Transmission of SCHC-compressed packets over IEEE 802.15.4 networks
draft-gomez-6lo-schc-15dot4-03

Abstract

A framework called Static Context Header Compression and fragmentation (SCHC) has been designed with the primary goal of supporting IPv6 over Low Power Wide Area Network (LPWAN) technologies [RFC8724]. One of the SCHC components is a header compression mechanism. If used properly, SCHC header compression allows a greater compression ratio than that achievable with traditional 6LoWPAN header compression [RFC6282]. For this reason, it may make sense to use SCHC header compression in some 6LoWPAN environments, including IEEE 802.15.4 networks. This document specifies how a SCHC-compressed packet can be carried over IEEE 802.15.4 networks.

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 11, 2023.

Copyright Notice

Copyright (c) 2022 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
1. Introduction

RFC 6282 is the main specification for IPv6 over Low power Wireless Personal Area Network (6LoWPAN) IPv6 header compression [RFC6282]. This RFC was designed assuming IEEE 802.15.4 as the layer below the 6LoWPAN adaptation layer, and it has also been reused (with proper adaptations) for IPv6 header compression over many other technologies relatively similar to IEEE 802.15.4 in terms of characteristics such as physical layer bit rate, layer 2 maximum payload size, etc.

Examples of such technologies comprise BLE, DECT-ULE, ITU G.9959, MS/
IP, NFC, and PLC. RFC 6282 provides additional functionality, such as a mechanism for UDP header compression.

In the best cases, RFC 6282 allows to compress a 40-byte IPv6 header down to a 2-byte compressed header (for link-local interactions) or a 3-byte compressed header (when global IPv6 addresses are used). On the other hand, an RFC 6282 compressed UDP header has a typical size of 4 bytes. Therefore, in advantageous conditions, a 48-byte uncompressed IPv6/UDP header may be compressed down to a 6-byte format (when using link-local addresses) or a 7-byte format (for global interactions) by using RFC 6282.

Recently, a framework called Static Context Header Compression (SCHC) has been designed with the primary goal of supporting IPv6 over Low Power Wide Area Network (LPWAN) technologies [RFC8724]. SCHC comprises header compression and fragmentation functionality tailored to the extraordinary constraints of LPWAN technologies, which are more severe than those exhibited by IEEE 802.15.4 or other relatively similar technologies. SCHC header compression allows a greater compression ratio than that of RFC 6282. If used properly, SCHC allows to compress an IPv6/UDP header down to e.g. a single byte. In addition, SCHC can be used to compress Constrained Application Protocol (CoAP) headers as well [RFC7252][RFC8824], which further increases the achievable performance improvement of using SCHC header compression, since there is no 6LoWPAN header compression mechanism defined for CoAP. Therefore, it may make sense to use SCHC header compression in some 6LoWPAN environments [I-D.toutain-6lo-6lo-and-schc], including IEEE 802.15.4 networks, considering its greater efficiency.

If SCHC header compression is added to the panoply of header compression mechanisms used in 6LoWPAN environments, then there is a need to signal when a packet header has been compressed by using SCHC. To this end, the present document specifies a 6LoWPAN Dispatch Type for SCHC header compression [RFC4944].

This document specifies how a SCHC-compressed packet can be carried over IEEE 802.15.4 networks. Note that, as per this document, and while SCHC defines fragmentation mechanisms as well, 6LoWPAN/6Lo fragmentation is used when necessary to transport SCHC-compressed packets over IEEE 802.15.4 networks [RFC4944][RFC8930][RFC8931].

TO-DO: indicate here any specific updates of RFC 8724 for use over IEEE 802.15.4. 
2. Terminology

2.1. Requirements language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP14 [RFC2119], [RFC8174], when, and only when, they appear in all capitals, as shown here.

2.2. Background on SCHC

The reader is expected to be familiar with the terms and concepts defined in the specification of SCHC (RFC 8724).

3. Architecture

3.1. Network topologies

IEEE 802.15.4 supports two main network topologies: the star topology, and the peer-to-peer (i.e., mesh) topology.

SCHC has been designed for LPWAN technologies, which are typically based on a star topology where constrained devices (e.g., sensors) communicate with a less constrained, central network gateway [RFC 8376]. However, as stated in [draft-ietf-lpwan-architecture], SCHC is generic and it can also be used in networking environments beyond the ones originally considered for SCHC.

SCHC compression is applicable to both star topology and mesh topology IEEE 802.15.4 networks.

3.2. Protocol stack

The traditional 6LoWPAN-based protocol stack for constrained devices (Figure 1, left) places the 6LoWPAN adaptation layer between IPv6 and an underlying technology such as IEEE 802.15.4. Suitable upper layer protocols include CoAP [RFC7252] and UDP. (Note that, while CoAP has also been specified over TCP, and TCP may play a significant role in IoT environments [RFC9006], 6LoWPAN header compression has not been defined for TCP.)

6LoWPAN can be envisioned as a set of two main sublayers, where the upper one provides header compression, while the lower one offers fragmentation.

This document defines an alternative approach for packet header compression over IEEE 802.15.4, which leads to a modified protocol
SCHC compression may be applied to the headers of different protocols or sets of protocols. Some examples include: i) IPv6 packet headers, ii) joint IPv6 and UDP packet headers, iii) joint IPv6, UDP and CoAP packet headers, etc.

4. Frame Format

This document defines the frame format to be used when a SCHC-compressed packet is carried over IEEE 802.15.4. Such format is carried as IEEE 802.15.4 frame payload. The format comprises a SCHC Dispatch Type, a SCHC Packet (i.e. a SCHC-compressed packet (RFC 8724), and Padding bits, if any). Figure 2 illustrates the described frame format.
4.1. SCHC Dispatch

Adding SCHC header compression to the panoply of header compression mechanisms used in 6LoWPAN/6Lo environments creates the need to signal when a packet header has been compressed by using SCHC. To this end, the present document specifies the SCHC Dispatch. The SCHC Dispatch indicates that the next field in the frame format is a SCHC-compressed header (SCHC Header in Figure 2, see 4.2)).

This document defines the SCHC Dispatch as a 6LoWPAN Dispatch Type for SCHC header compression [RFC4944]. With the aim to minimize overhead, the present document allocates a 1-byte pattern in Page 0 [RFC8025] for the SCHC Dispatch Type:

SCHC Dispatch Type bit pattern: 01000100 (Page 0) (Note: to be confirmed by IANA))

4.2. SCHC Header

SCHC Header (Figure 2) corresponds to a packet header that has been compressed by using SCHC. As defined in [RFC8724], the SCHC Header comprises a RuleID, and a compression residue. As per the present specification, a RuleID size between 1 and 16 bits is RECOMMENDED. In order to decide the RuleID size to be used in a network, the trade-off between (compressed) header overhead and the number of Rules needs to be carefully assessed.

4.3. Padding

If SCHC header compression leads to a SCHC Packet size of a non-integer number of bytes, padding bits of value equal to zero MUST be appended to the SCHC Packet as appropriate to align to an octet boundary.
5. SCHC compression for IPv6, UDP, and CoAP headers

SCHC header compression may be applied to the headers of different protocols or sets of protocols. Some examples include: i) IPv6 packet headers, ii) joint IPv6 and UDP packet headers, iii) joint IPv6, UDP and CoAP packet headers, etc.

5.1. SCHC compression for IPv6 and UDP headers

IPv6 and UDP header fields MUST be compressed as per Section 10 of RFC 8724.

IPv6 addresses are split into two 64-bit-long fields; one for the prefix and one for the Interface Identifier (IID).

To allow for a single Rule being used for both directions, RFC 8724 identifies IPv6 addresses and UDP ports by their role (Dev or App) and not by their position in the header (source or destination). This optimization can be used as is in some IEEE 802.15.4 networks (e.g., an IEEE 802.15.4 star topology where the peripheral devices (Devs) send/receive packets to/from a network-side entity (App)).

However, in some types of 6LoWPAN environments (e.g., when a sender and its destination are both peer nodes in a mesh topology network), additional functionality is needed to allow use of the Dev and App roles for C/D. In this case, each SCHC C/D entity needs to know its role (Dev or App) in addition to the Rule(s), and corresponding RuleIDs, for each endpoint it communicates with before such communication occurs [I-D.ietf-lpwan-architecture]. In such cases, the terms Uplink and Downlink that have been defined in RFC 8724 need to be understood in the context of each specific pair of endpoints.

5.1.1. Compression of IPv6 addresses

Compression of IPv6 source and destination prefixes MUST be performed as per Section 10.7.1 of RFC 8724. Additional guidance is given in the present section.

Compression of IPv6 source and destination IIDs MUST be performed as per Section 10.7.2 of RFC 8724. One particular consideration when SCHC C/D is used in IEEE 802.15.4 networks is that, in contrast with some LPWAN technologies, IEEE 802.15.4 data frame headers include both source and destination fields. If the Dev or App IID are based on an L2 address, in some cases the IID can be reconstructed with information coming from the L2 header. Therefore, in those cases, DevIID and AppIID CDAs can be used.
5.2. SCHC compression for CoAP headers

CoAP header fields MUST be compressed as per Sections 4 to 6 of RFC 8824. Additional guidance is given in this section.

For CoAP header compression/decompression, the SCHC Rules description uses direction information in order to reduce the number of Rules needed to compress headers.

As stated in 5.1, in some types of 6LoWPAN environments (e.g., when a sender and its destination are both peer nodes in a mesh topology network), each SCHC C/D entity needs to know its role (Dev or App), in addition to the Rule(s), and corresponding RuleIDs, for each endpoint it communicates with before such communication occurs [I-D.ietf-lpwan-architecture]. Therefore, in such cases, direction information will be specific to each pair of endpoints.

5.3. Header compression examples

TO-DO: provide examples for IPv6-only, IPv6/UDP and IPv6/UDP/CoAP.

6. Multihop communication

6LoWPAN defines two approaches for multihop communication: Route-Over and Mesh-Under [RFC6606]. In Route-Over, routing is performed at the IP layer. In Mesh-Under, routing functionality is located at the adaptation layer, below IP.

6.1. Route-Over

SCHC header compression MAY be used in a Route-Over network in a straightforward approach, whereby all network nodes MUST store all the Rules in use by any nodes in the network.

Alternatively, in a Route-Over network that uses the IPv6 Routing Protocol for Low-Power and Lossy Networks (RPL) [RFC6550], the RPL non-storing mode and [RFC8138] MAY be exploited in order to transmit SCHC-compressed packets. In this approach, a network node MUST store the Rules defined for its communication with other endpoints.

(Further details TBD.)

6.2. Mesh-Under

When SCHC header compression is used in a Mesh-Under network, a network node MUST store the Rules defined for its communication with other endpoints. In this case, a RuleID MAY be reused across disjoint pairs of endpoints, to identify different Rules used by such
disjoint pairs of endpoints, at the expense of increased RuleID management and device configuration complexity.

7. Fragmentation and reassembly

After applying SCHC header compression to a packet intended for transmission, if the size of the resulting SCHC Packet (Section 4) exceeds the IEEE 802.15.4 frame payload space available, such SCHC Packet MUST be fragmented, carried and reassembled by means of the fragmentation and reassembly functionality defined by 6LoWPAN [RFC4944] or 6Lo [RFC8930][RFC8931].

In a Route-Over multihop network, the 6LoWPAN fragment forwarding technique called Virtual Reassembly Buffer (VRB) [RFC8930] SHOULD be used. However, VRB might not be the best approach for a particular network, e.g., if at least one of the caveats described in Section 6 of RFC 8930 is unacceptable.

8. IANA Considerations

This document requests the allocation of the Dispatch Type Field bit pattern 01000100 (Page 0) as SCHC Dispatch Type.

9. Security Considerations

This document does not define SCHC header compression functionality beyond the one defined in RFC 8724. Therefore, the security considerations in section 12.1 of RFC 8724 and in section 9 of RFC 8824 apply.

As a safety measure, a SCHC decompressor implementing the present specification MUST NOT reconstruct a packet larger than 1500 bytes [RFC8724].

IEEE 802.15.4 networks support link-layer security mechanisms such as encryption and authentication. As in RFC 8824, the use of a cryptographic integrity-protection mechanism to protect the SCHC headers is REQUIRED.

10. Acknowledgments

Ana Minaburo and Laurent Toutain suggested for the first time the use of SCHC in environments where 6LoWPAN has traditionally been used. Laurent Toutain, Pascal Thubert, Dominique Barthel, Guangpeng Li, and Carsten Bormann made comments that helped shape this document.

Carles Gomez has been funded in part by the Spanish Government through project PID2019-106808RA-I00, and by Secretaria
11. References

11.1. Normative References


11.2. Informative References

[I-D.toutain-6lo-6lo-and-schc]

Authors’ Addresses

Carles Gomez
UPC
C/Esteve Terradas, 7
Castelldefels 08860
Spain
Email: carlesgo@entel.upc.edu

Ana Minaburo
Acklio
1137A avenue des Champs Blancs
Cesson-Sevigne Cedex 35510
France
Email: ana@ackl.io
IPv6 Neighbor Discovery Multicast Address Listener Registration
draft-ietf-6lo-multicast-registration-07

Abstract

This document updates RFC 8505 to enable a listener to register an IPv6 anycast or and subscribe to an IPv6 multicast address; the draft updates RFC 6550 (RPL) to add a new Non-Storing Multicast Mode and a new support for anycast addresses in Storing and Non-Storing Modes. This document extends RFC 9010 to enable the 6LR to inject the anycast and multicast addresses in RPL.

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 24 December 2022.

Copyright Notice

Copyright (c) 2022 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
license-info) in effect on the date of publication of this document.
Please review these documents carefully, as they describe your rights
and restrictions with respect to this document. Code Components
extracted from this document must include Revised BSD License text as
described in Section 4.e of the Trust Legal Provisions and are
provided without warranty as described in the Revised BSD License.

Table of Contents

1. Introduction ........................................... 3
2. Terminology ............................................ 4
   2.1. Requirements Language .............................. 4
   2.2. References ........................................ 5
   2.3. Glossary ......................................... 5
3. Overview ............................................... 6
4. Extending RFC 7400 ..................................... 9
5. Updating RFC 6550 ..................................... 10
   5.1. Updating MOP 3 .................................... 10
   5.2. New Non-Storing Multicast MOP ..................... 10
   5.3. RPL Anycast Operation ............................ 11
   5.4. New RPL Target Option Flags ....................... 13
6. Updating RFC 8505 ..................................... 13
   6.1. New EARO flag .................................... 13
   6.2. New EDAR Message Flag field ....................... 14
   6.3. Registering Extensions ............................ 15
7. Updating RFC 9010 ..................................... 18
8. Leveraging RFC 8928 ................................... 18
9. Node Uptime Option .................................... 19
10. Deployment considerations ............................. 21
11. Security Considerations ............................... 23
12. Backward Compatibility ............................... 23
13. IANA Considerations .................................. 24
   13.1. New EDAR Message Flags Subregistry ............. 24
   13.2. New EARO flags .................................. 24
   13.3. New RTO flags .................................. 25
   13.4. New RPL Mode of Operation ....................... 25
   13.5. New 6LoWPAN Capability Bits ..................... 26
   13.6. New Address Registration Option Status Values . 26
   13.7. New IPv6 Neighbor Discovery Option ............. 26
14. Acknowledgments ...................................... 27
15. Normative References ................................ 27
16. Informative References ............................... 29
Author’s Address .......................................... 30
1. Introduction

The design of Low Power and Lossy Networks (LLNs) is generally focused on saving energy, which is the most constrained resource of all. Other design constraints, such as a limited memory capacity, duty cycling of the LLN devices and low-power lossy transmissions, derive from that primary concern. The radio (both transmitting or simply listening) is a major energy drain and the LLN protocols must be adapted to allow the nodes to remain sleeping with the radio turned off at most times.

The "Routing Protocol for Low Power and Lossy Networks" [RFC6550] (RPL) provides IPv6 [RFC8200] routing services within such constraints. To save signaling and routing state in constrained networks, the RPL routing is only performed along a Destination-Oriented Directed Acyclic Graph (DODAG) that is optimized to reach a Root node, as opposed to along the shortest path between 2 peers, whatever that would mean in each LLN.

This trades the quality of peer-to-peer (P2P) paths for a vastly reduced amount of control traffic and routing state that would be required to operate an any-to-any shortest path protocol. Additionally, broken routes may be fixed lazily and on-demand, based on dataplane inconsistency discovery, which avoids wasting energy in the proactive repair of unused paths.

Section 12 of [RFC6550] details the "Storing Mode of Operation with multicast support" with source-independent multicast routing in RPL.

The classical "IPv6 Neighbor Discovery (IPv6 ND) Protocol" [RFC4861] [RFC4862] was defined for serial links and shared transit media such as Ethernet at a time when broadcast was cheap on those media while memory for neighbor cache was expensive. It was thus designed as a reactive protocol that relies on caching and multicast operations for the Address Discovery (aka Lookup) and Duplicate Address Detection (DAD) of IPv6 unicast addresses. Those multicast operations typically impact every node on-link when at most one is really targeted, which is a waste of energy, and imply that all nodes are awake to hear the request, which is inconsistent with power saving (sleeping) modes.

The original 6LoWPAN ND, "Neighbor Discovery Optimizations for 6LoWPAN networks" [RFC6775], was introduced to avoid the excessive use of multicast messages and enable IPv6 ND for operations over energy-constrained nodes. [RFC6775] changes the classical IPv6 ND model to proactively establish the Neighbor Cache Entry (NCE) associated to the unicast address of a 6LoWPAN Node (6LN) in the a 6LoWPAN Router(s) (6LR) that serves it. To that effect, [RFC6775]
defines a new Address Registration Option (ARO) that is placed in unicast Neighbor Solicitation (NS) and Neighbor Advertisement (NA) messages between the 6LN and the 6LR.

"Registration Extensions for 6LoWPAN Neighbor Discovery" [RFC8505] updates [RFC6775] into a generic Address Registration mechanism that can be used to access services such as routing and ND proxy and introduces the Extended Address Registration Option (EARO) for that purpose. This provides a routing-agnostic interface for a host to request that the router injects a unicast IPv6 address in the local routing protocol and provide return reachability for that address.

"Routing for RPL Leaves" [RFC9010] provides the router counterpart of the mechanism for a host that implements [RFC8505] to inject its unicast Unique Local Addresses (ULAs) and Global Unicast Addresses (GUAs) in RPL. But though RPL also provides multicast routing, 6LoWPAN ND supports only the registration of unicast addresses and there is no equivalent of [RFC9010] to specify the 6LR behavior upon the registration of one or more multicast address.

The "Multicast Listener Discovery Version 2 (MLDv2) for IPv6" [RFC3810] enables the router to learn which node listens to which multicast address, but as the classical IPv6 ND protocol, MLD relies on multicasting Queries to all nodes, which is unfit for low power operations. As for IPv6 ND, it makes sense to let the 6LNs control when and how they maintain the state associated to their multicast addresses in the 6LR, e.g., during their own wake time. In the case of a constrained node that already implements [RFC8505] for unicast reachability, it makes sense to extend to that support to register the multicast addresses they listen to.

This specification extends [RFC8505] and [RFC9010] to add the capability for the 6LN to register anycast and multicast addresses and for the 6LR to inject them in RPL when appropriate.

2. Terminology

2.1. Requirements Language

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

This document uses terms and concepts that are discussed in:

* "Neighbor Discovery for IP version 6" [RFC4861] and "IPv6 Stateless address Autoconfiguration" [RFC4862],

* Neighbor Discovery Optimization for Low-Power and Lossy Networks [RFC6775], as well as

* "Registration Extensions for 6LoWPAN Neighbor Discovery" [RFC8505] and

* "Using RPI Option Type, Routing Header for Source Routes, and IPv6-in-IPv6 Encapsulation in the RPL Data Plane" [RFC9008].

2.3. Glossary

This document uses the following acronyms:

6BBR 6LoWPAN Backbone Router
6BBR 6LoWPAN Border Router
6LN 6LoWPAN Node
6LR 6LoWPAN Router
6CIO Capability Indication Option
AMC Address Mapping Confirmation
AMR Address Mapping Request
ARO Address Registration Option
DAC Duplicate Address Confirmation
DAD Duplicate Address Detection
DAR Duplicate Address Request
EARO Extended Address Registration Option
EDAC Extended Duplicate Address Confirmation
EDAR Extended Duplicate Address Request
DODAG Destination-Oriented Directed Acyclic Graph
IR Ingress Replication
LLN Low-Power and Lossy Network
NA Neighbor Advertisement
NCE Neighbor Cache Entry
ND Neighbor Discovery
NS Neighbor Solicitation
ROVR Registration Ownership Verifier
RTO RPL Target Option
RA Router Advertisement
RS Router Solicitation
TID Transaction ID
TIO Transit Information Option
3. Overview

This specification inherits from [RFC6550], [RFC8505], and [RFC9010] to provide additional capabilities for anycast and multicast. Unless specified otherwise therein, the behavior of the 6LBR that acts as RPL Root, of the intermediate routers down the RPL graph, of the 6LR that act as access routers and of the 6LNs that are the RPL-unaware destinations, is the same as for unicast. In particular, forwarding a packet happens as specified in section 11 of [RFC6550], including loop avoidance and detection, though in the case of multicast multiple copies might be generated.

[RFC8505] is a pre-requisite to this specification. A node that implements this MUST also implement [RFC8505]. This specification does not introduce a new option; it modifies existing options and updates the associated behaviors to enable the Registration for Multicast Addresses as an extension to [RFC8505].

This specification also extends [RFC6550] and [RFC9010] in the case of a route-over multilink subnet based on the RPL routing protocol, to add multicast ingress replication in Non-Storing Mode and anycast support in both Storing and Non-Storing modes. A 6LR that implements the RPL extensions specified therein MUST also implement [RFC9010].

Figure 1 illustrates the classical situation of an LLN as a single IPv6 Subnet, with a 6LoWPAN Border Router (6LBR) that acts as Root for RPL operations and maintains a registry of the active registrations as an abstract data structure called an Address Registrar for 6LoWPAN ND.

The LLN may be a hub-and-spoke access link such as (Low-Power) Wi-Fi [IEEE Std 802.11] and Bluetooth (Low Energy) [IEEE Std 802.15.1], or a Route-Over LLN such as the Wi-SUN and 6TiSCH meshes [Wi-SUN] that leverages 6LoWPAN [RFC4919][RFC6282] and RPL [RFC6550] over [IEEE Std 802.15.4].
A leaf acting as a 6LN registers its unicast and anycast addresses a RPL router acting as a 6LR, using a layer-2 unicast NS message with an EARO as specified in [RFC8505]. The registration state is periodically renewed by the Registering Node, before the lifetime indicated in the EARO expires. As for unicast IPv6 addresses, the 6LR uses an EDAR/EDAC exchange with the 6LBR to notify the 6LBR of the presence of the listeners.

This specification updates the EARO with two new flags, the A flag for Anycast, and the M flag for Multicast, as detailed in Section 6.1. The existing R flag that requests reachability for the registered address gets new behavior. With this extension the 6LNs can now subscribe to the multicast addresses they listen to, using a new M flag in the EARO to signal that the registration is for a multicast address. Multiple 6LN may subscribe to the same multicast address to the same 6LR. Note the use of the term "subscribe": using the EARO registration mechanism, a node registers the unicast addresses that it owns, but subscribes to the multicast addresses that it listens to.

With this specification, the 6LNs can also register the anycast addresses they accept, using a new A flag in the EARO to signal that the registration is for an anycast address. As for multicast, multiple 6LN may register the same anycast address to the same 6LR.
If the R flag is set in the registration of one or more 6LNs for the same address, the 6LR injects the anycast addresses and multicast addresses of a scope larger than link-scope in RPL, based on the longest registration lifetime across the active registrations for the address.

In the RPL "Storing Mode of Operation with multicast support", the DAO messages for the multicast address percolate along the RPL preferred parent tree and mark a subtree that becomes the multicast tree for that multicast address, with 6LNs that subscribed to the address as the leaves. As prescribed in section 12 of [RFC6550], the 6LR forwards a multicast packet as an individual unicast MAC frame to each peer along the multicast tree, excepting to the node it received the packet from.

In the new RPL "Non-Storing Mode of Operation with multicast support" that is introduced here, the DAO messages announce the multicast addresses as Targets though never as Transit. The multicast distribution is an ingress replication whereby the Root encapsulates the multicast packets to all the 6LRs that are transit for the multicast address, using the same source-routing header as for unicast targets attached to the respective 6LRs.

Broadcasting is typically unreliable in LLNs (no ack) and forces a listener to remain awake, so it generally discouraged. The expectation is thus that in either mode, the 6LRs deliver the multicast packets as individual unicast MAC frames to each of the 6LNs that subscribed to the multicast address.

With this specification, anycast addresses can be injected in RPL in both Storing and Non-Storing modes. In Storing Mode the RPL router accepts DAO from multiple children for the same anycast address, but only forwards a packet to one of the children. In Non-Storing Mode, the Root maintains the list of all the RPL nodes that announced the anycast address as Target, but forwards a given packet to only one of them.

For backward compatibility, this specification allows to build a single DODAG signaled as MOP 1, that conveys anycast, unicast and multicast packets using the same source routing mechanism, more in Section 10.

It is also possible to leverage this specification between the 6LN and the 6LR for the registration of unicast, anycast and multicast IPv6 addresses in networks that are not necessarily LLNs, and/or where the routing protocol between the 6LR and above is not necessarily RPL. In that case, the distribution of packets between the 6LR and the 6LNs may effectively rely on a broadcast or multicast
support at the lower layer, e.g., using this specification as a replacement to MLD in an Ethernet bridged domain and still using either plain MAC-layer broadcast or snooping this protocol to control the flooding. It may also rely on overlay services to optimize the impact of Broadcast, Unknown and Multicast (BUM) over a fabric, e.g. registering with [I-D.thubert-bess-secure-evpn-mac-signaling] and forwarding with [I-D.ietf-bess-evpn-optimized-ir].

For instance, it is possible to operate a RPL Instance in the new "Non-Storing Mode of Operation with multicast support" (while possibly signaling a MOP of 1) and use "Multicast Protocol for Low-Power and Lossy Networks (MPL)" [RFC7731] for the multicast operation. MPL floods the DODAG with the multicast messages independently of the RPL DODAG topologies. Two variations are possible:

* In one possible variation, all the 6LNs set the R flag in the EARO for a multicast target, upon which the 6LRs send a unicast DAO message to the Root; the Root filters out the multicast messages for which there is no listener and only floods when there is.

* In a simpler variation, the 6LNs do not set the R flag and the Root floods all the multicast packets over the whole DODAG. Using configuration, it is also possible to control the behavior of the 6LR to ignore the R flag and either always or never send the DAO message, and/or to control the Root and specify which groups it should flood or not flood.

Note that if the configuration instructs the 6LR not to send the DAO, then MPL can really by used in conjunction with RPL Storing Mode as well.

4. Extending RFC 7400

This specification defines a new capability bit for use in the 6CIO as defined by "6LoWPAN-GHC: Generic Header Compression for IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs)" [RFC7400] and extended in [RFC8505] for use in IPv6 ND messages.

The new "Registration for xcast Address Supported" (X) flag indicates to the 6LN that the 6LR accepts unicast, multicast, and anycast address registrations as specified in this document and will ensure that packets for the Registered Address will be routed to the 6LNs that registered with the R flag set appropriately.

Figure 2 illustrates the X flag in its suggested position (8, counting 0 to 15 in network order in the 16-bit array), to be confirmed by IANA.
5. Updating RFC 6550

5.1. Updating MOP 3

RPL supports multicast operations in the "Storing Mode of Operation with multicast support" (MOP 3) which provides source-independent multicast routing in RPL, as prescribed in section 12 of [RFC6550]. MOP 3 is a storing Mode of Operation. This operation builds a multicast tree within the RPL DODAG for each multicast address. This specification provides additional details for the MOP 3 operation.

The expectation in MOP 3 is that the unicast traffic also follows the Storing Mode of Operation. But this is rarely the case in LLN deployments of RPL where the "Non-Storing Mode of Operation" (MOP 1) is the norm. Though it is preferred to build separate RPL Instances, one in MOP 1 and one in MOP 3, this specification allows to hybrid the Storing Mode for multicast and Non-Storing Mode for unicast in the same RPL Instance, more in Section 10.

Though it was implicit in [RFC6550], this specification clarifies that the freshness comparison based on the TID field is ignored for RPL multicast operations. A RPL router maintains a remaining Path Lifetime for each DAO that it receives for a multicast target, and sends its own DAO for that target with the longest remaining lifetime across its listening children.

5.2. New Non-Storing Multicast MOP

This specification adds a "Non-Storing Mode of Operation with ingress replication multicast support" (MOP to be assigned by IANA) whereby the non-storing Mode DAO to the Root may advertise a multicast address in the RPL Target Option (RTO), whereas the Transit Information Option (TIO) cannot.
In that mode, the RPL Root performs an ingress replication (IR) operation on the multicast packets, meaning that it transmits one copy of each multicast packet to each 6LR that is a transit for the multicast target, using the same source routing header and encapsulation as it would for a unicast packet for a RPL Unaware Leaf (RUL) attached to that 6LR.

For the intermediate routers, the packet appears as any source routed unicast packet. The difference shows only at the 6LR, that terminates the source routed path and forwards the multicast packet to all 6LNs that registered for the multicast address.

For a packet that is generated by the Root, this means that the Root builds a source routing header as shown in section 8.1.3 of [RFC9008], but for which the last and only the last address is multicast. For a packet that is not generated by the Root, the Root encapsulates the multicast packet as per section 8.2.4 of [RFC9008]. In that case, the outer header is purely unicast, and the encapsulated packet is purely multicast.

As with MOP 3, the freshness comparison based on the TID field is ignored for RPL MOP 5 multicast operations. The Root maintains a remaining Path Lifetime for each DAO that it receives, and the 6LRs generate the DAO for multicast addresses with the longest remaining lifetime across its registered 6LNs.

The route disappears when the associated path lifetime in the transit option times out, but the procedure to remove a unicast route based on TID cannot apply to multicast and anycast.

For this new mode as well, this specification allows to enable the operation in a MOP 1 brown field, more in Section 10.

5.3. RPL Anycast Operation

With multicast, the address has a recognizable format, and a multicast packet is to be delivered to all the active subscribers. In contrast, the format of an anycast address is not distinguishable from that of unicast. A legacy node may issue a DAO message without setting the A flag, the unicast behavior may apply to anycast traffic in a subDAGs. That message will be undistinguishable from a unicast advertisement and the anycast behavior in the dataplane can only happen if all the nodes that advertise the same anycast address are synchronized with the same TID. That way, the multiple paths can remain in the RPL DODAG.
With the A flag, this specification alleviates the issue of synchronizing the TIDs, and as for multicast, the freshness comparison based on the TID field is ignored. A target is routed as anycast by a parent (or the Root) that received at least one DAO message for that target with the A flag set to 1.

As opposed to multicast, the anycast operation described therein applies to both addresses and prefixes, and the A flag can be set for both. An external destination (address or prefix) that may be injected as a RPL target from multiple border routers SHOULD be injected as anycast in RPL to enable load balancing. A mobile target that is multihomed SHOULD in contrast be advertised as unicast over the multiple interfaces to favor the TID comparison and vs. the multipath load balancing.

For either multicast and anycast, there can be multiple registrations from multiple parties, each using a different value of the ROVR field that identifies the individual registration. The 6LR MUST maintain a registration state per value of the ROVR per multicast or anycast address, but inject the route into RPL only once for each address, and in the case of a multicast address, only if its scope is larger than link-scope (3 or more). Since the registrations are considered separate, the check on the TID that acts as registration sequence only applies to the registration with the same ROVR.

The 6LRs that inject multicast and anycast routes into RPL may not be synchronized to advertise same value of the Path Sequence in the RPL TIO. It results that the value the Path Sequence is irrelevant when the target is anycast or multicast, and that it MUST be ignored.

Like the 6LR, a RPL router in Storing Mode propagates the route to its parent(s) in DAO messages once and only once for each address, but it MUST retain a routing table entry for each of the children that advertised the address.

When forwarding multicast packets down the DODAG, the RPL router copies all the children that advertised the address in their DAO messages. In contrast, when forwarding anycast packets down the DODAG, the RPL router MUST copy one and only one of the children that advertised the address in their DAO messages, and forward to one parent if there is no such child.
5.4. New RPL Target Option Flags

[RFC6550] recognizes a multicast address by its format (as specified in section 2.7 of [RFC4291]) and applies the specified multicast operation if the address is recognized as multicast. This specification updates [RFC6550] to add the M and A flags to the RTO to indicate that the target address is to be processed as multicast or anycast, respectively.

An RTO that has the M flag set to 1 is called a multicast RTO. An RTO that has the A flag set to 1 is called an anycast RTO. An RTO that has both the A and the M flags set to 0 is called an unicast RTO. With this specification, the M and A flags are mutually exclusive and MUST NOT be both set to 1. The capability to set both flags is reserved and an RTO that is received with both flags set MUST be ignored.

The suggested position for the A and M flags are 2 and 3 counting from 0 to 7 in network order as shown in Figure 3, based on figure 4 of [RFC9010] which defines the flags in position 0 and 1:

```
0                   1                   2                   3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   Type = 0x05 | Option Length |F|X|A|M|ROVRsz | Prefix Length |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|                Target Prefix (Variable Length)                |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|                Registration Ownership Verifier (ROVR)           |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
Figure 3: Format of the RPL Target Option
```

6. Updating RFC 8505

6.1. New EARO flag

Section 4.1 of [RFC8505] defines the EARO as an extension to the ARO option defined in [RFC6775].
This specification adds a new M flag to the EARO flags field to signal that the Registered Address is a multicast address. When both the M and the R flags are set, the 6LR that conforms to this specification joins the multicast stream, e.g., by injecting the address in the RPL multicast support which is extended in this specification for Non-Storing Mode.

This specification adds a new A flag to the EARO flags field to signal that the Registered Address is an anycast address. When both the A and the R flags are set, the 6LR that conforms to this specification injects the anycast address in the RPL anycast support that is introduced in this specification for both Storing and Non-Storing Modes.

Figure 4 illustrates the A and M flags in their suggested positions (2 and 3, respectively, counting 0 to 7 in network order in the 8-bit array), to be confirmed by IANA.

```
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Type      |     Length    |    Status     |    Opaque     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|Rsv|A|M| I |R|T|     TID       |     Registration Lifetime     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|                                                               |
|                                                               |
| Registration Ownership Verifier                             ...
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 4: EARO Option Format

New and updated Option Fields:

Rsv 2-bit field: reserved, MUST be set to 0 and ignored by the receiver

A 1-bit flag: "Registration for Anycast Address"

M 1-bit flag: "Registration for Multicast Address"

6.2. New EDAR Message Flag field

Section 4 of [RFC6775] provides the same format for DAR and DAC messages but the status field is only used in DAC message and has to set to zero in DAC messages. [RFC8505] extends the DAC message as an EDAC but does not change the status field in the EDAR.
This specification repurposes the status field in the EDAR and a Flags field. It adds a new M flag to the EDAR flags field to signal that the Registered Address is a multicast address and a new A flag to signal that the Registered Address is an anycast address. As for EARO, the flags are mutually exclusive.

Figure 5 illustrates the A and M flags in their suggested positions (0 and 1, respectively, counting 0 to 7 in network order in the 8-bit array), to be confirmed by IANA.

```
0                   1                   2                   3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Type      |CodePfx|CodeSfx|          Checksum             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|A|M| Reserved  |     TID       |     Registration Lifetime     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|                                                               |
|                                                               |
|Registration Ownership Verifier (ROVR)                        |
|                                                               |
|                                                               |
+                                                               +
|                                                               +
|Registered Address                                            |
|                                                               +
|                                                               +
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 5: Extended Duplicate Address Message Format

New and updated Option Fields:

Reserved  6-bit field: reserved, MUST be set to 0 and ignored by the receiver

A  1-bit flag: "Registration for Anycast Address"

M  1-bit flag: "Registration for Multicast Address"

6.3. Registering Extensions

With [RFC8505]:

Thubert                Expires 24 December 2022                [Page 15]
* A router that expects to reboot may send a final RA message, upon which nodes should register elsewhere or reregister to this router upon reboot. In all other cases, a node reboot is silent. When the node comes back to life, existing registration state might be lost if it was not persisted, e.g., in persistent memory.

* Only unicast addresses can be registered.

* The 6LN must register all its ULA and GUA with a NS(EARO).

* The 6LN may set the R flag in the EARO to obtain return reachability services by the 6LR, e.g., through ND proxy operations, or by injecting the route in a route-over subnet.

* The 6LR maintains a registration state per Registered Address, including an NCE with the Link Layer Address (LLA) of the Registered Node (the 6LN here).

This specification adds the following behavior:

* A new ARO Status is introduced to indicate a "Registration Refresh Request" (see Table 7).

This status is used in asynchronous NA(EARO) messages to indicate to peer 6LNs that they are requested to reregister all addresses that were previously registered to the originating node. The NA message may be sent to a unicast or a multicast link-scope address and should be contained within the L2 range where nodes may effectively register to this, e.g., a radio broadcast domain.

A device that wishes to refresh its state, e.g., upon reboot if it may have lost some registration state, may send an asynchronous NA(EARO) with this new status value. That asynchronous NA(ARO) SHOULD be sent to the all-nodes link-scope multicast address (FF02::1) and Target MUST be set to the link local address that was exposed previously by this node to accept registrations, and the TID MUST be set to 0.

In an unreliable environment, the multicast NA(EARO) message may be resent in a fast sequence, in which case the TID must be incremented each time. A 6LN that has recently processed the NA(ARO) ignores the NA(EARO) with a newer TID received within the duration of the fast sequence. That duration depends on the environment and has to be configured. By default, it is of 10 seconds.
A new IPv6 ND Node Uptime option (NUO) is introduced to be placed in IPv6 ND messages. The NUO carries a Node State Sequence Information (NSSI) and a Node Uptime. See Section 9 for the option details.

A node that receives the NUO checks whether it is indicative of a loss of state, such as an address registration, in the sender. If so, it may attempt to reform the state, e.g., by re-registering an address. A loss of state is inferred if the NSSI has changed since last sight, or the Node Uptime is less than the time since the state was installed.

Registration for multicast and anycast addresses is now supported. New flags are added to the EARO to signal when the registered address is anycast or multicast.

The Status field in the EDAR message that was reserved and not used in RFC 8505 is repurposed to transport the flags to signal multicast and anycast.

The 6LN MUST also register all the IPv6 multicast addresses that it listens to but the all-nodes link-scope multicast address FF02::1 [RFC4291] which is implicitly registered, and it MUST set the M flag in the EARO for those addresses.

The 6LN MAY set the R flag in the EARO to obtain the delivery of the multicast packets by the 6LR, e.g., by MLD proxy operations, or by injecting the address in a route-over subnet or in the Protocol Independent Multicast [RFC7761] protocol.

The 6LN MUST also register all the IPv6 anycast addresses that it supports and it MUST set the A flag in the EARO for those addresses.

The 6LR and the 6LBR are extended to accept more than one registration for the same address when it is anycast or multicast, since multiple 6LNs may subscribe to the same address of these types. In both cases, the Registration Ownership Verifier (ROVR) in the EARO identifies uniquely a registration within the namespace of the Registered Address.

The 6LR MUST also consider that all the nodes that registered an address to it (as known by the SLLAO) also registered to the all nodes link-scope multicast address FF02::1 [RFC4291].

The 6LR MUST maintain a registration state per tuple (IPv6 address, ROVR) for both anycast and multicast types of address. It SHOULD notify the 6LBR with an EDAR message, unless it
determined that the 6LBR is legacy and does not support this specification. In turn, the 6LBR MUST maintain a registration state per tuple (IPv6 address, ROVR) for both anycast and multicast types of address.

7. Updating RFC 9010

With [RFC9010]:

* The 6LR injects only unicast routes in RPL

* Upon a registration with the R flag set to 1 in the EARO, the 6LR injects the address in the RPL unicast support.

* Upon receiving a packet directed to a unicast address for which it has an active registration, the 6LR delivers the packet as a unicast layer-2 frame to the LLA the nodes that registered the unicast address.

This specification adds the following behavior:

* Upon a registration with the R and the M flags set to 1 in the EARO, if the scope of the multicast address is above link-scope [RFC7346], then the 6LR injects the address in the RPL multicast support and sets the M flag in the RTO.

* Upon a registration with the R and the A flags set to 1 in the EARO, the 6LR injects the address in the new RPL anycast support and sets the A flag in the RTO.

* Upon receiving a packet directed to a multicast address for which it has at least one registration, the 6LR delivers a copy of the packet as a unicast layer-2 frame to the LLA of each of the nodes that registered to that multicast address.

* Upon receiving a packet directed to a multicast address for which it has at least one registration, the 6LR delivers a copy of the packet as a unicast layer-2 frame to the LLA of exactly one of the nodes that registered to that multicast address.

8. Leveraging RFC 8928

Address-Protected Neighbor Discovery for Low-Power and Lossy Networks [RFC8928] was defined to protect the ownership of unicast IPv6 addresses that are registered with [RFC8505].
With [RFC8928], it is possible for a node to autoconfigure a pair of public and private keys and use them to sign the registration of addresses that are either autoconfigured or obtained through other methods.

The first hop router (the 6LR) may then validate a registration and perform source address validation on packets coming from the sender node (the 6LN).

Anycast and multicast addresses are not owned by one node. Multiple nodes may subscribe to the same address. Also, anycast and multicast addresses are not used to source traffic. In that context, the method specified in [RFC8928] cannot be used with autoconfigured keypairs to protect a single ownership.

For an anycast or a multicast address, it is still possible to leverage [RFC8928] to enforce the right to subscribe. A keypair MUST be associated with the address before it is deployed, and a ROVR MUST be generated from that keypair as specified in [RFC8928]. The address and the ROVR MUST then be installed in the 6LBR so it can recognize the address and compare the ROVR on the first registration.

The keypair MUST then be provisioned in each node that needs to subscribe to the anycast or multicast address, so the node can follow the steps in [RFC8928] to register the address.

9. Node Uptime Option

This specification introduces a new option that characterizes the uptime of the sender. The option may be used by routers in RA messages and by any node in NA, NA, and RS messages. It is used by the receiver to infer whether some state synchronization might be lost, e.g., due to reboot.

```
<table>
<thead>
<tr>
<th>0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>
```

Figure 6: Node Uptime Option Format

Type  To be assigned by IANA, see Table 8
Length  1
S  1-bit flag, set to indicate that the device is low-power and may sleep.

flags  5-bit, reserved. MUST be set to 0 by the sender and ignored by the receiver.

NSSI  10-bits unsigned integer: The Node State Sequence Information

Uptime Exponent  6-bits unsigned integer: The 2-exponent of the uptime unit

Uptime Mantissa  10-bits unsigned integer: The mantissa of the uptime value

The Node Uptime indicates how long the sender has been continuously up and running without loss of state. It is expressed by the Uptime Mantissa in units of 2 at the power of the Uptime Exponent milliseconds.

The initial value and the steps of the Uptime Exponent are chosen freely by the implementation, but the value MUST NOT decrease over time. This means that 2 expressions of time from the same node can be compared by aggregating the Exponent + Mantissa Uptime fields and considering the aggregate globally as a 16-bits unsigned integer.

<table>
<thead>
<tr>
<th>Mantissa</th>
<th>Exponent</th>
<th>Resolution</th>
<th>Uptime</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>1ms</td>
<td>1ms</td>
</tr>
<tr>
<td>5</td>
<td>10</td>
<td>1s</td>
<td>5 seconds</td>
</tr>
<tr>
<td>2</td>
<td>15</td>
<td>30s</td>
<td>1mn</td>
</tr>
<tr>
<td>2</td>
<td>21</td>
<td>33mn</td>
<td>1 hour</td>
</tr>
</tbody>
</table>

Table 1: Node Uptime Rough Values

The NSSI SHOULD be stored by this node in persistent memory by the sender and incremented when it reboots and lost state. When persisting is not possible, then the NSSI is randomly generated upon a loss of state. Any change in the value of the NSSI from a node is an indication that the node lost state and that the needful state should be reinstalled, e.g., addresses registered to that node should be registered again with a minimal temporization to avoid collisions.
10. Deployment considerations

With this specification, a RPL DODAG forms a realm, and multiple RPL DODAGs may federated in a single RPL Instance administratively. This means that a multicast address that needs to span a RPL DODAG MUST use a scope of Realm-Local whereas a multicast address that needs to span a RPL Instance MUST use a scope of Admin-Local as discussed in section 3 of "IPv6 Multicast Address Scopes" [RFC7346].

"IPv6 Addressing of IPv4/IPv6 Translators" [RFC6052] enables to embed IPv4 addresses in IPv6 addresses. The Root of a DODAG may leverage that technique to translate IPv4 traffic in IPv6 and route along the RPL domain. When encapsulating an packet with an IPv4 multicast Destination Address, it MUST use a multicast address with the appropriate scope, Realm-Local or Admin-Local.

"Unicast-Prefix-based IPv6 Multicast Addresses" [RFC3306] enables to form $2^{32}$ multicast addresses from a single /64 prefix. If an IPv6 prefix is associated to an Instance or a RPL DODAG, this provides a namespace that can be used in any desired fashion. It is for instance possible for a standard defining organization to form its own registry and allocate 32-bit values from that namespace to network functions or device types. When used within a RPL deployment that is associated with a /64 prefix the IPv6 multicast addresses can be automatically derived from the prefix and the 32-bit value for either a Realm-Local or an Admin-Local multicast address as needed in the configuration.

In a "green field" deployment where all nodes support this specification, it is possible to deploy a single RPL Instance using a multicast MOP for unicast, multicast and anycast addresses.

In a "brown field" where legacy devices that do not support this specification co-exist with upgraded devices, it is RECOMMENDED to deploy one RPL Instance in any Mode of Operation (typically MOP 1) for unicast that legacy nodes can join, and a separate RPL Instance dedicated to multicast and anycast operations using a multicast MOP.

To deploy a Storing Mode multicast operation using MOP 3 in a RPL domain, it is required that there is enough density of RPL routers that support MOP 3 to build a DODAG that covers all the potential listeners and include the spanning multicast trees that are needed to distribute the multicast flows. This might not be the case when extending the capabilities of an existing network.
In the case of the new Non-Storing multicast MOP, arguably the new support is only needed at the 6LRs that will accept multicast listeners. It is still required that each listener can reach at least one such 6LR, so the upgraded 6LRs must be deployed to cover all the 6LN that need multicast services.

Using separate RPL Instances for in the one hand unicast traffic and in the other hand anycast and multicast traffic allows to use different objective function, one favoring the link quality up for unicast collection and one favoring downwards link quality for multicast distribution.

But this might be impractical in some use cases where the signaling and the state to be installed in the devices are very constrained, the upgraded devices are too sparse, or the devices do not support more multiple instances.

When using a single RPL Instance, MOP 3 expects the Storing Mode of Operation for both unicast and multicast, which is an issue in constrained networks that typically use MOP 1 for unicast. This specification allows a mixed mode that is signaled as MOP 1 in the DIO messages for backward compatibility, where limited multicast and/or anycast is available, under the following conditions:

* There MUST be enough density of 6LRs that support the mixed mode to cover all the 6LNs that require multicast or anycast services. In Storing Mode, there MUST be enough density of 6LR that support the mixed mode to also form a DODAG to the Root.

* The RPL routers that support the mixed mode and are configured to operate in accordance with the desired operation in the network.

* The MOP signaled in the RPL DODAG Information Object (DIO) messages is MOP 1 to enable the legacy nodes to operate as leaves.

* The support of multicast and/or anycast in the RPL Instance SHOULD be signaled by the 6LRs to the 6LN using a 6CIO, see Section 4.

* Alternatively, the support of multicast in the RPL domain can be globally known by other means such as configuration or external information such as support of a version of an industry standard that mandates it. In that case, all the routers MUST support the mixed mode.
11. Security Considerations

This specification extends [RFC8505], and the security section of that document also applies to this document. In particular, the link layer SHOULD be sufficiently protected to prevent rogue access.

Section 8 leverages [RFC8928] to prevent an unwanted subscriber to register for an anycast of a multicast address. This mechanism comes with a keypair that is shared between all subscribers. A shared key is prone to be stolen and the level of protection can only go down with time.

It is possible to update the keys associated to an address in all the 6LNs, but the flow is not clearly documented and may not complete in due time for all nodes in LLN use cases. It may be simpler to install a new address with new keys over a period of time, and switch the traffic to that address when the migration is complete.

12. Backward Compatibility

A legacy 6LN will not register multicast addresses and the service will be the same when the network is upgraded. A legacy 6LR will not set the M flag in the 6CIO and an upgraded 6LN will not register multicast addresses.

Upon an EDAR message, a legacy 6LBR may not realize that the address being registered is anycast or multicast, and return that it is duplicate in the EDAC status. The 6LR MUST ignore a duplicate status in the EDAR for anycast and multicast addresses.

As detailed in Section 10, it is possible to add multicast on an existing MOP 1 deployment.

The combination of a multicast address and the M flag set to 0 in an RTO in a MOP 3 RPL Instance is understood by the receiver that supports this specification (the parent) as an indication that the sender (child) does not support this specification, but the RTO is accepted and processed as if the M flag was set for backward compatibility.

When the DODAG is operated in MOP 3, a legacy node will not set the M flag and still expect multicast service as specified in section 12 of [RFC6550]. In MOP 3 an RTO that is received with a target that is multicast and the M bit set to 0 MUST be considered as multicast and MUST be processed as if the M flag is set.
13. IANA Considerations

Note to RFC Editor, to be removed: please replace "This RFC" throughout this document by the RFC number for this specification once it is allocated. Also, the I Field is defined in [RFC9010] but is missing from the subregistry, so the bit positions must be added for completeness.

IANA is requested to make changes under the "Internet Control Message Protocol version 6 (ICMPv6) Parameters" [IANA.ICMP] and the "Routing Protocol for Low Power and Lossy Networks (RPL)" [IANA.RPL] registries, as follows:

13.1. New EDAR Message Flags Subregistry

IANA is requested to create a new "EDAR Message Flags" subregistry of the "Internet Control Message Protocol version 6 (ICMPv6) Parameters" registry as indicated in Table 2:

<table>
<thead>
<tr>
<th>Bit Number</th>
<th>Meaning</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 (suggested)</td>
<td>A flag: Registered</td>
<td>This RFC</td>
</tr>
<tr>
<td></td>
<td>Address is Anycast</td>
<td></td>
</tr>
<tr>
<td>1 (suggested)</td>
<td>M flag: Registered</td>
<td>This RFC</td>
</tr>
<tr>
<td></td>
<td>Address is Multicast</td>
<td></td>
</tr>
</tbody>
</table>

Table 2: EDAR Message flags

13.2. New EARO flags

IANA is requested to make additions to the "Address Registration Option Flags" [IANA.ICMP.ARO.FLG] of the "Internet Control Message Protocol version 6 (ICMPv6) Parameters" registry as indicated in Table 3:
<table>
<thead>
<tr>
<th>ARO flag</th>
<th>Meaning</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>2 (suggested)</td>
<td>A flag: Registration for Anycast Address</td>
<td>This RFC</td>
</tr>
<tr>
<td>3 (suggested)</td>
<td>M flag: Registration for Multicast Address</td>
<td>This RFC</td>
</tr>
<tr>
<td>4 and 5</td>
<td>&quot;I&quot; Field</td>
<td>RFC 8505</td>
</tr>
</tbody>
</table>

Table 3: New ARO flags

13.3. New RTO flags

IANA is requested to make additions to the "RPL Target Option Flags" [IANA.RPL.RTO.FLG] subregistry of the "Routing Protocol for Low Power and Lossy Networks (RPL)" registry as indicated in Table 4:

<table>
<thead>
<tr>
<th>Bit Number</th>
<th>Meaning</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>2 (suggested)</td>
<td>A flag: Registered Address is Anycast</td>
<td>This RFC</td>
</tr>
<tr>
<td>3 (suggested)</td>
<td>M flag: Registered Address is Multicast</td>
<td>This RFC</td>
</tr>
</tbody>
</table>

Table 4: New RTO flags

13.4. New RPL Mode of Operation

IANA is requested to make an addition to the "Mode of Operation" [IANA.RPL.MOP] subregistry of the "Routing Protocol for Low Power and Lossy Networks (RPL)" registry as indicated in Table 5:

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>5 (suggested)</td>
<td>Non-Storing Mode of Operation with ingress replication multicast support</td>
<td>This RFC</td>
</tr>
</tbody>
</table>

Table 5: New RPL Mode of Operation
13.5. New 6LoWPAN Capability Bits

IANA is requested to make an addition to the "6LoWPAN Capability Bits" [IANA.ICMP.6CIO] subregistry subregistry of the "Internet Control Message Protocol version 6 (ICMPv6) Parameters" registry as indicated in Table 6:

<table>
<thead>
<tr>
<th>Capability Bit</th>
<th>Meaning</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>8 (suggested)</td>
<td>X flag: Registration for Unicast, Multicast, and Anycast Addresses Supported</td>
<td>This RFC</td>
</tr>
</tbody>
</table>

Table 6: New 6LoWPAN Capability Bits

13.6. New Address Registration Option Status Values

IANA has made additions to the "Address Registration Option Status Values" subregistry under the "Internet Control Message Protocol version 6 (ICMPv6) Parameters" registry, as follows:

<table>
<thead>
<tr>
<th>Value (suggested)</th>
<th>Description</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>11 (suggested)</td>
<td>Registration Refresh Request</td>
<td>This RFC</td>
</tr>
</tbody>
</table>

Table 7: New Address Registration Option Status Values

13.7. New IPv6 Neighbor Discovery Option

IANA has made additions to the "IPv6 Neighbor Discovery Option Formats" subregistry under the "Internet Control Message Protocol version 6 (ICMPv6) Parameters" registry, as follows:

<table>
<thead>
<tr>
<th>Value (suggested)</th>
<th>Description</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>42 (suggested)</td>
<td>Node Uptime Option</td>
<td>This RFC</td>
</tr>
</tbody>
</table>

Table 8: New IPv6 Neighbor Discovery Option"
14. Acknowledgments

15. Normative References


[IANA.ICMP.ARO.FLG] IANA, "IANA Sub-Registry for the ARO Flags", IANA, https://www.iana.org/assignments/icmpv6-parameters/icmpv6-parameters.xhtml#icmpv6-address-registration-option-flags.


16. Informative References


[I-D.ietf-bess-evpn-optimized-ir]


[IEEE Std 802.15.4] IEEE standard for Information Technology, "IEEE Std 802.15.4, Part. 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks".


[IEEE Std 802.15.1] IEEE standard for Information Technology, "IEEE Standard for Information Technology - Telecommunications and Information Exchange Between Systems - Local and Metropolitan Area Networks - Specific Requirements. - Part 15.1: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Wireless Personal Area Networks (WPANs)".

Author’s Address

Thubert Expires 24 December 2022 [Page 30]
IPv6 over Constrained Node Networks (6lo) Applicability & Use cases
draft-ietf-6lo-use-cases-13

Abstract

This document describes the applicability of IPv6 over constrained node networks (6lo) and provides practical deployment examples. In addition to IEEE Std 802.15.4, various link layer technologies such as ITU-T G.9959 (Z-Wave), Bluetooth Low Energy, DECT-ULE, MS/TP, NFC, and PLC are used as examples. The document targets an audience who would like to understand and evaluate running end-to-end IPv6 over the constrained node networks for local or Internet connectivity.

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 12, 2023.

Copyright Notice

Copyright (c) 2022 IETF Trust and the persons identified as the document authors. All rights reserved.
1. Introduction

Running IPv6 on constrained node networks presents challenges, due to the characteristics of these networks such as small packet size, low power, low bandwidth, low cost, and large number of devices, among others [RFC4919][RFC7228]. For example, many IEEE Std 802.15.4 variants [IEEE802154] exhibit a frame size of 127 octets, whereas
IPv6 requires its underlying layer to support an MTU of 1280 bytes. Furthermore, those IEEE Std 802.15.4 variants do not offer fragmentation and reassembly functionality. (It is noted that IEEE Std 802.15.9-2021 provides a multiplexing and fragmentation layer for the IEEE Std 802.15.4[IEEE802159].) Therefore, an appropriate adaptation layer supporting fragmentation and reassembly must be provided below IPv6. Also, the limited IEEE Std 802.15.4 frame size and low energy consumption requirements motivate the need for packet header compression. The IETF IPv6 over Low-Power WPAN (6LoWPAN) working group published a suite of specifications that provide an adaptation layer to support IPv6 over IEEE Std 802.15.4 comprising the following functionality:

- Fragmentation and reassembly, address autoconfiguration, and a frame format [RFC4944],
- IPv6 (and UDP) header compression [RFC6282],
- Neighbor Discovery Optimization for 6LoWPAN [RFC6775][RFC8505].

As Internet of Things (IoT) services become more popular, the IETF 6lo working group [IETF_6lo] has defined adaptation layer functionality to support IPv6 over various link layer technologies other than IEEE Std 802.15.4, such as Bluetooth Low Energy (Bluetooth LE), ITU-T G.9959 (Z-Wave), Digital Enhanced Cordless Telecommunications - Ultra Low Energy (DECT-ULE), Master-Slave/Token Passing (MS/TP), Near Field Communication (NFC), and Power Line Communication (PLC). The 6lo adaptation layers use a variation of the 6LoWPAN stack applied to each particular link layer technology.

The 6LoWPAN working group produced the document entitled "Design and Application Spaces for 6LoWPANs" [RFC6568], which describes potential application scenarios and use cases for low-power wireless personal area networks. The present document aims to provide guidance to an audience who are new to the IPv6 over constrained node networks (6lo) concept and want to assess its application to the constrained node network of their interest. This 6lo applicability document describes a few sets of practical 6lo deployment scenarios and use cases examples. In addition, it considers various network design space dimensions such as deployment, network size, power source, connectivity, multi-hop communication, traffic pattern, security level, mobility, and QoS requirements.

This document provides the applicability and use cases of 6lo, considering the following aspects:
It covers various IoT-related wired/wireless link layer technologies providing practical information about such technologies.

It provides a general guideline on how the 6LoWPAN stack can be modified for a given L2 technology.

Various 6lo use cases and practical deployment examples are described.

2. 6lo Link layer technologies

2.1. ITU-T G.9959

The ITU-T G.9959 Recommendation [G.9959] targets low-power Wireless Personal Area Networks (WPANs), and defines physical layer and link layer functionality. Physical layers of 9.6 kbit/s, 40 kbit/s and 100 kbit/s are supported. G.9959 defines how a unique 32-bit HomeID network identifier is assigned by a network controller and how an 8-bit NodeID host identifier is allocated to each node. NodeIDs are unique within the network identified by the HomeID. The G.9959 HomeID represents an IPv6 subnet that is identified by one or more IPv6 prefixes [RFC7428]. ITU-T G.9959 can be used for smart home applications.

2.2. Bluetooth LE

Bluetooth LE was introduced in Bluetooth 4.0, enhanced in Bluetooth 4.1, and developed further in successive versions. The Bluetooth SIG has also published the Internet Protocol Support Profile (IPSP). The IPSP enables discovery of IP-enabled devices and establishment of link-layer connections for transporting IPv6 packets. IPv6 over Bluetooth LE is dependent on both Bluetooth 4.1 and IPSP 1.0 or newer.

Many devices such as mobile phones, notebooks, tablets and other handheld computing devices which support Bluetooth 4.0 or subsequent versions also support the low-energy variant of Bluetooth. Bluetooth LE is also being included in many different types of accessories that collaborate with mobile devices. An example of a use case for a Bluetooth LE accessory is a heart rate monitor that sends data via the mobile phone to a server on the Internet [RFC7668]. A typical usage of Bluetooth LE is smartphone-based interaction with constrained devices. Bluetooth LE was originally designed to enable star topology networks. However, recent Bluetooth versions support the formation of extended topologies, and IPv6 support for mesh networks of Bluetooth LE devices has been developed [RFC9159].
2.3. DECT-ULE

DECT-ULE is a low-power air interface technology that is designed to support both circuit-switched services, such as voice communication, and packet-mode data services at modest data rate.

The DECT-ULE protocol stack consists of the physical layer operating at frequencies in the dedicated 1880 - 1920 MHz frequency band depending on the region and uses a symbol rate of 1.152 Mbps. Radio bearers are allocated by use of FDMA/TDMA/TDD techniques.

In its generic network topology, DECT is defined as a cellular network technology. However, the most common configuration is a star network with a single Fixed Part (FP) defining the network with a number of Portable Parts (PP) attached. The Medium Access Control (MAC) layer supports traditional DECT as this is used for services like discovery, pairing, and security features. All these features have been reused from DECT.

The DECT-ULE device can switch to the ULE mode of operation, utilizing the new ULE MAC layer features. The DECT-ULE Data Link Control (DLC) provides multiplexing as well as segmentation and re-assembly for larger packets from layers above. The DECT-ULE layer also implements per-message authentication and encryption. The DLC layer ensures packet integrity and preserves packet order, but delivery is based on best effort.

The current DECT-ULE MAC layer standard supports low bandwidth data broadcast. However, the usage of this broadcast service has not yet been standardized for higher layers [RFC8105]. DECT-ULE can be used for smart metering in a home.

2.4. MS/TP

MS/TP is a MAC protocol for the RS-485 [TIA-485-A] physical layer and is used primarily in building automation networks.

An MS/TP device is typically based on a low-cost microcontroller with limited processing power and memory. These constraints, together with low data rates and a small MAC address space, are similar to those faced in 6LoWPAN networks. MS/TP differs significantly from 6LoWPAN in at least three respects: a) MS/TP devices are typically mains powered, b) all MS/TP devices on a segment can communicate directly so there are no hidden node or mesh routing issues, and c) the latest MS/TP specification provides support for large payloads, eliminating the need for fragmentation and reassembly below IPv6.
MS/TP is designed to enable multidrop networks over shielded twisted pair wiring. It can support network segments up to 1000 meters in length at a data rate of 115.2 kbit/s or segments up to 1200 meters in length at lower bit rates. An MS/TP interface requires only a Universal Asynchronous Receiver-Transmitter (UART), an RS-485 [TIA-485-A] transceiver with a driver that can be disabled, and a 5 ms resolution timer. The MS/TP MAC is typically implemented in software.

Because of its long-range (~1 km), MS/TP can be used to connect remote devices (such as district heating controllers) to the nearest building control infrastructure over a single link [RFC8163].

2.5. NFC

NFC technology enables simple and safe two-way interactions between electronic devices, allowing consumers to perform contactless transactions, access digital content, and connect electronic devices with a single touch. NFC complements many popular consumer-level wireless technologies, by utilizing the key elements in existing standards for contactless card technology (ISO/IEC 14443 A&B and JIS-X 6319-4).

Extending the capability of contactless card technology, NFC also enables devices to share information at a distance that is less than 10 cm with a maximum communication speed of 424 kbps. Users can share business cards, make transactions, access information from a smart poster or provide credentials for access control systems with a simple touch.

NFC’s bidirectional communication ability is suitable for establishing connections with other technologies by the simplicity of touch. In addition to the easy connection and quick transactions, simple data sharing is available [I-D.ietf-6lo-nfc]. NFC can be used for secure transfer in healthcare services.

2.6. PLC

PLC is a data transmission technique that utilizes power conductors as medium [I-D.ietf-6lo-plc]. Unlike other dedicated communication infrastructure, power conductors are widely available indoors and outdoors. Moreover, wired technologies cause less interference to the radio medium than wireless technologies and are more reliable than their wireless counterparts.

The table below shows some available open standards defining PLC.
Table 1: Some Available Open Standards in PLC

IEEE Std 1901 [IEEE1901] defines a broadband variant of PLC but it is only effective within short range. This standard addresses the requirements with high data rates such as Internet, HDTV, audio, gaming.

IEEE Std 1901.1 [IEEE1901.1] defines a medium frequency band (less than 12 MHz) broadband PLC technology for smart grid applications based on OFDM. By achieving an extended communication range with medium speeds, this standard can be applied both in indoor and outdoor scenarios, such as Advanced Metering Infrastructure (AMI), street lighting, electric vehicle charging, smart city.

IEEE Std 1901.2 [IEEE1901.2] defines a narrowband variant of PLC with lower data rate but significantly higher transmission range that could be used in an indoor or even an outdoor environment. A typical use case of PLC is smart grid.

G3-PLC [G3-PLC] is a narrowband PLC technology that is based on the ITU-T G.9903 Recommendation [G.9903]. The ITU-T G.9903 Recommendation contains the physical layer and data link layer specification for the G3-PLC narrowband OFDM power line communication transceivers, for communications via alternating current and direct current electric power lines over frequency bands below 500 kHz.

2.7. Comparison between 6lo link layer technologies

In the above subsections, various 6lo link layer technologies are described. The following table shows the dominant parameters of each use case corresponding to the 6lo link layer technology.
### Table 2: Comparison between 6lo link layer technologies

<table>
<thead>
<tr>
<th>Usage</th>
<th>Z-Wave</th>
<th>BLE</th>
<th>DECT-ULE</th>
<th>MS/TP</th>
<th>NFC</th>
<th>PLC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Home Automation</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Interact w/ Smart Phone</td>
<td>Home Automation</td>
<td></td>
<td></td>
<td>Building Automation</td>
<td>Health-care Service</td>
<td>Smart Grid</td>
</tr>
<tr>
<td>Meter Reading</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Star &amp; No mesh</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MS/TP No mesh</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>P2P L2-mesh</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Star Tree Mesh</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Topology &amp; Subnet</th>
<th>L2-mesh or L3-mesh</th>
<th>Star &amp; Mesh</th>
<th>Star &amp; No mesh</th>
<th>MS/TP No mesh</th>
<th>P2P L2-mesh</th>
<th>Star Tree Mesh</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Mobility Requirement</th>
<th>No</th>
<th>Low</th>
<th>No</th>
<th>No</th>
<th>Moderate</th>
<th>No</th>
</tr>
</thead>
<tbody>
<tr>
<td>Buffering Requirement</td>
<td>Low</td>
<td>Low</td>
<td>Low</td>
<td>Low</td>
<td>Low</td>
<td>Low</td>
</tr>
<tr>
<td>Latency, QoS Requirement</td>
<td>High</td>
<td>Low</td>
<td>Low</td>
<td>High</td>
<td>High</td>
<td>Low</td>
</tr>
<tr>
<td>Transmission Frequency Feature</td>
<td>Infrequent</td>
<td>Infrequent</td>
<td>Infrequent</td>
<td>Frequent</td>
<td>Infrequent</td>
<td>Infrequent</td>
</tr>
<tr>
<td>RFC # or Draft</td>
<td>RFC7428</td>
<td>RFC7668, RFC9159</td>
<td>RFC8105</td>
<td>RFC8163</td>
<td>draft-ietf-6lo-nfc</td>
<td>draft-ietf-6lo-plc</td>
</tr>
</tbody>
</table>

#### 3. Guidelines for adopting an IPv6 stack (6lo)

6lo aims at reusing and/or adapting existing 6LoWPAN functionality in order to efficiently support IPv6 over a variety of IoT L2 technologies. The following guideline targets new candidate constrained L2 technologies that may be considered for running a modified 6LoWPAN stack on top. The modification of the 6LoWPAN stack should be based on the following:

- **Addressing Model**: The addressing model determines whether the device is capable of forming IPv6 link-local and global addresses, and what is the best way to derive the IPv6 addresses for the constrained L2 devices. L2-address-derived IPv6 addresses are
specified in [RFC4944], but there exist implications for privacy. For global usage, a unique IPv6 address must be derived using an assigned prefix and a unique interface ID. [RFC8065] provides such guidelines. For MAC-derived IPv6 addresses, please refer to [RFC8163] for IPv6 address mapping examples. Broadcast and multicast support are dependent on the L2 networks. Most low-power L2 implementations map multicast to broadcast networks. So care must be taken in the design for when to use broadcast, trying to stick to unicast messaging whenever possible.

- **MTU Considerations:** The deployment should consider packet maximum transmission unit (MTU) needs over the link layer and should consider if fragmentation and reassembly of packets are needed at the 6LoWPAN layer. For example, if the link layer supports fragmentation and reassembly of packets, then the 6LoWPAN layer may not need to support fragmentation/reassembly. In fact, for greatest efficiency, choosing a low-power link layer that can carry unfragmented application packets would be optimal for packet transmission if the deployment can afford it. Please refer to 6lo RFCs [RFC7668], [RFC8163], and [RFC8105] for example guidance.

- **Mesh or L3-Routing:** 6LoWPAN specifications provide mechanisms to support mesh routing at L2, a configuration called mesh-under [RFC6606]. It is also possible to use an L3 routing protocol in 6LoWPAN, an approach known as route-over. [RFC6550] defines RPL, a L3 routing protocol for low power and lossy networks using directed acyclic graphs. 6LoWPAN is routing-protocol-agnostic and does not specify any particular L2 or L3 routing protocol to use with a 6LoWPAN stack.

- **Address Assignment:** 6LoWPAN developed a new version of IPv6 Neighbor Discovery [RFC4861][RFC4862]. 6LoWPAN Neighbor Discovery [RFC7775][RFC8505] inherits from IPv6 Neighbor Discovery for mechanisms such as Stateless Address Autoconfiguration (SLAAC) and Neighbor Unreachability Detection (NUD). A 6LoWPAN node is also expected to be an IPv6 host per [RFC8200] which means it should ignore consumed routing headers and Hop-by-Hop options; when operating in a RPL network [RFC6550], it is also beneficial to support IP-in-IP encapsulation [RFC9008]. The 6LoWPAN node should also support [RFC8505] and use it as the default Neighbor Discovery method. It is the responsibility of the deployment to ensure unique global IPv6 addresses for Internet connectivity. For local-only connectivity IPv6 Unique Local Address (ULA) may be used. [RFC7775][RFC8505] specifies the 6LoWPAN border router (6LBR), which is responsible for prefix assignment to the 6LoWPAN network. A 6LBR can be connected to the Internet or to an enterprise network via one of the interfaces. Please refer to [RFC7668] and [RFC8105] for examples of address assignment.
considerations. In addition, privacy considerations [RFC8065] must be consulted for applicability. In certain scenarios, the deployment may not support IPv6 address autoconfiguration due to regulatory and business reasons and may choose to offer a separate address assignment service. Address Protection for 6LoWPAN Neighbor Discovery (AP-ND) [RFC8928] enables Source Address Validation [RFC6620] and protects the address ownership against impersonation attacks.

- Broadcast Avoidance: 6LoWPAN Neighbor Discovery aims at reducing the amount of multicast traffic of classical Neighbor Discovery, since IP-level multicast translates into L2 broadcast in many L2 technologies. 6LoWPAN Neighbor Discovery relies on a proactive registration to avoid the use of multicast for address resolution. It also uses a unicast method for Duplicate Address Detection (DAD), and avoids multicast lookups from all nodes by using non-onlink prefixes. Router Advertisements (RAs) are also sent in unicast, in response to Router Solicitations (RSs).

- Host-to-Router interface: 6lo has defined registration extensions for 6LoWPAN Neighbor Discovery [RFC8505]. This effort provides a host-to-router interface by which a host can request its router to ensure reachability for the address registered with the router. Note that functionality has been developed to ensure that such a host can benefit from routing services in a RPL network [RFC9010].

- Proxy Neighbor Discovery: Further functionality also allows a device (e.g., an energy-constrained device that needs to sleep most of the time) to request proxy Neighbor Discovery services from a 6LoWPAN Backbone Router (6BBR) [RFC8505][RFC8929]. The latter RFC federates a number of links into a multilink subnet.

- Header Compression: IPv6 header compression [RFC6282] is a vital part of IPv6 over low power communication. Examples of header compression over different link-layer specifications are found in [RFC7668], [RFC8163], and [RFC8105]. A generic header compression technique is specified in [RFC7400]. For 6LoWPAN networks where RPL is the routing protocol, there exist 6LoWPAN header compression extensions which allow also compressing the RPL artifacts used when forwarding packets in the route-over mesh [RFC8138] [RFC9035].

- Security and Encryption: Though 6LoWPAN basic specifications do not address security at the network layer, the assumption is that L2 security must be present. In addition, application-level security is highly desirable. The working groups [IETF_ace] and [IETF_core] should be consulted for application and transport level security. The 6lo working group has worked on address
authentication [RFC8928] and secure bootstrapping is also being discussed in the IETF. However, there may be other security mechanisms available in a deployment through other standards such as hardware-level security or certificates for the initial booting process. Encryption is important if the implementation can afford it.

- Additional processing: [RFC8066] defines guidelines for ESC dispatch octets use in the 6LoWPAN header. An implementation may take advantage of the ESC header to offer a deployment specific processing of 6LoWPAN packets.

4. 6lo Deployment Scenarios

4.1. Wi-SUN usage of 6lo in network layer

Wireless Smart Ubiquitous Network (Wi-SUN)[Wi-SUN] is a technology based on IEEE Std 802.15.4g. Wi-SUN networks support star and mesh topologies, as well as hybrid star/mesh deployments, but these are typically laid out in a mesh topology where each node relays data for the network to provide network connectivity. Wi-SUN networks are deployed on both grid-powered and battery-operated devices [RFC8376].

The main application domains using Wi-SUN are smart utility and smart city networks. This includes, but is not limited to the following applications:

- Advanced metering infrastructure
- Distribution automation
- Home energy management
- Infrastructure management
- Intelligent transportation systems
- Smart street lighting
- Agriculture
- Structural health (bridges, buildings)
- Monitoring and asset management
- Smart thermostats, air conditioning, and heat controls
- Energy usage information displays
The Wi-SUN Alliance Field Area Network (FAN) covers primarily outdoor networks, and its specification is oriented towards meeting the more rigorous challenges of these environments. It has the following features:

- Open standards based on IEEE 802, IETF, TIA, and ETSI
- Architecture based on an IPv6 frequency hopping wireless mesh network with enterprise-level security
- Simple infrastructure of low cost, low complexity
- Enhanced network robustness, reliability, and resilience to interference, due to high redundancy and frequency hopping
- Enhanced scalability, long range, and low energy usage
- Supports multiple global license-exempt sub-GHz bands
- Multi-vendor interoperability
- Very low power modes in development permitting long-term battery operation of network nodes

The Wi-SUN Field Area Network specification defines an IPv6-based protocol suite including TCP/UDP, IPv6, 6lo adaptation layer, DHCPv6 for IPv6 address management, RPL, and ICMPv6.

4.2. Thread usage of 6lo in network layer

Thread is an IPv6-based networking protocol stack built on open standards, designed for smart home environments, and based on low-power IEEE Std 802.15.4 mesh networks. Because of its IPv6 foundation, Thread can support existing popular application layers and IoT platforms, provide end-to-end security, ease development and enable flexible and future-proof designs [Thread].

The Thread specification uses the IEEE Std 802.15.4 [IEEE802154] physical and MAC layers operating at 250 kbps in the 2.4 GHz band.

Thread devices use 6LoWPAN, as defined in [RFC4944][RFC6282], for transmission of IPv6 Packets over IEEE Std 802.15.4 networks. Header compression is used within the Thread network and devices transmitting messages compress the IPv6 header to minimize the size of the transmitted packet. The mesh header is supported for link-layer (i.e., mesh under) forwarding. The mesh header as used in Thread also allows efficient end-to-end fragmentation of messages rather than the hop-by-hop fragmentation specified in [RFC4944].
Mesh under routing in Thread is based on a distance vector protocol in a full mesh topology.

4.3. G3-PLC usage of 6lo in network layer

G3-PLC [G3-PLC] is a narrowband PLC technology that is based on the ITU-T G.9903 Recommendation [G.9903]. G3-PLC supports multi-hop mesh network topology, and facilitates highly reliable, long-range communication. With the abilities to support IPv6 and to cross transformers, G3-PLC is regarded as one of the next-generation narrowband PLC technologies. G3-PLC has got massive deployments over several countries, e.g., Japan and France.

The main application domains using G3-PLC are smart grid and smart cities. This includes, but is not limited to the following applications:

- Smart metering
- Vehicle-to-grid communication
- Demand response
- Distribution automation
- Home/Building energy management systems
- Smart street lighting
- Advanced metering infrastructure (AMI) backbone network
- Wind/Solar farm monitoring

In the G3-PLC specification, the 6lo adaption layer utilizes the 6LoWPAN functions (e.g., header compression, fragmentation and reassembly). However, due to the different characteristics of the PLC media, the 6LoWPAN adaptation layer cannot perfectly fulfill the requirements [I-D.ietf-6lo-plc]. The ESC dispatch type is used in the G3-PLC to provide native mesh routing and bootstrapping functionalities [RFC8066].

4.4. Netricity usage of 6lo in network layer

The Netricity program in the HomePlug Powerline Alliance [NETRICITY] promotes the adoption of products built on the IEEE Std 1901.2 low-frequency narrowband PLC standard, which provides for urban and long-distance communications and propagation through transformers of the distribution network using frequencies below 500 kHz. The technology
also addresses requirements that assure communication privacy and secure networks.

The main application domains using Netricity are smart grid and smart cities. This includes, but is not limited to the following applications:

- Utility grid modernization
- Distribution automation
- Meter-to-Grid connectivity
- Micro-grids
- Grid sensor communications
- Load control
- Demand response
- Net metering
- Street lighting control
- Photovoltaic panel monitoring

The Netricity system architecture is based on the physical and MAC layers of IEEE Std 1901.2. Regarding the 6lo adaptation layer and an IPv6 network layer, Netricity utilizes IPv6 protocol suite including 6lo/6LoWPAN header compression, DHCPv6 for IP address management, RPL routing protocol, ICMPv6, and unicast/multicast forwarding. Note that the L3 routing in Netricity uses RPL in non-storing mode with the MRHOF (Minimum Rank with Hysteresis Objective Function) objective function based on their own defined Estimated Transmission Time (ETT) metric.

5. 6lo Use Case Examples

As IPv6 stacks for constrained node networks use a variation of the 6LoWPAN stack applied to each particular link layer technology, various 6lo use cases can be provided. In this section, various 6lo use cases which are based on different link layer technologies are described.
5.1. Use case of ITU-T G.9959: Smart Home

Z-Wave is one of the main technologies that may be used to enable smart home applications. Born as a proprietary technology, Z-Wave was specifically designed for this particular use case. Recently, the Z-Wave radio interface (physical and MAC layers) has been standardized as the ITU-T G.9959 specification.

Example: Use of ITU-T G.9959 for Home Automation

A variety of home devices (e.g., light dimmers/switches, plugs, thermostats, blinds/curtains, and remote controls) are augmented with ITU-T G.9959 interfaces. A user may turn on/off or may control home appliances by pressing a wall switch or by pressing a button in a remote control. Scenes may be programmed, so that after a given event, the home devices adopt a specific configuration. Sensors may also periodically send measurements of several parameters (e.g., gas presence, light, temperature, humidity) which are collected at a sink device, or may generate commands for actuators (e.g., a smoke sensor may send an alarm message to a safety system).

The devices involved in the described scenario are nodes of a network that follows the mesh topology, which is suitable for path diversity to face indoor multipath propagation issues. The multihop paradigm allows end-to-end connectivity when direct range communication is not possible. Security support is required, especially for safety-related communication. When a user interaction (e.g., a button press) triggers a message that encapsulates a command, if the message is lost, the user may have to perform further interactions to achieve the desired effect (e.g., turning off a light). A reaction to a user interaction will be perceived by the user as immediate as long as the reaction takes place within 0.5 seconds [RFC5826].

5.2. Use case of Bluetooth LE: Smartphone-based Interaction

The key feature behind the current high Bluetooth LE momentum is its support in a large majority of smartphones in the market. Bluetooth LE can be used to allow the interaction between the smartphone and surrounding sensors or actuators. Furthermore, Bluetooth LE is also the main radio interface currently available in wearables. Since a smartphone typically has several radio interfaces that provide Internet access, such as Wi-Fi or 4G, the smartphone can act as a gateway for nearby devices such as sensors, actuators or wearables. Bluetooth LE may be used in several domains, including healthcare, sports/wellness, and home automation.

Example: Use of Bluetooth LE-based Body Area Network for fitness
A person wears a smartwatch for fitness purposes. The smartwatch has several sensors (e.g., heart rate, accelerometer, gyrometer, GPS, temperature), a display, and a Bluetooth LE radio interface. The smartwatch can show fitness-related statistics on its display. However, when a paired smartphone is in the range of the smartwatch, the latter can report almost real-time measurements of its sensors to the smartphone, which can forward the data to a cloud service on the Internet. 6lo enables this use case by providing efficient end-to-end IPv6 support. In addition, the smartwatch can receive notifications (e.g., alarm signals) from the cloud service via the smartphone. On the other hand, the smartphone may locally generate messages for the smartwatch, such as e-mail reception or calendar notifications.

The functionality supported by the smartwatch may be complemented by other devices such as other on-body sensors, wireless headsets or head-mounted displays. All such devices may connect to the smartphone creating a star topology network whereby the smartphone is the central component. Support for extended network topologies (e.g., mesh networks) is being developed as of the writing.

5.3. Use case of DECT-ULE: Smart Home

DECT is a technology widely used for wireless telephone communications in residential scenarios. Since DECT-ULE is a low-power variant of DECT, DECT-ULE can be used to connect constrained devices such as sensors and actuators to a Fixed Part, a device that typically acts as a base station for wireless telephones. In this case, additionally, the Fixed Part must have a data network connection. Therefore, DECT-ULE is especially suitable for the connected home space in application areas such as home automation, smart metering, safety, and healthcare. Since DECT-ULE uses dedicated bandwidth, it avoids this coexistence issues suffered by other technologies that use e.g., ISM frequency bands.

Example: Use of DECT-ULE for Smart Metering

The smart electricity meter of a home is equipped with a DECT-ULE transceiver. This device is in the coverage range of the Fixed Part of the home. The Fixed Part can act as a router connected to the Internet. This way, the smart meter can transmit electricity consumption readings through the DECT-ULE link with the Fixed Part, and the latter can forward such readings to the utility company using Wide Area Network (WAN) links. The meter can also receive queries from the utility company or from an advanced energy control system controlled by the user, which may also be connected to the Fixed Part via DECT-ULE.
5.4. Use case of MS/TP: Building Automation Networks

The primary use case for IPv6 over MS/TP (6LoBAC) is in building automation networks. [BACnet] is the open, international standard protocol for building automation, and MS/TP is defined in [BACnet] Clause 9. MS/TP was designed to be a low-cost, multi-drop field bus to interconnect the most numerous elements (sensors and actuators) of a building automation network to their controllers. A key aspect of 6LoBAC is that it is designed to co-exist with BACnet MS/TP on the same link, easing the ultimate transition of some BACnet networks to native end-to-end IPv6 transport protocols. New applications for 6LoBAC may be found in other domains where low cost, long distance, and low latency are required. Note that BACnet comprises various networking solutions other than MS/TP, including the recently emerged BACnet IP. However, the latter is based on high-speed Ethernet infrastructure, and thus it falls outside of the constrained node network scope.

Example: Use of 6LoBAC in Building Automation Networks

The majority of installations for MS/TP are for "terminal" or "unitary" controllers, i.e., single zone or room controllers that may connect to HVAC or other controls such as lighting or blinds. The economics of daisy-chaining a single twisted-pair between multiple devices is often preferred over home-run, Cat 5-style wiring.

A multi-zone controller might be implemented as an IP router between a traditional Ethernet link and several 6LoBAC links, fanning out to multiple terminal controllers.

The superior distance capabilities of MS/TP (~1 km) compared to other 6lo media may suggest its use in applications to connect remote devices to the nearest building infrastructure. For example, remote pumping or measuring stations with moderate bandwidth requirements can benefit from the low-cost and robust capabilities of MS/TP over other wired technologies such as DSL, and without the line-of-sight restrictions or hop-by-hop latency of many low-cost wireless solutions.

5.5. Use case of NFC: Alternative Secure Transfer

In different applications, a variety of secured data can be handled and transferred. Depending on the security level of the data, different transfer methods can be alternatively selected.

Example: Use of NFC for Secure Transfer in Healthcare Services with Tele-Assistance
A senior citizen who lives alone wears one to several wearable 6lo devices to measure heartbeat, pulse rate. Other 6lo devices are densely installed at home for movement detection. A 6LBR at home will send the sensed information to a connected healthcare center. Portable base stations with displays may be used to check the data at home, as well. Data is gathered in both periodic and event-driven fashion. In this application, event-driven data can be very time-critical. In addition, privacy also becomes a serious issue in this case, as the sensed data is very personal.

While the senior citizen is provided audio and video healthcare services by a tele-assistance based on LTE connections, the senior citizen can alternatively use NFC connections to transfer the personal sensed data to the tele-assistance. Hackers can overhear the data based on the LTE connection, but they cannot gather the personal data over the NFC connection.

5.6. Use case of PLC: Smart Grid

The smart grid concept is based on deploying numerous operational and energy measuring sub-systems in an electricity grid system. It comprises multiple administrative levels/segments to provide connectivity among these numerous components. Last mile connectivity is established over the Low Voltage (LV) segment, whereas connectivity over electricity distribution takes place in the High Voltage (HV) segment. Smart grid systems include Advanced Metering Infrastructure (AMI), Demand Response (DR), Home Energy Management System (HEMS), Wide Area Situational Awareness (WASA), among others.

Although other wired and wireless technologies are also used in Smart Grid, PLC enjoys the advantage of reliable data communication over electrical power lines that are already present, and the deployment cost can be comparable to wireless technologies. The 6lo-related scenarios for PLC mainly lie in the LV PLC networks with most applications in the area of advanced metering infrastructure, vehicle-to-grid communications, in-home energy management, and smart street lighting.

Example: Use of PLC for Advanced Metering Infrastructure

Household electricity meters transmit time-based data of electric power consumption through PLC. Data concentrators receive all the meter data in their corresponding living districts and send them to the Meter Data Management System (MDMS) through a WAN network (e.g., Medium-Voltage PLC, Ethernet, or GPRS) for storage and analysis. Two-way communications are enabled which means smart meters can do actions like notification of electricity charges according to the commands from the utility company.
With the existing power line infrastructure as communication medium, cost on building up the PLC network is naturally saved, and more importantly, labor and operational costs can be minimized from a long-term perspective. Furthermore, this AMI application speeds up electricity charging, reduces losses by restraining power theft, and helps to manage the health of the grid based on line loss analysis.

Example: Use of PLC (IEEE Std 1901.1) for WASA in Smart Grid

Many sub-systems of Smart Grid require low data rates, and narrowband variants (e.g., IEEE Std 1901.1) of PLC fulfill such requirements. Recently, more complex scenarios are emerging that require higher data rates.

A WASA sub-system is an appropriate example that collects large amounts of information about the current state of the grid over a wide area from electric substations as well as power transmission lines. The collected feedback is used for monitoring, controlling, and protecting all the sub-systems.

6. IANA Considerations

There are no IANA considerations related to this document.

7. Security Considerations

Security considerations are not directly applicable to this document. For the use cases, the security requirements described in the protocol specifications apply.

The 6lo stack uses the IPv6 addressing model and it is required to consider the implication for privacy with L2-address-driven IPv6 addresses. In a typical 6lo use case with a variety of secured data (e.g., personal healthcare data), it is also required to provide secure data transmissions. Even though the 6lo stack do not address security at the network layer, it is required to provide L2-level security and application-level security is highly desirable.

8. Acknowledgements

Carles Gomez has been funded in part by the Spanish Government through the Jose Castillejo CAS15/00336 grant, the TEC2016-79988-P grant, and the PID2019-106808RA-I00 grant, and by Secretaria d’Universitats i Recerca del Departament d’Empresa i Coneixement de la Generalitat de Catalunya 2017 through grant SGR 376. His contribution to this work has been carried out in part during his stay as a visiting scholar at the Computer Laboratory of the University of Cambridge.
Thomas Watteyne, Pascal Thubert, Xavier Vilajosana, Daniel Migault, Jianqiang Hou, Kerry Lynn, S.V.R. Anand, and Seyed Mahdi Darroudi have provided valuable feedback for this draft.

Das Subir and Michel Veillette have provided valuable information of jupiterMesh and Paul Duffy has provided valuable information of Wi-SUN for this draft. Also, Jianqiang Hou has provided valuable information of G3-PLC and Netricity for this draft. Take Aanstoot, Kerry Lynn, and Dave Robin have provided valuable information of MS/TP and practical use case of MS/TP for this draft.

Deoknyong Ko has provided relevant text of LTE-MTC and he shared his experience to deploy IPv6 and 6lo technologies over LTE MTC in SK Telecom.

9. Informative References


[IEEE1901.2]

[IEEE802154]
IEEE standard for Information Technology, "IEEE Standard for Low-Rate Wireless Networks".

[IEEE802159]
IEEE standard for Information Technology, "IEEE Std 802.15.9-2016 - IEEE Recommended Practice for Transport of Key Management Protocol (KMP) Datagrams".

[I-D.ietf-6lo-nfc]

[I-D.ietf-6lo-plc]

[IETF_6lo]

[IETF_ace]

[IETF_core]
"IETF Constrained RESTful Environments (core) working group", <https://datatracker.ietf.org/wg/core/charters/>.

[Wi-SUN]

[Thread]

[NETRICITY]
"Netricity program in HomePlug Powerline Alliance", <http://groups.homeplug.org/tech/Netricity>.


Appendix A. Design Space Dimensions for 6lo Deployment

[RFC6568] lists the dimensions used to describe the design space of wireless sensor networks in the context of the 6LoWPAN working group. The design space is already limited by the unique characteristics of a LoWPAN (e.g., low power, short range, low bit rate). In [RFC6568], the following design space dimensions are described: Deployment, Network size, Power source, Connectivity, Multi-hop communication, Traffic pattern, Mobility, Quality of Service (QoS). However, in this document, the following design space dimensions are considered:

- Deployment/Bootstrapping: 6lo nodes can be connected randomly, or in an organized manner. The bootstrapping has different characteristics for each link layer technology.

- Topology: Topology of 6lo networks may inherently follow the characteristics of each link layer technology. Point-to-point,
star, tree or mesh topologies can be configured, depending on the link layer technology considered.

- L2-Mesh or L3-Mesh: L2-mesh and L3-mesh may inherently follow the characteristics of each link layer technology. Some link layer technologies may support L2-mesh and some may not support.

- Multi-link subnet, single subnet: The selection of multi-link subnet and single subnet depends on connectivity and the number of 6lo nodes.

- Data rate: Typically, the link layer technologies of 6lo have low rate of data transmission. But, by adjusting the MTU, it can deliver higher upper layer data rate.

- Buffering requirements: Some 6lo use case may require higher data rate than the link layer technology support. In this case, a buffering mechanism, telling the application to throttle its generation of data, and compression of the data are possible to manage the data.

- Security and Privacy Requirements: Some 6lo use case can involve transferring some important and personal data between 6lo nodes. In this case, high-level security support is required.

- Mobility across 6lo networks and subnets: The movement of 6lo nodes depends on the 6lo use case. If the 6lo nodes can move or be moved around, a mobility management mechanism is required.

- Time synchronization requirements: The requirement of time synchronization of the upper layer service is dependent on the use case. For some 6lo use case related to health service, the measured data must be recorded with exact time.

- Reliability and QoS: Some 6lo use case requires high reliability, for example, real-time or health-related services.

- Traffic patterns: 6lo use cases may involve various traffic patterns. For example, some 6lo use cases may require short data lengths and random transmission. Some 6lo use case may require continuous data transmission and discontinuous data transmission.

- Security Bootstraping: Without the external operations, 6lo nodes must have a security bootstrapping mechanism.

- Power use strategy: to enable certain use cases, there may be requirements on the class of energy availability and the strategy followed for using power for communication [RFC7228]. Each link
layer technology defines a particular power use strategy which may be tuned [RFC8352]. Readers are expected to be familiar with [RFC7228] terminology.

- Update firmware requirements: Most 6lo use cases will need a mechanism for updating firmware. In these cases, support for over the air updates is required, probably in a broadcast mode when bandwidth is low and the number of identical devices is high.

- Wired vs. Wireless: Plenty of 6lo link layer technologies are wireless, except MS/TP and PLC. The selection of wired or wireless link layer technology is mainly dependent on the requirements of the 6lo use cases and the characteristics of wired/wireless technologies.

Authors’ Addresses

Yong-Geun Hong
Daejeon University
62 Daehak-ro, Dong-gu
Daejeon 34520
Korea

Phone: +82 42 280 4841
Email: yonggeun.hong@gmail.com

Carles Gomez
Universitat Politecnica de Catalunya/Fundacio i2cat
C/Esteve Terradas, 7
Castelldefels 08860
Spain

Email: carlesgo@entel.upc.edu

Younghwan Choi
ETRI
218 Gajeongno, Yuseong
Daejeon 34129
Korea

Phone: +82 42 860 1429
Email: yhc@etri.re.kr
Abdur Rashid Sangi
Huaiyin Institute of Technology
No.89 North Beijing Road, Qinghe District
Huaian 223001
P.R. China

Email: sangi_bahrian@yahoo.com

Samita Chakrabarti
San Jose, CA
USA

Email: samitac.ietf@gmail.com
Native Short Addressing for Low power and Lossy Networks Expansion
draft-li-6lo-native-short-address-03

Abstract

This document specifies a topological addressing scheme, Native Short Address (NSA) that enables IP packet transmission over links where the transmission of a full length address may not be desirable. Furthermore, packet forwarding is stateless, meaning that no routing table needs to be built, rather, the forwarding decision is based solely on the destination address structure. This document focuses on carrying IP packets across an LLN (Low power and Lossy Network), in which the topology is static, where nodes’ location is fixed, and the connection between nodes is also rather stable. This specifications details the NSA architecture, address allocation, forwarding mechanism, header format design, including length-variable fields, and IPv6 interconnection support.

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 3 December 2022.
Copyright Notice

Copyright (c) 2022 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 Notation ..................................... 4
3. Architectural Overview .................................... 4
4. NSA Allocation ............................................ 7
   4.1. NSA Addresses and IPv6 Addresses ..................... 11
   4.2. Limitation of Number of Children Nodes ................ 12
5. Forwarding in a NSA Network ............................... 12
   5.1. Forwarding toward an NSA endpoint .................... 13
   5.2. Forwarding toward an external IPv6 node ............... 15
6. NSA Header Format ......................................... 15
7. NSA Control Message ....................................... 17
   7.1. New Control Message ................................ 17
   7.2. Address Configuration based on 6LOWPAN-ND .......... 18
       7.2.1. NSA Request Address Option (NRAO) Format ........ 18
       7.2.2. NSA Assign Address Option (NAAO) Format .......... 19
8. IANA Considerations ....................................... 20
   8.1. Dispatch Type Field ................................ 20
   8.2. Allocation Function Registry ......................... 20
   8.3. ICMP NSA Control Message ............................. 21
   8.4. NSA Neighbor Discovery Options ....................... 21
9. Reliability Considerations ................................ 21
10. Security Considerations ................................. 22
11. Acknowledgements ....................................... 22
12. References ............................................... 22
   12.1. Normative References ............................... 22
   12.2. Informative References ............................. 23
Authors’ Addresses ........................................... 25
1. Introduction

There is an ongoing massive expansion of the network edge that is driven by the "Internet of Things" (IoT), especially over low-power links which often, in the past, did not support IP packet transmission.

Particularly driven by the requirements stemming from Industry 4.0 and Smart City deployments, more and more devices/things are connected to the Internet. Sensors in plants/parking bays/mines, temperature/humidity/flash sensors in museums, normally are located in a fixed position and are networked by low power and lossy links even in hard-wired networks. Comparing with traditional scenarios, scalability of the (edge) network along with lower power consumption are key technical requirements. Moreover, large-scale Low power Lossy Networks (LLNs) are expected to be able to carry IPv6 packets over their links, together with an efficient access to native IPv6 domains.

The work in [SIXLOWPAN]/[SIXLO]/[LPWAN] Working Groups addresses many fundamental issues for those type of deployments. Those deployments can be considered an instantiation of what [RFC8799] defines as "limited domains". For instance, the 6lowpan compression technology ([RFC4944] and [RFC6282]) addresses the problem of IPv6 transmission over LLNs, making it possible to interconnect IPv6-based IoT networks and the Internet. [RFC8138] introduces a framework for implementing multi-hop routing on an LLN using a compressed routing header, which works also with RPL (Routing Protocol for LLNs [RFC6550]). This technique enables the ability to forward IPv6 packets within the domain without the need of decompression. In addition, SCHC (Generic Framework for Static Context Header Compression and Fragmentation [RFC8724]) enables even more compression by using a common static context.

Although aforementioned technologies are suitable in general for all IoT scenarios, there could be more simplified solutions for those scenarios and applications with static network topologies and stable network connections leveraging on wired technologies [I-D.ietf-6lo-use-cases] (e.g. PLC [I-D.ietf-6lo-plc] or MS/TP [RFC8163], and Industrial IoT technologies like [RS485], etc.). In those kinds of deployments, topologies are planned in advance and well provisioned, with sensor nodes usually fixed in specific locations. This draft presents a topology based addressing mechanism with shorter packet header and simpler forwarding rules for those static IoT networks.

The specifications in this document leverage on the dispatch type field ([RFC4944], [RFC8025]) that allows to accommodate the proposed
address format. This means that except the addresses (source and destination) the other fields of the header will be compressed mostly according to LOWPAN_IPHC. The proposed addressing is independent of Unique Local Addresses [RFC4193], which has a dependency on specific link-layer conventions [RFC6282]. It is also different from stateful address allocation that requires all nodes to obtain addresses from a centralized DHCP server, which leads to increased network startup time and consumption of extra bandwidth. Compared to RPL-based routing [RFC6550], NSA avoids the extra overhead of address assignment by integrating address assignment and tree forming together. Furthermore, NSA provides much smaller forwarding table size than storing mode RPL.

2. Requirements Notation

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] and [RFC8174] when, and only when, they appear in all capitals, as shown here.

3. Architectural Overview

Native Short Address (NSA) is an efficient topology-based network layer address assignment and packet forwarding mechanism. The NSA nodes are aware of their own IPv6 address, constructed by IPv6 prefix and the NSA (see Section 4.1 and Section 5.2). Inside the NSA domain, nodes communicate with each other by using only NSA addresses. It is a smaller addressing space compared to the huge IPv6 addressing space. The NSA enables stateless forwarding. When IPv6 communication occurs between nodes inside the NSA domain and external IPv6 nodes, the border router, which plays as well the role of "root" in the addressing tree, performs network address translation (as per Section 5.2 and [RFC6282]). The architecture of NSA network is showed in Figure 1.
In the NSA network, there are 3 types of nodes, the root node, the forwarder node and the leaf node. There is typically only one root node in the NSA network.

* Root Node: The root node is responsible for the management of the whole NSA network and routing/forwarding both internal and external traffic. It stores the IPv6 prefix of the domain in order to perform the network address translation for external communications. It also stores the address Allocation Function (AF) and performs the address assignment for its children. After successful address assignment, the root will keep the state of its direct children. The root node functions as gateway between the NSA domain and the Internet. As such it also operates the translation between NSA header and IPv6 header (cf. Section 5).

* Forwarder Nodes: A forwarder is a node, different from the root node, containing at least one child. A forwarder node is basically the root of a subtree and its role is to forward traffic between its parent and its children according to the addressing. When handling a packet, if the destination is in one of its
subtrees, it forwards the packet to the right child, otherwise it simply sends it to its parent.

* Leaf Nodes: A leaf node is a node with no children. Its operation is simple since it is either a destination or source of every packet it handles. If it is the source of packets, it simply sends the packets to its parent.

Each node acquiring a native short address needs to send an Address Request (AR) message to its link layer neighbors and wait for the response. In the AR message, the node needs to designate a 'role' value (forwarder or leaf) and the "node-id". The latter is a unique identifier of each NSA node, including root, forwarders, and leaves. This document assumes the use of the link-layer address of the node as 'node-id'.

Forwarder and Leaf roles can be assigned similarly to IEEE 802.15.4, which distinguishes between Full-Function Devices (FFD) and reduced function devices (RFD) (cf., [ZigBee]). If a neighbor is neither a forwarder nor the root, it will drop the AR message silently. Otherwise, the neighbor will calculate an address based on parameters in the AR message. After the neighbor node assigns an address to the node, using a Allocation function (AF), it stores the suffix of that address as the interface ID towards the node. Then, it generates and sends Address Assignment (AA) message back and becomes the parent node.

This address assignment relies on the base mechanism described in 6lowpan-ND ([RFC6775]), but defines two new options of ND message, whose format is defined in Section 7.2.1 and Section 7.2.2.

The acceptance of the address assignment follows "first come first serve" principle. Once a node receives a valid AA response, it uses that assigned address as its own network layer address, thus becomes a child of the address assigner. It will then ignore replies from other neighbors.

If a node does not receive any response after RTR_SOLICITATION_INTERVAL (10 seconds defined in [RFC6775]), it will send the AR message again. It is RECOMMENDED that nodes re-send the AR message up to MAX_RTR_SOLICITATIONS (3 transmissions defined in [RFC6775]), if no answer is received, they SHOULD stop.

The overall design objective is centered on reducing the size (or completely avoid the usage) of routing/forwarding table with a topological addressing scheme. NSA eliminates compression/decompression of the address and also reduces the amount of information synchronization messages, so it actually reduces
computation complexity during packets parsing and forwarding. As such, NSA may save communication energy in an IoT LLN network.

NSA uses a context-independent address encoding mechanism. It does not carry any field about address context in the packet. It carries source and destination addresses as variable length fields whose size can be reduced to one octet each in the best case. This allows the NSA packet header to be smaller than LOWPAN_IPHC’s 7 octets (see Figure 2), down to 4 octets, representing around 40% reduction in the header size.

<table>
<thead>
<tr>
<th>Payload Length (variable length)</th>
<th>CID</th>
<th>SAC</th>
<th>SAM</th>
<th>M</th>
<th>DAC</th>
<th>DAM</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>I/O</td>
<td>AM</td>
<td>Src (variable length)</td>
<td>SCI</td>
<td>DCI</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Dst (variable length)</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

b. IPHC best case header

a. NSA best case header

Figure 2: Best case of NSA and LOWPAN_IPHC packet header.

There are three distinct NSA features that allow NSA to be efficient, namely:

1. Native Short Address allocation (see Section 4),
2. Stateless forwarding (see Section 5),
3. Compact header format design (see Section 6) that avoids context and compression.

4. NSA Allocation

The basic rules of allocation include:
* Each node’s address is prefixed by their parent’s address.

* The root/forwarder runs an AF (Allocation Function) to generate its children’s addresses.

* All nodes run the same AF in the same network instance.

* The maximum length of the NSA address should not exceed 64-bit.

Normally, the root role is assigned to the border router when the LLN bootstraps. An example of a possible result of an NSA deployment is shown in Figure 3.

![Diagram of NSA addresses allocation]

Figure 3: An example of NSA addresses allocation.

The allocation function AF(role, i) used in this document is defined in Figure 4. Every forwarder node stores and maintain two indexes, one for the children that are forwarders and one for the children that are leaves (starting at 0 for the first child in each role). Let’s call the first index ‘f’, as of forwarder, and the second ‘l’ as for leaves. The ‘+’ symbol indicates a concatenation operation. The b() operation indicates the binary string of ‘1’ with length equal to its argument, for instance b(3) returns ’111’.

Li, et al. Expires 3 December 2022 [Page 8]
AF\( (\text{role}, f, l) = \text{‘address of the node performing the function’} \\right) + (\text{role} == \text{leaf}? b(l++) : b(f++)) + (\text{role} == \text{leaf}? \text{‘1’} : \text{‘0’}), \)  

in which, \( f \) and \( l \) are the indexes of respectively the forwarders and the leaves at this layer (starting at 0).

Figure 4: Definition of the Allocation Function (AF) of forwarder/root nodes.

Taking the example of the topology in Figure 3, the proposed AF works as follows.

At the top level, there are 4 children of root, two are forwarders and the other two are leaves. Starting from the left most node and moving to the right, the root node applies the AF as follows:

* For the first child, which is a forwarder:
  - \( A(\text{‘forwarder’, 0, 0}) = \text{‘1’}(\text{root address}) + b(0) + \text{‘0’} = \text{‘1’} + \text{‘0’} = 10 \)
  - Index \( f \) is increased by one and is now equal 1 (\( f=1 \))

* For the second child, which is a leaf:
  - \( A(\text{‘leaf’, 1, 0}) = \text{‘1’}(\text{root address}) + b(0) + \text{‘1’} = \text{‘1’} + \text{‘1’} = 11 \)
  - Index \( l \) is increased by one and is now equal 1 (\( l=1 \))

* For the third child, which is a forwarder:
  - \( A(\text{‘forwarder’, 1, 1}) = \text{‘1’}(\text{root address}) + b(1) + \text{‘0’} = \text{‘1’} + \text{‘1’} + \text{‘0’} = 110 \)
  - Index \( f \) is increased by one and is now equal 2 (\( f=2 \))

* For the fourth child, which is a leaf:
  - \( A(\text{‘leaf’, 2, 1}) = \text{‘1’}(\text{root address}) + b(1) + \text{‘1’} = \text{‘1’} + \text{‘1’} + \text{‘1’} = 111 \)
  - Index \( l \) is increased by one and is now equal 2 (\( l=2 \))

The first level addresses have now been assigned. Let’s now have a look how the node 10 (the first forwarder child of the root) applies the same Allocation Function. Note that node 10 will use its own ‘\( f’
and 'l' indexes initialized to 0. Starting again from the left most
node, node 10 applies the AF as follows:

* For the first child, which is a forwarder:
  - A('forwarder', 0, 0) = '10'(node address) + b(0) + '0' = '10' + 
    '0' = 100
  - Index f is increased by one and is now equal 1 (f=1)

* For the second child, which is a leaf:
  - A('leaf', 1, 0) = '10'(node address) + b(0) + '1' = '10' + '1' = 101
  - Index l is increased by one and is now equal 1 (l=1)

* For the third child, which is a forwarder:
  - A('forwarder', 1, 1) = '10'(node address) + b(1) + '0' = '10' + 
    '1' + '0' = 1010
  - Index f is increased by one and is now equal 2 (f=2)

* For the fourth child, which is a leaf:
  - A('leaf', 2, 1) = '10'(node address) + b(1) + '1' = '10' + '1' + '1' = 1011
  - Index l is increased by one and is now equal 2 (l=2)

Note how the children of the same parent all have the same prefix (10
in this example). The proposed AF algorithmically assigns addresses
to the different nodes without the need to know the topology in
advance. However, the largest address of the network will depend on
the actual topology. Indeed, the maximum length of an address with
the proposed AF grows linearly at each level of the tree with the
number of siblings from the same parent. Let’s take again the
example in Figure 3 and let’s assume that the children of node 10 are
all leaves, for the largest address we need 2 bits to encode the
parent node prefix (10 in this case) to which we need to add a number
of '1' equal to the value of the l index which is the number of
leaves minus one (because the first leaf has index 0), in this case
since there are 4 leaves, the index value is 3 and we add the '111'
string, hence the address length would be 6 (2 for the prefix, 3 to
encode the 4th leaf address, and one for the final 1 the ends all
leaves addresses). In a more formal way the maximum address length
at each level can be calculated as:
Max_Length = length(Parent address)
      length(b(max(f,l)))
+ 1

Where f and l are the indexes counting respectively the forwarders
and the leaves at this level.

The Allocation Function can be different from the one defined in
Figure 4, but all nodes know which one to use by configuration. The
use of one and only one AF is allowed in an NSA domain. It is
RECOMMENDED that implementations support at least the AF proposed in
this document (cf. Section 8).

Different allocation functions may, for example, leverage on a priori
knowledge of the topology in order to optimize the maximum address
size and make it smaller. For instance, because the order of address
allocation has an impact on the size, the address of children with
the largest subtree should be allocated in the first place so to
reduce the average address length of the whole subtree. Also,
knowing the traffic in advance, or being able to have an estimation,
can help to minimize the size of addresses that have a lot of
traffic. This kind of optimization can be an option, the
specification of optimizations is out of the scope of this document
and may be defined in new Allocation Functions to be added to the
"Allocation Function Registry" (see Section 8).

4.1. NSA Addresses and IPv6 Addresses

Obtaining a full IPv6 address from a NSA address is pretty
straightforward. First the NSA address is concatenated to the
configured IPv6 prefix. Since the length of the NSA address is
smaller than or equal to 64 bits (the interface ID length in IPv6),
the node needs to pad it with zeros ("0") used as most significant
bits. The full IPv6 address will look like: IPv6 prefix +
"000...000" + NSA (or in IPv6 notation <IPv6 Prefix>::<NSA>). The
NSA is assigned by the root/forwarder as previously described.

In an IPv6 communication, the node will derive the NSA address as the
short source address from its own IPv6 address by simply removing the
IPv6 prefix and all leading zeros before the NSA part. The node will
compare the destination IPv6 address with its own IPv6 address. If
they have the same prefix, it means that the destination is in the
local NSA domain and its corresponding NSA address will be extracted
as the short destination address (and the I/O Flag can be set
accordingly). Otherwise, it will be a communication towards the
Internet. In that case, a mapping mechanism implemented in the root
node will generate a short address to be mapped to the full IPv6
destination address. For instance, the mapped short address can be
generated using the least significant bits of the original IPv6 address. As previously stated, the mapping mechanism is out of the scope of this document.

Since the short mapped address is generated on the root, when the node first opens the connection toward the external site, with a first packet, the destination address is set to the full, uncompressed, IPv6 address. Once the packet arrives at the root node, by performing the destination address lookup, the root will notice that a full IPv6 address is being used and will trigger the short address generation mechanism and create a new mapping. Such a mapping is communicated to the source node via a new dedicated ICMP message (see Section 7). Once the node originating the communication receives such a message it SHOULD use the mapped short address for any further communication.

NSA does not prevent the normal checksum calculation for the transport layer (namely TCP or UDP) or IPSec encapsulation. Indeed, any NSA node is aware of its full IP address, which can be used for the calculation. For communication to/from the Internet, NSA nodes store the mappings between the external remote address and the short mapped address, hence checksum calculation can be performed as usual.

4.2. Limitation of Number of Children Nodes

The maximum number of child nodes is determined by the specific AF used. IEEE 802.15.5 has explored the use of a per-branch setup, which, however, incurs scalability problems [LEE10]. NSA allocation design is more flexible and extensible than the one proposed in IEEE 802.15.5. The AF used as example in this document does not need any specific setup network by network, though it is still limited by the maximum length of addresses. For the special case of the parent connecting to huge amount of children, a variant of the proposed AF can be designed to fulfill the requirement and optimize the address allocation (as previously described).

5. Forwarding in a NSA Network

Internal and external communications in an NSA network work slightly different. For internal communications, among NSA endpoints, packets carry native short addresses and no special operation is needed. For external communications, the root is responsible to perform the translation between native short addresses and IPv6 addresses. For instance, for a packet entering into the NSA domain, the root will extract the native short address of the destination from the suffix of the IPv6 address, by removing all leading ‘0’s. It will also map the source IPv6 address to a mapped native short address, in order to make it more efficient for communication inside the NSA domain.
The root has to store the mapping between external IPv6 addresses and their assigned mapped Native Short Addresses. The method of generating those mapping is out of scope of this document, however, the addressing space for the external NSA has to be maintained separate from the internal NSA address space. Overlap are allowed since the two addressing space are distinguishable in the packets by the use of the I/O field, as explained later on.

The following paragraphs will detail the forwarding operations for both internal and external communication. The intra-network forwarding procedure depends on the specific AF used. Here we will use the AF previously introduced (see Figure 4) to illustrate the forwarding procedure.

5.1. Forwarding toward an NSA endpoint

To perform forwarding operations, NSA nodes access the I/O field in the NSA header (see Section 6). When its value is 1, the packet is destined to an internal NSA node, so it is an inner-domain packet. Otherwise, the packet is destined to an external IPv6 node. It is called an outer-domain packet. Intra-domain packets carry a native short addresses in the source and the destination address fields. More specifically the destination address field is the address of another node in the same NSA domain. As such an NSA node performs the following sequence of actions (also see Figure 5):

1. Get destination address from packet (abbreviated to DA) and the current node’s address (abbreviated to CA). Go to step 2.
2. If length of DA is smaller than length of CA, send the packet to parent node, exit. Otherwise, go to step 3.
3. If length of DA equals to length of CA, go to step 4. Otherwise, go to step 5.
4. If DA and CA are the same, the packet arrived at destination, exit. Otherwise, send the packet to parent node, exit.
5. Check whether CA is equal to the prefix of DA. If yes, go to step 6. Otherwise, send the packet to parent node, exit.
6. Calculate which child is the next hop address and forward packet to it. With the AF propose in this document such operation reduces to reading the DA’s bits starting from the position equals to the length of CA, then skip all ‘1’ until the first ‘0’ or the last bit of DA. The sub-string obtained in such a way is the address of direct child of current node.
7. If any exception happens in the above steps, drop the packet and send error notification.

```
/*\    DA: Destination Address
    |***|    CA: Current Node's Address
    \*/
```

```
+--------+--------+
| Parse DA from pkt |
+--------+--------+
\         \   yes
\         \   no
\         \   yes
\         \   no
\         \   yes
\         \   no
\         \   yes
```

```
<table>
<thead>
<tr>
<th>Len(DA)&lt;Len(CA)?         CA == DA ?</th>
<th>CA == PrefixOf(DA)?</th>
<th>Calculate next-hop &amp; Forward</th>
</tr>
</thead>
<tbody>
<tr>
<td>no</td>
<td>no</td>
<td>yes</td>
</tr>
<tr>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
```

```
| Len(DA)=Len(CA)?         CA == DA ? | CA == PrefixOf(DA)? |
|--------------------------|----------------------|-------------------------|
| no                       | yes                  | yes                     |
| yes                      | no                   | no                      |
```

```
+-------+----------------+
| Len(DA)=Len(CA)? | CA == DA ? |
+-------+----------------+
| no    | yes            |
| yes   | no             |
```

```
+--------+--------+
| Forward to Parent |
+--------+--------+
```

Figure 5: Flow Chart of Internal Forwarding Procedure

In the case of packet arriving from the Internet (external IPv6 domain toward the local NSA domain) header adaptation operation is
performed by the root node. Concerning the destination address, the root builds the native short address of the destination by removing the prefix and the leading '0's of the suffix of the destination address. Meanwhile, it checks whether it exists already a mapping between the source address and a mapped NSA address to be used as source address in the NSA packet. If not it creates one. Then the root creates the inner-domain packet. It uses the NSA address as destination setting the I/O field to 1 so to route the packet to as described above to the destination node. The mapped NSA address is used as source address and the fact that is a Mapped Address is signaled by setting to 1 the MA field.

5.2. Forwarding toward an external IPv6 node

In the case that the I/O field (cf. Section 6) is set to 0, the packet is destined to an external IPv6 node, it is an outer-domain packet. As such the destination address is either a full IPv6 address (for the first packet of a communication) or a mapped short address generated by the root node and not belonging to any node inside the NSA domain.

All NSA nodes (except root) just send packets that are destined outside the local domain (I/O field equal 0) to their parent, not even looking at the actual destination address. Eventually all packets will reach the root node, which acts as gateway. The root node is able to map the destination NSA address to the corresponding full IPv6 address. Also, the root node is able to rebuild the full source IPv6 address by concatenating the IPv6 prefix and the NSA address as explained in Section 5.2. Other fields of the header are also decompressed as described in Section 6. A full IPv6 header replaces the original NSA header in the packet, which is then forwarded according to traditional IPv6 protocol.

6. NSA Header Format

As explained in Section 4, the addresses in NSA are of variable length, in this section, we outline the design of the header format partially based on the format of 6lowPAN, accommodating the variable length property in the packet. The header format is shown in Figure 6.
The first 4 bits are new dispatch types that will be introduced in Section 8.

* TF: The same definition as in [RFC6282] Section 3.1.1.
* NH: The same definition as in [RFC6282] Section 3.1.1.
* HL: This field indicates the hop limit. When HL is 1, a hop limit field defined in [RFC2460] locates in in-line fields, while HL is 0 means no hop limit header in packet.

* Payload length is a variable length field. It normally occupies an octet assuming most packets are smaller than 252 octets. For larger packets, payload length may expand to 2 to 3 octets. The encoding method is defined as follows. When the first octet has value of:
  - 0^252: Indicates how many octets the payload consist of.
  - 253: Indicates that there is an extra octet for payload length, with the actual length value equal to the last octet value plus 252.
  - 254: Indicates that there is an extra two octets for payload length, with the actual length value obtained from the second and third octets interpreted as a 16 bits unsigned integer plus 252 (from the first octet).
  - 255: Reserved.

* I/O: Indicates whether this packet is destined to a inner-domain node (value '1') or an outer-domain node (value '0'), where the former means from an NSA or IPv6 node to a NSA destination, while the latter means to an external IPv6 node.
* MA: Indicates the source address is actually a Mapped Address generated by the root. When it is '1', the source address of the packet is a mapped address of an external IPv6 address, while if it is '0', the source address of the packet is an NSA address.

For length variable native short address encoding, for both Source Address (SA) and Destination Address (DA), the definition is:

* 0~252: if the address value locates in this interval, one octet is used to encode the value
* 253: indicates that the following 2 octets encode the address.
* 254: indicates that the following 4 octets encode the address.
* 255: indicates that the following octet defines the length of address in octets, followed by the address octets.

The sequence of in-line fields is as per [RFC8200] section 3.

7. NSA Control Message

7.1. New Control Message

This document specifies only one new NSA Control Message, namely the NSA Mapped Address Advertisement described in Section 4. The purpose of such a message is advertise the mapping of an IPv6 address into an NSA address. The map is performed by the root node and sent to the node originating the communication. The root keeps a copy of the mapping to be used for future packets. The format is as follows:

```
+-------------------------------+-------------------------------+-------------------------------+-------------------------------+
| Type | Code = 0x00 | Reserved | NSA Length |
+-------------------------------+-------------------------------+-------------------------------+-------------------------------+
| Target IPv6 Address (Fixed length 128 bits) |
+-------------------------------+-------------------------------+-------------------------------+-------------------------------+
| Target NSA Address (Variable length). |
+-------------------------------+-------------------------------+-------------------------------+-------------------------------+
```

* Type: Type value identifying NSA Control Message. Value to be assigned by IANA (cf. Section 8)
* Code: This field identifies the specific control message. In this case it is set to the value 0x00 "NSA Mapped Address for External IPv6 Address".

* Reserved: Set as 0 on transmission and ignored on reception.

* NSA Length: This field indicates the length of the Target NSA Address at the end of the message, expressed in octets.

The "NSA Mapped Address for External IPv6 Address" is a variable length message, however, the first five fields of the message, namely Type, Code Reserved, NSA Length, and Target IPv6 address, have a fixed length of 160 bits (20 octets), hence the length of the NSA address is sufficient to calculate the length of the entire packet: 20 octets + "NSA length".

7.2. Address Configuration based on 6LOWPAN-ND

According to [RFC6775], neighbor discovery is available in 6LoWPANs. This document specifies NSA address configuration mechanism based on RS (Router Solicitation) and solicited RA (Router Advertisement) defined in [RFC4861]. In order for an NSA node to request an address, it uses a newly defined 'Request Address Option (NRAO)' in RS messages. The corresponding solicited RA will contain the 'NSA Assign Address Option (NAAO)' with the assigned address.

7.2.1. NSA Request Address Option (NRAO) Format

This option will be carried in RS messages [RFC4861] when node initializes. The same RS messages MUST carry the Source Link-Layer Address Option (SLLAO) ([RFC4861], [RFC6775]) as well. The link-layer address in SLLAO (Source Link-Layer Address Option will be used to identify unique NSA node. The NRAO option, respecting the specifications in [RFC6775], has the following format:

```
+---------------+--------------+-------------------------------+
|     Type      |    Length    |   Expected Address Lifetime   |
+---------------+--------------+-------------------------------+
|                          Reserved                            |
+--------------------------------------------------------------+
```

* Type: 136

* Length: 8-bit unsigned integer. The length of the option (including the Type and Length fields) in units of 8 octets. This field is always set to 1.
* Expected Address Lifetime: The sender of RS notify the node that assigns the address for how long is expected to be valid. The receiver may ignore this field. As for [RFC6775] the unit is set to 60 seconds (1 minute). This field MUST be set to zero by sender if there is no requirement on the lifetime.

* Reserved: This field is not used. It MUST be initialized to zero by the sender and MUST be ignored by the receiver.

7.2.2. NSA Assign Address Option (NAAO) Format

This option will be carried in the RA message solicited by the an RS message as for the usual Neighbor Discovery workflow. The NAAO option, respecting the specifications in [RFC6775], has the following format:

```
+---------------+--------------+-----------------------------+
|     Type      |    Length    |        Address Lifetime      |
+---------------+--------------+-----------------------------+
| Prefix Length |                   Reserved                   |
+---------------+----------------------------------------------+
|                                                              |
|                                                              |
|                                                              |
|                    NSA with IPv6 Prefix                      |
+--------------------------------------------------------------+
```

* Type: 137

* Length: 8-bit unsigned integer. The length of the option (including the Type and Length fields) in units of 8 octets. This field is always set to the value 3.

* Address Lifetime: The maximum time for the NSA being valid. As for [RFC6775] the unit is set to 60 seconds (1 minute). The node with this address MUST stop using this address for packet transmission when the life time expires. When the Address Lifetime is zero, the node must drop the address immediately. When the lifetime field is 0xFFFF, the address will be valid forever until the node sends another NAAO to update the lifetime.

* Prefix Length: This field will notifies the receiver the length of the IPv6 prefix, expressed in octets.
Reserved: This field is not used. It MUST be initialized to zero by the sender and MUST be ignored by the receiver.

* NSA with IPv6 Prefix: This field is filled by the node with the IPv6 prefix (according with the length field), the NSA address as the least significant bits of the IPv6 address, and padding the remaining bits in the middle with zeros.

8. IANA Considerations

8.1. Dispatch Type Field

This document requires IANA to assign the range 01010000 to 01011111 in page 10 of the "Dispatch Type Field" registry as follows:

<table>
<thead>
<tr>
<th>Bit Pattern</th>
<th>Page</th>
<th>Header Type</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>0101TTNH</td>
<td>10</td>
<td>LOWPAN NSA IP(LOWPAN_NIP)</td>
<td>[This Document]</td>
</tr>
</tbody>
</table>

Figure 7: LOWPAN Dispatch Type Field requested allocation

8.2. Allocation Function Registry

This section provides guidance to the Internet Assigned Numbers Authority (IANA) regarding registration of values related to the NSA specification, in accordance with BCP 26 [RFC8126].

IANA is asked to create a registry named "Native Short Addresses (NSA) Parameters".

Such registry should be populated with a one octet sub registry named "Allocation Function" and used to identify the AF used in a NSA deployment. The sub registry is populated as follows:

<table>
<thead>
<tr>
<th>Value</th>
<th>AF Name</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x00</td>
<td>Native Allocation Function</td>
<td>[This Document]</td>
</tr>
<tr>
<td>0x01-0xFF</td>
<td>Un-assigned</td>
<td></td>
</tr>
</tbody>
</table>

Values can be assigned by IANA on a "First Come, First Served" basis according to [RFC8126].
8.3. ICMP NSA Control Message

IANA is requested to allocate an ICMPv6 type value from the "ICMPv6 Parameters" registry to be used by "NSA Control Message".

Also IANA is requested to create an "NSA Control Codes" sub registry, for the Code field of the ICMPv6 NSA Control Message.

New codes may be allocated through the "Specification Required" procedure as defined in [RFC8126]. The following code is currently defined (the others are to be marked as un-assigned):

<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x00</td>
<td>NSA Mapped Address for External IPv6 Address</td>
<td>[This Document]</td>
</tr>
</tbody>
</table>

8.4. NSA Neighbor Discovery Options

IANA is requested to allocate two values from the "IPv6 Neighbor Discovery Option Formats" registry to be used by NRAO and NAAO. Suggested values are respectively 136 and 137. [Note to RFC Editor: If IANA assign different values the authors will update the document accordingly]

<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>136</td>
<td>NSA Request Address Option</td>
<td>[This Document]</td>
</tr>
<tr>
<td>137</td>
<td>NSA Assign Address Option</td>
<td>[This Document]</td>
</tr>
</tbody>
</table>

9. Reliability Considerations

Because NSA uses algorithmically generated addresses based on the network topology, nodes do not generate and store forwarding table entries in the normal case. One of the potential issues is the risk of renumbering of addresses in case of topology changes. Because of the applicability domain of NSA, the common case of topology change is known in advance and can be planned, so to reduce disruption due to renumbering. Another case is temporary link failures where the underlying technology is still able to provide connectivity through alternative links, which is strictly related to the underlying technology, the network topology, the deployed redundancy, and the expected reliability.
More complex reliability scenarios and alternative solutions are beyond the scope of this document, which is focused only on the address allocation framework. Furthermore, specific reliability solutions can depend as well on the specific Allocation Function used (different from the one presented in this document). Reliability is discussed in more details in [I-D.li-nsa-reliability-00].

10. Security Considerations

An extended security analysis will be provided in future revision of this document. As of this point we consider that the security considerations of [RFC4944], [RFC6282] apply.

11. Acknowledgements

This document received many discussion and help from community people. Pascal Thubert’s technical questions steers this document being improved. Brian Carpenter reminds key issues about IPv6 address usage. Dominique Barthel, Adnan Rashid, Michael Richardson, provide technical comments for this document. There are other people helped on improving this document who want to be unnamed. The authors would present thanks to all of them.

12. References

12.1. Normative References


12.2. Informative References

[I-D.ietf-6lo-plc]
Hou, J., Liu, B., Hong, Y., Tang, X., and C. E. Perkins,
"Transmission of IPv6 Packets over PLC Networks", Work in
Progress, Internet-Draft, draft-ietf-6lo-plc-11, 18 May
2022, <https://www.ietf.org/archive/id/draft-ietf-6lo-plc-
11.txt>.

[I-D.ietf-6lo-use-cases]
Hong, Y., Gomez, C., Choi, Y., Sangi, A. R., and S.
Chakrabarti, "IPv6 over Constrained Node Networks (6lo)
Applicability & Use cases", Work in Progress, Internet-
Draft, draft-ietf-6lo-use-cases-12, 25 January 2022,
<https://www.ietf.org/archive/id/draft-ietf-6lo-use-cases-
12.txt>.
[I-D.li-nsa-reliability-00]


Authors’ Addresses

Guangpeng Li
Huawei Technologies
Beijing Road, Haidian District
Beijing
100095
China

Email: liguangpeng@huawei.com

David Lou
Huawei Technologies Duesseldorf GmbH
Riesstrasse 25
80992 Munich
Germany

Email: zhe.lou@huawei.com

Luigi Iannone
Huawei Technologies France S.A.S.U.
18, Quai du Point du Jour
92100 Boulogne-Billancourt
France

Email: luigi.iannone@huawei.com

Peng Liu
China Mobile
No. 53, Xibianmen Inner Street, Xicheng District
Beijing
100053

Li, et al. Expires 3 December 2022
China

Email: liupengyjy@chinamobile.com

Rong Long
China Mobile
No. 53, Xibianmen Inner Street, Xicheng District
Beijing
100053
China

Email: longrong@chinamobile.com
Abstract

Native Short Address (NSA [I-D.li-6lo-native-short-address]), proposes to algorithmically assign short addresses to nodes in a 6lo environment so to achieve stateless forwarding, hence, avoiding using a routing protocol. NSA is more suitable in case of stable and static wireline connectivity, in order to avoid renumbering due to topology changes. Even in such kind of scenarios, reliability remains an issue. This memo tackles specifically reliability in NSA deployments, analyzing possible broad solution categories to solve the issue.

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 2 December 2022.

Copyright Notice

Copyright (c) 2022 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 and Problem Statement .......................... 2
2. Requirements Notation ........................................ 3
3. Potential Solution Approaches .................................. 3
  3.1. Multi-Address Approach ................................... 3
    3.1.1. Topology building .................................... 4
    3.1.2. Link Failures ........................................ 7
    3.1.3. Node Failures ........................................ 11
    3.1.4. Nodes Forwarding Procedure ........................... 13
  3.2. Single-Address Approach ................................... 15
    3.2.1. Link Failure ......................................... 17
    3.2.2. Node Failure ......................................... 19
    3.2.3. Node Forwarding Procedure ........................... 19
4. Links/Nodes Failure Detection and Recovery ....................... 21
5. Robustness ..................................................... 22
6. Security Considerations .......................................... 22
7. IANA Considerations ............................................. 22
8. References ....................................................... 22
  8.1. Normative References ....................................... 22
  8.2. Informative References ..................................... 23
Authors’ Addresses .................................................. 24

1. Introduction and Problem Statement

The common characteristic of various topological addressing schemes ([I-D.daniel-6lowpan-hilow-hierarchical-routing], [I-D.li-6lo-native-short-address], [KIM07]) is the possibility of nodes to forward packets without the need of routing tables, and hence, without the need of routing protocols. In such context the addresses are build in such a way that a node is capable of forwarding a packet to the next hop by comparing the destination address with its own address. It is not required to build routing table on which to execute look-up algorithms, only neighbor awareness is sufficient. However, this kind of stateless forwarding typically works in a single topology with static paths, where high reliability is hard to reach. Once a link (or a node) fails, the traffic will not be routable and packets will be dropped, even in the presence of alternative physical paths. Indeed, in order to use these alternative paths renumbering is necessary to (re)build an alternative logical topology. Such a solution, while looking as a simple operation, may be not enough and complicate in practice, since it implies to put the system offline during the renumbering process.
What is desirable is to have some mechanism that with little extra effort may quickly enable the usage of alternative paths, without the need to put the system offline, hence providing the desired reliability. The present memo analyzes two possible approaches to guarantee reliability in NSA domains.

2. Requirements Notation

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] and [RFC8174] when, and only when, they appear in all capitals, as shown here.

3. Potential Solution Approaches

In order to improve the reliability of the system, the pre-requisite is to have redundant links. This means that nodes are connected likely in a meshed fashion, where some of the links are actively used, and others not. In a normal situation, in the context of NSA, the actively used links form a tree. This is the same concept of spanning trees used in layer 2 technologies (e.g. [IEEE802.1W]). When a problem is detected, various possibilities arise in order to logically guarantee connectivity by starting using previously unused links. In the specific case of NSA [I-D.li-6lo-native-short-address], all nodes, except the root, have at least one secondary parent, which is used only if a problem is detected when communicating with the primary one. In this way, when the link toward the primary parent is broken, an alternative link toward the secondary parent can be used. In such context two different approaches can be identified:

* Multi-Address: using multiple addresses per node, one for each alternative parent (logically creating multiple topologies).

* Single-Address: using one single address per node, even if an alternative parent is present.

Both approaches, with pros and cons, are described and analyzed hereafter.

3.1. Multi-Address Approach

In the multi-address case multiple logical topologies are built using different addresses and different links. In the following it is assumed the two logical topologies are built on top of the physical connectivity, however, the principles can be easily extended to more than two topologies.
3.1.1. Topology building

In the multi-address case two root nodes are used. Each root node is the root of a different tree covering all the nodes. Nodes have the same role in both topologies. Meaning that a node will have the same "Leaf" or "Forwarder" role in both topologies (see [I-D.li-6lo-native-short-address] for role definition). The Allocation Function (AF) used to assign addresses in the two parallel topologies might differ, however, attention should be given to guarantee that addresses in the two topologies are different. This can be easily achieved by using two different addresses for the root nodes. Indeed such addresses will be the prefix of the whole tree, which also mean that the address of the root nodes can be used to actually identify the different topologies. For both topologies, the address allocation works in the exact same way as described in [I-D.li-6lo-native-short-address], the only additional action to be taken is that a node cannot choose the same parent node in both topologies. This can be easily achieved by imposing that the two parent must not have the same "node-id".

Let’s make a simple example with the topology depicted in Figure 1, where there are two root nodes, named "R-1" and "R-2" and a set of few nodes with different roles, where "L" stands for leaf nodes and "F" stands for forwarder nodes. Physical links are not depicted in the figure but, as already mentioned, the assumption is that each node is connected at least to two potential parents.

```
+----+     +----+
| R-1 |     | R-2 |
+----+     +----+
          +----+     +----+
          | F-1 |     | L-1 |
          +----+     +----+
          +----+     +----+
| F-3 |     | L-3 |
+----+     +----+
          +----+     +----+
| L-4 |     | L-5 |
+----+     +----+
```

Figure 1: Simple Topology example.
Let’s now assume that R-1 has the address 1, which is used to allocate the address to other nodes. After applying the allocation function presented in [I-D.li-6lo-native-short-address], a possible outcome is the one presented in Figure 2, where the links selected to form the logical topology are shown, as well as the assigned addresses.

```
+----+               +----+
| 1  |----------------+   |R-2|
+----+               +----+
      \                     \
    +----+        +-----+    +-----+
   | 10 |        | 11  |      | 110|      | 111|
   +----+        +-----+    +-----+    +-----+
      \                      \              \
    +----+        +-----+    +-----+    +-----+
   | 100|        | 101|      | 1010|      | 1011|
   +----+        +-----+    +-----+    +-----+
      \                      \              \
    +----+        +-----+    +-----+    +-----+
   | 1001|       | 10011|     | 100111|     | 100111|
   +----+        +-----+    +-----+    +-----+
```

Figure 2: Possible NSA assignment and logical topology using R-1 as root.

In a similar way, assuming root R-2 has the address 01, and again applying the allocation function presented in [I-D.li-6lo-native-short-address], a possible outcome is the one presented in Figure 3, where the links selected to form the logical topology are shown, as well as the assigned addresses.
When everything is working without problem, one of the logical topologies can be used as primary topology, while using the second one only in case of link/node failures. A simple selection can be done for example with the rule:

* Interpreting root nodes’ addresses as integers, choose the tree with the smallest value.

Another approach could be to try to use some load balancing approach, where sockets open on the various nodes are bound to one of the available addresses based on some algorithm. The algorithm can be as simple as a random choice, however it has to be considered, that random local choices can uniformly distribute connections on different addresses, but it does not mean that the traffic is uniformly distributed on the network as a whole [SINGH20]. Such kind of optimization algorithm are out of the scope of this document. In the following let’s assume that a primary/secondary approach is used, where the topology in Figure 2 is the primary one.

In [I-D.li-6lo-native-short-address], the root node has always address 1. Such simple approach allows to easily switch between NSA and IPv6 addresses. Indeed, because all NSA addresses start with a
1, to obtain an IPv6 address it is sufficient to prepend a sufficient number of zeros to the NSA address, so align to the /64 suffix length, then prepend it with the whole /64 prefix of the network. Vice-versa, to obtain the corresponding NSA address from an IPv6 address, the /64 prefix is removed as well as all leading zeros in the remaining suffix. This is not anymore sufficient. Taking the example in Figure 3, the root node has to be aware that it does not have to remove the last leading zero. In order to maintain the simplicity of the design of NSA, the addresses of root nodes are assigned as follows:

* Each root has an address where the least significant bit is set to 1 and all the others to zero.
* Each root has a different address length that has to be known by the root.
* An address length of 1, means no leading zeros.
* An address length of n, means n-1 leading zeros followed by 1.

Coming back to our example, root R-1, has an address length equal to 1, hence its address is "1", as depicted in Figure 2, while R-2 has an address length equal to 2, hence its address is "01", as depicted in Figure 3. Only the root nodes have to be aware of the root address length, because they are the only one performing address translation (from NSA to IPv6 and vice-versa). Leafs and Forwarders do not need to be aware of the root address length, since it is implicit in the prefix used in the tree for the addresses. For instance, the leafs at the bottom of Figure 3 can easily understand that the root address length is 2 because their address starts with 01. The only requirement imposed by this solution on the nodes is to allow addresses that start with zeros ([I-D.li-6lo-native-short-address] only specifies addresses starting with 1).

3.1.2. Link Failures

In case of link failure there are three actions that need to be performed in order to ensure connectivity.

1. The parent node, with respect to the link in object, has to inform all the nodes between itself and the root that a certain sub-tree is not reachable anymore through it, by using the primary topology. This can be achieved by sending an ICMPv6 message toward the root, where each node the message traverses stores the sub-tree unreachability status, so that packets destined to that sub-tree are actually re-directed toward the
root. After this procedure, when a node sees a packet that is destined to the node in the unreachable sub-tree, it sends it up to the root.

2. The child node, with respect to the link in object, has to inform the root that its sub-tree can still be reached, if traffic is sent through the secondary topology using the secondary address of the node that is the root of the sub-tree. This can be achieved by sending an ICMPv6 message toward the root of the primary tree, but actually using the secondary tree, hence using the secondary address as a source of the message. With this operation the root of the primary tree is now aware that to reach a certain sub-tree, traffic has to be sent through the secondary tree to a specific address (the secondary address of the child on the broken link). In order actually ship a packet destined to an address in the primary tree through the secondary tree, two options are possible: encapsulation or routing.

* Encapsulation is pretty simple. Whenever there is a packet destined to the sub-tree with a redirect entry on the primary root, the root encapsulates (tunnels) the packet to the secondary address of the child node of the broken link and sends it to the secondary root. The packet will be forwarded according to the stateless NSA procedure until it reaches the intended node. There, it is decapsulated and the original packet is routed in the sub-tree until its final destination. In the other direction, all packets coming from the sub-tree can be encapsulated toward the primary root, hence being forwarded on the secondary tree, circumventing the broken link.

* Routing relies on some forwarding entries stored on the nodes along the path on the secondary tree. Basically, when the ICMPv6 message, sent by the child node on the broken link, is forwarded on the secondary tree, each node along the path stores the fact that they are part of a forwarding path toward the sub-tree specified in the ICMPv6 message itself. In this way, no additional encapsulation is necessary, since the packet can be forwarded from the primary root to the secondary root, who in turn will forward it to the child from which it received the ICMPv6 message, and so on until the message reaches the sub-tree where it is forwarded using the normal NSA stateless forwarding. In the opposite direction, for packets coming from the sub-tree, nodes along the alternate path on the secondary tree will simply forward the packets to the secondary root, who will forward them to the primary root.
The first solution (encapsulation) may increase the likelihood to have MTU issues. Indeed, an additional encapsulation will increase the packet size. The second solution does not create MTU issues, but needs to store state in nodes along the alternative paths. While the number of entries is certainly limited, because it is the number of sub-trees unreachable through the primary tree and using the node as part of the alternative path. This may be an issue on devices with strong memory constraints. Yet, if the state grows big it is the symptom of massive failures in the network, which may be a far bigger/urgent problem. In both cases the root nodes have to keep some state: the redirection rules for all unreachable sub-trees. This is not a problem since root gateways are usually more powerful than the other nodes and do not run on batteries. However, if the number of entries grows large, this is again a symptom of massive failures.

3. Optionally, for optimization purposes, the child node, with respect to the link in object, may inform all the nodes of its sub-tree that they should start use the secondary tree (i.e. the secondary address). This can be achieved by sending a specific ICMPv6 message to all of its children, who will do the same recursively. In this way communications will take advantage from the stateless forwarding. However, communication using the primary address, with the mechanism describe in the previous points must still be supported, for ongoing communications that would otherwise break and for any communication initiated from the Internet toward and address in the primary tree. For instance because only primary addresses are shared publicly (via DNS or other means).

All of the above-mentioned ICMPv6 messages are forwarded using NSA stateless forwarding procedure.

Using the example previously introduced, let’s assume that the link between F-1 and F-3 breaks (cf. Figure 1). This means that in the primary topology (see Figure 2) the link between nodes 10 and 100 is broken. According the procedure presented above, the following action are taken:

1. 10 sends an ICMPv6 message to the root. Root will register that 100-sub-tree is not reachable through 10 but has to be redirected.

2. 100 sends an ICMPv6 message to 1 (root of primary tree) using 01100 as source address (see Figure 3). This message will be forwarded first to 01, the root of the secondary tree, and then
to 1. Let’s assume encapsulation is used, now root 1 has an entry stating:

* For 100-sub-tree encapsulate to 01100 and forward to 01

3. 100 will send an ICMPv6 message to its children suggesting to use the secondary addresses.

At this point connection is guaranteed. Let’s assume the in the primary tree (see Figure 2) nodes 11 and 1001 where communicating to each other. Packets will flow in the following way:

* From 11 to 1001:

1. Packet is transmitted from 11 to 1 (on the primary tree).

2. Because of the redirect entry, 1 encapsulates packet toward 100 and transmits it to 01 (root secondary).

3. 01 will use NSA stateless forwarding to transmit the packet to 0110 (on the secondary tree).

4. 0110 will use NSA stateless forwarding to transmit the packet to 01100 (on the secondary tree).

5. 01100 will decapsulate, note the destination is on the primary tree, use the NSA stateless forwarding to transmit the packet to 1001 (on the primary tree).

* From 1001 to 11:

1. Packet is transmitted from 1001 to 100 (on the primary tree).

2. Because 100 knows the upstream link is broken it encapsulates the packet with source 01100 and destination 1 (root primary tree) then transmits the packet to 0110 (on the secondary tree).

3. 0110 will use NSA stateless forwarding to transmit the packet to 01 (on the secondary tree).

4. 01 will see that packet is destined to the primary root and transmits it to 1.

5. 1 will decapsulate, note the destination is on the primary tree, use the NSA stateless forwarding to transmit the packet to 11 (on the primary tree).
In case of communication toward/from the public Internet the procedure is similar. For outgoing packets the primary root will expand the NSA header to a full IPv6 header and forward it upstream. For incoming packets, the root will first reduce the IPv6 header to an NSA header then forward it as described above. NSA header expansion and IPv6 header reduction are operations described in [I-D.li-6lo-native-short-address].

3.1.3. Node Failures

In case that an entire node fails, several links will not be usable anymore. Nevertheless, the procedure described in the previous section can be still applied, what may change is who is performing the action. More specifically:

1. The parent of the failed node, has to inform all the nodes between itself the root that a certain sub-tree is not reachable anymore through it. This is the exact same procedure like in Section 3.1.2.

2. All of the children of the failed node, have to independently inform the root that its sub-tree can still be reached if traffic is sent through the secondary topology, using the secondary address of the node that is the root of the sub-tree. This is the exact same procedure like in Section 3.1.2, just done by all children.

3. All of the children of the node, optionally, for optimization purposes, may inform all the nodes of their sub-trees that they should start use the secondary tree (i.e. the secondary address). This is the exact same procedure like in Section 3.1.2, just done by all children.

Using again the example previously introduced, let’s assume that node F-3 fails (cf. Figure 1). This means that in the primary topology (see Figure 2) the links between nodes 10 and 100 is unusable, as well as the links between 100 and its three children, namely 1001, 10011, and 100111. According the procedure presented above, the following action are taken:

1. 10 sends an ICMPv6 message to the root. Root will register that 100-sub-tree is not reachable through 10 but has to be redirected.

2. The three children of 100 will perform the following:
   * 1001 sends an ICMPv6 message to 1 (root of primary tree) using 01100 as source address (see Figure 3). This message will be
forwarded first to 01, the root of the secondary tree, and then to 1. Let’s assume encapsulation is used, now root 1 has an entry stating:

- For 1001-sub-tree encapsulate to 0110101 and forward to 01

* 1001 sends an ICMPv6 message to 1 (root of primary tree) using 01100 as source address (see Figure 3). This message will be forwarded first to 01, the root of the secondary tree, and then to 1. Let’s assume encapsulation is used, now root 1 has an entry stating:

- For 10011-sub-tree encapsulate to 01101011 and forward to 01

* 10011 sends an ICMPv6 message to 1 (root of primary tree) using 01100 as source address (see Figure 3). This message will be forwarded first to 01, the root of the secondary tree, and then to 1. Let’s assume encapsulation is used, now root 1 has an entry stating:

- For 100111-sub-tree encapsulate to 011010111 and forward to 01

3. The children of 100, will send an ICMPv6 message to their children (if any) suggesting to use the secondary addresses.

At this point connection is guaranteed. Let’s assume, like in the example for the link failure, that in the primary tree (see Figure 2) nodes 11 and 1001 where communicating to each other. Packets will flow in the following path:

* From 11 to 1001:

1. Packet is transmitted from 11 to 1 (on the primary tree).

2. Because of the redirect entry, 1 encapsulates packet toward 100 and transmits it to 01 (root secondary).

3. 01 will use NSA stateless forwarding to transmit the packet to 0110101 (on the secondary tree).

4. 0110101 will decapsulate, note the destination is its own primary address, the packet will be decapsulate once more and delivered to the upper layer.

* From 1001 to 11:
1. Because 1001 knows the upstream link is broken it encapsulates the packet with source 0110101 and destination 1 (root primary tree) then, using NSA stateless forwarding, it will transmit the packet to 01 (on the secondary tree).

2. 01 will see that packet is destined to the primary root and transmits it to 1.

3. 1 will decapsulate, note the destination is on the primary tree, use the NSA stateless forwarding to transmit the packet to 11 (on the primary tree).

In case of communication toward/from the public Internet the procedure is the same as described in Section 3.1.2.

3.1.4. Nodes Forwarding Procedure

Nodes, other than leafs, have to forward packets according to the procedures described in the previous sections. Nevertheless, compared to the original specification the modifications are very limited. Hereafter, the forwarding procedure for both forwarder and root nodes is provided. The mention "NSA Native Forwarding" is used where the original procedure described in [I-D.li-6lo-native-short-address] is employed.

3.1.4.1. Forwarder Nodes

As describe in Figure 4, in the context of multiple topologies, when a a forwarder node receives a packet, it needs first to verify if there is any rule that redirects the packet. If it is not the case, it needs to check if there is an encapsulation rule, if it is the case then the packets needs to be encapsulated accordingly. Then normal NSA forwarding is applied.
3.1.4.2. Root Nodes

In the case of a root node, and in the context of multiple topologies, the NSA Native Forwarding is always applied for outward packets. Only in case of inward packets, the node has to check whether there is an encapsulation rule through an alternative topology to bypass a failed link/node. Figure 5 show this simple case.
3.2. Single-Address Approach

In this approach, starting from the root node, we can assign a single address to each node in the NSA network based on the address allocation function described in [I-D.li-6lo-native-short-address]. All nodes with assigned addresses will send a message to the root to register themselves so that the root has an overview of the nodes and the topology in the NSA network. The nodes with the links used to assign the addresses form the primary tree topology. By default, the node forwards the packet via the primary tree by using the native NSA forwarding method defined in [I-D.li-6lo-native-short-address].

The root will have a backup with the same address 1, and Virtual Router Redundancy Protocol (VRRP [RFC5798]) could be used to implement same address root redundancy. In order to provide reliability inside the topology, each node will have at least one alternative parent for redundancy. This alternative uplink is stored added to the already existing Neighbor Discovery table. For a forwarder node, once an alternative downlink is established, because it is an alternative parent, it has to record this downlink into the ND table as well. For leaf nodes, there will be only alternative...
uplink entries. All the alternative links will be reported to the root using ICMPv6 messages. Therefore for forwarder nodes, there will be alternative uplink(s) and alternative downlink(s) stored in the ND table, and leaves nodes will have a ND table only with alternative uplink(s). An example of ND table which includes the alternative parent(s)/children is shown in Figure 6). In particular, it shows what should be the ND table content for node 100 in the topology shown in Figure 7.

<table>
<thead>
<tr>
<th>Destination</th>
<th>Flags</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>I</td>
</tr>
<tr>
<td>10</td>
<td>PP</td>
</tr>
<tr>
<td>1000</td>
<td>PFC</td>
</tr>
<tr>
<td>10010</td>
<td>PFC</td>
</tr>
<tr>
<td>1001</td>
<td>PLC</td>
</tr>
<tr>
<td>10011</td>
<td>PLC</td>
</tr>
<tr>
<td>110</td>
<td>AP</td>
</tr>
<tr>
<td>10100</td>
<td>AFC</td>
</tr>
<tr>
<td>10101</td>
<td>ALC</td>
</tr>
</tbody>
</table>

Figure 6: Example of a ND Table of a forwarder node with address of '100'.

There can be more than one forwarder and leaf children, "Primary" here means that they belong to the primary topology, to differentiate them from the backup alternative role. The first entry of Figure 6 shows the address of the node itself ‘100’. This node’s parent on the primary tree is ‘10’ that is recorded in second entry and marked accordingly a Primary Parent (PP). There are two Primary Forwarder Children (PFC), namely ‘1000’ and ‘10010’, followed by two Primary Leaf Children (PLC), namely ‘1001’ and ‘10011’. Then one alternative parent (AP) follows, namely ‘110’. Finally, for the sake of clarity, two alternative children have been added to complete the table (not depicted in Figure 7), an Alternative Forwarder Child (AFC) with address 10100, and an Alternative Leaf Child (ALC) with address 10101.
As there is only one primary tree, in general, the packet forwarding will follow the normal NSA forwarding method if there is no link or node failure. Even when there are failures on the alternative links, the normal NSA forwarding method is not impacted. However, if there is a link failure on the primary tree, the forwarding behavior will change as described in the following.

3.2.1. Link Failure

Upon a link failure an ICMPv6 message will be generated to report the event to the root. The root will then compute a new forwarding path based on the current state and encapsulate (tunnel) the packet to nodes where broken links could be avoided.

![Diagram of link failure](image)

Figure 7: An example of link failure in single address topology.
In order to give an example to what happens to packets flowing downlink, let’s assume a packet initiated from node 1101 destined to node 1001, and that the link between node 10 and node 100 is broken. When the link fails, upon detection of the failure, node 10 will send an ICMPv6 message to the root, to make it aware of the failure. The packet forwarding will happen as follows:

1. Packet is transmitted from node 1101 to the root 1, using NSA native forwarding.

2. Root 1 is aware that the path to destinations in the 100 sub-tree are not reachable through normal NSA forwarding because of the link failure, hence computes an alternative path. In this example: 1 -> 110 -> 100 -> 1001. Since normal NSA forwarding does not allow to go first through node 110 and then node 100, the root 1 will encapsulate the addresses of node 110 and node 100 in an extension header so to perform segmented routing [I-D.geng-spring-sr-redundancy-protection].

3. Once packet reaches 100, the segment routing extension is dropped, and the packet is sent to its destination 1001 using NSA native forwarding.

In the unlikely case that the root is not yet aware of the link failure during the packet transmission, the packet forwarding will happen as follows:

1. Packet is transmitted from node 1101 to the root 1, using NSA native forwarding.

2. Packet is transmitted from root 1 to node 10, following the normal NSA forwarding method.

3. Node 10, which is aware about the link failure, redirects the packet back to the root with SRv6 encapsulation.

4. Root 1, which should in the meantime have received an ICMPv6 message notifying the failure of the link, receives the encapsulated packet and, after decapsulation, it operates like in the previous example. Since it is now aware that the path to destinations in the 100 sub-tree are not reachable through normal NSA forwarding because of the link failure, hence computes an alternative path. In this example: 1 -> 110 -> 100 -> 1001. Since normal NSA forwarding does not allow to go first through node 110 and then node 100, the root 1 will encapsulate the addresses of node 110 and node 100 in an extension header so to perform segmented routing [I-D.geng-spring-sr-redundancy-protection].
5. Once packet reaches 100, segment routing extension dropped, and packet is sent to its destination 1001 using NSA native forwarding.

Let’s now look at what happens to packets flowing in the opposite direction, when packets are sent from 1001 to 1101, with the same link failed, namely the link between 100 and 10. Upon link failure detection by 100, the node will send an ICMPv6 message through an alternative parent, toward the root, to report the link failure. The packet will be handled as follows:

1. Packet is transmitted from node 1001 to node 100 using NSA native forwarding.
2. Because of the failed link, node 100 sends the packet to an alternative parent node.
3. NSA native forwarding is used then. If the alternative parent is in the same sub-tree like the destination, the packet is forwarded downward to the right child, otherwise it is sent upward to the its own parent. This goes on recursively until the packet reaches the root in the worst case, where it is then sent downward to the correct forwarder child, until it reaches the destination. In our example the path would be: 100 -> 110 -> 1101.

3.2.2. Node Failure

As for the multiple-address case, a node failure can be seen as multiple link failures, basically all links the node connects to. In this case the parent of the failed node and its children will simply apply the same procedure described in the previous section.

3.2.3. Node Forwarding Procedure

3.2.3.1. Forwarder Node
As describe in Figure 8, in the context of single-address approach, when a forwarder node receives a packet, it performs the normal NSA native forwarding (after decapsulation, if needed). If case of link failure, the forwarder will take different actions depending on downlink or uplink failure, as depicted in the Section 3.2.1.

3.2.3.2. Root Node

In the case of a root node, and in the context of single-address approach, the NSA native forwarding is always applied, for outward packets. Only in case of inward packets, the node has to check whether there is a redirection needed. If it is the case, it will compute the path and define the segment routing header in order to forward the packet to avoid the broken link(s).
4. Links/Nodes Failure Detection and Recovery

Previous sections describe actions and possible solution to failures events, but never discussed how failures are detected. This memo assumes that depending on the specific technology in use and the level of desired reliability, the most suitable failure detection mechanism is used to trigger the above-described actions. It is considered not desirable to define one single failure detection technique to be used in the context of NSA, neither to define new ones.

The link failure could be detected leveraging layer 2 feedback, like for instance the lack of acknowledgement upon packet transmission. It can also be detected using existing network layer solution, like for instance using Bidirectional Forwarding Detection (BFD [RFC7130]) or IPv6 specific mechanisms [RFC5534].

Another aspect of the general failure management is to recover from failures, going back to the original state. In the context of NSA there are a couple of possible approaches that can be used. Use
native addresses lifetime. Addresses can be assigned associated with a lifetime. When such lifetime expires, node have to undergo the same initial procedure address allocation. This is also a good moment to check whether a certain link or node is back to normal functioning. If it is not the case, the algorithmic procedure will anyway create topologies that do not take into account failed links/nodes. A faster approach could be based, like in the case of failure detection, on periodic checks that may leverage on layer 2 features or on some neighbor discovery messages. The former method being more effective, the latter introducing communication overhead.

5. Robustness

Real robustness provided by the different approaches depends on the specific topology. The single-address solution may introduce more state. Indeed, the root has the overview of the NSA network. It knows all nodes’ addresses, the alternative links and the broken links. It is able to compute a usable path towards a destination. This comes with the benefit of potentially being able to find a higher number of alternative path, hence, in the end providing a stronger protection against multiple failures. The forwarder node and the leaf node are rather dummy and use NSA stateless forwarding. They only are aware of link state toward their direct neighbors, and take action accordingly. The multi-address approach leverages more on the stateless forwarding of NSA. The root is in general unaware of nodes’ addresses, and the network topology. In case of failure, a redirection rule is set on the root, hence there amount of state is proportional to the number of failures. This means less state, but may be less robust to multiple failures. Differently from the single address solution, a small amount state is also required on forwarder nodes, because if a link fails a redirect rule has to be used.

The above mentioned pros and cons need to be pondered when choosing a reliability solution to be deployed in an NSA domain.

6. Security Considerations

TBD

7. IANA Considerations

TBD.

8. References

8.1. Normative References
8.2. Informative References

[I-D.daniel-6lowpan-hilow-hierarchical-routing]
Park, S. D., "Hierarchical Routing over 6LoWPAN (HiLow)",
Work in Progress, Internet-Draft, draft-daniel-6lowpan-
hilow-hierarchical-routing-01, 18 June 2007,
<https://www.ietf.org/archive/id/draft-daniel-6lowpan-
hilow-hierarchical-routing-01.txt>.

[I-D.geng-spring-sr-redundancy-protection]
Geng, X., Chen, M., Yang, F., Garvia, P. C., and G.
Mishra, "SRv6 for Redundancy Protection",
Work in Progress, Internet-Draft, draft-geng-spring-sr-redundancy-
protection-05, 2 August 2021,
[IEEE802.1W]


Authors’ Addresses

Guangpeng Li
Huawei Technologies
Beiqing Road, Haidian District
Beijing
100095
China

Email: liguangpeng@huawei.com

David Lou
Huawei Technologies
Riesstrasse 25
80992 Munich
Germany

Email: zhe.lou@huawei.com

Luigi Iannone
Huawei Technologies France S.A.S.U.
18, Quai du Point du Jour
92100 Boulogne-Billancourt
France