

Internet Area WG
Internet-Draft
Intended status: Best Current Practice
Expires: January 24, 2019

R. Bonica
Juniper Networks
F. Baker
Unaffiliated
G. Huston
APNIC
R. Hinden
Check Point Software
O. Troan
Cisco
F. Gont
SI6 Networks
July 23, 2018

IP Fragmentation Considered Fragile
draft-bonica-intarea-frag-fragile-03

Abstract

This document provides an overview of IP fragmentation. It also explains how IP fragmentation reduces the reliability of Internet communication.

Finally, this document proposes alternatives to IP fragmentation and provides recommendations for application developers and network operators.

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 January 24, 2019.

Copyright Notice

Copyright (c) 2018 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 Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	3
2. IP Fragmentation	3
2.1. Links, Paths, MTU and PMTU	3
2.2. Upper-layer Protocols	5
3. Requirements Language	7
4. IP Fragmentation Reduces Reliability	7
4.1. Middle Box Failures	7
4.2. Partial Filtering	8
4.3. Telemetry and Monitoring and monitoring Failures	8
4.4. Suboptimal Load Balancing	9
4.5. Security Vulnerabilities	9
4.6. Blackholing Due to ICMP Loss	11
4.6.1. Transient Loss	12
4.6.2. Incorrect Implementation of Security Policy	12
4.6.3. Persistent Loss Caused By Anycast	13
4.7. Blackholing Due To Filtering	13
5. Alternatives to IP Fragmentation	14
5.1. Transport Layer Solutions	14
5.2. Application Layer Solutions	15
6. Applications That Rely on IPv6 Fragmentation	16
6.1. DNS	16
6.2. OSPFv3	17
6.3. Packet-in-Packet Encapsulations	17
7. Recommendations	17
7.1. For Application Developers	17
7.2. For Network Operators	18
8. IANA Considerations	18
9. Security Considerations	18
10. Acknowledgements	18
11. References	18
11.1. Normative References	18

11.2. Informative References	20
Appendix A. Contributors' Address	22
Authors' Addresses	22

1. Introduction

Operational experience [RFC7872] [Huston] reveals that IP fragmentation reduces the reliability of Internet communication. This document provides an overview of IP fragmentation. It also explains how IP fragmentation reduces the reliability of Internet communication.

Finally, this document proposes alternatives to IP fragmentation and provides recommendations for application developers and network operators.

2. IP Fragmentation

2.1. Links, Paths, MTU and PMTU

An Internet path connects a source node to a destination node. A path can contain links and intermediate systems. If a path contains more than one link, the links are connected in series and an intermediate system connects each link to the next. An intermediate system can be a router or a middle box.

Internet paths are dynamic. Assume that the path from one node to another contains a set of links and intermediate systems. If the network topology changes, that path can also change so that it includes a different set of links and intermediate systems.

Each link is constrained by the number of bytes that it can convey in a single IP packet. This constraint is called the link Maximum Transmission Unit (MTU). IPv4 [RFC0791] requires every link to have an MTU of 68 bytes or greater. IPv6 [RFC8200] requires every link to have an MTU of 1280 bytes or greater. These are called the IPv4 and IPv6 minimum link MTU's.

Each Internet path is constrained by the number of bytes that it can convey in a IP single packet. This constraint is called the Path MTU (PMTU). For any given path, the PMTU is equal to the smallest of its link MTU's. Because Internet paths are dynamic, PMTU is also dynamic.

For reasons described below, source nodes estimate the PMTU between themselves and destination nodes. A source node can produce extremely conservative PMTU estimates in which:

- o The estimate for each IPv4 path is equal to the IPv4 minimum link MTU.
- o The estimate for each IPv6 path is equal to the IPv6 minimum link MTU.

While these conservative estimates are guaranteed to be less than or equal to the actual PMTU, they are likely to be much less than the actual PMTU. This may adversely affect upper-layer protocol performance.

By executing Path MTU Discovery (PMTUD) [RFC1191] [RFC8201] procedures, a source node can maintain a less conservative, running estimate of the PMTU between itself and a destination node. According to these procedures, the source node produces an initial PMTU estimate. This initial estimate is equal to the MTU of the first link along the path to the destination node. It can be greater than the actual PMTU.

Having produced an initial PMTU estimate, the source node sends non-fragmentable IP packets to the destination node. If one of these packets is larger than the actual PMTU, a downstream router will not be able to forward the packet through the next link along the path. Therefore, the downstream router drops the packet and sends an Internet Control Message Protocol (ICMP) [RFC0792] [RFC4443] Packet Too Big (PTB) message to the source node. The ICMP PTB message indicates the MTU of the link through which the packet could not be forwarded. The source node uses this information to refine its PMTU estimate.

PMTUD produces a running estimate of the PMTU between a source node and a destination node. Because PMTU is dynamic, at any given time, the PMTU estimate can differ from the actual PMTU. In order to detect PMTU increases, PMTUD occasionally resets the PMTU estimate to the MTU of the first link along path to the destination node. It then repeats the procedure described above.

PMTUD has the following characteristics:

- o It relies on the network's ability to deliver ICMP PTB messages to the source node.
- o It is susceptible to attack because ICMP messages are easily forged [RFC5927].

FOOTNOTE: According to RFC 0791, every IPv4 host must be capable of receiving a packet whose length is equal to 576 bytes. However, the

IPv4 minimum link MTU is not 576. Section 3.2 of RFC 0791 explicitly states that the IPv4 minimum link MTU is 68 bytes.

FOOTNOTE: In the paragraphs above, the term "non-fragmentable packet" is introduced. A non-fragmentable packet can be fragmented at its source. However, it cannot be fragmented by a downstream node. An IPv4 packet whose DF-bit is set to zero is fragmentable. An IPv4 packet whose DF-bit is set to one is non-fragmentable. All IPv6 packets are also non-fragmentable.

FOOTNOTE: In the paragraphs above, the term "ICMP PTB message" is introduced. The ICMP PTB message has two instantiations. In ICMPv4 [RFC0792], the ICMP PTB message is Destination Unreachable message with Code equal to (4) fragmentation needed and DF set. This message was augmented by [RFC1191] to indicate the MTU of the link through which the packet could not be forwarded. In ICMPv6 [RFC4443], the ICMP PTB message is a Packet Too Big Message with Code equal to (0). This message also indicates the MTU of the link through which the packet could not be forwarded.

2.2. Upper-layer Protocols

When an upper-layer protocol submits data to the underlying IP module, and the resulting IP packet's length is greater than the PMTU, IP fragmentation may be required. IP fragmentation divides a packet into fragments. Each fragment includes an IP header and a portion of the original packet.

[RFC0791] describes IPv4 fragmentation procedures. IPv4 packets whose DF-bit is set to one cannot be fragmented. IPv4 packets whose DF-bit is set to zero can be fragmented at the source node or by any downstream router. [RFC8200] describes IPv6 fragmentation procedures. IPv6 packets can be fragmented at the source node only.

IPv4 fragmentation differs slightly from IPv6 fragmentation. However, in both IP versions, the upper-layer header appears in the first fragment only. It does not appear in subsequent fragments.

Upper-layer protocols can operate in the following modes:

- o Do not rely on IP fragmentation.
- o Rely on IP source fragmentation only (i.e., fragmentation at the source node).
- o Rely on IP source fragmentation and downstream fragmentation (i.e., fragmentation at any node along the path).

Upper-layer protocols running over IPv4 can operate in all of the above-mentioned modes. Upper-layer protocols running over IPv6 can operate in the first and second modes only.

Upper-layer protocols that operate in the first two modes (above) require access to the PMTU estimate. In order to fulfil this requirement, they can

- o Estimate the PMTU to be equal to the IPv4 or IPv6 minimum link MTU.
- o Access the estimate that PMTUD produced.
- o Execute PMTUD procedures themselves.
- o Execute Packetization Layer PMTUD (PLPMTUD) [RFC4821] [I-D.fairhurst-tsvwg-datagram-plpmtud] procedures.

According to PLPMTUD procedures, the upper-layer protocol maintains a running PMTU estimate. It does so by sending probe packets of various sizes to its peer and receiving acknowledgements. This strategy differs from PMTUD in that it relies on acknowledgement of received messages, as opposed to ICMP PTB messages concerning dropped messages. Therefore, PLPMTUD does not rely on the network's ability to deliver ICMP PTB messages to the source.

An upper-layer protocol that does not rely on IP fragmentation never causes the underlying IP module to emit

- o A fragmentable IP packet (i.e., an IPv4 packet with the DF-bit set to zero).
- o An IP fragment.
- o A packet whose length is greater than the PMTU estimate.

However, when the PMTU estimate is greater than the actual PMTU, the upper-layer protocol can cause the underlying IP module to emit a packet whose length is greater than the actual PMTU. When this occurs, a downstream router drops the packet and the source node refines its PMTU estimate, employing either PMTUD or PLPMTUD procedures.

When an upper-layer protocol that relies on IP source fragmentation only submits data to the underlying IP module, and the resulting packet is larger than the PMTU estimate, the underlying IP module fragments the packet and emits the fragments. However, the upper-layer protocol never causes the underlying IP module to emit

- o A fragmentable IP packet.
- o A packet whose length is greater than the PMTU estimate.

When the PMTU estimate is greater than the actual PMTU, the upper-layer protocol can cause the underlying IP module to emit a packet whose length is greater than the actual PMTU. When this occurs, a downstream router drops the packet and the source node refines its PMTU estimate, employing either PMTUD or PLPMTUD procedures.

An upper-layer protocol that relies on IP source fragmentation and downstream fragmentation can cause the underlying IP module to emit

- o A fragmentable IP packet.
- o An IP fragment.
- o A packet whose length is greater than the PMTU estimate.

A protocol that relies on IP source fragmentation and downstream fragmentation does not require access to the PMTU estimate. For these protocols, the underlying IP module:

- o Fragments all packets whose length exceeds the MTU of the first link along the path to the destination.
- o Sets the DF-bit to zero, so that downstream nodes can fragment the packet.

3. 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.

4. IP Fragmentation Reduces Reliability

This section explains how IP fragmentation reduces the reliability of Internet communication.

4.1. Middle Box Failures

Many middle boxes require access to the transport-layer header. However, when a packet is divided into fragments, the transport-layer header appears in the first fragment only. It does not appear in

subsequent fragments. This omission can prevent middle boxes from delivering their intended services.

For example, assume that a router diverts selected packets from their normal path towards network appliances that support deep packet inspection and lawful intercept. The router selects packets for diversion based upon the following 5-tuple:

- o IP Source Address.
- o IP Destination Address.
- o IPv4 Protocol or IPv6 Next Header.
- o transport-layer source port.
- o transport-layer destination port.

IP fragmentation causes this selection algorithm to behave suboptimally, because the transport-layer header appears only in the first fragment of each packet.

In another example, a middle box remarks a packet's Differentiated Services Code Point [RFC2474] based upon the above-mentioned 5-tuple. IP fragmentation causes this process to behave suboptimally, because the transport-layer header appears only in the first fragment of each packet.

In all of the above-mentioned examples, the middle box cannot deliver its intended service without reassembling fragmented packets.

4.2. Partial Filtering

IP fragments cause problems for firewalls whose filter rules include decision making based on TCP and UDP ports. As the port information is not in the trailing fragments the firewall may elect to accept all trailing fragments, which may admit certain classes of attack, or may elect to block all trailing fragments, which may block otherwise legitimate traffic, or may elect to reassemble all fragmented packets, which may be inefficient and negatively affect performance.

4.3. Telemetry and Monitoring and monitoring Failures

Stateless telemetry and monitoring strategies may require the transport-layer header to appear in every packet. However, when a packet is divided into fragments, the transport-layer header appears in the first fragment only. It does not appear in subsequent

fragments. This omission can prevent some stateless telemetry strategies from functioning correctly.

4.4. Suboptimal Load Balancing

Many stateless load-balancers require access to the transport-layer header. Assume that a load-balancer distributes flows among parallel links. In order to optimize load balancing, the load-balancer sends every packet or packet fragment belonging to a flow through the same link.

In order to assign a packet or packet fragment to a link, the load-balancer executes an algorithm. If the packet or packet fragment contains a transport-layer header, the load balancing algorithm accepts the following 5-tuple as input:

- o IP Source Address.
- o IP Destination Address.
- o IPv4 Protocol or IPv6 Next Header.
- o transport-layer source port.
- o transport-layer destination port.

However, if the packet or packet fragment does not contain a transport-layer header, the load balancing algorithm accepts only the following 3-tuple as input:

- o IP Source Address.
- o IP Destination Address.
- o IPv4 Protocol or IPv6 Next Header.

Therefore, non-fragmented packets belonging to a flow can be assigned to one link while fragmented packets belonging to the same flow can be divided between that link and another. This can cause suboptimal load balancing.

4.5. Security Vulnerabilities

Security researchers have documented several attacks that rely on IP fragmentation. The following are examples:

- o Overlapping fragment attack [RFC1858][RFC3128] [RFC5722]

- o Resource exhaustion attacks (such as the Rose Attack)
- o Attacks based on predictable fragment identification values [RFC7739]
- o Attacks based on bugs in the implementation of the fragment reassembly algorithm
- o Evasion of Network Intrusion Detection Systems (NIDS) [Ptacek1998]

In the overlapping fragment attack, an attacker constructs a series of packet fragments. The first fragment contains an IP header, a transport-layer header, and some transport-layer payload. This fragment complies with local security policy and is allowed to pass through a stateless firewall. A second fragment, having a non-zero offset, overlaps with the first fragment. The second fragment also passes through the stateless firewall. When the packet is reassembled, the transport layer header from the first fragment is overwritten by data from the second fragment. The reassembled packet does not comply with local security policy. Had it traversed the firewall in one piece, the firewall would have rejected it.

A stateless firewall cannot protect against the overlapping fragment attack. However, destination nodes can protect against the overlapping fragment attack by implementing the reassembly procedures described in RFC 1858, RFC 3128 and RFC 8200. These reassembly procedures detect the overlap and discard the packet.

The fragment reassembly algorithm is a stateful procedure for an otherwise stateless protocol. As such, it can be exploited for resource exhaustion attacks. An attacker can construct a series of fragmented packets, with one fragment missing from each packet so that the reassembly process cannot complete. Thus, this attack causes resource exhaustion on the destination node, possibly denying reassembly services to other flows. This type of attack can be mitigated by flushing fragment reassembly buffers when necessary, at the expense of possibly dropping legitimate fragments.

An IP fragment contains an "Identification" field that, together with the IP Source Address and Destination Address of a packet, identifies fragments that correspond to the same original datagram, so that they can be reassembled together by the receiving host. Many implementations have employed predictable values for the Identification field, thus making it easy for an attacker to forge malicious IP fragments that would cause the reassembly procedure for legitimate packets to fail.

Over the years multiple IPv4 and IPv6 implementations have been found to have flaws in their implementation of the IP fragment reassembly algorithm, typically resulting in buffer overflows. These buffer overflows have been exploitable for denial of service and remote code execution attacks.

NIDS aims at identifying malicious activity by analyzing network traffic. Ambiguity in the possible result of the fragment reassembly process may allow an attacker to evade these systems. Many of these systems try to mitigate some of these evasion techniques by e.g. Computing all possible outcomes of the fragment reassembly process, at the expense of increased processing requirements.

4.6. Blackholing Due to ICMP Loss

As stated above, an upper-layer protocol requires access the PMTU estimate if it:

- o Does not rely on IP fragmentation.
- o Relies on IP source fragmentation only (i.e., fragmentation at the source node).

In order to satisfy this requirement, the upper-layer protocol can:

- o Estimate the PMTU to be equal to the IPv4 or IPv6 minimum link MTU.
- o Access the estimate that PMTUD produced.
- o Execute PMTUD procedures itself.
- o Execute PLPMTUD procedures.

PMTUD relies upon the network's ability to deliver ICMP PTB messages to the source node. Therefore, if an upper-layer protocol relies on PMTUD, it also relies on the network's ability to deliver ICMP PTB messages to the source node.

According to [RFC4890], ICMP PTB messages must not be filtered. However, ICMP PTB delivery is not reliable. It is subject to both transient and persistent loss.

Transient loss of ICMP PTB messages causes PMTUD to perform less efficiently, but does not cause it to fail completely. When the conditions contributing to transient loss abate, the network regains its ability to deliver ICMP PTB messages and PMTUD regains its

ability to function. Section 4.6.1 of this document describes conditions that lead to transient loss of ICMP PTB messages.

However, persistent loss of ICMP PTB messages causes PMTUD to fail completely. Section 4.6.2 and Section 4.6.3 of this document describe conditions that lead to persistent loss of ICMP PTB messages.

The problem described in this section is specific to PMTUD. It does not occur when the upper-layer protocol obtains its PMTU estimate from PLPMTUD or any other source.

4.6.1. Transient Loss

The following factors can contribute to transient loss of ICMP PTB messages:

- o Network congestion.
- o Packet corruption.
- o Transient routing loops.
- o ICMP rate limiting.

The effect of rate limiting may be severe, as RFC 4443 recommends strict rate limiting of IPv6 traffic.

4.6.2. Incorrect Implementation of Security Policy

Incorrect implementation of security policy can cause persistent loss of ICMP PTB messages.

Assume that a Customer Premise Equipment (CPE) router implements the following zone-based security policy:

- o Allow any traffic to flow from the inside zone to the outside zone.
- o Do not allow any traffic to flow from the outside zone to the inside zone unless it is part of an existing flow (i.e., it was elicited by an outbound packet).

When a correct implementation of the above-mentioned security policy receives an ICMP PTB message, it examines the ICMP PTB payload in order to determine the original packet (i.e., the packet that elicited the ICMP PTB message) belonged to an existing flow. If the original packet belonged to an existing flow, the implementation

allows the ICMP PTB to flow from the outside zone to the inside zone. If not, the implementation discards the ICMP PTB message.

When a incorrect implementation of the above-mentioned security policy receives an ICMP PTB message, it discards the packet because its source address is not associated with an existing flow.

The security policy described above is implemented incorrectly on many consumer CPE routers.

4.6.3. Persistent Loss Caused By Anycast

Anycast can cause persistent loss of ICMP PTB messages. Consider the example below:

A DNS client sends a request to an anycast address. The network routes that DNS request to the nearest instance of that anycast address (i.e., a DNS Server). The DNS server generates a response and sends it back to the DNS client. While the response does not exceed the DNS server's PMTU estimate, it does exceed the actual PMTU.

A downstream router drops the packet and sends an ICMP PTB message the packet's source (i.e., the anycast address). The network routes the ICMP PTB message to the anycast instance closest to the downstream router. Sadly, that anycast instance may not be the DNS server that originated the DNS response. It may be another DNS server with the same anycast address. The DNS server that originated the response may never receive the ICMP PTB message and may never update its PMTU estimate.

4.7. Blackholing Due To Filtering

In RFC 7872, researchers sampled Internet paths to determine whether they would convey packets that contain IPv6 extension headers. Sampled paths terminated at popular Internet sites (e.g., popular web, mail and DNS servers).

The study revealed that at least 28% of the sampled paths did not convey packets containing the IPv6 Fragment extension header. In most cases, fragments were dropped in the destination autonomous system. In other cases, the fragments were dropped in transit autonomous systems.

Another recent study [Huston] confirmed this finding. It reported that 37% of sampled endpoints used IPv6-capable DNS resolvers that were incapable of receiving a fragmented IPv6 response.

It is difficult to determine why network operators drop fragments. Possible causes follow:

- o Hardware inability to process fragmented packets.
- o Failure to change a vendor defaults.
- o Unintentional misconfiguration.
- o Intentional configuration (e.g., network operators consciously chooses to drop IPv6 fragments in order to address the issues raised in Section 4.1 through Section 4.6, above.)

5. Alternatives to IP Fragmentation

5.1. Transport Layer Solutions

The Transport Control Protocol (TCP) [RFC0793]) can be operated in a mode that does not require IP fragmentation.

Applications submit a stream of data to TCP. TCP divides that stream of data into segments, with no segment exceeding the TCP Maximum Segment Size (MSS). Each segment is encapsulated in a TCP header and submitted to the underlying IP module. The underlying IP module prepends an IP header and forwards the resulting packet.

If the TCP MSS is sufficiently small, the underlying IP module never produces a packet whose length is greater than the actual PMTU. Therefore, IP fragmentation is not required.

TCP offers the following mechanisms for MSS management:

- o Manual configuration
- o PMTUD
- o PLPMTUD

For IPv6 nodes, manual configuration is always applicable. If the MSS is manually configured to 1220 bytes and the packet does not contain extension headers, the IP layer will never produce a packet whose length is greater than the IPv6 minimum link MTU (1280 bytes). However, manual configuration prevents TCP from taking advantage of larger link MTU's.

RFC 8200 strongly recommends that IPv6 nodes implement PMTUD, in order to discover and take advantage of path MTUs greater than 1280 bytes. However, as mentioned in Section 2.1, PMTUD relies upon the

network's ability to deliver ICMP PTB messages. Therefore, PMTUD is applicable only in environments where the risk of ICMP PTB loss is acceptable.

By contrast, PLPMTUD does not rely upon the network's ability to deliver ICMP PTB messages. However, in many loss-based TCP congestion control algorithms, the dropping of a packet may cause the TCP control algorithm to drop the congestion control window, or even re-start with the entire slow start process. For high capacity, long round-trip time, large volume TCP streams, the deliberate probing with large packets and the consequent packet drop may impose too harsh a penalty on total TCP throughput for it to be a viable approach. [RFC4821] defines PLPMTUD procedures for TCP.

While TCP will never cause the underlying IP module to emit a packet that is larger than the PMTU estimate, it can cause the underlying IP module to emit a packet that is larger than the actual PMTU. If this occurs, the packet is dropped, the PMTU estimate is updated, the segment is divided into smaller segments and each smaller segment is submitted to the underlying IP module.

The Datagram Congestion Control Protocol (DCCP) [RFC4340] and the Stream Control Protocol (SCP) [RFC4960] also can be operated in a mode that does not require IP fragmentation. They both accept data from an application and divide that data into segments, with no segment exceeding a maximum size. Both DCCP and SCP offer manual configuration, PMTUD and PLPMTUD as mechanisms for managing that maximum size. [I-D.fairhurst-tsvwg-datagram-plpmtud] proposes PLPMTUD procedures for DCCP and SCP.

Currently, User Data Protocol (UDP) [RFC0768] lacks a fragmentation mechanism of its own and relies on IP fragmentation. However, [I-D.ietf-tsvwg-udp-options] proposes a fragmentation mechanism for UDP.

5.2. Application Layer Solutions

[RFC8085] recognizes that IP fragmentation reduces the reliability of Internet communication. It also recognizes that UDP lacks a fragmentation mechanism of its own and relies on IP fragmentation. Therefore, [RFC8085] offers the following advice regarding applications the run over the UDP.

"An application SHOULD NOT send UDP datagrams that result in IP packets that exceed the Maximum Transmission Unit (MTU) along the path to the destination. Consequently, an application SHOULD either use the path MTU information provided by the IP layer or implement Path MTU Discovery (PMTUD) itself to determine whether the path to a

destination will support its desired message size without fragmentation."

RFC 8085 continues:

"Applications that do not follow the recommendation to do PMTU/PLPMTUD discovery SHOULD still avoid sending UDP datagrams that would result in IP packets that exceed the path MTU. Because the actual path MTU is unknown, such applications SHOULD fall back to sending messages that are shorter than the default effective MTU for sending (EMTU_S in [RFC1122]). For IPv4, EMTU_S is the smaller of 576 bytes and the first-hop MTU. For IPv6, EMTU_S is 1280 bytes. The effective PMTU for a directly connected destination (with no routers on the path) is the configured interface MTU, which could be less than the maximum link payload size. Transmission of minimum-sized UDP datagrams is inefficient over paths that support a larger PMTU, which is a second reason to implement PMTU discovery."

RFC 8085 assumes that for IPv4, an EMTU_S of 576 is sufficiently small, even though the IPv4 minimum link MTU is 68 bytes.

This advice applies equally to application that run directly over IP.

6. Applications That Rely on IPv6 Fragmentation

The following applications rely on IPv6 fragmentation:

- o DNS [RFC1035]
- o OSPFv3 [RFC5340]
- o Packet-in-packet encapsulations

Each of these applications relies on IPv6 fragmentation to a varying degree. In some cases, that reliance is essential, and cannot be broken without fundamentally changing the protocol. In other cases, that reliance is incidental, and most implementations already take appropriate steps to avoid fragmentation.

This list is not comprehensive, and other protocols that rely on IPv6 fragmentation may exist. They are not specifically considered in the context of this document.

6.1. DNS

DNS relies on UDP for efficiency, and the consequence is the use of IP fragmentation for large responses, as permitted by the DNS EDNS(0) options in the query. It is possible to mitigate the issue of

fragmentation-based packet loss by having queries use smaller EDNS(0) UDP buffer sizes, but then the operational issue of the partial level of support for DNS over TCP over IPv6 becomes a limiting factor of the efficacy of this approach in an IPv6 context [Damas].

Larger DNS responses can normally be avoided by aggressively pruning the Additional section of DNS responses. One scenario where such pruning is ineffective is in the use of DNSSEC, where large key sizes act to increase the response size to certain DNS queries. There is no effective response to this situation within the DNS other than using smaller cryptographic keys and adoption of DNSSEC administrative practices that attempt to keep DNS response as short as possible.

6.2. OSPFv3

OSPFv3 implementations can emit messages large enough to cause IPv6 fragmentation. However, in keeping with the recommendations of RFC8200, and in order to optimize performance, most OSPFv3 implementations restrict their maximum message size to the IPv6 minimum link MTU.

6.3. Packet-in-Packet Encapsulations

In this document, packet-in-packet encapsulations include IP-in-IP [RFC2003], Generic Routing Encapsulation (GRE) [RFC2784], GRE-in-UDP [RFC8086] and Generic Packet Tunneling in IPv6 [RFC2473]. [RFC4459] describes fragmentation issues associated with all of the above-mentioned encapsulations.

The fragmentation strategy described for GRE in [RFC7588] has been deployed for all of the above-mentioned encapsulations. This strategy does not rely on IPv6 fragmentation except in one corner case. (see Section 3.3.2.2 of RFC 7588 and Section 7.1 of RFC 2473). Section 3.3 of [RFC7676] further describes this corner case.

7. Recommendations

7.1. For Application Developers

Application developers SHOULD NOT develop applications that rely on IPv6 fragmentation.

Application-layer protocols then depend upon IPv6 fragmentation SHOULD be updated to break that dependency.

7.2. For Network Operators

As per RFC 4890, network operators MUST NOT filter ICMPv6 PTB messages unless they are known to be forged or otherwise illegitimate. As stated in Section 4.6, filtering ICMPv6 PTB packets causes PMTUD to fail. Operators MUST ensure proper PMTUD operation in their network, including making sure the network generates PTB packets when dropping packets too large compared to outgoing interface MTU.

Many upper-layer protocols rely on PMTUD.

8. IANA Considerations

This document makes no request of IANA.

9. Security Considerations

This document mitigates some of the security considerations associated with IP fragmentation by discouraging the use of IP fragmentation. It does not introduce any new security vulnerabilities, because it does not introduce any new alternatives to IP fragmentation. Instead, it recommends well-understood alternatives.

10. Acknowledgements

Thanks to Mikael Abrahamsson, Lorenzo Colitti, Mike Heard, Tom Herbert, Tatuya Jinmei, Paolo Lucente, Eric Nygren, and Joe Touch for their comments.

11. References

11.1. Normative References

- [RFC0768] Postel, J., "User Datagram Protocol", STD 6, RFC 768, DOI 10.17487/RFC0768, August 1980, <<https://www.rfc-editor.org/info/rfc768>>.
- [RFC0791] Postel, J., "Internet Protocol", STD 5, RFC 791, DOI 10.17487/RFC0791, September 1981, <<https://www.rfc-editor.org/info/rfc791>>.
- [RFC0792] Postel, J., "Internet Control Message Protocol", STD 5, RFC 792, DOI 10.17487/RFC0792, September 1981, <<https://www.rfc-editor.org/info/rfc792>>.

- [RFC0793] Postel, J., "Transmission Control Protocol", STD 7, RFC 793, DOI 10.17487/RFC0793, September 1981, <<https://www.rfc-editor.org/info/rfc793>>.
- [RFC1035] Mockapetris, P., "Domain names - implementation and specification", STD 13, RFC 1035, DOI 10.17487/RFC1035, November 1987, <<https://www.rfc-editor.org/info/rfc1035>>.
- [RFC1191] Mogul, J. and S. Deering, "Path MTU discovery", RFC 1191, DOI 10.17487/RFC1191, November 1990, <<https://www.rfc-editor.org/info/rfc1191>>.
- [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>>.
- [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>>.
- [RFC4821] Mathis, M. and J. Heffner, "Packetization Layer Path MTU Discovery", RFC 4821, DOI 10.17487/RFC4821, March 2007, <<https://www.rfc-editor.org/info/rfc4821>>.
- [RFC8085] Eggert, L., Fairhurst, G., and G. Shepherd, "UDP Usage Guidelines", BCP 145, RFC 8085, DOI 10.17487/RFC8085, March 2017, <<https://www.rfc-editor.org/info/rfc8085>>.
- [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>>.
- [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>>.
- [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>>.

11.2. Informative References

- [Damas] Damas, J. and G. Huston, "Measuring ATR", April 2018, <<http://www.potaroo.net/ispcol/2018-04/atr.html>>.
- [Huston] Huston, G., "IPv6, Large UDP Packets and the DNS (<http://www.potaroo.net/ispcol/2017-08/xtn-hdrs.html>)", August 2017.
- [I-D.fairhurst-tsvwg-datagram-plpmtud]
Fairhurst, G., Jones, T., Tuexen, M., and I. Ruengeler, "Packetization Layer Path MTU Discovery for Datagram Transports", draft-fairhurst-tsvwg-datagram-plpmtud-02 (work in progress), December 2017.
- [I-D.ietf-tsvwg-udp-options]
Touch, J., "Transport Options for UDP", draft-ietf-tsvwg-udp-options-05 (work in progress), July 2018.
- [Ptacek1998]
Ptacek, T. and T. Newsham, "Insertion, Evasion and Denial of Service: Eluding Network Intrusion Detection", 1998, <<http://www.aciri.org/vern/Ptacek-Newsham-Evasion-98.ps>>.
- [RFC1122] Braden, R., Ed., "Requirements for Internet Hosts - Communication Layers", STD 3, RFC 1122, DOI 10.17487/RFC1122, October 1989, <<https://www.rfc-editor.org/info/rfc1122>>.
- [RFC1858] Ziemba, G., Reed, D., and P. Traina, "Security Considerations for IP Fragment Filtering", RFC 1858, DOI 10.17487/RFC1858, October 1995, <<https://www.rfc-editor.org/info/rfc1858>>.
- [RFC2003] Perkins, C., "IP Encapsulation within IP", RFC 2003, DOI 10.17487/RFC2003, October 1996, <<https://www.rfc-editor.org/info/rfc2003>>.
- [RFC2473] Conta, A. and S. Deering, "Generic Packet Tunneling in IPv6 Specification", RFC 2473, DOI 10.17487/RFC2473, December 1998, <<https://www.rfc-editor.org/info/rfc2473>>.
- [RFC2474] Nichols, K., Blake, S., Baker, F., and D. Black, "Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers", RFC 2474, DOI 10.17487/RFC2474, December 1998, <<https://www.rfc-editor.org/info/rfc2474>>.

- [RFC2784] Farinacci, D., Li, T., Hanks, S., Meyer, D., and P. Traina, "Generic Routing Encapsulation (GRE)", RFC 2784, DOI 10.17487/RFC2784, March 2000, <<https://www.rfc-editor.org/info/rfc2784>>.
- [RFC3128] Miller, I., "Protection Against a Variant of the Tiny Fragment Attack (RFC 1858)", RFC 3128, DOI 10.17487/RFC3128, June 2001, <<https://www.rfc-editor.org/info/rfc3128>>.
- [RFC4340] Kohler, E., Handley, M., and S. Floyd, "Datagram Congestion Control Protocol (DCCP)", RFC 4340, DOI 10.17487/RFC4340, March 2006, <<https://www.rfc-editor.org/info/rfc4340>>.
- [RFC4459] Savola, P., "MTU and Fragmentation Issues with In-the-Network Tunneling", RFC 4459, DOI 10.17487/RFC4459, April 2006, <<https://www.rfc-editor.org/info/rfc4459>>.
- [RFC4890] Davies, E. and J. Mohacsi, "Recommendations for Filtering ICMPv6 Messages in Firewalls", RFC 4890, DOI 10.17487/RFC4890, May 2007, <<https://www.rfc-editor.org/info/rfc4890>>.
- [RFC4960] Stewart, R., Ed., "Stream Control Transmission Protocol", RFC 4960, DOI 10.17487/RFC4960, September 2007, <<https://www.rfc-editor.org/info/rfc4960>>.
- [RFC5340] Coltun, R., Ferguson, D., Moy, J., and A. Lindem, "OSPF for IPv6", RFC 5340, DOI 10.17487/RFC5340, July 2008, <<https://www.rfc-editor.org/info/rfc5340>>.
- [RFC5722] Krishnan, S., "Handling of Overlapping IPv6 Fragments", RFC 5722, DOI 10.17487/RFC5722, December 2009, <<https://www.rfc-editor.org/info/rfc5722>>.
- [RFC5927] Gont, F., "ICMP Attacks against TCP", RFC 5927, DOI 10.17487/RFC5927, July 2010, <<https://www.rfc-editor.org/info/rfc5927>>.
- [RFC7588] Bonica, R., Pignataro, C., and J. Touch, "A Widely Deployed Solution to the Generic Routing Encapsulation (GRE) Fragmentation Problem", RFC 7588, DOI 10.17487/RFC7588, July 2015, <<https://www.rfc-editor.org/info/rfc7588>>.

- [RFC7676] Pignataro, C., Bonica, R., and S. Krishnan, "IPv6 Support for Generic Routing Encapsulation (GRE)", RFC 7676, DOI 10.17487/RFC7676, October 2015, <<https://www.rfc-editor.org/info/rfc7676>>.
- [RFC7739] Gont, F., "Security Implications of Predictable Fragment Identification Values", RFC 7739, DOI 10.17487/RFC7739, February 2016, <<https://www.rfc-editor.org/info/rfc7739>>.
- [RFC7872] Gont, F., Linkova, J., Chown, T., and W. Liu, "Observations on the Dropping of Packets with IPv6 Extension Headers in the Real World", RFC 7872, DOI 10.17487/RFC7872, June 2016, <<https://www.rfc-editor.org/info/rfc7872>>.
- [RFC8086] Yong, L., Ed., Crabbe, E., Xu, X., and T. Herbert, "GRE-in-UDP Encapsulation", RFC 8086, DOI 10.17487/RFC8086, March 2017, <<https://www.rfc-editor.org/info/rfc8086>>.

Appendix A. Contributors' Address

Authors' Addresses

Ron Bonica
Juniper Networks
2251 Corporate Park Drive
Herndon, Virginia 20171
USA

Email: rbonica@juniper.net

Fred Baker
Unaffiliated
Santa Barbara, California 93117
USA

Email: FredBaker.IETF@gmail.com

Geoff Huston
APNIC
6 Cordelia St
Brisbane, 4101 QLD
Australia

Email: gih@apnic.net

Robert M. Hinden
Check Point Software
959 Skyway Road
San Carlos, California 94070
USA

Email: bob.hinden@gmail.com

Ole Troan
Cisco
Philip Pedersens vei 1
N-1366 Lysaker
Norway

Email: ot@cisco.com

Fernando Gont
SI6 Networks
Evaristo Carriego 2644
Haedo, Provincia de Buenos Aires
Argentina

Email: fgont@si6networks.com

V6OPS Working Group
Internet-Draft
Intended status: Standards Track
Expires: December 7, 2018

G. Fioccola
Telecom Italia
G. Van de Velde
Nokia
M. Cociglio
Telecom Italia
P. Muley
Nokia
June 5, 2018

IPv6 Performance Measurement with Alternate Marking Method
draft-fioccola-v6ops-ipv6-alt-mark-01

Abstract

This document describes how the alternate marking method in [RFC8321] can be used as the passive performance measurement method in an IPv6 domain, and will discuss the strengths and the weaknesses of the implementation options available to network operations. It proposes how to extend [RFC7837] to apply alternate marking technique.

Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

Status of This Memo

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

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

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

This Internet-Draft will expire on December 7, 2018.

Copyright Notice

Copyright (c) 2018 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 Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	2
2. IPv6 application of Alternate Marking	3
2.1. IPv6 Extension Headers as Marking Field	3
2.2. Other Possibilities	5
2.2.1. IPv6 Addresses as Marking Field	5
2.2.2. IPv6 Flow Label as Marking Field	5
3. Alternate Marking Method Operation	6
3.1. Single Mark Measurement	6
3.2. Double Mark Measurement	7
4. Security Considerations	7
5. IANA Considerations	7
6. Acknowledgements	7
7. References	7
7.1. Normative References	7
7.2. Informative References	7
Authors' Addresses	9

1. Introduction

This document reports a summary on the possible implementation options for the application of the alternate marking method in an IPv6 domain.

[RFC8321] describes passive performance measurement method, which can be used to measure packet loss, latency and jitter on live traffic. Because this method is based on marking consecutive batches of packets the method often referred as Alternate Marking Method.

This document defines how the alternate marking method can be used to measure packet loss and delay metrics of IPv6 tunneled packets or SRv6 policies.

The IPv6 Header Format defined in [RFC8200] introduces the format of IPv6 addresses, the Extension Headers in the base IPv6 Header and the availability of a 20-bit flow label, that can be considered for the application of the Alternate Marking methodology.

2. IPv6 application of Alternate Marking

The application of the alternate marking requires a marking field. The alternatives that can be taken into consideration for the choice of the marking field are the following:

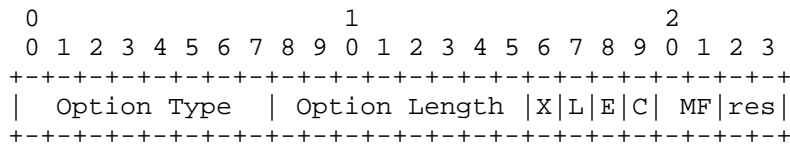
- o Extension Header
- o IPv6 Address
- o Flow Label

2.1. IPv6 Extension Headers as Marking Field

A new type of EH may be a solution space proposal (e.g. [RFC8250] and [RFC7837] give a chance).

A possibility can be to use a Hop-By-Hop(HBH) Extension Header(EH). The assumption is that a HBH EH with an alternate marking measurement option can be defined. The router processing can be optimized to handle this use case.

Using a new EH assumes that ALL routers in the domain support this type of headers, which complicates backward compatibility of the technology. The extension of an existing EH (e.g. [RFC7837]) can overcome this issue.



Mark Field (MF) is:

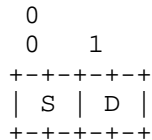


Figure 1: ConEx HBH Option Layout with Mark Field

where:

- o S - Single mark method;
- o D - Double mark method.

The Figure 1 defines a new ConEx HBH (Hop-By-Hop) Option Layout.

This proposal starts from ConEx Destination Option Layout defined in [RFC7837], where the Reserved (res) field is made by four bits that are not used in that specification, in fact they are set to zero by the sender and are ignored by the receiver.

This document aims to introduce the Mark Field (2 bits from 4 bits res field). So the Mark Field (MF) reduces the number of Reserved bits and the Reserved (res) field is now made by 2 bits.

It is important to highlight that the Destination Option Layout is used as Hop-By-Hop Option Layout, since the alternate marking methodology in [RFC8321] allows, by definition, Hop-By-Hop performance measurements.

[I-D.krishnan-conex-ipv6] also tried to introduce a ConEx HBH Options and inspired this proposal.

[I-D.fear-ippm-mpdm] introduces Marking Performance and Diagnostic Metrics (M-PDM) and aims to combine [RFC8250] with [RFC8321], while the extension of [RFC7837], proposed in this document, is optimized to include only marking method without any considerations on how to report and manage, this can be done in-band or out-of-band depending on the case.

2.2. Other Possibilities

This section reports the other possibilities that have been discussed.

2.2.1. IPv6 Addresses as Marking Field

There is an advantage of using destination addresses (DA) to encode the alternate marking method. In addition to identifying a host, a destination address is also and more fundamentally identifying an exit point from the forwarding domain. It indicates where processing for forwarding to the DA stops, and where other processing of the packet is to occur. Using the DA to encode this alternate marking processing means that it is easy to retrofit into existing devices and models. There is no need to replace existing IPv6 forwarding devices, because they already support DA based forwarding.

However using DA for marking seems a lot expensive.

2.2.2. IPv6 Flow Label as Marking Field

Considering the Flow Label, [RFC6294] makes a survey of Proposed Use Cases for the IPv6 Flow Label. The flow label is an immutable field recommended to contain a pseudo-random value, however, often it has the default value of zero. [RFC6436] and [RFC6437] open the door for IPv6 Flow Label to be used in a controlled environment and [RFC6438] describes the use of the IPv6 Flow Label field for load distribution purpose, especially across Equal Cost Multi-Path (ECMP) and/or Link Aggregation Group (LAG) paths. In addition it is possible to mention [I-D.krishnan-6man-header-reserved-bits] that tried to set aside 4 bits from the flow label field for future expansion.

There are few drawbacks to use Flow Label instead of an EH solution or IPv6 Addresses for IPv6 alternate marking, in particular an easier backward compatibility and less bits on the wire. In this way nothing breaks if a transit router does not have the capability of understanding the Flow Label context.

Since the flow-label based load balancing has been defined, the application of the Alternate Marking method to the flow label could be realised with two fundamental assumptions:

- o The original flow-label reconstructed when leaving the controlled domain.
- o The usage of IPv6 tunnels (IPv6inIPv6, IPSec, IPv6 UDP, etc..) or SRv6 policies.

In this case, the controlled domain reflects to the fact that it is a network operator choice that grabs control of packet handling within its own network. In fact, regarding the flow label, four options can be supposed:

- 1) Just do not do anything with Flow Label (leave it default).
- 2) Entropy only and NO alternate marking for performance measurements.
- 3) Alternate marking only and NO usage of entropy.
- 4) Alternate marking and entropy (in this case the entropy SHOULD be based upon a subset of bits because otherwise paths may be changed when the marking changes).

3. Alternate Marking Method Operation

[RFC8321] describes in detail the methodology, that we briefly illustrate also here.

3.1. Single Mark Measurement

As explained in the [RFC8321], marking can be applied to delineate blocks of packets based either on equal number of packets in a block or based on equal time interval. The latter method offers better control as it allows better account for capabilities of downstream nodes to report statistics related to batches of packets and, at the same time, time resolution that affects defect detection interval.

If the Single Mark measurement used, then the D flag MUST be set to zero on transmit and ignored by monitoring point.

The S flag is used to create alternate flows to measure the packet loss by switching value of the S flag. Delay metrics MAY be calculated with the alternate flow using any of the following methods:

- o First/Last Batch Packet Delay calculation: timestamps are collected based on order of arrival so this method is sensitive to packet loss and re-ordering.
- o Average Packet Delay calculation: an average delay is calculated by considering the average arrival time of the packets within a single block. This method only provides single metric for the duration of the block and it doesn't give information about the delay distribution.

3.2. Double Mark Measurement

Double Mark method allows more detailed measurement of delays for the monitored flow but it requires more nodal and network resources. If the Double Mark method used, then the S flag MUST be used to create the alternate flow. The D flag MUST be used to mark single packets to measure delay jitter.

The first marking (S flag alternation) is needed for packet loss and also for average delay measurement. The second marking (D flag is put to one) creates a new set of marked packets that are fully identified and dedicated for delay. This method is useful to have not only the average delay but also to know more about the statistic distribution of delay values.

4. Security Considerations

tbc

5. IANA Considerations

tbc

6. Acknowledgements

The authors would like to thank Fred Baker, Ole Troan, Robert Hinden, Suresh Krishnan, Brian Carpenter, Roberta Maglione, Tom Herbert, Mark Smith, Joel Halpern, Fernando Gont, Xiaohu Xu and Joel Jaeggli for their comments and feedbacks.

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/info/rfc2119>>.

7.2. Informative References

[I-D.fear-ippm-mpdm]
Elkins, N., Fioccola, G., and m. mackermann@bcbsm.com, "IPv6 Marking and Performance and Diagnostic Metrics (MPDM)", draft-fear-ippm-mpdm-00 (work in progress), June 2018.

- [I-D.krishnan-6man-header-reserved-bits]
Krishnan, S. and J. Halpern, "Reserving bits in the IPv6 header for future use", draft-krishnan-6man-header-reserved-bits-00 (work in progress), October 2010.
- [I-D.krishnan-conex-ipv6]
Krishnan, S., Kuehlewind, M., and C. Ucendo, "Options for Conex marking in IPv6 packets", draft-krishnan-conex-ipv6-02 (work in progress), March 2011.
- [RFC6294] Hu, Q. and B. Carpenter, "Survey of Proposed Use Cases for the IPv6 Flow Label", RFC 6294, DOI 10.17487/RFC6294, June 2011, <<https://www.rfc-editor.org/info/rfc6294>>.
- [RFC6436] Amante, S., Carpenter, B., and S. Jiang, "Rationale for Update to the IPv6 Flow Label Specification", RFC 6436, DOI 10.17487/RFC6436, November 2011, <<https://www.rfc-editor.org/info/rfc6436>>.
- [RFC6437] Amante, S., Carpenter, B., Jiang, S., and J. Rajahalme, "IPv6 Flow Label Specification", RFC 6437, DOI 10.17487/RFC6437, November 2011, <<https://www.rfc-editor.org/info/rfc6437>>.
- [RFC6438] Carpenter, B. and S. Amante, "Using the IPv6 Flow Label for Equal Cost Multipath Routing and Link Aggregation in Tunnels", RFC 6438, DOI 10.17487/RFC6438, November 2011, <<https://www.rfc-editor.org/info/rfc6438>>.
- [RFC7837] Krishnan, S., Kuehlewind, M., Briscoe, B., and C. Ralli, "IPv6 Destination Option for Congestion Exposure (ConEx)", RFC 7837, DOI 10.17487/RFC7837, May 2016, <<https://www.rfc-editor.org/info/rfc7837>>.
- [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>>.
- [RFC8250] Elkins, N., Hamilton, R., and M. Ackermann, "IPv6 Performance and Diagnostic Metrics (PDM) Destination Option", RFC 8250, DOI 10.17487/RFC8250, September 2017, <<https://www.rfc-editor.org/info/rfc8250>>.

[RFC8321] Fioccola, G., Ed., Capello, A., Cociglio, M., Castaldelli, L., Chen, M., Zheng, L., Mirsky, G., and T. Mizrahi, "Alternate-Marking Method for Passive and Hybrid Performance Monitoring", RFC 8321, DOI 10.17487/RFC8321, January 2018, <<https://www.rfc-editor.org/info/rfc8321>>.

Authors' Addresses

Giuseppe Fioccola
Telecom Italia
Torino
Italy

Email: giuseppe.fioccola@telecomitalia.it

Gunter Van de Velde
Nokia
Antwerp
BE

Email: gunter.van_de_velde@nokia.com

Mauro Cociglio
Telecom Italia
Torino
Italy

Email: mauro.cociglio@telecomitalia.it

Praveen Muley
Nokia
Mountain View
USA

Email: praveen.muley@nokia.com

IPv6 Operations
Internet-Draft
Intended status: Informational
Expires: February 22, 2019

J. Linkova
Google
M. Stucchi
RIPE NCC
August 21, 2018

Using Conditional Router Advertisements for Enterprise Multihoming
draft-ietf-v6ops-conditional-ras-08

Abstract

This document discusses the most common scenarios of connecting an enterprise network to multiple ISPs using an address space assigned by an ISP and how the approach proposed in the "ietf-rtgwg-enterprise-pa-multihoming" draft could be applied in those scenarios. The problem of enterprise multihoming without address translation of any form has not been solved yet as it requires both the network to select the correct egress ISP based on the packet source address and hosts to select the correct source address based on the desired egress ISP for that traffic. The "ietf-rtgwg-enterprise-pa-multihoming" document proposes a solution to this problem by introducing a new routing functionality (Source Address Dependent Routing) to solve the uplink selection issue and using Router Advertisements to influence the host source address selection. While the above-mentioned document focuses on solving the general problem and on covering various complex use cases, this document adopts the approach proposed in the "ietf-rtgwg-enterprise-pa-multihoming" draft to provide a solution for a limited number of common use cases. In particular, the focus is on scenarios where an enterprise network has two Internet uplinks used either in primary/backup mode or simultaneously and hosts in that network might not yet properly support multihoming as described in RFC8028.

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 February 22, 2019.

Copyright Notice

Copyright (c) 2018 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 Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	3
1.1. Requirements Language	4
2. Common Enterprise Multihoming Scenarios	4
2.1. Two ISP Uplinks, Primary and Backup	4
2.2. Two ISP Uplinks, Used for Load Balancing	5
3. Conditional Router Advertisements	5
3.1. Solution Overview	5
3.1.1. Uplink Selection	5
3.1.2. Source Address Selection and Conditional RAs	5
3.2. Example Scenarios	8
3.2.1. Single Router, Primary/Backup Uplinks	8
3.2.2. Two Routers, Primary/Backup Uplinks	9
3.2.3. Single Router, Load Balancing Between Uplinks	12
3.2.4. Two Router, Load Balancing Between Uplinks	12
3.2.5. Topologies with Dedicated Border Routers	13
3.2.6. Intra-Site Communication during Simultaneous Uplinks Outage	15
3.2.7. Uplink Damping	15
3.2.8. Routing Packets when the Corresponding Uplink is Unavailable	16
3.3. Solution Limitations	16
3.3.1. Connections Preservation	17
4. IANA Considerations	17
5. Security Considerations	17
5.1. Privacy Considerations	18
6. Acknowledgements	18
7. References	18
7.1. Normative References	18
7.2. Informative References	20

Appendix A. Change Log	20
Authors' Addresses	20

1. Introduction

Multihoming is an obvious requirement for many enterprise networks to ensure the desired level of network reliability. However, using more than one ISP (and address space assigned by those ISPs) introduces the problem of assigning IP addresses to hosts. In IPv4 there is no choice but using [RFC1918] address space and NAT ([RFC3022]) at the network edge ([RFC4116]). Using Provider Independent (PI) address space is not always an option, since it requires running BGP between the enterprise network and the ISPs. Administrative overhead of obtaining and managing PI address space can also be a concern. As IPv6 hosts can, by design, have multiple addresses of the global scope ([RFC4291]), multihoming using provider address looks even easier for IPv6: each ISP assigns an IPv6 block (usually /48) and hosts in the enterprise network have addresses assigned from each ISP block. However using IPv6 PA blocks in multihoming scenario introduces some challenges, including but not limited to:

- o Selecting the correct uplink based on the packet source address;
- o Signaling to hosts that some source addresses should or should not be used (e.g. an uplink to the ISP went down or became available again).

The document [I-D.ietf-rtgwg-enterprise-pa-multihoming] discusses these and other related challenges in detail in relation to the general multihoming scenario for enterprise networks and proposes a solution which relies heavily on the rule 5.5 of the default address selection algorithm ([RFC6724]). The rule 5.5 makes hosts prefer source addresses in a prefix advertised by the next-hop and therefore is very useful in multihomed scenarios when different routers may advertise different prefixes. While [RFC6724] defines the Rule 5.5 as optional, the recent [RFC8028] recommends that multihomed hosts SHOULD support it. Unfortunately that rule has not been widely implemented when this document was written. Therefore network administrators in enterprise networks can't yet assume that all devices in their network support the rule 5.5, especially in the quite common BYOD ("Bring Your Own Device") scenario. However, while it does not seem feasible to solve all the possible multihoming scenarios without relying on rule 5.5, it is possible to provide IPv6 multihoming using provider-assigned (PA) address space for the most common use cases. This document discusses how the general approach described in [I-D.ietf-rtgwg-enterprise-pa-multihoming] can be applied to solve multihoming scenarios when:

- o An enterprise network has two or more ISP uplinks;
- o Those uplinks are used for Internet access in active/backup or load sharing mode w/o any sophisticated traffic engineering requirements;
- o Each ISP assigns the network a subnet from its own PA address space
- o Hosts in the enterprise network are not expected to support the Rule 5.5 of the default address selection algorithm ([RFC6724]).

1.1. Requirements Language

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

2. Common Enterprise Multihoming Scenarios

2.1. Two ISP Uplinks, Primary and Backup

This scenario has the following key characteristics:

- o The enterprise network is using uplinks to two (or more) ISPs for Internet access;
- o Each ISP assigns IPv6 PA address space for the network;
- o Uplink(s) to one ISP is a primary (preferred) one. All other uplinks are backup and are not expected to be used while the primary one is operational;
- o If the primary uplink is operational, all Internet traffic should flow via that uplink;
- o When the primary uplink fails the Internet traffic needs to flow via the backup uplinks;
- o Recovery of the primary uplink needs to trigger the traffic switchover from the backup uplinks back to primary one;
- o Hosts in the enterprise network are not expected to support the Rule 5.5 of the default address selection algorithm ([RFC6724]).

2.2. Two ISP Uplinks, Used for Load Balancing

This scenario has the following key characteristics:

- o The enterprise network is using uplinks to two (or more) ISPs for Internet access;
- o Each ISP assigns an IPv6 PA address space;
- o All the uplinks may be used simultaneously, with the traffic flows being randomly (not necessarily equally) distributed between them;
- o Hosts in the enterprise network are not expected to support the Rule 5.5 of the default address selection algorithm ([RFC6724]).

3. Conditional Router Advertisements

3.1. Solution Overview

3.1.1. Uplink Selection

As discussed in [I-D.ietf-rtgwg-enterprise-pa-multihoming], one of the two main problems to be solved in the enterprise multihoming scenario is the problem of the next-hop (uplink) selection based on the packet source address. For example, if the enterprise network has two uplinks, to ISP_A and ISP_B, and hosts have addresses from subnet_A and subnet_B (belonging to ISP_A and ISP_B respectively) then packets sourced from subnet_A must be sent to ISP_A uplink while packets sourced from subnet_B must be sent to ISP_B uplink. Sending packets with source addresses belonging to one ISP address space to another ISP might cause those packets to be filtered out if those ISPs or their uplinks implement anti-spoofing ingress filtering ([RFC2827], [RFC3704]).

While some work is being done in the Source Address Dependent Routing (SADR) (such as [I-D.ietf-rtgwg-dst-src-routing]), the simplest way to implement the desired functionality currently is to apply a policy which selects a next-hop or an egress interface based on the packet source address. Most SMB/Enterprise grade routers have such functionality available currently.

3.1.2. Source Address Selection and Conditional RAs

Another problem to be solved in the multihoming scenario is the source address selection on hosts. In the normal situation (all uplinks are up/operational) hosts have multiple global unique addresses and can rely on the default address selection algorithm ([RFC6724]) to pick up a source address, while the network is

responsible for choosing the correct uplink based on the source address selected by a host as described in Section 3.1.1. However, some network topology changes (i.e. changing uplink status) might affect the global reachability for packets sourced from the particular prefixes and therefore such changes have to be signaled back to the hosts. For example:

- o An uplink to an ISP_A went down. Hosts should not use addresses from ISP_A prefix;
- o A primary uplink to ISP_A which was not operational has come back up. Hosts should start using the source addresses from ISP_A prefix.

[I-D.ietf-rtgwg-enterprise-pa-multihoming] provides a detailed explanation on why SLAAC (Stateless Address Autoconfiguration, [RFC4862]) and RAs (Router Advertisements, [RFC4861]) are the most suitable mechanism for signaling network topology changes to hosts and thereby influencing the source address selection. Sending a router advertisement to change the preferred lifetime for a given prefix provides the following functionality:

- o deprecating addresses (by sending an RA with the preferred_lifetime set to 0 in the corresponding PIO (Prefix Information option, [RFC4861])) to indicate to hosts that that addresses from that prefix should not be used;
- o making a previously unused (deprecated) prefix usable again (by sending an RA containing a PIO with non-zero preferred lifetime) to indicate to hosts that addresses from that prefix can be used again.

It should be noted that only preferred lifetime for the affected prefix needs to be changed. As the goal is to influence the source address selection algorithm on hosts, not preventing them from forming addresses from a specific prefix, the valid lifetime should not be changed. Actually it would not even be possible for unauthenticated RAs (which is the most common deployment scenario) as Section 5.5.3 of [RFC4862] prevents hosts from setting valid lifetime for addresses to zero unless RAs are authenticated.

To provide the desired functionality, first-hop routers are required to

- o send RA triggered by defined event policies in response to uplink status change event; and

- o while sending periodic or solicited RAs, set the value in the given RA field (e.g. PIO preferred lifetime) based on the uplink status.

The exact definition of the 'uplink status' depends on the network topology and may include conditions like:

- o uplink interface status change;
 - o presence of a particular route in the routing table;
 - o presence of a particular route with a particular attribute (next-hop, tag etc) in the routing table;
 - o protocol adjacency change.
- etc.

In some scenarios, when two routers are providing first-hop redundancy via VRRP (Virtual Router Redundancy Protocol, [RFC5798]), the master-backup status can be considered as a condition for sending RAs and changing the preferred lifetime value. See Section 3.2.2 for more details.

If hosts are provided with ISP DNS servers IPv6 addresses via RDNSS (Router Advertisement Options for DNS Configuration, [RFC8106]) it might be desirable for the conditional RAs to update the Lifetime field of the RDNSS option as well.

The trigger is not only forcing the router to send an unsolicited RA to propagate the topology changes to all hosts. Obviously the RA fields values (like PIO Preferred Lifetime or DNS Server Lifetime) changed by the particular trigger need to stay the same until another event happens causing the value to be updated. E.g. if the ISP_A uplink failure causes the prefix to be deprecated, all solicited and unsolicited RAs sent by the router need to have the Preferred Lifetime for that PIO set to 0 until the uplink comes back up.

It should be noted that the proposed solution is quite similar to the existing requirement L-13 for IPv6 Customer Edge Routers ([RFC7084]) and the documented behavior of homenet devices ([RFC7788]). It is using the same mechanism of deprecating a prefix when the corresponding uplink is not operational, applying it to enterprise network scenario.

3.2. Example Scenarios

This section illustrates how the conditional RAs solution can be applied to most common enterprise multihoming scenarios, described in Section 2.

3.2.1. Single Router, Primary/Backup Uplinks



Figure 1: Single Router, Primary/Backup Uplinks

Let's look at a simple network topology where a single router acts as a border router to terminate two ISP uplinks and as a first-hop router for hosts. Each ISP assigns a /48 to the network, and the ISP_A uplink is a primary one, to be used for all Internet traffic, while the ISP_B uplink is a backup, to be used only when the primary uplink is not operational.

To ensure that packets with source addresses from ISP_A and ISP_B are only routed to ISP_A and ISP_B uplinks respectively, the network administrator needs to configure a policy on R1:

```

IF (packet_source_address is in 2001:db8:1::/48)
  and
  (packet_destination_address is not in (2001:db8:1::/48 or 2001:db8:2::/48))
THEN
  default next-hop is ISP_A_uplink

IF (packet_source_address is in 2001:db8:2::/48)
  and
  (packet_destination_address is not in (2001:db8:1::/48 or 2001:db8:2::/48))
THEN
  default next-hop is ISP_B_uplink
  
```


Under normal circumstances it is desirable that all traffic be sent via the ISP_A uplink, therefore hosts (the host H1 in the example topology figure) should be using source addresses from 2001:db8:1:1::/64. When/if ISP_A uplink fails, hosts should stop using the 2001:db8:1:1::/64 prefix and start using 2001:db8:2:1::/64 until the ISP_A uplink comes back up. To achieve this the router advertisement configuration on the R1 device for the interface facing H1 needs to have the following policy:

```
prefix 2001:db8:1:1::/64 {  
    IF (ISP_A_uplink is up)  
        THEN  
            preferred_lifetime = 604800  
        ELSE  
            preferred_lifetime = 0  
}  
  
prefix 2001:db8:2:1::/64 {  
    IF (ISP_A_Uplink is up)  
        THEN  
            preferred_lifetime = 0  
        ELSE  
            preferred_lifetime = 604800  
}
```

A similar policy needs to be applied to the RDNSS Lifetime if ISP_A and ISP_B DNS servers are used.

3.2.2. Two Routers, Primary/Backup Uplinks

Let's look at a more complex scenario where two border routers are terminating two ISP uplinks (one each), acting as redundant first-hop routers for hosts. The topology is shown on Fig.2

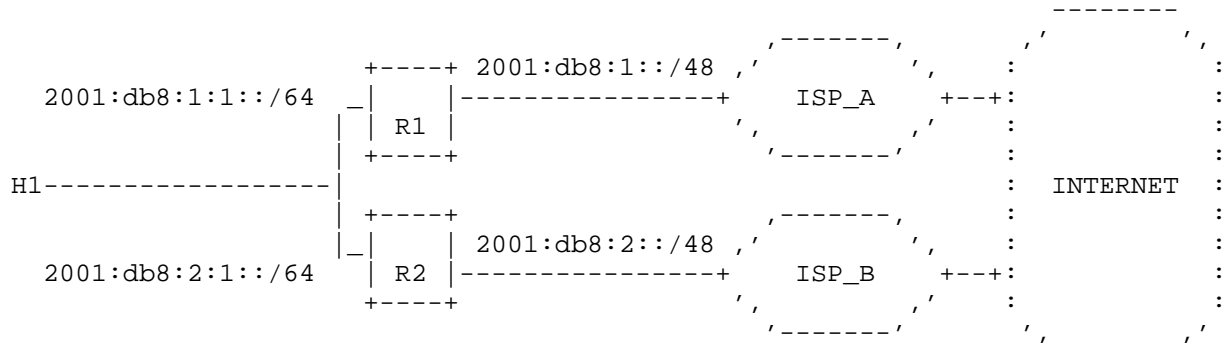


Figure 2: Two Routers, Primary/Backup Uplinks

In this scenario R1 sends RAs with PIO for 2001:db8:1:1::/64 (ISP_A address space) and R2 sends RAs with PIO for 2001:db8:2:1::/64 (ISP_B address space). Each router needs to have a forwarding policy configured for packets received on its hosts-facing interface:

```

IF (packet_source_address is in 2001:db8:1::/48)
  and
  (packet_destination_address is not in (2001:db8:1::/48 or 2001:db8:2::/48))
  THEN
    default next-hop is ISP_A_uplink

IF (packet_source_address is in 2001:db8:2::/48)
i and
  (packet_destination_address is not in (2001:db8:1::/48 or 2001:db8:2::/48))
  THEN
    default next-hop is ISP_B_uplink
  
```

In this case there is more than one way to ensure that hosts are selecting the correct source address based on the uplink status. If VRRP is used to provide first-hop redundancy and the master router is the one with the active uplink, then the simplest way is to use the VRRP mastership as a condition for router advertisement. So, if ISP_A is the primary uplink, the routers R1 and R2 need to be configured in the following way:

R1 is the VRRP master by default (when ISP_A uplink is up). If ISP_A uplink is down, then R1 becomes a backup (the VRRP interface status tracking is expected to be used to automatically modify the VRRP priorities and trigger the mastership switchover). Router

advertisements on R1's interface facing H1 needs to have the following policy applied:

```
prefix 2001:db8:1:1::/64 {  
    IF (vrrp_master)  
        THEN  
            preferred_lifetime = 604800  
        ELSE  
            preferred_lifetime = 0  
    }  
}
```

R2 is VRRP backup by default. Router advertisement on R2 interface facing H1 needs to have the following policy applied:

```
prefix 2001:db8:2:1::/64 {  
    IF(vrrp_master)  
        THEN  
            preferred_lifetime = 604800  
        ELSE  
            preferred_lifetime = 0  
    }  
}
```

If VRRP is not used or interface status tracking is not used for mastership switchover, then each router needs to be able to detect the uplink failure/recovery on the neighboring router, so that RAs with updated preferred lifetime values are triggered. Depending on the network setup various triggers like a route to the uplink interface subnet or a default route received from the uplink can be used. The obvious drawback of using the routing table to trigger the conditional RAs is that some additional configuration is required. For example, if a route to the prefix assigned to the ISP uplink is used as a trigger, then the conditional RA policy would have the following logic:

R1:

```
prefix 2001:db8:1:1::/64 {  
    IF (ISP_A_uplink is up)  
        THEN  
            preferred_lifetime = 604800  
        ELSE  
            preferred_lifetime = 0  
    }  
}
```

R2:

```
prefix 2001:db8:2:1::/64 {
    IF (ISP_A_uplink_route is present)
        THEN
            preferred_lifetime = 0
        ELSE
            preferred_lifetime = 604800
}
```

3.2.3. Single Router, Load Balancing Between Uplinks

Let's look at the example topology shown in Figure 1, but with both uplinks used simultaneously. In this case R1 would send RAs containing PIOs for both prefixes, 2001:db8:1:1::/64 and 2001:db8:2:1::/64, changing the preferred lifetime based on particular uplink availability. If the interface status is used as uplink availability indicator, then the policy logic would look like the following:

```
prefix 2001:db8:1:1::/64 {
    IF (ISP_A_uplink is up)
        THEN
            preferred_lifetime = 604800
        ELSE
            preferred_lifetime = 0
}
prefix 2001:db8:2:1::/64 {
    IF (ISP_B_uplink is up)
        THEN
            preferred_lifetime = 604800
        ELSE
            preferred_lifetime = 0
}
```

R1 needs a forwarding policy to be applied to forward packets to the correct uplink based on the source address similar to one described in Section 3.2.1.

3.2.4. Two Router, Load Balancing Between Uplinks

In this scenario the example topology is similar to the one shown in Figure 2, but both uplinks can be used at the same time. It means that both R1 and R2 need to have the corresponding forwarding policy to forward packets based on their source addresses.

Each router would send RAs with PIO for the corresponding prefix, setting preferred_lifetime to a non-zero value when the ISP uplink is up, and deprecating the prefix by setting the preferred lifetime to 0 in case of uplink failure. The uplink recovery would trigger another

RA with non-zero preferred lifetime to make the addresses from the prefix preferred again. The example RA policy on R1 and R2 would look like:

R1:

```
prefix 2001:db8:1:1::/64 {  
    IF (ISP_A_uplink is up)  
        THEN  
            preferred_lifetime = 604800  
        ELSE  
            preferred_lifetime = 0  
    }  
}
```

R2:

```
prefix 2001:db8:2:1::/64 {  
    IF (ISP_B_uplink is up)  
        THEN  
            preferred_lifetime = 604800  
        ELSE  
            preferred_lifetime = 0  
    }  
}
```

3.2.5. Topologies with Dedicated Border Routers

For simplicity, all topologies above show the ISP uplinks terminated on the first-hop routers. Obviously, the proposed approach can be used in more complex topologies when dedicated devices are used for terminating ISP uplinks. In that case VRRP mastership or interface status can not be used as a trigger for conditional RAs and route presence as described above (Section 3.2.2) should be used instead.

Let's look at the example topology shown on the Figure 3:

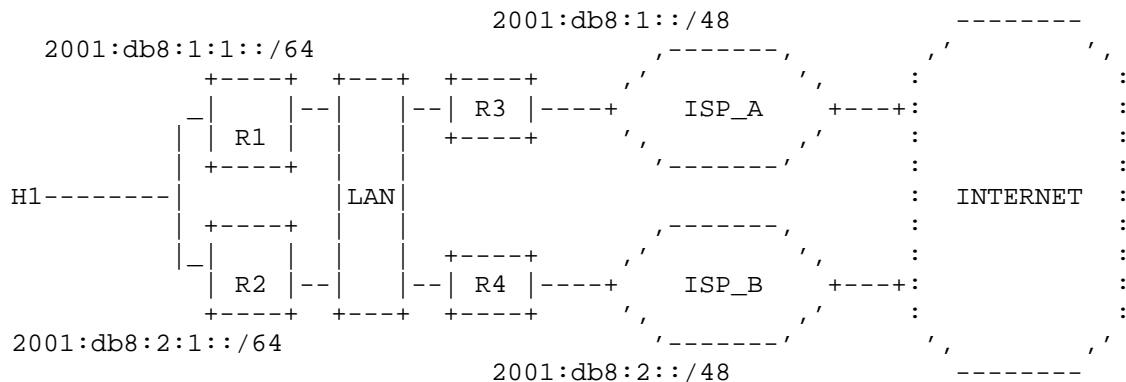


Figure 3: Dedicated Border Routers

For example, if ISP_A is a primary uplink and ISP_B is a backup one then the following policy might be used to achieve the desired behaviour (H1 is using ISP_A address space, 2001:db8:1:1::/64 while ISP_A uplink is up and only using ISP_B 2001:db8:2:1::/64 prefix if the uplink is non-operational):

R1 and R2 policy:

```

prefix 2001:db8:1:1::/64 {
    IF (ISP_A_uplink_route is present)
        THEN
            preferred_lifetime = 604800
        ELSE
            preferred_lifetime = 0
    }

prefix 2001:db8:2:1::/64 {
    IF (ISP_A_uplink_route is present)
        THEN
            preferred_lifetime = 0
        ELSE
            preferred_lifetime = 604800
    }

```

For the load-balancing case the policy would look slightly different: each prefix has non-zero preferred_lifetime only if the corresponding ISP uplink route is present:

```
prefix 2001:db8:1:1::/64 {
    IF (ISP_A_uplink_route is present)
        THEN
            preferred_lifetime = 604800
        ELSE
            preferred_lifetime = 0
}

prefix 2001:db8:2:1::/64 {
    IF (ISP_B_uplink_route is present)
        THEN
            preferred_lifetime = 604800
        ELSE
            preferred_lifetime = 0
}
```

3.2.6. Intra-Site Communication during Simultaneous Uplinks Outage

Prefix deprecation as a result of an uplink status change might lead to a situation when all global prefixes are deprecated (all ISP uplinks are not operational for some reason). Even when there is no Internet connectivity it might be still desirable to have intra-site IPv6 connectivity (especially when the network in question is an IPv6-only one). However while an address is in a deprecated state, its use is discouraged, but not strictly forbidden ([RFC4862]). In such a scenario all IPv6 source addresses in the candidate set ([RFC6724]) are deprecated, which means that they still can be used (as there are no preferred addresses available) and the source address selection algorithm can pick up one of them, allowing the intra-site communication. However some OSes might just fall back to IPv4 if the network interface has no preferred IPv6 global addresses. Therefore if intra-site connectivity is vital during simultaneous outages of multiple uplinks, administrators might consider using ULAs (Unique Local Addresses, [RFC4193]) or provisioning additional backup uplinks to protect the network from double-failure cases.

3.2.7. Uplink Damping

If an actively used uplink (primary one or one used in load balancing scenario) starts flapping, it might lead to the undesirable situation of flapping addresses on hosts (every time the uplink goes up hosts receive an RA with non-zero preferred PIO lifetime, and every time the uplink goes down all addresses in the affected prefix become deprecated). This would, undoubtedly, negatively impact the user experience, not to mention the impact of spikes of duplicate address detection traffic every time an uplink comes back up. Therefore it's recommended that router vendors implement some form of damping policy for conditional RAs and either postpone sending an RA with non-zero

lifetime for a PIO when the uplink comes up for a number of seconds or even introduce accumulated penalties/exponential backoff algorithm for such delays. (In the case of a multiple simultaneous uplink failure scenario, when all but one uplinks are down and the last remaining is flapping it might result in all addresses being deprecated for a while after the flapping uplink recovers.)

3.2.8. Routing Packets when the Corresponding Uplink is Unavailable

Deprecating IPv6 addresses by setting the preferred lifetime to 0 discourage but not strictly forbid its usage in new communications. A deprecated address may still be used for existing connections ([RFC4862]). Therefore when an ISP uplink goes down the corresponding border router might still receive packets with source addresses belonging to that ISP address space while there is no available uplink to send those packets to.

The expected router behaviour would depend on the uplink selection mechanism. For example if some form of SADR is used then such packets will be dropped as there is no route to the destination. If policy-based routing is used to set a next-hop then the behaviour would be implementation-dependent and may vary from dropping the packets to forwarding them based on the routing table entries. It should be noted that there is no return path to the packet source (as the ISP uplink is not operational) therefore even if the outgoing packets are sent to another ISP the return traffic might not be delivered.

3.3. Solution Limitations

It should be noted that the proposed approach is not a "silver bullet" for all possible multihoming scenarios. It would work very well for networks with relatively simple topologies and straightforward routing policies. The more complex the network topology and the corresponding routing policies, the more configuration would be required to implement the solution.

Another limitation is related to the load balancing between the uplinks. In the scenario in which both uplinks are active, hosts would select the source prefix using the Default Address Selection algorithm ([RFC6724]), and therefore the load between two uplinks most likely would not be evenly distributed. (However, the proposed mechanism does allow a creative way of controlling uplinks load in software defined networks where controllers might selectively deprecate prefixes on some hosts but not others to move egress traffic between uplinks). Also the prefix selection does not take into account any other uplinks properties (such as latency etc), so egress traffic might not be sent to the nearest uplink if the

corresponding prefix is selected as a source. In general, if not all uplinks are equal and some uplinks are expected to be preferred over others, then the network administrator should ensure that prefixes from non-preferred ISP(s) are kept deprecated (so primary/backup setup is used).

3.3.1. Connections Preservation

The proposed solution is not designed to preserve connection state after an uplink failure. If all uplinks to an ISP go down, all sessions to/from addresses from that ISP address space are interrupted as there is no egress path for those packets and there is no return path from the Internet to the corresponding prefix. In this regard it is similar to IPv4 multihoming using NAT, where an uplink failure and failover to another uplink means that a public IPv4 address changes and all existing connections are interrupted.

An uplink recovery, however, does not necessarily lead to connections interruption. In the load sharing/balancing scenario an uplink recovery does not affect any existing connections at all. In the active/backup topology when the primary uplink recovers from the failure and the backup prefix is deprecated, the existing sessions (established to/from the backup ISP addresses) can be preserved if the routers are configured as described in Section 3.2.1 and send packets with the backup ISP source addresses to the backup uplink even when the primary one is operational. As a result, the primary uplink recovery makes the usage of the backup ISP addresses discouraged but still possible.

It should be noted that in IPv4 multihoming with NAT, when the egress interface is chosen without taking packet source address into account (as internal hosts usually have addresses from [RFC1918] space), sessions might not be preserved after an uplink recovery unless packet forwarding is integrated with existing NAT sessions tracking.

4. IANA Considerations

This memo asks the IANA for no new parameters.

5. Security Considerations

This memo introduces no new security considerations. It relies on Router Advertisements ([RFC4861]) and SLAAC ([RFC4862]) mechanism and inherits their security properties. If an attacker is able to send a rogue RA they could deprecate IPv6 addresses on hosts or influence source address selection processes on hosts.

The potential attack vectors are including but not limited to:

- o An attacker sends a rogue RA deprecating IPv6 addresses on hosts;
- o An attacker sends a rogue RA making addresses preferred while the corresponding ISP uplink is not operational;
- o An attacker sends a rogue RA making addresses preferred for a backup ISP, steering traffic to undesirable (e.g. more expensive) uplink.

Therefore the network administrators SHOULD secure Router Advertisements, e.g., by deploying RA guard [RFC6105].

5.1. Privacy Considerations

This memo introduces no new privacy considerations.

6. Acknowledgements

Thanks to the following people (in alphabetical order) for their review and feedback: Mikael Abrahamsson, Lorenzo Colitti, Marcus Keane, Erik Kline, David Lamparter, Dusan Mudric, Erik Nordmark, Dave Thaler.

7. References

7.1. Normative References

- [RFC1918] Rekhter, Y., Moskowitz, B., Karrenberg, D., de Groot, G., 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>>.
- [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>>.
- [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>>.
- [RFC3022] Srisuresh, P. and K. Egevang, "Traditional IP Network Address Translator (Traditional NAT)", RFC 3022, DOI 10.17487/RFC3022, January 2001, <<https://www.rfc-editor.org/info/rfc3022>>.

- [RFC3704] Baker, F. and P. Savola, "Ingress Filtering for Multihomed Networks", BCP 84, RFC 3704, DOI 10.17487/RFC3704, March 2004, <<https://www.rfc-editor.org/info/rfc3704>>.
- [RFC4116] Abley, J., Lindqvist, K., Davies, E., Black, B., and V. Gill, "IPv4 Multihoming Practices and Limitations", RFC 4116, DOI 10.17487/RFC4116, July 2005, <<https://www.rfc-editor.org/info/rfc4116>>.
- [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>>.
- [RFC4291] Hinden, R. and S. Deering, "IP Version 6 Addressing Architecture", RFC 4291, DOI 10.17487/RFC4291, February 2006, <<https://www.rfc-editor.org/info/rfc4291>>.
- [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>>.
- [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>>.
- [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>>.
- [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>>.
- [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>>.

7.2. Informative References

- [I-D.ietf-rtgwg-dst-src-routing]
Lamparter, D. and A. Smirnov, "Destination/Source Routing", draft-ietf-rtgwg-dst-src-routing-06 (work in progress), October 2017.
- [I-D.ietf-rtgwg-enterprise-pa-multihoming]
Baker, F., Bowers, C., and J. Linkova, "Enterprise Multihoming using Provider-Assigned Addresses without Network Prefix Translation: Requirements and Solution", draft-ietf-rtgwg-enterprise-pa-multihoming-07 (work in progress), June 2018.
- [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>>.
- [RFC5798] Nadas, S., Ed., "Virtual Router Redundancy Protocol (VRRP) Version 3 for IPv4 and IPv6", RFC 5798, DOI 10.17487/RFC5798, March 2010, <<https://www.rfc-editor.org/info/rfc5798>>.
- [RFC7084] Singh, H., Beebe, W., Donley, C., and B. Stark, "Basic Requirements for IPv6 Customer Edge Routers", RFC 7084, DOI 10.17487/RFC7084, November 2013, <<https://www.rfc-editor.org/info/rfc7084>>.
- [RFC7788] Stenberg, M., Barth, S., and P. Pfister, "Home Networking Control Protocol", RFC 7788, DOI 10.17487/RFC7788, April 2016, <<https://www.rfc-editor.org/info/rfc7788>>.

Appendix A. Change Log

Initial Version: July 2017

Authors' Addresses

Jen Linkova
Google
Mountain View, California 94043
USA

Email: furry@google.com

Massimiliano Stucchi
RIPE NCC
Stationsplein, 11
Amsterdam 1012 AB
The Netherlands

Email: mstucchi@ripe.net

Network Working Group
Internet-Draft
Intended status: Informational
Expires: November 27, 2018

Z. Kahn, Ed.
LinkedIn
J. Brzozowski, Ed.
Comcast
R. White, Ed.
LinkedIn
May 26, 2018

Requirements for IPv6 Routers
draft-ietf-v6ops-ipv6rtr-reqs-04

Abstract

The Internet is not one network, but rather a collection of networks. The interconnected nature of these networks, and the nature of the interconnected systems that make up these networks, is often more fragile than it appears. Perhaps "robust but fragile" is an overstatement, but the actions of each vendor, implementor, and operator in such an interconnected environment can have a major impact on the stability of the overall Internet (as a system). The widespread adoption of IPv6 could, particularly, disrupt network operations, in a way that impacts the entire system.

This time of transition is an opportune time to take stock of lessons learned through the operation of large-scale networks on IPv4, and consider how to apply these lessons to IPv6. This document provides an overview of the design and architectural decisions that attend IPv6 deployment, and a set of IPv6 requirements for routers, switches, and middleboxes deployed in IPv6 networks. The hope of the editors and contributors is to provide the necessary background to guide equipment manufacturers, protocol implementors, and network operators in effective IPv6 deployment.

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 November 27, 2018.

Copyright Notice

Copyright (c) 2018 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 Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	3
1.1. Contributors	3
1.2. Acknowledgments	4
1.3. Use and Applicability	4
2. Review of the Internet Architecture	5
2.1. Robustness Principle	5
2.2. Complexity	7
2.2.1. Elegance	7
2.2.2. Trade-offs	8
2.3. Layered Structure	9
2.4. Routers	10
3. Requirements Related to Device Management and Security	12
3.1. Programmable Device Access	12
3.2. Human Readable Device Access	13
3.3. Supporting Zero Touch Provisioning for Connected Devices	13
3.4. Device Protection against Denial of Service Attacks	15
4. Requirements Related to Telemetry	15
4.1. Device State and Traceability	16
4.2. Topology State and Traceability	16
4.3. Flow State and Traceability	17
5. Requirements Related to IPv6 Forwarding and Addressing	17
5.1. The IPv6 Address is not a Host Identifier	17
5.2. Router IPv6 Addresses	18
5.3. The Maximum Transmission Unit	19
5.4. ICMP Considerations	20
5.5. Machine Access to the Forwarding Table	21
5.6. Processing IPv6 Extension Headers	22
5.7. IPv6 Operation by Default	22
5.8. IPv6 Only Operation	22

5.9. Prefix Length Handling in IPv6 Packet Forwarding	23
5.10. IPv6 Mobility Support	23
6. Security Considerations	23
6.1. Robustness and Security	23
6.2. Programmable Device Access and Security	24
6.3. Zero Touch Provisioning and Security	24
6.4. Defaulting to IPv6 Forwarding and Security	24
7. IANA Considerations	25
8. Conclusion	25
9. References	25
9.1. Normative References	25
9.2. Informative References	25
Authors' Addresses	32

1. Introduction

This memo defines and discusses requirements for devices that perform forwarding for Internet Protocol version 6 (IPv6). The "use and applicability" section below contains more information on the specific target of this draft, and the envisioned use of the draft.

Readers should recognize that while this memo applies to IPv6, routers and middleboxes IPv6 packets will often also process IPv4 packets, forward based on MPLS labels, and potentially process many other protocols. This memo will only discuss IPv4, MPLS, and other protocols as they impact the behavior of an IPv6 forwarding device; no attempt is made to specify requirements for protocols other than IPv6. The reader should, therefore, not count on this document as a "sole source of truth," but rather use this document as a guide.

For IPv4 router requirements, readers are referred to [RFC1812]. For simplicity, the term "devices" is used interchangeably with the phrase "routers and middleboxes" and the term "routers" throughout this document. These three terms represent stylistic differences, rather than substantive differences.

This document is broken into the following sections: a review of Internet architecture and principles, requirements relating to device management, requirements related to telemetry, requirements related to IPv6 forwarding and addressing, and future considerations. Following these sections, a short conclusion is provided for review.

1.1. Contributors

Shawn Zandi, Pete Lumbis, Fred Baker, James Woodyatt, Erik Muller, Lee Howard, and Joe Clarke contributed significant text and ideas to this draft.

1.2. Acknowledgments

The editors and contributors would like to thank Ron Bonica, Lorenzo Coitti, Brian E. Carpenter, Tim Chown, Peter Lothberg, and Mikael Abrahamsson for their comments, edits, and ideas on the text of this draft.

1.3. Use and Applicability

The conceived use of this draft is as a reference point. The first part of the draft is designed to help IPv6 implementors and network operators to understand Internet and Internetworking technologies, so they can better understand the context of IPv6. The second part of this draft outlines a common set of requirements for devices which are designed to forward IPv6 traffic. This can include (but is not limited to) the devices described below.

- o Devices which are primarily designed to forward traffic between more than two interfaces. These are normally referred to by the Internet community as routers or, in some cases, intermediate systems.
- o Devices which are designed to modify packets rather than "just" forwarding them. These are often referred to by the Internet community as middleboxes. See [RFC7663] for a fuller definition of middleboxes.

This draft is not designed to apply to consumer devices, such as smart devices (refrigerators, light bulbs, garage door openers, etc.), Internet of Things (IoT) devices, cell phones primarily used as an end user device (such as checking email, social media, games, and use as a voice device), and other devices of this class. It is up to each provider or equipment purchaser to determine how best to apply this document to their environment.

The intended use of this document is for operators to be able to point to a common set of functionality which should be available across all IPv6 implementations. Several members of the community have argued there is no common set of IPv6 features; rather each deployment of IPv6 calls for different feature sets. However, the authors of this draft believe outlining a common set of features expected of every IPv6 forwarding device is useful. Specifically:

- o If every IPv6 deployment situation is unique, and requires a different set of features, there will not be a solid definition of what an IPv6 forwarding device is, or performs. This fragments the concept of IPv6 forwarding devices in an unhelpful way, especially as IPv6 deployment is already seen as difficult.

- o It encourages developers and vendors to code a multitude of different IPv6 stacks, one for each possible set of features. This fragments the experience with these stacks, potentially preventing the development of a well designed, fully featured stacks the entire community can rely on.

Because this document is designed to be a reference point rather than a best common practice or a standard, this document does not use [RFC2119] upper case "must" and "should" throughout. Rather, it uses lower case "must" and "should" throughout, anticipating operators will find such guidance clear and useful.

2. Review of the Internet Architecture

The Internet relies on a number of basic concepts and considerations. These concepts are not explicitly called out in any specification, nor do they necessarily impact protocol design or packet forwarding directly. This section provides an overview of these concepts and considerations to help the reader understand the larger context of this document.

2.1. Robustness Principle

Every point where multiple protocols interact, is an interaction surface that can threaten the robustness of the overall system. While it may seem the global Internet has achieved a level of stability that makes it immune to such considerations, the reality is every network is a complex system, and is therefore subject to massive non repeatable unanticipated failures. Postel's Robustness Principle countered this problem with a simple statement, explicated in [RFC1122]: "Be conservative in what you do, and liberal in what you accept from others."

However, since this time, it has been noted that following this law allows errors in protocols to accumulate over time, with overall negative effects on the system as a whole. [RFC1918] describes several points in conjunction with this principle that bear updating based on further experience with large-scale protocol and network deployments within the Internet community, including:

- o Applications should deal with error states gracefully; an application should not degrade in a way that will cause the failure of adjacent systems when possible. For instance, when a routing protocol implementation fails, it should not do so in a way that will cause the spreading of or continued existence of false reachability information, nor should it fail in a way that overloads adjacent routers or interacting protocols and causing a cascading failure.

- o It is best to assume the network is filled with poor implementations and malevolent actors, both of which will find every possible failure mode over time.
- o It is best to assume every technology will be used to the limits of its technical capabilities, rather than assuming a particular protocol's scope of use will align (in any way) with the intent of the original designer(s). [RFC5218] defines a wildly successful protocol as one that "far exceeds its original goals, in terms of purpose (being used in scenarios far beyond the initial design), in terms of scale (being deployed on a scale much greater than originally envisaged), or both." Successful implementations attract more functionality, much like a few nodes in a scale free graph eventually become connectivity hubs.
- o Protocols and implementations change over time. A corollary of the assumption that protocols will be used until they reach their technical limits is that protocols will change over time as they gain new functionality. [RFC5218] points out several problems with "wild success" in a protocol: undesirable side effects, performance problems, and becoming a high value attack target. Protocol and implementation design should take into account use cases that have not yet been thought of by building flexibility into protocols. Protocols should also remained focused on a narrow range of use cases; it is often wise to invent a new protocol than to extend a single protocol into a broad set of use cases.
- o Protocols are sometimes replaced or updated to new versions in order to add new capabilities or features. Updating a protocol requires great care in providing for a transition mechanism between older and newer versions. [RFC8170] provides sound advice on protocol transition planning and mechanisms.
- o Obscure, but legal, protocol features are often ignored or left unimplemented. Protocols must handle receiving unexpected information gracefully so they do not fail because of incomplete or partial implementations. Protocols should avoid specifying contradictory states, or features that will cause interoperability issues if multiple implementations choose to implement different feature sets.
- o Monocultures are almost always bad. While multiple implementations can represent an interaction surface which increases complexity, particularly if a broad set of protocol capabilities and/or implementation features are used, using the same implementation at every point in a deployment results in a mono-culture. In a monoculture, a single event can trigger a

defect in every router, causing a network failure. Mono-cultures must be carefully balanced against interaction surfaces; often this is best accomplished by using multiple implementations and minimal, widely implemented, and well understood protocol features.

A summary of the points above might be this: It is important to work within the bounds of what is actually implemented in any given protocol, and to leave corner cases for another day. It is often easy to assume "virtual oceans" are easier to boil than physical ones, or for an ocean to appear much smaller because it is being implemented in software. This is often deceptive. It is never helpful to boil the ocean whether in a design, an implementation, or a protocol.

2.2. Complexity

Complexity, as articulated by Mike O'Dell (see [RFC3439]), is "the primary mechanism which impedes efficient scaling, and as a result is the primary driver of increases in both capital expenditures (CAPEX) and operational expenditures (OPEX)." At the same time, complexity cannot be "solved," but rather must be "managed." The simplest and most obvious solution to any problem is often easy to design, deploy, and manage. It's also often wrong and/or broken. As much as developers, designers, and operators might like to make things as simple as possible, hard problems require complex solutions. See Alderson and Doyle [COMPLEXHARD] for a discussion of the relationship between hard problems and complex solutions.

The following sections contain observations which apply to the management of complexity in both protocol and network design.

2.2.1. Elegance

Elegance should be the goal of protocol and network design. Rather than seeking out simple solutions because they are simple, seek out solutions that will solve the problem in the simplest way possible (and no simpler). Often this will require:

- o Ensuring the goal is actually the goal. Many times the goal is taken from the operational realm into the protocol design realm before enough thought has been applied to ensure the correct problem is being addressed.
- o Seeing the problem from different angles, trying to break the problem up in multiple ways; and trying, abandoning, and rebuilding ideas and implementations until a better way is found.

- o Sometimes the complexity of the solution will overwhelm the use case; sometimes it is better to leave the apparent problem unsolved, or allow the community time and space to find a simpler solution.

2.2.2. Trade-offs

There are always trade-offs. For any protocol, network, or operational design decision, there will always be a trade-off between at least two competing goals. If some problem appears to have a single solution without trade-offs, this doesn't mean the trade-offs don't exist. Rather, it means the trade-offs haven't been discovered yet. In the area of protocol and network design, these trade-offs often take the form of common "choose two of three" situations, such as "quick, cheap, high quality." In network and protocol design, the trade-offs are often:

- o The amount of state carried in the system and the speed at which it changes, or simply the state. The amount of state required to operate a system as it scales tends to be nonlinear. Some instances of this are described in [RFC3439] section 2.2.1, the Amplification Principle.
- o The number of interaction surfaces between the components that make up the complete system, and the depth of those interaction surfaces. Some examples of surfaces are described in [RFC3439]section 2.2.2, the Coupling Principle. Layering is essentially a form of abstraction; all abstractions are subject to the law of leaky abstractions, [LEAKYABS] which states: "all nontrivial abstractions leak."
- o The desired optimization, including efficient use of network resources, optimal support for business objectives, and optimal support for a specific set of applications.

These three make up a "triangle problem." For instance, to increase the optimization of traffic flow through a network generally requires adding more state to the control plane, leading to problems in complexity due to amplification. To reduce amplification, the control plane (or perhaps the various functions the control plane serves) can be broken up into subsystems, or modules. Breaking the control plane up into subsystems, however, introduces interaction surfaces between the components, which is another form of complexity. [RFC7980] provides a good overview of network complexity; in particular, section 3 of that document provides some examples of complexity trade-offs.

2.3. Layered Structure

The Internet data plane is organized around broad top and bottom layers, and much thinner middle layer. This is illustrated in the figure below.

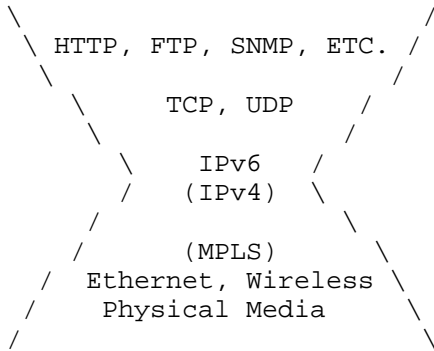


Figure 1

This layering emulates or mirrors many naturally occurring systems; it is a common strategy for managing complexity (see Meyer's presentation on complexity). [COMPLEXLAYER] The single protocol in the center, IPv6, serves to separate the complexity of the lower layers from the complexity of the upper layers. This center layer of the Internet ecosystem has traditionally been called the Network Layer, in reference to the Department of Defense (DoD) [DoD] and OSI models. [OSI] The Internet ecosystem includes two different protocols in this central location.

- o IPv4, an older network protocol that, it is anticipated, will be replaced over time as the Internet ecosystem standardizes on IPv6
- o IPv6, a newer network protocol that is being adopted

MPLS is often used as a "middle" sub-transport layer, and at other times as "middle" sub data link layer; hence MPLS is difficult to classify within the strictly hierarchical model depicted here. These protocols are often treated as if they exist in strict hierarchical layers with a well defined and followed Application Programming Interface (API), data models, Remote Procedure Calls (RPCs), sockets, etc. The reality, however, is there are often solid reasons for violating these layers, creating interaction surfaces that are often deeper than intended or understood without some experience. Beyond this, such layering mechanisms act as information abstractions. It is well known that all such abstractions leak (see above on the law of leaky abstractions). Because of these intentional and

unintentional leakages of information, the interactions between protocols is often subtle.

2.4. Routers

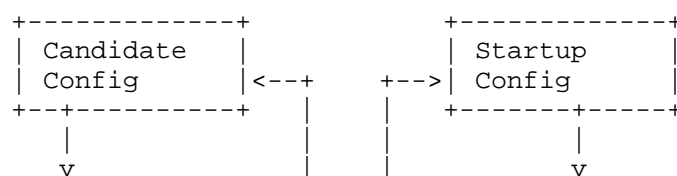
A router connects to two or more logical interfaces and at least one physical interface. A router processes packets by:

- o Receiving a packet through an interface
- o Stripping the data link, physical header, or tunnel encapsulation off the packet
- o Examining the packet for errors, and determining if this packet needs to be punted to another process on the router
- o Looking up the destination in a local forwarding table
- o Rewriting the data link and/or physical layer header
- o Transmitting the packet out an interface

When consulting the forwarding table, the router searches for a match based on:

- o The longest prefix containing the destination address (this is the most common matching element)
- o A label, such as a flow label or MPLS label
- o The source address or other header fields (not common)

The router then examines the information in the matching entry to determine the next hop, or rather the next logically connected device to forward the packet to. The next hop will either be another router, which will presumably carry the packet closer to the final destination, or it will be the destination host itself. The following figure provides a conceptual model of a router; not all routers actually have this set of tables and interactions, and some have many more moving parts. This model is simply used as a common reference to promote understanding.



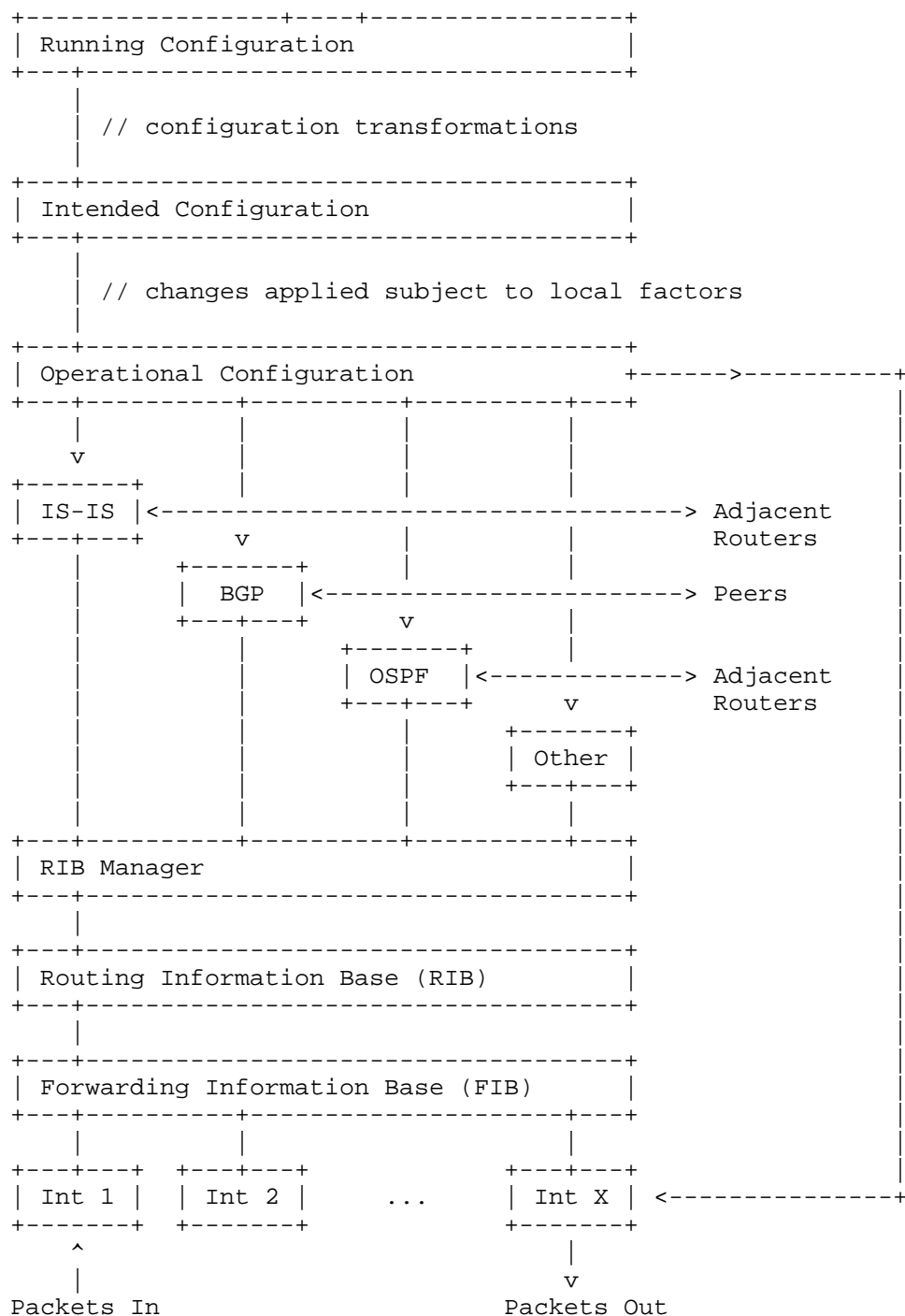


Figure 2

The configuration datastores in this figure follow [RFC8342].

3. Requirements Related to Device Management and Security

Network engineering began in the era of Command Line Interfaces (CLIs), and has generally stayed with these CLIs even as the Graphical User Interface (GUI) has become the standard way of interacting with almost every other computing device. Direct human interaction with routers and middleboxes in large-scale and complex environments, however, tends to result in an unacceptably low Mean Time Between Mistakes (MTBM), directly impacting the overall availability of the network. In reaction to this, operators have increased their reliance on automation, specifically targeting machine to machine interfaces, such as Remote Procedure Calls (RPCs) and Application Programming Interface (API) solutions, to manage and configure routers and middleboxes. This section considers the various components of device management.

Across all interface types, devices should provide and use complete, idempotent, stateless configurations. Further, default settings should be accessible in some way, even if they are hidden by default for configuration readability.

3.1. Programmable Device Access

Configuration primarily relates to the startup, candidate, running, intended, and operational configurations in the router model shown above. In order to deploy networks at scale, operators rely on automated management of router configuration. This effort has traditionally focused on screen scraping and other proprietary methods of "reading" and "writing" configuration information through a CLI. In the future, operators expect to move towards open source/open standards YANG models, regardless of how these are encoded and/or carried (or marshaled).

Vendors and implementors should implement machine readable interfaces with overlays to support human interaction, rather than human readable interfaces with overlays to support machine to machine interaction. Emphasis should be placed on machine to machine interaction for day to day operations, rather than on human readable interfaces, which are largely used in the process of troubleshooting. Within the realm of machine to machine interfaces, emphasis should be placed on marshaling information in YANG models.

To support automated router configuration, IPv6 routers and routers should support YANG configuration, including (but not limited to):

- o Openconfig models [OPENCONF] related to the protocols configured on the device, interface state, and device state
- o [RFC8343]: A YANG Data Model for Interface Management
- o [RFC7224]: IANA Interface Type YANG Module
- o [RFC8344]: A YANG Data Model for IP Management
- o [RFC7317]: A YANG Data Model for System Management
- o [RFC8349]: A YANG Data Model for Routing Management (NMDA Version)

3.2. Human Readable Device Access

To operate a network at scale, operators rely on the ability to access routers and middleboxes to troubleshoot and gather state manually through a number of different interfaces. These interfaces should provide current device configuration, current device state (such as interface state, packets drops, etc.), and current control plane contents (such as the RIB in the figure above). In other words, manual interfaces should provide information about the router (the whole device stack).

To support manual state gathering and troubleshooting, IPv6 routers and middleboxes should support:

- o TELNET ([RFC0854]): TELNET should be disabled by default, but should be available for operational purposes as required or as configured by the operator
- o SSH ([RFC4253]): SSH should be the default access for IPv6 capable routers
- o All network devices supporting IPv6 must support access through an Ethernet management port

3.3. Supporting Zero Touch Provisioning for Connected Devices

To operate a network at scale, operators rely on protocols and mechanisms that reduce provisioning time to a minimum. The preferred state is zero touch provisioning; plug a new router in and it just works without any manual configuration. The closer an operator can come to this ideal, the more MTBM and Operational Expenses (OPEX) can be reduced -- important goals in the real world. IPv6 routers should support several standards, including, but not limited to:

- o [I-D.ietf-dhc-rfc3315bis]: Dynamic Configuration Protocol for IPv6 must be supported.
- o [RFC4862]: IPv6 Stateless Address Autoconfiguration (SLAAC) must be supported, and must be enabled by default on all router interfaces. SLAAC must be able to be disabled by operators who prefer to use some other mechanism for address management and assignment (specifically for customer facing edge ports).
- o [RFC7217]: Semantically Opaque Interface Identifiers should be supported unless there's a need to embed MAC address.
- o [RFC7934]: Host Address Availability, the ability to assign multiple addresses to a host, should be supported.
- o [RFC7527]: Enhanced Duplicate Address Detection should be supported.
- o [RFC7527]: Enhanced Duplicate Address Detection may be disabled for manually configured interfaces.
- o [RFC8028]: First-Hop Router Selection by Hosts, specifically section 2.1, which says a router should be able to send a PIO with both the L and A bits cleared.
- o [RFC3810]: Routers supporting IPv6 must support Multicast Listener Discovery Version 2
- o [RFC7772]: Routers supporting IPv6 should support Reducing Energy Consumption of Router Advertisements
- o [RFC8273]: Routers supporting IPv6 should support Unique IPv6 Prefix per Host

The provisioning of Domain Name Systems (DNS) system information is a contentious topic, based on provider, operating system, interface, and other requirements. This document therefore addresses the mechanisms that must be included in IPv6 router implementations, but leaves the option of what to configure and deploy to the network operator. Routers supporting IPv6, and intended for user facing connections, must support:

- o [RFC3646]: DNS Configuration options for Dynamic Host Configuration Protocol for IPv6 (DHCPv6) if DHCPv6 is supported.
- o [RFC8106]: IPv6 Router Advertisement Options for DNS Configuration. This includes the ability to send Router Advertisements (RAs) with DNS information.

Whether these are enabled by default, or require extra configuration, is left as an exercise for providers and implementation developers to determine on a case by case basis.

3.4. Device Protection against Denial of Service Attacks

Denial of Service (DoS) and Distributed Denial of Service (DDoS) attacks are unfortunately common in the Internet globally; these types of attacks cost network operators a great deal in opportunity and operational costs in prevention and responses. To provide for effective counters to DoS and DDoS attacks directly on routers:

- o Manufacturers and system integrators should test and clearly report the packet/traffic load handling capabilities of devices with and without various encryption methods enabled
- o Routers should be able to police traffic destined to the control plane based on the rate of traffic received, including the ability to police individual flows, targeted services, etc., at individual rates as described in [RFC6192]
- o Ideally, devices should be able to statefully filter traffic destined to the control plane

There are other useful techniques for dealing with DDoS attacks at the network level, including: transferring sessions to a new address and abandoning the address under attack, using BGP communities to spread the attack over multiple ingress ports and "consume" it, and requiring mutual authentication before allocating larger resource pools to a connection. These techniques are not "device level," and hence are not considered further here.

4. Requirements Related to Telemetry

Telemetry relates to information devices push to systems used to monitor and track the state of the network. This applies to individual devices as well as the network as a system. Two major challenges face operators in the area of telemetry:

- o Information that is laid out primarily for human, rather than machine, consumption. While human consumption of telemetry is important in some situations, this information should be supplied in a form that focuses on machine readability with an overlay or interpreter that allows human consumption.
- o Software systems that require information to be queried (or polled or even pushed) on a per-item basis. This form of organization can produce a lot of information, and a lot of individual packets,

very quickly, overwhelming monitoring systems and consuming a large amount of available network resources. Instead, telemetry should be focused on bulk collection.

There are three broad categories of telemetry: device state and traceability, topology state and traceability, and flow traceability. These three roughly correspond to the management plane, the control plane, and the forwarding plane of the network. Each of the sections below considers one of these three telemetry types.

4.1. Device State and Traceability

Ideally, the entire network could be monitored using a single modeling language to ease implementation of telemetry systems and increase the pace at which new software can be deployed in production environments. In real deployments, it is often impossible to reach this ideal; however, reducing the languages and methods used, while focusing on machine readability, can greatly ease the deployment and management of a large-scale network. Specifically, IPv6 routers should support:

- o [RFC6241] and [RFC8040]: NETCONF/RESTCONF transporting telemetry formatted according to YANG (see above)
- o [I-D.ietf-i2rs-yang-l2-network-topology]: An I2RS model for layer 2 topologies
- o [I-D.ietf-netconf-yang-push]: YANG Datastore Subscription
- o [RFC5424]: Syslog
- o gRPC based telemetry interfaces [GRPC]
- o Simple Network Management Protocol (SNMP) MIBs as appropriate

Syslog and SNMP access for telemetry should be considered "legacy," and should not be the focus of new telemetry access development efforts.

4.2. Topology State and Traceability

IPv6 routers are part of a system of devices that, combined, make up the entire network. Viewing the network as a system is often crucial for operational purposes. For instance, being able to understand changes in the topology and utilization of a network can lead to insights about traffic flow and network growth that lead to a greater understanding of how the network is operating, where problems are developing, and how to improve the network's performance. To support

systemic monitoring of the network topology, IPv6 devices should support at least:

- o [RFC5424]: North-Bound Distribution of Link-State and Traffic Engineering (TE) Information using BGP
- o [I-D.ietf-i2rs-yang-l2-network-topology]: An I2RS model for layer 2 topologies
- o [RFC8346]: An I2RS model for layer 3 topologies
- o [RFC8345]: A Data Model for Network Topologies

4.3. Flow State and Traceability

Network operators frequently need to observe and understand the types, sources, and destinations of traffic passing through devices. For example, information about traffic flows may be used to identify abuse (such as DDOS attacks) or to plan network expansions based on traffic patterns. To support insight and analysis of this traffic, IPv6 devices should support IPFIX as described in [RFC7011], PSAMP as described in [RFC5474], or some other flow state mechanism.

In-situ Operational and Management (iOAM) is a technology that being developed at the time of this writing; see [I-D.ietf-ippm-ioam-data]. Operators and vendors should consider the deployment of iOAM to provide deeper information about flow and topology information.

5. Requirements Related to IPv6 Forwarding and Addressing

There are a number of capabilities that a device should have to be deployed into an IPv6 network, and several forwarding plane considerations operators and vendors need to bear in mind. The sections below explain these considerations.

5.1. The IPv6 Address is not a Host Identifier

The IPv6 address is commonly treated as a host identifier; it is not. Rather, it is an interface identifier that describes the topological point where a particular host connects to the Internet. Specifically:

- o The IPv6 address will change when a device changes where it connects to the network.
- o A single host can have multiple addresses. For instance, a host may have one address per interface, or multiple addresses assigned

through different mechanisms, or through multiple connection points.

- o A single IPv6 address may represent many hosts, as in the case of a group of hosts reachable through a multicast address, or a set of services reachable through an anycast address.

Because the host address may change at any time, it is generally harmful to embed IPv6 addresses inside upper layer headers to identify a particular host.

5.2. Router IPv6 Addresses

Internet Routing Registries may allocate a network operator a wide range of prefix lengths (see [RFC6177] for further information). Within this allocation, network operators will often suballocate address space along nibble boundaries (/48, /52, /56, /60, and /64) for ease of configuration and management. Several common practices are:

- o Each multiaccess interface is allocated a /64
- o Point-to-point links are allocated a /64, but should be addressed with a longer prefix length to prevent certain kinds of denial of service attacks ([RFC6547] originally mandated 64 bit prefix lengths on point-to-point links; [RFC6164] explains possible security issues with assigning a 64 bit prefix length to a point-to-point, and recommends a /127 instead)
- o Although aggregation is typically only performed to the nibble boundaries noted above, variances are possible
- o Loopback addresses are assigned a /128

Given these common practices, routers designed to run IPv6 should support the following addressing conventions:

- o The default prefix length on any interface other than a loopback should be a /64
- o Configuring a prefix length longer than a /64 on any multi-access interface should require additional configuration steps to prevent manual configuration errors
- o Routers must not assume IPv6 prefix lengths only on nibble boundaries

- o Routers should support any prefix length shorter or greater than /64
- o Loopback interfaces should default to a /128 prefix length unless some additional configuration is undertaken to override this default setting
- o Routers must be able to generate link local addresses on all links and/or interfaces using stateless address autoconfiguration (see [RFC6434]).

5.3. The Maximum Transmission Unit

The long history of the Maximum Transmission Unit (MTU) in networks is not a happy one. Specific problems with MTU sizing include:

- o Many different default sizes on different media types, from very small (576 bytes on X.25) to very large (17914 bytes on 16Mbps Token Ring)
- o Many different ways to calculate the MTU on any given link; for instance a 9000 byte MTU can be calculated as 8184 bytes on one operating system, 8972 on another, and 9000 on a third
- o The increasing use of tunnel encapsulations in the network; for instance MPLS over GRE over IP over...
- o The wide variety of default MTUs across many different end hosts and operating systems
- o The general ineffectiveness of path MTU discovery to operate correctly in the face of packet filters and rate limiters (see the section on ICMP filtering below)
- o Lower speed links at the network edge which require a lot of time to serialize a packet with a large MTU
- o Increased jitter caused by the disparity between large and small packet size across a lower bandwidth links

The final point requires some further elucidation. The time required to serialize various packets at various speeds are:

- o 64 byte packet onto a 10Mb/s link: .5ms
- o 1500 byte packet onto a 10Mb/s link: 1.2ms
- o 9000 byte packet onto a 10Mb/s link: 7.2ms

- o 64 byte packet onto a 100Mb/s link: .05ms
- o 1500 byte packet onto a 100Mb/s link: .12ms
- o 9000 byte packet onto a 100Mb/s link: .72ms

A 64 byte packet trapped behind a single 1500 byte packet on a 10Mb/s link suffers 1.2ms of serialization delay. Each additional 1500 byte packet added to the queue in front of the 64 byte packet adds an additional 1.2ms of delay. In contrast, a 64 byte packet trapped behind a single 9000 byte packet on a 10Mb/s link suffers 7.7ms of serialization delay. Each additional 9000 byte packet added to the queue adds an additional 7.2ms of serialization delay. The practical result is that larger MTU sizes on lower speed links can add a significant amount of delay and jitter into a flow. On the other hand, increasing the MTU on higher speed links appears to add negligible additional delay and jitter.

The result is that it costs less in terms of overall systemic performance to use higher MTUs on higher speed links than on lower speed links. Based on this, increasing the MTU across any particular link may not increase overall end-to-end performance, but can greatly enhance the performance of local applications (such as a local BGP peering session, or a large/long standing elephant flow used to transfer data across a local fabric), while also providing room for tunnel encapsulations to be added with less impact on lower MTU end systems.

The general rule of thumb is to assume the largest size MTU should be used on higher speed transit only links in order to support a wide array of available link sizes, default MTUs, and tunnel encapsulations. Routers designed for a network core, data center core, or use on the global Internet should support at least 9000 byte MTUs on all interfaces. MTU detection mechanisms, such as IS-IS hello padding, described in [RFC3719], should be enabled to ensure correct point-to-point MTU configuration. Devices should also support:

- o [RFC8201]: Path MTU Discovery for IP version 6
- o [RFC4821]: Packetization Layer Path MTU Discovery

5.4. ICMP Considerations

Internet Control Message Protocol (ICMP) is described in [RFC0792] and [RFC4443]. ICMP is often used to perform a traceroute through a network (normally by using a TTL expired ICMP message), for Path MTU discovery, and, in IPv6, for autoconfiguration and neighbor

discovery. ICMP is often blocked by middleboxes of various kinds and/or ICMP filters configured on the ingress edge of a provider network, most often to prevent the discovery of reachable hosts and network topology. Routers implementing IPv6:

- o Should rate limit the generation of ICMP echo and echo responses by default (for instance, using a token bucket method as described in [RFC4443]). The device should support the configuration of not generating ICMP echo, echo response, and time exceeded packets to prevent topology discovery.
- o Should rate limit the generation of ICMP error messages with a token bucket method as described in [RFC4443]. Rate limits should be narrow enough to (a) protect the device's ability to generate packets and (b) reduce the usefulness of ICMP error packets as part of a distributed denial of service attack. Limits should be generous enough to allow successful path MTU discovery and traceroute. For example, in a small/mid-size device, the possible defaults could be bucket size=100, refill rate=100/s. Larger devices can afford more generous rate limits.
- o Should implement the filtering suggestions in [I-D.gont-opsec-icmp-ingress-filtering]
- o Should not filter Destination Unreachable or Packet Too Big ICMP error messages by default, as this has negative impacts on many aspects of IPv6 operation, particularly path MTU discovery.

There are implications for path MTU discovery and other useful mechanisms in filtering and rate limiting ICMP. The trade-off here is between allowing unlimited ICMP, which would allow path MTU detection to work, or limiting ICMP in a way that prevents negative side effects for individual devices, and hence the operational capabilities of the network as a whole. Operators rightly limit ICMP to reduce the attack surface against their network, as well as the opportunity for "perfect storm" events that inadvertently reduce the capability of routers and middleboxes. Hence ICMP can be treated as "quasi-reliable" in many situations; existence of an ICMP message can prove, for instance, that a particular host is unreachable. The non-existence of an ICMP message, however, does not prove a particular host exists or does not.

5.5. Machine Access to the Forwarding Table

In order to support treating the "network as a whole" as a single programmable system, it is important for each router have the ability to directly program forwarding information. This programmatic interface allows controllers, which are programmed to support

specific business logic and applications, to modify and filter traffic flows without interfering with the distributed control plane. While there are several programmatic interfaces available, this document suggests that the I2RS interface to the RIB be supported in all IPv6 routers. Specifically, these drafts should be supported to enable network programmability:

- o [I-D.ietf-i2rs-fb-rib-data-model]: Filter-Based RIB Data Model
- o [I-D.ietf-i2rs-fb-rib-info-model]: Filter-Based RIB Information Model
- o [I-D.ietf-i2rs-rib-data-model]: A YANG Data Model for Routing Information Base (RIB)
- o [RFC7922]: I2RS Traceability

5.6. Processing IPv6 Extension Headers

(To be added)

5.7. IPv6 Operation by Default

If a device forwards and/or originates IPv4 packets by default (without explicit configuration by the operator), it should forward and/or originate IPv6 packets by default. See the security considerations section below for reflections on the automatic configuration of IPv6 forwarding in parallel with IPv4.

5.8. IPv6 Only Operation

While the transition to IPv6 only networks may take years (or perhaps decades), a number of operators are moving to deploy IPv6 on internal networks supporting transport and data center fabric applications more quickly. Routers and middleboxes that support IPv6 should support IPv6 only operation, including:

- o Link Local addressing must be configurable and usable as the primary address on all interfaces on a device.
- o IPv4 and/or MPLS should not be required for proper device operation. For instance, an IPv4 address should not be required to determine the router ID for any protocol. See [RFC6540] section 2.
- o Any control plane protocol implementations must support the recommendations in [RFC7404] for operation using link local addresses only.

5.9. Prefix Length Handling in IPv6 Packet Forwarding

Routers must support IPv6 destination lookups in the forwarding process on a single bit prefix length increments, in accordance with [RFC7608].

5.10. IPv6 Mobility Support

Mobile IPv6 [RFC6275] and associated specifications, including [RFC3776] and [RFC4877] allow a node to change its point of attachment within the Internet, while maintaining (and using) a permanent address. All communication using the permanent address continues to proceed as expected even as the node moves around. At the present time, Mobile IP has seen only limited implementation. More usage and deployment experience is needed with mobility before any specific approach can be recommended for broad implementation in hosts and routers. Consequently, routers may support [RFC6275] and associated specifications (these specifications are not required for IPv6 routers).

6. Security Considerations

This document addresses several ways in which devices designed to support IPv6 forwarding. Some of the recommendations here are designed to increase device security; for instance, see the section on device access. Others may intersect with security, but are not specifically targeted at security, such as running IPv6 link local only on links. These are not discussed further here, as they improve the security stance of the network. Other areas discussed in this draft are more nuanced. This section gathers the intersection between operational concerns and security concerns into one place.

ICMP security is already considered in the section on ICMP; it will not be considered further here. Link local only addressing will increase security by removing transit only links within the network as a reachable destination.

6.1. Robustness and Security

Robustness, particularly in the area of error handling, largely improves security if designed and implemented correctly. Many attacks take advantage of mistakes in implementations and variations in protocols. In particular, any feature that is unevenly implemented among a number of implementations often offers an attack surface. Hence, reducing protocol complexity helps reduce the breadth of attack surfaces.

Another point to consider at the intersection of robustness and security is the issue of monocultures. Monocultures are in and of themselves a potential attack surface, in that finding a single failure mode can be exploited to take an entire network (or operator) down. On the other hand, reducing the number of implementations for any particular protocol will decrease the set of "random" features deployed in the network. These two goals will often be opposed to one another. Network designers and operators need to consider these two sides of this trade-off, and make an intelligent decision about how much diversity to implement versus how to control the attack surface represented by deploying a wide array of implementations.

6.2. Programmable Device Access and Security

Programmable interfaces, including programmable configuration, telemetry, and machine interface to the routing table, introduce a large attack surface; operators should be careful to ensure this attack surface is properly secured. Specifically:

- o Prevent external access to any administrative access points used for device programmability
- o Use AAA systems to ensure only valid devices and/or users access devices
- o Rate limit the change rate and protect management interfaces from DoS and DDoS attacks

Such interfaces should be treated no differently than SSH, SFTP, and other interfaces available to manage routers and middleboxes.

6.3. Zero Touch Provisioning and Security

Zero touch provisioning opens a new attack surface; insider attackers can simply install a new device, and assume it will be autoconfigured into the network. A "simple" solution would be to install door locks, but this will likely not be enough; defenses need to be layered to be effective. It is recommended that devices installed in the network need to contain a hardware or software identification system that allows the operator to identify devices that are installed in the network.

6.4. Defaulting to IPv6 Forwarding and Security

Operators should be aware that devices which forward IPv6 by default can introduce a new attack surface or new threats without explicit configuration. Operators should verify that IPv6 policies, including filtering, match or fulfill the same intent as any existing IPv4

policies when deploying devices capable of forwarding both IPv4 and IPv6.

7. IANA Considerations

This document has no actions for IANA.

8. Conclusion

The deployment of IPv6 throughout the Internet marks a point in time where it is good to review the overall Internet architecture, and assess the impact on operations of these changes. This document provides an overview of a lot of these changes and lessons learned, as well as providing pointers to many of the relevant documents to understand each topic more deeply.

9. References

9.1. 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>>.

9.2. Informative References

- [COMPLEXHARD]
Alderson, D. and J. Doyle, "Contrasting Views of Complexity and Their Implications For Network-Centric Infrastructures", 2010, <<http://ieeexplore.ieee.org/abstract/document/5477188/?reload=true>>.
- [COMPLEXLAYER]
Meyer, D., "Macro Trends, Architecture, and the Hidden Nature of Complexity", 2010, <<http://www.slideshare.net/dmm613/macro-trends-complexityandsdn-32951199>>.
- [DoD]
Wikipedia, "The Internet Protocol Suite", 2016, <https://en.wikipedia.org/wiki/Internet_protocol_suite>.
- [GRPC]
gRPC, "gRPC", 2016, <<http://www.grpc.io>>.

- [I-D.gont-opsec-icmp-ingress-filtering]
Gont, F., Hunter, R., Massar, J., and W. LIU, "Defeating Attacks which employ Forged ICMPv4/ICMPv6 Error Messages", draft-gont-opsec-icmp-ingress-filtering-03 (work in progress), July 2017.
- [I-D.ietf-dhc-rfc3315bis]
Mrugalski, T., Siodelski, M., Volz, B., Yourtchenko, A., Richardson, M., Jiang, S., Lemon, T., and T. Winters, "Dynamic Host Configuration Protocol for IPv6 (DHCPv6) bis", draft-ietf-dhc-rfc3315bis-13 (work in progress), April 2018.
- [I-D.ietf-i2rs-fb-rib-data-model]
Hares, S., Kini, S., Dunbar, L., Krishnan, R., Bogdanovic, D., and R. White, "Filter-Based RIB Data Model", draft-ietf-i2rs-fb-rib-data-model-01 (work in progress), March 2017.
- [I-D.ietf-i2rs-fb-rib-info-model]
Kini, S., Hares, S., Dunbar, L., Ghanwani, A., Krishnan, R., Bogdanovic, D., and R. White, "Filter-Based RIB Information Model", draft-ietf-i2rs-fb-rib-info-model-00 (work in progress), June 2016.
- [I-D.ietf-i2rs-rib-data-model]
Wang, L., Chen, M., Dass, A., Ananthakrishnan, H., Kini, S., and N. Bahadur, "A YANG Data Model for Routing Information Base (RIB)", draft-ietf-i2rs-rib-data-model-15 (work in progress), May 2018.
- [I-D.ietf-i2rs-yang-l2-network-topology]
Dong, J. and X. Wei, "A YANG Data Model for Layer-2 Network Topologies", draft-ietf-i2rs-yang-l2-network-topology-04 (work in progress), March 2018.
- [I-D.ietf-ippm-ioam-data]
Brockners, F., Bhandari, S., Pignataro, C., Gredler, H., Leddy, J., Youell, S., Mizrahi, T., Mozes, D., Lapukhov, P., Chang, R., daniel.bernier@bell.ca, d., and J. Lemon, "Data Fields for In-situ OAM", draft-ietf-ippm-ioam-data-02 (work in progress), March 2018.
- [I-D.ietf-netconf-yang-push]
Clemm, A., Voit, E., Prieto, A., Tripathy, A., Nilsen-Nygaard, E., Bierman, A., and B. Lengyel, "YANG Datastore Subscription", draft-ietf-netconf-yang-push-15 (work in progress), February 2018.

- [LEAKYABS] Spolsky, J., "The Law of Leaky Abstractions", 2002, <<https://www.joelonsoftware.com/2002/11/11/the-law-of-leaky-abstractions/>>.
- [OPENCONF] OpenConfig, "Openconfig release YANG models", 2016, <<https://github.com/openconfig/public/tree/master/release>>.
- [OSI] Wikipedia, "OSI Model", 2016, <https://en.wikipedia.org/wiki/OSI_model>.
- [RFC0792] Postel, J., "Internet Control Message Protocol", STD 5, RFC 792, DOI 10.17487/RFC0792, September 1981, <<https://www.rfc-editor.org/info/rfc792>>.
- [RFC0854] Postel, J. and J. Reynolds, "Telnet Protocol Specification", STD 8, RFC 854, DOI 10.17487/RFC0854, May 1983, <<https://www.rfc-editor.org/info/rfc854>>.
- [RFC1122] Braden, R., Ed., "Requirements for Internet Hosts - Communication Layers", STD 3, RFC 1122, DOI 10.17487/RFC1122, October 1989, <<https://www.rfc-editor.org/info/rfc1122>>.
- [RFC1812] Baker, F., Ed., "Requirements for IP Version 4 Routers", RFC 1812, DOI 10.17487/RFC1812, June 1995, <<https://www.rfc-editor.org/info/rfc1812>>.
- [RFC1918] Rekhter, Y., Moskowitz, B., Karrenberg, D., de Groot, G., 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>>.
- [RFC3439] Bush, R. and D. Meyer, "Some Internet Architectural Guidelines and Philosophy", RFC 3439, DOI 10.17487/RFC3439, December 2002, <<https://www.rfc-editor.org/info/rfc3439>>.
- [RFC3646] Droms, R., Ed., "DNS Configuration options for Dynamic Host Configuration Protocol for IPv6 (DHCPv6)", RFC 3646, DOI 10.17487/RFC3646, December 2003, <<https://www.rfc-editor.org/info/rfc3646>>.

- [RFC3719] Parker, J., Ed., "Recommendations for Interoperable Networks using Intermediate System to Intermediate System (IS-IS)", RFC 3719, DOI 10.17487/RFC3719, February 2004, <<https://www.rfc-editor.org/info/rfc3719>>.
- [RFC3776] Arkko, J., Devarapalli, V., and F. Dupont, "Using IPsec to Protect Mobile IPv6 Signaling Between Mobile Nodes and Home Agents", RFC 3776, DOI 10.17487/RFC3776, June 2004, <<https://www.rfc-editor.org/info/rfc3776>>.
- [RFC3810] Vida, R., Ed. and L. Costa, Ed., "Multicast Listener Discovery Version 2 (MLDv2) for IPv6", RFC 3810, DOI 10.17487/RFC3810, June 2004, <<https://www.rfc-editor.org/info/rfc3810>>.
- [RFC4253] Ylonen, T. and C. Lonvick, Ed., "The Secure Shell (SSH) Transport Layer Protocol", RFC 4253, DOI 10.17487/RFC4253, January 2006, <<https://www.rfc-editor.org/info/rfc4253>>.
- [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>>.
- [RFC4821] Mathis, M. and J. Heffner, "Packetization Layer Path MTU Discovery", RFC 4821, DOI 10.17487/RFC4821, March 2007, <<https://www.rfc-editor.org/info/rfc4821>>.
- [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>>.
- [RFC4877] Devarapalli, V. and F. Dupont, "Mobile IPv6 Operation with IKEv2 and the Revised IPsec Architecture", RFC 4877, DOI 10.17487/RFC4877, April 2007, <<https://www.rfc-editor.org/info/rfc4877>>.
- [RFC5218] Thaler, D. and B. Aboba, "What Makes for a Successful Protocol?", RFC 5218, DOI 10.17487/RFC5218, July 2008, <<https://www.rfc-editor.org/info/rfc5218>>.
- [RFC5424] Gerhards, R., "The Syslog Protocol", RFC 5424, DOI 10.17487/RFC5424, March 2009, <<https://www.rfc-editor.org/info/rfc5424>>.

- [RFC5474] Duffield, N., Ed., Chiou, D., Claise, B., Greenberg, A., Grossglauser, M., and J. Rexford, "A Framework for Packet Selection and Reporting", RFC 5474, DOI 10.17487/RFC5474, March 2009, <<https://www.rfc-editor.org/info/rfc5474>>.
- [RFC6164] Kohno, M., Nitzan, B., Bush, R., Matsuzaki, Y., Colitti, L., and T. Narten, "Using 127-Bit IPv6 Prefixes on Inter-Router Links", RFC 6164, DOI 10.17487/RFC6164, April 2011, <<https://www.rfc-editor.org/info/rfc6164>>.
- [RFC6177] Narten, T., Huston, G., and L. Roberts, "IPv6 Address Assignment to End Sites", BCP 157, RFC 6177, DOI 10.17487/RFC6177, March 2011, <<https://www.rfc-editor.org/info/rfc6177>>.
- [RFC6192] Dugal, D., Pignataro, C., and R. Dunn, "Protecting the Router Control Plane", RFC 6192, DOI 10.17487/RFC6192, March 2011, <<https://www.rfc-editor.org/info/rfc6192>>.
- [RFC6241] Enns, R., Ed., Bjorklund, M., Ed., Schoenwaelder, J., Ed., and A. Bierman, Ed., "Network Configuration Protocol (NETCONF)", RFC 6241, DOI 10.17487/RFC6241, June 2011, <<https://www.rfc-editor.org/info/rfc6241>>.
- [RFC6275] Perkins, C., Ed., Johnson, D., and J. Arkko, "Mobility Support in IPv6", RFC 6275, DOI 10.17487/RFC6275, July 2011, <<https://www.rfc-editor.org/info/rfc6275>>.
- [RFC6434] Jankiewicz, E., Loughney, J., and T. Narten, "IPv6 Node Requirements", RFC 6434, DOI 10.17487/RFC6434, December 2011, <<https://www.rfc-editor.org/info/rfc6434>>.
- [RFC6540] George, W., Donley, C., Liljenstolpe, C., and L. Howard, "IPv6 Support Required for All IP-Capable Nodes", BCP 177, RFC 6540, DOI 10.17487/RFC6540, April 2012, <<https://www.rfc-editor.org/info/rfc6540>>.
- [RFC6547] George, W., "RFC 3627 to Historic Status", RFC 6547, DOI 10.17487/RFC6547, February 2012, <<https://www.rfc-editor.org/info/rfc6547>>.
- [RFC7011] Claise, B., Ed., Trammell, B., Ed., and P. Aitken, "Specification of the IP Flow Information Export (IPFIX) Protocol for the Exchange of Flow Information", STD 77, RFC 7011, DOI 10.17487/RFC7011, September 2013, <<https://www.rfc-editor.org/info/rfc7011>>.

- [RFC7217] Gont, F., "A Method for Generating Semantically Opaque Interface Identifiers with IPv6 Stateless Address Autoconfiguration (SLAAC)", RFC 7217, DOI 10.17487/RFC7217, April 2014, <<https://www.rfc-editor.org/info/rfc7217>>.
- [RFC7224] Bjorklund, M., "IANA Interface Type YANG Module", RFC 7224, DOI 10.17487/RFC7224, May 2014, <<https://www.rfc-editor.org/info/rfc7224>>.
- [RFC7317] Bierman, A. and M. Bjorklund, "A YANG Data Model for System Management", RFC 7317, DOI 10.17487/RFC7317, August 2014, <<https://www.rfc-editor.org/info/rfc7317>>.
- [RFC7404] Behringer, M. and E. Vyncke, "Using Only Link-Local Addressing inside an IPv6 Network", RFC 7404, DOI 10.17487/RFC7404, November 2014, <<https://www.rfc-editor.org/info/rfc7404>>.
- [RFC7527] Asati, R., Singh, H., Beebee, W., Pignataro, C., Dart, E., and W. George, "Enhanced Duplicate Address Detection", RFC 7527, DOI 10.17487/RFC7527, April 2015, <<https://www.rfc-editor.org/info/rfc7527>>.
- [RFC7608] Boucadair, M., Petrescu, A., and F. Baker, "IPv6 Prefix Length Recommendation for Forwarding", BCP 198, RFC 7608, DOI 10.17487/RFC7608, July 2015, <<https://www.rfc-editor.org/info/rfc7608>>.
- [RFC7663] Trammell, B., Ed. and M. Kuehlewind, Ed., "Report from the IAB Workshop on Stack Evolution in a Middlebox Internet (SEMI)", RFC 7663, DOI 10.17487/RFC7663, October 2015, <<https://www.rfc-editor.org/info/rfc7663>>.
- [RFC7772] Yourtchenko, A. and L. Colitti, "Reducing Energy Consumption of Router Advertisements", BCP 202, RFC 7772, DOI 10.17487/RFC7772, February 2016, <<https://www.rfc-editor.org/info/rfc7772>>.
- [RFC7922] Clarke, J., Salgueiro, G., and C. Pignataro, "Interface to the Routing System (I2RS) Traceability: Framework and Information Model", RFC 7922, DOI 10.17487/RFC7922, June 2016, <<https://www.rfc-editor.org/info/rfc7922>>.
- [RFC7934] Colitti, L., Cerf, V., Cheshire, S., and D. Schinazi, "Host Address Availability Recommendations", BCP 204, RFC 7934, DOI 10.17487/RFC7934, July 2016, <<https://www.rfc-editor.org/info/rfc7934>>.

- [RFC7980] Behringer, M., Retana, A., White, R., and G. Huston, "A Framework for Defining Network Complexity", RFC 7980, DOI 10.17487/RFC7980, October 2016, <<https://www.rfc-editor.org/info/rfc7980>>.
- [RFC7991] Hoffman, P., "The "xml2rfc" Version 3 Vocabulary", RFC 7991, DOI 10.17487/RFC7991, December 2016, <<https://www.rfc-editor.org/info/rfc7991>>.
- [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>>.
- [RFC8040] Bierman, A., Bjorklund, M., and K. Watsen, "RESTCONF Protocol", RFC 8040, DOI 10.17487/RFC8040, January 2017, <<https://www.rfc-editor.org/info/rfc8040>>.
- [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>>.
- [RFC8170] Thaler, D., Ed., "Planning for Protocol Adoption and Subsequent Transitions", RFC 8170, DOI 10.17487/RFC8170, May 2017, <<https://www.rfc-editor.org/info/rfc8170>>.
- [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>>.
- [RFC8342] Bjorklund, M., Schoenwaelder, J., Shafer, P., Watsen, K., and R. Wilton, "Network Management Datastore Architecture (NMDA)", RFC 8342, DOI 10.17487/RFC8342, March 2018, <<https://www.rfc-editor.org/info/rfc8342>>.
- [RFC8343] Bjorklund, M., "A YANG Data Model for Interface Management", RFC 8343, DOI 10.17487/RFC8343, March 2018, <<https://www.rfc-editor.org/info/rfc8343>>.
- [RFC8344] Bjorklund, M., "A YANG Data Model for IP Management", RFC 8344, DOI 10.17487/RFC8344, March 2018, <<https://www.rfc-editor.org/info/rfc8344>>.

- [RFC8345] Clemm, A., Medved, J., Varga, R., Bahadur, N., Ananthakrishnan, H., and X. Liu, "A YANG Data Model for Network Topologies", RFC 8345, DOI 10.17487/RFC8345, March 2018, <<https://www.rfc-editor.org/info/rfc8345>>.
- [RFC8346] Clemm, A., Medved, J., Varga, R., Liu, X., Ananthakrishnan, H., and N. Bahadur, "A YANG Data Model for Layer 3 Topologies", RFC 8346, DOI 10.17487/RFC8346, March 2018, <<https://www.rfc-editor.org/info/rfc8346>>.
- [RFC8349] Lhotka, L., Lindem, A., and Y. Qu, "A YANG Data Model for Routing Management (NMDA Version)", RFC 8349, DOI 10.17487/RFC8349, March 2018, <<https://www.rfc-editor.org/info/rfc8349>>.

Authors' Addresses

Zaid Ali Kahn (editor)
LinkedIn
CA
USA

Email: zaid@linkedin.com

John Brzozowski (editor)
Comcast
USA

Email: John_Brzozowski@comcast.com

Russ White (editor)
LinkedIn
Oak Island, NC 28465
USA

Email: russ@riw.us

v6ops
Internet-Draft
Intended status: Informational
Expires: December 28, 2018

J. Palet Martinez
The IPv6 Company
June 26, 2018

NAT64/464XLAT Deployment Guidelines in Operator and Enterprise Networks
draft-palet-v6ops-nat64-deployment-02

Abstract

This document describes how NAT64 and 464XLAT can be deployed in an IPv6 operator (cellular and broadband) or enterprise network and the issues to be considered when having an IPv6-only access link, regarding: a) DNS64, b) applications or devices that use literal IPv4 addresses or non-IPv6 compliant APIs, and c) IPv4-only hosts or applications.

Status of This Memo

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

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

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

This Internet-Draft will expire on December 28, 2018.

Copyright Notice

Copyright (c) 2018 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 Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as

described in the Simplified BSD License.

Table of Contents

1. Introduction	3
2. Requirements Language	4
3. NAT64 Deployment Scenarios	4
3.1. Known to Work	5
3.1.1. Service Provider NAT64 with DNS64	5
3.1.2. Service Provider offering 464XLAT, with DNS64	7
3.1.3. Service Provider offering 464XLAT, without DNS64	9
3.2. Known to Work Under Special Conditions	10
3.2.1. Service Provider NAT64 without DNS64	10
3.2.2. Service Provider NAT64; DNS64 in the IPv6 hosts	11
3.2.3. Service Provider NAT64; DNS64 in the IPv4-only remote network	12
3.3. Comparing the Scenarios	12
4. Issues to be Considered	13
4.1. DNSSEC Considerations and Possible Approaches	14
4.1.1. Not using DNS64	15
4.1.2. DNSSEC validator aware of DNS64	16
4.1.3. Stub validator	16
4.1.4. CLAT with DNS proxy and validator	16
4.1.5. ACL of clients	17
4.1.6. Mapping-out IPv4 addresses	17
4.2. DNS64 and Reverse Mapping	17
4.3. Using 464XLAT with/without DNS64	17
4.4. Manual Configuration of Foreign DNS	18
4.5. Well-Known Prefix (WKP) vs Network-Specific Prefix (NSP)	19
4.6. IPv4 literals and old APIs	19
4.7. IPv4-only Hosts or Applications	20
4.8. CLAT Translation Considerations	20
5. Summary of Deployment Recommendations for NAT64	20
6. Deployment of NAT64 in Enterprise Networks	23
7. Security Considerations	24
8. IANA Considerations	24
9. Acknowledgements	24
10. ANNEX A: Example of Broadband Deployment with 464XLAT	25
11. ANNEX B: CLAT Implementation	28
12. References	29
12.1. Normative References	29
12.2. Informative References	30
Author's Address	31

1. Introduction

NAT64 ([RFC6146]) describes a stateful IPv6 to IPv4 translation, which allows IPv6-only hosts to contact IPv4 servers using unicast UDP, TCP or ICMP, by means of a single or a set of IPv4 public addresses assigned to the translator, to be shared by the IPv6-only clients.

The translation of the packet headers is done using the IP/ICMP Translation Algorithm defined in [RFC7915] and algorithmically translating the IPv4-hosts addresses to IPv6 ones following [RFC6052].

To avoid changes in both, the IPv6-only hosts and the IPv4-only server, NAT64 requires also the use of a DNS64 ([RFC6147]), in charge for the synthesis of AAAA records from the A records.

However, the use of NAT64 and/or DNS64 present three issues:

- a. Because DNS64 ([RFC6147]) modifies DNS answers, and DNSSEC is designed to detect such modifications, DNS64 ([RFC6147]) can potentially break DNSSEC, depending on a number of factors, such as the location of the DNS64 function (at a DNS server or validator, at the end host, ...), how as been configured, if the end-hosts is validating, etc.
- b. Because the need of using DNS64 ([RFC6147]), there is a major issue for NAT64 ([RFC6146]), as doesn't work when literal addresses or non-IPv6 compliant APIs are being used.
- c. NAT64 alone, doesn't provide a solution for IPv4-only hosts or applications located within a network which are connected to a service provider IPv6-only access.

The same issues are true if part of an enterprise or similar network, is connected to other parts of the same network or third party networks by means of IPv6-only links.

According to that, across this document, the use of "operator network" is interchangeable with equivalent cases of enterprise (or similar) networks.

This document looks into different possible NAT64 ([RFC6146]) deployment scenarios, including 464XLAT ([RFC6877]) ones, in operators (broadband and cellular) and enterprise networks, and provides guidelines to avoid the above-mentioned issues.

Towards that, this document first looks into the possible NAT64

deployment scenarios (split in "known to work" and "known to work under special conditions"), providing a quick and generic comparison table among them. Then describes the issues that an operator need to understand on different matters that will allow to define what is the best approach/scenario for each specific network case. A summary provides some recommendations and decision points and then a clarification of the usage of this document for enterprise networks is provided. Finally, an Annex provides an example of a broadband deployment using 464XLAT.

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. NAT64 Deployment Scenarios

Section 7 of DNS64 ([RFC6147]), provides 3 scenarios, looking at the location of the DNS64. However, since the publication of that document, there are new possible scenarios and NAT64 use cases that need to be considered now, despite they were specifically ruled out of the original NAT64/DNS64 work.

Consequently, the perspective in this document is to broader those scenarios, including a few new ones. However, in order to be able to reduce the number of possible cases, we work under the assumption that the service provider wants to make sure that all the customers have a service without failures. This means considering the worst possible case:

- a. There are hosts that will be validating DNSSEC.
- b. Literal addresses and non-IPv6 compliant APIs are being used.
- c. There are IPv4-only hosts or applications beyond the IPv6-only link.

We use a common set of possible "participant entities":

1. An IPv6-only access network (IPv6).
2. An IPv4-only remote network/server/services (IPv4).
3. The NAT64 function (NAT64) in the service provider.

4. The DNS64 function (DNS64) in the service provider.
5. An external service provider offering the NAT64 and/or the DNS64 function (extNAT64/extDNS64).
6. 464XLAT customer side translator (CLAT).

We split the possible scenarios in two general categories:

1. Known to work.
2. Known to work under special conditions.

3.1. Known to Work

The scenarios in this category are known to work. Each one may have different pros and cons, and in some cases the trade-offs, maybe acceptable for some operators.

3.1.1. Service Provider NAT64 with DNS64

In this scenario, the service provider offers both, the NAT64 and the DNS64 function.

This is probably the most common scenario, however also has the implications related the DNSSEC.

This scenario also fails to solve the issue of literal addresses or non-IPv6 compliant APIs, as well as the issue of IPv4-only hosts or applications inside the IPv6-only access network.

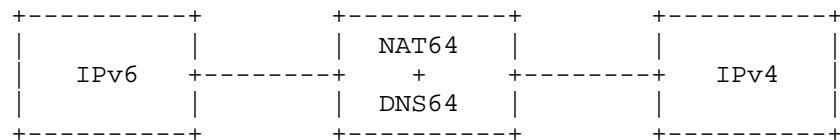


Figure 1: NAT64 with DNS64

A totally equivalent scenario will be if the service provider offers only the DNS64 function, and the NAT64 function is provided by an outsourcing agreement with an external provider. All the considerations in the previous paragraphs of this section are the same for this sub-case.

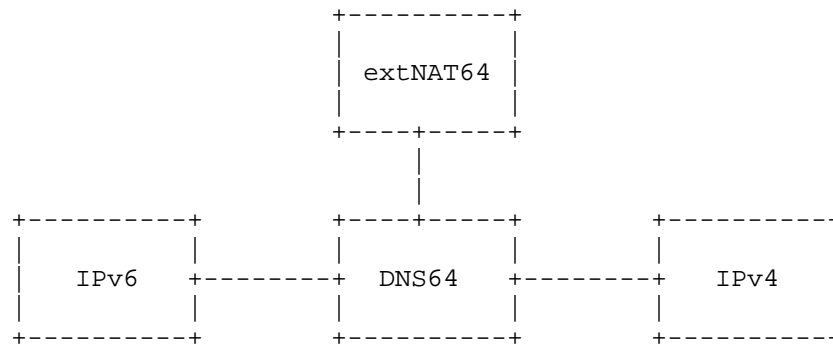


Figure 2: NAT64 in external service provider

As well, is equivalent to the scenario where the outsourcing agreement with the external provider is to provide both the NAT64 and DNS64 functions. Once more, all the considerations in the previous paragraphs of this section are the same for this sub-case.

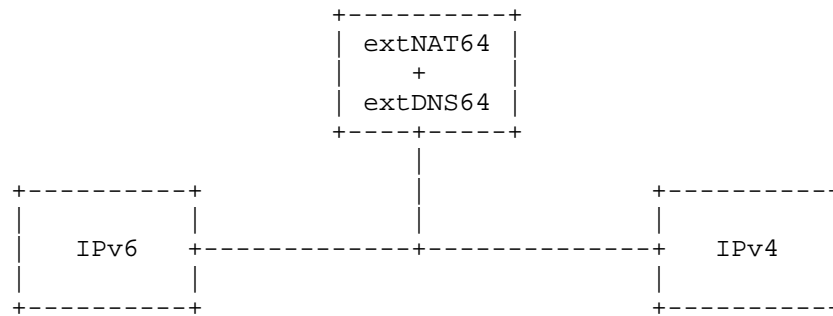


Figure 3: NAT64 and DNS64 in external provider

One more equivalent scenario will be if the service provider offers the NAT64 only, and the DNS64 function is from an external provider with or without a specific agreement among them. This is an scenario already feasible today, as several "global" service providers provide free DNS64 services and users often configure manually their DNS. This will only work if both the NAT64 and the DNS64 are using the same WKP (Well-Known Prefix) or NSP (Network-Specific Prefix). All the considerations in the previous paragraphs of this section are the same for this sub-case.

Of course, if the external DNS64 is agreed with the service provider, then we are in the same case as in the previous ones already depicted in this scenario.

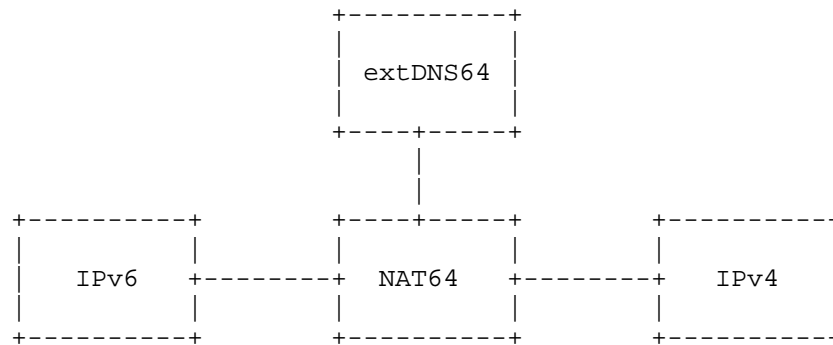


Figure 4: NAT64; DNS64 by external provider

3.1.2. Service Provider offering 464XLAT, with DNS64

464XLAT ([RFC6877]) describes an architecture that provides IPv4 connectivity across a network, or part of it, when it is only natively transporting IPv6.

In order to do that, 464XLAT ([RFC6877]) relies on the combination of existing protocols:

1. The customer-side translator (CLAT) is a stateless IPv4 to IPv6 translator (NAT46) ([RFC7915]) implemented in the end-user device or CE, located at the "customer" edge of the network.
2. The provider-side translator (PLAT) is a stateful NAT64 ([RFC6146]), implemented typically at the opposite edge of the operator network, that provides access to both IPv4 and IPv6 upstreams.
3. Optionally, DNS64 ([RFC6147]), implemented as part of the PLAT allows an optimization (a single translation at the NAT64, instead of two translations - NAT46+NAT64), when the application at the end-user device supports IPv6 DNS (uses AAAA RR).

Note that even in the 464XLAT ([RFC6877]) terminology, the provider-side translator is referred as PLAT, for simplicity and uniformity, in this document is always referred as NAT64.

In this scenario the service provider deploys 464XLAT with DNS64.

As a consequence, the DNSSEC issues remain.

464XLAT ([RFC6877]) is a very simple approach to cope with the major NAT64+DNS64 drawback: Not working with applications or devices that

use literal IPv4 addresses or non-IPv6 compliant APIs.

464XLAT ([RFC6877]) has been used initially in IPv6 cellular networks, providing an IPv6-only access network. By supporting CLAT, the end-user device applications can access IPv4-only end-networks/applications, despite those applications or devices use literal IPv4 addresses or non-IPv6 compliant APIs.

In addition to that, in the same example of the cellular network above, if the User Equipment (UE) provides tethering, other devices behind it will be presented with a traditional NAT44, in addition to the native IPv6 support, so clearly it allows IPv4-only hosts inside the IPv6-only access network.

Furthermore, as indicated in [RFC6877] (464XLAT), can be used in broadband IPv6 network architectures, by implementing the CLAT functionality at the CE.

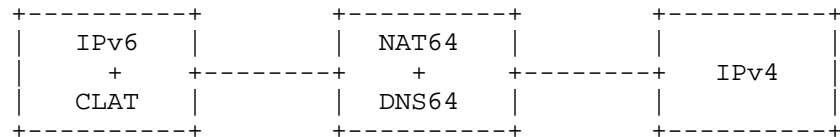


Figure 5: 464XLAT with DNS64

An equivalent scenario will be if the service provider offers only the DNS64 function, and the NAT64 function is provided by an outsourcing agreement with an external provider. All the considerations in the previous paragraphs of this section are the same for this sub-case.

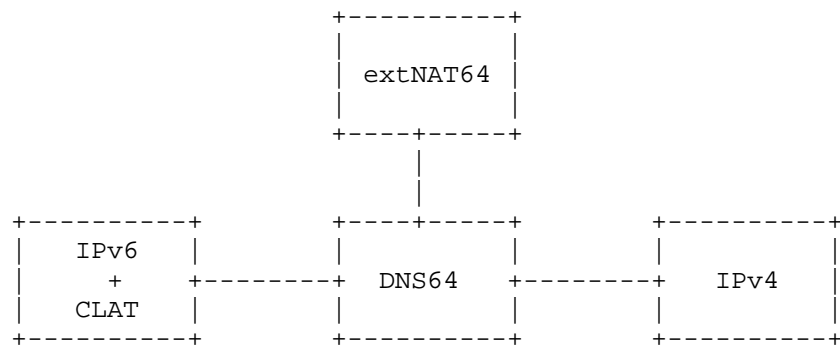


Figure 6: 464XLAT with DNS64; NAT64 in external provider

As well, is equivalent to the scenario where the outsourcing

agreement with the external provider is to provide both the NAT64 and DNS64 functions. Once more, all the considerations in the previous paragraphs of this section are the same for this sub-case.

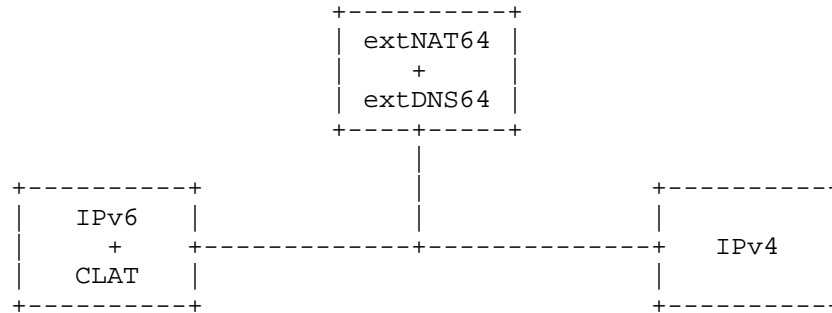


Figure 7: 464XLAT with DNS64; NAT64 and DNS64 in external provider

3.1.3. Service Provider offering 464XLAT, without DNS64

The major advantage of this scenario, using 464XLAT without DNS64, is that the service provider ensures that DNSSEC is never broken.

In this scenario, as in the previous one, there are no issues related to IPv4-only hosts inside the IPv6-only access network, neither to the usage of IPv4 literals or non-IPv6 compliant APIs.

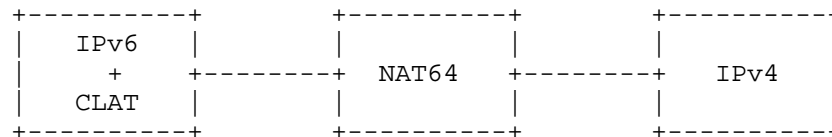


Figure 8: 464XLAT without DNS64

This is equivalent to the scenario where there is an outsourcing agreement with an external provider for the NAT64 function. All the considerations in the previous paragraphs of this section are the same for this sub-case.

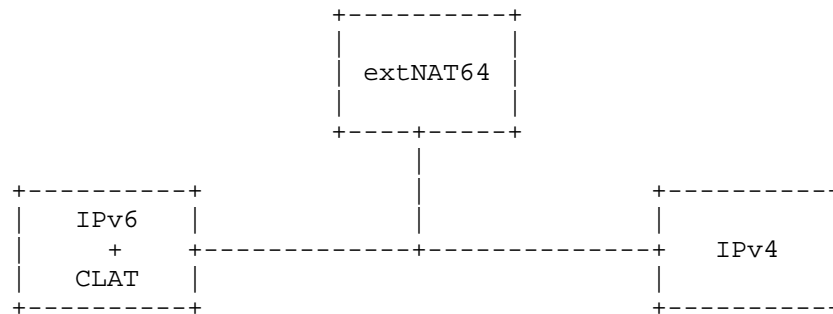


Figure 9: 464XLAT without DNS64; NAT64 in external provider

3.2. Known to Work Under Special Conditions

The scenarios in this category are known not to work unless significant effort is devoted to solve the issues, or are intended to solve problems across "closed" networks, instead of as a general Internet access usage. In addition to the different pros, cons and trade-offs, which may be acceptable for some operators, they have implementation difficulties, as they are beyond the original expectations of the NAT64/DNS64 original intent.

3.2.1. Service Provider NAT64 without DNS64

In this scenario, the service provider offers a NAT64, however there is no DNS64 function support.

As a consequence, an IPv6 host in the IPv6-only access network, will not be able to detect the presence of DNS64 by means of [RFC7050], neither learning the IPv6 prefix to be used for the NAT64.

This can be sorted out as indicated in Section 4.1.1.

However, despite that, because the lack of the DNS64 function, the IPv6 host will not be able to obtain AAAA synthesized records, so the NAT64 becomes useless.

An exception to this "useless" scenario will be manually configure mappings between the A records of each of the IPv4-only remote hosts and the corresponding AAAA records, with the WKP (Well-Known Prefix) or NSP (Network-Specific Prefix) used by the service provider NAT64, as if they were synthesized by a DNS64.

This mapping could be done by several means, typically at the authoritative DNS server, or at the service provider resolvers by means of DNS RPZ (Response Policy Zones). The latest, may have

implications in DNSSEC, if the zone is signed. Also, if the service provider is using a NSP, having the mapping at the authoritative server, will mean that may create troubles to other parties trying to use different NSP or the WKP, unless multiple DNS "views" are also being used at the authoritative servers.

Generally, the mappings alternative, will only make sense if a few set of IPv4-only remote hosts need to be accessed by a single network or reduced set of them, which support IPv6-only in the access, with some kind of mutual agreement for using this procedure, so it doesn't care if they become a trouble for other parties across Internet ("closed services").

In any case, this scenario doesn't solve the issue of literal addresses or non-IPv6 compliant APIs, neither it solves the problem of IPv4-only hosts within that IPv6-only access network.

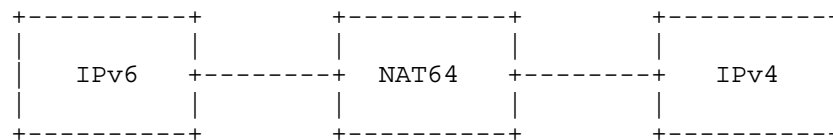


Figure 10: NAT64 without DNS64

3.2.2. Service Provider NAT64; DNS64 in the IPv6 hosts

In this scenario, the service provider offers the NAT64, but not the DNS64. However, the IPv6 hosts have a built-in DNS64 function.

This may become common if the DNS64 function is implemented in all the IPv6 hosts/stacks, which is not the actual situation. At this way, the DNSSEC validation is performed on the A record, and then the host can use the DNS64 function so to be able to use the NAT64, without any DNSSEC issues.

This scenario fails to solve the issue of literal addresses or non-IPv6 compliant APIs, unless the IPv6 hosts also supports Happy Eyeballs v2 ([RFC8305], Section 7.1), which may solve that issue.

However, this scenario still fails to solve the problem of IPv4-only hosts or applications inside the IPv6-only access network.



Figure 11: NAT64; DNS64 in IPv6 hosts

3.2.3. Service Provider NAT64; DNS64 in the IPv4-only remote network

In this scenario, the service provider offers the NAT64 only. The remote IPv4-only network offers the DNS64 function.

This is not common, and looks like doesn't make too much sense that a remote network, not deploying IPv6, is providing a DNS64 function and as in the case of the scenario depicted in Section 3.2.1, it will only work if both sides are using the WKP or the same NSP so, the same considerations apply. It can be also tuned to behave as in Section 3.1.1

This scenario still fails to solve the issue of literal addresses or non-IPv6 compliant APIs.

This scenario also fails to solve the problem of IPv4-only hosts or applications inside the IPv6-only access network.



Figure 12: NAT64; DNS64 in the IPv4-only

3.3. Comparing the Scenarios

This section compares the different scenarios, including the possible variations (each one represented in the precedent sections by a different Figure), looking at the following parameters:

- a. DNSSEC: Are there host validating DNSSEC?.
- b. Literal/APIs: Are there applications using literals or non-IPv6 compliant APIs?.
- c. IPv4-only: Are there hosts or applications using IPv4-only?.
- d. Foreign DNS: Is the Scenario surviving if the user change the

DNS?.

In the next table, the columns represent each of the scenario from the previous sections, by the Figure number. The possible values are:

- Scenario "bad" for that item.
- + Scenario "good" for that item.

Needs to be noted that in some cases "countermeasures", alternative or special configurations, may be available for the items designated as "bad", so this comparison is making a generic case, as a quick comparison guide. In some cases, a "bad" item is not necessarily a negative aspect, all it depends on the specific needs/characteristics of the network where the deployment will take place. For instance in a network which has only IPv6-only hosts and apps using only DNS and IPv6-compliant APIs, there is no impact using only NAT64 and DNS64, but if the hosts may validate DNSSEC, that item is still relevant.

Item / Figure	1	2	3	4	5	6	7	8	9	10	11	12
DNSSEC	-	-	-	-	-	-	-	+	+	+	+	+
Literal/APIs	-	-	-	-	+	+	+	+	+	-	-	-
IPv4-only	-	-	-	-	+	+	+	+	+	-	-	-
Foreign DNS	-	-	-	-	+	+	+	+	+	-	+	-

Figure 13: Scenario Comparision

As a general conclusion, we should note that if the network must support applications using literals, non-IPv6-compliant APIs, or IPv4-only hosts or applications, only the scenarios with 464XLAT will provide a solution. Further to that, those scenarios will also keep working if the user change the DNS setup. Clearly also, depending on if DNS64 is used or not, DNSSEC may be broken for those hosts doing DNSSEC validation.

4. Issues to be Considered

This section reviews the different issues that an operator needs to consider towards a NAT64/464XLAT deployment, as they may bring to decision points about how to approach that deployment.

4.1. DNSSEC Considerations and Possible Approaches

As indicated in Section 8 of [RFC6147] (DNS64, Security Considerations), because DNS64 modifies DNS answers and DNSSEC is designed to detect such modifications, DNS64 can break DNSSEC.

If a device connected to an IPv6-only WAN queries for a domain name in a signed zone, by means of a recursive name server that supports DNS64, and the result is a synthesized AAAA record, and the recursive name server is configured to perform DNSSEC validation and has a valid chain of trust to the zone in question, it will cryptographically validate the negative response from the authoritative name server. This is the expected DNS64 behavior: The recursive name server actually lies to the client device. However, in most of the cases, the client will not notice it, because generally they don't perform validation themselves and instead, rely on the recursive name servers.

A validating DNS64 resolver in fact, increase the confidence on the synthetic AAAA, as it has validated that a non-synthetic AAAA for sure, doesn't exist. However, if the client device is NAT64-oblivious (most common case) and performs DNSSEC validation on the AAAA record, it will fail as it is a synthesized record.

The best possible scenario from DNSSEC point of view is when the client requests the DNS64 server to perform the DNSSEC validation (by setting the DO bit to 1 and the CD bit to 0). In this case, the DNS64 server validates the data thus tampering may only happen inside the DNS64 server (which is considered as a trusted part, thus its likelihood is low) or between the DNS64 server and the client. All other parts of the system (including transmission and caching) are protected by DNSSEC ([Threat-DNS64]).

Similarly, if the client querying the recursive name server is another name server configured to use it as a forwarder, and is performing DNSSEC validation, it will also fail on any synthesized AAAA record.

All those considerations are extensively covered in Sections 3, 5.5 and 6.2 of [RFC6147].

The ideal solution to avoid DNSSEC issues, will be that all the signed zones also provide IPv6 connectivity, together with the corresponding AAAA records, which is out of the control of the operator needing to deploy NAT64.

An alternative solution, which was the one considered while developing [RFC6147], is that validators will be DNS64-aware, so

could perform the necessary discovery and do their own synthesis. That was done under the expectation that it was sufficiently early in the validator-deployment curve that it would be ok to break certain DNSSEC assumptions for networks who were really stuck in a NAT64/DNS64-needing world.

Previous data seems to indicate, that the figures of DNSSEC broken by using DNS64 will be around 1.7% ([About-DNS64]).

As already indicated, the scenarios in the previous section, are in fact somehow simplified, looking at the worst possible case (or saying it in a different way: "trying to look for the most perfect approach"), because breaking DNSSEC will not happen if the end-host is not doing validation, which is the case today in 1.7% of the cases. So a decision point for the operator must depend on "do I really care for that percentage of cases or can I provide alternative solutions for them?". Some possible solutions may be taken, as depicted in the next sections.

4.1.1. Not using DNS64

The ideal solution will be to avoid using DNS64, but as already indicated this is not possible in all the scenarios.

However, not having a DNS64, means that it is not possible to heuristically discover the NAT64 ([RFC7050]) and consequently, an IPv6 host in the IPv6-only access network, will not be able to detect the presence of the DNS64, neither to learn the IPv6 prefix to be used for the NAT64.

The learning of the IPv6 prefix could be solved by means of adding the relevant AAAA records to the `ipv4only.arpa.` zone of the service provider recursive servers, i.e., if using the WKP (`64:ff9b::/96`):

```
ipv4only.arpa. SOA      . . 0 0 0 0 0
ipv4only.arpa. NS       .
ipv4only.arpa. AAAA     64:ff9b::192.0.0.170
ipv4only.arpa. AAAA     64:ff9b::192.0.0.171
ipv4only.arpa. A        192.0.0.170
ipv4only.arpa. A        192.0.0.171
```

An alternative option to the above, is the use of DNS RPZ (Response Policy Zones).

One more alternative, only valid in environments with PCP support (for both the hosts or CEs and for the service provider network), to follow [RFC7225] (Discovering NAT64 IPv6 Prefixes using PCP).

Other alternatives may be available in the future, such as DHCPv6 options.

It may be convenient to support at the same time several of the approaches described, in order to ensure that clients with different ways to configure the NAT64 prefix, obtain it. This is also convenient even if DNS64 is being used.

4.1.2. DNSSEC validator aware of DNS64

In general, DNS servers with DNS64 function, by default, will not synthesize AAAA responses if the DNSSEC OK (DO) flag was set in the query. In this case, as only an A record is available, it means that the CLAT will take the responsibility, as in the case of literal IPv4 addresses, to keep that traffic flow end-to-end as IPv4, so DNSSEC is not broken. However, this will not work if a CLAT is not present as the hosts will not be able to use IPv4 (scenarios without 464XLAT).

4.1.3. Stub validator

If the DO flag is set and the client device performs DNSSEC validation, and the Checking Disabled (CD) flag is set for a query, as the DNS64 recursive server will not synthesize AAAA responses, the client could perform the DNSSEC validation with the A record and then may query the network for a NAT64 prefix ([RFC7050]) in order to synthesize the AAAA ([RFC6052]). This allows the client device to avoid using the CLAT and still use NAT64 even with DNSSEC.

If the end-host is IPv4-only, this will not work if a CLAT is not present (scenarios without 464XLAT).

Some devices/OSs may implement, instead of CLAT, a similar function by using Bump-in-the-Host ([RFC6535]), implemented as part of Happy Eyeballs v2 (Section 7.1 of [RFC8305]). In this case, the considerations in the above paragraphs are also applicable.

4.1.4. CLAT with DNS proxy and validator

If a CE includes CLAT support and also a DNS proxy, as indicated in Section 6.4 of [RFC6877], the CE could behave as a stub validator on behalf of the client devices, following the same approach described in the precedent section (Stub validator). So, the DNS proxy actually lie to the client devices, which in most of the cases will not notice it unless they perform validation themselves. Again, this allow the client devices to avoid using the CLAT and still use NAT64 with DNSSEC.

Once more, this will not work without a CLAT (scenarios without

464XLAT).

4.1.5. ACL of clients

In cases of dual-stack clients, stub resolvers should send the AAAA queries before the A ones. So, such clients, if DNS64 is enabled, will never get A records, even for IPv4-only servers, and they may be in the path before the NAT64 and accessible by IPv4. If DNSSEC is being used for all those flows, specific addresses or prefixes can be left-out the DNS64 synthesis by means of ACLs.

Once more, this will not work without a CLAT (scenarios without 464XLAT).

4.1.6. Mapping-out IPv4 addresses

If there are well-known specific IPv4 addresses or prefixes using DNSSEC, they can be mapped-out of the DNS64 synthesis.

Even if this is not related to DNSSEC, this "mapping-out" feature is actually, quite commonly used to ensure that [RFC1918] addresses (for example used by LAN servers) are not synthesized to AAAA.

Once more, this will not work without a CLAT (scenarios without 464XLAT).

4.2. DNS64 and Reverse Mapping

When a client device, using a name server configured to perform DNS64, tries to reverse-map a synthesized IPv6 address, the name server responds with a CNAME record pointing the domain name used to reverse-map the synthesized IPv6 address (the one under ip6.arpa), to the domain name corresponding to the embedded IPv4 address (under in-addr.arpa).

This is the expected behavior, so no issues to be considered regarding DNS reverse mapping.

4.3. Using 464XLAT with/without DNS64

In the case the client device is IPv6-only (either because the stack is IPv6-only, or because it is connected via an IPv6-only LAN) and the remote server is IPv4-only (either because the stack is IPv4-only, or because it is connected via an IPv4-only LAN), only NAT64 combined with DNS64 will be able to provide access among both. Because DNS64 is then required, DNSSEC validation will be only possible if the recursive name server is validating the negative response from the authoritative name server and the client is not

performing validation.

However, when the client device is dual-stack and/or connected in a dual-stack LAN by means of a CLAT (or has the built-in CLAT), DNS64 is an option.

1. With DNS64: If DNS64 is used, most of the IPv4 traffic (except if using literal IPv4 addresses or non-IPv6 compliant APIs) will not use the CLAT, so will use the IPv6 path and only one translation will be done at the NAT64. This may break DNSSEC, unless measures as described in the precedent sections are taken.
2. Without DNS64: If DNS64 is not used, all the IPv4 traffic will make use of the CLAT, so two translations are required (NAT46 at the CLAT and NAT64 at the PLAT), which adds some overhead in terms of the extra NAT46 translation, however avoids the AAAA synthesis and consequently will never break DNSSEC.

Note that the extra translation, when DNS64 is not used, takes place at the CLAT, which means no extra overhead for the operator, and no perceptible impact for a CE in a broadband network, while it may have some impact in a battery powered device. This cost for a battery powered device, is possibly comparable to the cost when the device is doing a local address synthesis (see Section 7.1 of [RFC8305]).

4.4. Manual Configuration of Foreign DNS

When clients, in a service provider network, use DNS servers from other networks, for example manually configured by users, they may support or not DNS64, so the considerations in Section 4.3 will apply as well.

Even in the case that the external DNS supports DNS64 function, we may be in the situation of providing incorrect configurations parameters, for example un-matching WKP or NSP, or a case such the one described in Section 3.2.3.

Having a CLAT and using an external DNS without DNS64, ensures that everything will work, so the CLAT must be considered as an advantage against user configuration errors.

However, it needs to be reinforced, that if there is not a CLAT (scenarios without 464XLAT), an external DNS without DNS64 support, will not only guarantee that DNSSEC is broken, but also disallow any access to IPv4-only networks, so will behave as in the Section 3.2.1.

4.5. Well-Known Prefix (WKP) vs Network-Specific Prefix (NSP)

[RFC6052] (IPv6 Addressing of IPv4/IPv6 Translators), Section 3, discusses some considerations which are useful to decide if an operator should use the WKP or an NSP.

Taking in consideration that discussion and other issues, we can summarize the possible decision points as:

- a. The WKP MUST NOT be used to represent non-global IPv4 addresses. If this is required, because the network to be translated use non-global addresses then an NSP is required.
- b. The WKP MAY appear in inter-domain routing tables, if the operator provides NAT64 to peers, however special considerations related to BGP filtering are then required and IPv4-embedded IPv6 prefixes longer than the WKP MUST NOT be advertised in BGP. An NSP may be a more appropriate option in those cases.
- c. If several NAT64s use the same prefix, packets from the same flow may be routed to different NAT64s in case of routing changes. This can be avoided either by using different prefixes for each NAT64, or by ensuring that all the NAT64s coordinate their state. Using an NSP could facilitate that.
- d. If DNS64 is required and users may change their DNS configuration, and deliberately choose an alternative DNS64, most probably alternative DNS64 will use by default the WKP. If an NSP is used by the NAT64, the users will not be able to use the operator NAT64.

4.6. IPv4 literals and old APIs

A hosts or application using literal IPv4 addresses or older APIs, behind a network with IPv6-only access, will not work unless a CLAT is present.

A possible alternative approach is described as part of Happy Eyeballs v2 Section 7.1 ([RFC8305]), or if not supporting HEv2, directly using Bump-in-the-Host ([RFC6535]), and then a DNS64 function.

Those alternatives will solve the problem for and end-hosts, however, if that end-hosts is providing "tethering" or an equivalent service to others hosts, that need to be considered as well. In other words, in a case of a cellular network, it resolves the issue for the UE itself, but may be not for hosts behind it.

Otherwise, 464XLAT is the only valid approach to resolve this issue.

4.7. IPv4-only Hosts or Applications

An IPv4-only hosts or application behind a network with IPv6-only access, will not work unless a CLAT is present. 464XLAT is the only valid approach to resolve this issue.

4.8. CLAT Translation Considerations

As described in Section 6.3 of [RFC6877] (IPv6 Prefix Handling), if the CLAT can be configured with a dedicated /64 prefix for the NAT46 translation, then it will be possible to do a more efficient stateless translation.

However, if this dedicated prefix is not available, the CLAT will need to do a stateful translation, for example performing stateful NAT44 for all the IPv4 LAN packets, so they appear as coming from a single IPv4 address, and then in turn, stateless translated to a single IPv6 address.

The obvious recommended setup, in order to maximize the CLAT performance, is to configure the dedicated translation prefix. This can be easily achieved automatically, if the broadband CE or end-user device is able to obtain a shorter prefix by means of DHCPv6-PD ([RFC3633]) so, the CE can use a /64 for that. This is also possible when broadband is provided by a cellular access.

The above recommendation is often not possible for cellular networks, when connecting smartphones (as UEs), as they don't use DHCPv6-PD ([RFC3633]) an instead a single /64 is provided for each PDP context and use /64 prefix sharing ([RFC6877]). So, in this case, the UEs typically have a build-in CLAT client, which is doing a stateful NAT44 before the stateless NAT46.

5. Summary of Deployment Recommendations for NAT64

It can be argued that none of the possible transition mechanisms is perfect, and somehow, we may consider that actually this is a good thing as a way to push for the IPv6 deployment, or otherwise, it may be further delayed, with clear undesirable effects for the global Internet.

However, for an operator, being in business means minimizing the adverse transition effects, and provide the most perfect one reasonably balanced with cost (CAPEX/OPEX), and at the same time looking for a valid long-term vision.

NAT64/464XLAT has demonstrated to be a valid choice in several scenarios, with hundreds of millions of users, offering different choices of deployment, depending on each network case, needs and requirements.

Depending on those requirements, DNS64 may be a required function, while in other cases the adverse effects may be counterproductive. Similarly, in some cases NAT64, together with DNS64, may be a valid solution, when for sure there is no need to support hosts or applications which are IPv4-only (Section 4.6, Section 4.7). However, in other cases the limitations they have, may suggest the operator to look into 464XLAT as a more complete solution.

Service providers willing to deploy NAT64, need to take into account the considerations of this document in order to better decide what is more appropriate for their own specific case.

In the case of broadband managed networks (CE provided or suggested/ supported by the operator), in order to fully support the actual user needs (IPv4-only devices and applications, usage of literals and old APIs), they SHOULD consider the 464XLAT scenario and in that case, MUST support the customer-side translator (CLAT).

If the operator offers DNS services, in order to increase performance by reducing the double translation for all the IPv4 traffic, they MAY support DNS64 and avoid, as much as possible, breaking DNSSEC. In this case, if the DNS service is offering DNSSEC validation, then it MUST be in such way that it is aware of the DNS64. This is considered de simpler and safer approach, and MAY be combined as well with the other possible solutions described in this document:

- o DNS infrastructure MUST be aware of DNS64 (Section 4.1.2).
- o Devices running CLAT SHOULD follow the indications in Section 4.1.3 (Stub validator). However, this may be out of the control of the operator.
- o CEs SHOULD include a DNS proxy and validator (Section 4.1.4).
- o Section 4.1.5 (ACL of clients) and Section 4.1.6 (Mapping-out IPv4 addresses) MAY be considered by each operator, depending on their own infrastructure.

This "increased performance" approach has the disadvantage of potentially breaking DNSSEC for a small percentage of validating end-hosts versus the small impact of a double translation taking place in the CE. If CE performance is not an issue, which is the most frequent case, then a much safer approach is to not use DNS64 at all,

and consequently ensure that all the IPv4 traffic is translated at the CLAT (Section 4.3).

If DNS64 is not used, at least one of the alternatives described in Section 4.1.1, MUST be followed.

The operator need to consider that if the user can modify the DNS configuration (which most probably is impossible to avoid), and instead of configuring a DNS64 choose an external regular DNS (non-DNS64), an scenario with only NAT64 will not work with any IPv4-only remote host, while it will continue working in the case of 464XLAT (Section 4.4).

Similar considerations need to be taken regarding the usage of a NAT64 Well-Known vs Network-Specific Prefix (Section 4.5), in the sense of, if using DNS64, they MUST match and if the user can change the DNS config, they will, most probably, not.

The ideal configuration for CEs supporting CLAT, is that they support DHCPv6-PD ([RFC3633]) and internally reserve one /64 for the stateless NAT46 translation. The operator MUST ensure that the customers get allocated prefixes shorter than /64 in order to support this optimization. One way or the other, this is not impacting the performance of the operator network.

As indicated in Section 7 of [RFC6877] (Deployment Considerations), operators MAY follow those suggestions in order to take advantage of traffic engineering.

In the case of cellular networks, the considerations regarding DNSSEC may appear as out-of-scope, because UEs OSs, commonly don't support DNSSEC, however applications running on them may do, or it may be an OS "built-in" support in the future. Moreover, if those devices offer tethering, other client devices may be doing the validation, hence the relevance of a proper DNSSEC support by the operator network.

Furthermore, cellular networks supporting 464XLAT ([RFC6877]) and "Discovery of the IPv6 Prefix Used for IPv6 Address Synthesis" ([RFC7050]), allow a progressive IPv6 deployment, with a single APN supporting all types of PDP context (IPv4, IPv6, IPv4v6), in such way that the network is able to automatically serve all the possible combinations of UEs.

One last consideration is that many networks may have different scenarios at the same time, for example, customers requiring 464XLAT, others not requiring it, customers requiring DNS64, others not, etc. In general, the different issues and approaches described in this

document can be implemented at the same time for different customers or parts of the network, so not representing any problem for complex cases.

Finally, if the operator chooses to secure the NAT64 prefix, it MUST follow the advice from Section 3.1.1. of [RFC7050] (Validation of Discovered Pref64::/n).

6. Deployment of NAT64 in Enterprise Networks

The recommendations of this document can be used as well in enterprise networks, campus and other similar scenarios, when the NAT64 (and/or DNS64) are under the control of that network, and for whatever reasons, there is a need to provide "IPv6-only access" to any part of that network or it is IPv6-only connected to third party networks.

An example of that is the IETF meetings network itself, where a NAT64 and DNS64 are provided, presenting in this case the same issues as per Section 3.1.1. If there is a CLAT in the IETF network, then there is no need to use DNS64 and it falls under the considerations of Section 3.1.3. Both scenarios have been tested and verified already in the IETF network itself.

Next figures are only meant to represent a few of the possible scenarios, not pretending to be the only ones that are feasible.

The following figure provides an example of and IPv6-only enterprise network connected with dual-stack to Internet and using local NAT64 and DNS64.

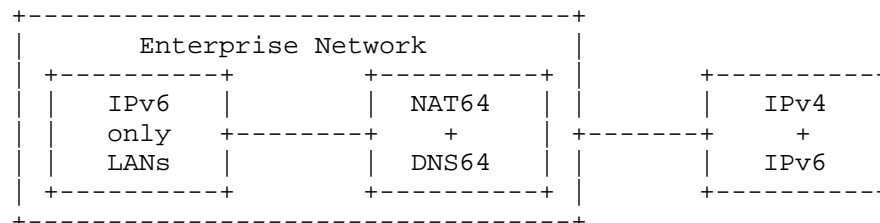


Figure 14: IPv6-only enterprise with NAT64 and DNS64

The following figure provides an example of dual-stack enterprise network connected with dual-stack to Internet and using CLAT without DNS64.

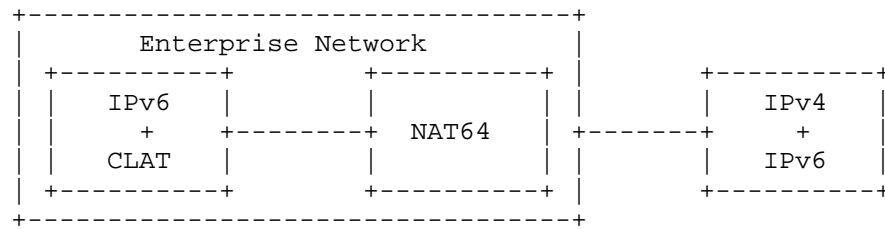


Figure 15: Dual-stack enterprise with CLAT without DNS64

Finally, the following figure provides an example of an IPv6-only provider with NAT64, and a dual-stack enterprise network by means of their own CLAT without DNS64.

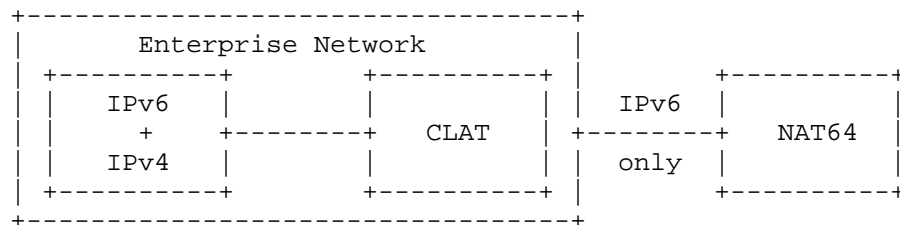


Figure 16: Dual-stack enterprise with CLAT without DNS64

7. Security Considerations

This document does not have any new specific security considerations.

8. IANA Considerations

This document does not have any new specific IANA considerations.

Note: This section is assuming that <https://www.rfc-editor.org/errata/eid5152> is resolved, otherwise, this section may include the required text to resolve the issue.

9. Acknowledgements

The author would like to acknowledge the inputs of Gabor Lencse, Andrew Sullivan, Lee Howard, Barbara Stark, Fred Baker and TBD ...

Conversations with Marcelo Bagnulo, one of the co-authors of NAT64 and DNS64, as well as several emails in mailing lists from Mark Andrews, have been very useful for this work.

Christian Huitema inspired working in this document by suggesting

that DNS64 should never be used, during a discussion regarding the deployment of CLAT in the IETF network.

10. ANNEX A: Example of Broadband Deployment with 464XLAT

This section summarizes how an operator may deploy an IPv6-only network for residential/SOHO customers, supporting IPv6 inbound connections, and IPv4-as-a-Service (IPv4aaS) by using 464XLAT.

Note that an equivalent setup could also be provided for enterprise customers. In case they need IPv4 inbound connections, several mechanisms, depending on specific customer needs, allow that.

Conceptually, most of the operator network could be IPv6-only (represented in the next pictures as "IPv6-only Internet"). This part of the network connects the IPv6-only subscribers (by means of IPv6-only access links), to the IPv6 upstream providers, as well as to the IPv4-Internet by means of the NAT64 (PLAT in the 464XLAT terminology).

The traffic flow from and back to the CE to services available in the IPv6 Internet (or even dual-stack remote services, when IPv6 is being used), is purely native IPv6 traffic, so no special considerations about it.

Looking at the picture from the DNS perspective, there are remote networks with are IPv4-only, and typically will have only IPv4 DNS (DNS/IPv4), or at least will be seen as that from the CE perspective. At the operator side, the DNS, as seen from the CE, is only IPv6 (DNS/IPv6) and has also a DNS64 function.

In the customer LANs side, there is actually one network, which of course could be split in different segments, and the most common setup will be those segments being dual-stack (global IPv6 addresses and [RFC1918] for IPv4, as usual in any regular residential/SOHO IPv4 network today). In the figure it is represented as tree segments, just to show that the three possible setups are valid (IPv6-only, IPv4-only and dual-stack).

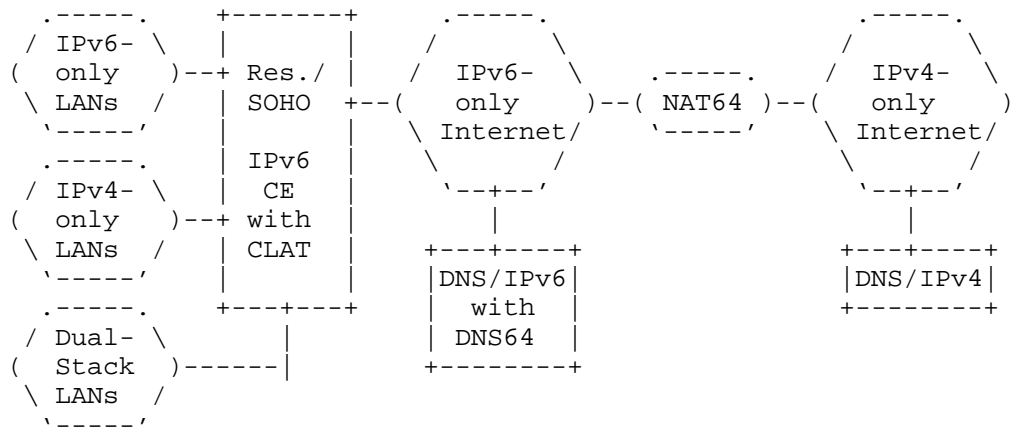


Figure 17: CE setup with built-in CLAT with DNS64

In addition to the regular CE setup, which will be typically access-technology dependent, the steps for the CLAT configuration can be summarized as:

1. Discovery of the PLAT (NAT64) prefix: It may be done using [RFC7050], or in those networks where PCP is supported, by means of [RFC7225], or other alternatives that may be available in the future (such as DHCPv6 options).
2. If the CLAT allows stateless NAT46 translation, a /64 from the pool typically provided to the CE by means of DHCPv6-PD [RFC3633], need to be set aside for that translation. Otherwise, the CLAT is forced to perform an intermediate stateful NAT44 before the a stateless NAT46, as described in Section 4.8.

The operator network need to ensure that the correct responses are provided for the discovery of the PLAT prefix, as well as it is highly recommended follows [RIPE-690], in order to ensure that multiple /64s are available including the one needed for the NAT46 translation.

The operator need to understand other issues, described across this document, in order to take the relevant decisions. For example, if several NAT64 are needed in the context of scalability/high-availability, an NSP should be considered (Section 4.5).

More complex scenarios are possible, for example, if a network offers multiple NAT64 prefixes, destination-based NAT64 prefixes, etc.

If the operator decides not to provide DNS64, then this setup turns

into the one in the following Figure. This will be also the setup that, if the user has changed the DNS and consequently is not using the operator DNS64, it will be seen from the perspective of the CE.

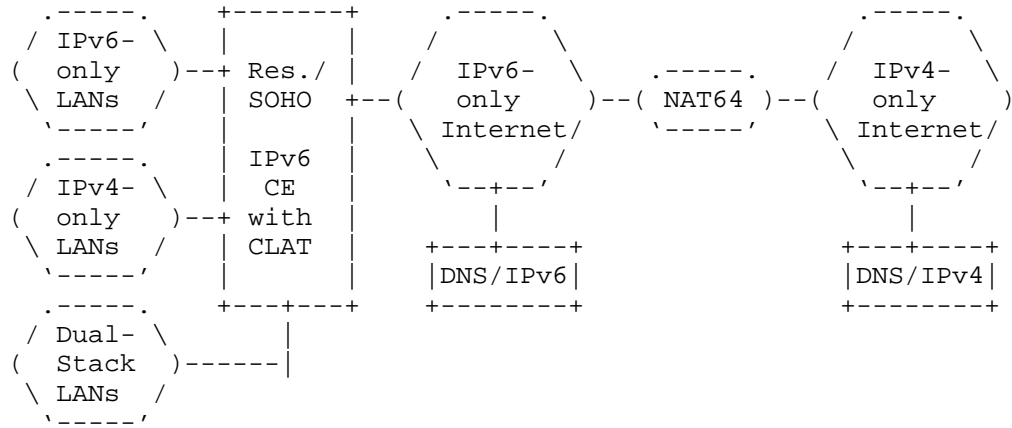


Figure 18: CE setup with built-in CLAT without DNS64

In this case the discovery of the PLAT prefix need to be arranged as indicated in Section 4.1.1.

In this case the CE doesn't have a built-in CLAT, or the customer can choose to setup the IPv6 operator-managed CE in bridge mode (and optionally use its own external router), or for example there is an access technology that requires some kind of media converter (ONT for FTTH, CableModem for DOCSIS, etc.), the complete setup will look as in the next figure. Obviously, there will be some intermediate configuration steps for the bridge, depending on the specific access technology/protocols, which should not modify the steps already described in the previous cases for the CLAT configuration.

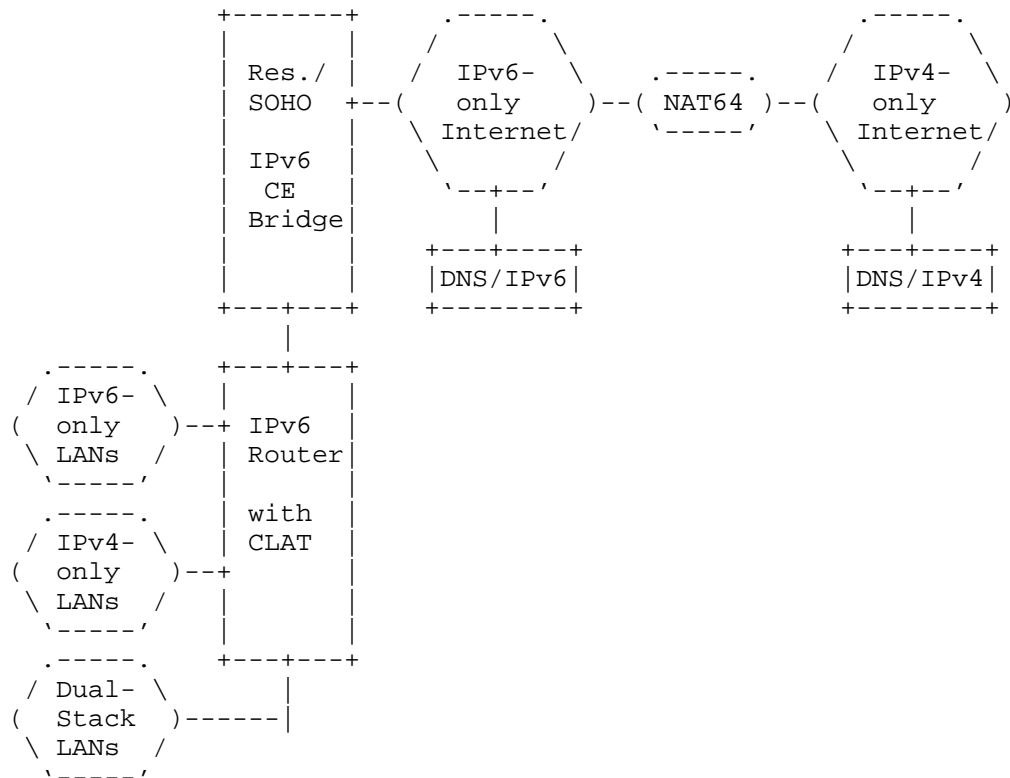


Figure 19: CE setup with bridged CLAT without DNS64

It should be avoided that several routers (i.e., the operator provided CE and a downstream user provided router) enable simultaneously routing and/or CLAT, in order to avoid multiple NAT44 and NAT46 levels, as well as ensuring the correct operation of multiple IPv6 subnets, so it is suggested to use HNCP ([RFC8375]).

Note that the procedure described here for the CE setup, can be simplified if the CE follows draft-ietf-v6ops-transition-ipv4aas ... TBD.

11. ANNEX B: CLAT Implementation

TBD.

A CLAT CE implementation basically requires support of [RFC7915] for the NAT46 functionality, [RFC7050] for the PLAT prefix discovery (and/or [RFC7225] for PCP), and if stateless NAT46 is supported, mechanisms to ensure that multiple /64 are available, such as

DHCPv6-PD [RFC3633].

There are several OpenSource implementations of CLAT, such as:

Android: https://github.com/ddrown/android_external_android-clat.

Linux: <https://github.com/toreanderson/clatd>.

OpenWRT: <https://github.com/openwrt-routing/packages/blob/master/nat46/files/464xlat.sh>.

VPP: <https://git.fd.io/vpp/tree/src/plugins/nat>.

12. References

12.1. Normative References

- [RFC1918] Rekhter, Y., Moskowitz, B., Karrenberg, D., de Groot, G., 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>>.
- [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>>.
- [RFC3633] Troan, O. and R. Droms, "IPv6 Prefix Options for Dynamic Host Configuration Protocol (DHCP) version 6", RFC 3633, DOI 10.17487/RFC3633, December 2003, <<https://www.rfc-editor.org/info/rfc3633>>.
- [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>>.
- [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>>.

- [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>>.
- [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>>.
- [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>>.
- [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>>.
- [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>>.
- [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>>.
- [RFC8375] Pfister, P. and T. Lemon, "Special-Use Domain 'home.arpa.'", RFC 8375, DOI 10.17487/RFC8375, May 2018, <<https://www.rfc-editor.org/info/rfc8375>>.

12.2. Informative References

- [About-DNS64] J. Linkova, "Let's talk about IPv6 DNS64 & DNSSEC", 2016, <<https://blog.apnic.net/2016/06/09/lets-talk-ipv6-dns64-dnssec/>>.

[RIPE-690]

RIPE, "Best Current Operational Practice for Operators: IPv6 prefix assignment for end-users - persistent vs non-persistent, and what size to choose", October 2017, <<https://www.ripe.net/publications/docs/ripe-690>>.

[Threat-DNS64]

G. Lencse and Y. Kadobayashi, "Methodology for the identification of potential security issues of different IPv6 transition technologies: Threat analysis of DNS64 and stateful NAT64", September 2018.

Author's Address

Jordi Palet Martinez
The IPv6 Company
Molino de la Navata, 75
La Navata - Galapagar, Madrid 28420
Spain

Email: jordi.palet@theipv6company.com
URI: <http://www.theipv6company.com/>

v6ops
Internet-Draft
Intended status: Informational
Expires: May 7, 2020

J. Palet Martinez
The IPv6 Company
November 4, 2019

IPv6 Point-to-Point Links
draft-palet-v6ops-p2p-links-04

Abstract

This document describes different alternatives for configuring IPv6 point-to-point links, considering the prefix size, numbering choices and prefix pool to be used.

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 May 7, 2020.

Copyright Notice

Copyright (c) 2019 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 Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	2
2. Requirements Language	3
3. The Ping-Pong Problem in Point-to-Point Links	3
4. Prefix Size Choices	3
4.1. Rationale for using /64	3
4.2. Rationale for using /127	4
4.3. Rationale for using /126 and Other Options	5
4.4. A Possible Middle-Term Choice	5
5. Numbering Choices	5
5.1. GUA (Global Unicast Addresses)	5
5.2. ULA (Unique Local Addresses)	5
5.3. Link-Local Addresses Only	6
6. Prefix Pool Choices	7
7. /64 from Customer Prefix for point-to-point links	7
7.1. Numbering Interfaces	7
7.2. Routing Aggregation of the Point-to-Point Links	8
7.3. DHCPv6 Considerations	9
7.4. Router Considerations	9
8. Security Considerations	10
9. IANA Considerations	10
10. Acknowledgements	10
11. References	10
11.1. Normative References	10
11.2. Informative References	11
Author's Address	12

1. Introduction

There are different alternatives for numbering IPv6 point-to-point links, and from an operational perspective, there may have different advantages or disadvantages that need to be taken in consideration under the scope of each specific network architecture design.

[RFC6164] describes using /127 prefixes for inter-router point-to-point links, using two different address pools, one for numbering the point-to-point links and another one for delegating the prefixes at the end of the point-to-point link. However, this doesn't exclude other choices.

This document describes alternative approaches, for the prefix size, the numbering of the link and the prefix pool.

The proposed approaches are suitable for those point-to-point links connecting ISP to customers, but not limited to those cases, and in fact, all them are being used by a relevant number of networks worldwide, in several different scenarios (service providers,

enterprise networks, etc.).

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. The Ping-Pong Problem in Point-to-Point Links

Some point-to-point links may present the ping-pong problem, (a forwarding loop). The fundamental root cause of this problem is an IPv6 implementations not performing full Neighbor Discovery (NS/NA) on addresses that the prefix says could exist on the link.

IPv6 implementations are assuming that all addresses within the prefix must exist at the other end of the point-to-point link, and send the traffic straight onto the link. If the address doesn't exist, and there is a covering route back in the other direction, the ping-pong problem occurs.

Full Neighbor Discovery is doing more than just resolving the link-layer address of an IPv6 address. Neighbor Discovery is also determining if the address exists. Even if a point-to-point link doesn't have link-layer addresses to resolve, ND determining if an address exists on the link is very beneficial because it will prevent the ping-pong problem occurring entirely regardless of the IPv6 prefix length being used on the link.

4. Prefix Size Choices

[RFC7608] already discusses about the IPv6 prefix length recommendations for forwarding, and the need for routing and forwarding implementations to ensure that longest-prefix-match works on any prefix length. So, in this document, we concentrate in the most commonly used choices, not excluding other options.

4.1. Rationale for using /64

The IPv6 Addressing Architecture ([RFC4291]) specifies that all the Interface Identifiers for all the unicast addresses (except for 000/3) are required to be 64 bits long and to be constructed in Modified EUI-64 format.

The same document also mandates the usage of the predefined subnet-router anycast address, which has cleared to zero all the bits that

do not form the subnet prefix.

Using /64 is the most common scenario and currently the best practice by the number of service providers using this approach compared to others.

Using a /64 has the advantage of being future proof and avoids renumbering, in the event that new standards take advantage of the 64 bits for other purposes, or the link becomes a point-to-multipoint, or there is a need to use more addresses in the link (e.g., monitoring equipment, managed bridges).

It has been raised also the issue of some hardware having limitations in using prefixes longer than /64, for example using extra hardware resources.

Section 5. of [RFC6164] describes possible issues when using /64 for the point-to-point links, such as the ping-pong and the neighbor cache exhaustion. However, it also states that they can be mitigated by other means, including the latest ICMPv6 [RFC4443] ND [RFC4861]. Indeed, considering the publication date of that document, those issues should not be any longer a concern. The fact is that many operators worldwide, today use /64 without any concerns, as vendors have taken the necessary code updates.

Consequently, we shall conclude that it is a valid approach to use /64 prefixes for the point-to-point links.

4.2. Rationale for using /127

[RFC6164] already do a complete review of reasons why /127 is a good approach vs other options. However, it needs to be considered that it was published a number of years ago, and most of the hardware today already incorporate mitigations.

It should be noted that, when using a /127 prefix, configuration of each of the addresses within the /127 prefix, at each respective end of the link, must be actively validated by the network operator. A missing /127 address from one end of the link, with a local route pointing out that end of the link that covers the missing /127 address, such as a default route, causes a "ping-pong" scenario to exist for the missing /127 address. The link could still be successfully carrying transit traffic, and IPv6 will not report any errors, because IPv6 doesn't require or nor check to ensure all interfaces attached to a link has addresses from all prefixes assigned to the link, excepting the Link-Local prefix per [RFC4291].

It is a valid approach to use /127 for the point-to-point links,

however is not future proof considering the comments from the previous section, and older equipment may not support it.

4.3. Rationale for using /126 and Other Options

/126 was considered by [RFC3627], and despite this document has been obsoleted, because was considering /127 as harmful, the considerations in Section 4.3 are still valid.

The same document describes options such as /112 and /120, and all those are commonly used in worldwide IPv6 deployments [IPv6-Survey], though in a lesser degree than /64 or /127.

Consequently, we shall conclude that /126, /120 and /112 are valid approaches for the point-to-point links.

4.4. A Possible Middle-Term Choice

A possible "middle-term" approach, will be to allocate a /64 for each point-to-point link, but use just one /127 out of it, making it future proof and at the same time avoiding possible issues indicated in the previous sections.

5. Numbering Choices

IPv6 provides different unicast addressing scopes which can be considered when numbering a point-to-point link.

It has been reported that certain hardware may consume resources when using numbered links. This is a very specific situation that may need to be consider on a case by case basis.

5.1. GUA (Global Unicast Addresses)

Using GUA is the most common approach. It provides full functionality for both end-points of the point-to-point link and consequently, facilitates troubleshooting.

5.2. ULA (Unique Local Addresses)

Some networks use ULAs for numbering the point-to-point links. This approach may cause numerous problems when carrying Internet traffic and therefore, is strongly discouraged. For example, if the CE needs to send an ICMPv6 message to a host outside that network (to the Internet), the packet with ULA source address will not get thru and PMTUD will break, which in turn will completely break that IPv6 connection when the MTU is not the same for all the path.

ULAs are IPv6 private addresses, not intended to be used as source or destination addresses across the Internet. This issue also exists in IPv4 when using [RFC1918] addresses on links carrying IPv4 Internet traffic. [RFC6752] discusses this issue for IPv4, with much of the discussion applying similarly to IPv6 and ULAs.

However, this approach is valid if, following Section 2.2 of [RFC4443], and despite using ULA for the point-to-point link, the router is configured with at least one GUA and the source of the ICMPv6 messages are always a GUA, per the IPv6 Default Address Selection algorithm [RFC6724].

5.3. Link-Local Addresses Only

Some networks leave the point-to-point links with only Link-Local addresses used at both ends of the link. This is sometimes improperly referred as "unnumbered", because the Link-Local addresses are also "numbers". Furthermore, [RFC4291] requires that all interfaces attached to a link have at least a Link-Local "number" or address from the Link-Local prefix.

[RFC7404] (Using Only Link-Local Addressing inside an IPv6 Network) discusses pros and cons of this alternative, which in general apply for the point-to-point links.

While this choice might work if the point-to-point link is terminated in a router, which typically will get configured with a suitable routable GUA or ULA, it will not work for devices that can't be further configured, for example if they do not support DHCPv6-PD. This is the case for hosts, when the Operating System is not expected to be a DHCPv6-PD client and are therefore left without any usable GUA to allow traffic forwarding.

In the case of a router, the route for the assigned prefix is pointed towards the link-local address on the router WAN port and the default route on the router is pointed towards the link-local address on the upstream network equipment port.

This choice seems easier to implement, compared the previous ones, but it also brings some drawbacks, such as difficulties with troubleshooting and monitoring. For example, link local addresses do not appear in traceroute, so it makes more difficult to locate the exact point of failure.

It is more useful in scenarios where it is known that only a router will be attached to the point-to-point link, and where the configured address of the router is known. Non-routers connecting to a network, which can't initiate DHCPv6-PD might experience problems and will

stay unnumbered upon connection, if a /64 prefix is not used to number the link. This may be also the case for routers, which will not be able to complete the DHCPv6-PD in unnumbered links.

The considerations indicated in the previous section, regarding not using ULA as source address of ICMPv6 messages, and instead ensuring there is at least one GUA configured for that, also apply if link-locals are used for the point-to-point link.

6. Prefix Pool Choices

The logic choice seems to use a dedicated pool of IPv6 addresses, as this is the way we are "used to" with IPv4. Actually, this is done often by means of different IPv6 pools at every PoP in a service provider network.

A possible benefit of using a dedicated IPv6 pool, is that allows applying security policies without harming the customers. This is only true if customers always have a CE at their end of the WAN link.

However, the fact that the default IPv6 link size is /64 and commonly multiple /64's are assigned to a single customer, provides an interesting alternative approach for combining "best practices" described in the precedent sections.

The following section depicts this alternative.

7. /64 from Customer Prefix for point-to-point links

Using a /64 from the customer prefix, in addition to the advantages already indicated when using /64, simplifies the addressing plan.

The use of /64 also facilitates an easier way for routing the shorter aggregated prefix into the point-to-point link. Consequently it simplifies the "view" of a more unified addressing plan, providing an easier path for following up any issue when operating IPv6 networks and typically, will have a great impact in saving expensive hardware resources (lower usage of TCAM, typically by half).

This mechanism would not work in broadcast layer two media that rely on ND, because it will try ND for all the addresses within the shorter prefix that is being routed thru the point-to-point link.

7.1. Numbering Interfaces

Often, in point-to-point links, hardware tokens are not available, or there is the need to keep certain bits (u, g) cleared, so the links can be manually numbered sequentially with most of the bits cleared

to zero. This numbering makes as well easier to remember the interfaces, which typically will become numbered as 0 (with 63 leading zero bits) for the provider side and 1 (with 63 leading zero bits) for the customer side.

Using interface identifiers as 0 and 1 is not only a very simple approach, but also a very common practice. Other different choices can as well be used as required in each case.

On the other hand, using the EUI-64, makes it more difficult to remember and handle the interfaces, but provides an additional degree of protection against port (actually address) scanning as described at [RFC7707].

7.2. Routing Aggregation of the Point-to-Point Links

Following this approach and assuming that a shorter prefix is typically delegated to a customer, for example a /48, it is possible to simplify the routing aggregation of the point-to-point links. Towards this, the point-to-point link may be numbered using the first /64 of the /48 delegated to the customer.

Let's see a practical example:

- o A service provider uses the prefix 2001:db8::/32 and is using 2001:db8:aaaa::/48 for a given customer.
- o Instead of allocating the point-to-point link from a different addressing pool, it may use 2001:db8:aaaa::/64 (which is the first /64 subnet from the 2001:db8:aaaa::/48) to number the link.
- o This means that, in the case the non-EUI-64 approach is used, the point-to-point link may be numbered as 2001:db8:aaaa::1/64 for the provider side and 2001:db8:aaaa::2/64 for the customer side.
- o Note that using the first /64 and interface identifiers 1 and 2 is a very common practice. However other values may be chosen according to each case specific needs.

In this way, as the same address pool is being used for both, the prefix and the point-to-point link, one of the advantages of this approach is to make very easy the recognition of the point-to-point link that belongs to a given customer prefix, or in the other way around, the recognition of the prefix that is linked by a given point-to-point link.

For example, making a trace-route to debug any issue to a given address in the provider network, will show a straight view, and it

becomes unnecessary one extra step to check a database that correlate an address pool for the point-to-point links and the customer prefixes, as all they are the same.

Moreover, it is possible to use the shorter prefix as the provider side numbering for the point-to-point link and keep the /64 for the customer side. In our example, it will become:

- o Point-to-point link at provider side: 2001:db8:aaaa::1/48
- o Point-to-point link at customer side: 2001:db8:aaaa::2/64

This provides one additional advantage as in some platforms the configuration may be easier saving one step for the route of the delegated prefix (no need for two routes to be configured, one for the delegated prefix, one for the point-to-point link). It is possible because the longest-prefix-match rule.

The behavior of this type of configuration has been successfully deployed in different operator and enterprise networks, using commonly available implementations with different routing protocols, including RIP, BGP, IS-IS, OSPF, along static routing, and no failures or interoperability issues have been reported.

7.3. DHCPv6 Considerations

As stated in [RFC3633], "the requesting router MUST NOT assign any delegated prefixes or subnets from the delegated prefix(es) to the link through which is received the DHCP message from the delegating router", however the approach described in this document is still useful in other DHCPv6 scenarios or non-DHCPv6 scenarios.

Furthermore, [RFC3633] was updated by Prefix Exclude Option for DHCPv6-based Prefix Delegation ([RFC6603]), precisely to define a new DHCPv6 option, which covers the case described by this document.

Moreover, [RFC3769] has no explicit requirement that avoids the approach described in this document.

7.4. Router Considerations

This approach is being used by operators in both, residential/SOHO and enterprise networks, so the routers at the customer end for those networks MUST support [RFC6603] if DHCPv6-PD is used.

In the case of Customer Edge Routers there is a specific requirement ([RFC7084]) WPD-8 (Prefix delegation Requirements), marked as SHOULD for [RFC6603]. However, in a scenario where the approach described

in this document is followed, together with DHCPv6-PD, the CE Router MUST support [RFC6603].

8. Security Considerations

This document does not have any new specific security considerations.

9. IANA Considerations

This document does not have any new specific IANA considerations.

10. Acknowledgements

The author would like to acknowledge the inputs of Mikael Abrahamsson, Brian Carpenter, Eric Vyncke, Mark Smith and TBD.

Acknowledge is also due to my co-authors of RIPE-690 (Best Current Operational Practice for Operators: IPv6 prefix assignment for end-users - persistent vs non-persistent, and what size to choose, <https://www.ripe.net/publications/docs/ripe-690>) and global community, which provided valuable inputs which have been key for this document.

Acknowledgement to co-authors, Cesar Olvera and Miguel Angel Diaz, of a previous related document (draft-palet-v6ops-point2point, 2006), as well as inputs for that version from Alain Durand, Chip Popoviciu, Daniel Roesen, Fred Baker, Gert Doering, Olaf Bonness, Ole Troan, Pekka Savola and Vincent Jardin, are also granted.

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>>.
- [RFC3633] Troan, O. and R. Droms, "IPv6 Prefix Options for Dynamic Host Configuration Protocol (DHCP) version 6", RFC 3633, DOI 10.17487/RFC3633, December 2003, <<https://www.rfc-editor.org/info/rfc3633>>.
- [RFC3769] Miyakawa, S. and R. Droms, "Requirements for IPv6 Prefix Delegation", RFC 3769, DOI 10.17487/RFC3769, June 2004, <<https://www.rfc-editor.org/info/rfc3769>>.

- [RFC4291] Hinden, R. and S. Deering, "IP Version 6 Addressing Architecture", RFC 4291, DOI 10.17487/RFC4291, February 2006, <<https://www.rfc-editor.org/info/rfc4291>>.
- [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>>.
- [RFC6603] Korhonen, J., Ed., Savolainen, T., Krishnan, S., and O. Troan, "Prefix Exclude Option for DHCPv6-based Prefix Delegation", RFC 6603, DOI 10.17487/RFC6603, May 2012, <<https://www.rfc-editor.org/info/rfc6603>>.
- [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>>.
- [RFC7084] Singh, H., Beebe, W., Donley, C., and B. Stark, "Basic Requirements for IPv6 Customer Edge Routers", RFC 7084, DOI 10.17487/RFC7084, November 2013, <<https://www.rfc-editor.org/info/rfc7084>>.
- [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>>.

11.2. Informative References

- [IPv6-Survey] Palet Martinez, J., "IPv6 Deployment Survey (Residential/Household Services)", January 2018, <<https://indico.uknol.org.uk/event/41/contribution/5/material/slides/0.pdf>>.
- [RFC1918] Rekhter, Y., Moskowitz, B., Karrenberg, D., de Groot, G., 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>>.

- [RFC3627] Savola, P., "Use of /127 Prefix Length Between Routers Considered Harmful", RFC 3627, DOI 10.17487/RFC3627, September 2003, <<https://www.rfc-editor.org/info/rfc3627>>.
- [RFC6164] Kohno, M., Nitzan, B., Bush, R., Matsuzaki, Y., Colitti, L., and T. Narten, "Using 127-Bit IPv6 Prefixes on Inter-Router Links", RFC 6164, DOI 10.17487/RFC6164, April 2011, <<https://www.rfc-editor.org/info/rfc6164>>.
- [RFC6752] Kirkham, A., "Issues with Private IP Addressing in the Internet", RFC 6752, DOI 10.17487/RFC6752, September 2012, <<https://www.rfc-editor.org/info/rfc6752>>.
- [RFC7404] Behringer, M. and E. Vyncke, "Using Only Link-Local Addressing inside an IPv6 Network", RFC 7404, DOI 10.17487/RFC7404, November 2014, <<https://www.rfc-editor.org/info/rfc7404>>.
- [RFC7608] Boucadair, M., Petrescu, A., and F. Baker, "IPv6 Prefix Length Recommendation for Forwarding", BCP 198, RFC 7608, DOI 10.17487/RFC7608, July 2015, <<https://www.rfc-editor.org/info/rfc7608>>.
- [RFC7707] Gont, F. and T. Chown, "Network Reconnaissance in IPv6 Networks", RFC 7707, DOI 10.17487/RFC7707, March 2016, <<https://www.rfc-editor.org/info/rfc7707>>.

Author's Address

Jordi Palet Martinez
The IPv6 Company
Molino de la Navata, 75
La Navata - Galapagar, Madrid 28420
Spain

EMail: jordi.palet@theipv6company.com
URI: <http://www.theipv6company.com/>

IPv6 Operations (v6ops)
Internet-Draft
Intended status: Informational
Expires: September 3, 2018

J. Palet Martinez
The IPv6 Company
H. M.-H. Liu
D-Link Systems, Inc.
March 2, 2018

Transition Requirements for IPv6 Customer Edge Routers to support IPv4
as a Service
draft-palet-v6ops-transition-ipv4aas-00

Abstract

This document specifies the transition requirements for an IPv6 Customer Edge (CE) router, either provided by the service provider or thru the retail market.

Specifically, this document extends the "Basic Requirements for IPv6 Customer Edge Routers" ([RFC7084]) in order to allow the provisioning of IPv6 transition services for the support of IPv4 as a Service (IPv4aaS) by means of new transition mechanisms, which were not available at the time [RFC7084] was published. The document only covers transition technologies for delivering IPv4 in IPv6-only access networks, commonly called IPv4 "as-a-service" (IPv4aaS), as required in a world where IPv4 addresses are no longer available, so hosts in the customer LANs with IPv4-only or IPv6-only applications or devices, requiring to communicate with IPv4-only services at the Internet, are still able to do so.

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 September 3, 2018.

Copyright Notice

Copyright (c) 2018 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 Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	2
1.1. Requirements Language	3
2. Terminology	4
3. Usage Scenarios	4
4. End-User Network Architecture	6
5. Requirements	9
5.1. General Requirements	9
5.2. LAN-Side Configuration	9
5.3. Transition Technologies Support for IPv4 Service Continuity (IPv4 as a Service - IPv4aaS)	9
5.3.1. 464XLAT	10
5.3.2. Lightweight 4over6 (lw4o6)	10
5.3.3. MAP-E	11
5.3.4. MAP-T	11
6. IPv4 Multicast Support	12
7. Code Considerations	12
8. Security Considerations	12
9. Acknowledgements	12
10. References	13
10.1. Normative References	13
10.2. Informative References	15
Authors' Addresses	15

1. Introduction

This document defines basic IPv6 transition features for a residential or small-office router, referred to as an "IPv6 Transition CE router with IPv4aaS support", in order to establish an industry baseline for transition features to be implemented on such a router.

These routers are based on "Basic Requirements for IPv6 Customer Edge Routers" ([RFC7084]), so the scope of this documents is to ensure the IPv4 "service continuity" support, in the LAN side and the access to IPv4-only Internet services from an IPv6-only access WAN even from IPv6-only applications or devices in the LAN side.

This document covers the IP transition technologies required when ISPs have already an IPv6-only access network, which is becoming a common situation in a world where IPv4 addresses are no longer available, so the service providers need to provision IPv6-only WAN access, while at the same time ensuring that both IPv4-only and IPv6-only devices or applications in the customer LANs, can still reach IPv4-only devices or applications in Internet, which still don't have IPv6 support.

This document specifies the transition mechanisms to be supported by an IPv6 transition CE router, and relevant provisioning or configuration information differences from [RFC7084].

This document is not a recommendation for service providers to use any specific transition mechanism.

Automatic provisioning of more complex topology than a single router with multiple LAN interfaces may be handled by means of HNCP ([RFC7788]), which is out of the scope of this document.

The CE vendors need to consider that the situation of lack of IPv4 addresses and the IPv6 deployment, is a global issue, so the CEs fulfilling the requirements of this document aren't only those provided by the service providers to the customers, but also the customers may need to replace existing ones by themselves thru the retail market.

1.1. Requirements Language

Take careful note: Unlike other IETF documents, the key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are not used as described in RFC 2119 [RFC2119]. This document uses these keywords not strictly for the purpose of interoperability, but rather for the purpose of establishing industry-common baseline functionality. As such, the document points to several other specifications (preferable in RFC or stable form) to provide additional guidance to implementers regarding any protocol implementation required to produce a successful IPv6 Transition CE router that interoperates successfully with a particular subset of currently deploying and planned common IPv6 access networks.

2. Terminology

This document uses the same terminology as in [RFC7084], with two minor clarifications.

The term "IPv6 transition Customer Edge Router with IPv4aaS" (shortened as "IPv6 transition CE") is defined as an "IPv6 Customer Edge Router" that provides transition support to allow IPv4-IPv6 coexistence either beyond the WAN, in the LAN or both.

The "WAN Interface" term used across this document, means that can also support link technologies based in Internet-layer (or higher-layers) "tunnels", such as IPv4-in-IPv6 tunnels.

3. Usage Scenarios

The situation before described, where there is ongoing IPv6 deployment and lack of IPv4 addresses, is not happening at the same pace at every country, and even within every country, every ISP. For different technical, financial, commercial/marketing and socio-economical reasons, each network is transitioning at their own pace, and nobody has a magic crystal ball, to make a guess.

Different studies also show that this is a changing situation, because in a single country, may be not all operators provide IPv6 support, and customer churn implies that the same customers at some point may have IPv6 service and may not have it, if changing ISP, and viceversa.

So it is clear that, to cover all those evolving situations, it is required an IPv6 transition CE which, at least from the perspective of the transition support, can keep accommodating to those changes, as it may be or not provided by the service provider. Even may be a point when, having one working seamlessly among different operators means lower cost for changing them, and so, increase and facilitate competition.

Moreover, because some services will remain as IPv4-only for an undetermined time and some service providers may also delay their IPv6 support, again for an undetermined period of time, there is an uncertainty about how much time there will be a percentage of IPv4 traffic between end-users and end-services, that definitively needs to be "serviced", so there will be a need to provide CEs with support "IPv4 as a Service" for some time.

This document is consequently, based on those premises, in order to ensure the continued transition from networks that today may provide access with dual-stack or IPv6-in-IPv4, as described in [RFC7084],

and as an "extension" to it, evolving to an IPv6-only access with IPv4-as-a-Service.

Considering that situation and different possible usage cases, the IPv6 Transition CE router described in this document is expected to be used typically, in any of the following scenarios:

1. Residential/household users. Common usage is any kind of Internet access (web, email, streaming, online gaming, etc.).
2. Residential with Small Office/Home Office (SOHO). Same usage as for the first scenario.
3. Small Office/Home Office (SOHO). Same usage as for the first scenario.
4. Small and Medium Enterprise (SME). Same usage as for the first scenario.
5. Residential/household with advanced requirements. Same basic usage as for the first scenario, however there may be requirements for exporting services to the WAN (IP cameras, web, DNS, email, VPN, etc.).
6. Small and Medium Enterprise (SME) with advanced requirements. Same basic usage as for the first scenario, however there may be requirements for exporting services to the WAN (IP cameras, web, DNS, email, VPN, etc.).

The above list is not intended to be comprehensive of all the possible usage scenarios, just the main ones. In fact, combinations of the above usages are also possible, for example a residential with SOHO and advanced requirements, as well as situations where the same CE is used at different times in different scenarios or even different services providers that may use a different transition mechanism.

The mechanisms for exporting IPv6 services are commonly "naturally" available in any IPv6 router, as when using GUA, unless they are blocked by firewall rules, which may require some manual configuration by means of a GUI and/or CLI.

However, in the case of IPv4, because the usage of private addresses and NAT, it typically requires some degree of manual configuration such as setting up a DMZ, virtual servers, or port/protocol forwarding. In general, CE routers already provide GUI and/or CLI to manually configure them, or the possibility to setup the CE in bridge mode, so another CE behind it, takes care of that. It is out of the

scope of this document the definition of any requirements for that.

The main difference for an IPv6 Transition CE router to support one or several of the above indicated scenarios, is related to the packet processing capabilities, performance, even other details such as the number of WAN/LAN interfaces, their maximum speed, memory for keeping tables or tracking connections, etc. So, it is out of the scope of this document to classify them.

For example, an SME may have just 10 employees (micro-SME), which commonly will be considered same as a SOHO, but a small SME can have up to 50 employees, or 250 for a medium one. Depending on the IPv6 Transition CE router capabilities or even how it is being configured (for instance, using SLAAC or DHCPv6), it may support even a higher number of employees if the traffic in the LANs is low, or switched by another device(s), or the WAN bandwidth requirements are low, etc. The actual bandwidth capabilities of access with technologies such as FTTH, cable and even 3GPP/LTE, allows the support of such usages, and indeed, is a very common situation that access networks and the IPv6 Transition CE provided by the service provider are the same for SMEs and residential users.

There is also no difference in terms of who actually provides the IPv6 Transition CE router. In most of the cases is the service provider, and in fact is responsible, typically, of provisioning/managing at least the WAN side. However, commonly the user has access to configure the LAN interfaces, firewall, DMZ, and many other aspects. In fact, in many cases, the user must supply, or at least can replace the IPv6 Transition CE router, which makes even more relevant that all the IPv6 Transition CE routers, support the same requirements defined in this document, despite if they are provided directly by the service provider or acquired thru the retail market.

The IPv6 Transition CE router described in this document is not intended for usage in other scenarios such as bigger Enterprises, Data Centers, Content Providers, etc. So, even if the documented requirements meet their needs, may have additional requirements, which are out of the scope of this document.

4. End-User Network Architecture

According to the descriptions in the precedent sections, an end-user network will likely support both IPv4 and IPv6. It is not expected that an end user will change their existing network topology with the introduction of IPv6. There are some differences in how IPv6 works and is provisioned; these differences have implications for the network architecture.

A typical IPv4 end-user network consists of a "plug and play" router with NAT functionality and a single link behind it, connected to the service provider network.

From the perspective of an "IPv4 user" behind an IPv6 transition Customer Edge Router with IPv4aaS, this doesn't change.

However, while a typical IPv4 NAT deployment by default blocks all incoming connections and may allow opening of ports using a Universal Plug and Play Internet Gateway Device (UPnP IGD) [UPnP-IGD] or some other firewall control protocol, in the case of an IPv6-only access, the latest may not be feasible depending on specific transition mechanism details. PCP (Port Control Protocol, [RFC6887]) may be an alternative solution, as well.

Another consequence of using IPv4 private address space in the end-user network is that it provides stable addressing; that is, it never changes even when you change service providers, and the addresses are always there even when the WAN interface is down or the customer edge router has not yet been provisioned. In the case of an IPv6-only access, there is no change on that if the transition mechanism keeps running the NAT interface towards the LAN side.

Many existing routers support dynamic routing (which learns routes from other routers), and advanced end-users can build arbitrary, complex networks using manual configuration of address prefixes combined with a dynamic routing protocol. Once again, this is true for both, IPv4 and IPv6.

In general, the end-user network architecture for IPv6 should provide equivalent or better capabilities and functionality than the current IPv4 architecture.

The end-user network is a stub network, in the sense that is not providing transit to other external networks. However HNCP ([RFC7788]) allows support for automatic provisioning of downstream routers. Figure 1 illustrates the model topology for the end-user network.

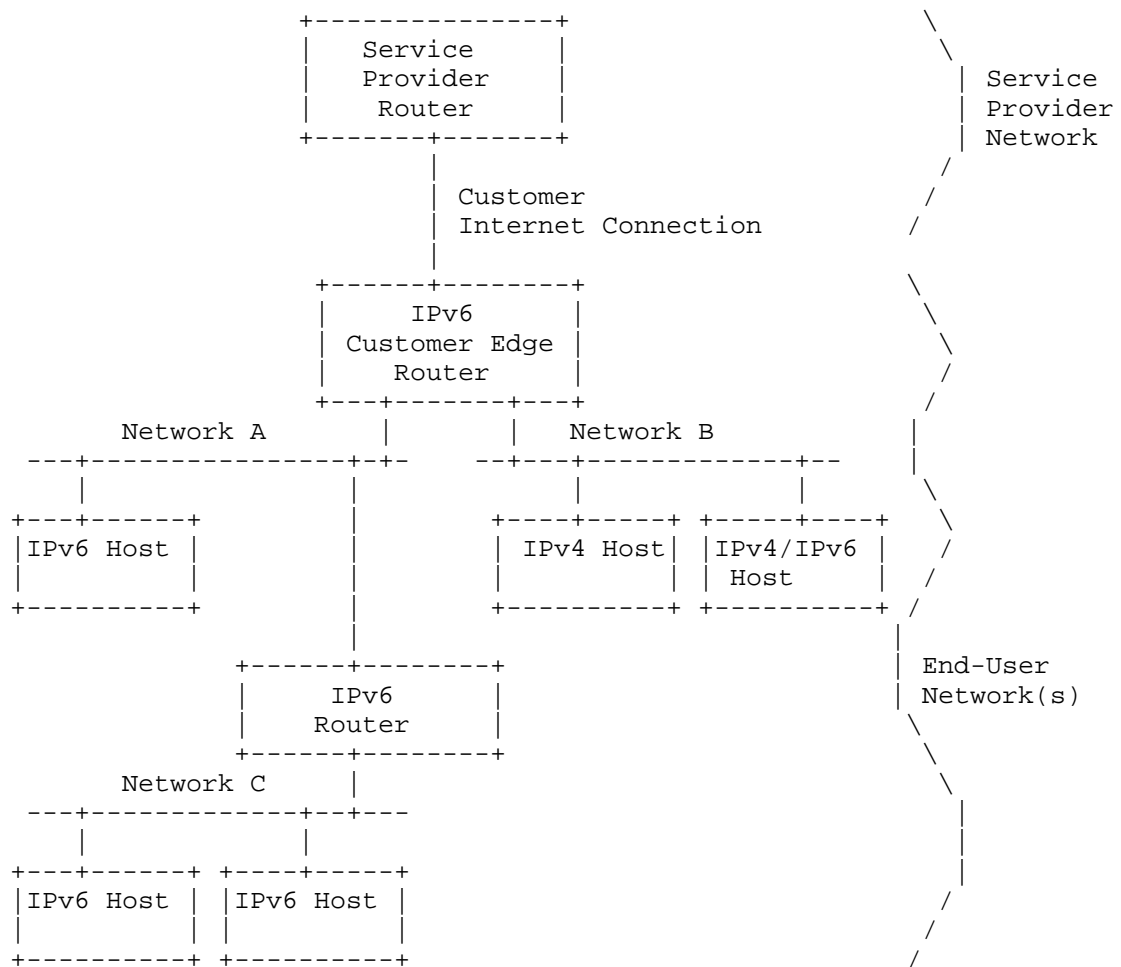


Figure 1: An Example of a Typical End-User Network

This architecture describes the:

- o Basic capabilities of an IPv6 Transition CE router
- o Provisioning of the WAN interface connecting to the service provider
- o Provisioning of the LAN interfaces

The IPv6 Transition CE router may be manually configured in an arbitrary topology with a dynamic routing protocol or using HNCP ([RFC7788]). Automatic provisioning and configuration is described

for a single IPv6 Transition CE router only.

5. Requirements

The IPv6 Transition CE router must comply with all the requirements stated in [RFC7084].

5.1. General Requirements

A new general requirement is added:

G-6 The IPv6-only CE router MUST comply with [RFC7608].

5.2. LAN-Side Configuration

A new LAN requirement is:

L-15 The IPv6 CE router SHOULD implement a DNS proxy as described in [RFC5625].

5.3. Transition Technologies Support for IPv4 Service Continuity (IPv4 as a Service - IPv4aaS)

The main target of this document is the support of IPv6-only WAN access, and while needed, the support of IPv4-only devices and applications in the customers LANs, in one side of the picture. In the other side, some remote services may stay IPv4-only, so a solution is also required for both the IPv4-only and the IPv6-only devices inside the CE are able to reach the IPv4-only services. Consequently, transition technologies to resolve both sides of the picture are considered.

In order to seamlessly provide the IPv4 Service Continuity in Customer LANs, allowing an automated IPv6 transition mechanism provisioning, a new general transition requirement is added.

General transition requirement:

TRANS-1: The IPv6 Transition CE router MUST support the DHCPv6 S46 priority option described in [RFC8026] if more than one S46 mechanisms is supported.

The following sections describe the requirements for supporting additional transition mechanisms not included in [RFC7084].

5.3.1. 464XLAT

464XLAT [RFC6877] is a technique to provide IPv4 access service to IPv6-only edge networks without encapsulation.

The IPv6 Transition CE router SHOULD support CLAT functionality. If 464XLAT is supported, it MUST be implemented according to [RFC6877]. The following CE Requirements also apply:

464XLAT requirements:

- 464XLAT-1: The IPv6 Transition CE router MUST verify if the WAN link supports native IPv4, and if that's not available, MUST enable the CLAT (in order to automatically configure [RFC6877]), unless there is a match with a valid OPTION_S46_PRIORITY (following section 1.4 of [RFC8026]), which will allow configuring any of the other transition mechanisms.
- 464XLAT-2: The IPv6 Transition CE router MUST perform IPv4 Network Address Translation (NAT) on IPv4 traffic translated using the CLAT, unless a dedicated /64 prefix has been acquired using DHCPv6-PD [RFC3633].
- 464XLAT-3: The IPv6 Transition CE router MUST implement [RFC7050] in order to discover the PLAT-side translation IPv4 and IPv6 prefix(es)/suffix(es). In environments with PCP support, the IPv6 Transition CE SHOULD follow [RFC7225] to learn the PLAT-side translation IPv4 and IPv6 prefix(es)/suffix(es) used by an upstream PCP-controlled NAT64 device.

5.3.2. Lightweight 4over6 (lw4o6)

Lw4o6 [RFC7596] specifies an extension to DS-Lite, which moves the NAPT function from the DS-Lite tunnel concentrator to the tunnel client located in the IPv6 Transition CE router, removing the requirement for a CGN function in the tunnel concentrator and reducing the amount of centralized state.

The IPv6 Transition CE router SHOULD implement lw4o6 functionality. If DS-Lite is implemented, lw4o6 MUST be supported as well. If lw4o6 is supported, it MUST be implemented according to [RFC7596]. This document takes no position on simultaneous operation of lw4o6 and native IPv4. The following IPv6 Transition CE router Requirements also apply:

Lw4o6 requirements:

- LW4O6-1: The IPv6 Transition CE router MUST support configuration of lw4o6 via the lw4o6 DHCPv6 options [RFC7598]. The IPv6 Transition CE router MAY use other mechanisms to configure lw4o6 parameters. Such mechanisms are outside the scope of this document.
- LW4O6-2: The IPv6 Transition CE router MUST support the DHCPv4-over-DHCPv6 (DHCP 4o6) transport described in [RFC7341].
- LW4O6-3: The IPv6 Transition CE router MAY support Dynamic Allocation of Shared IPv4 Addresses as described in [RFC7618].

5.3.3. MAP-E

MAP-E [RFC7597] is a mechanism for transporting IPv4 packets across an IPv6 network using IP encapsulation, including a generic mechanism for mapping between IPv6 addresses and IPv4 addresses as well as transport-layer ports.

The IPv6 Transition CE router SHOULD support MAP-E functionality. If MAP-E is supported, it MUST be implemented according to [RFC7597]. The following CE Requirements also apply:

MAP-E requirements:

- MAPE-1: The IPv6 Transition CE router MUST support configuration of MAP-E via the MAP-E DHCPv6 options [RFC7598]. The IPv6 Transition CE router MAY use other mechanisms to configure MAP-E parameters. Such mechanisms are outside the scope of this document.

5.3.4. MAP-T

MAP-T [RFC7599] is a mechanism similar to MAP-E, differing from it in that MAP-T uses IPv4-IPv6 translation, rather than encapsulation, as the form of IPv6 domain transport.

The IPv6 Transition CE router SHOULD support MAP-T functionality. If MAP-T is supported, it MUST be implemented according to [RFC7599]. The following IPv6 Transition CE Requirements also apply:

MAP-T requirements:

- MAPT-1: The CE router MUST support configuration of MAP-T via the MAP-T DHCPv6 options [RFC7598]. The IPv6 Transition CE router MAY use other mechanisms to configure MAP-T parameters. Such mechanisms are outside the scope of this

document.

6. IPv4 Multicast Support

Actual deployments support IPv4 multicast for services such as IPTV. In the transition phase it is expected that multicast services will still be provided using IPv4 to the customer LANs.

In order to support the delivery of IPv4 multicast services to IPv4 clients over an IPv6 multicast network, the IPv6 Transition CE router SHOULD support [RFC8114] and [RFC8115].

7. Code Considerations

One of the apparent main issues for vendors to include new functionalities, such as support for new transition mechanisms, is the lack of space in the flash (or equivalent) memory. However, it has been confirmed from existing open source implementations (OpenWRT/LEDE, Linux, others), that adding the support for the new transitions mechanisms, requires around 10-12Kbytes (because most of the code base is shared among several transition mechanisms already supported by [RFC7084]), as a single data plane is common to all them, which typically means about 0,15% of the existing code size in popular CEs already in the market.

It is also clear that the new requirements don't have extra cost in terms of RAM memory, neither other hardware requirements such as more powerful CPUs.

The other issue seems to be the cost of developing the code for those new functionalities. However at the time of writing this document, it has been confirmed that there are several open source versions of the required code for supporting the new transition mechanisms, and even several vendors already have implementations and provide it to ISPs, so the development cost is negligent, and only integration and testing cost may become a minor issue.

8. Security Considerations

The IPv6 Transition CE router must comply with the Security Considerations as stated in [RFC7084].

9. Acknowledgements

Thanks to James Woodyatt, Mohamed Boucadair, Masanobu Kawashima, Mikael Abrahamsson, Barbara Stark, Ole Troan and Brian Carpenter for their review and comments in previous versions of this document.

10. References

10.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC2131] Droms, R., "Dynamic Host Configuration Protocol", RFC 2131, DOI 10.17487/RFC2131, March 1997, <<https://www.rfc-editor.org/info/rfc2131>>.
- [RFC3633] Troan, O. and R. Droms, "IPv6 Prefix Options for Dynamic Host Configuration Protocol (DHCP) version 6", RFC 3633, DOI 10.17487/RFC3633, December 2003, <<https://www.rfc-editor.org/info/rfc3633>>.
- [RFC3704] Baker, F. and P. Savola, "Ingress Filtering for Multihomed Networks", BCP 84, RFC 3704, DOI 10.17487/RFC3704, March 2004, <<https://www.rfc-editor.org/info/rfc3704>>.
- [RFC4213] Nordmark, E. and R. Gilligan, "Basic Transition Mechanisms for IPv6 Hosts and Routers", RFC 4213, DOI 10.17487/RFC4213, October 2005, <<https://www.rfc-editor.org/info/rfc4213>>.
- [RFC5625] Bellis, R., "DNS Proxy Implementation Guidelines", BCP 152, RFC 5625, DOI 10.17487/RFC5625, August 2009, <<https://www.rfc-editor.org/info/rfc5625>>.
- [RFC5969] Townsley, W. and O. Troan, "IPv6 Rapid Deployment on IPv4 Infrastructures (6rd) -- Protocol Specification", RFC 5969, DOI 10.17487/RFC5969, August 2010, <<https://www.rfc-editor.org/info/rfc5969>>.
- [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>>.
- [RFC6334] Hankins, D. and T. Mrugalski, "Dynamic Host Configuration Protocol for IPv6 (DHCPv6) Option for Dual-Stack Lite", RFC 6334, DOI 10.17487/RFC6334, August 2011, <<https://www.rfc-editor.org/info/rfc6334>>.

- [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>>.
- [RFC6887] Wing, D., Ed., Cheshire, S., Boucadair, M., Penno, R., and P. Selkirk, "Port Control Protocol (PCP)", RFC 6887, DOI 10.17487/RFC6887, April 2013, <<https://www.rfc-editor.org/info/rfc6887>>.
- [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>>.
- [RFC7084] Singh, H., Beebe, W., Donley, C., and B. Stark, "Basic Requirements for IPv6 Customer Edge Routers", RFC 7084, DOI 10.17487/RFC7084, November 2013, <<https://www.rfc-editor.org/info/rfc7084>>.
- [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>>.
- [RFC7341] Sun, Q., Cui, Y., Siodelski, M., Krishnan, S., and I. Farrer, "DHCPv4-over-DHCPv6 (DHCP 4o6) Transport", RFC 7341, DOI 10.17487/RFC7341, August 2014, <<https://www.rfc-editor.org/info/rfc7341>>.
- [RFC7596] Cui, Y., Sun, Q., Boucadair, M., Tsou, T., Lee, Y., and I. Farrer, "Lightweight 4over6: An Extension to the Dual-Stack Lite Architecture", RFC 7596, DOI 10.17487/RFC7596, July 2015, <<https://www.rfc-editor.org/info/rfc7596>>.
- [RFC7597] Troan, O., Ed., Dec, W., Li, X., Bao, C., Matsushima, S., Murakami, T., and T. Taylor, Ed., "Mapping of Address and Port with Encapsulation (MAP-E)", RFC 7597, DOI 10.17487/RFC7597, July 2015, <<https://www.rfc-editor.org/info/rfc7597>>.
- [RFC7598] Mrugalski, T., Troan, O., Farrer, I., Perreault, S., Dec, W., Bao, C., Yeh, L., and X. Deng, "DHCPv6 Options for Configuration of Software Address and Port-Mapped Clients", RFC 7598, DOI 10.17487/RFC7598, July 2015, <<https://www.rfc-editor.org/info/rfc7598>>.

- [RFC7599] Li, X., Bao, C., Dec, W., Ed., Troan, O., Matsushima, S., and T. Murakami, "Mapping of Address and Port using Translation (MAP-T)", RFC 7599, DOI 10.17487/RFC7599, July 2015, <<https://www.rfc-editor.org/info/rfc7599>>.
- [RFC7608] Boucadair, M., Petrescu, A., and F. Baker, "IPv6 Prefix Length Recommendation for Forwarding", BCP 198, RFC 7608, DOI 10.17487/RFC7608, July 2015, <<https://www.rfc-editor.org/info/rfc7608>>.
- [RFC7618] Cui, Y., Sun, Q., Farrer, I., Lee, Y., Sun, Q., and M. Boucadair, "Dynamic Allocation of Shared IPv4 Addresses", RFC 7618, DOI 10.17487/RFC7618, August 2015, <<https://www.rfc-editor.org/info/rfc7618>>.
- [RFC8026] Boucadair, M. and I. Farrer, "Unified IPv4-in-IPv6 Software Customer Premises Equipment (CPE): A DHCPv6-Based Prioritization Mechanism", RFC 8026, DOI 10.17487/RFC8026, November 2016, <<https://www.rfc-editor.org/info/rfc8026>>.
- [RFC8114] Boucadair, M., Qin, C., Jacquenet, C., Lee, Y., and Q. Wang, "Delivery of IPv4 Multicast Services to IPv4 Clients over an IPv6 Multicast Network", RFC 8114, DOI 10.17487/RFC8114, March 2017, <<https://www.rfc-editor.org/info/rfc8114>>.
- [RFC8115] Boucadair, M., Qin, J., Tsou, T., and X. Deng, "DHCPv6 Option for IPv4-Embedded Multicast and Unicast IPv6 Prefixes", RFC 8115, DOI 10.17487/RFC8115, March 2017, <<https://www.rfc-editor.org/info/rfc8115>>.

10.2. Informative References

- [RFC7788] Stenberg, M., Barth, S., and P. Pfister, "Home Networking Control Protocol", RFC 7788, DOI 10.17487/RFC7788, April 2016, <<https://www.rfc-editor.org/info/rfc7788>>.
- [UPnP-IGD]
UPnP Forum, "InternetGatewayDevice:2 Device Template Version 1.01", December 2010, <<http://upnp.org/specs/gw/igd2/>>.

Authors' Addresses

Jordi Palet Martinez
The IPv6 Company
Molino de la Navata, 75
La Navata - Galapagar, Madrid 28420
Spain

EMail: jordi.palet@theipv6company.com
URI: <http://www.theipv6company.com/>

Hans M.-H. Liu
D-Link Systems, Inc.
17595 Mount Herrmann St.
Fountain Valley, California 92708
US

EMail: hans.liu@dlinkcorp.com
URI: <http://www.dlink.com/>

Network Working Group
Internet-Draft
Intended status: Informational
Expires: July 5, 2021

F. Templin, Ed.
Boeing Research & Technology
January 1, 2021

IPv6 Prefix Delegation and Multi-Addressing Models
draft-templin-v6ops-pdhost-27

Abstract

Requesting nodes typically acquire IPv6 prefixes from a prefix delegation service for the network. The requesting node can provision the prefix according to whether it acts as a router on behalf of any downstream networks and/or as a host on behalf of its local applications. In the latter case, the requesting node can use portions of the delegated prefix for its own multi-addressing purposes. This document therefore considers prefix delegation models for both the classic routing and various multi-addressing use cases.

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 July 5, 2021.

Copyright Notice

Copyright (c) 2021 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 Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	2
2. Terminology	6
3. Multi-Addressing Considerations	6
4. Multi-Addressing Alternatives for Delegated Prefixes	7
5. Address Autoconfiguration Considerations	8
6. MLD/DAD Implications	8
7. Dynamic Routing Protocol Implications	9
8. IPv6 Neighbor Discovery Implications	9
9. Prefix Delegation Services	9
10. IANA Considerations	10
11. Security Considerations	10
12. Acknowledgements	10
13. References	11
13.1. Normative References	11
13.2. Informative References	12
Appendix A. Change Log	13
Author's Address	14

1. Introduction

IPv6 Neighbor Discovery (ND) is the process by which nodes on the link discover each other's presence as well as advertise and receive configuration information. IPv6 Prefix Delegation (PD) entails 1) the communication of a prefix from a delegation service to a requesting node, 2) a representation of the prefix in the network's Routing Information Base (RIB) and the first-hop router's Forwarding Information Base (FIB), and 3) a control messaging service to maintain prefix lifetimes. Following delegation, the prefix is available for the requesting node's exclusive use and is not shared with any other nodes. This document considers prefix delegation models and multiaddressing considerations for requesting nodes that act as a router on behalf of any downstream networks and/or as a host on behalf of their local applications.

For nodes that connect downstream-attached networks (e.g., a cellphone that connects a "tethered" Internet of Things (IoT), a laptop computer with a complex internal network of virtual machines, etc.), the classic routing model applies as shown in Figure 1:

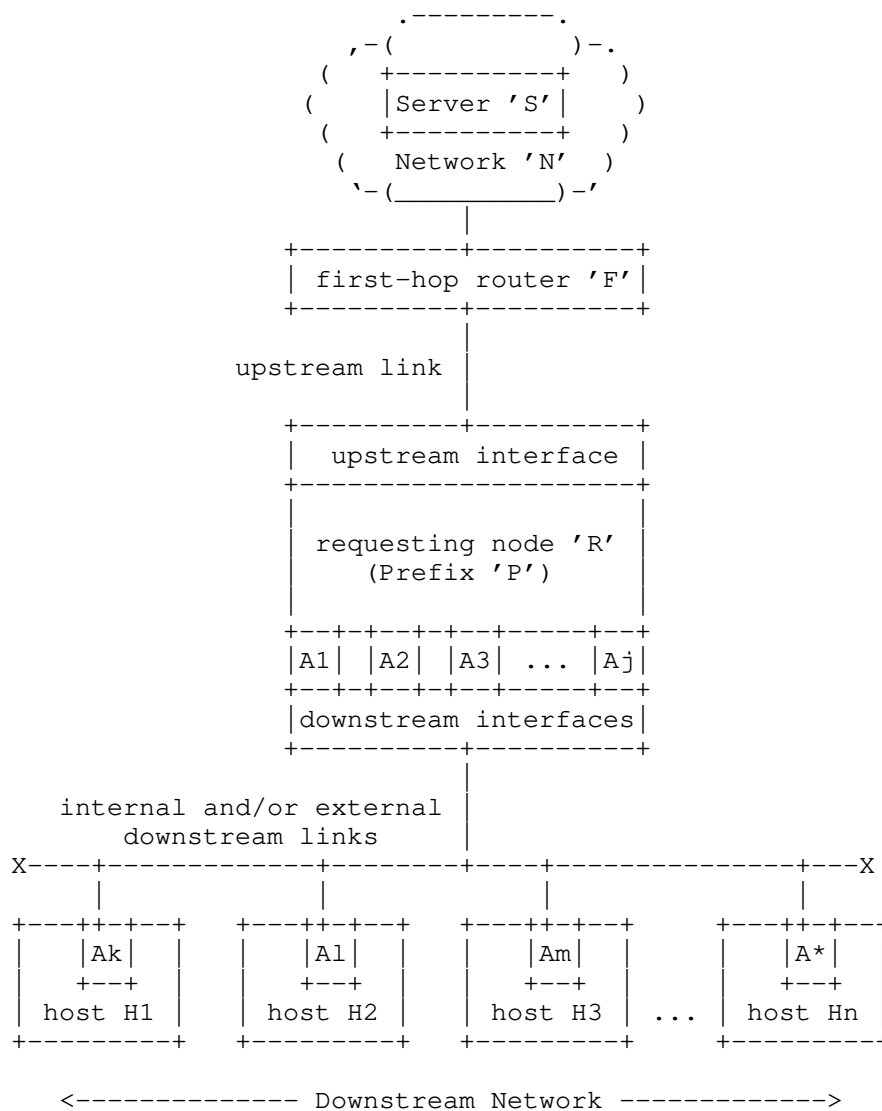


Figure 1: Classic Routing Model

In the classic routing model, requesting node 'R' has one or more upstream interfaces and connects zero or more internal and/or external downstream networks. When 'R' requests a prefix delegation, the following sequence of events transpires:

- o Server 'S' located in network 'N' delegates prefix 'P' to requesting node 'R'.

- o 'P' is injected into the RIB for 'N', and first hop router 'F' configures a FIB entry with 'R' as the next hop.
- o R' receives 'P' and assigns zero or more addresses 'A(*)' taken from 'P' to its downstream interfaces
- o 'R' advertises zero or more sub-prefixes taken from 'P' to hosts 'H(i)' on downstream networks.
- o 'R' delegates zero or more sub-prefixes taken from 'P' to requesting nodes in downstream networks.
- o 'R' acts as a router for hosts 'H(i)' on downstream networks and as a host on behalf of its local applications.

This document also considers the case when 'R' uses portions of 'P' for its own internal multi-addressing purposes. [RFC7934] provides Best Current Practice (BCP) motivations for the benefits of multi-addressing, while an operational means for providing nodes with multiple addresses is given in [RFC8273]. The following multi-addressing alternatives for delegated prefixes compliment this framework.

In a first alternative, when requesting node 'R' receives prefix 'P', it can assign addresses taken from 'P' to downstream virtual interfaces (e.g., a loopback) as shown in Figure 2:

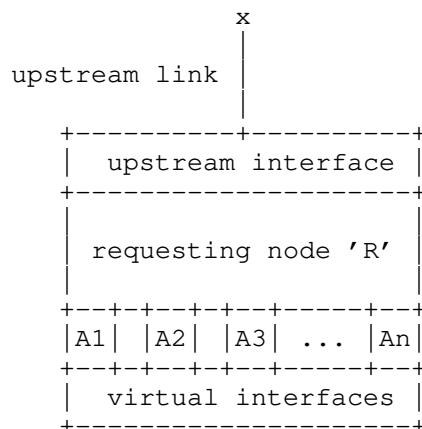


Figure 2: Address Assignment to Downstream Virtual Interfaces

In a second alternative, 'R' could assign IPv6 addresses taken from 'P' to the upstream interface over which the prefix was received as shown in Figure 3:

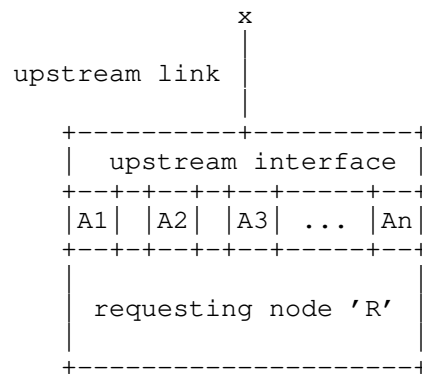


Figure 3: Upstream Interface Address Assignment

In a third alternative, 'R' could assign IPv6 addresses taken from 'P' to its local applications which appear as "psuedo" virtual interfaces as shown in Figure 4:

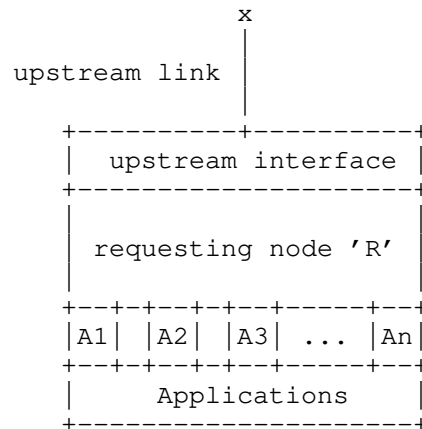


Figure 4: Application Addressing Model

With these IPv6 PD-based multi-addressing considerations, the node can configure an unlimited supply of addresses to make them available for local applications without requiring coordination with other nodes on upstream interfaces. The following sections present considerations for nodes that employ IPv6 PD mechanisms.

2. Terminology

The terms "node", "host" and "router" are the same as defined in [RFC8200]. The terms Router Solicitation (RS), Router Advertisement (RA), Neighbor Solicitation (NS), Neighbor Advertisement (NA), Redirect and Prefix Information Option (PIO) are the same as defined in [RFC4861]. All other terminology in the normative references applies, while the following terms are defined within the context of this document:

shared prefix

an IPv6 prefix that may be advertised to more than one node on the link. The router that advertises the prefix must consider the prefix as on-link so that the IPv6 ND address resolution function will identify the correct neighbor for each packet.

individual prefix

an IPv6 prefix that is advertised to exactly one node on the link, where the node may be unaware that the prefix is individual and may not participate in prefix maintenance procedures. The router that advertises the prefix can consider the prefix as on-link or not on-link. In the former case, the router performs address resolution and only forwards those packets that match one of the node's configured addresses so that the node will not receive unwanted packets. In the latter case, the router can simply forward all packets matching the prefix to the node which must then drop any packets that do not match one of its configured addresses. An example individual prefix service is documented in [RFC8273].

delegated prefix

an IPv6 prefix that is explicitly conveyed to a node for its own exclusive use, where the node is an active participant in prefix delegation and maintenance procedures. The first-hop router simply forwards all packets matching the prefix to the requesting node. The requesting node associates the prefix with downstream and/or internal virtual interfaces (i.e., and not the upstream interface).

3. Multi-Addressing Considerations

IPv6 allows nodes to assign multiple addresses to a single interface. [RFC7934] discusses options for multi-addressing as well as use cases where multi-addressing may be desirable. Address configuration options for multi-addressing include Stateless Address AutoConfiguration (SLAAC) [RFC4862], Dynamic Host Configuration Protocol for IPv6 (DHCPv6) address configuration [RFC8415], manual configuration, etc.

Nodes configure addresses from a shared or individual prefix and assign them to the upstream interface over which the prefix was received. When the node assigns the addresses, it is required to use Multicast Listener Discovery (MLD) [RFC3810] to join the appropriate solicited-node multicast group(s) and to use the Duplicate Address Detection (DAD) algorithm [RFC4862] to ensure that no other node configures a duplicate address.

In contrast, a node that configures addresses from a delegated prefix can assign them without invoking MLD/DAD on an upstream interface, since the prefix has been delegated to the node for its own exclusive use and is not shared with any other nodes.

4. Multi-Addressing Alternatives for Delegated Prefixes

When a node receives a delegated prefix, it has many alternatives for provisioning the prefix to its local interfaces and/or downstream networks. [RFC7278] discusses alternatives for provisioning a prefix obtained by a User Equipment (UE) device under the 3rd Generation Partnership Program (3GPP) service model. This document considers the more general case when the node receives a delegated prefix explicitly provided for its own exclusive use.

When the node receives the prefix, it can distribute the prefix to internal (virtual) or external (physical) downstream networks and optionally configure addresses for itself on downstream interfaces. The node then acts as a router on behalf of its downstream networks.

The node could instead (or in addition) use portions of the delegated prefix for its own multi-addressing purposes. In a first alternative, the node can assign as many addresses as it wants from the prefix to downstream virtual interfaces.

In a second alternative, the node can assign as many addresses as it wants from the prefix to the upstream interface over which the prefix was received, but in normal practice does not assign the prefix itself (or subnets from the prefix) to the upstream interface. If the node assigned the prefix to the upstream interface, any neighbors on the upstream link receiving an RA could configure addresses from the prefix and a default route with the node as the next hop. This could create a loop where upstream link neighbors send packets to the node which in turn forwards them to another upstream link neighbor. Still, there may be cases where the node provides services for dependent neighbors on the upstream link that have no other means of connecting to the network. ([RFC8415] chose to remain silent on this subject since it is operational rather than functional in nature.)

In a third alternative, the node can assign addresses taken from the delegated prefix to its local applications. The applications themselves then serve as virtual interfaces. (Note that, in the future, the practice of assigning unique non-link-local IPv6 addresses to applications could obviate the need for transport protocol port numbers.)

In these multi-addressing cases, the node normally assigns the prefix itself to a virtual interface such as a loopback so that unused portions of the prefix are correctly identified as unreachable. The node then acts as a host on behalf of its local applications even though neighbors on the upstream link consider it as a router.

5. Address Autoconfiguration Considerations

Nodes autoconfigure addresses according to Section 6 of IPv6 Node Requirements [RFC8504].

Nodes that connect to a network that spans more than just a single LAN configure at least one non-link-local address, i.e., for network management and error reporting purposes.

Nodes recognize the Subnet Router Anycast address [RFC4291] for each delegated prefix. Therefore, the node's use of the Subnet Router Anycast address must be indistinguishable from the behavior of an ordinary router when viewed from the outside world.

6. MLD/DAD Implications

When a node configures addresses for itself from a shared or individual prefix (and when the interface variable 'DupAddrDetectTransmits' is non-zero [RFC4862]), the node performs MLD/DAD by sending multicast messages over the upstream interface to test whether there is another node on the link that configures a duplicate address. When there are many such addresses and/or many such nodes, this could result in substantial multicast traffic that affects all nodes on the link.

When a node configures addresses for itself from a delegated prefix and assigns them on downstream interfaces, it can configure as many addresses as it wants without performing MLD/DAD for any of the addresses over the upstream interface.

When a node configures addresses for itself from a delegated prefix and assigns them on the upstream interface over which the prefix was received, the node honors MLD/DAD procedures according to the interface's 'DupAddrDetectTransmits' variable.

7. Dynamic Routing Protocol Implications

Nodes that receive delegated prefixes can be configured to either participate or not participate in a dynamic routing protocol over the upstream interface. When there are many nodes on the upstream link, dynamic routing protocol participation might be impractical due to scaling limitations, and may also be exacerbated by factors such as node mobility.

Unless it participates in a dynamic routing protocol, the node initially has only a default route pointing to a neighbor via an upstream interface. This means that packets sent by the node over an upstream interface will initially go through a default router even if there is a better first-hop node on the link. The node may subsequently receive Redirect messages from the default router that identify a better first-hop.

8. IPv6 Neighbor Discovery Implications

According to [RFC4861], when a node receives a shared or individual prefix with "L=1" and has a packet to send to an IPv6 destination within the prefix, it is required to use the IPv6 ND address resolution function to resolve the link-layer address of a neighbor on the link that configures the address.

Also according to [RFC4861], when a node receives a shared or individual prefix with "L=0" and has a packet to send to an IPv6 destination within the prefix, it sends the packet to a default router since "L=0" makes no statement about on-link or off-link properties of the prefix.

When a node requires a delegated prefix, it acts as a simple host by sending RS messages over the upstream interface in the manner described in Section 4.2 of [RFC7084] and invokes prefix delegation services as discussed in Section 9. The node considers the upstream interface as a non-advertising interface [RFC4861], i.e., it does not send RA messages over the upstream interface. The node further does not perform the IPv6 ND address resolution function over the upstream interface, since the delegated prefix is by definition not associated with the upstream interface.

9. Prefix Delegation Services

Selection of prefix delegation services must be considered according to specific use cases. An example service is that offered by standard DHCPv6 Prefix Delegation [RFC8415]. Alternative services based on IPv6 ND messaging have also been proposed [I-D.templin-6man-dhcpv6-ndopt][I-D.naveen-slaac-prefix-management].

Other, non-router, mechanisms may exist, such as proprietary IPAMs, [I-D.ietf-anima-prefix-management] and [I-D.li-opsawg-address-pool-management-arch]. Requirements for extending an IPv6 /64 Prefix from a Third Generation Partnership Project (3GPP) Mobile Interface to a LAN Link are discussed in [RFC7278].

10. IANA Considerations

This document introduces no IANA considerations.

11. Security Considerations

Security considerations for IPv6 Neighbor Discovery [RFC4861] and any applicable PD mechanisms apply to this document. Nodes that manage their delegated prefixes such that MLD/DAD procedures are not needed on the upstream interface can avoid introducing multicast messaging congestion on the upstream link. Also, routers that delegate prefixes keep only a single neighbor cache entry for each prefix delegation recipient, meaning that the router's neighbor cache cannot be subject to address resolution-based resource exhaustion attacks.

For shared and individual prefixes, if the advertising router considers the prefix as on-link the IPv6 ND address resolution function will prevent unwanted IPv6 packets from reaching the node. For delegated prefixes and individual prefixes that are not considered on-link, the router delivers all packets that match the prefix to the node. In that case, the node may receive unwanted IPv6 packets via an upstream interface for which it has no matching configured address. The node then drops the packets and observes the ICMPv6 "Destination Unreachable - Address/Port unreachable" procedures discussed in [RFC4443].

The node may also receive IPv6 packets via an upstream interface that do not match any of the node's delegated prefixes. In that case, the node drops the packets and observes the ICMPv6 "Destination Unreachable - No route to destination" procedures discussed in [RFC4443]. Dropping the packets is necessary to avoid a reflection attack that would cause the node to forward packets received from an upstream interface via the same or a different upstream interface.

12. Acknowledgements

This work was motivated by discussions on the v6ops list. Mark Smith, Ricardo Pelaez-Negro, Edwin Cordeiro, Fred Baker, Ron Bonica, Naveen Lakshman, Ole Troan, Bob Hinden, Brian Carpenter, Joel Halpern, Albert Manfredi, Dusan Mudric, Paul Marks, Joe Touch, Alex

Petrescu, Lorenzo Colitti, Tatuya Jinmei and Naveen Kottapalli provided useful comments that have greatly improved the document.

This work is aligned with the NASA Safe Autonomous Systems Operation (SASO) program under NASA contract number NNA16BD84C.

This work is aligned with the FAA as per the SE2025 contract number DTFAWA-15-D-00030.

This work is aligned with the Boeing Commercial Airplanes (BCA) Internet of Things (IoT) and autonomy programs.

This work is aligned with the Boeing Information Technology (BIT) MobileNet program.

13. References

13.1. Normative References

- [RFC3810] Vida, R., Ed. and L. Costa, Ed., "Multicast Listener Discovery Version 2 (MLDv2) for IPv6", RFC 3810, DOI 10.17487/RFC3810, June 2004, <<https://www.rfc-editor.org/info/rfc3810>>.
- [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>>.
- [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>>.

- [RFC8415] Mrugalski, T., Siodelski, M., Volz, B., Yourtchenko, A., Richardson, M., Jiang, S., Lemon, T., and T. Winters, "Dynamic Host Configuration Protocol for IPv6 (DHCPv6)", RFC 8415, DOI 10.17487/RFC8415, November 2018, <<https://www.rfc-editor.org/info/rfc8415>>.

13.2. Informative References

- [I-D.ietf-anima-prefix-management]
Jiang, S., Du, Z., Carpenter, B., and Q. Sun, "Autonomic IPv6 Edge Prefix Management in Large-scale Networks", draft-ietf-anima-prefix-management-07 (work in progress), December 2017.
- [I-D.li-opsawg-address-pool-management-arch]
Li, C., Xie, C., Kumar, R., Fioccola, G., Xu, W., LIU, W., Ma, D., and J. Bi, "Coordinated Address Space Management architecture", draft-li-opsawg-address-pool-management-arch-01 (work in progress), July 2018.
- [I-D.naveen-slaac-prefix-management]
Kottapalli, N., "IPv6 Stateless Prefix Management", draft-naveen-slaac-prefix-management-00 (work in progress), November 2018.
- [I-D.templin-6man-dhcpv6-ndopt]
Templin, F., "A Unified Stateful/Stateless Configuration Service for IPv6", draft-templin-6man-dhcpv6-ndopt-10 (work in progress), June 2020.
- [I-D.templin-6man-rio-redirect]
Templin, F. and j. woodyatt, "Route Information Options in IPv6 Neighbor Discovery", draft-templin-6man-rio-redirect-08 (work in progress), June 2019.
- [RFC4291] Hinden, R. and S. Deering, "IP Version 6 Addressing Architecture", RFC 4291, DOI 10.17487/RFC4291, February 2006, <<https://www.rfc-editor.org/info/rfc4291>>.
- [RFC7084] Singh, H., Beebee, W., Donley, C., and B. Stark, "Basic Requirements for IPv6 Customer Edge Routers", RFC 7084, DOI 10.17487/RFC7084, November 2013, <<https://www.rfc-editor.org/info/rfc7084>>.

- [RFC7278] Byrne, C., Drown, D., and A. Vizdal, "Extending an IPv6 /64 Prefix from a Third Generation Partnership Project (3GPP) Mobile Interface to a LAN Link", RFC 7278, DOI 10.17487/RFC7278, June 2014, <<https://www.rfc-editor.org/info/rfc7278>>.
- [RFC7934] Colitti, L., Cerf, V., Cheshire, S., and D. Schinazi, "Host Address Availability Recommendations", BCP 204, RFC 7934, DOI 10.17487/RFC7934, July 2016, <<https://www.rfc-editor.org/info/rfc7934>>.
- [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>>.
- [RFC8504] Chown, T., Loughney, J., and T. Winters, "IPv6 Node Requirements", BCP 220, RFC 8504, DOI 10.17487/RFC8504, January 2019, <<https://www.rfc-editor.org/info/rfc8504>>.

Appendix A. Change Log

<< RFC Editor - remove prior to publication >>

Changes from -25 to -26:

- o Version and reference update

Changes from -24 to -25:

- o Version and reference update

Changes from -23 to -24:

- o Version and reference update

Changes from -22 to -23:

- o Changed DHCPv6 references to RFC8415. Deprecate RFC3315 and RFC3633.
- o New text on assignment of addresses and prefixes on the upstream interface.

Changes from -21 to -22:

- o Changes to address list comments contributed by Lorenzo Colitti, Tatuya Jinmei, Brian Carpenter and Fred Baker.

- o Deleted section on ICMPv6 - now defer to normative reference [RFC4443].
- o Discuss 'DupAddrDetectTransmits' variable implications under MLD/DAD considerations.

Changes from -20 to -21:

- o Re-worked classic routing model section
- o Included multi-addressing case where addresses may be assigned to applications
- o Removed strong/weak end system discussions

Changes from -19 to -20:

- o figure 1 updates to show Server as being somewhere in the network
- o Introductory material to show relation to other RFCs on multi-addressing

Changes from -18 to -19:

- o added new section on Prefix Delegation Services

Changes from -17 to -18:

- o re-worked discussion on the prefix delegation service in Section 1
- o updated figures in Section 1

Changes from -16 to -17:

- o added supporting text in the introduction to discuss the Delegating Router's relationship with the Requesting Router and with supporting infrastructure in the operator's network
- o updated figures in introduction to include representation of operator's network
- o added new section on Address Autoconfiguration Considerations

Author's Address

Fred L. Templin (editor)
Boeing Research & Technology
P.O. Box 3707
Seattle, WA 98124
USA

Email: fltemplin@acm.org