, making the communication slightly more efficient. Additionally, it encourages dual-stack endpoints to utilize the IPv6 and NAT64 path to reach IPv4-only destinations. This operational pattern is useful for identifying IPv6-related issues and IPv4 dependencies at an earlier stage of the IPv6-mostly migration. For that reason administrators might find it beneficial to run DNS64 even if it is not strictly necessary, provided the the above mentioned drawbacks are addressed.

## 5. Solution Analysis

### 5.1. IPv6-Only Compared to Dual-Stack

IPv6-Mostly networks offer significant advantages over traditional dual-stack models where endpoints have both IPv4 and IPv6 addresses:

- \* The deployment of dual-stack networks does not fundamentally resolve the core issue of IPv4 address exhaustion. The IPv6-Mostly approach, however, offers a mechanism to significantly reduce the IPv4 address space allocated to endpoints and reclaim existing IPv4 space. The degree of possible reduction is dependent on endpoint capabilities, specifically support of DHCPv4 Option 108 and CLAT. Real-world IPv6-Mostly deployments (2025 data) show that approximately 60-70% of endpoints support IPv6-only operation. This capability potentially allows for the use of IPv4 subnets that are 2 to 4 times smaller than those required for dual-stack. It is important to note that migrating a dual-stack network to an IPv6-only mode does not reduce the amount of public IPv4 space required for network elements such as NAT pools. However, for large networks (e.g., enterprises and datacenters) facing private IPv4 address exhaustion, the IPv6-Mostly strategy drastically reduces internal IPv4 demand, helping to ensure business continuity.
- \* Controlled and incremental phase-out of IPv4: IPv6-Mostly allows for the controlled phase out IPv4 from many endpoints, streamlining operations and improving overall network reliability at a measured and operator-controlled pace. While IPv6-Mostly

deployments do not inherently resolve the IPv4 address exhaustion issue (as previously noted), the approach effectively paves the road toward IPv6-only networks.

- \* **Reduced Dependency on DHCPv4:** As more devices operate seamlessly in IPv6-only mode, the criticality of DHCPv4 service diminishes significantly. This allows operators to scale down DHCPv4 infrastructure or, in some cases, even operate it with less stringent service level objectives (SLOs), optimizing costs and resource allocation.
- \* **Simplified troubleshooting due reduced impact of Happy Eyeballs [RFC8305]:** when both the client and the server are dual-stack, communication can happen either over IPv6 or over IPv4 and the protocol choice can change over time. This may obscure network issues or make them intermittent, complicating the troubleshooting. In IPv6-Mostly network IPv6-only hosts are much less affected by such an issue. Although modern versions of Happy Eyeballs algorithm support falling back to IPv4 even in case of IPv6-only network, the delay before switching to IPv4 is so long that the fallback does not happen during regular operations. It should be noted, however, that when an IPv6-only client communicates with an IPv4-only destination, the traffic may traverse CLAT or be sent as IPv6 towards the NAT64, depending on how the specific application is written.

The introduction of NAT64 within the IPv6-Mostly model may appear as a drawback, as it inherits many of the limitations associated with NAT44, such as translating flows from different hosts to the same public IPv4 address, or scalability challenges caused by stateful operation. However, it's important to recognize that most IPv4-only or dual-stack networks already rely on NAT44 due to IPv4 address scarcity. For these networks, transitioning to an IPv6-Mostly architecture would simply require enabling NAT64 functionality on existing NAT44 devices.

In a dual-stack network, flows originating from IPv6-only capable endpoints to IPv4-only destinations are, in most cases, translated by NAT44. Within an IPv6-Mostly environment, this role is fulfilled by NAT64. Consequently, the overall load on the NAT infrastructure remains effectively unchanged, with NAT64 essentially replacing NAT44 for a subset of traffic flows.

It is also worth emphasizing that NAT64 is employed solely for facilitating access to IPv4 resources. As the availability of IPv6-enabled resources continues to increase, the volume of traffic traversing NAT64 is expected to diminish proportionally.

As discussed in Section 7.3 coexistence of IPv6-only and IPv4-only hosts on the same link might present challenges for onlink service discovery and onlink peer2peer communications. It might be consider a disadvantage of IPv6-Mostly model for networks where onlink communication between IPv4-only and IPv6-only devices is desirable.

## 5.2. IPv6-Mostly Compared to a Dedicated IPv6-Only Network

Traditional IPv6-only adoption involves separate networks alongside dual-stack ones. IPv6-Mostly approach offers significant improvements, such as:

- \* **Enhanced Scalability:** Separate IPv6-only networks double the number of SSIDs in wireless environments, causing channel congestion and degrading performance. IPv6-Mostly doesn't require additional SSIDs. Similarly, it allows IPv4 and IPv6-only devices to coexist on the same wired VLANs, eliminating the need of additional layer2 segments, VLAN IDs, and their respective layer3 configurations.
- \* **Operational Simplicity:** Managing one network segment for all clients (regardless of IPv4 needs) simplifies operations, improves user experience (no more confusing SSID choices), and reduces support tickets related to mismatched connections. For wired connections dynamic VLAN assignment becomes easier without device-specific IPv6 capability tracking.
- \* **Optimized IPv4 Consumption:** User-selected dual-stack networks often lead to unnecessary IPv4 use, as users often connect IPv6-only capable devices to a dual-stack network. IPv6-Mostly network allocates IPv4 addresses only when devices don't advertise IPv6-only capability (DHCPv4 Option 108).
- \* **Improved Problem Visibility:** User-selected fallback to dual-stack networks can mask issues with IPv6-only operation, hindering problem reporting and resolution. IPv6-Mostly forces users to work through any issues, improving identification and enabling fixes for smoother long-term transition.
- \* **Flexible, Incremental Transition:** IPv6-Mostly allows for gradual migration on a per-segment or even on per-device basis. Devices become IPv6-only only when deemed fully compatible with that mode.

## 6. Incremental Rollout Considerations

Migrating endpoints to IPv6-only fundamentally changes network dynamics by removing the IPv4 safety net. This includes the masking effect of traditional Happy Eyeballs algorithm [RFC6555]. IPv6 connectivity issues become far more prominent, including those previously hidden within dual-stack environments. Operators should be prepared to discover and troubleshoot issues in both endpoints and network infrastructure, even if the dual-stack network appeared problem-free.

Some rollout considerations are discussed in the following sections.

### 6.1. Opt-In and Opt-Out Modes

Before user-facing deployment, the administrator SHOULD consider a dedicated IPv6-mostly proof-of-concept network for early adopters. While this temporarily sacrifices some IPv6-mostly benefits (Section 5.2 ), it provides valuable operational experience and early issue detection. In a nutshell, the recommended approach contains two phases:

- \* Opt-In Phase: Invite tech-savvy early adopters to join an IPv6-mostly network and report issues. While response rates may be low, dedicated participants provide valuable troubleshooting data.
- \* Opt-Out Phase: Incrementally enable PREF64 signalling as well as Option 108. Allow selective disabling for problematic endpoints, in extreme case even by providing a separate network or rolling back IPv6-mostly in that particular subnet. Requiring users to file a problem report to opt-out provides a mechanism for gaining visibility into user experience on IPv6-mostly network. This process facilitates the identification of related issues and expedites issue resolution.

In some scenarios (see Section 7.5.1 ) the administrator MAY keep a dual-stack network as a last resort fallback mechanism but SHOULD prevent users from connecting to it accidentally (e.g. it should be a hidden SSID with authentication enabled). For recurring temporary networks, for instance during a conference, it might be a good practice to change the network SSID of the last resort fallback network between each event.

## 6.2. Per-Device and Per-Subnet Incremental Rollout

Limited control over endpoint configuration necessitates a per-subnet rollout, incrementally enabling first PREF64 signalling and then option 108 processing in DHCP.

In most networks all hosts on a given subnet receive the same RAs (unless mechanisms like per-client RA [RFC8273] are deployed), so signalling PREF64 via an RA option doesn't allow any per-endpoint opt out mechanism.

Some operating systems provide a mechanism to enable option 108 processing. In these cases, if the endpoints are managed, per-device rollout may be possible. Note that some OSes enable Option 108 support by default without providing configuration knobs to turn it off. Such systems become IPv6-only the moment Option 108 is activated on the server side.

The following approach is RECOMMENDED:

- \* PREF64 signalling: Enable PREF64 option for the router advertisement and/or DNS64 (see above). Especially if DNS64 is used, this might affect behaviour of some devices as the path via NAT64 would get generally preferred over native IPv4 path. Rollback at this stage affects the entire subnet.
- \* DHCP Server-Side Activation: Enable Option 108 processing. Some OSes automatically switch to IPv6-only. Rollback at this stage affects the entire subnet.
- \* Controlled Endpoint Activation: Enable Option 108 on managed endpoints with per-device rollback possible.

## 6.3. Rollback Approach

For quick rollback, the administrator SHOULD start with a minimal Option 108 value (300 seconds, Section 3.4 of [RFC8925]) and possibly increase this value as the IPv6-mostly network proves reliable, reducing the likelihood of full-scale rollback. In most deployments, the load of DHCP infrastructure caused by each endpoint restarting the DHCP handshake every 300 seconds would be negligible. However for large networks containing a high number of IPv6-only capable clients, the default value of 300 seconds may be too low. Even if each IPv6-only capable client sends only one DHCPDISCOVER every 300 seconds, the collective traffic can still create a substantial load on the DHCP infrastructure. Administrators are advised to monitor this load and adjust the Option 108 value as necessary.

## 7. Operational Considerations

### 7.1. Address Assignment Policy

As outlined in Section 6.3 of [RFC6877] , CLAT requires either a dedicated IPv6 prefix or, if unavailable, a dedicated IPv6 address. Currently (2024), all implementations use SLAAC for CLAT address acquisition. Therefore, to enable CLAT functionality within IPv6-mostly network segments, first-hop routers MUST be configured to advertise a Prefix Information Option (PIO, [RFC4861]) containing a globally routable SLAAC-suitable prefix with the 'Autonomous Address-Configuration' (A) flag set to one.

### 7.2. Extension Headers

Being an IPv6-specific concept, IPv6 extension headers are often neglected or even explicitly prohibited by security policies in dual-stack networks. The issues caused by blocking extension headers might be masked by the presence of Happy Eyeballs but become highly visible when there is no IPv4 to fallback to.

The network SHOULD permit at least the following extension headers:

- \* Fragment Header (Section 4.5 of [RFC8200] ).Section 7.5.5 discusses the fragmentation in more details.
- \* ESP Header, which is used for IPSec traffic, such as VPN and Wi-Fi Calling.

### 7.3. Onlink Communication and Service Discovery

Shared link is often used for automatic discovery of neighboring devices and their services by means of different protocols like Multicast DNS [RFC6762]. Many devices and/or services are built on an assumption that devices on the same link also share the same set of address families or at least they have one common address family shared between all of them.

In IPv6-Mostly, this assumption is not valid as there might be both IPv4-only and IPv6-only devices on the same link. As per Section 20 of [RFC6762], this means that the link has two unrelated ".local." zones, one for each address family. Discovery of devices across different address families is impossible, unless some sort of relay is deployed on the link.

This problem, however, is limited to networks where client-to-client onlink communications are desired and permitted by security policies. For networks where client isolation is enabled (it's often a case for

enterprise networks and public WiFi), this issue is not a concern. If onlink peer2peer communication is required, the operator needs to ensure all participating devices are either dual-stack or IPv6-mostly.

#### 7.4. NAT64 IPv4 Pools

It is not uncommon for client IP addresses to be used as a part of an authentication and access control mechanisms on the server side. Additionally, web servers are known to use client IP addresses for session tracking. To ensure that migrating a client from dual-stack to IPv6-only mode does not impact user experience:

- \* If NAT44 is employed, then NAT64 and NAT44 must share the same pool of IPv4 addresses.
- \* Whenever possible, the same NAT pool address SHOULD be assigned to all concurrent sessions originating from the same internal host (a configuration often referred to as 'sticky' NAT).

#### 7.5. Typical Issues

IPv6-mostly networks expose hidden issues by removing the IPv4 safety net. While implementation bugs vary greatly and are beyond the scope of this document, this section focuses on common problems caused by configuration, topology, or design choices. It's crucial to note that these issues likely pre-exist in dual-stack networks, but remain unnoticed due to IPv4 fallback.

##### 7.5.1. Hosts with Disabled or Disfunctional IPv6

Historically, tech support often advised disabling IPv6 as a quick workaround, leading to devices with disabled IPv6. Similarly, corporate IT may have disabled or filtered IPv6 under the assumption that it's not widely used. Such endpoints requesting Option 108 will fail to connect in an IPv6-mostly network, as they won't receive IPv4 addresses and IPv6 is disabled.

Administrators controlling endpoints SHOULD ensure those endpoints have IPv6 enabled and operational before transitioning the network to IPv6-mostly mode. This includes verifying protocol enablement as well as validation of any central or discretely managed host based firewalls that could potentially interfere with proper IPv6 function that may be masked by the presence of IPv4, and therefore exposed with its removal.

### 7.5.2. Network Extension

IPv4's NAT44 allows endpoints to extend connectivity to downstream systems without upstream network awareness or permission. This creates challenges in IPv6-mostly deployments where endpoints lack IPv4 addresses:

Solutions and trade-offs:

- \* Using DHCPv6-PD to allocate prefixes to endpoints ([RFC9663]). Provides downstream systems with IPv6 addresses and native connectivity.
- \* Enabling the CLAT function on the endpoint. This scenario is similar to the Wireline Network Architecture described in Section 4.1 of [RFC6877] . The downstream systems would receive IPv4 addresses and their IPv4 traffic would be translated to IPv6 by the endpoint. However this approach leads to the downstream systems using IPv4 only and not benefiting from end2end IPv6 connectivity. To enable IPv6 benefits, combine this with IPv6 Prefix Delegation as discussed above.
- \* Bridging and ND Proxy: The endpoint bridges IPv6 traffic and masks downstream devices behind its MAC address. This can lead to scalability issues ( Section 7.5.3 ) due to the single MAC being mapped to many IPv6 addresses.

### 7.5.3. Multiple Addresses per Device

Unlike IPv4, where endpoints typically have a single IPv4 address per interface, IPv6 endpoints inherently use multiple addresses:

- \* Link-local address.
- \* Temporary address (common on mobile devices for privacy)
- \* Stable address (for long-term identification)
- \* CLAT address (in IPv6-mostly/IPv6-only networks)

Endpoints with containers, namespaces, or ND proxy functions may have even more addresses. This poses challenges for network infrastructure devices (SAVI switches, wireless access points, etc.) that map MAC addresses to IPv6 addresses, often with limits to prevent resource exhaustion or DoS attacks. When the number of IPs per MAC limit is exceeded, infrastructure devices behavior varies across implementations, leading to inconsistent connectivity loss and other unexpected behavior: while some systems drop new addresses,

others delete older entries, causing previously functional addresses to lose connectivity. In all those cases endpoints and applications don't receive explicit signalling about the address becoming unusable.

This problem, while impacting dual-stack endpoints as well, is much less visible for dual-stack devices:

- \* On a dual-stack endpoint Happy Eyeballs might obscure the issue by falling back to IPv4.
- \* Using dedicated IPv6 addresses for CLAT instances increases the number of IPv6 addresses required on IPv6-only nodes compared to dual-stack ones.
- \* Some traffic flows (e.g. belonging to IPv4-only applications) would be using IPv4 on a dual-stack endpoint but go through CLAT on an IPv6-only one. Such flows would be unaffected on a dual-stack network but experience an outage if the CLAT address is blocked by the network.

Allocating prefixes to endpoints via DHCP-PD ([RFC9663]) allows to eliminate the issue and to address the corresponding scalability concerns. Operators of large-scale IPv6-mostly networks might benefit from using DHCPv6 prefix delegation for IPv6-only and IPv6-enabled endpoints. In that case the administrator also need to ensure that the first-hop routers set the P-flag [RFC9762] in RAs. However, the deployment model described in [RFC9663] might not yet be supported by all endpoints. The network administrator SHOULD ensure that the deployed network infrastructure devices allow sufficient number of IPv6 addresses to be mapped to a client's MAC and SHOULD monitor for events, indicating that the limit has been reached (such as syslog messages, etc).

#### 7.5.4. Host Mobility and Renumbering

Networks employing dynamic VLAN assignment (e.g., based on 802.1x or MAC-based authentication) can cause endpoints to move between VLANs and IPv6 subnets. As client operating systems do not always handle changes in link-layer state (e.g., VLAN changes) correctly, this mobility often leads to inconsistent IP stack behavior on operating systems, resulting in the persistence of old subnet addresses and potential connectivity issues due to incorrect source address selection. In such networks, the administrator SHOULD configure the link-local addresses of the first-hop routers (such as link-local addresses assigned to router interfaces or the corresponding VRRPv3 (Virtual Router Redundancy Protocol, [RFC9568]) virtual link-local IP

address) to be unique within the mobility domain (the set of links the host can move between). For example, if there are two VLANs using subnets 2001:db8:1:a::/64 and 2001:db8:1:b::/64, the administrator might configure the virtual router link-local addresses as 'fe80::2001:db8:1:a' and 'fe80::2001:db8:1:b', respectively. [I-D.link-6man-gulla] provides further analysis and discusses this approach in more detail.

#### 7.5.5. Fragmentation

As the basic IPv6 header is 20 bytes longer than the IPv4 header, translating from IPv4 to IPv6 can result in packets exceeding the path MTU on the IPv6 side. In that case NAT64 creates IPv6 packets with the Fragment Header (see Section 4 of [RFC6145] for more details. As per [RFC6145], by default the translator fragments IPv4 packets so that they fit in 1280-byte IPv6 packets. It means that all IPv4 packets larger than 1260 bytes are fragmented (or dropped if the DF bit is set).

Administrators SHOULD maximize the path MTU on the IPv6 side (from the translator to IPv6-only hosts) to minimize fragmentation. NAT64 devices SHOULD be configured to use the actual path MTU on the IPv6 side when fragmenting IPv4 packets.

Another common case of IPv6 fragmentation is the use of protocols like DNS and RADIUS, where the server response needs to be sent as a single UDP datagram. Network security policies MUST allow IPv6 fragments for permitted UDP traffic (e.g., DNS, RADIUS) where single-datagram responses are required. Allowing IPv6 fragments for permitted TCP traffic is RECOMMENDED unless the network infrastructure reliably performs TCP MSS clamping.

To ensure reliable Path MTU Discovery (PMTUD, [RFC8201]), network security policies SHOULD permit the passage of ICMPv6 Type 2 (Packet Too Big, [RFC4443]) messages to and from all IPv6-enabled devices. This recommendation is fundamentally no different from the established practice in IPv4 networks, where ICMP Type 3 Code 4 (Fragmentation Needed and DF set) messages are essential for PMTUD.

##### 7.5.5.1. Atomic Fragments

Section 4 of [RFC6145] specified that an IPv4 packet without the Don't Fragment (DF) bit set, when translated to IPv6, should include an IPv6 Fragment Header. Such packets are known as atomic fragments. However, [RFC7915] obsoleted [RFC6145] and deprecated the algorithm that generates these IPv6 atomic fragments. It has been observed that certain [RFC7915] non-compliant PLAT implementations continue to

generate IPv6 atomic fragments. This behavior causes unintended connectivity issues, particularly on endpoints where firewalls are enabled. Administrators should verify that the PLAT devices do not generate atomic fragments by default. If atomic fragments are generated, the administrator should configure the PLAT devices to disable atomic fragments generation, provided a configuration mechanism is available.

#### 7.5.6. IPv4 UDP Packets with Zero Checksum

Translating IPv4 UDP packet with zero checksum to IPv6 requires the translator to calculate the checksum and therefore has performance implication. As per Section 4.5 of [RFC7915], translators are allowed to drop IPv4 UDP packet with zero checksum. Additionally, Section 1.2 of [RFC7915] says: "Fragmented IPv4 UDP packets that do not contain a UDP checksum (i.e., the UDP checksum field is zero) are not of significant use on the Internet, and in general will not be translated by the IP/ICMP translator...However, when the translator is configured to forward the packet without a UDP checksum, the fragmented IPv4 UDP packets will be translated."

It has been observed that some systems (such as 3GPP ePDG, Evolved Packet Data Gateway) might generate IPv4 UDP packets with zero checksum. Those packets could be dropped by the PLAT devices, impacting WiFi calling on IPv6-only endpoints connecting to those ePDGs. As Section 4.5 of [RFC7915] recommends that the translators have a configuration knob to calculate an IPv6 checksum and forward the packet instead of dropping it, the administrator should consider enabling that option if such issue is encountered.

#### 7.5.7. Representing IPv6 Addresses by CLAT

Certain CLAT implementations face challenges when translating incoming IPv6 packets with native (non-synthesized) source addresses (e.g. ICMPv6 packets sent by intermediate hops on the path). This lack of standardized translation mechanisms can lead to:

- \* Incomplete Traceroute: Omission of IPv6-only hops between the endpoint and NAT64 translator, hindering troubleshooting.
- \* Path MTU Discovery Issues: Potential disruptions in the PMTU discovery process.

[I-D.ietf-v6ops-icmpext-xlat-v6only-source] proposes a solution for signalling the actual non-synthesized IPv6 source address while translating ICMPv6 error messages.

#### 7.5.8. IPv4-Dependencies in Network Admission Control

Certain layer 2 network devices (e.g., wireless access points, controllers) may enforce a policy where a host's network access is contingent upon successful DHCP IPv4 address assignment or tied to results of DHCP snooping. Consequently, hosts supporting Option 108 may experience network access denial or disconnection, as they are not assigned IPv4 addresses.

#### 7.5.9. Custom DNS Configuration on Endpoints

In IPv6-mostly networks without PREF64 in RAs, hosts rely on DNS64 ([RFC7050] to discover the NAT64 prefix for CLAT operation. [RFC8880] requires that queries for AAAA resource records of "ipv4only.arpa." MUST be sent to the recursive resolvers provided by the network, not to the resolvers configured manually, local recursive resolvers etc. However some implementations do not comply with [RFC8880] and, when configured with custom DNS resolvers (e.g., public or corporate DNS) may bypass the network-provided DNS64, preventing NAT64 prefix discovery and hindering CLAT functionality.

Where feasible, administrators SHOULD include PREF64 in RAs within IPv6-mostly networks to minimize reliance on DNS64. Administrators need to be aware of the potential for CLAT failures when endpoints use custom resolvers in environments lacking PREF64.

Similarly, some VPN clients are known to override the DNS configuration on the endpoints, preventing those devices from discovering the NAT64 prefix via [RFC7050] mechanism. If the network doesn't provide the NAT64 prefix in RAs or the endpoint doesn't support [RFC8781], the endpoint might not be able to discover the NAT64 prefix and therefore would fail to enable CLAT.

### 8. Security Considerations

The proposed deployment scenario inherits security considerations of IPv6 (see [RFC9099]). As IPv6-only device can not fall back to IPv4 if IPv6 connectivity is not available or impacted by a malicious actor. Therefore any attack affecting IPv6 connectivity would have much more drastic outcome, comparing to dual-stack networks.

By signalling a rogue NAT64 prefix to a host, a malicious actor can:

- \* cause a DoS attack, if the network does not provide NAT64 functions (for that prefix or at all);
- \* implement MitM attack by intercepting traffic to the rogue prefix.

To countermeasure this attack vector, the network administrators SHOULD configure the first-hop routers to include PREF64 information in Router Advertisements as per [RFC8781] and ensure that layer 2 security measures such as RA-Guard ([RFC6105]) are in place. Section 7 of [RFC8781] discusses this topic in more details. Additionally, [I-D.ietf-v6ops-clatn] discusses security implications of enabling CLAT if native IPv4 connectivity is available and recommends disabling CLAT in that case.

Security considerations of using DHCP Option 108 are documented in Section 6 of [RFC8925].

## 9. Privacy Considerations

This document does not introduce any new privacy considerations. IPv6-mostly networks have the same privacy considerations as other Dual-stacked or IPv6-only networks.

## 10. IANA Considerations

This memo does not introduce any requests to IANA.

## 11. References

### 11.1. 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>>.
- [RFC6105] Levy-Abegnoli, E., Van de Velde, G., Popoviciu, C., and J. Mohacsi, "IPv6 Router Advertisement Guard", RFC 6105, DOI 10.17487/RFC6105, February 2011, <<https://www.rfc-editor.org/info/rfc6105>>.
- [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>>.
- [RFC6333] Durand, A., Droms, R., Woodyatt, J., and Y. Lee, "Dual-Stack Lite Broadband Deployments Following IPv4 Exhaustion", RFC 6333, DOI 10.17487/RFC6333, August 2011, <<https://www.rfc-editor.org/info/rfc6333>>.

- [RFC6555] Wing, D. and A. Yourtchenko, "Happy Eyeballs: Success with Dual-Stack Hosts", RFC 6555, DOI 10.17487/RFC6555, April 2012, <<https://www.rfc-editor.org/info/rfc6555>>.
- [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>>.
- [RFC7050] Savolainen, T., Korhonen, J., and D. Wing, "Discovery of the IPv6 Prefix Used for IPv6 Address Synthesis", RFC 7050, DOI 10.17487/RFC7050, November 2013, <<https://www.rfc-editor.org/info/rfc7050>>.
- [RFC7335] Byrne, C., "IPv4 Service Continuity Prefix", RFC 7335, DOI 10.17487/RFC7335, August 2014, <<https://www.rfc-editor.org/info/rfc7335>>.
- [RFC8200] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", STD 86, RFC 8200, DOI 10.17487/RFC8200, July 2017, <<https://www.rfc-editor.org/info/rfc8200>>.
- [RFC8106] Jeong, J., Park, S., Beloeil, L., and S. Madanapalli, "IPv6 Router Advertisement Options for DNS Configuration", RFC 8106, DOI 10.17487/RFC8106, March 2017, <<https://www.rfc-editor.org/info/rfc8106>>.
- [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>>.
- [RFC8781] Colitti, L. and J. Linkova, "Discovering PREF64 in Router Advertisements", RFC 8781, DOI 10.17487/RFC8781, April 2020, <<https://www.rfc-editor.org/info/rfc8781>>.
- [RFC8585] Palet Martinez, J., Liu, H. M.-H., and M. Kawashima, "Requirements for IPv6 Customer Edge Routers to Support IPv4-as-a-Service", RFC 8585, DOI 10.17487/RFC8585, May 2019, <<https://www.rfc-editor.org/info/rfc8585>>.
- [RFC8925] Colitti, L., Linkova, J., Richardson, M., and T. Mrugalski, "IPv6-Only Preferred Option for DHCPv4", RFC 8925, DOI 10.17487/RFC8925, October 2020, <<https://www.rfc-editor.org/info/rfc8925>>.

- [RFC9099] Vyncke, ., Chittimaneni, K., Kaeo, M., and E. Rey, "Operational Security Considerations for IPv6 Networks", RFC 9099, DOI 10.17487/RFC9099, August 2021, <<https://www.rfc-editor.org/info/rfc9099>>.

## 11.2. Informative References

- [RFC1918] Rekhter, Y., Moskowitz, B., Karrenberg, D., de Groot, G. J., and E. Lear, "Address Allocation for Private Internets", BCP 5, RFC 1918, DOI 10.17487/RFC1918, February 1996, <<https://www.rfc-editor.org/info/rfc1918>>.
- [RFC4443] Conta, A., Deering, S., and M. Gupta, Ed., "Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification", STD 89, RFC 4443, DOI 10.17487/RFC4443, March 2006, <<https://www.rfc-editor.org/info/rfc4443>>.
- [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>>.
- [RFC4862] Thomson, S., Narten, T., and T. Jinmei, "IPv6 Stateless Address Autoconfiguration", RFC 4862, DOI 10.17487/RFC4862, September 2007, <<https://www.rfc-editor.org/info/rfc4862>>.
- [RFC6052] Bao, C., Huitema, C., Bagnulo, M., Boucadair, M., and X. Li, "IPv6 Addressing of IPv4/IPv6 Translators", RFC 6052, DOI 10.17487/RFC6052, October 2010, <<https://www.rfc-editor.org/info/rfc6052>>.
- [RFC6145] Li, X., Bao, C., and F. Baker, "IP/ICMP Translation Algorithm", RFC 6145, DOI 10.17487/RFC6145, April 2011, <<https://www.rfc-editor.org/info/rfc6145>>.
- [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>>.
- [RFC6535] Huang, B., Deng, H., and T. Savolainen, "Dual-Stack Hosts Using "Bump-in-the-Host" (BIH)", RFC 6535, DOI 10.17487/RFC6535, February 2012, <<https://www.rfc-editor.org/info/rfc6535>>.

- [RFC6762] Cheshire, S. and M. Krochmal, "Multicast DNS", RFC 6762, DOI 10.17487/RFC6762, February 2013, <<https://www.rfc-editor.org/info/rfc6762>>.
- [RFC7225] Boucadair, M., "Discovering NAT64 IPv6 Prefixes Using the Port Control Protocol (PCP)", RFC 7225, DOI 10.17487/RFC7225, May 2014, <<https://www.rfc-editor.org/info/rfc7225>>.
- [RFC7755] Anderson, T., "SIIT-DC: Stateless IP/ICMP Translation for IPv6 Data Center Environments", RFC 7755, DOI 10.17487/RFC7755, February 2016, <<https://www.rfc-editor.org/info/rfc7755>>.
- [RFC7915] Bao, C., Li, X., Baker, F., Anderson, T., and F. Gont, "IP/ICMP Translation Algorithm", RFC 7915, DOI 10.17487/RFC7915, June 2016, <<https://www.rfc-editor.org/info/rfc7915>>.
- [RFC8201] McCann, J., Deering, S., Mogul, J., and R. Hinden, Ed., "Path MTU Discovery for IP version 6", STD 87, RFC 8201, DOI 10.17487/RFC8201, July 2017, <<https://www.rfc-editor.org/info/rfc8201>>.
- [RFC8273] Brzozowski, J. and G. Van de Velde, "Unique IPv6 Prefix per Host", RFC 8273, DOI 10.17487/RFC8273, December 2017, <<https://www.rfc-editor.org/info/rfc8273>>.
- [RFC8305] Schinazi, D. and T. Pauly, "Happy Eyeballs Version 2: Better Connectivity Using Concurrency", RFC 8305, DOI 10.17487/RFC8305, December 2017, <<https://www.rfc-editor.org/info/rfc8305>>.
- [RFC8880] Cheshire, S. and D. Schinazi, "Special Use Domain Name 'ipv4only.arpa'", RFC 8880, DOI 10.17487/RFC8880, August 2020, <<https://www.rfc-editor.org/info/rfc8880>>.
- [RFC9568] Lindem, A. and A. Dogra, "Virtual Router Redundancy Protocol (VRRP) Version 3 for IPv4 and IPv6", RFC 9568, DOI 10.17487/RFC9568, April 2024, <<https://www.rfc-editor.org/info/rfc9568>>.
- [RFC9663] Colitti, L., Linkova, J., Ed., and X. Ma, Ed., "Using DHCPv6 Prefix Delegation (DHCPv6-PD) to Allocate Unique IPv6 Prefixes per Client in Large Broadcast Networks", RFC 9663, DOI 10.17487/RFC9663, October 2024, <<https://www.rfc-editor.org/info/rfc9663>>.

- [RFC9762] Colitti, L., Linkova, J., Ma, X., Ed., and D. Lamparter, "Using Router Advertisements to Signal the Availability of DHCPv6 Prefix Delegation to Clients", RFC 9762, DOI 10.17487/RFC9762, June 2025, <<https://www.rfc-editor.org/info/rfc9762>>.
- [RFC9872] Buraglio, N., Jensen, T., and J. Linkova, "Recommendations for Discovering IPv6 Prefix Used for IPv6 Address Synthesis", RFC 9872, DOI 10.17487/RFC9872, September 2025, <<https://www.rfc-editor.org/info/rfc9872>>.
- [I-D.ietf-v6ops-claton]  
Colitti, L., Linkova, J., and T. Jensen, "464XLAT Customer-side Translator (CLAT): Node Behavior and Recommendations", Work in Progress, Internet-Draft, draft-ietf-v6ops-claton-10, 18 October 2025, <<https://datatracker.ietf.org/doc/html/draft-ietf-v6ops-claton-10>>.
- [I-D.link-6man-gulla]  
Linkova, J., "Using Prefix-Specific Link-Local Addresses to Improve SLAAC Robustness", Work in Progress, Internet-Draft, draft-link-6man-gulla-01, 3 March 2025, <<https://datatracker.ietf.org/doc/html/draft-link-6man-gulla-01>>.
- [I-D.ietf-v6ops-icmpext-xlat-v6only-source]  
Lamparter, D. and J. Linkova, "Using Dummy IPv4 Address and Node Identification Extensions for IP/ICMP translators (XLATs)", Work in Progress, Internet-Draft, draft-ietf-v6ops-icmpext-xlat-v6only-source-00, 4 July 2025, <<https://datatracker.ietf.org/doc/html/draft-ietf-v6ops-icmpext-xlat-v6only-source-00>>.

#### Acknowledgements

Thanks to Brian Carpenter, Stuart Cheshire, Joe Clarke, Lorenzo Colitti, Jeremy Duncan, David Farmer, Warren Kumari, Jordi Palet, Tom Petch, Michael Richardson, Matsuzaki Yoshinobu for the discussions, the feedback, and all contribution.

#### Authors' Addresses

Nick Buraglio  
Energy Sciences Network  
IL  
United States of America  
Email: [buraglio@forwardingplane.net](mailto:buraglio@forwardingplane.net), [buraglio@es.net](mailto:buraglio@es.net)

Ondrej Caletka  
RIPE NCC  
Stationsplein 11  
Amsterdam  
Netherlands  
Email: Ondrej.Caletka@ripe.net

Jen Linkova  
Google  
1 Darling Island Rd  
Pyrmont NSW 2009  
Australia  
Email: furry13@gmail.com, furry@google.com