

IPv6 operations  
Internet-Draft  
Updates: 6877, 8585 (if approved)  
Intended status: Standards Track  
Expires: 26 January 2026

J. Linkova  
Google  
T. Jensen  
25 July 2025

464XLAT Customer-side Translator (CLAT): Node Recommendations  
draft-ietf-v6ops-claton-06

## Abstract

464XLAT [RFC6877] defines an architecture for providing IPv4 connectivity across an IPv6-only network. The solution contains two key elements: provider-side translator (PLAT) and customer-side translator (CLAT). This document complements [RFC6877] and updates Requirements for IPv6 Customer Edge Routers to Support IPv4-as-a-Service (RFC8585) by providing recommendations for the node developers on enabling and disabling CLAT functions.

## 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 26 January 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 . . . . .	2
2. Requirements Language . . . . .	3
3. Terminology . . . . .	3
4. Multiple Interfaces Considerations . . . . .	4
5. Enabling CLAT . . . . .	4
6. Disabling CLAT . . . . .	5
7. CLAT Addresses Considerations . . . . .	6
7.1. CLAT IPv4 Addresses . . . . .	6
7.2. CLAT IPv6 Addresses . . . . .	7
8. CLAT and Multiple Prefixes per Interface . . . . .	10
8.1. Link Renumbering . . . . .	12
9. MTU Considerations . . . . .	13
10. Updates to RFC8585 . . . . .	13
11. Security Considerations . . . . .	15
12. Privacy Considerations . . . . .	16
13. IANA Considerations . . . . .	16
14. References . . . . .	16
14.1. Normative References . . . . .	16
14.2. Informative References . . . . .	18
Appendix A. Enabling and Disabling CLAT: Flowchart . . . . .	20
Acknowledgements . . . . .	20
Authors' Addresses . . . . .	21

## 1. Introduction

464XLAT is widely deployed in 3GPP networks (as described in Section 4.2 of [RFC6877]) where User Equipment (UE) devices (such as mobile phones and CE routers) perform the CLAT function, providing a private IPv4 address and default route for the applications and tethered devices. Enabling 464XLAT allowed mobile operators to transition UE devices to IPv6-only mode, where UE devices WAN interfaces have only IPv6 addresses, and no IPv4 addresses.

Until recently, IPv6-only hosts were rather uncommon outside of mobile networks and datacenters. Even if the network provides PLAT in the form of NAT64 ([RFC6146]), hosts (desktops, laptops, etc.) still needed the network to provide IPv4 addresses, as otherwise

applications which require IPv4 would fail. However, as more and more operating systems outside of the 3GPP world support CLAT, it becomes possible to migrate those devices to IPv6-only mode, while still providing IPv4 as a service via 464XLAT. Networks such as public Wi-Fi, enterprise networks, or even home networks can deploy 464XLAT as described in Section 4.2 of [RFC6877]:

- \* PLAT functions are performed by NAT64 network devices.
- \* CLAT is performed by the UA itself.

Another 464XLAT deployment model is a Wireline one (section 4.1 of [RFC6877]), when a CPE router is connected to an IPv6-only network and provides CLAT functions for IPv4-enabled downstream devices. [RFC8585] specifies 464XLAT support requirements for such devices.

For both scenarios to work, the node performing CLAT (such as a host or a CPE router) need to enable the CLAT when connecting to an IPv6-only network. This document complements [RFC6877] and updates [RFC8585] by providing recommendations for the node developers on enabling and disabling CLAT functions.

## 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 [RFC6877].

- \* CLAT Node: a node (a host or a router) which performs CLAT functions by running one or multiple CLAT instances (e.g. one CLAT instance per interface).
- \* IPv4-only application: An application which requires the presence of an IPv4 address and/or IPv4 default route to operate. Examples include but are not limited to applications using IPv4 literals or opening IPv4-only sockets.
- \* IPv6-only network: A network that does not assign IPv4 addresses to hosts and facilitates connectivity to IPv4-only destinations using NAT64 ([RFC6146]). In this document, the term "IPv6-only network" specifically refers to networks that provide NAT64 as it's required by the 464XLAT architecture ([RFC6877]).

- \* Native IPv4 (such as in 'native IPv4 connectivity' or 'native IPv4 default gateway'): IPv4 connectivity or default gateway provided by the network without using any form of IPv4-as-a-service or translation mechanisms (such as 464XLAT).
- \* ULA: Unique Local Addresses, [RFC4193].

Currently the only available form of PLAT for 464XLAT deployments is NAT64. Therefore this document uses those terms interchangeably.

#### 4. Multiple Interfaces Considerations

A node may have multiple IPv6-only interfaces (for example, a mobile phone can be connected to an IPv6-only Wi-Fi network and to an IPv6-only mobile network). In that case the node SHOULD run an independent, dedicated CLAT instance on each interface connected to a network equipped with PLAT. Consequently, each CLAT instance SHOULD install a separate default IPv4 route on each CLAT-enabled interface. The metrics of IPv4 routes SHOULD be consistent with the metrics of IPv6 default routes.

#### 5. Enabling CLAT

For performance and security reasons CLAT MUST NOT be enabled if the node has IPv4 native connectivity over the given interface. Therefore recommendations provided in this section are only applicable to the IPv6-only interfaces of a given node (the node has no native IPv4 default route pointing to that interface).

To enable CLAT, an IPv6-only node needs to discover the PLAT-side translation IPv6 prefix, also known as the NAT64 prefix (see Section 6.3 of [RFC6877]). The PREF64 Router Advertisement (RA) option ([RFC8781]) provides that information and can be used as a strong indication that the network supports PLAT (NAT64) functionality. Therefore an IPv6-only node SHOULD enable CLAT as soon as a Router Advertisement containing PREF64 option is received.

If RAs received by the node do not contain a PREF64 option, the node MAY use other mechanisms to detect the PLAT presence and obtain the NAT64 prefix (such as [RFC7050]). When discovering the NAT64 prefix using the mechanism defined in [RFC7050], the node MUST follow recommendations provided in [RFC8880]. Specifically, the node MUST send the query to the DNS servers for the specific network interface per Section 7.1 of [RFC8880]. In particular, 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.

Any delay in enabling CLAT on an IPv6-only node would be impactful for IPv4-only applications, as such applications cannot benefit from 464XLAT until CLAT is operational. Therefore it's desirable that the node enables CLAT as soon as network support for PLAT is detected while IPv4 connectivity is not yet detected. The node SHOULD enable CLAT after discovering the NAT64 prefix, unless by that time the node has already obtained a non-link-local IPv4 address. The node SHOULD NOT wait for an explicit (DHCP Option 108) or an implicit (DHCP timeouts) indication that native IPv4 connectivity is not available. However, to mitigate attacks described in Section 7 of [RFC7050], the node MAY delay enabling CLAT if the NAT64 prefix was discovered via DNS ([RFC7050]) only. The delay is implementation specific. If IPv4 connectivity becomes available later, the node MUST disable CLAT (unless explicitly configured to keep it running) as discussed in the following section.

If the node supports multiple IPv4 continuity solutions, the node MUST follow recommendations from Section 4 of [RFC7335] to avoid IPv4 address space conflicts.

## 6. Disabling CLAT

It is possible that after the CLAT instance has started, native IPv4 becomes available (e.g. an IPv4 address received via DHCP). Unless explicitly configured otherwise, the node MUST disable CLAT immediately upon obtaining a native IPv4 default gateway.

While disabling CLAT is impactful for all applications and traffic flows already utilizing CLAT, it is recommended not only from a performance perspective, but also from a security point of view. Because IPv4-only networks are inherently IPv6-ignorant, they might lack IPv6 layer2 security features, such as RA Guard, that would prevent spoofed RAs. An attacker can send an RA containing the PREF64 option, while the network doesn't provide any PLAT functionality. If the node keeps CLAT enabled and uses it for IPv4-only destinations, that traffic could be dropped (availability attack) or routed through an attacker-controlled route (active attack on insecure traffic or hoarding secure traffic for "Harvest Now, Decrypt Later" attacks for non-PQC secure traffic, see Section 8 of [I-D.ietf-pquip-pqc-engineers]). Rogue RAs can also disturb traffic to destinations that support both IPv4 and IPv6 by causing IPv6 through an attacker's PLAT to be used instead of the legitimate network owner's IPv4 path.

There are some corner cases when the administrator might prefer the node to use CLAT even if the native IPv4 connectivity is available (e.g. for performance reasons, if IPv4 as a Service performs better than native IPv4). However for the reasons described above such

behaviour MUST be explicitly enabled by the administrator via a configuration knob and MUST NOT be a default behaviour, especially for unmanaged nodes.

## 7. CLAT Addresses Considerations

### 7.1. CLAT IPv4 Addresses

There are two different 464XLAT deployments models:

- \* A dedicated prefix model, which Section 4.1 of [RFC6877] calls "wireline network architecture". In that case, the node performing CLAT functions also extends the network downstream and provides network connectivity services to other connected systems. Those systems can be physical (e.g. various clients connected to a CPE router), or logical (e.g. virtual systems running on a node, while the host system acts as a router and performs CLAT). In all those cases, systems behind the CLAT node usually use [RFC1918] addresses. Despite the word "wireline", this architecture is applicable for wireless or 3GPP routers: such router can provide IPv4 connectivity to connected devices, performing CLAT functions for traffic sent/received via the IPv6-only uplink interface.
- \* A single-address model, which section 4.2 of [RFC6877] calls "wireless network architecture". In that case, the CLAT instance provides an IPv4 address and the default route to the local node's network stack only. It should be noted that [RFC6877] implies that the second deployment scenario is limited to 3GPP cases, while currently it is also deployed in other types of networks, such as enterprise networks and Wi-Fi hotspots, where hosts (as mobile phones, laptops and desktops) use CLAT to provide connectivity to IPv4-only local applications. Despite the word "wireless" in the name, this architecture is applicable for wired networks as well (e.g. desktops or servers using wired connections).

In the single-address model, the CLAT instance needs a single IPv4 CLAT address and a single CLAT-only IPv6 address (which is distinct from the one or more IPv6 addresses used by the node running CLAT for its own native IPv6 connectivity, see Section 7.2). The node providing CLAT functions to local applications SHOULD use IPv4 addresses from the dedicated 192.0.0.0/29 range ([RFC7335]), reserved for IPv4 continuity solutions including but not limited to 464XLAT. If the node runs multiple CLAT instances (see Section 4), the node SHOULD use different local IPv4 addresses for each CLAT instance. This approach limits the number of CLAT instances per node to 8, which seems to be more than sufficient at the time of writing. If in the future some deployment scenarios require more than 8 CLAT instances per node, a new larger IPv4 range will be requested from IANA.

The node MUST NOT send packets on wire from the local CLAT address.

The host SHOULD use 255.255.255.255 as a netmask for the CLAT address. That allows all 8 addresses from 192.0.0.0/29 to be used, if needed.

It should be noted that 192.0.0.0/29 is shared between multiple IPv4 continuity solutions such as 464XLAT and DS-Lite (see [RFC7335]). For example, Section 10 of [RFC6333] reserves 192.0.0.1 for the Dual-Stack Lite default router. However, as per Section 4 of [RFC7335], the host MUST NOT enable two active IPv4 continuity solutions simultaneously in a way that would cause a node to have overlapping 192.0.0.0/29 address space. Therefore, as long as the host is not using DS-Lite, it MAY use 192.0.0.1 as a CLAT address.

## 7.2. CLAT IPv6 Addresses

Section 6.3 of [RFC6877] recommends that the CLAT instance acquires a dedicated /64 for translating between IPv4 and IPv6, and only uses a single interface IPv6 address if a dedicated prefix is not available via DHCPv6-PD. However, deployments where each node can obtain a dedicated /64 just for CLAT are rather uncommon, especially in environments like enterprise networks, Wi-Fi hotspots, etc. Quite often the CLAT instance uses a single IPv6 address as a source for all IPv4 traffic translated by CLAT. In particular, in a single address model (see Section 7.1) the CLAT instance only needs a single CLAT IPv6 address, so obtaining a /64 is wasteful. For instance, a home network that gets a /60 from its ISP can only connect up to 15 CLAT-enabled devices before it runs out of available prefixes. Even in a dedicated prefix model, the CLAT instance can first perform stateful NAT44 to translate all IPv4 addresses from the dedicated prefix to a single IPv4 address, and then perform stateless CLAT.

This document updates [RFC6877] by removing the requirement to acquire a dedicated /64 prefix for the purpose of sending and receiving statelessly translated packets. The following recommendations are made instead:

- \* In a single-address model, the CLAT instance SHOULD NOT obtain a dedicated /64 for the purpose of sending and receiving statelessly translated packets.
- \* In a dedicated prefix model, the CLAT instance MAY do one of the following:
  - Perform stateful NAT44 to translate all IPv4 addresses from the dedicated prefix to a single IPv4 address, then perform stateless CLAT.
  - Obtain a dedicated IPv6 address for each CLAT IPv4 address.
  - Obtain a dedicated /64 prefix via DHCPv6-PD.

In a single-address model, the CLAT instance SHOULD obtain a dedicated IPv6 address used exclusively for CLAT functions. This is required as when the node receives a packet from a NAT64 source, the node needs to differentiate between native IPv6 traffic and traffic which needs to be passed to the CLAT instance. For example, an ICMPv6 Echo Reply packet from 64:ff9b::203.0.113.1 can be a response to either an IPv6 ping to 64:ff9b::203.0.113.1, or an IPv4 ping to 203.0.113.1, translated by CLAT. Using a dedicated IPv6 source address for CLAT traffic allows the node to make that distinction without keeping state and operate in the stateless mode (see Section 1.3 of [RFC7915]).

The node SHOULD treat its CLAT IPv6 addresses as any other IPv6 address and comply with [RFC4861] and [RFC4862]. In particular:

- \* Performing Duplicate Address Detection for each dedicated CLAT address (Section 5.4 of [RFC4862]);
  - Justification: performing DAD minimizes loss of connectivity in the unlikely event of address collision. Additionally, real world deployment experience shows that network infrastructure devices mandate a DAD packet from the client before enabling network access.
- \* Processing received unicast Neighbor Solicitations (NSes) as well as multicast ones sent to the solicited-node multicast address ([RFC4861]) for the node CLAT addresses.

- Justification: If a node doesn't respond to unicast NSeS, anytime the first-hop router gets a packet for the CLAT address and its Neighbor Cache entry is 'STALE' (Section 7.3.2 of [RFC4861]), the Neighbor Unreachability Detection process (Section 7.3.3 of [RFC4861]) will delete that CLAT address's cache entry. This forces the address resolution process to restart from scratch. Until resolution finishes, traffic for the CLAT address might drop, leading to a degraded user experience, especially for applications sensitive to jitter and packet loss.
- \* Sending Gratuitous Neighbor Advertisements ([RFC9131]) for the CLAT addresses.
  - Justification: not following this recommendation leads to user-visible packet loss when the CLAT instance starts receiving traffic after period of inactivity, or when connected to the network for the first time. The problem is discussed in Section 2 of [RFC9131].
- \* Registering the CLAT addresses assigned via SLAAC or statically using DHCPv6 ([RFC9686]), if the network supports the registration.
  - Justification: not registering CLAT addresses reduces traffic visibility for network operators, which complicates troubleshooting and forensics, as discussed in [RFC9686].

If the dedicated CLAT address is obtained via Stateless Address Autoconfiguration (SLAAC, [RFC4862]), the CLAT instance SHOULD ensure that the address is checksum-neutral for the given local IPv4 CLAT address and the NAT64 prefix (this means that the local IPv4 address needs to be assigned/known before the IPv6 address is configured). Using a checksum-neutral CLAT address provides the following benefits:

- \* Better performance as CLAT doesn't need to recalculate the checksum.
- \* If a protocol uses the standard IP checksum, CLAT doesn't need to recalculate the checksum. That improves the chances of the protocol working via CLAT even if CLAT is not aware of the protocol's semantics.

To protect user privacy and prevent user tracking through CLAT addresses, the node SHOULD generate a different interface id for the CLAT address when connecting to different networks, even if the NAT64 prefix and the local IPv4 CLAT address do not change. In particular, the node SHOULD generate a random CLAT address every time the network attachment changes to another network.

## 8. CLAT and Multiple Prefixes per Interface

IPv6 multihoming, particularly when multiple routers on the same link advertise different prefixes PIOs, presents a complex and not yet fully resolved challenge.

When routers on a given link are managed independently (e.g., by different ISPs), the resulting set of configuration parameters received by a host can be difficult to utilize without creating a complex and fragile state machine. For example, if router\_A advertises a PIO with prefix\_A and PREF64\_A, while router\_B advertises a PIO with prefix\_B and a PREF64\_B, it is crucial that the CLAT bundles the information received from each router. A CLAT instance must use PREF64\_A and generate a CLAT address from prefix\_A, sending translated packets to router\_A. Alternatively, it must use PREF64\_B, generate an address from prefix\_B, and send translated packets to router\_B. Mixing configuration information from different routers (e.g., generating a CLAT address from prefix\_A but using PREF64\_B for translation) can lead to packet loss. For example, if packets with source addresses from prefix\_A are sent to router\_B, that router (or the uplink network) might drop the packets according to BCP 38 ([RFC2827]). Similarly, if the CLAT instance uses PREF64\_A, advertised by router\_A, but those packets are sent to router\_B, that router might not be configured to translate packets for that prefix.

This document does not aim to define CLAT behavior for every possible multi-router/multi-prefix scenario. Instead, this section provides recommendations for common scenarios, leaving numerous corner cases out of scope.

This section implies that a router is identified by its link-local address, used as a source address for RAs. For example, "detecting multiple routers" means that the node received RAs from multiple link-local addresses.

A node discovering multiple routers on the same interface advertising the \_same PIOs and NAT64 prefix\_, SHOULD only create one CLAT instance using one of the PIOs to form a CLAT address.

A node discovering multiple routers on the same interface signalling the `_different PIOs and NAT64 prefixes_`, MAY create one CLAT instance for each tuple of PIOs and NAT64 prefix (both PIO and NAT64 prefix in a given tuple MUST be advertised by the same router), or only a single CLAT instance using the NAT64 prefix discovered through the selected IPv6 default router and the address formed from a PIO advertised by that router.

When a node creates a single CLAT instance and must choose between multiple PIOs, the node SHOULD select a single PIO using the same algorithm as for choosing the source address for a destination within the selected NAT64 prefix ([RFC6724], updated by [I-D.ietf-6man-rfc6724-update]).

Discussion: This approach, leveraging the default source address selection algorithm (Section 5 of [RFC6724]), typically results in the policy table (rule 6) and longest prefix match (rule 8) being used for prefix selection. This ensures CLAT address selection aligns with default source address selection for native IPv6 flows, offering the following advantages:

- \* When using the well-known NAT64 prefix (64:ff9b::/96), non-ULA prefixes are preferred over ULA prefixes by default. This is beneficial as ULA source packets may not reach PLAT devices.
- \* For network-specific NAT64 prefixes within the known-local ULA range ([I-D.ietf-6man-rfc6724-update]), the ULA prefix is preferred. This can be advantageous in home and enterprise environments where administrators intend to perform NAT64 for specific source prefixes only.
- \* For network-specific NAT64 prefixes within the operator's global non-ULA range, the longest prefix match selects the PIO, ensuring CLAT uses the operator's source address for traffic to the operator's PLAT in multi-prefix environments.
- \* In managed environments, operators can customize CLAT behavior by modifying the policy table if the default prefix selection is unsuitable.

Creating a single CLAT instance significantly simplifies the CLAT state machine. However, this approach may concentrate all traffic from that instance onto the same first-hop router and NAT64 device in some multihomed topologies. As traffic shifts from CLAT to native IPv6, this drawback becomes less significant and does not justify the added complexity of multiple instances.

### 8.1. Link Renumbering

As discussed above, a single CLAT instance per interface, using a single PIO, is typically sufficient, even if the link has multiple assigned subnets. However, PIO selection can significantly impact user experience during link renumbering.

[RFC8978] discusses various examples of "flash renumbering," where the IPv6 prefix assigned to the link changes without explicit host notification. [I-D.ietf-6man-slaac-renum] and [I-D.link-6man-gulla] discuss methods to mitigate the impact of flash renumbering. These methods generally rely on hosts with addresses from both old and new prefixes ceasing use of the old prefix and adopting the new prefix. For nodes running CLAT instances, this requires disabling instances using addresses from the old prefix and creating an instance using an address from the new prefix.

The CLAT node SHOULD use at least the following signals to detect link renumbering events:

- \* A prefix used to form the CLAT address becomes deprecated or invalid ([RFC4862]).
- \* The router (or routers) advertizing the PIO used to form the CLAT address
  - has changed its state from being reachable or probably reachable to being unknown or suspect (i.e., its neighbor cache entry moved to the INCOMPLETE state or ceased to exist, see Section 6.3.6 of [RFC4861]).
  - ceased to be a router (see Section 7.3.3 of [RFC4861]).

Upon receiving a signal indicating a possible renumbering event, the node SHOULD disable the CLAT instance(s) affected by the renumbering, and create new instance(s). In case of implicit signals (provided by the Neighbor Unreachability Detection, [RFC4861], rather than by a Router Advertisement deprecating or invalidating a prefix), the node MAY send Router Solicitations to obtain the most up-to-date network configuration information. When sending Router Solicitations the node MUST follow recommendations specified in Section 6.3.7 of [RFC4861]. The node MAY react to a potential renumbering event in a "make-before-break" manner, when old instances are still running until all required information to enable new ones becomes available.

## 9. MTU Considerations

The IPv4 header is 20 bytes long (or longer if IP options are present), while the IPv6 header is 40 bytes. This means that when CLAT translates an IPv4 packet to IPv6, it usually adds 20 bytes to the packet size. However, when CLAT translates a fragmented IPv4 packet, then Fragment Header needs to be added to the resulting IPv6 packet (Section 4.1 of [RFC7915]). The length of IPv6 Fragment Extension header is 8 bytes (Section 4.5 of [RFC8200]). Therefore, to minimize undesirable IP fragmentation ([RFC8900]), the CLAT instance in a single-address mode SHOULD present IPv4-only applications with an IPv4 MTU which is 28 bytes smaller than the IPv6 MTU of the interface the instance is running on.

## 10. Updates to RFC8585

This document makes the following changes to Section 3.2.1 of [RFC8585]:

OLD TEXT:

===

464XLAT requirements:

===

NEW TEXT:

===

464XLAT requirements:

464XLAT-0: The IPv6 Transition CE Router SHOULD follow recommendations provided in draft-link-v6ops-claton.

===

OLD TEXT:

===

464XLAT-4: The IPv6 Transition CE Router MUST implement [RFC7050] ("Discovery of the IPv6 Prefix Used for IPv6 Address Synthesis") in order to discover the provider-side translator (PLAT) translation IPv4 and IPv6 prefix(es)/suffix(es).

===

NEW TEXT:

===

464XLAT-4: The IPv6 Transition CE Router MUST implement [RFC8781] ("Discovering PREF64 in Router Advertisements") and SHOULD implement [RFC7050] ("Discovery of the IPv6 Prefix Used for IPv6 Address Synthesis") in order to discover the provider-side translator (PLAT) translation IPv4 and IPv6 prefix(es)/suffix(es).

===

OLD TEXT:

===

464XLAT-6: If the network provides several choices for the discovery/learning of the NAT64 prefix, the priority to use one or the other MUST follow this order: 1) [RFC7225] and 2) [RFC7050].

The NAT64 prefix could be discovered by means of the method defined in [RFC7050] only if the service provider uses DNS64 [RFC6147]. It may be the case that the service provider does not use or does not trust DNS64 [RFC6147] because the DNS configuration at the CE (or hosts behind the CE) can be modified by the customer. In that case, the service provider may opt to configure the NAT64 prefix by means of the option defined in [RFC7225]. This can also be used if the service provider uses DNS64 [RFC6147].

===

NEW TEXT

===

464XLAT-6: If the network provides several choices for the discovery/learning of the NAT64 prefix, the priority to use one or the other MUST follow this order: 1)[RFC7225] 2)[RFC8781] and 3) [RFC7050].

464XLAT-7: If the IPv6 Transition CE Router performs CLAT functions it SHOULD also include the PLAT prefix in Router Advertisements ([RFC8781]) sent via the LAN interfaces. If the IPv6 Transition CE Router acts as a DHCP server it SHOULD enable DHCP Option 108 ([RFC8925]) processing. The router SHOULD have a configuration knob to disable DHCP Option 108 processing.

[RFC8781] allows the service provider to signal NAT64 prefix independently from DNS64 presence. At the same time the NAT64 prefix could be discovered by means of the method defined in [RFC7050] only if the service provider uses DNS64 [RFC6147]. It may be the case that the service provider does not use or does not trust DNS64 [RFC6147] because the DNS configuration at the CE (or hosts behind the CE) can be modified by the customer. In that case, the service provider may opt to configure the NAT64 prefix by means of the option defined in [RFC7225]. This can also be used if the service provider uses DNS64 [RFC6147].

===

## 11. Security Considerations

If a malicious actor spoofs PLAT presence signals (such as an RA with PREF64 option) or DNS responses for DNS-based NAT64 prefix detection ([RFC7050]), traffic of IPv4-only applications using CLAT can be affected:

- \* if there is no PLAT (NAT64) devices, traffic to NAT64 destinations would be dropped.
- \* If the attacker intercepts traffic for the NAT64 prefix (e.g. by providing the victim with a bogus NAT64 prefix and steering traffic for those destinations towards themselves), the attacker might be able to perform man-in-the-middle attacks.

Using the PREF64 RA option to detect PLAT presence and the NAT64 prefix is less prone to such attacks than DNS-based detection ([RFC7050]), as the attacker needs to be on-link and be able to bypass layer-2 security features such as RA Guard. Therefore Section 5 recommends the PREF64 RA option as a preferred way to detect PLAT presence.

In networks lacking explicit IPv6 deployment and consequently IPv6 security features, administrators may inadvertently expose link-local IPv6 connectivity when endpoints have IPv6 enabled. This unintended exposure can facilitate PLAT presence signal falsification, enabling malicious actors to conduct Man-in-the-Middle (MitM) or Denial-of-Service (DoS) attacks. This document mitigates this risk by requiring endpoints to disable CLAT when the network provides non-link-local IPv4 connectivity, as outlined in Section 6.

## 12. Privacy Considerations

If the instance utilizes the same CLAT address for an extensive period of time or, much worse, uses the same CLAT address when connecting to different networks, eavesdroppers and information collectors can correlate various network activity to the same node. To mitigate that risk and make address-based network-activity correlation more difficult, the node SHOULD generate a different interface id for the CLAT address when connecting to different networks (see Section 7.2).

It should be noted that the node's CLAT IPv6 address is only used (and visible to observers) when the traffic is carried from the CLAT node to the PLAT device. In the vast majority of the cases it means that address is never visible outside of the network Internet edge, so to perform address-based network-activity correlation the observer needs to be located in the same network as the CLAT node.

To summarize, this document does not introduce any new privacy considerations.

## 13. IANA Considerations

This memo does not introduce any requests to IANA.

## 14. References

### 14.1. 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>>.
- [RFC4193] Hinden, R. and B. Haberman, "Unique Local IPv6 Unicast Addresses", RFC 4193, DOI 10.17487/RFC4193, October 2005, <<https://www.rfc-editor.org/info/rfc4193>>.
- [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>>.

- [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>>.
- [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>>.
- [RFC6724] Thaler, D., Ed., Draves, R., Matsumoto, A., and T. Chown, "Default Address Selection for Internet Protocol Version 6 (IPv6)", RFC 6724, DOI 10.17487/RFC6724, September 2012, <<https://www.rfc-editor.org/info/rfc6724>>.
- [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>>.
- [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>>.
- [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>>.
- [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>>.

- [RFC8028] Baker, F. and B. Carpenter, "First-Hop Router Selection by Hosts in a Multi-Prefix Network", RFC 8028, DOI 10.17487/RFC8028, November 2016, <<https://www.rfc-editor.org/info/rfc8028>>.
- [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>>.
- [RFC8981] Gont, F., Krishnan, S., Narten, T., and R. Draves, "Temporary Address Extensions for Stateless Address Autoconfiguration in IPv6", RFC 8981, DOI 10.17487/RFC8981, February 2021, <<https://www.rfc-editor.org/info/rfc8981>>.
- [RFC9131] Linkova, J., "Gratuitous Neighbor Discovery: Creating Neighbor Cache Entries on First-Hop Routers", RFC 9131, DOI 10.17487/RFC9131, October 2021, <<https://www.rfc-editor.org/info/rfc9131>>.
- [RFC9686] Kumari, W., Krishnan, S., Asati, R., Colitti, L., Linkova, J., and S. Jiang, "Registering Self-Generated IPv6 Addresses Using DHCPv6", RFC 9686, DOI 10.17487/RFC9686, December 2024, <<https://www.rfc-editor.org/info/rfc9686>>.
- [I-D.ietf-6man-rfc6724-update] Buraglio, N., Chown, T., and J. Duncan, "Prioritizing known-local IPv6 ULAs through address selection policy", Work in Progress, Internet-Draft, draft-ietf-6man-rfc6724-update-23, 7 July 2025, <<https://datatracker.ietf.org/doc/html/draft-ietf-6man-rfc6724-update-23>>.

## 14.2. 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>>.
- [RFC2827] Ferguson, P. and D. Senie, "Network Ingress Filtering: Defeating Denial of Service Attacks which employ IP Source Address Spoofing", BCP 38, RFC 2827, DOI 10.17487/RFC2827, May 2000, <<https://www.rfc-editor.org/info/rfc2827>>.
- [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>>.
- [RFC8900] Bonica, R., Baker, F., Huston, G., Hinden, R., Troan, O., and F. Gont, "IP Fragmentation Considered Fragile", BCP 230, RFC 8900, DOI 10.17487/RFC8900, September 2020, <<https://www.rfc-editor.org/info/rfc8900>>.
- [RFC8978] Gont, F., 貼or軒, J., and R. Patterson, "Reaction of IPv6 Stateless Address Autoconfiguration (SLAAC) to Flash-Renumbering Events", RFC 8978, DOI 10.17487/RFC8978, March 2021, <<https://www.rfc-editor.org/info/rfc8978>>.
- [I-D.ietf-pquip-pqc-engineers]  
Banerjee, A., Reddy, K. T., Schoiniakakis, D., Hollebeek, T., and M. Ounsworth, "Post-Quantum Cryptography for Engineers", Work in Progress, Internet-Draft, draft-ietf-pquip-pqc-engineers-13, 1 July 2025, <<https://datatracker.ietf.org/doc/html/draft-ietf-pquip-pqc-engineers-13>>.
- [I-D.ietf-6man-slaac-renum]  
Gont, F., Zorz, J., Patterson, R., and J. Linkova, "Improving the Robustness of Stateless Address Autoconfiguration (SLAAC) to Flash Renumbering Events", Work in Progress, Internet-Draft, draft-ietf-6man-slaac-renum-09, 3 March 2025, <<https://datatracker.ietf.org/doc/html/draft-ietf-6man-slaac-renum-09>>.
- [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>>.

## Appendix A. Enabling and Disabling CLAT: Flowchart

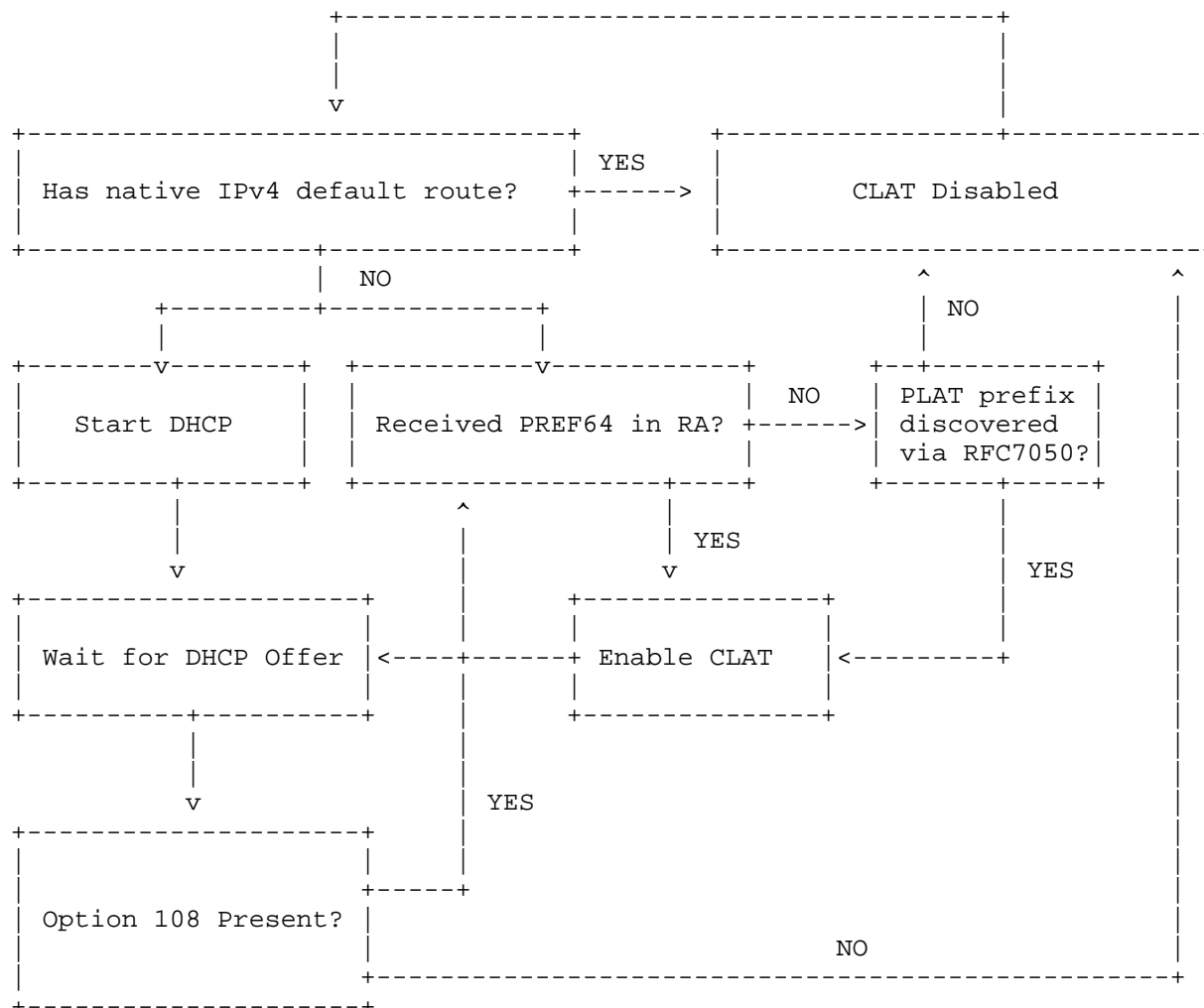


Figure 1: Enabling and Disabling CLAT Instance

## Acknowledgements

Thanks to Ondrej Caletka, Stuart Cheshire, Lorenzo Colitti, Jeremy Duncan, Jason Healy, Ed Horley, KAWASHIMA Masanobu, Ted Lemon, George Michaelson, Jordi Palet, Dieter Siegmund, Philipp S. Tiesel, Eric Vyncke for the discussions, the input, and all contribution.

Authors' Addresses

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

Tommy Jensen  
Email: tojens.ietf@gmail.com