v6ops N. Buraglio
Internet-Draft Energy Sciences Network
Intended status: Informational T. Jensen
Expires: 26 December 2025 Microsoft
J. Linkova
Google
24 June 2025

~~Prefer use of RFC8781 for~~Recommendation for the ~~-~~Discovery of IPv6 Prefix Used for IPv6 Address Synthesis

draft-ietf-v6ops-prefer8781-03

Abstract

On networks providing IPv4-IPv6 translation (NAT64, RFC7915), hosts and other endpoints might need to know the IPv6 prefix(es) used for translation (the NAT64 prefix). While “Discovery of the IPv6 Prefix Used for IPv6 Address Synthesis” (RFC 7050)~~RFC7050~~ define~~sd~~ a DNS64-based prefix discovery mechanism, more robust methods have been specified since ~~emerged~~then. This document provides ~~updated~~ guidelines for NAT64 prefix discovery, ~~deprecating the RFC7050 approach in favor of modern~~ with a preference for more deterministic alternatives compared the RFC 7050 heuristic, ~~(e.g., RFC8781)~~ whenever available.

Commenté [MB1]: The writeup should include a justification why this is used, not BCP.

Commenté [MB2]: The abstract cites 8781 only as an example. Better to align the title with the spirit of the recommendation.

Commenté [MB3]: NAT64 is defined in rfc6146.

Other than 6146, there is even no menton of «NAT64» in 6146

I think that you would like to generalize that concept. Maybe better to not use the term here, but clarify in the terminology section

Commenté [MB4]: As there may have many in theory (e.g., per-destination NAT64)

Commenté [MB5]: Abstract should be self-containd

Commenté [MB6]: This may be confusing as we are not obsoleting 7050, which is today the widely used approach.

About This Document

This note is to be removed before publishing as an RFC.

The latest revision of this draft can be found at https://github.com/buraglio/draft-nbtjjl-v6ops-prefer8781. Status information for this document may be found at https://datatracker.ietf.org/doc/draft-ietf-v6ops-prefer8781/.

Discussion of this document takes place on the v6ops Working Group mailing list (mailto:v6ops@ietf.org), which is archived at https://datatracker.ietf.org/wg/v6ops/about/. Subscribe at https://www.ietf.org/mailman/listinfo/v6ops/.

Source for this draft and an issue tracker can be found at https://github.com/buraglio/draft-nbtjjl-v6ops-prefer8781.

Status of This Memo

This Internet-Draft will expire on 26 December 2025.

Copyright Notice

Table of Contents

1. Introduction

NAT64 ~~devices~~ functions translating between IPv4 and IPv6 packet headers ~~(~~[~~RFC7915~~RFC6146]~~)~~ ~~employ~~ use a NAT64 prefix to map IPv4 addresses into the IPv6 address space, and vice versa. When a network provides NAT64 ~~services~~, it is advantageous for hosts and endpoints to acquire the network's NAT64 prefixes (PREF64). Discovering ~~the~~ PREF64s enables endpoints to:

* I~~i~~mplement the customer-side translator (CLAT) function~~s~~ of the 464XLAT architecture [RFC6877].~~;~~
* Support applications referrals, with IPv4 literals.

Commenté [MB7]: Do we need to cite both systematically?

I see that some text uses only hosts, while other only endpoints.

Other parts of the text use «node».

Please use a consistent terminology through the doc

Commenté [MB8]: Maybe used Pref64::/n to be consistent with RFC7050

Commenté [MB9]: Please check 7051

Commenté [MB10]: SDP, etc.

* T~~t~~ranslate ~~the ~~IPv4 literals to ~~an ~~IPv6 literals (Section 7.1 of [RFC8305]);

* ~~perform ~~Perform local DNS64 ~~(~~[RFC6147]~~) functions~~.

Dynamic PREF64 discovery is ~~often essential~~useful to avoid stale prefixes, particularly for unmanaged or mobile endpoints~~, where static configuration is impractical~~. ~~While ~~[RFC7050] ~~introduced ~~introduces the first DNS64-based mechanism for PREF64 discovery based in the [RFC7051] analysis. However, ~~, ~~subsequent methods have been developed to address its limitations.

For instance, [RFC8781] defines a Neighbor Discovery ~~(~~[RFC4861]~~)~~ option for Router Advertisements (RAs) to convey PREF64 information to hosts. This approach offers several advantages (Section 3 of [RFC8781]), including fate sharing with other host network configuration parameters.

Due to fundamental shortcomings of the [RFC7050] mechanism (Section 3), [RFC8781] is the preferred solution for new deployments. Implementations should strive for consistent PREF64 acquisition methods. The DNS64-based mechanism of [RFC7050] should be employed only when RA-based PREF64 delivery is unavailable, or as a fallback for legacy systems incapable of processing the PREF64 RA ~~option~~Option.

2. Conventions and Definitions

CLAT: A customer-side translator~~ (XLAT)~~, defined in [RFC6877]~~], that complies with [RFC7915]~~.

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

NAT64: a mechanism for translating IPv6 packets to IPv4 packets and vice versa. The translation is done by translating the packet headers according to the IP/ICMP Translation Algorithm defined in [RFC7915]. NAT64 translators can operate in stateless or stateful mode ([RFC6144]).

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].

Router Advertisement (RA): A packet used by Neighbor Discovery [RFC4861] and SLAAC to advertise the presence of the routers, together with other IPv6 configuration information.

SLAAC: StateLess Address AutoConfiguration, [RFC4862]

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.

Commenté [MB11]: Consistent with 8781 use

Commenté [MB12]: This is used once in the document. Do we really need to list it here?

Commenté [MB13]: Indicate this is a generalized definition.

Commenté [MB14]: Please use the notation used in RFC7050

Commenté [MB15]: This citation is confusing as this may interpreted as if PREF64 is defined there as well, which is not the case.

Commenté [MB16]: I don't think we need this.

3. Existing ~~I~~issues with RFC 7050

DNS-based method of discovering the NAT64 prefix introduces some challenges, which make this approach less preferable than ~~most recently~~latest developed alternatives (such as PREF64 RA ~~option~~Option, [RFC8781]). This section outlines the key issues, associated with [RFC7050].

Commenté [MB17]: Why not referring to the analysis at rfc7051#section-5.1.3?

3.1. Dependency on Network-Provided Recursive Resolvers

Commenté [MB18]: You may say that this problematic if NSP used, and less an issue if WKP.

Fundamentally, the presence of the NAT64 and the exact value of the prefix used for the translation are network-specific attributes. Therefore, [RFC7050] requires the device to use the DNS64 resolvers provided by the network. If the device or an application is configured to use other recursive resolvers or runs a local recursive resolver, the corresponding name resolution APIs and libraries are required to recognize 'ipv4only.arpa.' as a special name and give it special treatment. This issue and remediation approach are discussed in [RFC8880]. However, it has been observed that very few [RFC7050] implementations support [RFC8880] requirements for special treatment of 'ipv4only.arpa.'. As a result, configuring such systems and applications to use resolvers other than the one provided by the network breaks the PREF64 discovery, leading to degraded user experience.

Commenté [MB19]: Which device?

VPN clients ~~often~~ may override the host's DNS configuration, for example, by configuring enterprise DNS servers as the host's recursive resolvers and forcing all name resolution through the VPN. These enterprise DNS servers typically lack DNS64 functionality and, therefore, cannot provide information about the PREF64 used within the local network. Consequently, this prevents the host from discovering the necessary PREF64, negatively impacting its connectivity on IPv6-only networks

Commenté [MB20]: Because otherwise we will need a reference to back this.

3.2. Network Stack Initialization Delay

When using SLAAC, an IPv6 host typically requires a single RA to acquire its network configuration. For IPv6-only hosts, timely PREF64 discovery is critical, particularly for those performing local DNS64 or NAT64 functions, such as CLAT. Until ~~the~~ a PREF64 is obtained, the host's IPv4-only applications and communication to IPv4-only destinations are impaired. The mechanism defined in [RFC7050] does not bundle PREF64 information with other network configuration parameters.

Commenté [MB21]: As there might be multiple interfaces

3.3. Latency in Updates Propagation

Section 3 of [RFC7050] requires that the node ~~SHALL~~ shall cache the replies received during the PREF64 discovery and ~~SHOULD~~ should repeat the discovery process ten seconds before the TTL of the Well-Known Name's synthetic AAAA resource record expires. As a result, once ~~the~~ a PREF64 is discovered, it will be used until the TTL expired, or until the node disconnects from the network. There is no mechanism for an operator to force the PREF64 rediscovery on the node without disconnecting the node from the network. If the operator needs to change the PREF64

Commenté [MB22]: Avoid redundant normative language, or use «quote» value used in the network, they need to proactively reduce the TTL value returned by the DNS64 server. This method has two significant drawbacks:

* Many networks utilize external DNS64 servers and therefore have no control over the TTL value.

* The PREF64 changes need to be planned and executed at least TTL seconds in advance. If the operator needs to notify nodes that a particular prefix must not be used (e.g.,~~e.g.~~ during a network outage or if the nodes learnt a rogue PREF64 as a result of an attack), it might not be possible without interrupting the network connectivity for the affected nodes.

Commenté [MB23]: I don't understand this point? Isn't this the operator that offers the NAT64 as well?

3.4. Multihoming Implications

According to Section 3 of [RFC7050], a node ~~MUST~~ must examine all received AAAA resource records to discover one or more PREF64s and ~~MUST~~must utilize all learned prefixes. However, this approach presents challenges in some multihomed topologies where different DNS64 servers belonging to different ISPs might return different PREF64s. In such cases, it is crucial that traffic destined for synthesized addresses is ~~routed~~ forwarded to the correct NAT64 ~~device~~ function and the source address selected for those flows belongs to the prefix from that ISP's address space. In other words, the node needs to associate ~~the~~a discovered PREF64 with upstream information, including the IPv6 prefix and default gateway. Currently, there is no reliable way for a node to map a DNS64 response (and the prefix learned from it) to a specific upstream in a multihoming scenario. Consequently, the node might inadvertently select an incorrect source address for a given PREF64 and/or send traffic to the incorrect uplink.

3.5. Security Implications

As discussed in Section 7 of [RFC7050], the DNS-based PREF64 discovery is prone to DNS spoofing attacks. In addition to creating a wider attack surface for IPv6 deployments, [RFC7050] has other security challenges worth noting to justify declaring it legacy.

3.5.1. Definition of S~~s~~ecure C~~c~~hannel

[RFC7050] requires a node's communication channel with a DNS64 server to be a "secure channel" which it defines to mean "a communication channel a node has between itself and a DNS64 server protecting DNS protocol-related messages from interception and tampering." This need is redundant when another communication mechanism of IPv6-related configuration, specifically ~~Router Advertisements~~RAs, can already be defended against tampering by RA-Guard ~~RA Guard~~ [RFC6105]. Requiring nodes to implement two defense mechanisms when only one is necessary when [RFC8781] is used in place of [RFC7050] creates unnecessary risk.

3.5.2. Secure C~~c~~hannel ~~example~~ Example of IPsec

One of the two examples that [RFC7050] defines to qualify a communication channel with a DNS64 server is the use of an "IPsec-based virtual private network (VPN) tunnel". As of the time of this writing, this is not supported as a practice by any common operating system DNS client. While they could, there have also since been multiple mechanisms defined for performing DNS-specific encryption such as those defined in [RFC9499] that would be more appropriately scoped to the applicable DNS traffic. These are also compatible with encrypted

DNS advertisement by the network using Discovery of Network-designated Resolvers [RFC9463] that would ensure the clients know in advance that the DNS64 server supported the encryption mechanism.

3.5.3. Secure C~~c~~hannel ~~example ~~Example of L~~l~~ink ~~layer ~~Layer E~~e~~ncryption

The other example given by [RFC7050] that would allow a communication channel with a DNS64 server to qualify as a "secure channel" is the use of a "link layer utilizing data encryption technologies". As of the time of this writing, most common link layer implementations use data encryption already with no extra effort needed on the part of network nodes. While this appears to be a trivial way to satisfy this requirement, it also renders the requirement meaningless since any node along the path can still read the higher-layer DNS traffic containing the translation prefix. This seems to be at odds with the definition of "secure channel" as explained in Section 2.2 of [RFC7050].

4. Recommendations for PREF64 Discovery

4.1. Deployment Recommendations

Operators deploying NAT64 ~~networks ~~SHOULD provide PREF64 information in Router Advertisements ~~as ~~per [RFC8781].

4.1.1. Mobile N~~n~~etwork ~~considerations~~Considerations

Use of [RFC8781] may not be ~~currently ~~practical for networks that have more complex network control signaling or rely on slower network component upgrade cycles, such as mobile networks. These environments are encouraged to incorporate [RFC8781] when made practical by infrastructure upgrades or software stack feature additions.

4.2. Clients Implementation Recommendations

Clients SHOULD try obtain PREF64 information from Router Advertisements ~~as ~~per [RFC8781] instead of using [RFC7050] method. In the absence of the PREF64 information in RAs, a client MAY choose to fall back to the discovery heuristic defined in [RFC7050].

X. Operational Considerations

5. Security Considerations

Commenté [MB24]: I don't understand the 9499 citation here.

Commenté [MB25]: Support of 8787 requires changes to all access nodes (PGW, UPF), which has a cost.

Commenté [MB26]: Host?

Commenté [MB27]: As it is not sure it will retrieve a prefix

Commenté [MB28]: Please note that 8781 has already the following order preferecnce:

==

When different PREF64s are discovered using multiple mechanisms, hosts **SHOULD** select one source of information only. The **RECOMMENDED** order is:¶
•PCP-discovered prefixes [RFC7225], if supported;¶
•PREF64s discovered via the RA Option;¶
•PREF64s resolving an IPv4-only fully qualified domain name [RFC7050]¶

==

Some text to explain how is this is different from that reco

Commenté [MB29]: Please add ops impacts (access nodes), transition path.

Refer to more guidance at https://datatracker.ietf.org/doc/html/draft-opsarea-rfc5706bis-02#name-operational-considerations-

Obtaining PREF64 information ~~from~~ using RAs~~Router Advertisements~~ improves the overall security of an IPv6-only client as it mitigates all attack vectors related to spoofed or rogue DNS response, as discussed in Section 7 of [RFC7050]. Security considerations related to obtaining PREF64 information from RAs are discussed in Section 7 of [RFC8781].

6. IANA Considerations

~~It is expected that there will be a long tail of both clients and networks still relying on [RFC7050] as a sole mechanism to discover PREF64 information. Therefore IANA still need to maintain "ipv4only.arpa." as described in [RFC7050] and this document has no IANA actions.~~This document does not make any request to IANA.

7. References

7.1. 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/rfc/rfc2119>.

[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/rfc/rfc7050>.

[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/rfc/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/rfc/rfc8781>.

7.2. Informative References

[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/rfc/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/rfc/rfc4862>.

[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/rfc/rfc6105>.

[RFC6144] Baker, F., Li, X., Bao, C., and K. Yin, "Framework for IPv4/IPv6 Translation", RFC 6144, DOI 10.17487/RFC6144, April 2011, <https://www.rfc-editor.org/rfc/rfc6144>.

[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/rfc/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/rfc/rfc6147>.

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

[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/rfc/rfc7915>.

[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/rfc/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/rfc/rfc8880>.

[RFC9463] Boucadair, M., Ed., Reddy.K, T., Ed., Wing, D., Cook, N., and T. Jensen, "DHCP and Router Advertisement Options for the Discovery of Network-designated Resolvers (DNR)", RFC 9463, DOI 10.17487/RFC9463, November 2023, <https://www.rfc-editor.org/rfc/rfc9463>.

[RFC9499] Hoffman, P. and K. Fujiwara, "DNS Terminology", BCP 219, RFC 9499, DOI 10.17487/RFC9499, March 2024, <https://www.rfc-editor.org/rfc/rfc9499>.

Acknowledgments

The authors would like to than the following people for their valuable contributions: Lorenzo Colitti, Tom Costello, Charles Eckel, Nick Heatley, Gabor Lencse and Peter Schmitt.

Authors' Addresses

Nick Buraglio
Energy Sciences Network
Email: buraglio@forwardingplane.net

Tommy Jensen
Microsoft
Email: tojens.ietf@gmail.com

Jen Linkova
Google
Email: furry13@gmail.com