6Lo Working Group
Internet-Draft
Intended status: Standards Track
Expires: May 4, 2017

C. Gomez
S. Darroudi
UPC/i2cat
T. Savolainen
Nokia

October 31, 2016

# IPv6 Mesh over Bluetooth(R) Low Energy using IPSP
# draft-gomez-6lo-blemesh-02

## Abstract

RFC 7668 describes the adaptation of 6LoWPAN techniques to enable IPv6 over Bluetooth low energy networks that follow the star topology. However, recent Bluetooth specifications allow the formation of extended topologies as well. This document specifies the mechanisms needed to enable IPv6 over mesh networks composed of Bluetooth low energy links established by using the Bluetooth Internet Protocol Support Profile.

## 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 http://datatracker.ietf.org/drafts/current/.

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

This Internet-Draft will expire on May 4, 2017.

## Copyright Notice

Table of Contents



## 1. Introduction

Bluetooth low energy (hereinafter, Bluetooth LE) was first introduced in the Bluetooth 4.0 specification. Bluetooth LE (which has been marketed as Bluetooth Smart) is a low-power wireless technology designed for short-range control and monitoring applications. Bluetooth LE is currently implemented in a wide range of consumer electronics devices, such as smartphones and wearable devices. Given the high potential of this technology for the Internet of Things, the Bluetooth Special Interest Group (Bluetooth SIG) and the IETF have produced specifications in order to enable IPv6 over Bluetooth LE, such as the Internet Protocol Support Profile (IPSP) [IPSP], and RFC 7668, respectively. Bluetooth 4.0 only supports Bluetooth LE networks that follow the star topology. In consequence, RFC 7668 was specifically developed and optimized for that type of network topology. However, subsequent Bluetooth specifications allow the formation of extended topologies [BTCorev4.1], such as the mesh topology. The functionality described in RFC 7668 is not sufficient and would fail to enable IPv6 over mesh networks composed of Bluetooth LE links. This document specifies the mechanisms needed to enable IPv6 over mesh networks composed of Bluetooth LE links. This specification also allows to run IPv6 over Bluetooth LE star topology

networks, albeit without all the topology-specific optimizations contained in RFC 7668.

### 1.1. Terminology and Requirements Language

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

The terms 6LoWPAN Node (6LN), 6LoWPAN Router (6LR) and 6LoWPAN Border Router (6LBR) are defined as in [RFC6775], with an addition that Bluetooth LE central and Bluetooth LE peripheral (see Section 2) can both be adopted by a 6LN, a 6LR or a 6LBR.

## 2. Bluetooth LE Networks and the IPSP

Bluetooth LE defines two Generic Access Profile (GAP) roles of relevance herein: the Bluetooth LE central role and the Bluetooth LE peripheral role. A device in the central role, which is called central from now on, has traditionally been able to manage multiple simultaneous connections with a number of devices in the peripheral role, called peripherals hereinafter. Bluetooth 4.1 introduced the possibility for a peripheral to be connected to more than one central simultaneously, therefore allowing extended topologies beyond the star topology for a Bluetooth LE network. In addition, a device may simultaneously be a central in a set of link layer connections, as well as a peripheral in others. On the other hand, the IPSP enables discovery of IP-enabled devices and the establishment of a link layer connection for transporting IPv6 packets. The IPSP defines the Node and Router roles for devices that consume/originate IPv6 packets and for devices that can route IPv6 packets, respectively. Consistently with Bluetooth 4.1, a device may implement both roles simultaneously.

This document assumes a mesh network composed of Bluetooth LE links, where link layer connections have been established between neighboring IPv6-enabled devices. The IPv6 forwarding devices of the mesh have to implement both Node and Router roles, while simpler leaf-only nodes can implement only the Node role. In an IPv6-enabled mesh of Bluetooth LE links, a node is a neighbor of another node, and vice versa, if a link layer connection has been established between both by using the IPSP functionality for discovery and link layer connection establishment for IPv6 packet transport.

## 3. Specification of IPv6 mesh over Bluetooth LE networks

### 3.1. Protocol stack

Figure 1 illustrates the protocol stack for IPv6 mesh over Bluetooth LE networks. There are two main differences with the IPv6 over Bluetooth LE stack in RFC 7668: a) the adaptation layer below IPv6 (labelled as "6Lo for mesh of Bluetooth LE") is now adapted for mesh networks of Bluetooth LE links, and b) the protocol stack for IPv6 mesh networks of Bluetooth LE links includes IPv6 routing functionality.

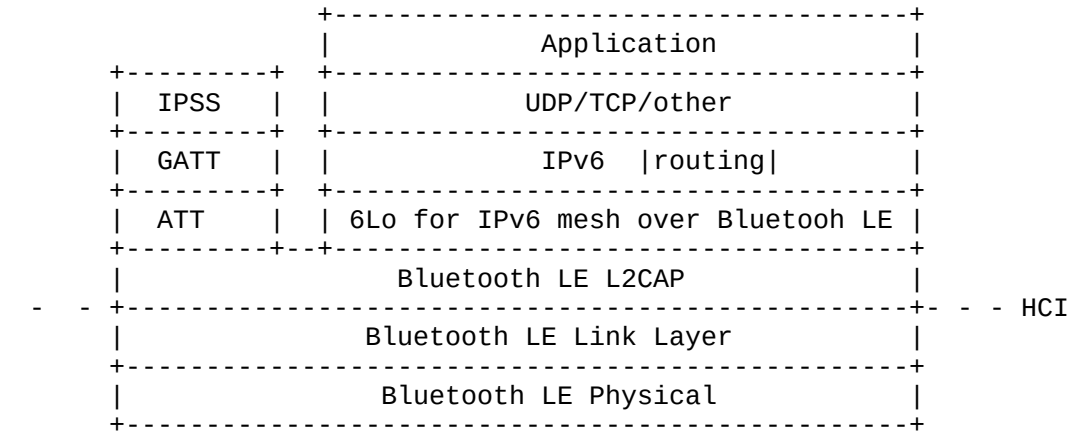
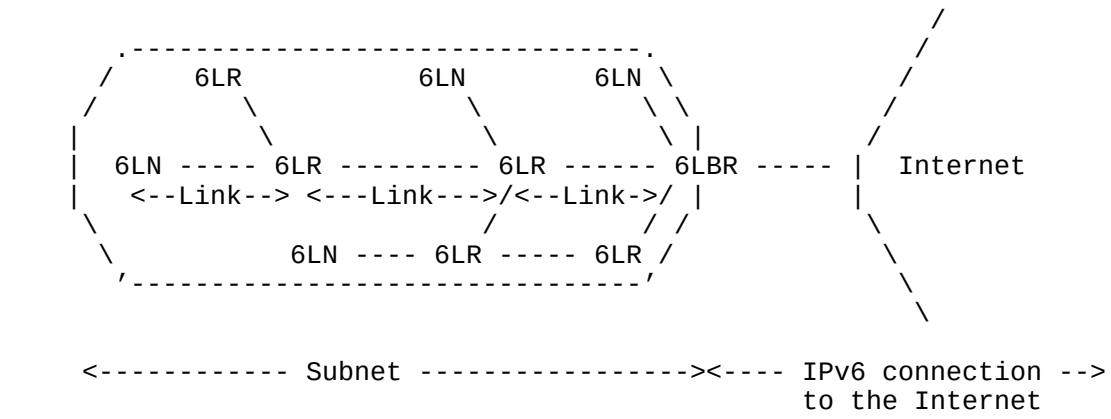



Figure 1: Protocol stack for IPv6 mesh over Bluetooth LE.

### 3.2. Subnet model

For IPv6 mesh over Bluetooth LE, a multilink model has been chosen, as further illustrated in Figure 2. As IPv6 over Bluetooth LE is intended for constrained nodes, and for Internet of Things use cases and environments, the complexity of implementing a separate subnet on each peripheral-central link and routing between the subnets appears to be excessive. In this specification, the benefits of treating the collection of point-to-point links between a central and its connected peripherals as a single multilink subnet rather than a multiplicity of separate subnets are considered to outweigh the multilink model’s drawbacks as described in [RFC4903].




to the Internet

Figure 2: Example of an IPv6 mesh over a Bluetooth LE network connected to the Internet

One or more 6LBRs are connected to the Internet. 6LNs are connected to the network through a 6LR or a 6LBR. A prefix is used on the whole subnet.

IPv6 mesh networks over Bluetooth LE MUST follow a route-over approach. This document does not specify the routing protocol to be used in an IPv6 mesh over Bluetooth LE.

## 3.3. Link model

### 3.3.1. Stateless address autoconfiguration

6LN, 6LR and 6LBR IPv6 addresses in an IPv6 mesh over Bluetooth LE are configured as per section 3.2.2 of RFC 7668.

Multihop DAD functionality as defined in section 8.2 of RFC 6775, or some substitute mechanism (see section 3.3.2), MUST be supported.

### 3.3.2. Neighbor Discovery

'Neighbor Discovery Optimization for IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs)' [RFC6775] describes the neighbor discovery approach as adapted for use in several 6LoWPAN topologies, including the mesh topology. The route-over functionality of RFC 6775 MUST be supported.

The following aspects of the Neighbor Discovery optimizations [RFC6775] are applicable to Bluetooth LE 6LNs:

1. A Bluetooth LE 6LN MUST NOT register its link-local address. A Bluetooth LE 6LN MUST register its non-link-local addresses with its routers by sending a Neighbor Solicitation (NS) message with the Address Registration Option (ARO) and process the Neighbor Advertisement (NA) accordingly. The NS with the ARO option MUST be sent irrespective of the method used to generate the IID. The ARO option requires use of an EUI-64 identifier [RFC6775]. In the case of Bluetooth LE, the field SHALL be filled with the 48-bit device address used by the Bluetooth LE node converted into 64-bit Modified EUI-64 format [RFC4291].

If the 6LN registers for a same compression context multiple addresses that are not based on Bluetooth device address, the header compression efficiency will decrease.

2. For sending Router Solicitations and processing Router Advertisements the Bluetooth LE 6LNs MUST, respectively, follow Sections 5.3 and 5.4 of the [RFC6775].

3. The router behavior for 6LRs and 6LBRs is described in Section 6 of RFC 6775. However, as per this specification, routers SHALL NOT use multicast NSs to discover other routers' link layer addresses.

4. Border router behavior is described in Section 7 of RFC 6775.

RFC 6775 defines substitutable mechanisms for distributing prefixes and context information (section 8.1 of RFC 6775), as well as for Duplicate Address Detection across a route-over 6LoWPAN (section 8.2 of RFC 6775). Implementations of this specification MUST support the features described in sections 8.1 and 8.2 of RFC 6775 unless some alternative ("substitute") from some other specification is supported.

### 3.3.3. Header compression

Header compression as defined in RFC 6282 [RFC6282], which specifies the compression format for IPv6 datagrams on top of IEEE 802.15.4, is REQUIRED as the basis for IPv6 header compression on top of Bluetooth LE. All headers MUST be compressed according to RFC 6282 [RFC6282] encoding formats.

To enable efficient header compression, when the 6LBR sends a Router Advertisement it MUST include a 6LoWPAN Context Option (6CO) [RFC6775] matching each address prefix advertised via a Prefix Information Option (PIO) [RFC4861] for use in stateless address autoconfiguration.

The specific optimizations of RFC 7668 for header compression, which exploit the star topology and ARO, cannot be generalized in a mesh network composed of Bluetooth LE links. Still, a subset of those optimizations can be applied in some cases in such a network. In particular, the latter comprise link-local interactions, non-link-local packet transmissions originated and performed by a 6LN, and non-link-local packet transmissions originated by a 6LN neighbor and sent to a 6LN. For the rest of packet transmissions, context-based compression MAY be used.

When a device transmits a packet to a neighbor, the sender MUST fully elide the source IID if the source IPv6 address is the link-local address based on the sender's Bluetooth device address (SAC=0, SAM=11). The sender also MUST fully elide the destination IPv6 address if it is the link-local-address based on the neighbor's Bluetooth device address (DAC=0, DAM=11).

When a 6LN transmits a packet, with a non-link-local source address that the 6LN has registered with ARO in the next-hop router for the indicated prefix, the source address MUST be fully elided if it is the latest address that the 6LN has registered for the indicated prefix (SAC=1, SAM=11). If the source non-link-local address is not the latest registered by the 6LN, then the 64-bits of the IID SHALL be fully carried in-line (SAC=1, SAM=01) or if the first 48-bits of the IID match with the latest address registered by the 6LN, then the last 16-bits of the IID SHALL be carried in-line (SAC=1, SAM=10).

When a router transmits a packet to a neighboring 6LN, with a non-link-local destination address, the router MUST fully elide the destination IPv6 address if the destination address is the latest registered by the 6LN with ARO for the indicated context (DAC=1, DAM=11). If the destination address is a non-link-local address and not the latest registered, then the 6LN MUST either include the IID part fully in-line (DAM=01) or, if the first 48-bits of the IID match to the latest registered address, then elide those 48-bits (DAM=10).

### 3.3.4. Unicast and multicast mapping

The Bluetooth LE Link Layer does not support multicast. Hence, traffic is always unicast between two Bluetooth LE neighboring nodes. If a node needs to send a multicast packet to several neighbors, it has to replicate the packet and unicast it on each link. However, this may not be energy efficient, and particular care must be taken if the node is battery powered. A router (i.e. a 6LR or a 6LBR) MUST keep track of neighboring multicast listeners, and it MUST NOT forward multicast packets to neighbors that have not registered as listeners for multicast groups the packets belong to.

## 4. IANA Considerations

There are no IANA considerations related to this document.

## 5. Security Considerations

The security considerations in RFC 7668 apply.

IPv6 mesh networks over Bluetooth LE require a routing protocol to find end-to-end paths. Unfortunately, the routing protocol may generate additional opportunities for threats and attacks to the network.

RFC 7416 [RFC 7416] provides a systematic overview of threats and attacks on the IPv6 Routing Protocol for Low-Power and Lossy Networks (RPL), as well as countermeasures. In that document, described threats and attacks comprise threats due to failures to authenticate, threats due to failure to keep routing information, threats and attacks on integrity, and threats and attacks on availability. Reported countermeasures comprise confidentiality attack, integrity attack, and availability attack countermeasures.

While this specification does not state the routing protocol to be used in IPv6 mesh over Bluetooth LE networks, the guidance of RFC 7416 is useful when RPL is used in such scenarios. Furthermore, such guidance may partly apply for other routing protocols as well.

## 6. Acknowledgements



The authors of this document are grateful to all RFC 7668 authors, since this document borrows many concepts (albeit, with necessary extensions) from RFC 7668.

The authors also thank Alain Michaud, Mark Powell and Martin Turon for their comments, which helped improve the document.

Carles Gomez has been supported in part by the Spanish Government Ministerio de Economia y Competitividad through project TEC2012-32531, and FEDER.

## 7. References

### 7.1. Normative References

[BTCorev4.1] Bluetooth Special Interest Group, "Bluetooth Core Specification Version 4.1", December 2013, <https://www.bluetooth.org/en-us/specification/adopted-specifications>.

[IPSP] Bluetooth Special Interest Group, "Bluetooth Internet Protocol Support Profile Specification Version 1.0.0", December 2014, <https://www.bluetooth.org/en-us/specification/adopted-specifications>.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <http://www.rfc-editor.org/info/rfc2119>.

[RFC4291] Hinden, R. and S. Deering, "IP Version 6 Addressing Architecture", RFC 4291, DOI 10.17487/RFC4291, February 2006, <http://www.rfc-editor.org/info/rfc4291>.

[RFC4861] Narten, T., Nordmark, E., Simpson, W., and H. Soliman, "Neighbor Discovery for IP version 6 (IPv6)", RFC 4861, DOI 10.17487/RFC4861, September 2007, <http://www.rfc-editor.org/info/rfc4861>.

[RFC6282] Hui, J., Ed. and P. Thubert, "Compression Format for IPv6 Datagrams over IEEE 802.15.4-Based Networks", RFC 6282, DOI 10.17487/RFC6282, September 2011, <http://www.rfc-editor.org/info/rfc6282>.

[RFC6775] Shelby, Z., Ed., Chakrabarti, S., Nordmark, E., and C. Bormann, "Neighbor Discovery Optimization for IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs)", RFC 6775, DOI 10.17487/RFC6775, November 2012, <http://www.rfc-editor.org/info/rfc6775>.

[RFC7668] Nieminen, J., Savolainen, T., Isomaki, M., Patil, B., Shelby, Z., and C. Gomez, "IPv6 over BLUETOOTH(R) Low Energy", RFC 7668, DOI 10.17487/RFC7668, October 2015, <http://www.rfc-editor.org/info/rfc7668>.

### 7.2. Informative References

[RFC4903] Thaler, D., "Multi-Link Subnet Issues", RFC 4903, DOI 10.17487/RFC4903, June 2007, <http://www.rfc-editor.org/info/rfc4903>.

[RFC7416] Tsao, T., Alexander, R., Dohler, M., Daza, V., Lozano, A., and M. Richardson, Ed., "A Security Threat Analysis for the Routing Protocol for Low-Power and Lossy Networks (RPLs)", RFC 7416, DOI 10.17487/RFC7416, January 2015, <http://www.rfc-editor.org/info/rfc7416>.

## Authors' Addresses

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

Email: carlesgo@entel.upc.edu

Seyed Mahdi Darroudi
Universitat Politecnica de Catalunya/Fundacio i2cat
C/Esteve Terradas, 7
Castelldefels 08860
Spain

Email: sm.darroudi@entel.upc.edu

Teemu Savolainen
Nokia Technologies
Hatanpaan valtatie 30
Tampere 33100
Finland

Email: teemu.savolainen@nokia.com

6Lo Working Group
Internet-Draft
Intended status: Standards Track
Expires: May 3, 2017

Y-G. Hong
ETRI
C. Gomez
UPC/i2cat
Y-H. Choi
ETRI
D-Y. Ko
SKtelecom

October 30, 2016

# IPv6 over Constrained Node Networks(6lo) Applicability & Use cases draft-hong-6lo-use-cases-03

## Abstract

This document describes the applicability of IPv6 over constrained node networks (6lo) and use cases. It describes the practical deployment scenarios of 6lo technologies with the consideration of 6lo link layer technologies and identifies the requirements. In addition to IEEE 802.15.4, various link layer technologies such as ITU-T G.9959 (Z-Wave), BLE, DECT-ULE, MS/TP, NFC, LTE MTC, and IEEE 802.15.4e(6tisch) are widely used at constrained node networks for typical services. Based on these link layer technologies, IPv6 over networks of resource-constrained nodes has various and practical use cases. To efficiently implement typical services, the applicability and consideration of several design space dimensions are described.

## 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 http://datatracker.ietf.org/drafts/current/.

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

This Internet-Draft will expire on May 3, 2017.

Table of Contents

## 1. Introduction

Running IPv6 on constrained node networks has different features from general node networks due to the characteristics of constrained node networks such as small packet size, short link-layer address, low bandwidth, network topology, low power, low cost, and large number of devices [RFC4919]. For example, because some IEEE 802.15.4 link layers have a frame size of 127 octets and IPv6 requires the layer below to support an MTU of 1280 bytes, an appropriate fragmentation and reassembly adaptation layer must be provided at the layer of below IPv6. Also, the limited size of IEEE 802.15.4 frame and low energy consumption requirements make the need for header compression. IETF 6lowpan (IPv6 over Low powerWPAN) working group published, an adaptation layer for sending IPv6 packets over IEEE 802.15.4 [RFC4944], compression format for IPv6 datagrams over IEEE 802.15.4-based networks [RFC6282], and Neighbor Discovery Optimization for 6lowpan [RFC6775].

As IoT (Internet of Things) services become more popular, various link layer technologies 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 LTE Machine Type Communication are actively used. And the transmission of IPv6 packets over these link layer technologies is required. A number of IPv6-over-foo documents have been developed in the IETF 6lo (IPv6 over Networks of Resource-constrained Nodes) and 6tisch (IPv6 over the TSCH mode of IEEE 802.15.4e) working groups.

In the 6lowpan working group, the [RFC6568], "Design and Application Spaces for 6LoWPANs" was published and it describes potential application scenarios and use cases for low-power wireless personal area networks. In this document, various design space dimensions such as deployment, network size, power source, connectivity, multi-hop communication, traffic pattern, security level, mobility, and QoS were analyzed. And it described a fundamental set of 6lowpan application scenarios and use cases: Industrial monitoring-Hospital storage rooms, Structural monitoring-Bridge safety monitoring, Connected home-Home Automation, Healthcare-Healthcare at home by tele-assistance, Vehicle telematics-telematics, and Agricultural monitoring-Automated vineyard.

Even though the [RFC6568] describes some potential application scenarios and use cases and it lists the design space in the context of 6lowpan, it does not cover the different use cases and design space in the context of the 6lo working group. The RFC6568 assumed that the link layer technology is the IEEE802.15.4 and the described application scenarios and use cases were based on the IEEE 802.15.4

technologies. Due to various link layer technologies such as ITU-T G.9959 (Z-Wave), BLE, DECT-ULE, MS/TP, NFC, LTE MTC, and IEEE 802.15.4e(6tisch), potential application scenarios and use cases of 6lo will go beyond the RFC6568.

This document provides the applicability and use cases of 6lo, considering the following:

- 6lo applicability and use cases MAY be uniquely different from those of 6lowpan.
- 6lo applicability and use cases SHOULD cover various IoT related wire/wireless link layer technologies providing practical information of such technologies.
- 6lo applicability and use cases SHOULD describe characteristics and typical use cases of each link layer technology, and then 6lo use cases's applicability.

## 2. Conventions and Terminology

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

## 3. 6lo Link layer technologies

### 3.1. ITU-T G.9959

The ITU-T G.9959 recommendation [G.9959] targets low-power Personal Area Networks (PANs). 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].

### 3.2. Bluetooth Low Energy

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

Devices such as mobile phones, notebooks, tablets and other handheld computing devices which will include Bluetooth 4.1 chipsets will probably also have the low-energy variant of Bluetooth. Bluetooth LE will also be included in many different types of accessories that collaborate with mobile devices such as phones, tablets and notebook computers. 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].

### 3.3. 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 PHY layer operating at frequencies in the 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 MAC layer supports traditional DECT as this is used for services like discovery, pairing, security features etc. 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 [I-D.ietf-6lo-dect-ule].

### 3.4. Master-Slave/Token-Passing

MS/TP is a contention-free access method for the RS-485 physical layer, which is used extensively in building automation networks.

An MS/TP device is typically based on a low-cost microcontroller with limited processing power and memory. Together with low data rates and a small address space, these constraints are similar to those

faced in 6LoWPAN networks and suggest some elements of that solution might be leveraged. MS/TP differs significantly from 6LoWPAN in at least three respects: a) MS/TP devices typically have a continuous source of power, b) all MS/TP devices on a segment can communicate directly so there are no hidden node or mesh routing issues, and c) recent changes to MS/TP provide support for large payloads, eliminating the need for link-layer fragmentation and reassembly.

MS/TP is designed to enable multidrop networks over shielded twisted pair wiring. It can support a data rate of 115,200 baud on segments up to 1000 meters in length, or segments up to 1200 meters in length at lower baud rates. An MS/TP link requires only a UART, an RS-485 transceiver with a driver that can be disabled, and a 5ms resolution timer. These features make MS/TP a cost-effective field bus for the most numerous and least expensive devices in a building automation network [I-D.ietf-6lo-6lobac].

### 3.5. 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). NFC can be compatible with existing contactless card infrastructure and it enables a consumer to utilize one device across different systems.

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 ideal for establishing connections with other technologies by the simplicity of touch. In addition to the easy connection and quick transactions, simple data sharing is also available [I-D.ietf-6lo-nfc].

### 3.6. LTE MTC

LTE category defines the overall performance and capabilities of the UE(User Equipment). For example, the maximum down rate of category 1 UE and category 2 UE are 10.3 Mbit/s and 51.0 Mbit/s respectively. There are many categories in LTE standard. 3GPP standards defined the category 0 to be used for low rate IoT service in release 12. Since

category 1 and category 0 could be used for low rate IoT service, these categories are called LTE MTC (Machine Type Communication) [LTE_MTC].

LTE MTC have the advantages compared to above category 2 to be used for low rate IoT service such as low power and low cost.

The below figure shows the primary characteristics of LTE MTC.

| Category | Max. Date Rate Down | Max. Date Rate Up |
|---|---|---|
| Category 0 | 1.0 Mbit/s | 1.0 Mbit/s |
| Category 1 | 10.3 Mbit/s | 5.2 Mbit/s |

Table 1: Primary characteristics of LTE MTC

## 3.7. IEEE 802.15.4e

The Timeslotted Channel Hopping (TSCH) mode was introduced in the IEEE 802.15.4-2015 standard. In a TSCH network, all nodes are synchronized. Time is sliced up into timeslots. The duration of a timeslot, typically 10ms, is large enough for a node to send a full-sized frame to its neighbor, and for that neighbor to send back an acknowledgment to indicate successful reception. Timeslots are grouped into one of more slotframes, which repeat over time.

All the communication in the network is orchestrated by a communication schedule which indicates to each node what to do in each of the timeslots of a slotframe: transmit, listen or sleep. The communication schedule can be built so that the right amount of link-layer resources (the cells in the schedule) are scheduled to satisfy the communication needs of the applications running on the network, while keeping the energy consumption of the nodes very low. Cells can be scheduled in a collision-free way, introducing a high level of determinism to the network.

A TSCH network exploits channel hopping: subsequent packets exchanged between neighbor nodes are done so on a different frequency. This means that, if a frame isn’t received, the transmitter node will re-transmitt the frame on a different frequency. The resulting "channel hopping" efficiently combats external interference and multi-path fading.

The main benefits of IEEE 802.15.4 TSCH are:

- ultra high reliability. Off-the-shelf commercial products offer over 99.999% end-to-end reliability.

- ultra low-power consumption. Off-the-shelf commercial products offer over a decade of battery lifetime.

## 4. 6lo Deployment Scenarios

In this clause, we will describe some 6lo deployment scenrios such as Smart Grid activity in WiSun

[TBD]

## 5. Design Space

The [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 the RFC 6568, the following design space dimensions are described; Deployment, Network size, Power source, Connectivity, Multi-hop communication, Traffic pattern, Mobility, Quality of Service (QoS).

The design space dimensions of 6lo are a little different from those of the RFC 6568 due to the different characteristics of 6lo link layer technologies. The following design space dimensions can be 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: Originally, the link layer technologies of 6lo have low rate of data transmission. But, by adjusting the MTU, it can deliver higher data rate.

- Buffering requirements: Some 6lo use case may require more data rate than the link layer technology support. In this case, a buffering mechanism to manage the data is required.

- Security 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 is dependent on the 6lo use case. If the 6lo nodes can move or moved around, it requires a mobility management mechanism.

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

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

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

- Security Bootstrapping: Without the external operations, 6lo nodes must have the 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 [I-D.ietf-lwig-energy-efficient]. Readers are expected to be familiar with RFC 7228 terminology.

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

# 6. 6lo Use Cases

## 6.1. 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 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

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, etc.) 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, specially 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. a light is turned off). A reaction to a user interaction will be perceived by the user as immediate as long as the reaction takes place after less than 0.5 seconds [RFC5826].

Dominant parameters in home automation scenarios with ITU-T G.9959:

- Deployment/Bootstrapping: Pre-planned.
- Topology: Mesh topology.
- L2-mesh or L3-mesh: ITU-T G.9959 provides support for L2-mesh, and L3-mesh can also be used (the latter requires an IP-based routing protocol).
- Multi-link subnet, single subnet: Multi-link subnet.
- Data rate: Small data rate, infrequent transmissions.

- Buffering requirements: Low requirement.
- Security requirements: Data privacy and security must be provided. Encryption is required.
- Mobility: Most devices are static. A few devices (e.g. remote control) are portable.
- Time Synchronization: TBD.
- Reliability and QoS: Moderate to high level of reliability support. Actions as a result of human-generated traffic should occur after less than 0.5 seconds.
- Traffic patterns: Periodic (sensor readings) and aperiodic (user-triggered interaction).
- Security Bootstrapping: Required.
- Power use strategy: Mix of P1 (Low-power) devices and P9 (Always-on) devices.
- Update firmware requirements: TBD.

### 6.2. Use case of Bluetooth Low Energy: Smartphone-Based Interaction with Constrained Devices

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: 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, etc.), 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. 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.

Dominant parameters in fitness scenarios with Bluetooth LE:

- Deployment/Bootstrapping: Pre-planned.
- Topology: Star topology.
- L2-mesh or L3-mesh: No.
- Multi-link subnet, single subnet: Multi-link subnet.
- Data rate: TBD.
- Buffering requirements: Low requirement.
- Security requirements: For health-critical information, data privacy and security must be provided. Encryption is required. Some types of notifications sent by the smartphone may not need.
- Mobility: Low.
- Time Synchronization: the link layer, which is based on TDMA, provides a basis for time synchronization.
- Reliability and QoS: a relatively low ratio of message losses is acceptable for periodic sensor readings. End-to-end latency of sensor readings should be low for critical notifications or alarms, generated by either the smartphone or an Internet cloud service.
- Traffic patterns: periodic (sensor readings) and aperiodic (smartphone-generated notifications).
- Security Bootstrapping: Required.
- Power use strategy: P1 (Low-power) devices.
- Update firmware requirements: TBD.

### 6.3. 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. Therefore, DECT-ULE is specially suitable for the connected home space in application areas such as home automation, smart metering, safety, healthcare, etc.

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.

Dominant parameters in smart metering scenarios with DECT-ULE:

- Deployment/Bootstrapping: Pre-planned.
- Topology: Star topology.
- L2-mesh or L3-mesh: No.
- Multi-link subnet, single subnet: Multi-link subnet.
- Data rate: Small data rate, infrequent transmissions.
- Buffering requirements: Low requirement.
- Security requirements: Data privacy and security must be provided. Encryption is required.
- Mobility: No.
- Time Synchronization: TBD.
- Reliability and QoS: bounded latency, stringent reliability service agreements [I-D.ietf-roll-applicability-ami].

- o Traffic patterns: Periodic (meter reading notifications sent by the meter) and aperiodic (user- or company-triggered queries to the meter, and messages triggered by local events such as power outage or leak detection [I-D.ietf-roll-applicability-ami]).

- o Security Bootstrapping: required.

- o Power use strategy: P0 (Normally-off) for devices with long sleep intervals (i.e. greater than ~10 seconds) which then may need to resynchronize again, and P1 (Low-power) for short sleep intervals. P9 (Always-on) for the Fixed Part (FP), which is the central node in the star topology.

- o Update firmware requirements: TBD.

### 6.4. Use case of MS/TP:

[TBD]

Example: [TBD]

- o Power use strategy: P9 (Always-on).

### 6.5. Use case of NFC: Alternative Secure Transfer

According to applications, various secured data can be handled and transferred. Depending on security level of the data, methods for transfer can be alternatively selected. The personal data having serious issues should be transferred securely, but data transfer by using Wi-Fi and Bluetooth connections cannot always be secure because of their a little long radio frequency range. Hackers can overhear the personal data transfer behind hidden areas. Therefore, methods need to be alternatively selected to transfer secured data. Voice and video data, which are not respectively secure and requires long transmission range, can be transferred by 3G/4G technologies, such as WCDMA, GSM, and LTE. Big size data, which are not secure and requires high speed and broad bandwidth, can be transferred by Wi-Fi and wired network technologies. However, the personal data, which pose serious issues if mishandled while transferred in wireless domain, can be securely transferred by NFC technology. It has very short frequency range - nearly single touch communication.

Example: Secure Transfer by Using NFC in Healthcare Services with Tele-Assistance

A senior citizen who lives alone wears one to several wearable 6lo devices to measure heartbeat, pulse rate, etc. The 6lo devices are densely installed at home for movement detection. An LoWPAN Border

Router (LBR) at home will send the sensed information to a connected healthcare center. Portable base stations with LCDs 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. At this moment, hidden hackers can overhear the data based on the LTE connection, but they cannot gather the personal data over the NFC connection.

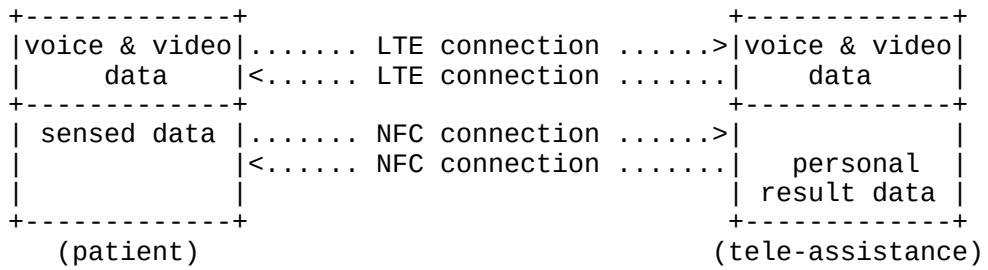



Figure 1: Alternative Secure Transfer in Healthcare Services

Dominant parameters in secure transfer by using NFC in healthcare services:

- Deployment/Bootstrapping: Pre-planned. MP2P/P2MP (data collection), P2P (local diagnostic).
- Topology: Small, NFC-enabled device connected to the Internet.
- L2-mesh or L3-mesh: NFC does not support L2-mesh, L3-mesh can be configured.
- Multi-link subnet, single subnet: a single hop for gateway; patient’s body network is mesh topology.
- Data rate: Small data rate.
- Buffering requirements: Low requirement.
- Security requirements: Data privacy and security must be provided. Encryption is required.

- Mobility: Moderate (patient’s mobility).
- Time Synchronization: Highly required.
- Reliability and QoS: High level of reliability support (life-or-death implication), role-based.
- Traffic patterns: Short data length and periodic (randomly).
- Security Bootstrapping: Highly required.
- Other Issues: Plug-and-play configuration is required for mainly non-technical end-users. Real-time data acquisition and analysis are important. Efficient data management is needed for various devices that have different duty cycles, and for role-based data control. Reliability and robustness of the network are also essential.
- Power use strategy: TBD.
- Update firmware requirements: TBD.

### 6.6. Use case of LTE MTC

Wireless link layer technologies can be divided into short range connectivity and long range connectivity. BLE, ITU-T G.9959 (Z-Wave), DECT-ULE, MS/TP, NFC are used for short range connectivity. LTE MTC is used for long range connectivity. And there is another long range connectivity technology. It is LPWAN (Low Power Wide Area Network) technology such as LoRa, Sigfox, etc. Therefore, the use case of LTE MTC could be used in LPWAN.

Example: Use of wireless backhaul for LoRa gateway

LoRa is one of the most promising technologies of LPWAN. LoRa network architecture has a star of star topology. LoRa gateway relay the messages from LoRa end device to application server and vice versa. LoRa gateway can has two types of backhaul, wired and wireless backhaul.

If LoRa gateway has wireless backhaul, it should have LTE modem. Since the modem cost of LTE MTC is cheaper than the modem cost of above LTE category 2, it is helpful to design to use LTE MTC. Since the maximum date rate of LoRa end device is 50kbps, it is sufficient to use LTE MTC without using category 2.

Dominant parameters in LoRa gateway scenarios with above example:

- Deployment/Bootstrapping: Pre-planned.
- Topology: Star topology.
- L2-mesh or L3-mesh: No.
- Multi-link subnet, single subnet: Single subnet.
- Data rate: depends on 3GPP specification.
- Buffering requirements: High requirement.
- Security requirements: No, because data security is already provided in LoRa specification.
- Mobility: Static.
- Time Synchronization: Highly required.
- Reliability and QoS: TBD.
- Traffic patterns: Random.
- Security Bootstrapping: required.
- Power use strategy: P9 (Always-on).
- Update firmware requirements: TBD.

Example: Use of controlling car

Car sharing services are becoming more popular. Customers wish to control the car with smart phone application. For example, customers wish to lock/unlock the car door with smart phone application, because customers may not have a car key. Customers wish to blow with smart phone application to locate the car easily.

Therefore, rental car should have a long range connectivity capable modem such as LoRa end device and LTE UE. However, LoRa may not be used because LoRa has low reliability and may not be supported in an indoor environment such as a basement parking lot. And since message size for car control is very small, it is sufficient to use LTE MTC but category 2.

Dominant parameters in controlling car scenarios with above example:

- Deployment/Bootstrapping: Pre-planned.

- Topology: Star topology.
- L2-mesh or L3-mesh: No.
- Multi-link subnet, single subnet: Single subnet.
- Data rate: depends on 3GPP specification.
- Buffering requirements: High requirement.
- Security requirements: High requirement.
- Mobility: Always dynamic .
- Time Synchronization: Highly required.
- Reliability and QoS: TBD.
- Traffic patterns: Random.
- Security Bootstrapping: required.
- Power use strategy: P1 (Low-power).

### 6.7. Use case of IEEE 802.15.4e:

[TBD]

Example: [TBD]

## 7. IANA Considerations

There are no IANA considerations related to this document.

## 8. Security Considerations

[TBD]

## 9. Acknowledgements

Carles Gomez has been funded in part by the Spanish Government (Ministerio de Educacion, Cultura y Deporte) through the Jose Castillejo grant CAS15/00336. 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.

Samita Chakrabarti, Thomas Watteyne, Pascal Thubert, Abdur Rashid Sangi, Xavier Vilajosana, Daniel Migault, and Take Aanstoot have provided valuable feedback for this draft.

## 10. References

### 10.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <http://www.rfc-editor.org/info/rfc2119>.

[RFC4919] Kushalnagar, N., Montenegro, G., and C. Schumacher, "IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs): Overview, Assumptions, Problem Statement, and Goals", RFC 4919, DOI 10.17487/RFC4919, August 2007, <http://www.rfc-editor.org/info/rfc4919>.

[RFC4944] Montenegro, G., Kushalnagar, N., Hui, J., and D. Culler, "Transmission of IPv6 Packets over IEEE 802.15.4 Networks", RFC 4944, DOI 10.17487/RFC4944, September 2007, <http://www.rfc-editor.org/info/rfc4944>.

[RFC5826] Brandt, A., Buron, J., and G. Porcu, "Home Automation Routing Requirements in Low-Power and Lossy Networks", RFC 5826, DOI 10.17487/RFC5826, April 2010, <http://www.rfc-editor.org/info/rfc5826>.

[RFC6282] Hui, J., Ed. and P. Thubert, "Compression Format for IPv6 Datagrams over IEEE 802.15.4-Based Networks", RFC 6282, DOI 10.17487/RFC6282, September 2011, <http://www.rfc-editor.org/info/rfc6282>.

[RFC6568] Kim, E., Kaspar, D., and JP. Vasseur, "Design and Application Spaces for IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs)", RFC 6568, DOI 10.17487/RFC6568, April 2012, <http://www.rfc-editor.org/info/rfc6568>.

[RFC6775] Shelby, Z., Ed., Chakrabarti, S., Nordmark, E., and C. Bormann, "Neighbor Discovery Optimization for IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs)", RFC 6775, DOI 10.17487/RFC6775, November 2012, <http://www.rfc-editor.org/info/rfc6775>.

[RFC7228] Bormann, C., Ersue, M., and A. Keranen, "Terminology for Constrained-Node Networks", RFC 7228, DOI 10.17487/RFC7228, May 2014, <http://www.rfc-editor.org/info/rfc7228>.

[RFC7428] Brandt, A. and J. Buron, "Transmission of IPv6 Packets over ITU-T G.9959 Networks", RFC 7428, DOI 10.17487/RFC7428, February 2015, <http://www.rfc-editor.org/info/rfc7428>.

[RFC7668] Nieminen, J., Savolainen, T., Isomaki, M., Patil, B., Shelby, Z., and C. Gomez, "IPv6 over BLUETOOTH(R) Low Energy", RFC 7668, DOI 10.17487/RFC7668, October 2015, <http://www.rfc-editor.org/info/rfc7668>.

## 10.2. Informative References

[I-D.ietf-6lo-dect-ule]
Mariager, P., Petersen, J., Shelby, Z., Logt, M., and D. Barthel, "Transmission of IPv6 Packets over DECT Ultra Low Energy", draft-ietf-6lo-dect-ule-07 (work in progress), October 2016.

[I-D.ietf-6lo-6lobac]
Lynn, K., Martocci, J., Neilson, C., and S. Donaldson, "Transmission of IPv6 over MS/TP Networks", draft-ietf-6lo-6lobac-05 (work in progress), June 2016.

[I-D.ietf-6lo-nfc]
Choi, Y., Youn, J., and Y. Hong, "Transmission of IPv6 Packets over Near Field Communication", draft-ietf-6lo-nfc-05 (work in progress), October 2016.

[I-D.ietf-lwig-energy-efficient]
Gomez, C., Kovatsch, M., Tian, H., and Z. Cao, "Energy-Efficient Features of Internet of Things Protocols", draft-ietf-lwig-energy-efficient-05 (work in progress), October 2016.

[I-D.ietf-roll-applicability-ami]
Cam-Winget, N., Hui, J., and D. Popa, "Applicability Statement for the Routing Protocol for Low Power and Lossy Networks (RPL) in AMI Networks", draft-ietf-roll-applicability-ami-15 (work in progress), October 2016.

[G.9959] "International Telecommunication Union, "Short range narrow-band digital radiocommunication transceivers - PHY and MAC layer specifications", ITU-T Recommendation", January 2015.

[LTE_MTC] "3GPP TS 36.306 V13.0.0, 3rd Generation Partnership Project; Technical Specification Group Radio Access Network; Evolved Universal Terrestrial Radio Access (E-UTRA); User Equipment (UE) radio access capabilities (Release 13)", December 2015.

Authors' Addresses

Yong-Geun Hong
ETRI
161 Gajeong-Dong Yuseung-Gu
Daejeon 305-700
Korea

Phone: +82 42 860 6557
Email: yghong@etri.re.kr

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 305-700
Korea

Phone: +82 42 860 1429
Email: yhc@etri.re.kr

Deoknyong Ko
SKtelecom
9-1 Byundang-gu Sunae-dong, Seongnam-si
Gyeonggi-do 13595
Korea

Phone: +82 10 3356 8052
Email: engineer@sk.com

6Lo Working Group
Internet-Draft
Intended status: Standards Track
Expires: September 11, 2017

K. Lynn, Ed.
Verizon Labs
J. Martocci
Johnson Controls
C. Neilson
Delta Controls
S. Donaldson
Honeywell
March 10, 2017

# Transmission of IPv6 over MS/TP Networks
# draft-ietf-6lo-6lobac-08

## Abstract

Master-Slave/Token-Passing (MS/TP) is a medium access control method for the RS-485 physical layer and is used primarily in building automation networks. This specification defines the frame format for transmission of IPv6 packets and the method of forming link-local and statelessly autoconfigured IPv6 addresses on MS/TP 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 http://datatracker.ietf.org/drafts/current/.

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

This Internet-Draft will expire on September 11, 2017.

## Copyright Notice

Table of Contents



## 1. Introduction

Master-Slave/Token-Passing (MS/TP) is a medium access control (MAC) protocol for the RS-485 [TIA-485-A] physical layer and is used primarily in building automation networks. This specification defines the frame format for transmission of IPv6 [RFC2460] packets and the method of forming link-local and statelessly autoconfigured IPv6 addresses on MS/TP networks. The general approach is to adapt elements of the 6LoWPAN specifications [RFC4944], [RFC6282], and [RFC6775] to constrained wired networks, as noted below.

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.

The following sections provide a brief overview of MS/TP, then describe how to form IPv6 addresses and encapsulate IPv6 packets in MS/TP frames. This specifcation (subsequently referred to as "6LoBAC") includes a REQUIRED header compression mechanism that is based on LOWPAN_IPHC [RFC6282] and improves MS/TP link utilization.

## 1.1. Requirements Language

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

## 1.2. Abbreviations Used

ASHRAE: American Society of Heating, Refrigerating, and Air-Conditioning Engineers (http://www.ashrae.org)

BACnet: An ISO/ANSI/ASHRAE Standard Data Communication Protocol for Building Automation and Control Networks

CRC: Cyclic Redundancy Code

MAC: Medium Access Control

MSDU: MAC Service Data Unit (MAC client data)

MTU: Maximum Transmission Unit; the size of the largest network layer protocol data unit that can be communicated in a single network transaction

UART: Universal Asynchronous Transmitter/Receiver

## 1.3. MS/TP Overview

This section provides a brief overview of MS/TP, as specified in ANSI/ASHRAE Standard 135-2016 [BACnet] Clause 9. The latest version of [BACnet] integrates changes to legacy MS/TP (approved as [Addendum_an]) that provide support for larger frame sizes and improved error handling. [BACnet] Clause 9 also covers physical layer deployment options.

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 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.

The differential signaling used by [TIA-485-A] requires a contention-free MAC. MS/TP uses a token to control access to a multidrop bus. Only an MS/TP master node can initiate the unsolicited transfer of data, and only when it holds the token. After sending at most a configured maximum number of data frames, a master node passes the token to the next master node (as determined by MAC address). If present on the link, legacy MS/TP implementations (including any slave nodes) ignore the frame format defined in this specification.

[BACnet] Clause 9 defines a range of Frame Type values used to designate frames that contain data and data CRC fields encoded using Consistent Overhead Byte Stuffing [COBS] (see Appendix B). The purpose of COBS encoding is to eliminate preamble sequences from the Encoded Data and Encoded CRC-32K fields. The Encoded Data is covered by a 32-bit CRC [CRC32K] (see Appendix C) that is also COBS encoded.

MS/TP COBS-encoded frames have the following format:

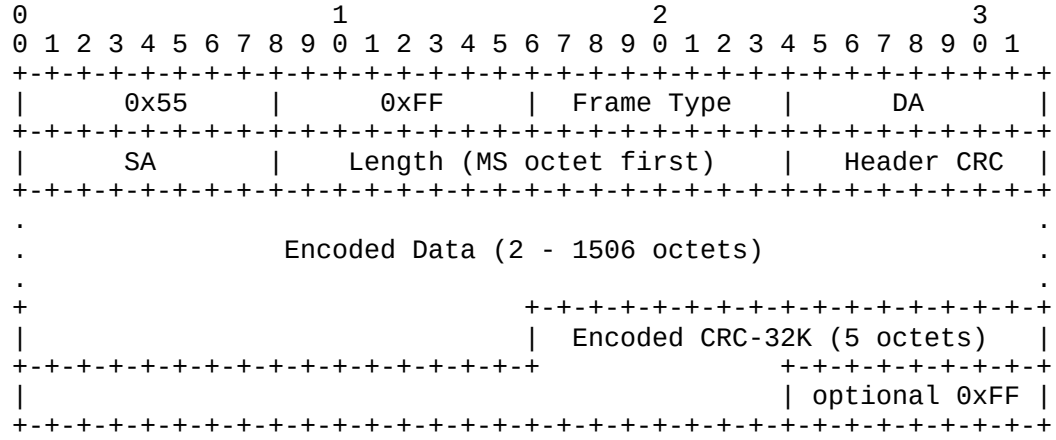



Figure 1: MS/TP COBS-Encoded Frame Format

MS/TP COBS-encoded frame fields are defined as follows:

| | |
|---|---|
| Preamble | two octet preamble: 0x55, 0xFF |
| Frame Type | one octet |
| Destination Address | one octet address |
| Source Address | one octet address |
| Length | two octets, most significant octet first |
| Header CRC | one octet |
| Encoded Data | 2 - 1506 octets (see Section 4 and Appendix B) |
| Encoded CRC-32K | five octets (see Appendix C) |
| (pad) | (optional) at most one octet of trailer: 0xFF |

The Frame Type is used to distinguish between different types of MAC frames. The types relevant to this specification (in decimal) are:

```
   0  Token
   1  Poll For Master
   2  Reply To Poll For Master
   3  Test_Request
   4  Test_Response
      ...
  34  IPv6 over MS/TP (LoBAC) Encapsulation
```

Frame Types 8 - 31 and 35 - 127 are reserved for assignment by ASHRAE. Frame Types 32 - 127 designate COBS-encoded frames that convey Encoded Data and Encoded CRC-32K fields. See Section 2 for additional details.

The Destination and Source Addresses are each one octet in length. See Section 3 for additional details.

For COBS-encoded frames, the Length field indicates the size of the [COBS] Encoded Data field in octets, plus three. (This adjustment is required in order for legacy MS/TP devices to ignore COBS-encoded frames.) See Section 4 and Appendices for additional details.

The Header CRC field covers the Frame Type, Destination Address, Source Address, and Length fields. The Header CRC generation and check procedures are specified in [BACnet] Annex G.1.

Use of the optional 0xFF trailer octet is discussed in [BACnet] Clause 9.

### 1.4. Goals and Constraints

The main goals of this specification are a) to enable IPv6 directly on wired end devices in building automation and control networks by leveraging existing standards to the greatest extent possible, and b) to co-exist with legacy MS/TP implementations. Co-existence allows MS/TP networks to be incrementally upgraded to support IPv6.

In order to co-exist with legacy devices, no changes are permitted to the MS/TP addressing modes, frame header format, control frames, or Master Node state machine as specified in [BACnet] Clause 9.

## 2. Profile for IPv6 over MS/TP

ASHRAE has assigned an MS/TP Frame Type value of 34 to indicate IPv6 over MS/TP (LoBAC) Encapsulation. This falls within the range of values that designate COBS-encoded data frames.

### 2.1. Mandatory Features

[BACnet] Clause 9 specifies mandatory to implement features of MS/TP devices. E.g., it is mandatory that all MS/TP nodes respond to a Test_Request with a Test_Response frame. All MS/TP master nodes must implement the Master Node state machine and handle Token, Poll For Master, and Reply to Poll For Master control frames. 6LoBAC nodes are MS/TP master nodes that implement a Receive Frame state machine capable of handling COBS-encoded frames.

6LoBAC nodes must support a data rate of 115.2 kbit/s and may support lower data rates as specified in [BACnet] Clause 9. The method of selecting the data rate is outside the scope of this specification.

### 2.2. Configuration Constants

The following constants are used by the Receive Frame state machine.

Nmin_COBS_length The minimum valid Length value of any LoBAC encapsulated frame: 5

Nmax_COBS_length The maximum valid Length value of any LoBAC encapsulated frame: 1509

### 2.3. Configuration Parameters

The following parameters are used by the Master Node state machine.

Nmax_info_frames The default maximum number of information frames the node may send before it must pass the token: 1

Nmax_master The default highest allowable address for master nodes: 127

The mechanisms for setting parameters or monitoring MS/TP performance are outside the scope of this specification.

## 3. Addressing Modes

MS/TP node (MAC) addresses are one octet in length and assigned dynamically. The method of assigning MAC addresses is outside the scope of this specification. However, each MS/TP node on the link MUST have a unique address in order to ensure correct MAC operation.

[BACnet] Clause 9 specifies that addresses 0 through 127 are valid for master nodes. The method specified in Section 6 for creating a MAC-address-derived Interface Identifier (IID) ensures that an IID of all zeros can never be generated.

A Destination Address of 255 (all nodes) indicates a MAC-layer broadcast. MS/TP does not support multicast, therefore all IPv6 multicast packets MUST be broadcast at the MAC layer and filtered at the IPv6 layer. A Source Address of 255 MUST NOT be used.

Hosts learn IPv6 prefixes via router advertisements according to [RFC4861].

## 4. Maximum Transmission Unit (MTU)

Upon transmission, the network layer MTU is formatted according to Section 5 and becomes the MAC service data unit (MSDU). The MSDU is then COBS encoded by MS/TP. Upon reception, the steps are reversed. [BACnet] Clause 9 supports MSDUs up to 2032 octets in length.

IPv6 [RFC2460] requires that every link in the internet have an MTU of 1280 octets or greater. Additionally, a node must be able to accept a fragmented packet that, after reassembly, is as large as 1500 octets. This specification defines an MTU length of at least 1280 octets and at most 1500 octets. Support for an MTU length of 1500 octets is RECOMMENDED.

## 5. LoBAC Adaptation Layer

This section specifies an adaptation layer to support compressed IPv6 headers as specified in Section 10. IPv6 header compression MUST be implemented on all nodes. Implementations MAY also support Generic Header Compression [RFC7400] for transport layer headers.

The LoBAC encapsulation format defined in this section describes the MSDU of an IPv6 over MS/TP frame. The LoBAC payload (i.e., an IPv6 packet) follows an encapsulation header stack. LoBAC is a subset of the LoWPAN encapsulation defined in [RFC4944] as updated by [RFC6282] so the use of "LOWPAN" in literals below is intentional. The primary difference between LoWPAN and LoBAC encapsulation is omission of the Mesh, Broadcast, Fragmentation, and LOWPAN_HC1 headers in the latter.

All LoBAC encapsulated datagrams transmitted over MS/TP are prefixed by an encapsulation header stack consisting of a Dispatch value followed by zero or more header fields. The only sequence currently defined for LoBAC is the LOWPAN_IPHC header followed by payload, as shown below:


Figure 2: A LoBAC Encapsulated LOWPAN_IPHC Compressed IPv6 Datagram

The Dispatch value is treated as an unstructured namespace. Only a single pattern is used to represent current LoBAC functionality.

| Pattern | Header Type |
|---|---|
| 01 1xxxxx | LOWPAN_IPHC - LOWPAN_IPHC compressed IPv6 [RFC6282] |

Figure 3: LoBAC Dispatch Value Bit Pattern

Other IANA-assigned 6LoWPAN Dispatch values do not apply to 6LoBAC unless otherwise specified.

## 6. Stateless Address Autoconfiguration

This section defines how to obtain an IPv6 Interface Identifier. This specification distinguishes between two types of IID, MAC-address-derived and semantically opaque.

A MAC-address-derived IID is the RECOMMENDED type for use in forming a link-local address, as it affords the most efficient header compression provided by the LOWPAN_IPHC [RFC6282] format specified in Section 10. The general procedure for creating a MAC-address-derived IID is described in [RFC4291] Appendix A, "Creating Modified EUI-64 Format Interface Identifiers", as updated by [RFC7136].

The Interface Identifier for link-local addresses SHOULD be formed by concatenating the node’s 8-bit MS/TP MAC address to the seven octets 0x00, 0x00, 0x00, 0xFF, 0xFE, 0x00, 0x00. For example, an MS/TP MAC address of hexadecimal value 0x4F results in the following IID:

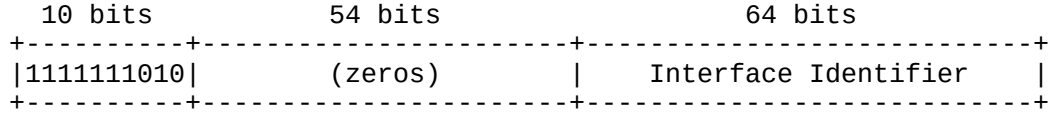
| 0 1 | 1 3 | 3 4 | 4 6 |
|---|---|---|---|
| 0 5 | 6 1 | 2 7 | 8 3 |
| 0000000000000000 | 0000000011111111 | 1111111000000000 | 0000000001001111 |

A semantically opaque IID having 64 bits of entropy is RECOMMENDED for each globally scoped address and MAY be locally generated according to one of the methods cited in Section 12. A node that generates a 64-bit semantically opaque IID MUST register the IID with its local router(s) by sending a Neighbor Solicitation (NS) message with the Address Registration Option (ARO) and process Neighbor Advertisements (NA) according to [RFC6775].

An IPv6 address prefix used for stateless autoconfiguration [RFC4862] of an MS/TP interface MUST have a length of 64 bits.

## 7. IPv6 Link Local Address

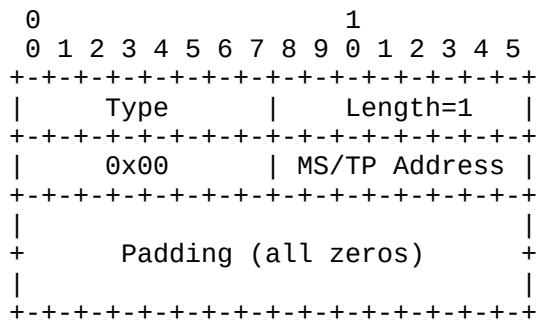
The IPv6 link-local address [RFC4291] for an MS/TP interface is formed by appending the Interface Identifier, as defined above, to the prefix FE80::/64.




## 8. Unicast Address Mapping

The address resolution procedure for mapping IPv6 non-multicast addresses into MS/TP MAC-layer addresses follows the general description in Section 7.2 of [RFC4861], unless otherwise specified.

The Source/Target Link-layer Address option has the following form when the addresses are 8-bit MS/TP MAC-layer (node) addresses.




Option fields:

Type:

1: for Source Link-layer address.

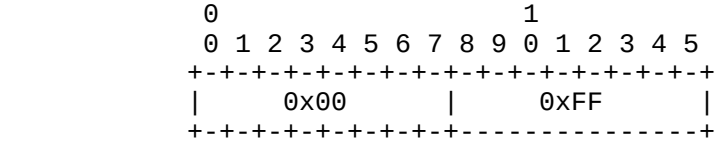
2: for Target Link-layer address.

Length: This is the length of this option (including the type and length fields) in units of 8 octets. The value of this field is 1 for 8-bit MS/TP MAC addresses.

MS/TP Address: The 8-bit address in canonical bit order [RFC2469]. This is the unicast address the interface currently responds to.

## 9. Multicast Address Mapping

All IPv6 multicast packets MUST be sent to MS/TP Destination Address 255 (broadcast) and filtered at the IPv6 layer. When represented as a 16-bit address in a compressed header (see Section 10), it MUST be formed by padding on the left with a zero octet:




## 10. Header Compression

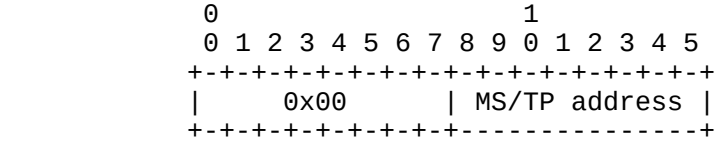
6LoBAC REQUIRES LOWPAN_IPHC IPv6 compression, which is specified in [RFC6282] and included herein by reference. This section will simply identify substitutions that should be made when interpreting the text of [RFC6282].

In general the following substitutions should be made:

- Replace instances of "6LoWPAN" with "MS/TP network"

- Replace instances of "IEEE 802.15.4 address" with "MS/TP address"

When a 16-bit address is called for (i.e., an IEEE 802.15.4 "short address") it MUST be formed by padding the MS/TP address to the left with a zero octet:




If LOWPAN_IPHC compression [RFC6282] is used with context, the router(s) directly attached to the MS/TP segment MUST disseminate the 6LoWPAN Context Option (6CO) according to [RFC6775], Section 7.2.

## 11. IANA Considerations

This document uses values previously reserved by [RFC4944] and [RFC6282] and makes no further requests of IANA.

Note to RFC Editor: this section may be removed upon publication.

## 12. Security Considerations

See [RFC8065] for a general discussion of privacy threats faced by constrained nodes.

[RFC8065] makes a distinction between "stable" and "temporary" addresses. The former are long-lived and typically advertised by servers. The latter are typically used by clients and SHOULD be changed frequently to mitigate correlation of activities over time. Nodes that engage in both activities SHOULD support simultaneous use of multiple addresses per device.

Globally scoped addresses that contain MAC-address-derived IIDs may expose a network to address scanning attacks. For this reason, it is RECOMMENDED that a 64-bit semantically opaque IID be generated for each globally scoped address in use according to, for example, [RFC3315], [RFC3972], [RFC4941], [RFC5535], or [RFC7217].

## 13. Acknowledgments

We are grateful to the authors of [RFC4944] and members of the IETF 6LoWPAN working group; this document borrows liberally from their work. Ralph Droms and Brian Haberman provided indispensable guidance and support from the outset. Peter van der Stok, James Woodyatt, Carsten Bormann, and Dale Worley provided detailed reviews. Stuart Cheshire invented the very clever COBS encoding. Michael Osborne made the critical observation that encoding the data and CRC32K fields separately would allow the CRC to be calculated on-the-fly. Alexandru Petrescu, Brian Frank, Geoff Mulligan, and Don Sturek offered valuable comments.

## 14. References

### 14.1. Normative References

[BACnet] American Society of Heating, Refrigerating, and Air-Conditioning Engineers, "BACnet - A Data Communication Protocol for Building Automation and Control Networks", ANSI/ASHRAE Standard 135-2016, January 2016, <http://www.techstreet.com/ashrae/standards/ashrae-135-2016?product_id=1918140#jumps>.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <http://www.rfc-editor.org/info/rfc2119>.

[RFC2460] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", RFC 2460, DOI 10.17487/RFC2460, December 1998, <http://www.rfc-editor.org/info/rfc2460>.

[RFC3315] Droms, R., Ed., Bound, J., Volz, B., Lemon, T., Perkins, C., and M. Carney, "Dynamic Host Configuration Protocol for IPv6 (DHCPv6)", RFC 3315, DOI 10.17487/RFC3315, July 2003, <http://www.rfc-editor.org/info/rfc3315>.

[RFC3972] Aura, T., "Cryptographically Generated Addresses (CGA)", RFC 3972, DOI 10.17487/RFC3972, March 2005, <http://www.rfc-editor.org/info/rfc3972>.

[RFC4291] Hinden, R. and S. Deering, "IP Version 6 Addressing Architecture", RFC 4291, DOI 10.17487/RFC4291, February 2006, <http://www.rfc-editor.org/info/rfc4291>.

[RFC4861] Narten, T., Nordmark, E., Simpson, W., and H. Soliman, "Neighbor Discovery for IP version 6 (IPv6)", RFC 4861, DOI 10.17487/RFC4861, September 2007, <http://www.rfc-editor.org/info/rfc4861>.

[RFC4862] Thomson, S., Narten, T., and T. Jinmei, "IPv6 Stateless Address Autoconfiguration", RFC 4862, DOI 10.17487/RFC4862, September 2007, <http://www.rfc-editor.org/info/rfc4862>.

[RFC4941] Narten, T., Draves, R., and S. Krishnan, "Privacy Extensions for Stateless Address Autoconfiguration in IPv6", RFC 4941, DOI 10.17487/RFC4941, September 2007, <http://www.rfc-editor.org/info/rfc4941>.

[RFC4944] Montenegro, G., Kushalnagar, N., Hui, J., and D. Culler, "Transmission of IPv6 Packets over IEEE 802.15.4 Networks", RFC 4944, DOI 10.17487/RFC4944, September 2007, <http://www.rfc-editor.org/info/rfc4944>.

[RFC5535] Bagnulo, M., "Hash-Based Addresses (HBA)", RFC 5535, DOI 10.17487/RFC5535, June 2009, <http://www.rfc-editor.org/info/rfc5535>.

[RFC6282] Hui, J., Ed. and P. Thubert, "Compression Format for IPv6 Datagrams over IEEE 802.15.4-Based Networks", RFC 6282, DOI 10.17487/RFC6282, September 2011, <http://www.rfc-editor.org/info/rfc6282>.

[RFC6775] Shelby, Z., Ed., Chakrabarti, S., Nordmark, E., and C. Bormann, "Neighbor Discovery Optimization for IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs)", RFC 6775, DOI 10.17487/RFC6775, November 2012, <http://www.rfc-editor.org/info/rfc6775>.

[RFC7136] Carpenter, B. and S. Jiang, "Significance of IPv6 Interface Identifiers", RFC 7136, DOI 10.17487/RFC7136, February 2014, <http://www.rfc-editor.org/info/rfc7136>.

[RFC7217] Gont, F., "A Method for Generating Semantically Opaque Interface Identifiers with IPv6 Stateless Address Autoconfiguration (SLAAC)", RFC 7217, DOI 10.17487/RFC7217, April 2014, <http://www.rfc-editor.org/info/rfc7217>.

[RFC7400] Bormann, C., "6LoWPAN-GHC: Generic Header Compression for IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs)", RFC 7400, DOI 10.17487/RFC7400, November 2014, <http://www.rfc-editor.org/info/rfc7400>.

## 14.2. Informative References

[Addendum_an] American Society of Heating, Refrigerating, and Air-Conditioning Engineers, "ANSI/ASHRAE Addenda an, at, au, av, aw, ax, and az to ANSI/ASHRAE Standard 135-2012, BACnet - A Data Communication Protocol for Building Automation and Control Networks", July 2014, <https://www.ashrae.org/File%20Library/docLib/StdsAddenda/07-31-2014_135_2012_an_at_au_av_aw_ax_az_Final.pdf>.

[COBS] Cheshire, S. and M. Baker, "Consistent Overhead Byte Stuffing", IEEE/ACM TRANSACTIONS ON NETWORKING, VOL.7, NO.2 , April 1999, <http://www.stuartcheshire.org/papers/COBSforToN.pdf>.

[CRC32K] Koopman, P., "32-Bit Cyclic Redundancy Codes for Internet Applications", IEEE/IFIP International Conference on Dependable Systems and Networks (DSN 2002) , June 2002, <https://users.ece.cmu.edu/~koopman/networks/dsn02/dsn02_koopman.pdf>.

[IEEE.802.3_2012] IEEE, "802.3-2012", IEEE 802.3-2012, DOI 10.1109/ieeestd.2012.6419735, January 2013, <http://ieeexplore.ieee.org/servlet/opac?punumber=6419733>.

[RFC2469] Narten, T. and C. Burton, "A Caution On The Canonical Ordering Of Link-Layer Addresses", RFC 2469, DOI 10.17487/RFC2469, December 1998, <http://www.rfc-editor.org/info/rfc2469>.

[RFC8065] Thaler, D., "Privacy Considerations for IPv6 Adaptation-Layer Mechanisms", RFC 8065, DOI 10.17487/RFC8065, February 2017, <http://www.rfc-editor.org/info/rfc8065>.

[TIA-485-A] Telecommunications Industry Association, "TIA-485-A, Electrical Characteristics of Generators and Receivers for Use in Balanced Digital Multipoint Systems (ANSI/TIA/EIA-485-A-98) (R2003)", March 2003, <https://global.ihs.com/doc_detail.cfm?item_s_key=00032964>.

## Appendix A. Abstract MAC Interface

This Appendix is informative and not part of the standard.

[BACnet] Clause 9 provides support for MAC-layer clients through its SendFrame and ReceivedDataNoReply procedures. However, it does not define a network-protocol independent abstract interface for the MAC. This is provided below as an aid to implementation.

### A.1. MA-DATA.request

#### A.1.1. Function

This primitive defines the transfer of data from a MAC client entity to a single peer entity or multiple peer entities in the case of a broadcast address.

#### A.1.2. Semantics of the Service Primitive

The semantics of the primitive are as follows:

```
MA-DATA.request (
                 destination_address,
                 source_address,
                 data,
                 type
                )
```

The ’destination_address’ parameter may specify either an individual or a broadcast MAC entity address. It must contain sufficient information to create the Destination Address field (see Section 1.3) that is prepended to the frame by the local MAC sublayer entity. The

'source_address' parameter, if present, must specify an individual MAC address. If the source_address parameter is omitted, the local MAC sublayer entity will insert a value associated with that entity.

The 'data' parameter specifies the MAC service data unit (MSDU) to be transferred by the MAC sublayer entity. There is sufficient information associated with the MSDU for the MAC sublayer entity to determine the length of the data unit.

The 'type' parameter specifies the value of the MS/TP Frame Type field that is prepended to the frame by the local MAC sublayer entity.

#### A.1.3. When Generated

This primitive is generated by the MAC client entity whenever data shall be transferred to a peer entity or entities. This can be in response to a request from higher protocol layers or from data generated internally to the MAC client, such as a Token frame.

#### A.1.4. Effect on Receipt

Receipt of this primitive will cause the MAC entity to insert all MAC specific fields, including Destination Address, Source Address, Frame Type, and any fields that are unique to the particular media access method, and pass the properly formed frame to the lower protocol layers for transfer to the peer MAC sublayer entity or entities.

### A.2. MA-DATA.indication

#### A.2.1. Function

This primitive defines the transfer of data from the MAC sublayer entity to the MAC client entity or entities in the case of a broadcast address.

#### A.2.2. Semantics of the Service Primitive

The semantics of the primitive are as follows:

```
  MA-DATA.indication (
                      destination_address,
                      source_address,
                      data,
                      type
                     )
```

The 'destination_address' parameter may be either an individual or a

broadcast address as specified by the Destination Address field of the incoming frame. The 'source_address' parameter is an individual address as specified by the Source Address field of the incoming frame.

The 'data' parameter specifies the MAC service data unit (MSDU) as received by the local MAC entity. There is sufficient information associated with the MSDU for the MAC sublayer client to determine the length of the data unit.

The 'type' parameter is the value of the MS/TP Frame Type field of the incoming frame.

#### A.2.3. When Generated

The MA_DATA.indication is passed from the MAC sublayer entity to the MAC client entity or entities to indicate the arrival of a frame to the local MAC sublayer entity that is destined for the MAC client. Such frames are reported only if they are validly formed, received without error, and their destination address designates the local MAC entity. Frames destined for the MAC Control sublayer are not passed to the MAC client.

#### A.2.4. Effect on Receipt

The effect of receipt of this primitive by the MAC client is unspecified.

## Appendix B. Consistent Overhead Byte Stuffing [COBS]

This Appendix is informative and not part of the standard.

[BACnet] Clause 9 corrects a long-standing issue with the MS/TP specification; namely that preamble sequences were not escaped whenever they appeared in the Data or Data CRC fields. In rare cases, this resulted in dropped frames due to loss of frame synchronization. The solution is to encode the Data and 32-bit Data CRC fields before transmission using Consistent Overhead Byte Stuffing [COBS] and decode these fields upon reception.

COBS is a run-length encoding method that nominally removes '0x00' octets from its input. Any selected octet value may be removed by XOR'ing that value with each octet of the COBS output. [BACnet] Clause 9 specifies the preamble octet '0x55' for removal.

The minimum overhead of COBS is one octet per encoded field. The worst-case overhead in long fields is bounded to one octet per 254 as described in [COBS].

Frame encoding proceeds logically in two passes. The Encoded Data field is prepared by passing the MSDU through the COBS encoder and XOR'ing the preamble octet '0x55' with each octet of the output. The Encoded CRC-32K field is then prepared by calculating a CRC-32K over the Encoded Data field and formatting it for transmission as described in Appendix C. The combined length of these fields, minus two octets for compatibility with legacy MS/TP devices, is placed in the MS/TP header Length field before transmission.

Example COBS encoder and decoder functions are shown below for illustration. Complete examples of use and test vectors are provided in [BACnet] Annex T.

<CODE BEGINS>

```
#include <stddef.h>
#include <stdint.h>

/*
 * Encodes 'length' octets of data located at 'from' and
 * writes one or more COBS code blocks at 'to', removing any
 * 'mask' octets that may present be in the encoded data.
 * Returns the length of the encoded data.
 */

size_t
cobs_encode (uint8_t *to, const uint8_t *from, size_t length,
```

```
                uint8_t mask)
  {
    size_t code_index = 0;
    size_t read_index = 0;
    size_t write_index = 1;
    uint8_t code = 1;
    uint8_t data, last_code;

    while (read_index < length) {
      data = from[read_index++];
      /*
       * In the case of encountering a non-zero octet in the data,
       * simply copy input to output and increment the code octet.
       */
      if (data != 0) {
        to[write_index++] = data ^ mask;
        code++;
        if (code != 255)
          continue;
      }
      /*
       * In the case of encountering a zero in the data or having
       * copied the maximum number (254) of non-zero octets, store
       * the code octet and reset the encoder state variables.
       */
      last_code = code;
      to[code_index] = code ^ mask;
      code_index = write_index++;
      code = 1;
    }
    /*
     * If the last chunk contains exactly 254 non-zero octets, then
     * this exception is handled above (and returned length must be
     * adjusted). Otherwise, encode the last chunk normally, as if
     * a "phantom zero" is appended to the data.
     */
    if ((last_code == 255) && (code == 1))
      write_index--;
    else
      to[code_index] = code ^ mask;

    return write_index;
  }
```

```
#include <stddef.h>
#include <stdint.h>

/*
 * Decodes 'length' octets of data located at 'from' and
 * writes the original client data at 'to', restoring any
 * 'mask' octets that may present in the encoded data.
 * Returns the length of the encoded data or zero if error.
 */
size_t
cobs_decode (uint8_t *to, const uint8_t *from, size_t length,
             uint8_t mask)
{
  size_t read_index = 0;
  size_t write_index = 0;
  uint8_t code, last_code;

  while (read_index < length) {
    code = from[read_index] ^ mask;
    last_code = code;
    /*
     * Sanity check the encoding to prevent the while() loop below
     * from overrunning the output buffer.
     */
    if (read_index + code > length)
      return 0;

    read_index++;
    while (--code > 0)
      to[write_index++] = from[read_index++] ^ mask;
    /*
     * Restore the implicit zero at the end of each decoded block
     * except when it contains exactly 254 non-zero octets or the
     * end of data has been reached.
     */
    if ((last_code != 255) && (read_index < length))
      to[write_index++] = 0;
  }
  return write_index;
}
```

<CODE ENDS>

## Appendix C. Encoded CRC-32K [CRC32K]

This Appendix is informative and not part of the standard.

Extending the payload of MS/TP to 1500 octets required upgrading the Data CRC from 16 bits to 32 bits. P.Koopman has authored several papers on evaluating CRC polynomials for network applications. In [CRC32K], he surveyed the entire 32-bit polynomial space and noted some that exceed the [IEEE.802.3_2012] polynomial in performance. [BACnet] Clause 9 specifies one of these, the CRC-32K (Koopman) polynomial.

The specified use of the calc_crc32K() function is as follows. Before a frame is transmitted, 'crc_value' is initialized to all ones. After passing each octet of the [COBS] Encoded Data through the function, the ones complement of the resulting 'crc_value' is arranged in LSB-first order and is itself [COBS] encoded. The length of the resulting Encoded CRC-32K field is always five octets.

Upon reception of a frame, 'crc_value' is initialized to all ones. The octets of the Encoded Data field are accumulated by the calc_crc32K() function before decoding. The Encoded CRC-32K field is then decoded and the resulting four octets are accumulated by the calc_crc32K() function. If the result is the expected residue value 'CRC32K_RESIDUE', then the frame was received correctly.

An example CRC-32K function in shown below for illustration. Complete examples of use and test vectors are provided in [BACnet] Annex G.3.

<CODE BEGINS>

```
#include <stdint.h>

/* See BACnet Addendum 135-2012an, section G.3.2 */
#define CRC32K_INITIAL_VALUE (0xFFFFFFFF)
#define CRC32K_RESIDUE (0x0843323B)

/* CRC-32K polynomial, 1 + x**1 + ... + x**30 (+ x**32) */
#define CRC32K_POLY (0xEB31D82E)

/*
 * Accumulate 'data_value' into the CRC in 'crc_value'.
 * Return updated CRC.
 *
 * Note: crc_value must be set to CRC32K_INITIAL_VALUE
 * before initial call.
 */
uint32_t
calc_crc32K (uint8_t data_value, uint32_t crc_value)
{
  int b;

  for (b = 0; b < 8; b++) {
    if ((data_value & 1) ^ (crc_value & 1)) {
      crc_value >>= 1;
      crc_value ^= CRC32K_POLY;
    } else {
      crc_value >>= 1;
    }
    data_value >>= 1;
  }
  return crc_value;
}
```

<CODE ENDS>

## Appendix D. Example 6LoBAC Frame Decode

This Appendix is informative and not part of the standard.

```
BACnet MS/TP, Src (2), Dst (1), IPv6 Encapsulation
    Preamble 55: 0x55
    Preamble FF: 0xff
    Frame Type: IPv6 Encapsulation (34)
    Destination Address: 1
    Source Address: 2
    Length: 537
    Header CRC: 0x1c [correct]
    Extended Data CRC: 0x9e7259e2 [correct]
6LoWPAN
    IPHC Header
        011. .... = Pattern: IP header compression (0x03)
        ...1 1... .... .... = Traffic class and flow label:
                              Version, traffic class, and flow label
                              compressed (0x0003)
        .... .0.. .... .... = Next header: Inline
        .... ..00 .... .... = Hop limit: Inline (0x0000)
        .... .... 1... .... = Context identifier extension: True
        .... .... .1.. .... = Source address compression: Stateful
        .... .... ..01 .... = Source address mode:
                              64-bits inline (0x0001)
        .... .... .... 0... = Multicast address compression: False
        .... .... .... .1.. = Destination address compression:
                              Stateful
        .... .... .... ..10 = Destination address mode:
                              16-bits inline (0x0002)
        0000 .... = Source context identifier: 0x00
        .... 0000 = Destination context identifier: 0x00
        [Source context: aaaa:: (aaaa::)]
        [Destination context: aaaa:: (aaaa::)]
    Next header: ICMPv6 (0x3a)
    Hop limit: 63
    Source: aaaa::1 (aaaa::1)
    Destination: aaaa::ff:fe00:1 (aaaa::ff:fe00:1)
Internet Protocol Version 6, Src: aaaa::1 (aaaa::1),
                              Dst: aaaa::ff:fe00:1 (aaaa::ff:fe00:1)
    0110 .... .... .... .... .... .... .... = Version: 6
    .... 0000 0000 .... .... .... .... .... = Traffic class:
                                               0x00000000
    .... 0000 00.. .... .... .... .... .... = Differentiated
                                               Services Field:
                                               Default (0x00000000)
    .... .... ..0. .... .... .... .... .... = ECN-Capable Transport
```

```
                                                  (ECT): Not set
        .... .... ...0 .... .... .... .... .... = ECN-CE: Not set
        .... .... .... 0000 0000 0000 0000 0000 = Flowlabel: 0x00000000
        Payload length: 518
        Next header: ICMPv6 (58)
        Hop limit: 63
        Source: aaaa::1 (aaaa::1)
        Destination: aaaa::ff:fe00:1 (aaaa::ff:fe00:1)
    Internet Control Message Protocol v6
        Type: Echo (ping) request (128)
        Code: 0
        Checksum: 0x783f [correct]
        Identifier: 0x2ee5
        Sequence: 2
        [Response In: 5165]
        Data (510 bytes)
            Data: e4dbe8553ba0040008090a0b0c0d0e0f1011121314151617...
            [Length: 510]
```

```
Frame (547 bytes):
55 ff 22 01 02 02 19 1c 56 2d 83 56 6f 6a 54 54   U."....V-.VojTT
54 54 54 54 57 54 56 54 d5 50 2d 6a 7b b0 5c 57   TTTTWTVT.P-j{.\W
b1 8e bd 00 6e f5 51 ac 5d 5c 5f 5e 59 58 5b 5a   ....n.Q.]\_^YX[Z
45 44 47 46 41 40 43 42 4d 4c 4f 4e 49 48 4b 4a   EDGFA@CBMLONIHKJ
75 74 77 76 71 70 73 72 7d 7c 7f 7e 79 78 7b 7a   utwvqpsr}|.~yx{z
65 64 67 66 61 60 63 62 6d 6c 6f 6e 69 68 6b 6a   edgfa`cbmlonihkj
15 14 17 16 11 10 13 12 1d 1c 1f 1e 19 18 1b 1a   ................
05 04 07 06 01 00 03 02 0d 0c 0f 0e 09 08 0b 0a   ................
35 34 37 36 31 30 33 32 3d 3c 3f 3e 39 38 3b 3a   54761032=<?>98;:
25 24 27 26 21 20 23 22 2d 2c 2f 2e 29 28 2b 2a   %$'&! #"-,/.)(+*
d5 d4 d7 d6 d1 d0 d3 d2 dd dc df de d9 d8 db da   ................
c5 c4 c7 c6 c1 c0 c3 c2 cd cc cf ce c9 c8 cb ca   ................
f5 f4 f7 f6 f1 f0 f3 f2 fd fc ff fe f9 f8 fb fa   ................
e5 e4 e7 e6 e1 e0 e3 e2 ed ec ef ee e9 e8 eb ea   ................
95 94 97 96 91 90 93 92 9d 9c 9f 9e 99 98 9b 9a   ................
85 84 87 86 81 80 83 82 8d 8c 8f 8e 89 88 8b 8a   ................
b5 b4 b7 b6 b1 b0 b3 b2 bd bc bf be b9 b8 bb ba   ................
a5 a4 a7 a6 a1 a0 a3 a2 ad ac af ae a9 a8 ab aa   ................
ab 54 57 56 51 50 53 52 5d 5c 5f 5e 59 58 5b 5a   .TWVQPSR]\_^YX[Z
45 44 47 46 41 40 43 42 4d 4c 4f 4e 49 48 4b 4a   EDGFA@CBMLONIHKJ
75 74 77 76 71 70 73 72 7d 7c 7f 7e 79 78 7b 7a   utwvqpsr}|.~yx{z
65 64 67 66 61 60 63 62 6d 6c 6f 6e 69 68 6b 6a   edgfa`cbmlonihkj
15 14 17 16 11 10 13 12 1d 1c 1f 1e 19 18 1b 1a   ................
05 04 07 06 01 00 03 02 0d 0c 0f 0e 09 08 0b 0a   ................
35 34 37 36 31 30 33 32 3d 3c 3f 3e 39 38 3b 3a   54761032=<?>98;:
25 24 27 26 21 20 23 22 2d 2c 2f 2e 29 28 2b 2a   %$'&! #"-,/.)(+*
d5 d4 d7 d6 d1 d0 d3 d2 dd dc df de d9 d8 db da   ................
c5 c4 c7 c6 c1 c0 c3 c2 cd cc cf ce c9 c8 cb ca   ................
f5 f4 f7 f6 f1 f0 f3 f2 fd fc ff fe f9 f8 fb fa   ................
e5 e4 e7 e6 e1 e0 e3 e2 ed ec ef ee e9 e8 eb ea   ................
95 94 97 96 91 90 93 92 9d 9c 9f 9e 99 98 9b 9a   ................
85 84 87 86 81 80 83 82 8d 8c 8f 8e 89 88 8b 8a   ................
b5 b4 b7 b6 b1 b0 b3 b2 bd bc bf be b9 b8 bb ba   ................
a5 a4 a7 a6 a1 a0 a3 a2 ad ac af ae a9 a8 50 cb   ..............P.
27 0c b7                                          '..
```

```
Decoded Data and CRC32K (537 bytes):
78 d6 00 3a 3f 00 00 00 00 00 00 01 00 01 80   x..:?...........
00 78 3f 2e e5 00 02 e4 db e8 55 3b a0 04 00 08   .x?.......U;....
09 0a 0b 0c 0d 0e 0f 10 11 12 13 14 15 16 17 18   ................
19 1a 1b 1c 1d 1e 1f 20 21 22 23 24 25 26 27 28   ....... !"#$%&'(
29 2a 2b 2c 2d 2e 2f 30 31 32 33 34 35 36 37 38   )*+,-./012345678
39 3a 3b 3c 3d 3e 3f 40 41 42 43 44 45 46 47 48   9:;<=>?@ABCDEFGH
49 4a 4b 4c 4d 4e 4f 50 51 52 53 54 55 56 57 58   IJKLMNOPQRSTUVWX
59 5a 5b 5c 5d 5e 5f 60 61 62 63 64 65 66 67 68   YZ[\]^_‘abcdefgh
69 6a 6b 6c 6d 6e 6f 70 71 72 73 74 75 76 77 78   ijklmnopqrstuvwx
79 7a 7b 7c 7d 7e 7f 80 81 82 83 84 85 86 87 88   yz{|}~..........
89 8a 8b 8c 8d 8e 8f 90 91 92 93 94 95 96 97 98   ................
99 9a 9b 9c 9d 9e 9f a0 a1 a2 a3 a4 a5 a6 a7 a8   ................
a9 aa ab ac ad ae af b0 b1 b2 b3 b4 b5 b6 b7 b8   ................
b9 ba bb bc bd be bf c0 c1 c2 c3 c4 c5 c6 c7 c8   ................
c9 ca cb cc cd ce cf d0 d1 d2 d3 d4 d5 d6 d7 d8   ................
d9 da db dc dd de df e0 e1 e2 e3 e4 e5 e6 e7 e8   ................
e9 ea eb ec ed ee ef f0 f1 f2 f3 f4 f5 f6 f7 f8   ................
f9 fa fb fc fd fe ff 00 01 02 03 04 05 06 07 08   ................
09 0a 0b 0c 0d 0e 0f 10 11 12 13 14 15 16 17 18   ................
19 1a 1b 1c 1d 1e 1f 20 21 22 23 24 25 26 27 28   ....... !"#$%&'(
29 2a 2b 2c 2d 2e 2f 30 31 32 33 34 35 36 37 38   )*+,-./012345678
39 3a 3b 3c 3d 3e 3f 40 41 42 43 44 45 46 47 48   9:;<=>?@ABCDEFGH
49 4a 4b 4c 4d 4e 4f 50 51 52 53 54 55 56 57 58   IJKLMNOPQRSTUVWX
59 5a 5b 5c 5d 5e 5f 60 61 62 63 64 65 66 67 68   YZ[\]^_‘abcdefgh
69 6a 6b 6c 6d 6e 6f 70 71 72 73 74 75 76 77 78   ijklmnopqrstuvwx
79 7a 7b 7c 7d 7e 7f 80 81 82 83 84 85 86 87 88   yz{|}~..........
89 8a 8b 8c 8d 8e 8f 90 91 92 93 94 95 96 97 98   ................
99 9a 9b 9c 9d 9e 9f a0 a1 a2 a3 a4 a5 a6 a7 a8   ................
a9 aa ab ac ad ae af b0 b1 b2 b3 b4 b5 b6 b7 b8   ................
b9 ba bb bc bd be bf c0 c1 c2 c3 c4 c5 c6 c7 c8   ................
c9 ca cb cc cd ce cf d0 d1 d2 d3 d4 d5 d6 d7 d8   ................
d9 da db dc dd de df e0 e1 e2 e3 e4 e5 e6 e7 e8   ................
e9 ea eb ec ed ee ef f0 f1 f2 f3 f4 f5 f6 f7 f8   ................
f9 fa fb fc fd 9e 72 59 e2                        ......rY.
```

```
Decompressed 6LoWPAN IPHC (558 bytes):
60 00 00 00 02 06 3a 3f aa aa 00 00 00 00 00 00   ‘.....:?........
00 00 00 00 00 00 00 01 aa aa 00 00 00 00 00 00   ................
00 00 00 ff fe 00 00 01 80 00 78 3f 2e e5 00 02   ..........x?....
e4 db e8 55 3b a0 04 00 08 09 0a 0b 0c 0d 0e 0f   ...U;...........
10 11 12 13 14 15 16 17 18 19 1a 1b 1c 1d 1e 1f   ................
20 21 22 23 24 25 26 27 28 29 2a 2b 2c 2d 2e 2f    !"#$%&’()*+,-./
30 31 32 33 34 35 36 37 38 39 3a 3b 3c 3d 3e 3f   0123456789:;<=>?
40 41 42 43 44 45 46 47 48 49 4a 4b 4c 4d 4e 4f   @ABCDEFGHIJKLMNO
50 51 52 53 54 55 56 57 58 59 5a 5b 5c 5d 5e 5f   PQRSTUVWXYZ[\]^_
60 61 62 63 64 65 66 67 68 69 6a 6b 6c 6d 6e 6f   ‘abcdefghijklmno
70 71 72 73 74 75 76 77 78 79 7a 7b 7c 7d 7e 7f   pqrstuvwxyz{|}~.
80 81 82 83 84 85 86 87 88 89 8a 8b 8c 8d 8e 8f   ................
90 91 92 93 94 95 96 97 98 99 9a 9b 9c 9d 9e 9f   ................
a0 a1 a2 a3 a4 a5 a6 a7 a8 a9 aa ab ac ad ae af   ................
b0 b1 b2 b3 b4 b5 b6 b7 b8 b9 ba bb bc bd be bf   ................
c0 c1 c2 c3 c4 c5 c6 c7 c8 c9 ca cb cc cd ce cf   ................
d0 d1 d2 d3 d4 d5 d6 d7 d8 d9 da db dc dd de df   ................
e0 e1 e2 e3 e4 e5 e6 e7 e8 e9 ea eb ec ed ee ef   ................
f0 f1 f2 f3 f4 f5 f6 f7 f8 f9 fa fb fc fd fe ff   ................
00 01 02 03 04 05 06 07 08 09 0a 0b 0c 0d 0e 0f   ................
10 11 12 13 14 15 16 17 18 19 1a 1b 1c 1d 1e 1f   ................
20 21 22 23 24 25 26 27 28 29 2a 2b 2c 2d 2e 2f    !"#$%&’()*+,-./
30 31 32 33 34 35 36 37 38 39 3a 3b 3c 3d 3e 3f   0123456789:;<=>?
40 41 42 43 44 45 46 47 48 49 4a 4b 4c 4d 4e 4f   @ABCDEFGHIJKLMNO
50 51 52 53 54 55 56 57 58 59 5a 5b 5c 5d 5e 5f   PQRSTUVWXYZ[\]^_
60 61 62 63 64 65 66 67 68 69 6a 6b 6c 6d 6e 6f   ‘abcdefghijklmno
70 71 72 73 74 75 76 77 78 79 7a 7b 7c 7d 7e 7f   pqrstuvwxyz{|}~.
80 81 82 83 84 85 86 87 88 89 8a 8b 8c 8d 8e 8f   ................
90 91 92 93 94 95 96 97 98 99 9a 9b 9c 9d 9e 9f   ................
a0 a1 a2 a3 a4 a5 a6 a7 a8 a9 aa ab ac ad ae af   ................
b0 b1 b2 b3 b4 b5 b6 b7 b8 b9 ba bb bc bd be bf   ................
c0 c1 c2 c3 c4 c5 c6 c7 c8 c9 ca cb cc cd ce cf   ................
d0 d1 d2 d3 d4 d5 d6 d7 d8 d9 da db dc dd de df   ................
e0 e1 e2 e3 e4 e5 e6 e7 e8 e9 ea eb ec ed ee ef   ................
f0 f1 f2 f3 f4 f5 f6 f7 f8 f9 fa fb fc fd         ..............
```

Authors' Addresses

Kerry Lynn (editor)
Verizon Labs
50 Sylvan Rd
Waltham , MA 02451
USA

Phone: +1 781 296 9722
Email: kerlyn@ieee.org

Jerry Martocci
Johnson Controls, Inc.
507 E. Michigan St
Milwaukee , WI 53202
USA

Email: jpmartocci@sbcglobal.net

Carl Neilson
Delta Controls, Inc.
17850 56th Ave
Surrey , BC V3S 1C7
Canada

Phone: +1 604 575 5913
Email: cneilson@deltacontrols.com

Stuart Donaldson
Honeywell Automation & Control Solutions
6670 185th Ave NE
Redmond , WA 98052
USA

Email: stuart.donaldson@honeywell.com

6lo
Internet-Draft
Updates: 6775, 8505 (if approved)
Intended status: Standards Track
Expires: 24 September 2020

P. Thubert, Ed.
Cisco Systems
C.E. Perkins
Blue Meadow Networking
E. Levy-Abegnoli
Cisco Systems
23 March 2020

# IPv6 Backbone Router
## draft-ietf-6lo-backbone-router-20

## Abstract

This document updates RFC 6775 and RFC 8505 in order to enable proxy services for IPv6 Neighbor Discovery by Routing Registrars called Backbone Routers. Backbone Routers are placed along the wireless edge of a Backbone, and federate multiple wireless links to form a single Multi-Link Subnet.

## 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 September 2020.

Table of Contents



## 1. Introduction

IEEE STD. 802.1 [IEEEstd8021] Ethernet Bridging provides an efficient and reliable broadcast service for wired networks; applications and protocols have been built that heavily depend on that feature for their core operation. Unfortunately, Low-Power Lossy Networks (LLNs) and local wireless networks generally do not provide the broadcast capabilities of Ethernet Bridging in an economical fashion.

As a result, protocols designed for bridged networks that rely on multicast and broadcast often exhibit disappointing behaviours when employed unmodified on a local wireless medium (see [I-D.ietf-mboned-ieee802-mcast-problems]).

Wi-Fi [IEEEstd80211] Access Points (APs) deployed in an Extended Service Set (ESS) act as Ethernet Bridges [IEEEstd8021], with the property that the bridging state is established at the time of association. This ensures connectivity to the end node (the Wi-Fi STA) and protects the wireless medium against broadcast-intensive Transparent Bridging reactive Lookups. In other words, the association process is used to register the MAC Address of the STA to the AP. The AP subsequently proxies the bridging operation and does not need to forward the broadcast Lookups over the radio.

In the same way as Transparent Bridging, IPv6 [RFC8200] Neighbor Discovery [RFC4861] [RFC4862] Protocol (IPv6 ND) is a reactive protocol, based on multicast transmissions to locate an on-link correspondent and ensure the uniqueness of an IPv6 address. The mechanism for Duplicate Address Detection (DAD) [RFC4862] was designed for the efficient broadcast operation of Ethernet Bridging. Since broadcast can be unreliable over wireless media, DAD often fails to discover duplications [I-D.yourtchenko-6man-dad-issues]. In practice, the fact that IPv6 addresses very rarely conflict is mostly attributable to the entropy of the 64-bit Interface IDs as opposed to the succesful operation of the IPv6 ND duplicate address detection and resolution mechanisms.

The IPv6 ND Neighbor Solicitation (NS) [RFC4861] message is used for DAD and address Lookup when a node moves, or wakes up and reconnects to the wireless network. The NS message is targeted to a Solicited-Node Multicast Address (SNMA) [RFC4291] and should in theory only reach a very small group of nodes. But in reality, IPv6 multicast messages are typically broadcast on the wireless medium, and so they are processed by most of the wireless nodes over the subnet (e.g., the ESS fabric) regardless of how few of the nodes are subscribed to the SNMA. As a result, IPv6 ND address Lookups and DADs over a large wireless and/or a LowPower Lossy Network (LLN) can consume enough bandwidth to cause a substantial degradation to the unicast traffic service.

Because IPv6 ND messages sent to the SNMA group are broadcast at the radio MAC Layer, wireless nodes that do not belong to the SNMA group still have to keep their radio turned on to listen to multicast NS messages, which is a waste of energy for them. In order to reduce their power consumption, certain battery-operated devices such as IoT sensors and smartphones ignore some of the broadcasts, making IPv6 ND operations even less reliable.

These problems can be alleviated by reducing the IPv6 ND broadcasts over wireless access links. This has been done by splitting the broadcast domains and routing between subnets, at the extreme by assigning a /64 prefix to each wireless node (see [RFC8273]). But deploying a single large subnet can still be attractive to avoid renumbering in situations that involve large numbers of devices and mobility within a bounded area.

A way to reduce the propagation of IPv6 ND broadcast in the wireless domain while preserving a large single subnet is to form a Multi-Link Subnet (MLSN). Each Link in the MLSN, including the backbone, is its own broadcast domain. A key property of MLSNs is that Link-Local unicast traffic, link-scope multicast, and traffic with a hop limit of 1 will not transit to nodes in the same subnet on a different link, something that may produce unexpected behavior in software that expects a subnet to be entirely contained within a single link.

This specification considers a special type of MLSN with a central backbone that federates edge (LLN) links, each Link providing its own protection against rogue access and tempering or replaying packets. In particular, the use of classical IPv6 ND on the backbone requires that the all nodes are trusted and that rogue access to the backbone is prevented at all times (see Section 11).

In that particular topology, ND proxies can be placed at the boundary of the edge links and the backbone to handle IPv6 ND on behalf of Registered Nodes and forward IPv6 packets back and forth. The ND proxy enables the continuity of IPv6 ND operations beyond the backbone, and enables communication using Global or Unique Local Addresses between any pair of nodes in the MLSN.

The 6LoWPAN Backbone Router (6BBR) is a Routing Registrar [RFC8505] that provides proxy-ND services. A 6BBR acting as a Bridging Proxy provides a proxy-ND function with Layer-2 continuity and can be collocated with a Wi-Fi Access Point (AP) as prescribed by IEEE Std 802.11 [IEEEstd80211]. A 6BBR acting as a Routing Proxy is applicable to any type of LLN, including LLNs that cannot be bridged onto the backbone, such as IEEE Std 802.15.4 [IEEEstd802154].

Knowledge of which address to proxy for can be obtained by snooping the IPV6 ND protocol (see [I-D.bi-savi-wlan]), but it has been found to be unreliable. An IPv6 address may not be discovered immediately due to a packet loss, or if a "silent" node is not currently using one of its addresses. A change of state (e.g., due to movement) may be missed or misordered, leading to unreliable connectivity and incomplete knowledge of the state of the network.

With this specification, the address to be proxied is signaled explicitly through a registration process. A 6LoWPAN node (6LN) registers all its IPv6 Addresses using NS messages with an Extended Address Registration Option (EARO) as specified in [RFC8505] to a 6LoWPAN Router (6LR) to which it is directly attached. If the 6LR is a 6BBR then the 6LN is both the Registered Node and the Registering Node. If not, then the 6LoWPAN Border Router (6LBR) that serves the LLN proxies the registration to the 6BBR. In that case, the 6LN is the Registered Node and the 6LBR is the Registering Node. The 6BBR performs IPv6 Neighbor Discovery (IPv6 ND) operations on its Backbone interface on behalf of the 6LNs that have registered addresses on its LLN interfaces without the need of a broadcast over the wireless medium.

A Registering Node that resides on the backbone does not register to the SNMA groups associated to its Registered Addresses and defers to the 6BBR to answer or preferably forward to it as unicast the corresponding multicast packets.

## 2. Terminology

### 2.1. BCP 14

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. New Terms

This document introduces the following terminology:

Federated: A subnet that comprises a Backbone and one or more (wireless) access links, is said to be federated into one Multi-Link Subnet. The proxy-ND operation of 6BBRs over the Backbone extends IPv6 ND operation over the access links.

Sleeping Proxy: A 6BBR acts as a Sleeping Proxy if it answers IPv6 ND Neighbor Solicitations over the Backbone on behalf of the Registering Node that is in a sleep state and cannot answer in due time.

Routing Proxy: A Routing Proxy provides IPv6 ND proxy functions and enables the MLSN operation over federated links that may not be compatible for bridging. The Routing Proxy advertises its own MAC Address as the Target Link Layer Address (TLLA) in the proxied NAs

over the Backbone, and routes at the Network Layer between the federated links.

Bridging Proxy: A Bridging Proxy provides IPv6 ND proxy functions while preserving forwarding continuity at the MAC Layer. In that case, the MAC Address and the mobility of the Registering Node is visible across the bridged Backbone. The Bridging Proxy advertises the MAC Address of the Registering Node as the TLLA in the proxied NAs over the Backbone, and proxies ND for all unicast addresses including Link-Local Addresses. Instead of replying on behalf of the Registering Node, a Bridging Proxy will preferably forward the NS Lookup and NUD messages that target the Registered Address to the Registering Node as unicast frames and let it respond in its own.

Binding Table: The Binding Table is an abstract database that is maintained by the 6BBR to store the state associated with its registrations.

Binding: A Binding is an abstract state associated to one registration, in other words one entry in the Binding Table.

## 2.3. Abbreviations

This document uses the following abbreviations:

6BBR: 6LoWPAN Backbone Router
6LBR: 6LoWPAN Border Router
6LN: 6LoWPAN Node
6LR: 6LoWPAN Router
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
ID: Identifier
LLN: Low-Power and Lossy Network
NA: Neighbor Advertisement
MAC: Medium Access Control
NCE: Neighbor Cache Entry
ND: Neighbor Discovery
NDP: Neighbor Discovery Protocol
NS: Neighbor Solicitation

NS(DAD): NDP NS message used for the purpose of duplication avoidance (multicast)
NS(Lookup): NDP NS message used for the purpose of address resolution (multicast)
NS(NUD): NDP NS message used for the purpose of unreachability detection (unicast)
NUD: Neighbor Unreachability Detection
ROVR: Registration Ownership Verifier
RPL: IPv6 Routing Protocol for LLNs
RA: Router Advertisement
RS: Router Solicitation
SNMA: Solicited-Node Multicast Address
LLA: Link Layer Address (aka MAC address)
SLLA: Source Link Layer Address
TLLA: Target Link Layer Address
TID: Transaction ID

## 2.4. References

In this document, readers will encounter terms and concepts that are discussed in the following documents:

Classical IPv6 ND: "Neighbor Discovery for IP version 6" [RFC4861], "IPv6 Stateless Address Autoconfiguration" [RFC4862] and "Optimistic Duplicate Address Detection" [RFC4429],

IPv6 ND over multiple links: "Neighbor Discovery Proxies (proxy-ND)" [RFC4389] and "Multi-Link Subnet Issues" [RFC4903],

6LoWPAN: "Problem Statement and Requirements for IPv6 over Low-Power Wireless Personal Area Network (6LoWPAN) Routing" [RFC6606], and

6LoWPAN ND: Neighbor Discovery Optimization for Low-Power and Lossy Networks [RFC6775], "Registration Extensions for 6LoWPAN Neighbor Discovery" [RFC8505], and " Address Protected Neighbor Discovery for Low-power and Lossy Networks" [I-D.ietf-6lo-ap-nd].

# 3. Overview

This section and its subsections present a non-normative high level view of the operation of the 6BBR. The following sections cover the normative part.

Figure 1 illustrates a backbone link that federates a collection of LLNs as a single IPv6 Subnet, with a number of 6BBRs providing proxy-ND services to their attached LLNs.

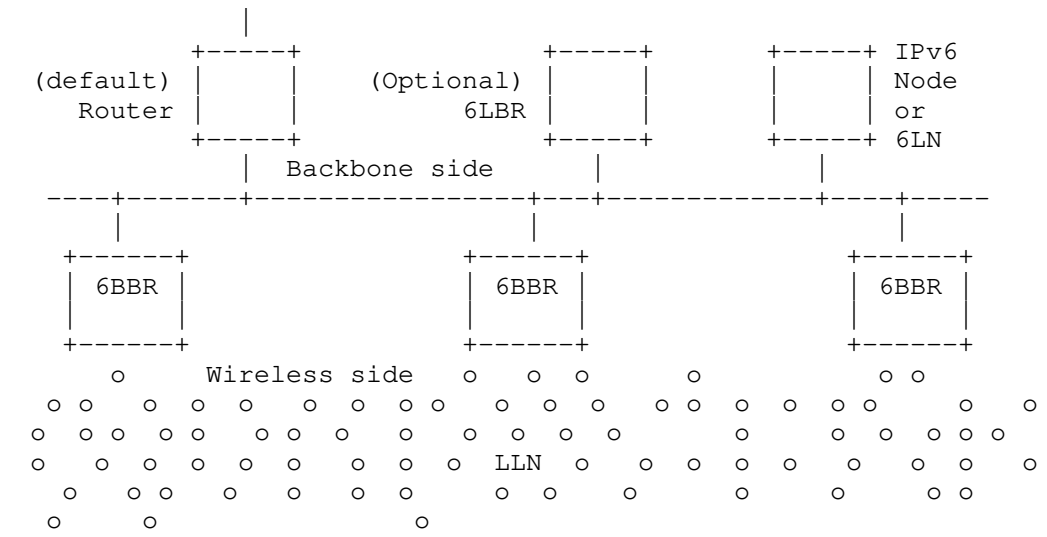



Figure 1: Backbone Link and Backbone Routers

The LLN may be a hub-and-spoke access link such as (Low-Power) IEEE STD. 802.11 (Wi-Fi) [IEEEstd80211] and IEEE STD. 802.15.1 (Bluetooth) [IEEEstd802151], or a Mesh-Under or a Route-Over network [RFC8505]. The proxy state can be distributed across multiple 6BBRs attached to the same Backbone.

The main features of a 6BBR are as follows:

* Multi-Link-subnet functions (provided by the 6BBR on the backbone) performed on behalf of Registered Nodes, and

* Routing registrar services that reduce multicast within the LLN:

  - Binding Table management
  - failover, e.g., due to mobility

Each Backbone Router (6BBR) maintains a data structure for its Registered Addresses called a Binding Table. The abstract data that is stored in the Binding Table includes the Registered Address, anchor information on the Registering Node such as connecting interface, Link-Local Address and Link-Layer Address of the Registering Node on that interface, the EARO including ROVR and TID, a state that can be either Reachable, Tentative, or Stale, and other information such as a trust level that may be configured, e.g., to protect a server. The combined Binding Tables of all the 6BBRs on a backbone form a distributed database of Registered Nodes that reside in the LLNs or on the IPv6 Backbone.

Unless otherwise configured, a 6BBR does the following:

* Create a new entry in a Binding Table for a new Registered Address and ensure that the Address is not duplicated over the Backbone.

* Advertise a Registered Address over the Backbone using an NA message, either unsolicited or as a response to a NS message. This includes joining the multicast group associated to the SNMA derived from the Registered Address as specified in section 7.2.1. of [RFC4861] over the Backbone.

* The 6BBR MAY respond immediately as a Proxy in lieu of the Registering Node, e.g., if the Registering Node has a sleeping cycle that the 6BBR does not want to interrupt, or if the 6BBR has a recent state that is deemed fresh enough to permit the proxied response. It is preferred, though, that the 6BBR checks whether the Registering Node is still responsive on the Registered Address. To that effect:

  - as a Bridging Proxy:
    the 6BBR forwards the multicast DAD and Address Lookup messages as a unicast MAC-Layer frames to the MAC address of the Registering Node that matches the Target in the ND message, and forwards as is the unicast Neighbor Unreachability Detection (NUD) messages, so as to let the Registering Node answer with the ND Message and options that it sees fit;
  - as a Routing Proxy:
    the 6BBR checks the liveliness of the Registering Node, e.g., using a NUD verification, before answering on its behalf.

* Deliver packets arriving from the LLN, using Neighbor Solicitation messages to look up the destination over the Backbone.

* Forward or bridge packets between the LLN and the Backbone.

* Verify liveness for a registration, when needed.

The first of these functions enables the 6BBR to fulfill its role as a Routing Registrar for each of its attached LLNs. The remaining functions fulfill the role of the 6BBRs as the border routers that federate the Multi-link IPv6 subnet.

The operation of IPv6 ND and of proxy-ND are not mutually exclusive on the Backbone, meaning that nodes attached to the Backbone and using IPv6 ND can transparently interact with 6LNs that rely on a 6BBR to proxy ND for them, whether the 6LNs are reachable over an LLN or directly attached to the Backbone.

The [RFC8505] registration mechanism used to learn addresses to be proxied may co-exist in a 6BBR with a proprietary snooping or the traditional bridging functionality of an Access Point, in order to support legacy LLN nodes that do not support this specification.

The registration to a proxy service uses an NS/NA exchange with EARO. The 6BBR operation resembles that of a Mobile IPv6 (MIPv6) [RFC6275] Home Agent (HA). The combination of a 6BBR and a MIPv6 HA enables full mobility support for 6LNs, inside and outside the links that form the subnet.

The 6BBRs performs IPv6 ND functions over the backbone as follows:

* The EARO [RFC8505] is used in the IPv6 ND exchanges over the Backbone between the 6BBRs to help distinguish duplication from movement. Extended Duplicate Address Messages (EDAR and EDAC) may also be used to communicate with a 6LBR, if one is present. Address duplication is detected using the ROVR field. Conflicting registrations to different 6BBRs for the same Registered Address are resolved using the TID field which forms an order of registrations.

* The Link Layer Address (LLA) that the 6BBR advertises for the Registered Address on behalf of the Registered Node over the Backbone can belong to the Registering Node; in that case, the 6BBR (acting as a Bridging Proxy (see Section 8)) bridges the unicast packets. Alternatively, the LLA can be that of the 6BBR on the Backbone interface, in which case the 6BBR (acting as a Routing Proxy (see Section 7)) receives the unicast packets at Layer 3 and routes over.

### 3.1. Updating RFC 6775 and RFC 8505

This specification adds the EARO as a possible option in RS, NS(DAD) and NA messages over the backbone. This document specifies the use of those ND messages by 6BBRs over the backbone, at a high level in Section 6 and in more detail in Section 9.

Note: [RFC8505] requires that the registration NS(EARO) contains an Source Link Layer Address Option (SLLAO). [RFC4862] requires that the NS(DAD) is sent from the unspecified address for which there cannot be a SLLAO. Consequently, an NS(DAD) cannot be confused with a registration.

This specification allows to deploy a 6LBR on the backbone where EDAR and EDAC messages coexist with classical ND. It also adds the capability to insert IPv6 ND options in the EDAR and EDAC messages. A 6BBR acting as a 6LR for the Registered Address can insert an SLLAO

in the EDAR to the 6LBR in order to avoid a Lookup back. This enables the 6LBR to store the MAC address associated to the Registered Address on a Link and to serve as a mapping server as described in [I-D.thubert-6lo-unicast-lookup].

This specification allows for an address to be registered to more than one 6BBR. Consequently a 6LBR that is deployed on the backbone MUST be capable of maintaining state for each of the 6BBR having registered with the same TID and same ROVR.

## 3.2. Access Link

The simplest Multi-Link Subnet topology from the Layer 3 perspective occurs when the wireless network appears as a single hop hub-and-spoke network as shown in Figure 2. The Layer 2 operation may effectively be hub-and-spoke (e.g., Wi-Fi) or Mesh-Under, with a Layer 2 protocol handling the complex topology.

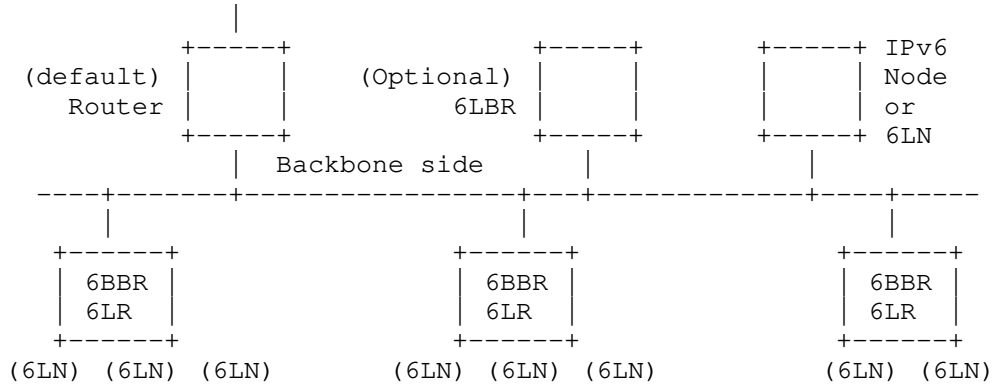



Figure 2: Access Link Use case

Figure 3 illustrates a flow where 6LN forms an IPv6 Address and registers it to a 6BBR acting as a 6LR [RFC8505]. The 6BBR applies ODAD (see Section 3.6) to the registered address to enable connectivity while the message flow is still in progress.

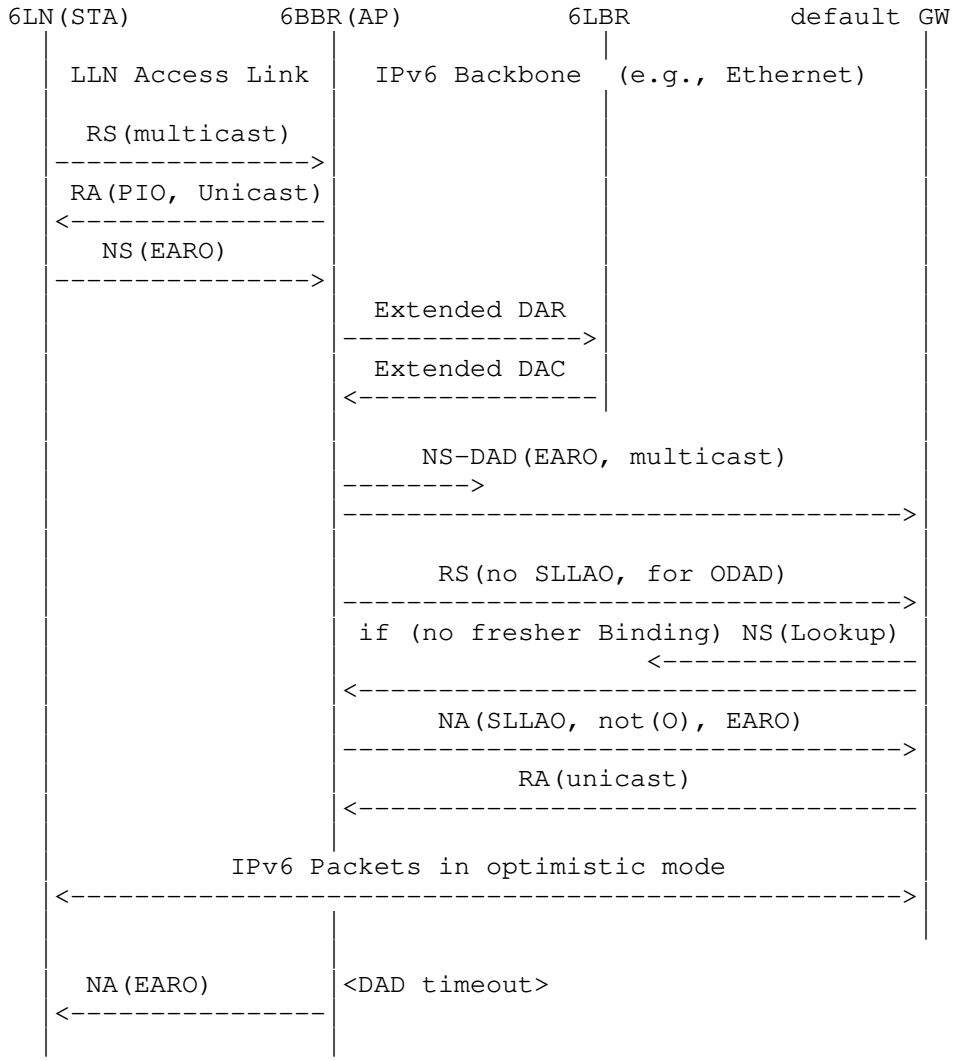



Figure 3: Initial Registration Flow to a 6BBR acting as Routing Proxy

In this example, a 6LBR is deployed on the backbone link to serve the whole subnet, and EDAR / EDAC messages are used in combination with DAD to enable coexistence with IPv6 ND over the backbone.

The RS sent initially by the 6LN (e.g., a Wi-Fi STA) is transmitted as a multicast but since it is intercepted by the 6BBR, it is never effectively broadcast. The multiple arrows associated to the ND messages on the Backbone denote a real Layer 2 broadcast.

### 3.3. Route-Over Mesh

A more complex Multi-Link Subnet topology occurs when the wireless network appears as a Layer 3 Mesh network as shown in Figure 4. A so-called Route-Over routing protocol exposes routes between 6LRs towards both 6LRs and 6LNs, and a 6LBR acts as Root of the Layer 3 Mesh network and proxy-registers the LLN addresses to the 6BBR.

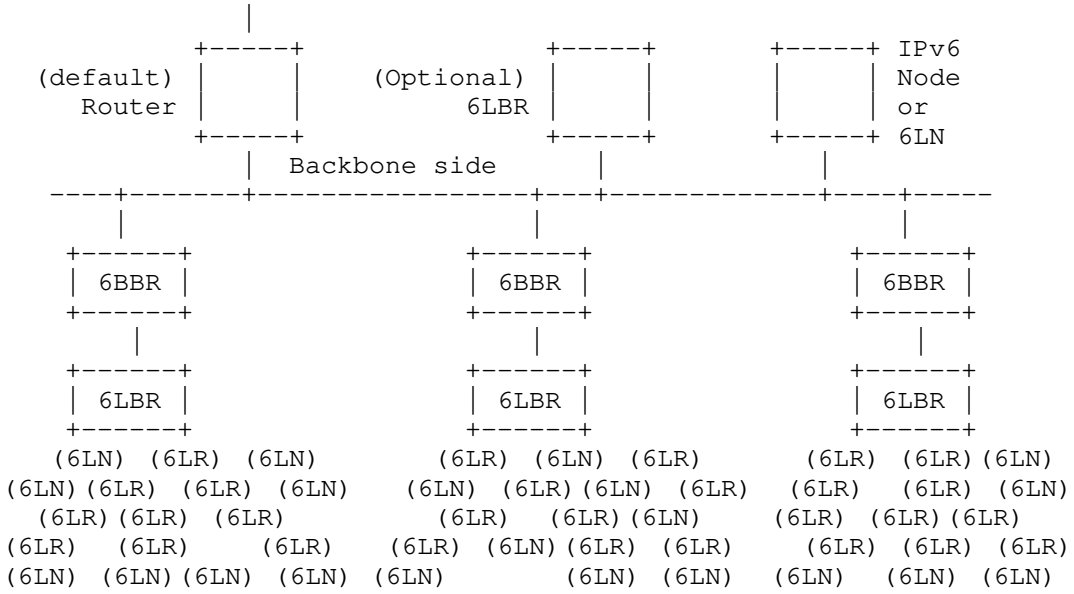



Figure 4: Route-Over Mesh Use case

Figure 5 illustrates IPv6 signaling that enables a 6LN (the Registered Node) to form a Global or a Unique-Local Address and register it to the 6LBR that serves its LLN using [RFC8505] using a neighboring 6LR as relay. The 6LBR (the Registering Node) then proxies the [RFC8505] registration to the 6BBR to obtain proxy-ND services from the 6BBR.

The RS sent initially by the 6LN is a transmitted as a multicast and contained within 1-hop broadcast range where hopefully a 6LR is found. The 6LR is expected to be already connected to the LLN and capable to reach the 6LBR, possibly multiple hops away, using unicast messages.

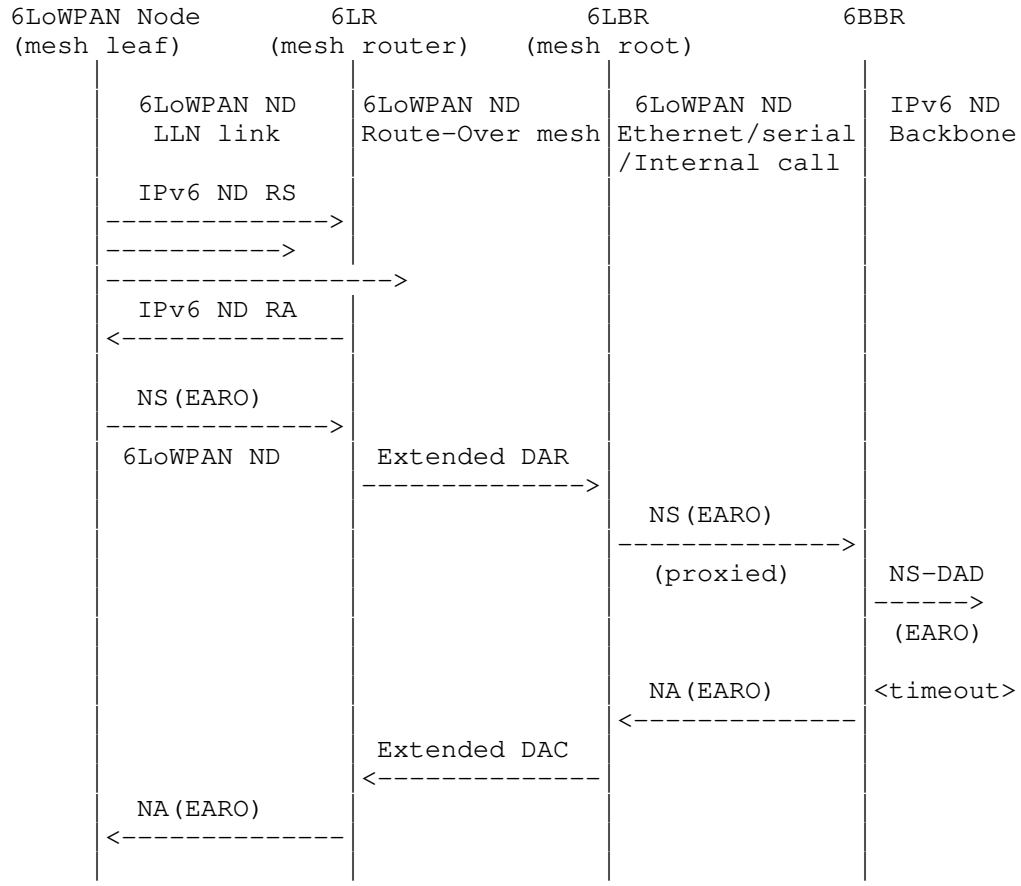



Figure 5: Initial Registration Flow over Route-Over Mesh

As a non-normative example of a Route-Over Mesh, the 6TiSCH architecture [I-D.ietf-6tisch-architecture] suggests using the RPL [RFC6550] routing protocol and collocating the RPL root with a 6LBR that serves the LLN. The 6LBR is also either collocated with or directly connected to the 6BBR over an IPv6 Link.

## 3.4. The Binding Table

Addresses in an LLN that are reachable from the Backbone by way of the 6BBR function must be registered to that 6BBR, using an NS(EARO) with the R flag set [RFC8505]. The 6BBR answers with an NA(EARO) and maintains a state for the registration in an abstract Binding Table.

An entry in the Binding Table is called a "Binding". A Binding may be in Tentative, Reachable or Stale state.

The 6BBR uses a combination of [RFC8505] and IPv6 ND over the Backbone to advertise the registration and avoid a duplication. Conflicting registrations are solved by the 6BBRs, transparently to the Registering Nodes.

Only one 6LN may register a given Address, but the Address may be registered to Multiple 6BBRs for higher availability.

Over the LLN, Binding Table management is as follows:

* De-registrations (newer TID, same ROVR, null Lifetime) are accepted with a status of 4 ("Removed"); the entry is deleted;

* Newer registrations (newer TID, same ROVR, non-null Lifetime) are accepted with a status of 0 (Success); the Binding is updated with the new TID, the Registration Lifetime and the Registering Node; in Tentative state the EDAC response is held and may be overwritten; in other states the Registration Lifetime timer is restarted and the entry is placed in Reachable state.

* Identical registrations (same TID, same ROVR) from the same Registering Node are accepted with a status of 0 (Success). In Tentative state, the response is held and may be overwritten, but the response is eventually produced, carrying the result of the DAD process;

* Older registrations (older TID, same ROVR) from the same Registering Node are discarded;

* Identical and older registrations (not-newer TID, same ROVR) from a different Registering Node are rejected with a status of 3 (Moved); this may be rate limited to avoid undue interference;

* Any registration for the same address but with a different ROVR is rejected with a status of 1 (Duplicate).

The operation of the Binding Table is specified in detail in Section 9.

### 3.5. Primary and Secondary 6BBRs

A Registering Node MAY register the same address to more than one 6BBR, in which case the Registering Node uses the same EARO in all the parallel registrations. On the other hand, there is no provision in 6LoWPAN ND for a 6LN (acting as Registered Node) to select its 6LBR (acting as Registering Node), so it cannot select more than one either. To allow for this, NS(DAD) and NA messages with an EARO received over the backbone that indicate an identical Binding in

another 6BBR (same Registered address, same TID, same ROVR) are silently ignored but for the purpose of selecting the primary 6BBR for that registration.

A 6BBR may be either primary or secondary. The primary is the 6BBR that has the highest EUI-64 Address of all the 6BBRs that share a registration for the same Registered Address, with the same ROVR and same Transaction ID, the EUI-64 Address being considered as an unsigned 64bit integer. A given 6BBR can be primary for a given Address and secondary for another Address, regardless of whether or not the Addresses belong to the same 6LN.

In the following sections, it is expected that an NA is sent over the backbone only if the node is primary or does not support the concept of primary. More than one 6BBR claiming or defending an address generates unwanted traffic but no reachability issue since all 6BBRs provide reachability from the Backbone to the 6LN.

If a Registering Node loses connectivity to its or one of the 6BBRs to which it registered an address, it retries the registration to the (one or more) available 6BBR(s). When doing that, the Registering Node MUST increment the TID in order to force the migration of the state to the new 6BBR, and the reselection of the primary 6BBR if it is the node that was lost.

### 3.6. Using Optimistic DAD

Optimistic Duplicate Address Detection [RFC4429] (ODAD) specifies how an IPv6 Address can be used before completion of Duplicate Address Detection (DAD). ODAD guarantees that this behavior will not cause harm if the new Address is a duplicate.

Support for ODAD avoids delays in installing the Neighbor Cache Entry (NCE) in the 6BBRs and the default router, enabling immediate connectivity to the registered node. As shown in Figure 3, if the 6BBR is aware of the Link-Layer Address (LLA) of a router, then the 6BBR sends a Router Solicitation (RS), using the Registered Address as the IP Source Address, to the known router(s). The RS is sent without a Source LLA Option (SLLAO), to avoid invalidating a preexisting NCE in the router.

Following ODAD, the router may then send a unicast RA to the Registered Address, and it may resolve that Address using an NS(Lookup) message. In response, the 6BBR sends an NA with an EARO and the Override flag [RFC4861] that is not set. The router can then determine the freshest EARO in case of conflicting NA(EARO) messages, using the method described in section 5.2.1 of [RFC8505]. If the NA(EARO) is the freshest answer, the default router creates a Binding

with the SLLAO of the 6BBR (in Routing Proxy mode) or that of the Registering Node (in Bridging Proxy mode) so that traffic from/to the Registered Address can flow immediately.

## 4. Multi-Link Subnet Considerations

The Backbone and the federated LLN Links are considered as different links in the Multi-Link Subnet, even if multiple LLNs are attached to the same 6BBR. ND messages are link-scoped and are not forwarded by the 6BBR between the backbone and the LLNs though some packets may be reinjected in Bridging Proxy mode (see Section 8).

Legacy nodes located on the backbone expect that the subnet is deployed within a single link and that there is a common Maximum Transmission Unit (MTU) for intra-subnet communication, the Link MTU. They will not perform the IPv6 Path MTU Discovery [RFC8201] for a destination within the subnet. For that reason, the MTU MUST have the same value on the Backbone and all federated LLNs in the MLSN. As a consequence, the 6BBR MUST use the same MTU value in RAs over the Backbone and in the RAs that it transmits towards the LLN links.

## 5. Optional 6LBR serving the Multi-Link Subnet

A 6LBR can be deployed to serve the whole MLSN. It may be attached to the backbone, in which case it can be discovered by its capability advertisement (see section 4.3. of [RFC8505]) in RA messages.

When a 6LBR is present, the 6BBR uses an EDAR/EDAC message exchange with the 6LBR to check if the new registration corresponds to a duplication or a movement. This is done prior to the NS(DAD) process, which may be avoided if the 6LBR already maintains a conflicting state for the Registered Address.

If this registration is duplicate or not the freshest, then the 6LBR replies with an EDAC message with a status code of 1 ("Duplicate Address") or 3 ("Moved"), respectively. If this registration is the freshest, then the 6LBR replies with a status code of 0. In that case, if this registration is fresher than an existing registration for another 6BBR, then the 6LBR also sends an asynchronous EDAC with a status of 4 ("Removed") to that other 6BBR.

The EDAR message SHOULD carry the SLLAO used in NS messages by the 6BBR for that Binding, and the EDAC message SHOULD carry the Target Link Layer Address Option (TLLAO) associated with the currently accepted registration. This enables a 6BBR to locate the new position of a mobile 6LN in the case of a Routing Proxy operation, and opens the capability for the 6LBR to serve as a mapping server in the future.

Note that if Link-Local Addresses are registered, then the scope of uniqueness on which the address duplication is checked is the total collection of links that the 6LBR serves as opposed to the sole link on which the Link-Local Address is assigned.

## 6. Using IPv6 ND Over the Backbone Link

On the Backbone side, the 6BBR MUST join the SNMA group corresponding to a Registered Address as soon as it creates a Binding for that Address, and maintain that SNMA membership as long as it maintains the registration. The 6BBR uses either the SNMA or plain unicast to defend the Registered Addresses in its Binding Table over the Backbone (as specified in [RFC4862]). The 6BBR advertises and defends the Registered Addresses over the Backbone Link using RS, NS(DAD) and NA messages with the Registered Address as the Source or Target address.

The 6BBR MUST place an EARO in the IPv6 ND messages that it generates on behalf of the Registered Node. Note that an NS(DAD) does not contain an SLLAO and cannot be confused with a proxy registration such as performed by a 6LBR.

IPv6 ND operates as follows on the backbone:

* Section 7.2.8 of [RFC4861] specifies that an NA message generated as a proxy does not have the Override flag set in order to ensure that if the real owner is present on the link, its own NA will take precedence, and that this NA does not update the NCE for the real owner if one exists.

* A node that receives multiple NA messages updates an existing NCE only if the Override flag is set; otherwise the node will probe the cached address.

* When an NS(DAD) is received for a tentative address, which means that two nodes form the same address at nearly the same time, section 5.4.3 of [RFC4862] cannot detect which node first claimed the address and the address is abandoned.

* In any case, [RFC4862] indicates that a node never responds to a Neighbor Solicitation for a tentative address.

This specification adds information about proxied addresses that helps sort out a duplication (different ROVR) from a movement (same ROVR, different TID), and in the latter case the older registration from the fresher one (by comparing TIDs).

When a Registering Node moves from one 6BBR to the next, the new 6BBR sends NA messages over the backbone to update existing NCEs. A node that supports this specification and that receives multiple NA messages with an EARO option and the same ROVR MUST favor the NA with the freshest EARO over the others.

The 6BBR MAY set the Override flag in the NA messages if it does not compete with the Registering Node for the NCE in backbone nodes. This is assured if the Registering Node is attached via an interface that cannot be bridged onto the backbone, making it impossible for the Registering Node to defend its own addresses there. This may also be signaled by the Registering Node through a protocol extension that is not in scope for this specification.

When the Binding is in Tentative state, the 6BBR acts as follows:

* an NS(DAD) that indicates a duplication can still not be asserted for first come, but the situation can be avoided using a 6LBR on the backbone that will serialize the order of appearance of the address and ensure first-come/first-serve.

* an NS or an NA that denotes an older registration for the same Registered Node is not interpreted as a duplication as specified in section 5.4.3 and 5.4.4 of [RFC4862], respectively.

When the Binding is no longer in Tentative state, the 6BBR acts as follows:

* an NS or an NA with an EARO that denotes a duplicate registration (different ROVR) is answered with an NA message that carries an EARO with a status of 1 (Duplicate), unless the received message is an NA that carries an EARO with a status of 1.

In any state, the 6BBR acts as follows:

* an NS or an NA with an EARO that denotes an older registration (same ROVR) is answered with an NA message that carries an EARO with a status of 3 (Moved) to ensure that the stale state is removed rapidly.

This behavior is specified in more detail in Section 9.

This specification enables proxy operation for the IPv6 ND resolution of LLN devices and a prefix that is used across a Multi-Link Subnet MAY be advertised as on-link over the Backbone. This is done for backward compatibility with existing IPv6 hosts by setting the L flag in the Prefix Information Option (PIO) of RA messages [RFC4861].

For movement involving a slow reattachment, the NUD procedure defined in [RFC4861] may time out too quickly. Nodes on the backbone SHOULD support [RFC7048] whenever possible.

## 7. Routing Proxy Operations

A Routing Proxy provides IPv6 ND proxy functions for Global and Unique Local addresses between the LLN and the backbone, but not for Link-Local addresses. It operates as an IPv6 border router and provides a full Link-Layer isolation.

In this mode, it is not required that the MAC addresses of the 6LNs are visible at Layer 2 over the Backbone. It is thus useful when the messaging over the Backbone that is associated to wireless mobility becomes expensive, e.g., when the Layer 2 topology is virtualized over a wide area IP underlay.

This mode is definitely required when the LLN uses a MAC address format that is different from that on the Backbone (e.g., EUI-64 vs. EUI-48). Since a 6LN may not be able to resolve an arbitrary destination in the MLSN directly, a prefix that is used across a MLSN MUST NOT be advertised as on-link in RA messages sent towards the LLN.

In order to maintain IP connectivity, the 6BBR installs a connected Host route to the Registered Address on the LLN interface, via the Registering Node as identified by the Source Address and the SLLA option in the NS(EARO) messages.

When operating as a Routing Proxy, the 6BBR MUST use its Layer 2 Address on its Backbone Interface in the SLLAO of the RS messages and the TLLAO of the NA messages that it generates to advertise the Registered Addresses.

For each Registered Address, multiple peers on the Backbone may have resolved the Address with the 6BBR MAC Address, maintaining that mapping in their Neighbor Cache. The 6BBR SHOULD maintain a list of the peers on the Backbone which have associated its MAC Address with the Registered Address. If that Registered Address moves to another 6BBR, the previous 6BBR SHOULD unicast a gratuitous NA to each such peer, to supply the LLA of the new 6BBR in the TLLA option for the Address. A 6BBR that does not maintain this list MAY multicast a gratuitous NA message; this NA will possibly hit all the nodes on the Backbone, whether or not they maintain an NCE for the Registered Address. In either case, the 6BBR MAY set the Override flag if it is known that the Registered Node cannot attach to the backbone, so as to avoid interruptions and save probing flows in the future.

If a correspondent fails to receive the gratuitous NA, it will keep sending traffic to a 6BBR to which the node was previously registered. Since the previous 6BBR removed its Host route to the Registered Address, it will look up the address over the backbone, resolve the address with the LLA of the new 6BBR, and forward the packet to the correct 6BBR. The previous 6BBR SHOULD also issue a redirect message [RFC4861] to update the cache of the correspondent.

## 8. Bridging Proxy Operations

A Bridging Proxy provides IPv6 ND proxy functions between the LLN and the backbone while preserving the forwarding continuity at the MAC Layer. It acts as a Layer 2 Bridge for all types of unicast packets including link-scoped, and appears as an IPv6 Host on the Backbone.

The Bridging Proxy registers any Binding including for a Link-Local address to the 6LBR (if present) and defends it over the backbone in IPv6 ND procedures.

To achieve this, the Bridging Proxy intercepts the IPv6 ND messages and may reinject them on the other side, respond directly or drop them. For instance, an ND(Lookup) from the backbone that matches a Binding can be responded directly, or turned into a unicast on the LLN side to let the 6LN respond.

As a Bridging Proxy, the 6BBR MUST use the Registering Node's Layer 2 Address in the SLLAO of the NS/RS messages and the TLLAO of the NA messages that it generates to advertise the Registered Addresses. The Registering Node's Layer 2 address is found in the SLLA of the registration NS(EARO), and maintained in the Binding Table.

The Multi-Link Subnet prefix SHOULD NOT be advertised as on-link in RA messages sent towards the LLN. If a destination address is seen as on-link, then a 6LN may use NS(Lookup) messages to resolve that address. In that case, the 6BBR MUST either answer the NS(Lookup) message directly or reinject the message on the backbone, either as a Layer 2 unicast or a multicast.

If the Registering Node owns the Registered Address, meaning that the Registering Node is the Registered Node, then its mobility does not impact existing NCEs over the Backbone. In a network where proxy registrations are used, meaning that the Registering Node acts on behalf of the Registered Node, if the Registered Node selects a new Registering Node then the existing NCEs across the Backbone pointing at the old Registering Node must be updated. In that case, the 6BBR SHOULD attempt to fix the existing NCEs across the Backbone pointing at other 6BBRs using NA messages as described in Section 7.

This method can fail if the multicast message is not received; one or more correspondent nodes on the Backbone might maintain an stale NCE, and packets to the Registered Address may be lost. When this condition happens, it is eventually discovered and resolved using NUD as defined in [RFC4861].

## 9. Creating and Maintaining a Binding

Upon receiving a registration for a new Address (i.e., an NS(EARO) with the R flag set), the 6BBR creates a Binding and operates as a 6LR according to [RFC8505], interacting with the 6LBR if one is present.

An implementation of a Routing Proxy that creates a Binding MUST also create an associated Host route pointing to the registering node in the LLN interface from which the registration was received.

Acting as a 6BBR, the 6LR operation is modified as follows:

* Acting as Bridging Proxy the 6LR MUST proxy ND over the backbone for registered Link-Local Addresses.

* EDAR and EDAC messages SHOULD carry a SLLAO and a TLLAO, respectively.

* An EDAC message with a status of 9 (6LBR Registry Saturated) is assimilated as a status of 0 if a following DAD process protects the address against duplication.

This specification enables nodes on a Backbone Link to co-exist along with nodes implementing IPv6 ND [RFC4861] as well as other non-normative specifications such as [I-D.bi-savi-wlan]. It is possible that not all IPv6 addresses on the Backbone are registered and known to the 6LBR, and an EDAR/EDAC echange with the 6LBR might succeed even for a duplicate address. Consequently the 6BBR still needs to perform IPv6 ND DAD over the backbone after an EDAC with a status code of 0 or 9.

For the DAD operation, the Binding is placed in Tentative state for a duration of TENTATIVE_DURATION (Section 12), and an NS(DAD) message is sent as a multicast message over the Backbone to the SNMA associated with the registered Address [RFC4862]. The EARO from the registration MUST be placed unchanged in the NS(DAD) message.

If a registration is received for an existing Binding with a non-null Registration Lifetime and the registration is fresher (same ROVR, fresher TID), then the Binding is updated, with the new Registration Lifetime, TID, and possibly Registering Node. In Tentative state

(see Section 9.1), the current DAD operation continues unaltered. In other states (see Section 9.2 and Section 9.3 ), the Binding is placed in Reachable state for the Registration Lifetime, and the 6BBR returns an NA(EARO) to the Registering Node with a status of 0 (Success).

Upon a registration that is identical (same ROVR, TID, and Registering Node), the 6BBR does not alter its current state. In Reachable State it returns an NA(EARO) back to the Registering Node with a status of 0 (Success). A registration that is not as fresh (same ROVR, older TID) is ignored.

If a registration is received for an existing Binding and a registration Lifetime of zero, then the Binding is removed, and the 6BBR returns an NA(EARO) back to the Registering Node with a status of 0 (Success). An implementation of a Routing Proxy that removes a binding MUST remove the associated Host route pointing on the registering node.

The old 6BBR removes its Binding Table entry and notifies the Registering Node with a status of 3 (Moved) if a new 6BBR claims a fresher registration (same ROVR, fresher TID) for the same address. The old 6BBR MAY preserve a temporary state in order to forward packets in flight. The state may for instance be a NCE formed based on a received NA message. It may also be a Binding Table entry in Stale state and pointing at the new 6BBR on the backbone, or any other abstract cache entry that can be used to resolve the Link-Layer Address of the new 6BBR. The old 6BBR SHOULD also use REDIRECT messages as specified in [RFC4861] to update the correspondents for the Registered Address, pointing to the new 6BBR.

### 9.1. Operations on a Binding in Tentative State

The Tentative state covers a DAD period over the backbone during which an address being registered is checked for duplication using procedures defined in [RFC4862].

For a Binding in Tentative state:

* The Binding MUST be removed if an NA message is received over the Backbone for the Registered Address with no EARO, or containing an EARO that indicates an existing registration owned by a different Registering Node (different ROVR). In that case, an NA is sent back to the Registering Node with a status of 1 (Duplicate) to indicate that the binding has been rejected. This behavior might be overridden by policy, in particular if the registration is trusted, e.g., based on the validation of the ROVR field (see [I-D.ietf-6lo-ap-nd]).

* The Binding MUST be removed if an NS(DAD) message is received over the Backbone for the Registered Address with no EARO, or containing an EARO with a different ROVR that indicates a tentative registration by a different Registering Node. In that case, an NA is sent back to the Registering Node with a status of 1 (Duplicate). This behavior might be overridden by policy, in particular if the registration is trusted, e.g., based on the validation of the ROVR field (see [I-D.ietf-6lo-ap-nd]).

* The Binding MUST be removed if an NA or an NS(DAD) message is received over the Backbone for the Registered Address containing an EARO with a that indicates a fresher registration ([RFC8505]) for the same Registering Node (same ROVR). In that case, an NA MUST be sent back to the Registering Node with a status of 3 (Moved).

* The Binding MUST be kept unchanged if an NA or an NS(DAD) message is received over the Backbone for the Registered Address containing an EARO with a that indicates an older registration ([RFC8505]) for the same Registering Node (same ROVR). The message is answered with an NA that carries an EARO with a status of 3 (Moved) and the Override flag not set. This behavior might be overridden by policy, in particular if the registration is not trusted.

* Other NS(DAD) and NA messages from the Backbone are ignored.

* NS(Lookup) and NS(NUD) messages SHOULD be optimistically answered with an NA message containing an EARO with a status of 0 and the Override flag not set (see Section 3.6). If optimistic DAD is disabled, then they SHOULD be queued to be answered when the Binding goes to Reachable state.

When the TENTATIVE_DURATION (Section 12) timer elapses, the Binding is placed in Reachable state for the Registration Lifetime, and the 6BBR returns an NA(EARO) to the Registering Node with a status of 0 (Success).

The 6BBR also attempts to take over any existing Binding from other 6BBRs and to update existing NCEs in backbone nodes. This is done by sending an NA message with an EARO and the Override flag not set over the backbone (see Section 7 and Section 8).

### 9.2. Operations on a Binding in Reachable State

The Reachable state covers an active registration after a successful DAD process.

If the Registration Lifetime is of a long duration, an implementation might be configured to reassess the availability of the Registering Node at a lower period, using a NUD procedure as specified in [RFC7048]. If the NUD procedure fails, the Binding SHOULD be placed in Stale state immediately.

For a Binding in Reachable state:

* The Binding MUST be removed if an NA or an NS(DAD) message is received over the Backbone for the Registered Address containing an EARO that indicates a fresher registration ([RFC8505]) for the same Registered Node (i.e., same ROVR but fresher TID). A status of 4 (Removed) is returned in an asynchronous NA(EARO) to the Registering Node. Based on configuration, an implementation may delay this operation by a timer with a short setting, e.g., a few seconds to a minute, in order to a allow for a parallel registration to reach this node, in which case the NA might be ignored.

* NS(DAD) and NA messages containing an EARO that indicates a registration for the same Registered Node that is not as fresh as this binding MUST be answered with an NA message containing an EARO with a status of 3 (Moved).

* An NS(DAD) with no EARO or with an EARO that indicates a duplicate registration (i.e., different ROVR) MUST be answered with an NA message containing an EARO with a status of 1 (Duplicate) and the Override flag not set, unless the received message is an NA that carries an EARO with a status of 1, in which case the node refrains from answering.

* Other NS(DAD) and NA messages from the Backbone are ignored.

* NS(Lookup) and NS(NUD) messages SHOULD be answered with an NA message containing an EARO with a status of 0 and the Override flag not set. The 6BBR MAY check whether the Registering Node is still available using a NUD procedure over the LLN prior to answering; this behaviour depends on the use case and is subject to configuration.

When the Registration Lifetime timer elapses, the Binding is placed in Stale state for a duration of STALE_DURATION (Section 12).

### 9.3. Operations on a Binding in Stale State

The Stale state enables tracking of the Backbone peers that have a NCE pointing to this 6BBR in case the Registered Address shows up later.

If the Registered Address is claimed by another 6LN on the Backbone, with an NS(DAD) or an NA, the 6BBR does not defend the Address.

For a Binding in Stale state:

* The Binding MUST be removed if an NA or an NS(DAD) message is received over the Backbone for the Registered Address containing no EARO or an EARO that indicates either a fresher registration for the same Registered Node or a duplicate registration. A status of 4 (Removed) MAY be returned in an asynchronous NA(EARO) to the Registering Node.

* NS(DAD) and NA messages containing an EARO that indicates a registration for the same Registered Node that is not as fresh as this MUST be answered with an NA message containing an EARO with a status of 3 (Moved).

* If the 6BBR receives an NS(Lookup) or an NS(NUD) message for the Registered Address, the 6BBR MUST attempt a NUD procedure as specified in [RFC7048] to the Registering Node, targeting the Registered Address, prior to answering. If the NUD procedure succeeds, the operation in Reachable state applies. If the NUD fails, the 6BBR refrains from answering.

* Other NS(DAD) and NA messages from the Backbone are ignored.

When the STALE_DURATION (Section 12) timer elapses, the Binding MUST be removed.

## 10. Registering Node Considerations

A Registering Node MUST implement [RFC8505] in order to interact with a 6BBR (which acts as a routing registrar). Following [RFC8505], the Registering Node signals that it requires IPv6 proxy-ND services from a 6BBR by registering the corresponding IPv6 Address using an NS(EARO) message with the R flag set.

The Registering Node may be the 6LN owning the IPv6 Address, or a 6LBR that performs the registration on its behalf in a Route-Over mesh.

A 6LN MUST register all of its IPv6 Addresses to its 6LR, which is the 6BBR when they are connected at Layer 2. Failure to register an address may result in the address being unreachable by other parties. This would happen for instance if the 6BBR propagates the NS(Lookup) from the backbone only to the LLN nodes that do not register their addresses.

The Registering Node MUST refrain from using multicast NS(Lookup) when the destination is not known as on-link, e.g., if the prefix is advertised in a PIO with the L flag that is not set. In that case, the Registering Node sends its packets directly to its 6LR.

The Registering Node SHOULD also follow BCP 202 [RFC7772] in order to limit the use of multicast RAs. It SHOULD also implement Simple Procedures for Detecting Network Attachment in IPv6 [RFC6059] (DNA procedures) to detect movements, and support Packet-Loss Resiliency for Router Solicitations [RFC7559] in order to improve reliability for the unicast RS messages.

## 11. Security Considerations

The procedures in this document modify the mechanisms used for IPv6 ND and DAD and should not affect other aspects of IPv6 or higher-level-protocol operation. As such, the main classes of attacks that are in play are those which week to block neighbor discovery or to forcibly claim an address that another node is attempting to use. In the absence of cryptographic protection at higher layers, the latter class of attacks can have significant consequences, with the attacker being able to read all the "stolen" traffic that was directed to the target of the attack.

This specification applies to LLNs and a backbone in which the individual links are protected against rogue access, on the LLN by authenticating a node that attaches to the network and encrypting at the MAC layer the transmissions, and on the backbone side using the physical security and access control measures that are typically applied there, so packets may neither be forged or nor overheard.

In particular, the LLN MAC is required to provide secure unicast to/from the Backbone Router and secure broadcast from the routers in a way that prevents tampering with or replaying the ND messages.

For the IPv6 ND operation over the backbone, and unless the classical ND is disabled (e.g., by configuration), the classical ND messages are interpreted as emitted by the address owner and have precedence over the 6BBR that is only a proxy.

It results that the security threats that are detailed in section 11.1 of [RFC4861] fully apply to this specification as well. In very short:

* Any node that can send a packet on the backbone can take over any address including addresses of LLN nodes by claiming it with an NA message and the Override bit set. This means that the real owner will stop receiving its packets.

* Any node that can send a packet on the backbone can forge traffic and pretend it is issued from a address that it does not own, even if it did not claim the address using ND.

* Any node that can send a packet on the backbone can present itself as a preferred router to intercept all traffic outgoing the subnet. It may even expose a prefix on the subnet as not-on-link and intercept all the traffic within the subnet.

* If the rogue can receive a packet from the backbone it can also snoop all the intercepted traffic, be it by stealing an address or the role of a router.

This means that any rogue access to the backbone must be prevented at all times, and that nodes that are attached to the backbone must be fully trusted / never compromised.

Using address registration as the sole ND mechanism on a link and coupling it with [I-D.ietf-6lo-ap-nd] guarantees the ownership of a registered address within that link.

* The protection is based on a proof-of-ownership encoded in the ROVR field and protects against address theft and impersonation by a 6LN, because the 6LR can challenge the Registered Node for a proof-of-ownership.

* The protection extends to the full LLN in the case of an LLN Link, but does not extend over the backbone since the 6BBR cannot provide the proof-of-ownership when it defends the address.

A possible attack over the backbone can be done by sending an NS with an EARO and expecting the NA(EARO) back to contain the TID and ROVR fields of the existing state. With that information, the attacker can easily increase the TID and take over the Binding.

If the classical ND is disabled on the backbone and the use of [I-D.ietf-6lo-ap-nd] and a 6LBR are mandated, the network will benefit from the following new advantages:

Zero-trust security for ND flows within the whole subnet: the increased security that [I-D.ietf-6lo-ap-nd] provides on the LLN will also apply to the backbone; it becomes impossible for an attached node to claim an address that belongs to another node using ND, and the network can filter packets that are not originated by the owner of the source address (SAVI), as long as that the routers are known and trusted.

Remote ND DoS attack avoidance: the complete list of addresses in the network will be known to the 6LBR and available to the default router; with that information the router does not need to send a multicast NA(Lookup) in case of a Neighbor Cache miss for an incoming packet, which is a source of remote DoS attack against the network

Less IPv6 ND-related multicast on the backbone: DAD and NS(Lookup) become unicast queries to the 6LBR

Better DAD operation on wireless: DAD has been found to fail to detect duplications on large Wi-Fi infrastructures due to the unreliable broadcast operation on wireless; using a 6LBR enables a unicast lookup

Less Layer-2 churn on the backbone: Using the Routing Proxy approach, the Link-Layer address of the LLN devices and their mobility are not visible in the backbone; only the Link-Layer addresses of the 6BBR and backbone nodes are visible at Layer 2 on the backbone. This is mandatory for LLNs that cannot be bridged on the backbone, and useful in any case to scale down, stabilize the forwarding tables at Layer 2 and avoid the gratuitous frames that are typically broadcasted to fix the transparent bridging tables when a wireless node roams from an AP to the next.

This specification introduces a 6BBR that is a router on the path of the LLN traffic and a 6LBR that is used for the lookup. They could be interesting targets for an attacker. A compromised 6BBR can accept a registration but block the traffic, or refrain from proxying. A compromised 6LBR may accept unduly the transfer of ownership of an address, or block a new comer by faking that its address is a duplicate. But those attacks are possible in a classical network from a compromised default router and a DHCP server, respectively, and can be prevented using the same methods.

A possible attack over the LLN can still be done by compromising a 6LR. A compromised 6LR may modify the ROVR of EDAR messages in flight and transfer the ownership of the Registered Address to itself or a tier. It may also claim that a ROVR was validated when it really wasn't, and reattribute an address to self or to an attached 6LN. This means that 6LRs, as well as 6LBRs and 6BBRS must still be fully trusted / never compromised.

This specification mandates to check on the 6LBR on the backbone before doing the classical DAD, in case the address already exists. This may delay the DAD operation and should be protected by a short timer, in the order of 100ms or less, which will only represent a small extra delay versus the 1s wait of the DAD operation.

## 12. Protocol Constants

This Specification uses the following constants:

TENTATIVE_DURATION: 800 milliseconds

STALE_DURATION: see below

In LLNs with long-lived Addresses such as LPWANs, STALE_DURATION SHOULD be configured with a relatively long value to cover an interval when the address may be reused, and before it is safe to expect that the address was definitively released. A good default value can be 24 hours. In LLNs where addresses are renewed rapidly, e.g., for privacy reasons, STALE_DURATION SHOULD be configured with a relatively shorter value, by default 5 minutes.

## 13. IANA Considerations

This document has no request to IANA.

## 14. Acknowledgments

Many thanks to Dorothy Stanley, Thomas Watteyne and Jerome Henry for their various contributions. Also many thanks to Timothy Winters and Erik Nordmark for their help, review and support in preparation to the IESG cycle, and to Kyle Rose, Elwyn Davies, Barry Leiba, Mirja Kuhlewind, Alvaro Retana, Roman Danyliw and very especially Dominique Barthel and Benjamin Kaduk for their useful contributions through the IETF last call and IESG process.

## 15. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

[RFC4291] Hinden, R. and S. Deering, "IP Version 6 Addressing Architecture", RFC 4291, DOI 10.17487/RFC4291, February 2006, <https://www.rfc-editor.org/info/rfc4291>.

[RFC4429] Moore, N., "Optimistic Duplicate Address Detection (DAD) for IPv6", RFC 4429, DOI 10.17487/RFC4429, April 2006, <https://www.rfc-editor.org/info/rfc4429>.

[RFC4861] Narten, T., Nordmark, E., Simpson, W., and H. Soliman, "Neighbor Discovery for IP version 6 (IPv6)", RFC 4861,

DOI 10.17487/RFC4861, September 2007,
<https://www.rfc-editor.org/info/rfc4861>.

[RFC4862] Thomson, S., Narten, T., and T. Jinmei, "IPv6 Stateless Address Autoconfiguration", RFC 4862, DOI 10.17487/RFC4862, September 2007, <https://www.rfc-editor.org/info/rfc4862>.

[RFC6059] Krishnan, S. and G. Daley, "Simple Procedures for Detecting Network Attachment in IPv6", RFC 6059, DOI 10.17487/RFC6059, November 2010, <https://www.rfc-editor.org/info/rfc6059>.

[RFC6775] Shelby, Z., Ed., Chakrabarti, S., Nordmark, E., and C. Bormann, "Neighbor Discovery Optimization for IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs)", RFC 6775, DOI 10.17487/RFC6775, November 2012, <https://www.rfc-editor.org/info/rfc6775>.

[RFC7048] Nordmark, E. and I. Gashinsky, "Neighbor Unreachability Detection Is Too Impatient", RFC 7048, DOI 10.17487/RFC7048, January 2014, <https://www.rfc-editor.org/info/rfc7048>.

[RFC7559] Krishnan, S., Anipko, D., and D. Thaler, "Packet-Loss Resiliency for Router Solicitations", RFC 7559, DOI 10.17487/RFC7559, May 2015, <https://www.rfc-editor.org/info/rfc7559>.

[RFC7772] Yourtchenko, A. and L. Colitti, "Reducing Energy Consumption of Router Advertisements", BCP 202, RFC 7772, DOI 10.17487/RFC7772, February 2016, <https://www.rfc-editor.org/info/rfc7772>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

[RFC8200] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", STD 86, RFC 8200, DOI 10.17487/RFC8200, July 2017, <https://www.rfc-editor.org/info/rfc8200>.

[RFC8201] McCann, J., Deering, S., Mogul, J., and R. Hinden, Ed., "Path MTU Discovery for IP version 6", STD 87, RFC 8201, DOI 10.17487/RFC8201, July 2017, <https://www.rfc-editor.org/info/rfc8201>.

[RFC8505] Thubert, P., Ed., Nordmark, E., Chakrabarti, S., and C. Perkins, "Registration Extensions for IPv6 over Low-Power Wireless Personal Area Network (6LoWPAN) Neighbor Discovery", RFC 8505, DOI 10.17487/RFC8505, November 2018, <https://www.rfc-editor.org/info/rfc8505>.

## 16. Informative References

[RFC4389] Thaler, D., Talwar, M., and C. Patel, "Neighbor Discovery Proxies (ND Proxy)", RFC 4389, DOI 10.17487/RFC4389, April 2006, <https://www.rfc-editor.org/info/rfc4389>.

[RFC4903] Thaler, D., "Multi-Link Subnet Issues", RFC 4903, DOI 10.17487/RFC4903, June 2007, <https://www.rfc-editor.org/info/rfc4903>.

[RFC5415] Calhoun, P., Ed., Montemurro, M., Ed., and D. Stanley, Ed., "Control And Provisioning of Wireless Access Points (CAPWAP) Protocol Specification", RFC 5415, DOI 10.17487/RFC5415, March 2009, <https://www.rfc-editor.org/info/rfc5415>.

[RFC5568] Koodli, R., Ed., "Mobile IPv6 Fast Handovers", RFC 5568, DOI 10.17487/RFC5568, July 2009, <https://www.rfc-editor.org/info/rfc5568>.

[RFC6606] Kim, E., Kaspar, D., Gomez, C., and C. Bormann, "Problem Statement and Requirements for IPv6 over Low-Power Wireless Personal Area Network (6LoWPAN) Routing", RFC 6606, DOI 10.17487/RFC6606, May 2012, <https://www.rfc-editor.org/info/rfc6606>.

[RFC6275] Perkins, C., Ed., Johnson, D., and J. Arkko, "Mobility Support in IPv6", RFC 6275, DOI 10.17487/RFC6275, July 2011, <https://www.rfc-editor.org/info/rfc6275>.

[RFC6550] Winter, T., Ed., Thubert, P., Ed., Brandt, A., Hui, J., Kelsey, R., Levis, P., Pister, K., Struik, R., Vasseur, JP., and R. Alexander, "RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks", RFC 6550, DOI 10.17487/RFC6550, March 2012, <https://www.rfc-editor.org/info/rfc6550>.

[RFC6830] Farinacci, D., Fuller, V., Meyer, D., and D. Lewis, "The Locator/ID Separation Protocol (LISP)", RFC 6830, DOI 10.17487/RFC6830, January 2013, <https://www.rfc-editor.org/info/rfc6830>.

[RFC8273] Brzozowski, J. and G. Van de Velde, "Unique IPv6 Prefix per Host", RFC 8273, DOI 10.17487/RFC8273, December 2017, <https://www.rfc-editor.org/info/rfc8273>.

[I-D.yourtchenko-6man-dad-issues]
Yourtchenko, A. and E. Nordmark, "A survey of issues related to IPv6 Duplicate Address Detection", Work in Progress, Internet-Draft, draft-yourtchenko-6man-dad-issues-01, 3 March 2015, <https://tools.ietf.org/html/draft-yourtchenko-6man-dad-issues-01>.

[I-D.nordmark-6man-dad-approaches]
Nordmark, E., "Possible approaches to make DAD more robust and/or efficient", Work in Progress, Internet-Draft, draft-nordmark-6man-dad-approaches-02, 19 October 2015, <https://tools.ietf.org/html/draft-nordmark-6man-dad-approaches-02>.

[I-D.ietf-6man-rs-refresh]
Nordmark, E., Yourtchenko, A., and S. Krishnan, "IPv6 Neighbor Discovery Optional RS/RA Refresh", Work in Progress, Internet-Draft, draft-ietf-6man-rs-refresh-02, 31 October 2016, <https://tools.ietf.org/html/draft-ietf-6man-rs-refresh-02>.

[I-D.ietf-6lo-ap-nd]
Thubert, P., Sarikaya, B., Sethi, M., and R. Struik, "Address Protected Neighbor Discovery for Low-power and Lossy Networks", Work in Progress, Internet-Draft, draft-ietf-6lo-ap-nd-20, 9 March 2020, <https://tools.ietf.org/html/draft-ietf-6lo-ap-nd-20>.

[I-D.ietf-6tisch-architecture]
Thubert, P., "An Architecture for IPv6 over the TSCH mode of IEEE 802.15.4", Work in Progress, Internet-Draft, draft-ietf-6tisch-architecture-28, 29 October 2019, <https://tools.ietf.org/html/draft-ietf-6tisch-architecture-28>.

[I-D.ietf-mboned-ieee802-mcast-problems]
Perkins, C., McBride, M., Stanley, D., Kumari, W., and J. Zuniga, "Multicast Considerations over IEEE 802 Wireless Media", Work in Progress, Internet-Draft, draft-ietf-mboned-ieee802-mcast-problems-11, 11 December 2019, <https://tools.ietf.org/html/draft-ietf-mboned-ieee802-mcast-problems-11>.

[I-D.bi-savi-wlan]
Bi, J., Wu, J., Wang, Y., and T. Lin, "A SAVI Solution for WLAN", Work in Progress, Internet-Draft, draft-bi-savi-wlan-18, 17 November 2019, <https://tools.ietf.org/html/draft-bi-savi-wlan-18>.

[I-D.thubert-6lo-unicast-lookup]
Thubert, P. and E. Levy-Abegnoli, "IPv6 Neighbor Discovery Unicast Lookup", Work in Progress, Internet-Draft, draft-thubert-6lo-unicast-lookup-00, 25 January 2019, <https://tools.ietf.org/html/draft-thubert-6lo-unicast-lookup-00>.

[IEEEstd8021]
IEEE standard for Information Technology, "IEEE Standard for Information technology -- Telecommunications and information exchange between systems Local and metropolitan area networks Part 1: Bridging and Architecture".

[IEEEstd80211]
IEEE standard for Information Technology, "IEEE Standard for Information technology -- Telecommunications and information exchange between systems Local and metropolitan area networks-- Specific requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications".

[IEEEstd802151]
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)".

[IEEEstd802154]
IEEE standard for Information Technology, "IEEE Standard for Local and metropolitan area networks -- Part 15.4: Low-Rate Wireless Personal Area Networks (LR-WPANs)".

## Appendix A. Possible Future Extensions

With the current specification, the 6LBR is not leveraged to avoid multicast NS(Lookup) on the Backbone. This could be done by adding a lookup procedure in the EDAR/EDAC exchange.

By default the specification does not have a fine-grained trust model: all nodes that can authenticate to the LLN MAC or attach to the backbone are equally trusted. It would be desirable to provide a stronger authorization model, e.g., whereby nodes that associate their address with a proof-of-ownership [I-D.ietf-6lo-ap-nd] should be more trusted than nodes that do not. Such a trust model and related signaling could be added in the future to override the default operation and favor trusted nodes.

Future documents may extend this specification by allowing the 6BBR to redistribute Host routes in routing protocols that would operate over the Backbone, or in MIPv6 [RFC6275], or FMIP [RFC5568], or the Locator/ID Separation Protocol (LISP) [RFC6830] to support mobility on behalf of the 6LNs, etc... LISP may also be used to provide an equivalent to the EDAR/EDAC exchange using a Map Server / Map Resolver as a replacement to the 6LBR.

## Appendix B. Applicability and Requirements Served

This document specifies proxy-ND functions that can be used to federate an IPv6 Backbone Link and multiple IPv6 LLNs into a single Multi-Link Subnet. The proxy-ND functions enable IPv6 ND services for Duplicate Address Detection (DAD) and Address Lookup that do not require broadcasts over the LLNs.

The term LLN is used to cover multiple types of WLANs and WPANs, including (Low-Power) Wi-Fi, BLUETOOTH(R) Low Energy, IEEE STD 802.11ah and IEEE STD.802.15.4 wireless meshes, covering the types of networks listed in Appendix B.3 of [RFC8505] "Requirements Related to Various Low-Power Link Types".

Each LLN in the subnet is attached to an IPv6 Backbone Router (6BBR). The Backbone Routers interconnect the LLNs and advertise the Addresses of the 6LNs over the Backbone Link using proxy-ND operations.

This specification updates IPv6 ND over the Backbone to distinguish Address movement from duplication and eliminate stale state in the Backbone routers and Backbone nodes once a 6LN has roamed. This way, mobile nodes may roam rapidly from one 6BBR to the next and requirements in Appendix B.1 of [RFC8505] "Requirements Related to Mobility" are met.

A 6LN can register its IPv6 Addresses and thereby obtain proxy-ND services over the Backbone, meeting the requirements expressed in Appendix B.4 of [RFC8505], "Requirements Related to Proxy Operations".

The negative impact of the IPv6 ND-related broadcasts can be limited to one of the federated links, enabling the number of 6LNs to grow. The Routing Proxy operation avoids the need to expose the MAC addresses of the 6LNs onto the backbone, keeping the Layer 2 topology simple and stable. This meets the requirements in Appendix B.6 of [RFC8505] "Requirements Related to Scalability", as long has the 6BBRs are dimensioned for the number of registrations that each needs to support.

In the case of a Wi-Fi access link, a 6BBR may be collocated with the Access Point (AP), or with a Fabric Edge (FE) or a CAPWAP [RFC5415] Wireless LAN Controller (WLC). In those cases, the wireless client (STA) is the 6LN that makes use of [RFC8505] to register its IPv6 Address(es) to the 6BBR acting as Routing Registrar. The 6LBR can be centralized and either connected to the Backbone Link or reachable over IP. The 6BBR proxy-ND operations eliminate the need for wireless nodes to respond synchronously when a Lookup is performed for their IPv6 Addresses. This provides the function of a Sleep Proxy for ND [I-D.nordmark-6man-dad-approaches].

For the TimeSlotted Channel Hopping (TSCH) mode of [IEEEstd802154], the 6TiSCH architecture [I-D.ietf-6tisch-architecture] describes how a 6LoWPAN ND host could connect to the Internet via a RPL mesh Network, but doing so requires extensions to the 6LOWPAN ND protocol to support mobility and reachability in a secure and manageable environment. The extensions detailed in this document also work for the 6TiSCH architecture, serving the requirements listed in Appendix B.2 of [RFC8505] "Requirements Related to Routing Protocols".

The registration mechanism may be seen as a more reliable alternate to snooping [I-D.bi-savi-wlan]. It can be noted that registration and snooping are not mutually exclusive. Snooping may be used in conjunction with the registration for nodes that do not register their IPv6 Addresses. The 6BBR assumes that if a node registers at least one IPv6 Address to it, then the node registers all of its Addresses to the 6BBR. With this assumption, the 6BBR can possibly cancel all undesirable multicast NS messages that would otherwise have been delivered to that node.

Scalability of the Multi-Link Subnet [RFC4903] requires avoidance of multicast/broadcast operations as much as possible even on the Backbone [I-D.ietf-mboned-ieee802-mcast-problems]. Although hosts can connect to the Backbone using IPv6 ND operations, multicast RAs can be saved by using [I-D.ietf-6man-rs-refresh], which also requires the support of [RFC7559].

Authors' Addresses

Pascal Thubert (editor)
Cisco Systems, Inc
Building D
45 Allee des Ormes - BP1200
06254 MOUGINS - Sophia Antipolis
France

Phone: +33 497 23 26 34
Email: pthubert@cisco.com

Charles E. Perkins
Blue Meadow Networking
Saratoga, 95070
United States of America

Email: charliep@computer.org

Eric Levy-Abegnoli
Cisco Systems, Inc
Building D
45 Allee des Ormes - BP1200
06254 MOUGINS - Sophia Antipolis
France

Phone: +33 497 23 26 20
Email: elevyabe@cisco.com

6lo
Internet-Draft
Updates: 4944, 6282 (if approved)
Intended status: Standards Track
Expires: June 11, 2017

S. Chakrabarti

G. Montenegro
Microsoft
R. Droms

J. Woodyatt
Google

December 8, 2016

# 6lowpan ESC Dispatch Code Points and Guidelines
## draft-ietf-6lo-dispatch-iana-registry-07

Abstract

RFC4944 defines the ESC dispatch type to allow for additional dispatch octets in the 6lowpan header. The value of the ESC dispatch type was updated by RFC6282, however, its usage was not defined either in RFC6282 or in RFC4944. This document updates RFC4944 and RFC6282 by defining the ESC extension octet code points including registration of entries for known use cases at the time of writing of this document.

Table of Contents

## 1. Introduction

[RFC4944] section 5.1 defines the dispatch header and types. The ESC type is defined for using additional dispatch octets in the 6lowpan header. RFC 6282 modifies the value of the ESC dispatch type and that value is recorded in IANA registry [6LOWPAN-IANA]. However, the octets and usage following the ESC dispatch type are not defined in either [RFC4944] and [RFC6282]. In recent years with 6lowpan deployments, implementations and standards organizations have started using the ESC extension octets. This highlights the need for an updated IANA registration policy.

The following sections record the ITU-T specification for ESC dispatch octet code points as an existing known usage and propose the definition of ESC extension octets for future applications. The document also requests IANA actions for the first extension octet following the ESC dispatch type.

## 2. Terminology

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

## 3. Usage of ESC dispatch octets

RFC 4944 [RFC4944] first introduces this "ESC" dispatch header type for extension of dispatch octets. RFC 6282 [RFC6282] subsequently modified its value to [01 000000].

This document specifies that the first octet following the ESC dispatch type be used for extension type (extended dispatch values). Subsequent octets are left unstructured for the specific use of the extension type:

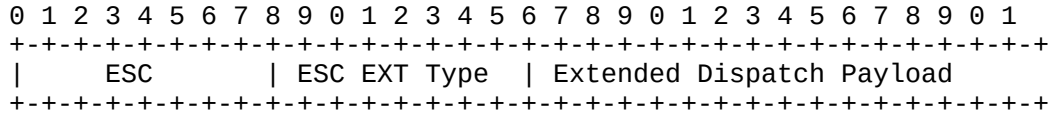



Figure 1: Frame Format with ESC dispatch type

ESC: The left-most octet is the ESC dispatch type containing

’01000000’

ESC Extension Type (EET): It is the first octet following the ESC dispatch type. Extension type defines the payload for the additional dispatch octets. The values are from 0 to 255. Values 0 and 255 are reserved for future use. The remaining values from 1 to 254 are assigned by IANA. The EET values are similar to dispatch values in the 6lowpan header except they are preceded by the ESC dispatch type. Thus, ESC extension types and dispatch values are using orthogonal code spaces. Though not desirable, multiple ESC dispatch types MAY appear in a 6lowpan header. Section 3.1 describes how to handle an unknown ESC dispatch type.

Extended Dispatch Payload (EDP): This part of the frame format must be defined by the corresponding extension type. A specification is required to define each usage of extension type and its corresponding Extension Payload. For the sake of interoperability, specifications of extension octets MUST NOT redefine the existing ESC Extension Type codes.

Section 5.1 in RFC4944 indicates that the Extension Type field may contain additional dispatch values larger than 63, as corrected by [4944-ERRATA]. For the sake of interoperability, the new dispatch type (EET) MUST NOT modify the behavior of existing dispatch types [RFC4944].

## 3.1. Interaction with other RFC4944 implementations

It is expected that existing implementations of RFC4944 are not capable of processing ESC extension data octets as defined in this document. However, implementers have to assume that existing implementation that attempt to process an EET unknown to them will simply drop the packet or ignore the ESC dispatch octets.

If an implementation following this document, during processing of the received packet reaches an ESC dispatch type for which it does not understand the extension octets (EET), it MUST drop that packet. However, it is important to clarify that a router node SHOULD forward a 6lowpan packet with the EET octets as long as it does not attempt to process any unknown ESC extension octets.

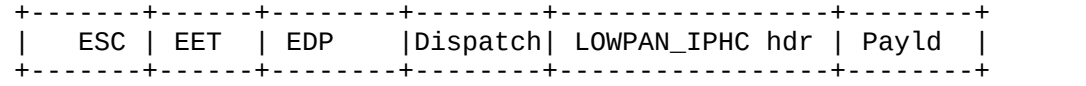
Multiple ESC extension octets may appear in a packet. The ESC dispatch types can appear as the first, last or middle dispatch octets. However, a packet will get dropped by any node that does not understand the EET at the beginning of the packet. Placing an EET toward the front of the packet has a greater probability of causing the packet to be dropped than placing the same EET later in the packet. Placement of an EET later in the packet increases the chance

that a legacy device will recognize and successfully process some dispatch type [RFC4944] before the EET. In this case, the legacy device will ignore the EET instead of dropping the entire packet.

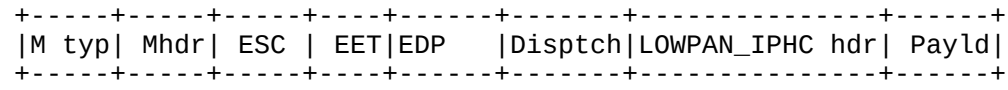
### 3.2. ESC Extension Octets Typical Sequence

ESC Extension octets sequence and order with respect to 6LoWPAN Mesh header and LoWPAN_IPHC header are described below. When LOWPAN_IPHC dispatch type is present, ESC dispatch types MUST appear before the LOWPAN_IPHC dispatch type in order to maintain backward compatibility with RFC6282 section 3.2. The following diagrams provide examples of ESC extension octet usages:

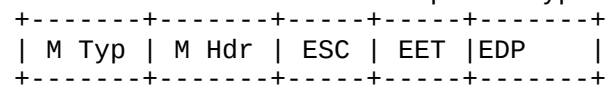
A LoWPAN encapsulated IPv6 Header compressed packet:



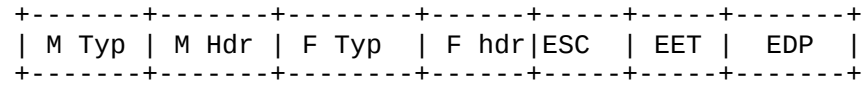

A LoWPAN_IPHC Header, Mesh header and an ESC extension octet:




A Mesh header with ESC dispatch types




With Fragment header




ESC dispatch type as a LowPAN encapsulation

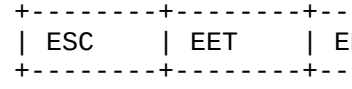



Figure 2: A 6lowpan packet with ESC dispatch types

### 3.3. ITU-T G.9903 ESC type usage

The ESC dispatch type is used in [G3-PLC] to provide native mesh routing and bootstrapping functionalities. The ITU-T recommendation [G3-PLC] section 9.4.2.3 defines commands which are formatted like ESC Extension type fields. The command ID values are 0x01 to 0x1F.

The frame format is defined as follows:

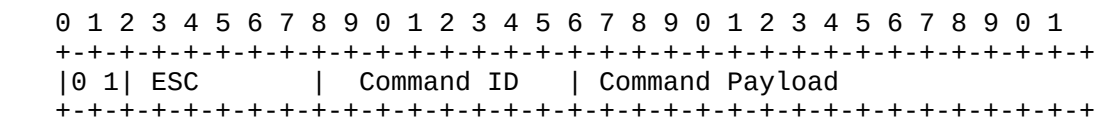



Figure 3: G.9903 Frame Format with ESC dispatch type

### 3.4. NALP and ESC dispatch types

According to RFC4944 [RFC4944] section 5.1, NALP dispatch octets are reserved for use as a kind of escape code for identification of non-6lowpan payloads. Since ESC dispatch types are part of 6lowpan dispatch types (extended), they are orthogonal to NALP octets.

This document clarifies that NALP dispatch codes only provide an escape method for non-6LoWPAN payloads when they appear as the initial octet of a LoWPAN encapsulation, and that the potential meaning of their appearance in any other location is reserved for future use.

## 4. IANA Considerations

This document requests IANA to register the ’ESC Extension Type’ values per the policy ’Specification Required’ [RFC5226], following the same policy as in the IANA Considerations section of [RFC4944]. For each Extension Type (except the Reserved values) the specification MUST define corresponding Extended Dispatch Payload frame octets for the receiver implementation to read the ESC dispatch types in an interoperable fashion.

[RFC5226] section 4.1 also indicates that "Specification Required" calls for a Designated Expert review of the public specification requesting registration of the ESC Extension Type values.

The allocation of code points should follow the guidelines on "Usage

of ESC dispatch octets" and the typical example sections. ESC Extension type code points MUST be used in conjunction with 6lo protocols following [RFC4944] or its derivatives. The requesting document MUST specify how the ESC dispatch octets will be used along with 6LOWPAN headers in their use cases.

The initial values for the ’ESC Extension Type’ fields are:

| Value | Description | Reference |
|---|---|---|
| 0 | Reserved for future use | This document |
| 1-31 | Used by ITU-T G.9903 and G.9905 Command IDs | ITU-T G.9903 & ITU-T G.9905 |
| 32-254 | Unassigned (Reserved for future IANA Assignment-- Spec Required) | This document |
| 255 | Reserved for future use | This document |

Figure 4: Initial Values for IANA Registry

## 5. Security Considerations

There are no additional security threats due to the assignments of ESC dispatch type usage described in this document. Furthermore, this document forbids defining any extended dispatch values or extension types that modify the behavior of existing Dispatch types.

## 6. Acknowledgements

The authors would like to thank the members of the 6lo WG for their comments. Many thanks to Carsten Bormann, Ralph Droms, Thierry Lys, Cedric Lavenu, Pascal Thubert for discussions regarding the bits allocation issues, which led to this document. Jonathan Hui and Robert Cragie provided extensive reviews and guidance for interoperability. The authors acknowledge the comments from the following people that helped shape this document: Paul Duffy, Don Sturek, Michael Richardson, Xavier Vilajosana, Scott Mansfield, Dale Worley and Russ Housley. Thanks to Brian Haberman, our document shepherd, for guidance in the IANA Considerations section.

This document was produced using the xml2rfc tool.

## 7. References

### 7.1. Normative References

[4944-ERRATA]
"https://www.rfc-editor.org/errata_search.php?rfc=4944".

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <http://www.rfc-editor.org/info/rfc2119>.

[RFC4944] Montenegro, G., Kushalnagar, N., Hui, J., and D. Culler, "Transmission of IPv6 Packets over IEEE 802.15.4 Networks", RFC 4944, DOI 10.17487/RFC4944, September 2007, <http://www.rfc-editor.org/info/rfc4944>.

[RFC6282] Hui, J., Ed. and P. Thubert, "Compression Format for IPv6 Datagrams over IEEE 802.15.4-Based Networks", RFC 6282, DOI 10.17487/RFC6282, September 2011, <http://www.rfc-editor.org/info/rfc6282>.

### 7.2. Informative References

[6LOWPAN-IANA]
"https://www.iana.org/assignments/_6lowpan-parameters/_6lowpan-parameters.xhtml".

[6loCHART]
"https://datatracker.ietf.org/wg/6lo/charter".

[G3-PLC] "http://www.itu.int/rec/T-REC-G.9903-201402-I".

[RFC5226] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 5226, DOI 10.17487/RFC5226, May 2008, <http://www.rfc-editor.org/info/rfc5226>.

Authors' Addresses

Samita Chakrabarti
San Jose, CA
USA

Email: samitac.ietf@gmail.com

Gabriel Montenegro
Microsoft
USA

Email: gabriel.montenegro@microsoft.com

Ralph Droms
USA

Email: rdroms.ietf@gmail.com

James Woodyatt
Google
Mountain View, CA
USA

Email: jhw@google.com

6Lo Working Group
Internet-Draft
Intended status: Standards Track
Expires: 7 September 2023

Y. Choi, Ed.
ETRI
Y-G. Hong
Daejon Univ
J-S. Youn
Dongeui Univ
6 March 2023

# Transmission of IPv6 Packets over Near Field Communication
# draft-ietf-6lo-nfc-22

## Abstract

Near Field Communication (NFC) is a set of standards for smartphones and portable devices to establish radio communication with each other by touching them together or bringing them into proximity, usually no more than 10 cm apart. NFC standards cover communications protocols and data exchange formats, and are based on existing radio-frequency identification (RFID) standards including ISO/IEC 14443 and FeliCa. The standards include ISO/IEC 18092 and those defined by the NFC Forum. The NFC technology has been widely implemented and available in mobile phones, laptop computers, and many other devices. This document describes how IPv6 is transmitted over NFC using 6LoWPAN techniques.

## 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 7 September 2023.

## Copyright Notice

Table of Contents



## 1. Introduction

NFC is a set of short-range wireless technologies, typically requiring a distance between sender and receiver of 10 cm or less. NFC operates at 13.56 MHz, and at rates ranging from 106 kbps to 424 kbps, as per the ISO/IEC 18000-3 air interface [ECMA-340]. NFC builds upon RFID systems by allowing two-way communication between endpoints. NFC always involves an initiator and a target; the initiator actively generates an RF field that can power a passive target. This enables NFC targets to take very simple form factors,

such as tags, stickers, key fobs, or cards, while avoiding the need for batteries. NFC peer-to-peer communication is possible, provided that both devices are powered.

NFC has its very short transmission range of 10 cm or less, so the other hidden NFC devices behind outside the range cannot receive NFC signals. Therefore, NFC often regarded as a secure communications technology.

In order to benefit from Internet connectivity, it is desirable for NFC-enabled devices to support IPv6, considering its large address space, along with tools for unattended operation, among other advantages. This document specifies how IPv6 is supported over NFC by using IPv6 over Low-power Wireless Personal Area Network (6LoWPAN) techniques [RFC4944], [RFC6282], [RFC6775]. 6LoWPAN is suitable, considering that it was designed to support IPv6 over IEEE 802.15.4 networks [IEEE802.15.4], and some of the characteristics of the latter are similar to those of NFC.

## 2. Conventions and Terminology

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.

This specification requires readers to be familiar with all the terms and concepts that are discussed in "IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs): Overview, Assumptions, Problem Statement, and Goals" [RFC4919], "Transmission of IPv6 Packets over IEEE 802.15.4 Networks" [RFC4944], "Neighbor Discovery Optimization for IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs) [RFC6775].

6LoWPAN Node (6LN):

A 6LoWPAN node is any host or router participating in a LoWPAN. This term is used when referring to situations in which either a host or router can play the role described.

6LoWPAN Router (6LR):

An intermediate router in the LoWPAN that is able to send and receive Router Advertisements (RAs) and Router Solicitations (RSs) as well as forward and route IPv6 packets. 6LoWPAN routers are present only in route-over topologies.

6LoWPAN Border Router (6LBR):

A border router located at the junction of separate 6LoWPAN networks or between a 6LoWPAN network and another IP network. There may be one or more 6LBRs at the 6LoWPAN network boundary. A 6LBR is the responsible authority for IPv6 prefix propagation for the 6LoWPAN network it is serving. An isolated LoWPAN also contains a 6LBR in the network, which provides the prefix(es) for the isolated network.

## 3. Overview of Near Field Communication Technology

This section presents an overview of NFC, focusing on the characteristics of NFC that are most relevant for supporting IPv6.

NFC enables simple, two-way, interaction between two devices, allowing users to perform contactless transactions, access digital content, and connect electronic devices with a single touch. NFC utilizes key elements in existing standards for contactless card Technology, such as ISO/IEC 14443 A&B and JIS-X 6319-4. NFC allows devices to share information at a distance up to 10 cm with a maximum physical layer bit rate of 424 kbps.

### 3.1. Peer-to-peer Mode of NFC

NFC defines three modes of operation: card emulation, peer-to-peer, and reader/writer. Only the peer-to-peer mode allows two NFC-enabled devices to communicate with each other to exchange information bidirectionally. The other two modes do not support two-way communications between two devices. Therefore, the peer-to-peer mode MUST used for IPv6 over NFC.

### 3.2. Protocol Stack of NFC

NFC defines a protocol stack for the peer-to-peer mode (Figure 1). The peer-to-peer mode is offered by the Activities Digital Protocol at the NFC Physical Layer. The NFC Logical Link Layer comprises the Logical Link Control Protocol (LLCP), and when IPv6 is used over NFC, it also includes an IPv6-LLCP Binding. IPv6 and its underlying adaptation Layer (i.e., IPv6-over-NFC adaptation layer) are placed directly on the top of the IPv6-LLCP Binding. An IPv6 datagram is transmitted by the Logical Link Control Protocol (LLCP) with guaranteed delivery, two-way transmission of information between the peer devices.

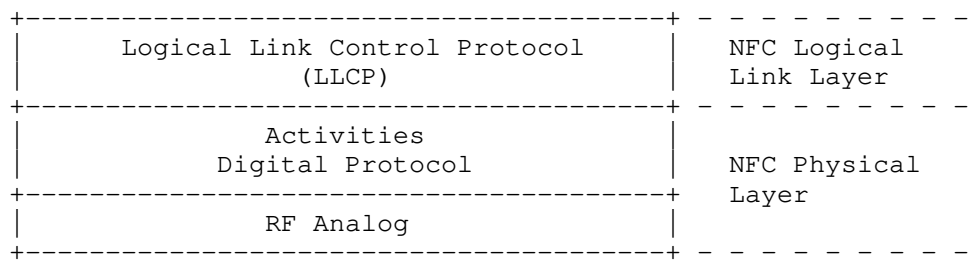



Figure 1: Protocol Stack of NFC

The LLCP consists of Logical Link Control (LLC) and MAC Mapping. The MAC Mapping integrates an existing RF protocol into the LLCP architecture. The LLC contains three components (Link Management, Connection-oriented Transmission, and Connectionless Transmission). The Link Management is responsible for serializing all connection-oriented and connectionless LLC PDU (Protocol Data Unit) exchanges and for aggregation and disaggregation of small PDUs. The Connection-oriented Transmission is responsible for maintaining all connection-oriented data exchanges including connection set-up and termination. However, NFC links do not guarantee perfect wireless link quality, so some type of delays or variation in delay would be expected in any case. The Connectionless Transmission is responsible for handling unacknowledged data exchanges.

In order to send an IPv6 packet over NFC, the packet MUST be passed down to the LLCP layer of NFC and carried by an Information Field in an LLCP Protocol Data Unit (I PDU). The LLCP does not support fragmentation and reassembly. For IPv6 addressing or address configuration, the LLCP MUST provide related information, such as link layer addresses, to its upper layer. The LLCP to IPv6 protocol binding MUST transfer the Source Service Access Point (SSAP) and Destination Service Access Point (DSAP) value to the IPv6 over NFC adaptation layer. SSAP is a Logical Link Control (LLC) address of the source NFC-enabled device with a size of 6 bits, while DSAP means an LLC address of the destination NFC-enabled device. Thus, SSAP is a source address, and DSAP is a destination address.

In addition, NFC links and host do not need to consider IP header bits for QoS signaling, or utilize these meaningfully.

### 3.3. NFC-enabled Device Addressing

According to [LLCP-1.4], NFC-enabled devices have two types of 6-bit addresses (i.e., SSAP and DSAP) to identify service access points. Several service access points can be installed on a NFC device. However, the SSAP and DSAP can be used as identifiers for NFC link connections with the IPv6 over NFC adaptation layer. Therefore, the SSAP can be used to generate an IPv6 interface identifier. Address values between 00h and 0Fh of SSAP and DSAP are reserved for identifying the well-known service access points, which are defined in the NFC Forum Assigned Numbers Register. Address values between 10h and 1Fh are assigned by the local LLC to services registered by local service environment. In addition, address values between 0x2 and 0x3f are assigned by the local LLC as a result of an upper layer service request. Therefore, the address values between 0x2 and 0x3f can be used for generating IPv6 interface identifiers.

### 3.4. MTU of NFC Link Layer

As mentioned in Section 3.2, when an IPv6 packet is transmitted, the packet MUST be passed down to LLCP of NFC and transported to an I PDU of LLCP of the NFC-enabled peer device.

The information field of an I PDU contains a single service data unit. The maximum number of octets in the information field is determined by the Maximum Information Unit (MIU) for the data link connection. The default value of the MIU for I PDUs is 128 octets. The local and remote LLCs each establish and maintain distinct MIU values for each data link connection endpoint. Also, an LLC may announce a larger MIU for a data link connection by transmitting an optional Maximum Information Unit Extension (MIUX) parameter within the information field. If no MIUX parameter is transmitted, the MIU value is 128 bytes. Otherwise, the MTU size in NFC LLCP MUST be calculated from the MIU value as follows:

$$MTU = MIU = 128 + MIUX.$$

According to [LLCP-1.4], Figure 2 shows an example of the MIUX parameter TLV. The Type and Length fields of the MIUX parameter TLV have each a size of 1 byte. The size of the TLV Value field is 2 bytes.

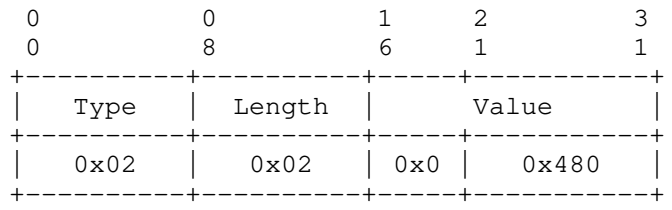



Figure 2: Example of MIUX Parameter TLV

When the MIUX parameter is used, the TLV Type field is 0x02 and the TLV Length field is 0x02. The MIUX parameter is encoded into the least significant 11 bits of the TLV Value field. The unused bits in the TLV Value field is set to zero by the sender and ignored by the receiver. The maximum possible value of the TLV Value field is 0x7FF, and the maximum size of the LLCP MTU is 2175 bytes. As per the present specification [LLCP-1.4], the MIUX value MUST be 0x480 to support the IPv6 MTU requirement (of 1280 bytes) [RFC8200].

## 4. Specification of IPv6 over NFC

NFC technology has requirements owing to low power consumption and allowed protocol overhead. 6LoWPAN standards [RFC4944], [RFC6775], and [RFC6282] provide useful functionality for reducing the overhead of IPv6 over NFC. This functionality consists of link-local IPv6 addresses and stateless IPv6 address auto-configuration (see Section 4.2 and Section 4.3), Neighbor Discovery (see Section 4.4) and header compression (see Section 4.6).

### 4.1. Protocol Stack

Figure 3 illustrates the IPv6 over NFC protocol stack. Upper layer protocols can be transport layer protocols (e.g., TCP and UDP), application layer protocols, and others capable of running on top of IPv6.

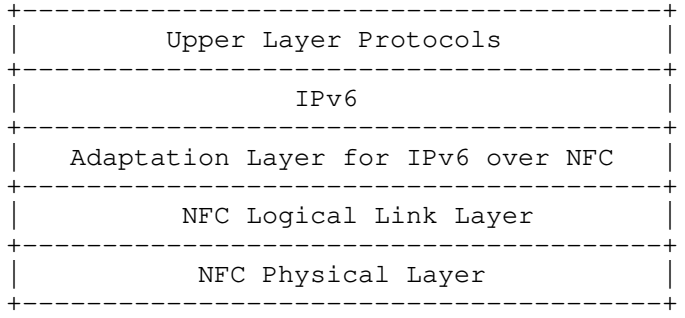


Figure 3: Protocol Stack for IPv6 over NFC

The adaptation layer for IPv6 over NFC supports neighbor discovery, stateless address auto-configuration, header compression, and fragmentation & reassembly, based on 6LoWPAN. Note that 6LoWPAN Header compression [RFC6282] does not define header compression for TCP. The latter can still be supported over IPv6 over NFC, albeit without the performance optimization of header compression.

## 4.2. Stateless Address Autoconfiguration

An NFC-enabled device performs stateless address autoconfiguration as per [RFC4862]. A 64-bit Interface identifier (IID) for an NFC interface is formed by utilizing the 6-bit NFC SSAP (see Section 3.3). In the viewpoint of address configuration, such an IID should guarantee a stable IPv6 address during the course of a single connection, because each data link connection is uniquely identified by the pair of DSAP and SSAP included in the header of each LLC PDU in NFC.

Following the guidance of [RFC7136], interface identifiers of all unicast addresses for NFC-enabled devices are 64 bits long and constructed by using the generation algorithm of random (but stable) identifier (RID) [RFC7217].

The RID is an output which is created by the F() algorithm with input parameters. One of the parameters is Net_Iface, and NFC Link Layer address (i.e., SSAP) MUST be a source of the Net_Iface parameter. The 6-bit address of SSAP of NFC is short and easy to be targeted by attacks of third party (e.g., address scanning). The F() algorithm with SHA-256 can provide secured and stable IIDs for NFC-enabled devices. In addition, an optional parameter, Network_ID is used to increase the randomness of the generated IID with NFC link layer address (i.e., SSAP). The secret key SHOULD be of at least 128 bits. It MUST be initialized to a pseudo-random number [RFC4086].

## 4.3. IPv6 Link-Local Address

The IPv6 link-local address for an NFC-enabled device is formed by appending the IID to the prefix fe80::/64, as depicted in Figure 4.

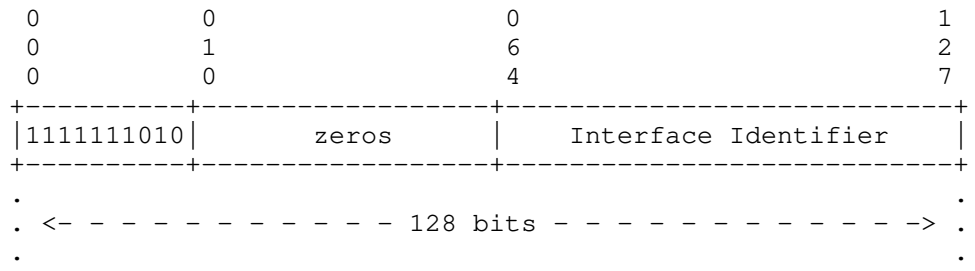



Figure 4: IPv6 link-local address in NFC

The "Interface Identifier" can be a random and stable IID.

## 4.4. Neighbor Discovery

Neighbor Discovery Optimization for 6LoWPANs ([RFC6775]) describes the neighbor discovery approach in several 6LoWPAN topologies, such as mesh topology. NFC supports mesh topologies, but most of all applications would use a simple multi-hop network topology or directly connected peer-to-peer network because NFC RF range is very short.

* When an NFC 6LoWPAN Node (6LN) is directly connected to an 6LBR, the 6LN MUST register its address with the 6LBR by sending Neighbor Solicitation (NS) with the Extended Address Registration Option (EARO) [RFC8505], and Neighbor Advertisement (NA) is started. When the 6LN and 6LBR are linked each other, an address is assigned to the 6LN. In this process, Duplicate Address Detection (DAD) is not required.

* When two or more NFC LNs are connected to the 6LBR, two cases of topologies can be formed. One is a multi-hop topology, and the other is a star topology based on the 6LBR. In multi-hop topology, LNs which have two or more links with neighbor nodes may act as routers. In star topology, any of LNs can be a router.

* For receiving Router Solicitations and sending Router Advertisements, the NFC 6LNs MUST follow Sections 5.3 and 5.4 of [RFC6775].

* When a NFC device is a 6LoWPAN Router (6LR) or a 6LBR, the NFC device MUST follow Section 6 and 7 of [RFC6775].

### 4.5. Dispatch Header

All IPv6-over-NFC encapsulated datagrams are prefixed by an encapsulation header stack consisting of a Dispatch value [IANA-6LoWPAN]. The only sequence currently defined for IPv6-over-NFC MUST be the LOWPAN_IPHC compressed IPv6 header (see Section 4.6) header followed by payload, as depicted in Figure 5 and Figure 6.

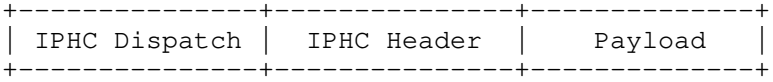



Figure 5: A IPv6-over-NFC Encapsulated LOWPAN_IPHC Compressed IPv6 Datagram

The dispatch value (length: 1 octet) is treated as an unstructured namespace. Only a single pattern is used to represent current IPv6-over-NFC functionality.

| Pattern | Header Type | Reference |
|---|---|---|
| 01 1xxxxx | LOWPAN_IPHC | [RFC6282] |

Figure 6: Dispatch Values

Other IANA-assigned 6LoWPAN Dispatch values do not apply to this specification.

### 4.6. Header Compression

Header compression as defined in [RFC6282], which specifies the compression format for IPv6 datagrams on top of IEEE 802.15.4, is REQUIRED in this document as the basis for IPv6 header compression on top of NFC. All headers MUST be compressed according to RFC 6282 encoding formats.

Therefore, IPv6 header compression in [RFC6282] MUST be implemented. Further, implementations MUST also support Generic Header Compression (GHC) of [RFC7400].

If a 16-bit address is required as a short address, it MUST be formed by padding the 6-bit NFC SSAP (NFC link-layer node address) to the left with zeros as shown in Figure 7.

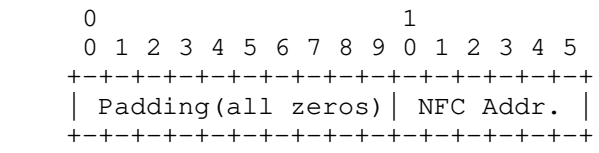



Figure 7: NFC short address format

## 4.7. Fragmentation and Reassembly Considerations

IPv6-over-NFC MUST NOT use fragmentation and reassembly (FAR) at the adaptation layer for the payloads as discussed in Section 3.4. The NFC link connection for IPv6 over NFC MUST be configured with an equivalent MIU size to support the IPv6 MTU requirement (of 1280 bytes). To this end, the MIUX value is 0x480.

## 4.8. Unicast and Multicast Address Mapping

The address resolution procedure for mapping IPv6 non-multicast addresses into NFC link-layer addresses follows the general description in Section 4.6.1 and 7.2 of [RFC4861], unless otherwise specified.

The Source/Target link-layer Address option has the following form when the addresses are 6-bit NFC SSAP/DSAP (NFC link-layer node addresses).

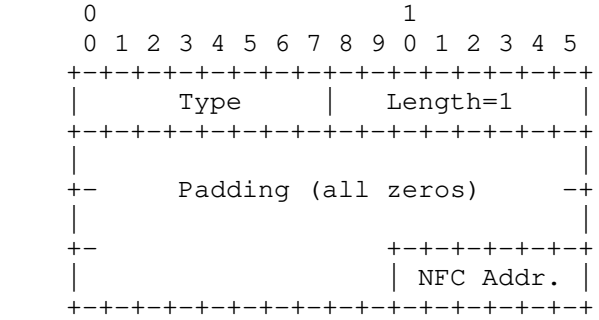



Figure 8: Unicast address mapping

Option fields:

Type:

- 1: for Source Link-layer address.

- 2: for Target Link-layer address.

Length:

- This is the length of this option (including the type and length fields) in units of 8 bits. The value of this field is 1 for 6-bit NFC node addresses.

NFC address:

- The 6-bit address in canonical bit order. This is the unicast address the interface currently responds to.

The NFC Link Layer does not support multicast. Therefore, packets are always transmitted by unicast between two NFC-enabled devices. Even in the case where a 6LBR is attached to multiple 6LNs, the 6LBR cannot do a multicast to all the connected 6LNs. If the 6LBR needs to send a multicast packet to all its 6LNs, it has to replicate the packet and unicast it on each link. However, this is not energy-efficient, and the central node, which is battery-powered, must take particular care of power consumption. To further conserve power, the 6LBR MUST keep track of multicast listeners at NFC link-level granularity (not at subnet granularity), and it MUST NOT forward multicast packets to 6LNs that have not registered as listeners for multicast groups the packets belong to. In the opposite direction, a 6LN always has to send packets to or through the 6LBR. Hence, when a 6LN needs to transmit an IPv6 multicast packet, the 6LN will unicast the corresponding NFC packet to the 6LBR.

## 5. Internet Connectivity Scenarios

### 5.1. NFC-enabled Device Network Connected to the Internet

Figure 9 illustrates an example of an NFC-enabled device network connected to the Internet. The distance between 6LN and 6LBR is typically 10 cm or less. For example, a laptop computer that is connected to the Internet (e.g. via Wi-Fi, Ethernet, etc.) may also support NFC and act as a 6LBR. Another NFC-enabled device may run as a 6LN and communicate with the 6LBR, as long as both are within each other's range.

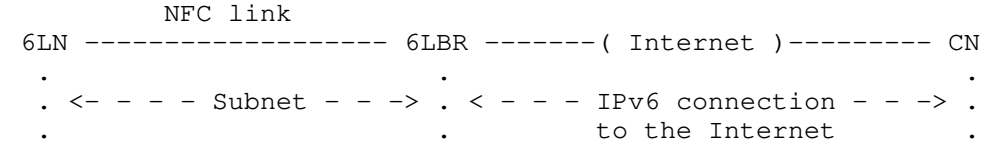


Figure 9: NFC-enabled device network connected to the Internet

Two or more 6LNs may be connected with a 6LBR, but each connection uses different IPv6 prefix. The 6LBR is acting as a router and forwarding packets between 6LNs and the Internet. Also, the 6LBR MUST ensure address collisions do not occur because the 6LNs are connected to the 6LBR like a start topology, so the 6LBR checks whether IPv6 addresses are duplicate or not, since 6LNs need to register their addresses with the 6LBR.

## 5.2. Isolated NFC-enabled Device Network

In some scenarios, the NFC-enabled device network may permanently be a simple isolated network as shown in the Figure 10.

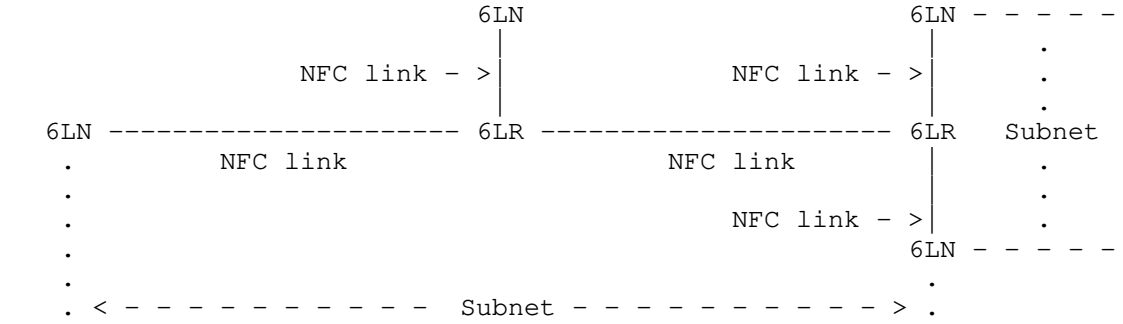



Figure 10: Isolated NFC-enabled device network

In multihop (i.e., more complex) topologies, the 6LR can also do the same task, but then Duplicate Address Detection (DAD) requires the extensions for multihop networks such as the ones in [RFC6775].

# 6. IANA Considerations

There are no IANA considerations related to this document.

# 7. Security Considerations

Neighbor Discovery in unencrypted wireless device networks may be susceptible to various threats as described in [RFC3756].

Per the NFC Logical Link Control Protocol [LLCP-1.4]:

* LLCP of NFC provides protection of user data to ensure confidentiality of communications. The confidentiality mechanism involves the encryption of user service data with a secret key that has been established during link activation.

* LLCP of NFC has two modes (i.e., ad-hoc mode and authenticated mode) for secure data transfer. Ad-hoc secure data transfer can be established between two communication parties without any prior knowledge of the communication partner. Ad-hoc secure data transfer can be vulnerable to Man-In-The-Middle (MITM) attacks. Authenticated secure data transfer provides protection against Man-In-The-Middle (MITM) attacks. In the initial bonding step, the two communicating parties store a shared secret along with a Bonding Identifier.

* For all subsequent interactions, the communicating parties re-use the shared secret and compute only the unique encryption key for that session. Secure data transfer is based on the cryptographic algorithms defined in the NFC Authentication Protocol [NAP-1.0].

Furthermore, NFC is considered by many to offer intrinsic security properties due to its short link range. When interface identifiers (IIDs) are generated, devices and users are required to consider mitigating various threats, such as correlation of activities over time, location tracking, device-specific vulnerability exploitation, and address scanning. However, IPv6-over-NFC uses a random (but stable) identifier (RID) [RFC7217] as an IPv6 interface identifier, and NFC applications use short-lived connections, and a different address is used for each connection, where the latter is of extremely short duration.

## 8. Acknowledgements

We are grateful to the members of the IETF 6lo working group.

Michael Richardson, Suresh Krishnan, Pascal Thubert, Carsten Bormann, Alexandru Petrescu, James Woodyatt, Dave Thaler, Samita Chakrabarti, Gabriel Montenegro, Erik Kline and Carles Gomez Montenegro have provided valuable feedback for this document.

## 9. References

### 9.1. Normative References

[LLCP-1.4] NFC Forum, "NFC Logical Link Control Protocol, Version 1.4", NFC Forum Technical Specification , January 2021, <https://nfc-forum.org/build/specifications>.

[NAP-1.0] NFC Forum, "NFC Authentication Protocol Candidate Technical Specification, Version 1.0", NFC Forum Technical Specification , December 2020, <https://nfc-forum.org/build/specifications>.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

[RFC4086] Eastlake 3rd, D., Schiller, J., and S. Crocker, "Randomness Requirements for Security", BCP 106, RFC 4086, DOI 10.17487/RFC4086, June 2005, <https://www.rfc-editor.org/info/rfc4086>.

[RFC4861] Narten, T., Nordmark, E., Simpson, W., and H. Soliman, "Neighbor Discovery for IP version 6 (IPv6)", RFC 4861, DOI 10.17487/RFC4861, September 2007, <https://www.rfc-editor.org/info/rfc4861>.

[RFC4862] Thomson, S., Narten, T., and T. Jinmei, "IPv6 Stateless Address Autoconfiguration", RFC 4862, DOI 10.17487/RFC4862, September 2007, <https://www.rfc-editor.org/info/rfc4862>.

[RFC4919] Kushalnagar, N., Montenegro, G., and C. Schumacher, "IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs): Overview, Assumptions, Problem Statement, and Goals", RFC 4919, DOI 10.17487/RFC4919, August 2007, <https://www.rfc-editor.org/info/rfc4919>.

[RFC4944] Montenegro, G., Kushalnagar, N., Hui, J., and D. Culler, "Transmission of IPv6 Packets over IEEE 802.15.4 Networks", RFC 4944, DOI 10.17487/RFC4944, September 2007, <https://www.rfc-editor.org/info/rfc4944>.

[RFC6282] Hui, J., Ed. and P. Thubert, "Compression Format for IPv6 Datagrams over IEEE 802.15.4-Based Networks", RFC 6282, DOI 10.17487/RFC6282, September 2011, <https://www.rfc-editor.org/info/rfc6282>.

[RFC6775] Shelby, Z., Ed., Chakrabarti, S., Nordmark, E., and C. Bormann, "Neighbor Discovery Optimization for IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs)", RFC 6775, DOI 10.17487/RFC6775, November 2012, <https://www.rfc-editor.org/info/rfc6775>.

[RFC7136] Carpenter, B. and S. Jiang, "Significance of IPv6 Interface Identifiers", RFC 7136, DOI 10.17487/RFC7136, February 2014, <https://www.rfc-editor.org/info/rfc7136>.

[RFC7217] Gont, F., "A Method for Generating Semantically Opaque Interface Identifiers with IPv6 Stateless Address Autoconfiguration (SLAAC)", RFC 7217, DOI 10.17487/RFC7217, April 2014, <https://www.rfc-editor.org/info/rfc7217>.

[RFC7400] Bormann, C., "6LoWPAN-GHC: Generic Header Compression for IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs)", RFC 7400, DOI 10.17487/RFC7400, November 2014, <https://www.rfc-editor.org/info/rfc7400>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

[RFC8200] Deering, S., Hinden, R., and RFC Publisher, "Internet Protocol, Version 6 (IPv6) Specification", STD 86, RFC 8200, DOI 10.17487/RFC8200, July 2017, <https://www.rfc-editor.org/info/rfc8200>.

[RFC8505] Thubert, P., Ed., Nordmark, E., Chakrabarti, S., and C. Perkins, "Registration Extensions for IPv6 over Low-Power Wireless Personal Area Network (6LoWPAN) Neighbor Discovery", RFC 8505, DOI 10.17487/RFC8505, November 2018, <https://www.rfc-editor.org/info/rfc8505>.

## 9.2. Informative References

[ECMA-340] "Near Field Communication - Interface and Protocol (NFCIP-1) 3rd Ed.", ECMA International , June 2013, <https://www.ecma-international.org/wp-content/uploads/ECMA-340_3rd_edition_june_2013.pdf>.

[IANA-6LoWPAN] Internet Assigned Numbers Authority (IANA), "IPv6 Low Power Personal Area Network Parameters", 3 December 2021, <https://www.iana.org/assignments/_6lowpan-parameters>.

[IEEE802.15.4] IEEE Computer Society, "IEEE Standard for Low-Rate Wireless Networks, IEEE Std. 802.15.4-2020", IEEE , July 2020, <https://standards.ieee.org/ieee/802.15.4/7029/>.

[RFC3756] Nikander, P., Ed., Kempf, J., and E. Nordmark, "IPv6 Neighbor Discovery (ND) Trust Models and Threats", RFC 3756, DOI 10.17487/RFC3756, May 2004, <https://www.rfc-editor.org/info/rfc3756>.

## Authors' Addresses

Younghwan Choi (editor)
Electronics and Telecommunications Research Institute
218 Gajeongno, Yuseung-gu
Daejeon
34129
South Korea
Phone: +82 42 860 1429
Email: yhc@etri.re.kr

Yong-Geun Hong
Daejon University
62 Daehak-ro, Dong-gu
Daejeon
34520
South Korea
Phone: +82 42 280 4841
Email: yonggeun.hong@gmail.com

Joo-Sang Youn
DONG-EUI University
176 Eomgwangno Busan_jin_gu
Busan
614-714
South Korea
Phone: +82 51 890 1993
Email: joosang.youn@gmail.com

6lo
Internet-Draft
Updates: 4944 (if approved)
Intended status: Standards Track
Expires: April 15, 2017

P. Thubert, Ed.
Cisco
R. Cragie
ARM

October 12, 2016

# 6LoWPAN Paging Dispatch
# draft-ietf-6lo-paging-dispatch-05

## Abstract

This specification updates RFC 4944 to introduce a new context switch mechanism for 6LoWPAN compression, expressed in terms of Pages and signaled by a new Paging Dispatch.

## 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 http://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 April 15, 2017.

## Copyright Notice

Table of Contents



## 1. Introduction

The design of Low Power and Lossy Networks (LLNs) is generally focused on saving energy, which often is a very constrained resource. Other constraints, such as memory capacity and duty cycle restrictions on LLN devices, usually derive from that primary concern. Energy is often available only from primary batteries that are expected to last for years, or is scavenged from the environment in very limited amounts. Any protocol that is intended for use in LLNs must be designed with a primary focus on saving energy, which is a strict requirement.

Controlling the amount of data transmission is one possible means of saving energy. In a number of LLN standards, the frame size is limited to much smaller values than the IPv6 maximum transmission unit (MTU) of 1280 bytes. In particular, an LLN that relies on the classical Physical Layer (PHY) of IEEE 802.15.4 [IEEE802154] is limited to 127 bytes per frame. The need to compress IPv6 packets over IEEE 802.15.4 led to the 6LoWPAN Header Compression [RFC6282] work (6LoWPAN-HC).

As more and more protocols need to be compressed, the encoding capabilities of the original dispatch defined in the 6lo adaptation layer framework ([RFC4944],[RFC6282]) becomes saturated. This specification introduces a new context switch mechanism for 6LoWPAN compression, expressed in terms of Pages and signaled by a new Paging Dispatch mechanism.

## 2. Terminology

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 [RFC2119].

The Terminology used in this document is consistent with and incorporates that described in Terms Used in Routing for Low-Power and Lossy Networks [RFC7102] and Terminology for Constrained-Node Networks [RFC7228].

## 3. Updating RFC 4944

This draft adapts 6LoWPAN while maintaining backward compatibility with IPv6 over IEEE 802.15.4 [RFC4944] by introducing a concept of a "parsing context" in the 6LoWPAN parser, a context being identified by a Page Number. This specification defines 16 Pages.

Pages are delimited in a 6LoWPAN packet by a Paging Dispatch value that indicates the next current Page. The Page Number is encoded in a Paging Dispatch with the Value Bit Pattern of 1111xxxx where xxxx is the Page Number, 0 to 15, as described in Figure 1:


Figure 1: Paging Dispatch with Page Number Encoding.

Values of the Dispatch byte defined in [RFC4944] are considered as belonging to the Page 0 parsing context, which is the default and does not need to be signaled explicitly at the beginning of a 6LoWPAN packet. This ensures backward compatibility with existing implementations of 6LoWPAN.

The Dispatch bits defined in Page 0 by [RFC4944] are free to be reused in Pages 1 to 15. This specification allocates some values in Page 1 in Section 4 and leaves the rest open for future allocations.

Values opened by this specification in Page 1 to 14 are to be assigned for new protocols whereas Page 15 is reserved for experimentations.

Note: This specification does not use the Escape Dispatch, which extends Page 0 to more values, but rather allocates another Dispatch

Bit Pattern (1111xxxx) for a new Paging Dispatch, that is present in all Pages, including Page 0 and Pages defined in future specifications, to indicate the next parsing context represented by its Page Number. The rationale for avoiding that approach is that there can be multiple occurrences of a new header indexed by this specification in a single frame and the overhead on an octet each time for the Escape Dispatch would be prohibitive.

A Page (say Page N) is said to be active once the Page N Paging Dispatch is parsed, and remains active until another Paging Dispatch is parsed.

## 4. Page 1 Paging Dispatch

This specification defines some special properties for Page 1, detailed below:

The Dispatch bits defined for LOWPAN_IPHC by the Compression Format for IPv6 Datagrams over IEEE 802.15.4-Based Networks [RFC6282] are defined with the same values in Page 1 so there is no need to switch context from Page 1 to Page 0 to decode a packet that is encoded per [RFC6282].

Mesh Headers represent Layer-2 information and are processed before any Layer-3 information that is encoded in Page 1. If a 6LoWPAN packet requires a Mesh header, the Mesh Header MUST always be placed in the packet before the first Page 1 Paging Dispatch, if any.

For the same reason, Fragment Headers as defined in [RFC4944] MUST always be placed in the packet before the first Page 1 Paging Dispatch, if any.

The NALP Dispatch Bit Pattern as defined in [RFC4944] is only defined for the first octet in the packet. Switching back to Page 0 for NALP inside a 6LoWPAN packet does not make sense.

As a result, there is no need for restoring the Page 0 parsing context after a context was switched to Page 1, so the value for the Page 0 Paging Dispatch of 11110000 may not actually occur in those packets that adhere to 6LoWPAN specifications available at the time of writing this specification.

## 5. Security Considerations

The security considerations of [RFC4944] and [RFC6282] apply.

# 6. IANA Considerations

## 6.1. Consuming Dispatch Types

This document allocates 16 values from the Dispatch type field registry that was created for [RFC4944]. The allocated values are from 11 110000 through 11 111111 and represent Page Numbers 0 through 15 as discussed in this document.

## 6.2. New Column in Dispatch Type Registry

This document extends the Dispatch type field registry that was created for [RFC4944] and updated by the [RFC6282], by adding a new column called "Page".

This document defines 16 Pages, "Page 0" to "Page 15".

The content of the incumbent registry is assigned to "Page 0".

This document also places in the registry associated to Page 1 the Dispatch type field values that are allocated for LOWPAN_IPHC by [RFC6282]. These values range from 01 100000 through 01 111111 and have the same definition in Page 1 as they do in Page 0; as a result, the registry entries for Page 0 and Page 1 are an exact overlap in this range.

Values ranging from 00000000 to 11101111 in Page 15 (that is all of Page 15 but the space used for Page switch) is reserved for experimentations and shall not be assigned.

The resulting registry may be represented as a table as follow (partial):

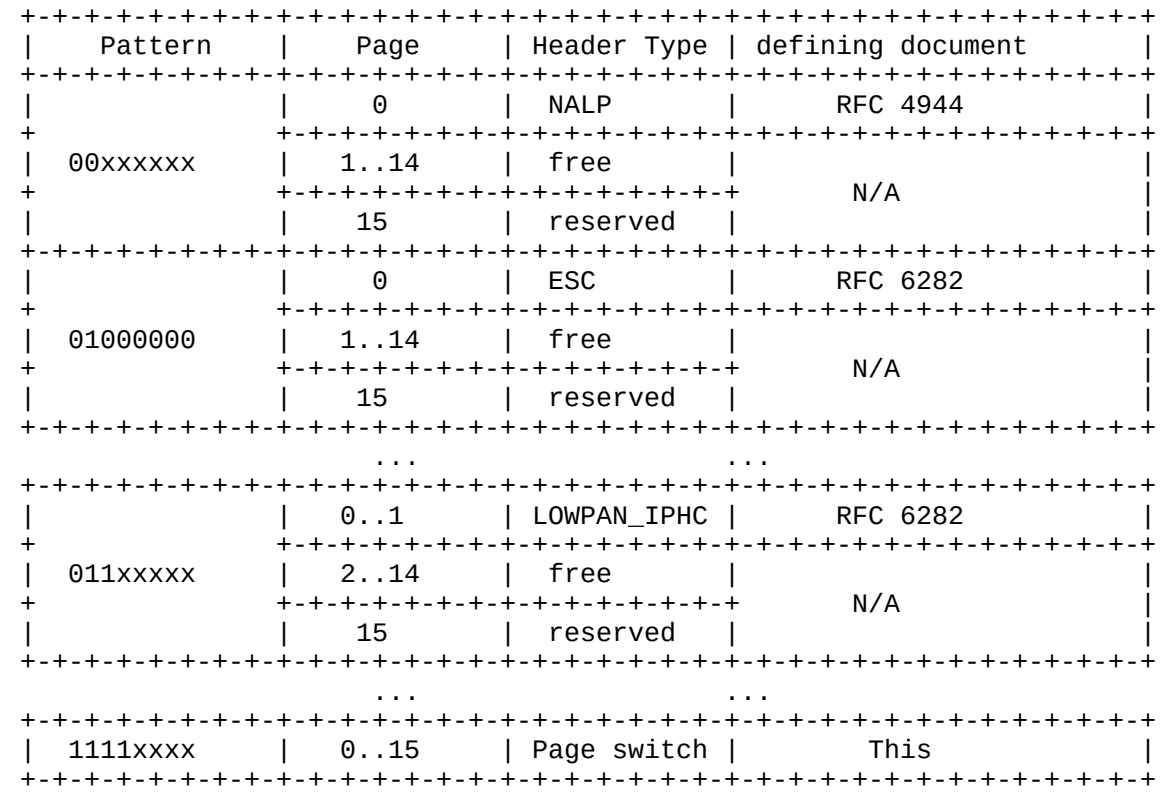



Figure 2: Integrating the new Page column

Future assignments in these registries are to be coordinated via IANA under the policy of "Specification Required" [RFC5226]. It is expected that this policy will allow for other (non-IETF) organizations to more easily obtain assignments.

## 7. Acknowledgments

The authors wish to thank Tom Phinney, Thomas Watteyne, Tengfei Chang, Martin Turon, James Woodyatt, Samita Chakrabarti, Jonathan Hui, Gabriel Montenegro and Ralph Droms for constructive reviews to the design in the 6lo Working Group.

## 8. References

## 8.1. Normative References

[IEEE802154]
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".

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <http://www.rfc-editor.org/info/rfc2119>.

[RFC4944] Montenegro, G., Kushalnagar, N., Hui, J., and D. Culler, "Transmission of IPv6 Packets over IEEE 802.15.4 Networks", RFC 4944, DOI 10.17487/RFC4944, September 2007, <http://www.rfc-editor.org/info/rfc4944>.

[RFC5226] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 5226, DOI 10.17487/RFC5226, May 2008, <http://www.rfc-editor.org/info/rfc5226>.

[RFC6282] Hui, J., Ed. and P. Thubert, "Compression Format for IPv6 Datagrams over IEEE 802.15.4-Based Networks", RFC 6282, DOI 10.17487/RFC6282, September 2011, <http://www.rfc-editor.org/info/rfc6282>.

## 8.2. Informative References

[RFC7102] Vasseur, JP., "Terms Used in Routing for Low-Power and Lossy Networks", RFC 7102, DOI 10.17487/RFC7102, January 2014, <http://www.rfc-editor.org/info/rfc7102>.

[RFC7228] Bormann, C., Ersue, M., and A. Keranen, "Terminology for Constrained-Node Networks", RFC 7228, DOI 10.17487/RFC7228, May 2014, <http://www.rfc-editor.org/info/rfc7228>.

## Authors' Addresses

Pascal Thubert (editor)
Cisco Systems
Building D - Regus
45 Allee des Ormes
BP1200
MOUGINS - Sophia Antipolis 06254
FRANCE

Phone: +33 4 97 23 26 34
Email: pthubert@cisco.com

Robert Cragie
ARM Ltd.
110 Fulbourn Road
Cambridge CB1 9NJ
UK

Email: robert.cragie@gridmerge.com

Network Working Group
Internet-Draft
Intended status: Informational
Expires: May 4, 2017

D. Thaler
Microsoft

October 31, 2016

# Privacy Considerations for IPv6 Adaptation Layer Mechanisms
# draft-ietf-6lo-privacy-considerations-04

## Abstract

This document discusses how a number of privacy threats apply to technologies designed for IPv6 over various link layer protocols, and provides advice to protocol designers on how to address such threats in adaptation layer specifications for IPv6 over such links.

## 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 http://datatracker.ietf.org/drafts/current/.

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

This Internet-Draft will expire on May 4, 2017.

## Copyright Notice

Table of Contents



## 1. Introduction

RFC 6973 [RFC6973] discusses privacy considerations for Internet protocols, and Section 5.2 of that document covers a number of privacy-specific threats. In the context of IPv6 addresses, Section 3 of [RFC7721] provides further elaboration on the applicability of the privacy threats.

When interface identifiers (IIDs) are generated without sufficient entropy compared to the link lifetime, devices and users can become vulnerable to the various threats discussed there, including:

- Correlation of activities over time, if the same identifier is used for traffic over period of time
- Location tracking, if the same interface identifier is used with different prefixes as a device moves between different networks
- Device-specific vulnerability exploitation, if the identifier helps identify a vendor or version or protocol and hence suggests what types of attacks to try
- Address scanning, which enables all of the above attacks by off-link attackers. (On some Non-Broadcast Multi-Access (NBMA) links where all nodes aren't already privy to all on-link addresses, address scans might also be done by on-link attackers, but in most cases address scans are not an interesting threat from on-link attackers and thus address scans generally apply only to routable addresses.)

For example, for links that may last for years, "enough" bits of entropy means at least 46 or so bits (see Section 2 for why) in a routable address; ideally all 64 bits of the IID should be used, although historically some bits have been excluded for reasons discussed in [RFC7421]. Link-local addresses can also be susceptible

to the same privacy threats from off-link attackers, since experience shows they are often leaked by upper-layer protocols such as SMTP, SIP, or DNS.

For these reasons, [I-D.ietf-6man-default-iids] recommends using an address generation scheme in [RFC7217], rather than addresses generated from a fixed link-layer address.

Furthermore, to mitigate the threat of correlation of activities over time on long-lived links, [RFC4941] specifies the notion of a "temporary" address to be used for transport sessions (typically locally-initiated outbound traffic to the Internet) that should not be linkable to a more permanent identifier such as a DNS name, user name, or fixed link-layer address. Indeed, the default address selection rules [RFC6724] now prefer temporary addresses by default for outgoing connections. If a device needs to simultaneously support unlinkable traffic as well as traffic that is linkable to such a stable identifier, this necessitates supporting simultaneous use of multiple addresses per device.

## 2. Amount of Entropy Needed in Global Addresses

In terms of privacy threats discussed in [RFC7721], the one with the need for the most entropy is address scans of routable addresses. To mitigate address scans, one needs enough entropy to make the probability of a successful address probe be negligible. Typically this is measured in the length of time it would take to have a 50% probability of getting at least one hit. Address scans often rely on sending a packet such as a TCP SYN or ICMP Echo Request, and determining whether the reply is an ICMP unreachable error (if no host exists with that address) or a TCP response or ICMP Echo Reply (if a host exists), or neither in which case nothing is known for certain.

Many privacy-sensitive devices support a "stealth mode" as discussed in Section 5 of [RFC7288] or are behind a network firewall that will drop unsolicited inbound traffic (e.g., TCP SYNs, ICMP Echo Requests, etc.) and thus no TCP RST or ICMP Echo Reply will be sent. In such cases, and when the device does not listen on a well-known TCP or UDP port known to the scanner, the effectiveness of an address scan is limited by the ability to get ICMP unreachable errors, since the attacker can only infer the presence of a host based on the absense of an ICMP unreachable error.

Generation of ICMP unreachable errors is typically rate limited to 2 per second (the default in routers such as Cisco routers running IOS 12.0 or later). Such a rate results in taking about a year to completely scan 26 bits of space.

The actual math is as follows. Let 2^N be the number of devices on the subnet. Let 2^M be the size of the space to scan (i.e., M bits of entropy). Let S be the number of scan attempts. The formula for a 50% chance of getting at least one hit in S attempts is: P(at least one success) = 1 - (1 - 2^N/2^M)^S = 1/2. Assuming 2^M >> S, this simplifies to: S * 2^N/2^M = 1/2, giving S = 2^(M-N-1), or M = N + 1 + log_2(S). Using a scan rate of 2 per second, this results in the following rule of thumb:

> Bits of entropy needed = log_2(# devices per link) + log_2(seconds of link lifetime) + 2

For example, for a network with at most 2^16 devices on the same long-lived link, and the average lifetime of a link being 8 years (2^28 seconds) or less, this results in a need for at least 46 bits of entropy (16+28+2) so that an address scan would need to be sustained for longer than the lifetime of the link to have a 50% chance of getting a hit.

Although 46 bits of entropy may be enough to provide privacy in such cases, 59 or more bits of entropy would be needed if addresses are used to provide security against attacks such as spoofing, as CGAs [RFC3972] and HBAs [RFC5535] do, since attacks are not limited by ICMP rate limiting but by the processing power of the attacker. See those RFCs for more discussion.

If, on the other hand, the devices being scanned for respond to unsolicited inbound packets, then the address scan is not limited by the ICMP unreachable rate limit in routers, since an adversary can determine the presence of a host without them. In such cases, more bits of entropy would be needed to provide the same level of protection.

## 3. Potential Approaches

The table below shows the number of bits of entropy currently available in various technologies:

| Technology | Reference | Bits of Entropy |
|---|---|---|
| 802.15.4 | [RFC4944] | 16+ or any EUI-64 |
| Bluetooth LE | [RFC7668] | 48 |
| DECT ULE | [I-D.ietf-6lo-dect-ule] | 40 or any EUI-48 |
| MS/TP | [I-D.ietf-6lo-6lobac] | 7 or 64 |
| ITU-T G.9959 | [RFC7428] | 8 |
| NFC | [I-D.ietf-6lo-nfc] | 5 |

Such technologies generally support either IEEE identifiers or so called "Short Addresses", or both, as link layer addresses. We discuss each in turn.

### 3.1. IEEE-Identifier-Based Addresses

Some technologies allow the use of IEEE EUI-48 or EUI-64 identifiers, or allow using an arbitrary 64-bit identifier. Using such an identifier to construct IPv6 addresses makes it easy to use the normal LOWPAN_IPHC encoding with stateless compression, allowing such IPv6 addresses to be fully elided in common cases.

Global addresses with interface identifiers formed from IEEE identifiers can have insufficient entropy to mitigate address scans unless the IEEE identifier itself has sufficient entropy, and enough bits of entropy are carried over into the IPv6 address to sufficiently mitigate the threats. Privacy threats other than "Correlation over time" can be mitigated using per-network randomized link-layer addresses with enough entropy compared to the link lifetime. A number of such proposals can be found at <https://mentor.ieee.org/privecsg/documents>, and Section 10.8 of [BTCorev4.1] specifies one for Bluetooth. Using routable IPv6 addresses derived from such link-layer addresses would be roughly equivalent to those specified in [RFC7217].

Correlation over time (for all addresses, not just routable addresses) can be mitigated if the link-layer address itself changes often enough, such as each time the link is established, if the link lifetime is short. For further discussion, see [I-D.huitema-6man-random-addresses].

Another potential concern is that of efficiency, such as avoiding Duplicate Address Detection (DAD) all together when IPv6 addresses are IEEE-identifier-based. Appendix A of [RFC4429] provides an analysis of address collision probability based on the number of bits of entropy. A simple web search on "duplicate MAC addresses" will show that collisions do happen with MAC addresses, and thus based on the analysis in [RFC4429], using sufficient bits of entropy in random addresses can provide greater protection against collision than using MAC addresses.

### 3.2. Short Addresses

A routable IPv6 address with an interface identifier formed from the combination of a "Short Address" and a set of well-known constant bits (such as padding with 0’s) lacks sufficient entropy to mitigate address scanning unless the link lifetime is extremely short. Furthermore, an adversary could also use statistical methods to

determine the size of the L2 address space and thereby make some inference regarding the underlying technology on a given link, and target further attacks accordingly.

When Short Addresses are desired on links that are not guaranteed to have a short enough lifetime, the mechanism for constructing an IPv6 interface identifier from a Short Address could be designed to sufficiently mitigate the problem. For example, if all nodes on a given L2 network have a shared secret (such as the key needed to get on the layer-2 network), the 64-bit IID might be generated using a one-way hash that includes (at least) the shared secret together with the Short Address. The use of such a hash would result in the IIDs being spread out among the full range of IID address space, thus mitigating address scans, while still allowing full stateless compression/elision.

For long-lived links, "temporary" addresses might even be generated in the same way by (for example) also including in the hash the Version Number from the Authoritative Border Router Option (Section 4.3 of [RFC6775]), if any. This would allow changing temporary addresses whenever the Version Number is changed, even if the set of prefix or context information is unchanged.

In summary, any specification using Short Addresses should carefully construct an IID generation mechanism so as to provide sufficient entropy compared to the link lifetime.

## 4. Recommendations

The following are recommended for adaptation layer specifications:

- Security (privacy) sections should say how address scans are mitigated. An address scan might be mitigated by having a link always be short-lived, or might be mitigated by having a large number of bits of entropy in routable addresses, or some combination. Thus, a specification should explain what the maximum lifetime of a link is in practice, and show how the number of bits of entropy is sufficient given that lifetime.

- Technologies should define a way to include sufficient bits of entropy in the IPv6 interface identifier, based on the maximum link lifetime. Specifying that randomized link-layer addresses can be used is one easy way to do so, for technologies that support such identifiers.

- Specifications should not simply construct an IPv6 interface identifier by padding a short address with a set of other well-known constant bits, unless the link lifetime is guaranteed to be

extremely short or the short address is allocated by the network (rather than being constant in the node). This also applies to link-local addresses if the same short address is used independent of network and is unique enough to allow location tracking.

- Specifications should make sure that an IPv6 address can change over long periods of time. For example, the interface identifier might change each time a device connects to the network (if connections are short), or might change each day (if connections can be long). This is necessary to mitigate correlation over time.

- If a device can roam between networks, and more than a few bits of entropy exist in the IPv6 interface identifier, then make sure that the interface identifier can vary per network as the device roams. This is necessary to mitigate location tracking.

## 5. IANA Considerations

This document has no actions for IANA.

## 6. Security Considerations

This entire document is about security considerations and how to specify possible mitigations.

## 7. Informative References

[RFC3972] Aura, T., "Cryptographically Generated Addresses (CGA)", RFC 3972, DOI 10.17487/RFC3972, March 2005, <http://www.rfc-editor.org/info/rfc3972>.

[RFC4429] Moore, N., "Optimistic Duplicate Address Detection (DAD) for IPv6", RFC 4429, DOI 10.17487/RFC4429, April 2006, <http://www.rfc-editor.org/info/rfc4429>.

[RFC4941] Narten, T., Draves, R., and S. Krishnan, "Privacy Extensions for Stateless Address Autoconfiguration in IPv6", RFC 4941, DOI 10.17487/RFC4941, September 2007, <http://www.rfc-editor.org/info/rfc4941>.

[RFC4944] Montenegro, G., Kushalnagar, N., Hui, J., and D. Culler, "Transmission of IPv6 Packets over IEEE 802.15.4 Networks", RFC 4944, DOI 10.17487/RFC4944, September 2007, <http://www.rfc-editor.org/info/rfc4944>.

[RFC5535] Bagnulo, M., "Hash-Based Addresses (HBA)", RFC 5535, DOI 10.17487/RFC5535, June 2009, <http://www.rfc-editor.org/info/rfc5535>.

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

[RFC6775] Shelby, Z., Ed., Chakrabarti, S., Nordmark, E., and C. Bormann, "Neighbor Discovery Optimization for IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs)", RFC 6775, DOI 10.17487/RFC6775, November 2012, <http://www.rfc-editor.org/info/rfc6775>.

[RFC6973] Cooper, A., Tschofenig, H., Aboba, B., Peterson, J., Morris, J., Hansen, M., and R. Smith, "Privacy Considerations for Internet Protocols", RFC 6973, DOI 10.17487/RFC6973, July 2013, <http://www.rfc-editor.org/info/rfc6973>.

[RFC7217] Gont, F., "A Method for Generating Semantically Opaque Interface Identifiers with IPv6 Stateless Address Autoconfiguration (SLAAC)", RFC 7217, DOI 10.17487/RFC7217, April 2014, <http://www.rfc-editor.org/info/rfc7217>.

[RFC7288] Thaler, D., "Reflections on Host Firewalls", RFC 7288, DOI 10.17487/RFC7288, June 2014, <http://www.rfc-editor.org/info/rfc7288>.

[RFC7421] Carpenter, B., Ed., Chown, T., Gont, F., Jiang, S., Petrescu, A., and A. Yourtchenko, "Analysis of the 64-bit Boundary in IPv6 Addressing", RFC 7421, DOI 10.17487/RFC7421, January 2015, <http://www.rfc-editor.org/info/rfc7421>.

[RFC7428] Brandt, A. and J. Buron, "Transmission of IPv6 Packets over ITU-T G.9959 Networks", RFC 7428, DOI 10.17487/RFC7428, February 2015, <http://www.rfc-editor.org/info/rfc7428>.

[RFC7668] Nieminen, J., Savolainen, T., Isomaki, M., Patil, B., Shelby, Z., and C. Gomez, "IPv6 over BLUETOOTH(R) Low Energy", RFC 7668, DOI 10.17487/RFC7668, October 2015, <http://www.rfc-editor.org/info/rfc7668>.

[RFC7721] Cooper, A., Gont, F., and D. Thaler, "Security and Privacy Considerations for IPv6 Address Generation Mechanisms", RFC 7721, DOI 10.17487/RFC7721, March 2016, <http://www.rfc-editor.org/info/rfc7721>.

[I-D.ietf-6man-default-iids]
Gont, F., Cooper, A., Thaler, D., and S. LIU, "Recommendation on Stable IPv6 Interface Identifiers", draft-ietf-6man-default-iids-16 (work in progress), September 2016.

[I-D.ietf-6lo-6lobac]
Lynn, K., Martocci, J., Neilson, C., and S. Donaldson, "Transmission of IPv6 over MS/TP Networks", draft-ietf-6lo-6lobac-05 (work in progress), June 2016.

[I-D.ietf-6lo-dect-ule]
Mariager, P., Petersen, J., Shelby, Z., Logt, M., and D. Barthel, "Transmission of IPv6 Packets over DECT Ultra Low Energy", draft-ietf-6lo-dect-ule-07 (work in progress), October 2016.

[I-D.ietf-6lo-nfc]
Choi, Y., Youn, J., and Y. Hong, "Transmission of IPv6 Packets over Near Field Communication", draft-ietf-6lo-nfc-05 (work in progress), October 2016.

[I-D.huitema-6man-random-addresses]
Huitema, C., "Implications of Randomized Link Layers Addresses for IPv6 Address Assignment", draft-huitema-6man-random-addresses-03 (work in progress), March 2016.

[BTCorev4.1]
Bluetooth Special Interest Group, "Bluetooth Core Specification Version 4.1", December 2013, <https://www.bluetooth.org/DocMan/handlers/DownloadDoc.ashx?doc_id=282159>.

## Author’s Address

Dave Thaler
Microsoft
One Microsoft Way
Redmond, WA 98052
USA

Email: dthaler@microsoft.com

6lo
Internet-Draft
Intended status: Standards Track
Expires: January 9, 2017

D. Migault
Ericsson
T. Guggemos
LMU Munich
C. Bormann
Universitaet Bremen TZI

July 8, 2016

# Diet-ESP: a flexible and compressed format for IPsec/ESP
draft-mglt-6lo-diet-esp-02.txt

## Abstract

This document defines Diet-ESP that adapts IPsec/ESP for IoT. Diet-ESP compresses fields of the Standard ESP. The compression is defined by profiles based ROHC and ROHCoverIPsec as well as parameters mentioned in the Diet-ESP Context agreed between the two Diet-ESP peers for example using IKEv2.

## 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 http://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 9, 2017.

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. Requirements notation

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

# 2. Introduction

## 2.1. IoT context

The IPsec/ESP [RFC4303] is represented in Figure 1. It was designed to: 1) provide general purposes security protocol with high level of security, 2) favor interoperability between implementations and 3) scale on large infrastructures.

In order to match these goals, ESP format favor mandatory fields with fixed sizes that are designed considering extreme or worst case scenarios. This results in a kind of "unique" packet format common to all considered scenarios using ESP. On the other hand ESP ends up carrying "unnecessary" or "larger than required" fields. This cost of additional bytes were considered as negligible versus interoperability, making ESP very successful over the years.

With IoT, requirements become slightly different. For most devices, like sensors, sending extra bytes directly impacts the battery and so the life time of the sensor. As a result, IoT may look at reducing the number of bytes sent on the wire.

This document describes Diet-ESP which compresses fields of the Standard ESP. The compression is defined by profiles based ROHC and ROHCoverIPsec as well as parameters mentioned in the Diet-ESP Context agreed between the two Diet-ESP peers for example using IKEv2 [RFC7296]

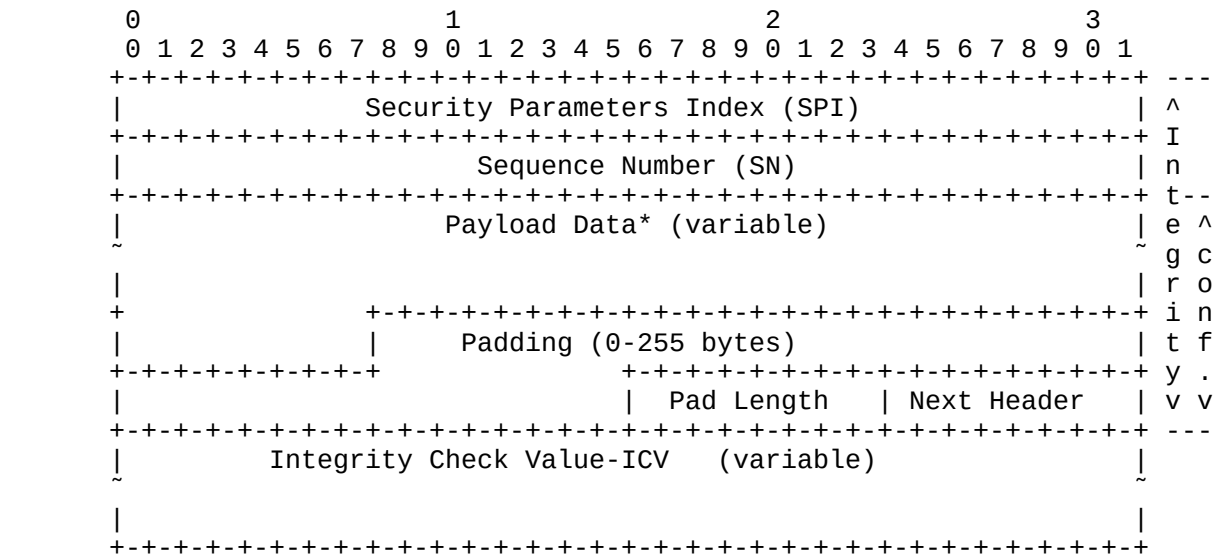



Figure 1: ESP Packet Description

## 2.2. Document Overview

This document describes how to compress ESP fields sent on the wire. Concerned fields are those of the ESP Header and the ESP Trailer. Compression of the Payload Data, including the Initialization Vector (IV) or the Integrity Check Value (ICV) are out of scope of the document.

The compression mechanisms defined in this document are based on ROHC [RFC3095], [RFC5225] and ROHCoverIPsec [RFC5856], [RFC5857], [RFC5858].

ROHC defines mechanisms to compress/decompress fields of an IP packet. These compressors are placed between the MAC layer and the IP layer. In the case of ESP, ROHC can be used to compress/ decompress SPI, SN. However, ROHC cannot be used to compress encrypted fields like Padding, Pad Length, Next Header -- and later the Clear Text Data before encryption. In fact, at the MAC layer, these fields are encrypted and their encrypted value is used generate the ICV. As a result, compression an ESP packet at the MAC layer requires to decrypt the packet to be able to compress fields like Clear Text Data, Pad Length in order to be able to eventually remove the Padding Field. Similarly, decompressing a compress ESP packet at the MAC layer would require to decrypt the received packet, decompress the packet the Clear Text Data as well as the other ESP fields, before forwarding the ESP packet to the IP stack. Note that

in some case decompression is not feasible. Consider for example an ESP implementation that generates a random Padding. If this field is removed by the compressor, it can hardly be recovered by the decompressor. Using a different Padding field would result is ESP rejecting the packet as the ICV check will not succeed. As a result ROHC cannot be used alone.

On the other hand, ROHCoverIPsec makes compression possible before the ESP payload is encrypted, and so the Clear Text Data can be compressed, but not the ESP related fields like Padding, Pad Length and Next Header.

ROHC and ROHCoverIPsec have been designed for bandwidth optimization, but not necessarily for constraint devices. As a result, defining ROHC and ROHCoverIPsec profiles is not sufficient to fulfill the complete set of Diet-ESP requirements listed in [I-D.mglt-6lo-diet-esp-requirements]. In fact Diet-ESP MUST result in an light implementation that does not require implementation of the full ROHC and ROHCoverIPsec frameworks.

In order to achieve ESP field compression, this document describes the Diet-ESP Context. This context contains all necessary parameters to compress an ESP packet. This Diet-ESP Context can be provided as input to proceed to Diet-ESP compression / decompression. This document uses the ROHC and ROHCoverIPsec framework to compress the ESP packet. The advantage of using ROHC and ROHCoverIPsec is that compression behavior follows a standardized compression framework. On the other hand, ROHC and ROHCoverIPsec frameworks are used in a stand alone mode, which means no ROHC communications between compressor and decompressor are considered. This enables specific and lighter implementations to perform Diet-ESP compression without implementing the ROHC or ROHCoverIPsec frameworks. All Diet-ESP implementations only have to agree on the Diet-ESP Context to become inter-operable.

The remaining of the document is as follows. Section 4 described the Diet-ESP Context. Section 5 describes how the parameters of the Diet-ESP Context are used by the ROHC and ROHCoverIPsec framework to compress the ESP packet. This requires definition of new profiles and extensions. Finally, Section 7 and Section 8 provides a security and privacy analysis on Diet-ESP over standard ESP. Informational material have been added into the Appendix section. Appendix A illustrates how an minimal Diet-ESP implementation may be used in IoT devices. Appendix B lists the differences between Diet-ESP and Standard ESP. Appendix C describes the interactions of Diet-ESP interacts with other compression protocols such as 6lowPAN and ROHC compression for other protocols than ESP. Finally, Appendix D

describes how Diet-ESP matches the requirements for Diet-ESP [I-D.mglt-6lo-diet-esp-requirements].

## 3. Terminology

This document uses the following terminology:

- IoT: Internet of Things

- LSB: Last Significant Bytes

- IP alignment: The necessary alignment for IPv4 (32 bits) resp. IPv6 (64 bits)

- Clear Text Data: designates the original data that are carried by ESP.

- Encrypted Payload: carries the encrypted Data Payload including cryptographic material like the IV and the ESP Trailer.

## 4. Diet-ESP Context

The Diet-ESP context provides the necessary parameters for the compressor and decompressor to perform the appropriated compression and decompression of the ESP packet. Table 1 the different parameters of the Diet-ESP Context.

| Context Field Name | Overview |
|---|---|
| ALIGN | Necessary Alignment for the specific device. |
| SPI_SIZE | Size in bytes of the SPI field in the sent packet. |
| SN_SIZE | Size in byte of the SN field in the sent packet. |
| NH | Presence of the Next Header field in the ESP Trailer. |
| PAD | Presence of the Pad Length field present in the ESP trailer. |

Table 1: Diet-ESP Context.

ALIGN:
Alignment is the minimum alignment accepted by the hardware. Constrains may come from various reasons. Hardware may have some specific requirements, but also operating systems. For most

servers CPU and OS have been designed with 32 bit or 64 bit alignment. As a result, IP headers have been standardized with 32 bits (resp. 64 bits in IPv6) alignment for each IP extension header. ESP is one of these extension headers with an Header (SPI and SN) of 64 bits and the Trailer (NH, PL, PAD) of (2 + PL) bytes. Since the trailer is part of the ESP extension header, it must provide the necessary padding for a correct alignment of the NH field to 32 (resp. 64) bits. The alignment may also be relevant if Block-Ciphers like AES-CBC needs an aligned payload to perform the encryption. To inter-oprate with the standard ESP, IP alignment must be 32 bit for IPv4 and 64 bit for IPv6.

Diet-ESP reduces the ALIGN value from 32 bits for IPv4 or 64 bits for IPv6 to 8, 16, 32 and 64 bit alignment.

Motivations to do so is to remove the Padding and other mandatory fields of the ESP packet. Then, most IoT embeds small 8 or 16 bit CPUs. Finally, even though ESP is an extension header, it is often the last extension header of a header-only IP packet. The ESP header is only read by the real receiver and is uninteresting for other devices like routers, placed between the to peers. As a result, there seems no real impact on the system if ESP extension header is not aligned.

Note that the benefices of ALIGN also depends on the used cryptographic mode. More specifically AES-CTR has a 8 bit block whereas AES-CBC has a 128 bit block. As a result the use of AES-CBC with small Clear Text Data results in large encrypted Data with embedded padding. In other words, the alignment for one packet is always MAX(CIPHER_BLOCK_SIZE, ALIGN).

SPI_SIZE:
ESP Security Policy Index is 4 byte long to identify the SAD-entry for incoming traffic. To interoperate with the standard ESP, the SPI_SIZE must be of 4 bytes.
Diet-ESP omits, leaves unchanged, or reduces the SPI sent on the wire to the 0, 1, 2, 3 or 4 LSB.
Compression only impacts the data sent on the wire and therefore SHOULD only deal with 4 byte decompressed SPIs in the SAD. This allows systems to send and receive multiple SPI_SIZE with different hosts. Decompressing the SPI at the receiver may involve IP addresses (see Appendix B.2.1).
Compressing the SPI has significant security impacts as detailed in Section 7. It should be guided by 1) the number of simultaneous inbound SA the device is expected to handle and 2) reliability of the IP addresses in order to identify the proper SA for incoming packets. More specifically, a sensor with a single connection to a Security Gateway, may bind incoming packets to the

proper SA based only in its IP addresses. In that case, the SPI may not be necessary. Other scenarios may consider using the SPI to index the SAs or may consider having multiple ESP channels with the same host from a single host. In that case one may choose a reduced length for the SPI. Note also that the value 0 for the SPI is not allowed to be sent on the wire as described in [RFC4303].

SN_SIZE:
ESP Sequence Number is 32 bit and extended SN is 64 bit long and used for anti-replay protection. To interoperate with standard ESP the SN_SIZE must be of 4 bytes.
Diet-ESP omits, leaves unchanged or reduces SN sent on the wire to 0, 1, 2, 3 or 4 LSB.
Decompressing the SN at the receiver is guided by a linear extrapolation of the expected received Sequence Number and the LSB-SN sent on the wire. To avoid packet overhead, this configuration is stored within the SA, whereas it remains valid during its lifetime. Therefore an implementer should consider the LSB window such that two consecutive received SN should not present a difference of more than the LSB window.
In some cases, the received SN may increase by a high number e.g. using the time as the SN or because of a high number of packet loss. See Section 7 for the related security considerations.

Note that SN and SPI MUST be aligned to a multiple of the Alignment value (ALIGN).

NH:
Next Header in ESP is used to identify the first header inside the ESP payload. To interoperate with standard ESP, the Next Header must be indicated and present.
Diet-ESP is able to remove the Next Header field from the ESP-Trailer.
Removing the Next Header is possible only if the underlying protocol can be derived from the Traffic Selector (TS) within the Security Association (SA). More specifically, the Next Header indicates whether the encrypted ESP payload is an IP packet, a UDP packet, a TCP packet or no next header. The NH can only be removed if this has been explicitly specified in the SA or if the device has a single application.
Note that removing the Next Header impacts how encryption is performed. For example, the use of AES-CBC [RFC3602] mode requires the last block to be padded, reaching a 128 bit alignment. In this case removing the Next Header increases the padding by the Next Header length, which is 8 bits. In this case, removing the Next Header provides few advantages, as it does not reduce the ESP packet length. With AES-CBC, the only advantage of

removing the Next Header would be for data with the last block of 15 bytes. In that case, ESP pads with 15 modulo 16 bytes, sets the 1 byte pad length field to 15 and add the one byte Next Header field. This leads to 15 + 15 + 1 + 1 = 32 bytes to be sent. On the other hand, removing the Next Header would require only the concatenation of the pad length byte with a 0 value, which leads to 16 bytes to be sent.
Other modes like AES-CTR [RFC3686] do not have block alignment requirements, so the only alignment constraint comes from the device hardware alignment (ALIGN). Suppose A designates the alignment constraint from OS, hardware, encryption, packet format...). A is fixed and consider then any data of length k * A + A - 1 bytes with k an integer. Sending this data using ESP takes advantage of removing the Next Header as it reduces the number of bytes to be sent by A over the traditional ESP. As a result, for 8 bit alignment hardware (A = 1) removing the Next Header always prevent an unnecessary byte to be sent.

PAD:
With ESP, all packets have a Pad Length field. This field is usually present because ESP requires IP alignment which is ensured with padding. In order to interoperate with the standard ESP, the Pad Length must be indicated and be present.
Diet-ESP considers removing the Padding and the Pad Length field. If PAD is present, then it is computed according to ALIGN.
In fact, some devices might use an 8 bits alignment, in which case padding is not necessary. Similarly, sensors may send application data of fixed length matching the alignment. Note that alignment may be required by the device (8-bit, 16-bit, or more generally 32-bit), but it may also be required by the encryption block size (AES-CBC uses 128 bit blocks). With ESP these scenarios would result in an unnecessary Pad Length field always set to zero. Diet-ESP considers those case with no padding, and thus the Pad Length field can be omitted.

Some additional parameters may be added to the Diet-ESP Context. Such parameters are defined in other documents, like [I-D.mglt-6lo-diet-esp-payload-compression] the compression of protocol headers inside the encrypted ESP payload.

## 5. Diet-ESP Protocol Description

This section defines Diet-ESP on the top of the ROHC and ROHCoverIPsec framework. Section 5.1 presents and explains the choice of these frameworks. This section is informational and its only goal is to position our work toward ROHC and ROHCoverIPsec.

### 5.1. Robust Header Compression (ROHC)

ROHC enables the compression of different protocols of all layers. It is designed as a framework, where protocol compression is defined as profile. Each profile is defined for a specific layer, and ROHC compression in [RFC3095] defines profiles for the following protocols: uncompressed, UDP/IP, ESP/IP and RTP/UDP/IP. The compression occurs between the IP and the MAC layer, and so remains independent of an eventual IP alignment.

The general idea of ROHC is to classify the different protocol fields. According to the classification, they can either completely and always be omitted, omitted only after the fields has been sent once and registered by the receiver or partly sent and be regenerated by the receiver. For example, a static field value may be negotiated out of band (for example IP version) and thus not be sent at all. In some cases, the value is not negotiated out of band and is carried in the first packet (for example SPI, UDP ports). As a result, the first packet is usually not so highly compressed with ROHC. Finally, some variable fields (for example Sequence Number) can be represented partially by their Last Significant Bits (LSB) and regenerated by the receiver.

The main issue encountered with ESP and ROHC is that ESP may contain encrypted data which makes compression between the IP and MAC layer complex to achieve. Therefore ROHC defines different compression of the ESP protocol (see Figure 2), so compression of the Clear Text Data can occur before the ESP encryption. Regular ROHC can compress the ESP header. If the packet is not encrypted, the rate of compression is extremely high as the whole packet including padding can be compressed in the regular ROHC stack, too. For encrypted payload ROHC defines ROHCoverIPsec ([RFC5856], [RFC5857], [RFC5858]) to compress the ESP payload before it is going to be encrypted. This leads to a second ESP stack, where another ROHC compressor (resp. decompressor) works (see Figure 2). Excluding the first packet which initializes the ROHC context, this makes ESP compression highly efficient.

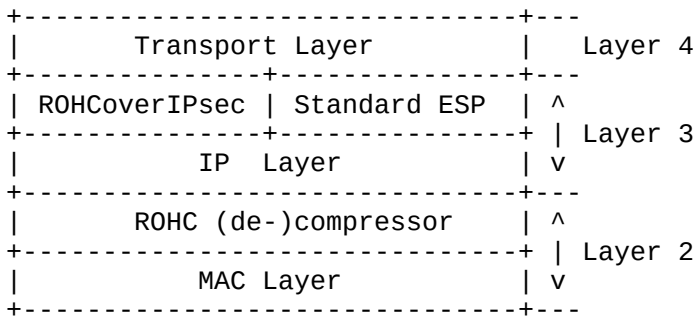



Figure 2: The two different ROHC layers in the TCP/IP stack.

The first drawback for ROHC and ROHCoverIPsec is, that it leads to two ROHC compression layers (ROHCoverIPsec before ESP encryption and ROHC before the MAC layer) in addition to two ESP implementations (Standard ESP and ROHCoverIPsec ESP). Both frameworks are quite complex and require a lot of resources which does not fit IoT requirements. Then ROHCoverIPsec also limits the compression of the ESP protocol, according to the IP restrictions. Padding remains necessary as IPsec is part of the IP stack which requires a 32 bits (resp. 64 bits for IPv6) aligned packet. This makes compression quite inefficient when small amount of data are sent.

Note that mechanism to compress encrypted fields may be possible with ROHC only and without ROHCoverIPsec. Such mechanisms may be possible for fields like the Next Header or the Padding and Pad Length when the data sent is of fixed size. As the sizes of the fields are known the compressor may simply remove these fields. However, even in this case, it almost doubles the amount of computation on the receiver's side. In fact, the ROHC compressor would almost decompress and re-encrypt the compressed ESP payload before forwarding it to the IP stack. In addition, since the receiver has to re-encrypt the decompressed information before integrity of the packet can be checked, one can easily construct a DoS attack. Flooding the receiver with invalid packet causes the receiver to perform the complex encryption and authentication algorithm for each packet.

## 5.2. Diet-ESP ROHC framework

This section defines how the compression of all ESP fields is performed within the ROHC and ROHCoverIPsec frameworks. More especially fields that are in the ESP Header (i.e. the SPI and the SN) and the ICV are compressed by the ROHC framework. The other fields, that is to say those of the ESP Trailer, are compressed by the ROHCoverIPsec framework.

Diet-ESP fits in the ROHC and the ROHCoverIPsec in a very specific way.

1 - Diet-ESP does not need any ROHC signaling between the peers. More specifically, ROHC Initialization and Refresh (IR), or ROHC IR-DYN or ROHC Feed back packet are not considered with Diet-ESP. The first reason is that fields are either STATIC or PATTERN and their value or profile is defined through the Diet-ESP Context agreement. This agreement is out of scope of ROHC, it is expected to be agreed by other protocols like IKEv2 and thus is considered as an out-of band agreement by the peers. Then, the profiles are applied for each Security Association that is unidirectional. In fact an IKEv2 negotiation results in two unidirectional SA. As a result, each SA the packets are sent in one direction only, which corresponds to the Unidirectional mode -- U-mode of ROHC.

2 - Diet-ESP only exchanges compressed data. How the compression / decompression occurs is defined by the Diet-ESP Context. Once the Diet-ESP Context has been agreed, both peers are in a Second Order (SO) State and exchange only compressed data.

3 - Diet-ESP only compresses ESP packets, it may include inner packet compression, but Diet-ESP does not make any assumption on the IP compression. This is made in order to make Diet-ESP interoperable with multiple IP compression protocols.

4 - Diet-ESP compresses partially STATIC fields as they are used as indexes by the receiver, and may not completely be removed.

### 5.3. Diet-ESP header classification

The ROHC header field classifications are defined in Appendix A.1 of [RFC3095] and Appendix A of [RFC5225].

| Field | ROHC class | Framework | Encoding Method | Diet-ESP Context Parameters |
|---|---|---|---|---|
| SPI | STATIC-DEF | ROHC | LSB | SPI_SIZE |
| SN | PATTERN | ROHC | LSB | SN_SIZE |
| Padding | PATTERN | ROHCoverIPsec | Removed | PAD, ALIGN |
| Pad Length | PATTERN | ROHCoverIPsec | Removed | PAD, ALIGN |
| Next Header | STATIC-DEF | ROHCoverIPsec | Removed | NH |

Table 2: Diet-ESP ROHC profile.

SPI:
   The SPI indexes the SA, is negotiated by the two peers (e.g. via IKEv2 or manually) and remains the same during the session. Therefore, as defined in Appendix A.6 of [RFC5225] this field is classified as STATIC-DEF. The compressed SPI consists in the SPI_SIZE LSB of the negotiated 32 bit SPI, and SPI_SIZE is provided by the Diet-ESP Context.

SN:
   The SN is used for anti-replay protection and is modified in every packet. In default cases, the ESP Sequence Number will be incremented by one for each packet sent. Therefore, as defined in Appendix A.6 of [RFC5225] this field is classified as PATTERN. The compressed SN consists in the SN_SIZE LSB of the 32 bit or 64 bit SN, and SN_SIZE is provided by the Diet-ESP Context.

Padding:
   Padding is used for alignment purposes and is computed on a per-packet basis. Therefore it is classified as PATTERN. The compressed Padding is defined by PAD and ALIGN provided by the the Diet-ESP Context. If PAD is set the Padding and Pad Length fields are removed. If PAD is unset, Padding is computed according to the ALIGN and the padding length is indicated in the PAD Length field.

Pad Length:
   Pad Length indicates the length of the Padding field and is computed on a per-packet basis. Therefore it is classified as PATTERN. See Padding for the compressed Pad Length.

Next Header:

The Next Header indicates the next layer in the inner ESP Payload. To be compressed the Next Header MUST remain the same during the session. This means that it MUST have been negotiated (e.g. by IKEv2) and can be derived from the Traffic Selectors. If this condition is met and the Next Header compression is requested by the peers with NH set in the Diet-ESP Context, then the Next Header field MUST be removed.

## 6. IANA Considerations

There are no IANA consideration for this document.

## 7. Security Considerations

This section lists security considerations related to the Diet-ESP protocol.

Security Parameter Index (SPI):
The Security Parameter Index (SPI) is used by the receiver to index the Security Association that contains appropriated cryptographic material. If the SPI is not found, the packet is rejected as no further checks can be performed. In Diet-ESP, the value of the SPI is not reduced, but compressed why the SPI value may not be fully provided between the compressor and the decompressor. On the other hand, its uncompressed value is provided to the ESP-procession and no weakness is introduced to ESP itself. On an implementation perspective, it is strongly recommended that decompression is deterministic. Compression and decompression adds some additional treatment to the ESP packet, which might be used by an attacker. In order to minimize the load associated to decompression, decompression is expected to be deterministic. The incoming compressed SPI with the associated IP addresses should output a single and unique uncompressed SPI value. If n uncompressed SPI values have to be considered, then the receiver could end in n signature checks which may be used by an attacker for a DoS attack. On a privacy perspective, until Diet-ESP is not deployed outside the scope of IoT and small devices, the use of a compressed SPI may provide an indication that one of the endpoint is a sensor. Such information may be used, for example, to evaluate the number of appliances deployed, or - in addition with other information, such as the time interval, the geographic location - be used to derive the type of data transmitted.

Sequence Numer (SN):
The Sequence Number (SN) is used as an anti-replay attack mechanism. Compression and decompression of the SN is already part of the standard ESP namely the Extended Sequence Number

(ESN). The SN in a standard ESP packet is 32 bit long, whether Diet-ESP enables to reduce it to 0 bytes and the main limitation to the compression a deterministic decompression. SN compression consists in indicating the least significant bits of the uncompressed SN on the wire. The size of the compressed SN must consider the maximum reordering index such that the probability that a later sent packet arrives before an earlier one. In addition the size of SN should also consider maximum consecutive packets lost during transmission. In the case of ESP, this number is set to 2^32 which is, in most real world case, largely over-provisioned. When the compression of the SN is not appropriately provisioned, the most significant bit value may be desynchronized between the sending and receiving parties. Although IKEv2 provides some re-synchronization mechanisms, in case of IoT the de-synchronization will most likely result in a renegotiation and thus DoS possibilities. Note that IoT communication may also use some external parameters, i.e. other than the compressed SN, to define whether a packet be considered or not and eventually derive the SN. One such scenario may be the use of time windows. Suppose a device is expected to send some information every hour or every week. In this case, for example, the SN may be compressed to zero bytes. Instead the SN may be derived by incrementing the SN every hour after the last received valid packet. Considering the time the packet is received make it possible to consider the time derivation of the sensor clock. Note also that the anti-replay mechanism needs to define the size of the anti-replay window. [RFC4303] provides guidance to set the window size and are similar to those used to define the size of the compressed SN

ESP Trailer:
Padding, Pad Length and Next Header are fields stored inside the encrypted payload. They are part of the ESP payload so that ESP can be seen as an IP option. IP extension headers must have 32 bit Byte-Alignment in IPv4 [43] and 64 bit Byte-Alignment in IPv6 [6]. The main motivation for the alignment is the improvement of packet processing on 32 or 64 bit processors. As a result, compression of these static fields does not impact the security of Diet-ESP compared to the one provided by ESP.

## 8. Privacy Considerations

Security Parameter Index (SPI):
Until Diet-ESP is not deployed outside the scope of IoT and small devices, the use of a compressed SPI may provide an indication that one of the endpoint is a sensor. Such information may be used, for example, to evaluate the number of appliances deployed, or - in addition with other information, such as the time

interval, the geographic location - be used to derive the type of data transmitted.

Sequence Number (SN): If incremented for each ESP packet, the SN may leak some information like the amount of transmitted data or the age of the sensor. The age of the sensor may be correlated with the software used and the potential bugs. On the other hand, re-keying will re-initialize the SN, but the cost of a re-keying may not be negligible and thus, frequent re-keying can be considered. In addition to the re-key operation, the SN may be generated in order to reduce the accuracy of the information leaked. In fact, the SN does not have to be incremented by one for each packet it just has to be an increasing function. Using a function such as a clock may prevent characterizing the age or the use of the sensor. Note that the use of such function may also impact the compression efficiency and result in larger compressed SN.

## 9. Acknowledgment

The current work on Diet-ESP results from exchange and cooperation between Orange, Ludwig-Maximilians-Universitaet Munich, Universite Pierre et Marie Curie. We thank Daniel Palomares and Carsten Bormann for their useful remarks, comments and guidances on the design. We thank Sylvain Killian for implementing an open source Diet-ESP on Contiki and testing it on the FIT IoT-LAB [fit-iot-lab] funded by the French Ministry of Higher Education and Research. We thank the IoT-Lab Team and the INRIA for maintaining the FIT IoT-LAB platform and for providing feed backs in an efficient way. We thank Ana Minaburo for her ROHC review.

## 10. References

### 10.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <http://www.rfc-editor.org/info/rfc2119>.

[RFC3095] Bormann, C., Burmeister, C., Degermark, M., Fukushima, H., Hannu, H., Jonsson, L-E., Hakenberg, R., Koren, T., Le, K., Liu, Z., Martensson, A., Miyazaki, A., Svanbro, K., Wiebke, T., Yoshimura, T., and H. Zheng, "RObust Header Compression (ROHC): Framework and four profiles: RTP, UDP, ESP, and uncompressed", RFC 3095, DOI 10.17487/RFC3095, July 2001, <http://www.rfc-editor.org/info/rfc3095>.

[RFC3602] Frankel, S., Glenn, R., and S. Kelly, "The AES-CBC Cipher Algorithm and Its Use with IPsec", RFC 3602, DOI 10.17487/RFC3602, September 2003, <http://www.rfc-editor.org/info/rfc3602>.

[RFC3686] Housley, R., "Using Advanced Encryption Standard (AES) Counter Mode With IPsec Encapsulating Security Payload (ESP)", RFC 3686, DOI 10.17487/RFC3686, January 2004, <http://www.rfc-editor.org/info/rfc3686>.

[RFC4104] Pana, M., Ed., Reyes, A., Barba, A., Moron, D., and M. Brunner, "Policy Core Extension Lightweight Directory Access Protocol Schema (PCELS)", RFC 4104, DOI 10.17487/RFC4104, June 2005, <http://www.rfc-editor.org/info/rfc4104>.

[RFC4301] Kent, S. and K. Seo, "Security Architecture for the Internet Protocol", RFC 4301, DOI 10.17487/RFC4301, December 2005, <http://www.rfc-editor.org/info/rfc4301>.

[RFC4303] Kent, S., "IP Encapsulating Security Payload (ESP)", RFC 4303, DOI 10.17487/RFC4303, December 2005, <http://www.rfc-editor.org/info/rfc4303>.

[RFC4309] Housley, R., "Using Advanced Encryption Standard (AES) CCM Mode with IPsec Encapsulating Security Payload (ESP)", RFC 4309, DOI 10.17487/RFC4309, December 2005, <http://www.rfc-editor.org/info/rfc4309>.

[RFC4555] Eronen, P., "IKEv2 Mobility and Multihoming Protocol (MOBIKE)", RFC 4555, DOI 10.17487/RFC4555, June 2006, <http://www.rfc-editor.org/info/rfc4555>.

[RFC5225] Pelletier, G. and K. Sandlund, "RObust Header Compression Version 2 (ROHCv2): Profiles for RTP, UDP, IP, ESP and UDP-Lite", RFC 5225, DOI 10.17487/RFC5225, April 2008, <http://www.rfc-editor.org/info/rfc5225>.

[RFC5857] Ertekin, E., Christou, C., Jasani, R., Kivinen, T., and C. Bormann, "IKEv2 Extensions to Support Robust Header Compression over IPsec", RFC 5857, DOI 10.17487/RFC5857, May 2010, <http://www.rfc-editor.org/info/rfc5857>.

[RFC5858] Ertekin, E., Christou, C., and C. Bormann, "IPsec Extensions to Support Robust Header Compression over IPsec", RFC 5858, DOI 10.17487/RFC5858, May 2010, <http://www.rfc-editor.org/info/rfc5858>.

[RFC7296] Kaufman, C., Hoffman, P., Nir, Y., Eronen, P., and T. Kivinen, "Internet Key Exchange Protocol Version 2 (IKEv2)", STD 79, RFC 7296, DOI 10.17487/RFC7296, October 2014, <http://www.rfc-editor.org/info/rfc7296>.

[RFC7321] McGrew, D. and P. Hoffman, "Cryptographic Algorithm Implementation Requirements and Usage Guidance for Encapsulating Security Payload (ESP) and Authentication Header (AH)", RFC 7321, DOI 10.17487/RFC7321, August 2014, <http://www.rfc-editor.org/info/rfc7321>.

### 10.2. Informational References

[fit-iot-lab]
"Future Internet of Things (FIT) IoT-LAB", <https://www.iot-lab.info>.

[I-D.mglt-6lo-diet-esp-payload-compression]
Migault, D. and T. Guggemos, "Diet-IPsec: ESP Payload Compression of IPv6 / UDP / TCP / UDP-Lite", January 2015.

[I-D.mglt-6lo-diet-esp-requirements]
Migault, D. and T. Guggemos, "Requirements for Diet-ESP the IPsec/ESP protocol for IoT", draft-mglt-6lo-diet-esp-requirements-01 (work in progress), February 2015.

[I-D.raza-6lowpan-ipsec]
Raza, S., Duquennoy, S., and G. Selander, "Compression of IPsec AH and ESP Headers for Constrained Environments", draft-raza-6lowpan-ipsec-01 (work in progress), September 2013.

[RFC5856] Ertekin, E., Jasani, R., Christou, C., and C. Bormann, "Integration of Robust Header Compression over IPsec Security Associations", RFC 5856, DOI 10.17487/RFC5856, May 2010, <http://www.rfc-editor.org/info/rfc5856>.

## Appendix A. Example of light Diet-ESP implementation for sensor

Diet-ESP has been designed to enable light implementation. This section illustrates the case of a sensor sending a specific amount of data periodically. This section is not normative and has only an illustrative purpose. In this scenario the sensor measures a temperature every minute and sends its value to a gateway, which is assumed to collect the data. The data is sent in an UDP packet and there is no other connection between the two peers. The communication between the sensor and the gateway should be secured by

a Diet-ESP connection in transport mode. Therefore the following context is chosen:

ALIGN: 8 bit
Sensors are not expected to be 32 or 64 bit CPU, and micro-controllers are expected to support 8 bit alignment.

SPI_SIZE: 0
As it is a single connection, the SA can be identified by using the IP addresses. As a result the SPI is not needed.

SN_SIZE: 0
Because only one packet every minute is sent, the packets will arrive at the receiver in an ordered way. The receiver can rebuild the SN which should be present in the packet, assuming the SN is incremented by one for each packet. Note that setting SN to 0 does not mean there is no anti replay protection. In fact, the SN is needed for the computation of the Diet-ESP ICV.

NH: Remove Next Header
Since the protocol is always UDP, the Next header can be omitted.

PAD: Remove Padding
With 8 bit alignment Padding has always a Pad Length of 0. Setting PAD to "Remove Padding" removes the Pad Length field.

Diet-ESP_ICV_SIZE: 4 bytes
The ICV is chosen to be 32 bits in order to find a fair trade-off between security and energy costs.

Encapsulating the outgoing Diet-ESP packet is proceeded as follows:

1) SAD lookup for outgoing traffic

2) Compress ESP payload incl. Transport Header (UDP)

3) Encrypt IP payload

4) Build ESP header

5) Calculate Diet-ESP ICV

6) Compress ESP header

7) Add ${Diet-ESP_ICV_SIZE} LSB of ICV to the packet.

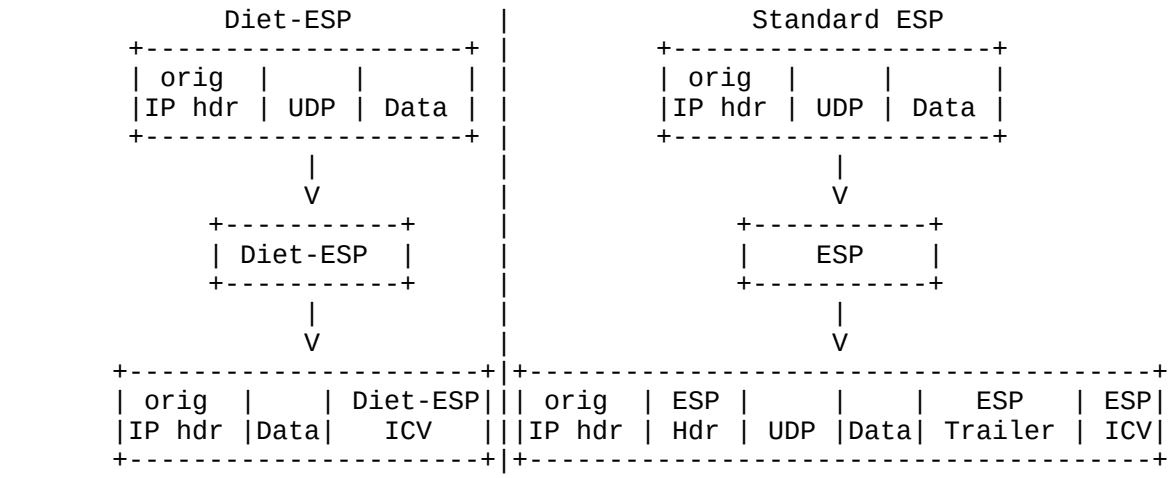



Figure 3: Minimal Example - Input and Output of the Diet-ESP function vs. Standard ESP.

Incoming Diet-ESP packet is processed as follows:

1) SAD lookup for incoming traffic traffic

2) Decompress ESP-header incl. Transport Header (UDP)

3) Calculate packet Diet-ESP ICV

4) Check integrity with ${Diet-ESP_ICV_SIZE} LSB of Diet-ESP ICV

5) Check anti-replay

6) Decrypt IP payload (excluding ICV)

7) Decompress ESP payload

## Appendix B. Difference between Diet-ESP and ESP

This section details how to use Diet-ESP to send and receive messages. The use of Diet-ESP is based on the IPsec architecture [RFC4301] and ESP [RFC4303]. We suppose the reader to be familiar with these documents and we list here possible adaptations that may be involved by Diet-ESP.

### B.1. Packet Alignment

Each ESP packet has a fixed alignment to 32 bits (resp. 64 bits in IPv6). For Diet-ESP each device has an internal parameter that defines the minimal acceptable alignment. ALIGN SHOULD be a the maximum of the peer's minimal alignment.

Diet-ESP Context with SPI_SIZE + SN_SIZE that is not a multiple of ALIGN MUST be rejected.

### B.2. SAD

#### B.2.1. Inbound Security Association Lookup

For devices that are configured with a single SPI_SIZE value can process inbound packet as defined in [RFC4301]. As such, no modifications is required by Diet-ESP.

Detecting Inbound Security Association: Identifying the SA for incoming packets is a one of the main reasons the SPI is send in each packet on the wire. For regular ESP (and AH) packets, the Security Association is detected as follows:

1. Search the SAD for a match on {SPI, destination address, source address}. If an SAD entry matches, then process the inbound ESP packet with that matching SAD entry. Otherwise, proceed to step 2.

2. Search the SAD for a match on {SPI, destination address}. If the SAD entry matches, then process the inbound ESP packet with that matching SAD entry. Otherwise, proceed to step 3.

3. Search the SAD for a match on only {SPI} if the receiver has chosen to maintain a single SPI space for AH and ESP, or on {SPI, protocol} otherwise. If an SAD entry matches, then process the inbound ESP packet with that matching SAD entry. Otherwise, discard the packet and log an audible event.

For device that are dealing with different SPI_SIZE SPI, the way inbound packets are handled differs from the [RFC4301]. In fact, when a inbound packet is received, the peer does not know the SPI_SIZE. As a result, it does not know the SPI that applies to the incoming packet. The different values could be the 0 (resp. 1, 2, 3 and 4) first bytes of the IP payload.

Since the size of the SPI is not known for incoming packets, the detection of inbound SAs has to be redefined in a Diet-ESP environment. In order to ensure a detection of a SA the above

described regular detection have to be done for each supported SPI size (in most cases 5 times). In most common cases this will return a unique Security Association.

If there is more than one SA matching the lookup, the authentication MUST be performed for all found SAs to detect the SA with the correct key. In case there is no match, the packet MUST be dropped. Of course this can lead into DoS vulnerability as an attacker recognizes an overlap of one or more IP-SPI combinations. Therefore it is highly recommended to avoid different values of the SPI_SIZE for one tuple of Source and Destination IP address. Furthermore this recommendation becomes mandatory if NULL authentication is supported. This is easy to implement as long as the sensors are not mobile and do not change their IP address.

The following optimizations MAY be considered for sensor that are not likely to perform mobility or multihoming features provided by MOBIKE [RFC4555] or any change of IP address during the lifetime of the SA.

Optimization 1 - SPI_SIZE is mentioned inside the SPI:
The SPI_SIZE is defined as part of the SPI sent in each packet. Therefore the receiver has to choose the most significant 2 bits of the SPI in the following way in order to recognize the right size for incoming Diet-ESP packets:

00: SPI_SIZE of 1 byte is used.

01: SPI_SIZE of 2 byte is used.

10: SPI_SIZE of 3 byte is used.

11: SPI_SIZE of 4 byte is used.

If the the value 0 is chosen for the SPI_SIZE this option is not feasible.

Optimization 2 - IP address based lookup:
IP address based search is one optimization one may choose to avoid several SAD lookups. It is based on the IP address and the stored SPI_SIZE, which MUST be the same value for each SA of one IP address. Otherwise it can’t neither be ensured that an SA is found nor that the correct one is found. Note that in case of mobile IP the SPI_SIZE MUST be updated for all SAs related to the new IP address which may cause renegotiation. Figure 4 shows this lookup described below.

1. Search most significant SA as follows:

1.1 Search the first SA for a match on {destination address, source address}. If an SA entry matches, then process to step 2. Otherwise, proceed to step 1.2.

1.2 Search the first SA for a match on {source address}. If an SA entry matches, then process to step 2. Otherwise, drop the packet.

2. Identify the size of the compressed SPI for the found SA, stored in the Diet-ESP context. Note that all SAs to one IP address MUST have the same value for the SPI_SIZE. Then go to step 3.

3. If the SPI_SIZE is NOT zero, read the SPI_SIZE SPI from the packet and perform a regular SAD lookup as described in [RFC4301]. If the SPI_SIZE is zero, the SA from step 1 is unique and can be used.

Note that some implementations may collect all SPI matching the IP addresses in step 2 to avoid an additional lookup over the whole SAD. This is implementation dependent.

If the sensor is likely to change its IP address, the outcome may be a given IP address associated to different SPI_SIZE. This case may occur if one IP address has been used by a device not anymore online, but the SA has not been removed. The IP has then been provided to another device. In this case the Diet-ESP Context SHOULD NOT be accepted by the Security Gateway when the new Diet-ESP Context is provided to the Security Gateway. At least the Security Gateway can check the previous peer is reachable and then delete the SA before accepting the new SA.

Another case may be that a sensor got two interfaces with different IP addresses, negotiates a different SPI_SIZE on each interface and then use MOBIKE to move the IPsec channels from one interface to the other. In this case, the Security Gateway SHOULD NOT accept the update, or force a renegotiation of the SPI_SIZE for all SAs, basically by re-keying the SAs.

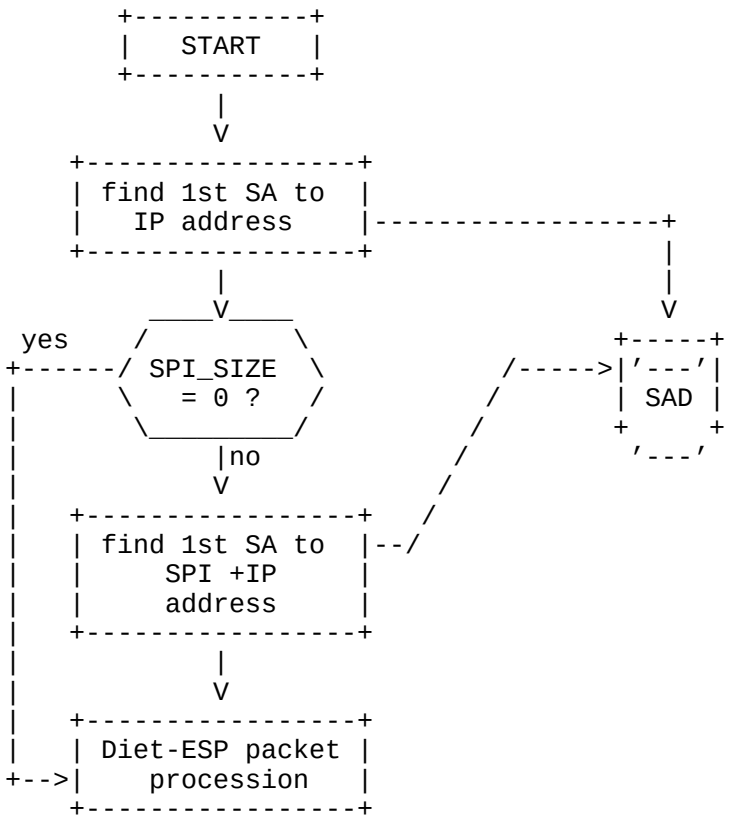



Figure 4: SAD lookup for incoming packets.

### B.2.2. Outgoing Security Association Lookup

Outgoing lookups for the SPI are performed in the same way as it is done in ESP. The Traffic Selector for the packet is searched and the right SA is read from the SA. The SPI used in the packet MUST be reduced to the value stored in SPI_SIZE.

## B.3. Sequence Number

Sequence number in ESP [RFC4303] can be of 4 bytes or 8 bytes for extended ESP. Diet-ESP introduces different sizes. One way to deal with this is to add a MAX_SN value that stores the maximum value the SN can have. Any new value of the SN will be check against this MAX_SN.

### B.4. Outgoing Packet processing

NH, TH, IH, P indicate fields or payloads that are removed from the Diet-ESP packet. How the Diet-ESP packet is generated depends on the length Payload Data LPD, BLCK the block size of the encryption algorithm and the device alignment ALIGN. We note M = MAX(BLCK, ALIGN).

- 1: Compress the headers inside the ESP payload.

- 2: if PAD and NH are set to present: Diet-ESP considers both fields Pad Length and Next Header. The Diet-ESP Payload is the encryption of the following clear text:
  Payload Data | Padding of Pad Length bytes | Pad Length field | Next Header field.
  The Pad Length value is (LPD + 2) mod [M].

- 3: if PAD is set to present and NH is set to removed: Diet-ESP considers the Pad Length field but removes the Next Header field. The ESP Payload is the encryption of the following clear text: Payload Data | Padding of Pad Length bytes | Pad Length field | Next Header field. The Pad Length value is (LPD + 1) mod [M].

- 4: if PAD is set to removed and NH is set to present: Diet-ESP considers the Next Header but do not consider the Pad Length field or the Padding Field. This is valid as long as (LPD + 1) mod [M] = 0. If M = 1 as it is the case for AES-CTR this equation is always true. On the other hand the use of specific block size requires the application to send specific length of application data.

- 5: if PAD and NH are set to removed: Diet-ESP does consider neither the Next Header field nor the Pad Length field nor the Padding Field. This is valid as long as LPD mod [M] = 0. If M = 1 as it is the case for AES-CTR this equation is always true. On the other hand the use of specific block size requires the application to send specific length of application data.

- 6: Encrypt the Diet-ESP payload.

- 7: Add ESP header.

- 8: Generate and add Diet-ESP ICV.

- 9: Compress ESP header.

### B.5. Inbound Packet processing

Decryption is for performed the other way around.

After SAD lookup, authenticating and decrypting the Diet-ESP payload the original packet is rebuild as follows:

- 1: Decompress ESP header.

- 2: Generate Diet-ESP ICV and check ICV send in the packet.

- 3: Check anti-replay

- 4: Remove compressed header.

- 5: Encrypt the Diet-ESP payload.

- 6: if PAD and NH are set to removed: Diet-ESP does consider neither the Next Header field nor the Pad Length field nor the Padding Field. The Next Header field of the IP packet is set to the protocol defined for incoming traffic within the Traffic Selector of the SA. Because there is no Padding it is disregarded.

- 7: if PAD is set to removed and NH is set to present: Diet-ESP considers the Next Header but do not consider the Pad Length field or the Padding Field. The Next Header field of the IP packet is set to the value within the Diet-ESP trailer.

- 8: if PAD is set to present and NH is set to removed: Diet-ESP considers the Pad Length field but removes the Next Header field. The Next Header field of the IP packet is set to the protocol defined for incoming traffic within the Traffic Selector of the SA. The Pad Length field is read and the Padding is removed from the Data Payload which results the original Data Payload.

- 9: if PAD and NH are set to present: Diet-ESP considers both fields Pad Length and Next Header. The Next Header field of the IP packet is set to the value within the Diet-ESP trailer. The Pad Length field is read and the Padding is removed from the Data Payload which results the original Data Payload.

- 10: Decompress the headers inside the ESP payload.

## Appendix C. Interaction with other Compression Protocols

Diet-ESP exclusively defines compression for the ESP protocol as well as the ESP payload. It does not consider compression of the IP protocol. ROHC or 6LoWPAN may be used by a sensor to compress the IP (resp. IPv6) header. Since compression usually occurs between the MAC and IP layers, there are no expected complications with this family of compression protocols.

### C.1. 6LoWPAN

Diet-ESP smoothly interacts with 6LoWPAN. Every 6LoWPAN compression header (NHC_EH) has an NH bit. This one is set to 1 if the following header is compressed with 6LoWPAN. Similarly, the NH bit is set to 0 if the following header is not compressed with 6LowPAN. Thus, interactions between 6LowPAN and Diet-ESP considers two case: 1) NH set to 0: 6LowPAN indicates the Diet-ESP payload is not compressed and 2) NH set to 1: 6LowPAN indicates the Diet-ESP payload is compressed.

Suppose 6LowPAN indicates the Next Header ESP is not compressed by 6LowPAN. If the peers have agreed to use Diet-ESP, then the ESP layer on each peers receives the expected Diet-ESP packet. Diet-ESP is fully compatible with 6LowPAN ESP compression disabled.

Suppose 6LowPAN indicates the Next Header ESP is compressed by 6LowPAN. ESP compression with 6LowPAN [I-D.raza-6lowpan-ipsec] considers the compression of the ESP Header, that is to say the compression of the SPI and SN fields. As a result 6LowPAN compression expects a 4 byte SPI and a 4 byte SN from the ESP layer. Similarly 6LowPAN decompression provides a 4 byte SPI and a 4 byte SN to the ESP layer. If the peers have agreed to use Diet-ESP and one of them uses 6LowPAN ESP compression, then the Diet-ESP MUST use SPI SIZE and the SN SIZE MUST be set to 4 bytes.

### C.2. ROHC

ROHC and ROHCoverIPsec have been used to describe Diet-ESP. This means the ROHC and ROHCoverIPsec concepts and terminology have been used to describe Diet-ESP. In that sens Diet-ESP is compatible with the ROHC and ROHCoverIPsec framework. The remaining of the section describes how Diet-ESP interacts with ROHC and ROHCoverIPsec profiles and payloads.

ROHC compress packets between the MAC and the IP layer. Compression can only be performed over non encrypted packets. As a results, this section considers the case of an ESP encrypted packet and an ESP non encrypted packet.

For encrypted ESP packet, ROHC profiles that enable ESP compression (e.g. profile 0x0003 and 0x1003) compresses only the ESP Header and the IP header. To enable ROHC compression a Diet-ESP packet MUST present an similar header as the ESP Header, that is a 4 byte SPI and a 4 byte SN. This is accomplished by setting SPI_SIZE = 4 and SN_SIZE = 4 in the Diet-ESP Context. Reversely, if the Diet-ESP packet presents a 4 byte SPI and a 4 byte SN, ROHC can proceed to the compression. Note that Diet-ESP does not consider the IP header, then ESP and Diet-ESP are encrypted, thus ROHC can hardly make the difference between Diet-ESP and ESP packets. For encrypted packets, the only difference at the MAC layer might be the alignment.

For non encrypted ESP packet, ROHC MAY proceed to the compression of different fields of ESP and other layers, as the payload appears in clear. ROHC compressor are unlikely to deal with ESP fields compressed by Diet-ESP. As a result, it is recommended not to combine Diet-ESP and ROHC ESP compression with non encrypted ESP packets.

### C.3. ROHCoverIPsec and 6LoWPANoverIPsec

ROHC or 6LoWPAN are not able to compress the ESP payload, as long as it is encrypted. Diet-ESP describes how to compress the ESP-Trailer, which is part of the encrypted payload can be compressed. 6LowPANoverIPsec (section 2 of [I-D.raza-6lowpan-ipsec]) and ROHCoverIPsec define the compression of the ESP payload, more specifically the upper layer headers (e.g. IP header or Transport layer header). These protocols need a second, modified ESP stack in order to make the payload compression possible. Then the packets with compressed payload are forwarded to this second ESP stack which can compress or decompress the payload.

Diet-ESP and its extensions also needs a modified ESP stack in order to perform the compression of ESP payload possible. In addition, fields that are subject to compression are most likely to be the same with Diet6ESP and 6LowPANoverIPsec and/or ROHCoverIPsec. Therefore, if a device implements Diet-ESP and 6LowPANoverIPsec and/or ROHCoverIPsec the developer needs to define an order the various frameworks perform the compression. Currently this order has not been defined, and Diet-ESP is unlikely to be compatible with 6LowPANoverIPsec and/or ROHCoverIPsec. Integration of Diet-ESP and 6LowPANoverIPsec and/or ROHCoverIPsec has not been considered in the current document as Diet-ESP has been designed to avoid implementations of 6LowPANoverIPsec and/or ROHCoverIPsec frameworks to be implemented into the devices. Diet-ESP has been designed to be more lightweight than 6LowPANoverIPsec and/or ROHCoverIPsec by avoiding negotiations between compressors and decompressors.

## Appendix D. Diet-ESP and Requirements

[I-D.mglt-6lo-diet-esp-requirements] lists the requirements for Diet-ESP. This section position Diet-ESP described in this document toward these requirements.

R1: Diet-ESP is completly based on IPsec/ESP and as such benefits from the standard IPsec security.

R2: Diet-ESP does not introduces vulnerabilities to IPsec/ESP. The only difference is that compression results in sending less bytes on the wire. In return the bytes sent over the wire are decompress to feed the IPsec stack with the appropriated bytes. Compression MAY require a non standard IPsec/ESP implementation, as some fields may have been removed. This fields are packet descriptors (like padding, length...), and are not related to the security of the standard IPsec/ESP.

R3: Diet-ESP is fully derived from IPsec/ESP ROHC and ROHCoverIPsec and as such relies on the security of these protocols.

R4: Diet-ESP is able to handle alignments of 8, 16, 32 and 64 bits.

R5: is not in the scope of Diet-ESP. Announcement of the Byte-Alignment should be performed by IKEv2.

R6: Diet-ESP does not modify how encryption occurs. It only changes the encrypted payload, which is one of the parameters for the encryption function. Therefore Diet-ESP is able to work with any encryption defined in [RFC7321] which also includes AES-CCM [RFC4309].

Combined Mode algorithm (e.g. AES-CCM, AES-GCM) have an additional parameter, called Addition Authentication Data (AAD). This AAD requires the uncompressed ESP header that is to say the full SPI and SN. These parameters are not removed by Diet-ESP. There are well known by the two peer. The ESP Header MUST be uncompressed before proceeding to encryption/ decryption.

R7: Diet-ESP can remove all static and compress fields from the protocol.

R8: The inner payload compression mechanisms are not defined in this document. This aspect is the purpose of [draft-mglt-inner-compression]

R9: Diet-ESP compressed packet fields are always a number of bytes -- that is Diet-ESP do not result in compressed fields that are not expressed in a natural number of bytes.

R10: Diet-ESP allows the developer define the maximum compression within the Diet-ESP context. The way the agreement is done, is out of scope of this document and is described in [draft-mglt-diet-esp-ikev2].

R11: Each field in the packet can be compressed separately, which provides high flexibility.

R12: Since Diet-ESP does not propose compression method flexibility. The proposed methods are generic enough and there is not advantage for this flexibility and so it does not seems appropriated for Diet-ESP.

R13: Each Diet-ESP client can have his own set of supported contexts. The negotiation is out of scope of this document and described in [draft-mglt-diet-esp-ikev2].

R14: Diet-ESP adds small complexity to Standard ESP, like described in Appendix B. In- and Outbound packet procession is straight-forward, like shown in Appendix B.5 and Appendix B.5. Appendix A provides a implementation guideline for a minimal use case. This one can be ported to any other use case.

R15: Diet-ESP is easy to configure and provides a default-context if a developer does not want to dive into the details of Diet-ESP.

R16: Diet-ESP can interact with 6LoWPAN and ROHC IP compression, but SHOULD be able to interact with all future compression applying after the IP layer as well.

R17: Compatibility with Standard ESP 1: Diet-ESP can be implemented instead of, nearby or like an add-on to an existing Standard ESP implementation.

R18: Compatibility with Standard ESP 2: Diet-ESP is able to work without compression and works with 32 and 64 bits alignment, which makes it compatible with Standard ESP.

## Appendix E. Document Change Log

[draft-mglt-6lo-diet-esp-00.txt]:
Changing affiliation

[draft-mglt-6lo-diet-esp-00.txt]:

Updating references

[draft-mglt-ipsecme-diet-esp-01.txt]:
Diet ESP described in the ROHC framework
ESP is not modified.

[draft-mglt-ipsecme-diet-esp-00.txt]:
NAT consideration added.
Comparison actualized to new Version of 6LoWPAN ESP.

[draft-mglt-dice-diet-esp-00.txt]: First version published.

## Authors' Addresses

Daniel Migault
Ericsson
8400 boulevard Decarie
Montreal, QC H4P 2N2
Canada

Email: daniel.migault@ericsson.com

Tobias Guggemos
LMU Munich
Oettingenstr. 67
80538 Muenchen, Bavaria
Germany

Email: guggemos@mnm-team.org

Carsten Bormann
Universitaet Bremen TZI
Postfach 330440
Bremen D-28359
Germany

Phone: +49-421-218-63921
Email: cabo@tzi.org

6lo
Internet-Draft
Intended status: Standards Track
Expires: January 9, 2017

D. Migault
Ericsson
T. Guggemos
LMU Munich
C. Bormann
Universitaet Bremen TZI

July 8, 2016

# Requirements for Diet-ESP the IPsec/ESP protocol for IoT
## draft-mglt-6lo-diet-esp-requirements-02.txt

Abstract

IPsec/ESP is used to secure end-to-end communications. This document lists the requirements Diet-ESP should meet to design IPsec/ESP for IoT.

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 http://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 9, 2017.

Table of Contents



## 1. Requirements notation

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

## 2. Introduction

IoT devices can carry all kind of small applications and some of them require a secure communication. They can be life critical devices (like a fire alarm), security critical devices (like home theft alarms) and home automation devices. Smart grid is one application where supplied electricity is based on information provided by each home. Similarly, home temperature might be determined by servo-controls based on information provided by temperature sensors.

Using IPsec [RFC4301] in the IoT world provides some advantages, such as:

- IPsec secures application communications transparently as security is handled at the IP layer. As such, applications do not need to be modified to be secured.

- IPsec does not depend on the transport layer. As a result, the security framework remains the same for all transport protocols, like UDP or TCP.

- IPsec is well designed for sleeping nodes as there are no sessions.

- IPsec defines security rules for the whole device, which outsource the device security to a designated area. Therefore IPsec can be seen like a tiny firewall securing all communication for an IoT device.

IPsec is mostly implemented in the kernel, whereas application are in the user space. This is often considered as a disadvantage for IPsec. However, as there are no real distinctions between these two spaces in IoT and that IoT devices are mostly designed to a specific and unique task, this may not be an issue anymore.

IoT constraints have not been considered in the early design of IPsec. In fact IPsec has mainly been designed to secure infrastructure. This document describes the requirements of Diet-ESP, the declination of IPsec/ESP for IoT, enabling optimized IPsec/ESP for the IoT.

## 3. Terminology

- IoT: Internet of Things

## 4. Protocol Design

Diet-ESP is based on IPsec/ESP and is adapted for IoT. Adaptation to IoT scenarios must not be at the expense of security, and the security evaluation of Diet-ESP should benefit as far as possible from the long experience of already existing protocols. As a result the protocol design requirements for Diet-ESP are as follows:

R1: Diet-ESP MUST benefit from the IPsec/ESP security.

R2: Diet-ESP MUST NOT introduce vulnerabilities over IPsec/ESP. This means that at some points IPsec/ESP is implemented. A foreseen way to reach that goal is to associate IPsec/ESP with compressors/decompressors.

R3: Diet-ESP SHOULD rely on existing protocol or frameworks.

## 5. Byte-Alignment

IP extension headers MUST have 32 bit Byte-Alignment in IPv4 (section 3.1 of [RFC0791] - Padding description) and a 64 bit Byte-Alignment in IPv6 (section 4 of [RFC2460]). As ESP [RFC4303] is such an extension header, padding is mandatory to meet the alignment constraint. This alignment is mostly caused by compiler and OS requirements dealing with a 32 or 64 Bit processor. In the world of IoT, processors and compilers are highly specialized and alignment is often not necessary 32 Bit, but 16 or 8 bit. As a result, the byte-alignment requirement is as follows:

R4: Diet-ESP MUST support Byte-Alignment that are different from 32 bits or 64 bits to prevent unnecessary padding.

R5: Each peer MUST be able to advertise and negotiate the Byte-Alignment, used for Diet-ESP. This could be done for example during the IKEv2 exchange.

## 6. Crypto-Suites

IEEE 802.15.4 defines AES-CCM*, that is AES-CTR and CBC-MAC, for link layer security with upper layer key-management. Therefore it is usually supported by hardware acceleration. This leads to the following crypto-suite requirement:

R6: Diet-ESP MUST support AES-CCM and MUST be able to take advantage of AES-CCM hardware acceleration. Diet-ESP MAY support other modes.

## 7. Compression

Sending data is very expensive regarding to power consumption, as illustrated in Appendix A. Compression can be performed at different layers. An encrypted ESP packet is an ESP Clear Text Data encrypted and eventually concatenated with the Initialization Vector IV to form an Encrypted Data Payload. This encrypted Data Payload is then placed between an ESP Header and an ESP Trailer. Eventually, this packet is authenticated with an ICV appended to ESP Trailer. Compression can be performed at the ESP layer that is to say for the fields of the ESP Header, ESP Trailer and the ICV. In addition, ESP Clear Text Data may also be compressed with non ESP mechanisms like ROHC [RFC3095], [RFC5225] for example, resulting in a smaller payload to be encrypted. If ESP is using encryption, these mechanisms MUST be performed over the ESP Clear Text Data before the ESP/Diet-ESP processing as missing of encrypted fields make decryption harder. As a result, compression requirements are as follows:

R7: Diet-ESP MUST be able to compress/remove all static ESP fields (SPI, Next Header) as well as the other fields SN, Padding, Pad Length or ICV.

R8: Diet-ESP SHOULD also allow compression mechanisms before the IPsec/ESP processing.

R9: Diet-ESP SHOULD NOT allow compressed fields, not aligned to 1 byte in order to prevent alignment complexity. In other words, Diet-ESP do not consider finer granularity than the byte.

## 8. Flexibility

Diet-ESP can compress some of the ESP fields as Diet-ESP is optimized for IoT. Which field may be compressed or not, depends on the scenario and current and future scenarios cannot been foreseen. As a result, the flexibility requirements are as follows:

R10: Diet-ESP MUST be able to compress any field independently from another.

R11: Diet-ESP SHOULD provide different ways to compress a single field, so the most appropriated way can be agreed between the peers.

R12: Each peer MUST be able to announce and negotiate the different compressed fields as well as the used method.

In fact Diet-ESP and ESP differs in the following point: ESP has been designed so that any ESP secured communication so any device is able to communicate with another. This means that ESP has been designed to work for large Security Gateway under thousands of connections, as well as devices with a single ESP communication. Because, ESP has been designed not to introduce any protocol limitations, counters and identifiers may become over-sized in an IoT context.

## 9. Code Complexity

IoT devices have limited space for memory and storage, which leads to the following requirement.

R13: Diet-ESP MUST be able to be implemented with minimal complexity. More especially, Diet-ESP MUST consider small implementation that implement only a subset of all Diet-ESP capabilities without requiring involving standard ESP, specific compressors and de-compressors.

## 10. Usability

Application Developer usually do not want to take care about the underlying protocols and security. In addition, the security configuration should remain feasible by a standard software developer. The usability requirements regarding Diet-ESP are as follows:

R14: Diet-ESP MUST remain independent from the application.

R15: Diet-ESP MUST detail for each field how compression impacts the security of the device. Although the creation of profiles is out of scope of Diet-ESP, it is expected that profiles may be defined latter by the usage.

## 11. Compatibility with IP compression Protocols

There are different protocols providing IP layer compression for constraint devices like IoT (6LoWPAN [RFC6282] ) or Mobile Devices (ROHC). The requirements regarding interactions of Diet-ESP and additional compression protocols are as follows:

R16: Diet-ESP MUST be able to interact with IP compression protocols. More especially, this means that a Diet-ESP packet MUST be able to be sent in a ROHC or a 6LowPAN packet. Diet-ESP document should explicitly detail how this can be achieved.

R17: Diet-ESP MUST also detail how compression of layers above IP with ROHC or 6LowPAN is compatible with Diet-ESP.

## 12. Compatibility with Standard ESP

IPsec/ESP is widely deployed by different vendors on different machines. IoT devices MAY have to communicate with Standard ESP implementations. The ESP compatibility requirements is as follows:

R18: Diet-ESP MUST be able to communicate with Standard ESP.

## 13. IANA Considerations

There are no IANA consideration for this document.

## 14. Security Considerations

Security Considerations have been expressed as one of the requirement.

## 15. Acknowledgment

The current work on Diet-ESP results from exchange and cooperation between Orange, Ludwig-Maximilians-Universitaet Munich, Universite Pierre et Marie Curie. We thank Daniel Palomares and Carsten Bormann for their useful remarks, comments and guidances on the design. We thank Sylvain Killian for implementing an open source Diet-ESP on Contiki and testing it on the FIT IoT-LAB [fit-iot-lab] funded by the French Ministry of Higher Education and Research. We thank the IoT-Lab Team and the INRIA for maintaining the FIT IoT-LAB platform and for providing feed backs in an efficient way.

## 16. Normative References

[fit-iot-lab] "Future Internet of Things (FIT) IoT-LAB", <https://www.iot-lab.info>.

[RFC0791] Postel, J., "Internet Protocol", STD 5, RFC 791, DOI 10.17487/RFC0791, September 1981, <http://www.rfc-editor.org/info/rfc791>.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <http://www.rfc-editor.org/info/rfc2119>.

[RFC2460] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", RFC 2460, DOI 10.17487/RFC2460, December 1998, <http://www.rfc-editor.org/info/rfc2460>.

[RFC3095] Bormann, C., Burmeister, C., Degermark, M., Fukushima, H., Hannu, H., Jonsson, L-E., Hakenberg, R., Koren, T., Le, K., Liu, Z., Martensson, A., Miyazaki, A., Svanbro, K., Wiebke, T., Yoshimura, T., and H. Zheng, "RObust Header Compression (ROHC): Framework and four profiles: RTP, UDP, ESP, and uncompressed", RFC 3095, DOI 10.17487/RFC3095, July 2001, <http://www.rfc-editor.org/info/rfc3095>.

[RFC4301] Kent, S. and K. Seo, "Security Architecture for the Internet Protocol", RFC 4301, DOI 10.17487/RFC4301, December 2005, <http://www.rfc-editor.org/info/rfc4301>.

[RFC4303] Kent, S., "IP Encapsulating Security Payload (ESP)", RFC 4303, DOI 10.17487/RFC4303, December 2005, <http://www.rfc-editor.org/info/rfc4303>.

[RFC5225] Pelletier, G. and K. Sandlund, "RObust Header Compression Version 2 (ROHCv2): Profiles for RTP, UDP, IP, ESP and UDP-Lite", RFC 5225, DOI 10.17487/RFC5225, April 2008, <http://www.rfc-editor.org/info/rfc5225>.

[RFC6282] Hui, J., Ed. and P. Thubert, "Compression Format for IPv6 Datagrams over IEEE 802.15.4-Based Networks", RFC 6282, DOI 10.17487/RFC6282, September 2011, <http://www.rfc-editor.org/info/rfc6282>.

## Appendix A. Power Consumption Example

IoT devices are often installed once and left untouched for a couple of years. Furthermore they often do not have a power supply wherefore they have to be fueled by a battery. This battery may have a limited capacity and maybe not replaceable. Therefore, power can be a limited resource in the world of IoT. Table 1 and Table 2 shows the costs for transmitting data and computation

Note these data are mentioned here with an illustrative purpose, for our motivations. These data may vary from one device to another, and may change over time.

| | power consumption |
|---|---|
| low-power radios < 10mW | (100nJ - 1uJ) / bit |

Table 1: Power consumption for data transmission.

| | power consumption |
|---|---|
| energy-efficient microprocessors | 0.5nJ / instruction |
| high-performance microprocessors | 200nJ / instruction |

Table 2: Power consumption for computation.

From these tables, sending 1 bit costs as much as 10-100 instructions in the CPU. Therefore there is a high interest to reduce the number of bits sent on the wire, even if it generates costs for computation.

## Appendix B. Document Change Log

[draft-mglt-6lo-diet-esp-requirements-01.txt]: Changing affiliation.

[draft-mglt-6lo-diet-esp-requirements-00.txt]: Published: Minor re-wordings

[draft-mglt-ipsecme-diet-ipsec-requirements-00.txt]: First version published.

## Authors' Addresses

Daniel Migault
Ericsson
8400 boulevard Decarie
Montreal, QC H4P 2N2
Canada

Email: daniel.migault@ericsson.com

Tobias Guggemos
LMU Munich
Am Osteroesch 9
87637 Seeg, Bavaria
Germany

Email: tobias.guggemos@gmail.com

Carsten Bormann
Universitaet Bremen TZI
Postfach 330440
Bremen D-28359
Germany

Phone: +49-421-218-63921
Email: cabo@tzi.org

6lo
Internet-Draft
Intended status: Standards Track
Expires: September 13, 2017

AR. Sangi
M. Chen
Huawei Technologies
C. Perkins
Futurewei

March 12, 2017

# Designating 6LBR for IID Assignment
# draft-rashid-6lo-iid-assignment-03

## Abstract

In IPv6 Stateless Address Autoconfiguration (SLAAC), randomizing the interface identifier (IID) is a common practice to promote privacy. If there are a very large number of nodes, as has been discussed in several use cases, the effect will to proportionately increase the number of IIDs. A duplicate address detection (DAD) cycle is needed for each configured IID, introducing more and more overhead into the network. Each failed DAD requires the initiating node to regenerate a new IID and undergo the DAD cycle again. This document proposes an optimized approach when higher privacy is required in a given network by allowing a 6LBR (6LoWPAN Border Router) to provide a unique IID, avoiding any potential duplication. Such practice also prevents failure of time-critical applications, by enabling 6LBR to provide a unique IID, in case of address collision.

Further improvements are proposed to enable multiple concurrent DAD cycles, and to return the randomized IID from 6LBR to 6LN in a space-efficient manner.

## 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 http://datatracker.ietf.org/drafts/current/.

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

This Internet-Draft will expire on September 13, 2017.

Table of Contents



## 1. Introduction

IPv6 addresses in SLAAC are formed by concatenating a network prefix, acquired from Router Advertisement (RA) messages, with a locally generated IID [RFC4862], [RFC2464]. Since the best method for generating IIDs varies depending on the network, none of the proposed mechanisms [RFC4941],[RFC7217] is considered a default mechanism. Using neighbour discovery (ND), the uniqueness of newly a generated IID is verified [RFC6775]. 6LBR performs DAD, and replies with a status. A failed DAD would require the initiating 6LN (6LoWPAN node) to regenerate an IID and wait for another DAD cycle, until the 6LN successfully registers a unique address [RFC6775].

A locally generated IID can be derived either from an embedded IEEE identifier [RFC4941], or randomly (based on a few variables) [RFC7217]. Since MAC reuse is unfortunately far more common than usually assumed [RFC7217][MAC-Duplication], IIDs derived from MAC address are likely to cause more than the expected number of DAD failures. As soon as the 6LN generates an IID, it sends the NS (Neighbor Solicitation) message to 6LR (LLN Router). Then 6LR proceeds to send an ICMPv6 based DAR (Duplicate Address Request) message to 6LBR. An LN sends out a NS after checking its local cache for duplication; before proceeding with DAR, the 6LR also protects against address duplication within a locally maintained Neighbor Cache Entry (NCE) [RFC7217].

Use cases including huge numbers of nodes and vast scale networks are discussed in [RFC5548], [RFC5827]. The use of arbitrary IIDs can resolve privacy concerns for a participating node, but a simple NS intended to be targeted to a small group of nodes can pollute a great deal of wireless bandwidth [I-D.vyncke-6man-mcast-not-efficient]. Multicast NS and NA are much more frequent in large scale radio environment with mobile devices [I-D.ietf-6lo-backbone-router]. Since the IIDs may be sporadically changed for privacy, the probability further increases that a duplicate IIDs would result in DAD failure and repeated DAD cycles.

On the other hand, waiting for 6LN to regenerate another IID due to a failed DAD might lead to failure of a time-critical application.

Address assignment can also be done using DNS (Domain Name Server), but doing so typically requires multicast traffic and introduces more control overhead. Unlike DNS, the 6LoWPAN ND works on layer 2 and our proposed mechanism implicitly provides assistance to the DAD process.

This document describes improvements to 6LoWPAN ND which enable 6LBR to grant a unique IID for failed DAD, to enable multiple concurrent DAD cycles, and to return an IID to 6LN in a space-efficient manner.

## 2. Terminology

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 [RFC2119]. This document uses terminology from [RFC6775], [RFC2464], [RFC8064], and [RFC7721].

SLLAO: Stateless Link-Local Address Option

RID: Random IDentifier

PRF: Pseudo Random Function

IID: Interface IDentifier

This document also uses the following terms:

EARO: Extended Address Registration Option

EDAR: Extended Duplicate Address Request

EDAC: Extended Duplicate Address Confirmation

LSB: Least Significant Bit

## 3. Likelihood of Address Collision

The following observations have motivated the design of this proposal:

- Manufacturer may not follow a fine grained randomness in MAC addresses.
- MAC addresses shorter than 64 bits are used in various constrained technologies.
- The frequency of an IID being changed depends on the degree of privacy that a particular application requires.
- Depending upon the method by which an IID is generated using MAC address, or with shorter MAC addresses, address collisions may become much more likely.

## 4. IID Assignment by 6LBR

MAC driven IIDs [RFC2464] reduce or eliminate the need for DAD, but in practice such IID generation is discouraged ([RFC8064], [RFC7721]), as common privacy concerns still persist, for instance:

- Network activity correlation,
- Location tracking,
- Address scanning, and
- Device-specific vulnerability exploitation.

Multiple approaches are proposed to suit different network constraints. The mechanisms specified in [RFC4941], which are mainly

based on MAC address or an appropriate simple random number generation algorithm can also be used to generate IID.

Considering the scalability of a network and enabling 6LBR to randomize an IID, the method for IID generation specified in [RFC7217] SHOULD be used because this method is appropriate to support periodically changing IIDs.

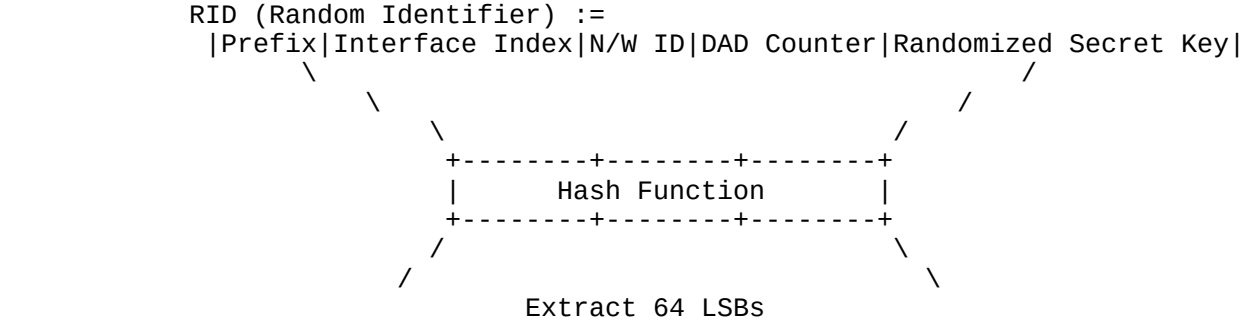



Figure 1: Using RFC7217 to generate IID

If DAD fails, the 6LBR will use public values for Prefix, Interface Index, and Network ID; the remaining two variables (DAD Counter, Randomized Secret Key) are local values. Neighbor solicitation using link-local address cannot be avoided, but only the newly generated IID needs to be forwarded to the LN.

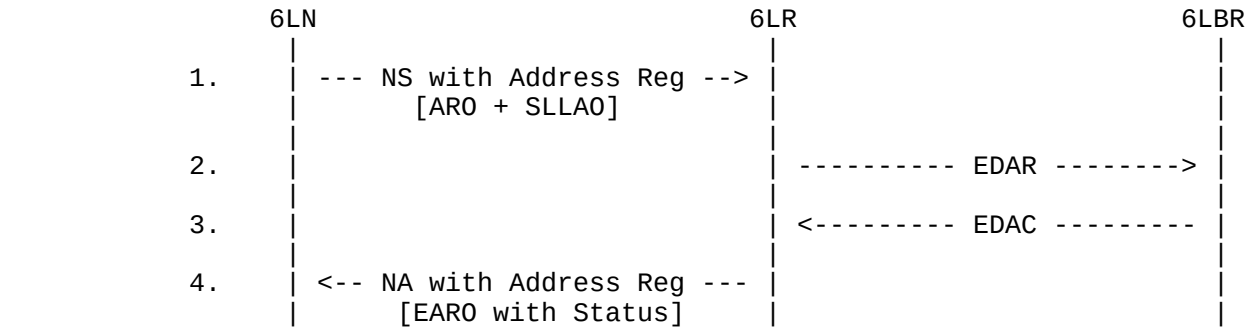



Figure 2: DAD cycle when 6LBR generates an IID

The approach in this draft is reactive rather than proactive; 6LBR only replies with a locally generated IID when DAD fails.

Appendix A [RFC7217] states that a Net_Iface parameter can either be:

o Interface Index,

- o Interface Name,
- o Link-Layer Address, or
- o Logical Network Service Identity.

EUI-64 of 6LN would be sent to 6LBR via 6LR within EARO and using that, a Link-Layer Address can be derived at 6LBR to input in PRF. For multiple interfaces, DAD_counter would be incremented as soon as the collision occurs.

### 4.1. Advantages of proposed algorithm

By reference to the algorithm in [RFC7217], the resulting IID offers the following advantages:

- o For a given interface, same prefix and subnet would always result in same IID,
- o It would always be a different IID generated when a different prefix is used, and
- o The DAD_Counter parameter is incremented in case of address collision, so that the resulting address would be different than the previous address.

### 4.2. Extended Duplicate Address Request (EDAR)

The Prefix is the same throughout each LoWPAN network. This draft uses that feature to reduce the size of the DAR:

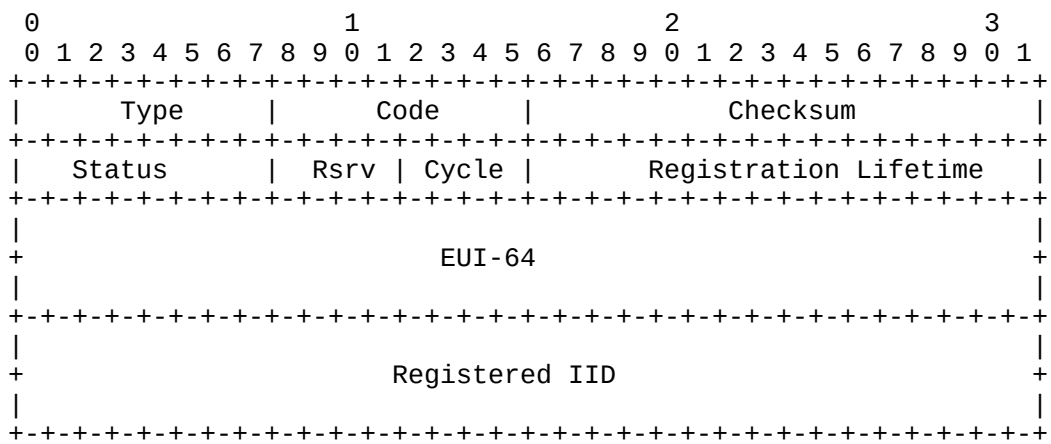



Figure 3: Extended Duplicate Address Request

The fields are similar to DAR in [RFC6775] except:

- Type: 159 (TBD)
- Cycle: 4 out of 8 reserved bits to identify the DAD cycle between given 6LR and 6LBR. The reference is used later by 6LR to extract IID provided by 6LBR.
- Unlike the DAR, the Registered IID (64 bit) is returned instead of Registered Address (128 bit).

### 4.3. Extended Duplicate Address Confirmation (EDAC)

EDAC reduces the space needed for returning the EUI-64:

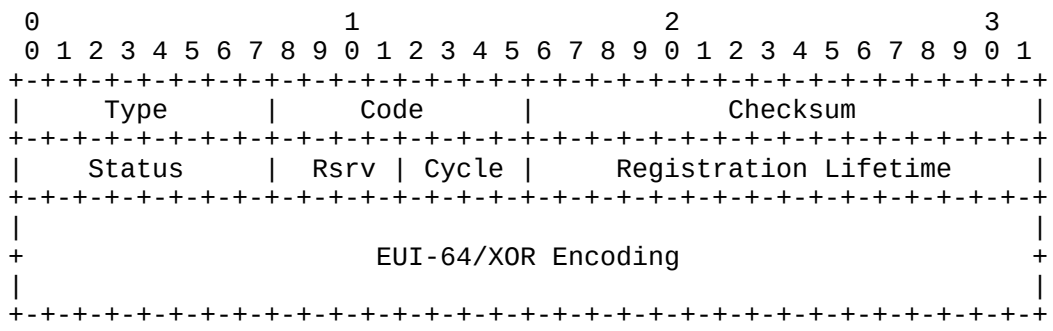



Figure 4: Extended Duplicate Address Confirmation

The fields are similar to DAC in [RFC6775] except:

- Type: 160 (TBD)
- Cycle: 4 out of 8 reserved bits identify the DAD cycle between the 6LR and 6LBR. These bits are used later by 6LR to extract the IID supplied by 6LBR.
- In case of a failed DAD, a 6LBR-generated IID is encoded using XOR with EUI-64; otherwise the same EUI-64 occupies the 64 bits.

### 4.4. Extended Address Registration Option

ARO and EARO can ONLY be initiated by host and 6LR, respectively. [RFC6775] expects the reply of a host initiated ARO from 6LR with the same ARO except for changing the status bit to indicate the duplication detection. EARO is introduced in this document; 6LR can send out this option if it receives EDAC instead of DAC from 6LBR.

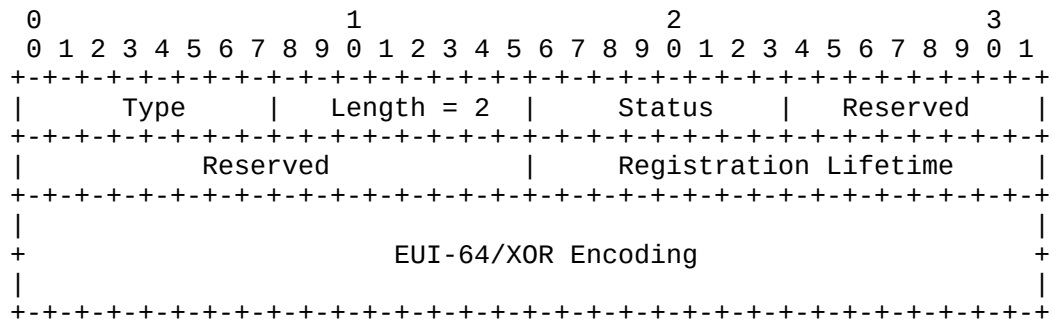



Figure 5: Extended Address Registration Option

- The fields are similar to ARO in [RFC6775] except:
- Type: 36 (TBD)
- EUI-64/XOR Encoding: a 64 bit IID generated by 6LBR is XOR'ed with EUI-64.

## 5. Multiple DAD cycles

In [RFC6775], 6LN is expected to generate an IID; so 6LR only acts on the first unique IID claim and silently discards any later claims for the same IID. In contrast, this document enables 6LBR to assign a unique IID in case of a duplicate IID claim by 6LR. For this purpose, a "Cycle" field is introduced to enable multiple concurrent DAD cycles that will be helpful for large-scale networks [RFC5548]. At 6LN, this "Cycle" field is also used when extracting both IID and EUI-64 that are XOR'ed by 6LBR. See Figure 3 and Figure 4 for the format of the Cycle field.

## 6. XOR Encoding

Each iteration of DAR and DAC [RFC6775] carries the entire 128 bit Registered Address during the DAD routine, even though the network Prefix is the same throughout each LoWPAN. This document enables eliding the network prefix part of the Registered Address as well in EDAC and EARO using simple XOR encoding. The encoded 64 bit field carries EUI-64 and randomized IID. See Figure 4 and Figure 5 for the format of the EUI-64/XOR encoding.

Under the proposed arrangement, 6LBR would only encode values, 6LN would only extract values and 6LR would do both.

At 6LR before sending EDAR to 6LBR:

o 6LR would use the 4 out of 8 Reserved "Cycle" bits of EDAR to keep track of multiple DAD cycles. These iterations are recorded at 6LR and that information is used to extract IID/EUI-64 from EDAC to be forwarded to the appropriate 6LN.

At 6LBR before sending to 6LR:

o If Status = 0 (Success), then 6LBR returns EDAC using all the values as received from EDAR.

o If Status = 1 (Duplicate), then 6LBR generates IID and XORs it with EUI-64 to return in the EDAC to 6LR.

At 6LR before sending to 6LN:

o If Status = 0 (Success) then keep the claimed address of 6LN as Destination Address for ARO to 6LN.

o If Status = 1 (Duplicate), then match the "Cycle" bits of EDAC to extract (using XOR) the EUI-64 address and use the extracted address as the Destination Address for EARO to 6LN.

Finally, at 6LN:

o If Status = 0 (Success), 6LN starts using the address that it claimed.

o If Status = 1 (Duplicate) then 6LN XORs the received EUI-64 address with its claimed EUI-64, which results in the newly generated IID sent by 6LBR.

## 7. IANA Considerations

### 7.1. EDAR and EDAC Messages, and EARO Option

The document requires two new ICMPv6 type numbers under the subregistry 'ICMPv6 "type" Numbers':

o Extended Duplicate Address Request (159)

o Extended Duplicate Address Confirmation (160)

This document requires a new ND option type under the subregistry "IPv6 Neighbor Discovery Option Formats":

o Extended Address Registration Option (36)

### 7.2. Additions to Status Field

One new value is required for the "Address Registration Option Status Values" sub-registry under the "IPv6 Neighbor Discovery Option Formats":

| Status | Description |
|---|---|
| 0 | Success |
| 1 | Duplicate Address |
| 2 | Neighbor Cache Full |
| 3 | 6LBR generated IID |
| 4-255 | Allocated using Standards Action [RFC5226] |

Addition to Status bits

## 8. Security Considerations

TBD

## 9. References

### 9.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <http://www.rfc-editor.org/info/rfc2119>.

[RFC2464] Crawford, M., "Transmission of IPv6 Packets over Ethernet Networks", RFC 2464, DOI 10.17487/RFC2464, December 1998, <http://www.rfc-editor.org/info/rfc2464>.

[RFC4862] Thomson, S., Narten, T., and T. Jinmei, "IPv6 Stateless Address Autoconfiguration", RFC 4862, DOI 10.17487/RFC4862, September 2007, <http://www.rfc-editor.org/info/rfc4862>.

[RFC4941] Narten, T., Draves, R., and S. Krishnan, "Privacy Extensions for Stateless Address Autoconfiguration in IPv6", RFC 4941, DOI 10.17487/RFC4941, September 2007, <http://www.rfc-editor.org/info/rfc4941>.

[RFC6775] Shelby, Z., Ed., Chakrabarti, S., Nordmark, E., and C. Bormann, "Neighbor Discovery Optimization for IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs)", RFC 6775, DOI 10.17487/RFC6775, November 2012, <http://www.rfc-editor.org/info/rfc6775>.

[RFC7217] Gont, F., "A Method for Generating Semantically Opaque Interface Identifiers with IPv6 Stateless Address Autoconfiguration (SLAAC)", RFC 7217, DOI 10.17487/RFC7217, April 2014, <http://www.rfc-editor.org/info/rfc7217>.

## 9.2. Informative References

[I-D.ietf-6lo-backbone-router] Thubert, P., "IPv6 Backbone Router", draft-ietf-6lo-backbone-router-03 (work in progress), January 2017.

[I-D.vyncke-6man-mcast-not-efficient] Vyncke, E., Thubert, P., Levy-Abegnoli, E., and A. Yourtchenko, "Why Network-Layer Multicast is Not Always Efficient At Datalink Layer", draft-vyncke-6man-mcast-not-efficient-01 (work in progress), February 2014.

[MAC-Duplication] Moore, HD., "The Wild West", September 2012, <https://speakerdeck.com/hdm/derbycon-2012-the-wild-west>.

[RFC5548] Dohler, M., Ed., Watteyne, T., Ed., Winter, T., Ed., and D. Barthel, Ed., "Routing Requirements for Urban Low-Power and Lossy Networks", RFC 5548, DOI 10.17487/RFC5548, May 2009, <http://www.rfc-editor.org/info/rfc5548>.

[RFC5827] Allman, M., Avrachenkov, K., Ayesta, U., Blanton, J., and P. Hurtig, "Early Retransmit for TCP and Stream Control Transmission Protocol (SCTP)", RFC 5827, DOI 10.17487/RFC5827, May 2010, <http://www.rfc-editor.org/info/rfc5827>.

[RFC7721] Cooper, A., Gont, F., and D. Thaler, "Security and Privacy Considerations for IPv6 Address Generation Mechanisms", RFC 7721, DOI 10.17487/RFC7721, March 2016, <http://www.rfc-editor.org/info/rfc7721>.

[RFC8064] Gont, F., Cooper, A., Thaler, D., and W. Liu, "Recommendation on Stable IPv6 Interface Identifiers", RFC 8064, DOI 10.17487/RFC8064, February 2017, <http://www.rfc-editor.org/info/rfc8064>.

Authors' Addresses

Abdur Rashid Sangi
Huawei Technologies
No.156 Beiqing Rd. Haidian District
Beijing 100095
P.R. China

Email: sangi_bahrian@yahoo.com

Mach(Guoyi) Chen
Huawei Technologies
No.156 Beiqing Rd. Haidian District
Beijing 100095
P.R. China

Email: mach.chen@huawei.com

Charles E. Perkins
Futurewei
2330 Central Expressway
Santa Clara 95050
USA

Email: charliep@computer.org

6lo
Internet-Draft
Updates: 6775 (if approved)
Intended status: Standards Track
Expires: November 5, 2016

P. Thubert, Ed.
cisco
E. Nordmark
Arista Networks
S. Chakrabarti
Ericsson

May 4, 2016

# An Update to 6LoWPAN ND
# draft-thubert-6lo-rfc6775-update-00

## Abstract

This specification proposes an update to 6LoWPAN Neighbor Discovery, to clarify the role of the protocol as a registration technique, and provide enhancements to the registration capabilities, in particular for the registration to a backbone router for proxy ND operations.

## 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 http://datatracker.ietf.org/drafts/current/.

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

This Internet-Draft will expire on November 5, 2016.

## Copyright Notice

Table of Contents



## 1. Introduction

The scope of this draft is an IPv6 Low Power Lossy Network (LLN), which can be a simple star or a more complex mesh topology. The LLN may be anchored at an IPv6 Backbone Router (6BBR). The Backbone Routers interconnect the LLNs over a Backbone Link and emulate that the LLN nodes are present on the Backbone using proxy-ND operations.

IPv6 Neighbor Discovery (ND) Optimization for IPv6 over Low-Power Wireless Personal Area Networks(6LoWPANs) [RFC6775] introduced a proactive registration mechanism to IPv6 ND services for nodes belonging to a LLN.

This specification modifies and extends the behaviour and protocol elements of [RFC6775] to enable additional capabilities, in particular the registration to a 6BBR for proxy ND operations [I-D.ietf-6lo-backbone-router].

## 2. Terminology

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

Readers are expected to be familiar with all the terms and concepts that are discussed in "Neighbor Discovery for IP version 6" [RFC4861], "IPv6 Stateless Address Autoconfiguration" [RFC4862], "IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs): Overview, Assumptions, Problem Statement, and Goals" [RFC4919], Neighbor Discovery Optimization for Low-power and Lossy Networks [RFC6775] and "Multi-link Subnet Support in IPv6" [I-D.ietf-ipv6-multilink-subnets].

Additionally, this document uses terminology from "Terms Used in Routing for Low-Power and Lossy Networks" [RFC7102] and [I-D.ietf-6tisch-terminology], as well as this additional terminology:

Backbone This is an IPv6 transit link that interconnects 2 or more Backbone Routers. It is expected to be deployed as a high speed backbone in order to federate a potentially large set of LLNS. Also referred to as a LLN backbone or Backbone network.

Backbone Router An IPv6 router that federates the LLN using a Backbone link as a backbone. A BBR acts as a 6LoWPAN Border Routers (6LBR) and an Energy Aware Default Router (NEAR).

Extended LLN This is the aggregation of multiple LLNs as defined in [RFC4919], interconnected by a Backbone Link via Backbone Routers, and forming a single IPv6 MultiLink Subnet.

Registration The process during which a wireless Node registers its address(es) with the Border Router so the 6BBR can proxy ND for it over the backbone.

Binding The state in the 6BBR that associates an IP address with a MAC address, a port and some other information about the node that owns the IP address.

Registered Node The node for which the registration is performed, which owns the fields in the EARO option.

Registering Node The node that performs the registration to the 6BBR, either for one of its own addresses, in which case it is Registered Node and indicates its own MAC Address as SLLA in the NS(ARO), or on behalf of a Registered Node that is

reachable over a LLN mesh. In the latter case, if the Registered Node is reachable from the 6BBR over a Mesh-Under mesh, the Registering Node indicates the MAC Address of the Registered Node as SLLA in the NS(ARO). Otherwise, it is expected that the Registered Device is reachable over a Route-Over mesh from the Registering Node, in which case the SLLA in the NS(ARO) is that of the Registering Node, which causes it to attract the packets from the 6BBR to the Registered Node and route them over the LLN.

Registered Address The address owned by the Registered Node node that is being registered.

## 3. Updating RFC 6775

The support of this specification is signaled in Router Advertisement (RA) messages by 6LoWPAN Router (6LR) (how: tbd). A Registering Node that supports this specification will favor registering to a 6LR that indicates support for this specification over that of [RFC6775].

### 3.1. Extended Address Registration Option

This specification extends the Address Registration Option (ARO) used for the process of address registration. The new ARO is referred to as Extended ARO (EARO), and its semantics are modified as follows:

The address that is being registered with a Neighbor Solicitation (NS) with an EARO is now the Target Address, as opposed to the Source Address as specified in [RFC6775]. This change enables a 6LBR to use an address of his as source to the proxy-registration of an address that belongs to a LLN Node to a 6BBR. This also limits the use of an address as source address before it is registered and the associated Duplicate Address Detection (DAD) is complete.

The Unique ID in the EARO option does no more have to be a MAC address. A new TLV format is introduced and a IANA registry is created for the type (TBD). This enables in particular the use of a Provable Temporary UID (PT-UID) as opposed to burn-in MAC address, the PT-UID providing a trusted anchor by the 6LR and 6LBR to protect the state associated to the node.

The specification introduces a Transaction ID (TID) field in the EARO. The TID MUST be provided by a node that supports this specification and a new T flag MUST be set to indicate so. The T bit can be used to determine whether the peer supports this specification.

### 3.2. Link-local Scope and Consequences

The use of link-local addresses as source address for the registration, and the expectation for the scope of those addresses, are clarified as follows:

A link is abstracted as a one-hop point-to-point communication medium. There is no need nor expectation that a link-local address is unique across the whole LLN. A 6LR assumes that the link-local address of a Registering Node is unique as long as the 6LR does not have a conflicting registration for that address.

An exchange between two nodes using link-local addresses implies that they are reachable over one hop and that at least one of the 2 nodes acts as a 6LR. A node MUST register a link-local address to a 6LR in order to obtain link scope reachability from that 6LR beyond the current exchange, and in particular to use it as Source Address to register other addresses.

A consequence of this model is that the Duplicate Address Detection (DAD) process between the 6LR and a 6LoWPAN Border Router (6LBR), which is based on a Duplicate Address Request (DAR) / Duplicate Address Confirmation (DAC) exchange as described in [RFC6775], does not take place for link-local addresses.

It is desired that a 6LR does not need to modify its state associated to the Source Address of an NS(EARO) message. For that reason, when possible, it is RECOMMENDED to use an address that is already registered with a 6LR as source for the NS(EARO) message.

When a Registering Node does not yet have an already-registered address, it MUST register a link-local address, using it as both the Source and the Target Address of an NS(EARO) message. In that case, it is RECOMMENDED to use a link-local address that is (expected to be) globally unique, e.g. derived from a burn-in MAC address.

Since there is no DAR/DAC exchange for link-local addresses, the 6LR may answer immediately to the registration of a link-local address, based solely on its existing state and the Source Link-Layer Option that MUST be placed in the NS(EARO) message as required in [RFC6775].

A node must register its IPv6 Global Unicast IPv6 Addresses (GUA) to a 6LR in order to obtain a global reachability for these addresses via that 6LR. In particular a Registering NODE registering a GUA SHOULD NOT use that GUA as Source Address for the registration to a 6LR that conforms this specification.

What makes this model practical in existing LLNs, which can grow to large number of nodes, is that a subnet may encompass multiple links, which can be LLN links or can be backbone links that federate a number of LLN links, effectively forming a non-broadcast multi-access (NBMA) multi-link subnet (MLSN).

## 4. Applicability and Requirements Served

This specification extends 6LoWPAN ND to sequence the registration and serves the requirements expressed Appendix A.1 by enabling the mobility of devices from one LLN to the next based on the complementary work in [I-D.ietf-6lo-backbone-router].

In the context of the the TimeSlotted Channel Hopping (TSCH) mode of [IEEE802154], the 6TiSCH architecture [I-D.ietf-6tisch-architecture] introduces how a 6LoWPAN ND host could connect to the Internet via a RPL mesh Network, but this requires additions to the 6LOWPAN ND protocol to support mobility and reachability in a secured and manageable environment. This specification details the new operations that are required to implement the 6TiSCH architecture and serves the requirements listed in Appendix A.2.

The term LLN is used loosely in this specification to cover multiple types of WLANs and WPANs, including Low-Power Wi-Fi, BLUETOOTH(R) Low Energy, IEEE802.11AH and IEEE802.15.4 wireless meshes, so as to address the requirements discussed in Appendix A.3

This specification can be used by any wireless node to associate at Layer-3 with a 6BBR and register its IPv6 addresses to obtain routing services including proxy-ND operations over the backbone, effectively providing a solution to the requirements expressed in Appendix A.4.

Efficiency aware IPv6 Neighbor Discovery Optimizations [I-D.chakrabarti-nordmark-6man-efficient-nd] suggests that 6LoWPAN ND [RFC6775] can be extended to other types of links beyond IEEE802.15.4 for which it was defined. The registration technique is beneficial when the Link-Layer technique used to carry IPv6 multicast packets is not sufficiently efficient in terms of delivery ratio or energy consumption in the end devices, in particular to enable energy-constrained sleeping nodes. The value of such extension is especially apparent in the case of mobile wireless nodes, to reduce the multicast operations that are related to classical ND ([RFC4861], [RFC4862]) and plague the wireless medium. This serves scalability requirements listed in Appendix A.6.

## 5. The Enhanced Address Registration Option (EARO)

With the ARO option defined in 6LoWPAN ND [RFC6775], the address being registered and its owner can be uniquely identified and matched with the Binding Table entries of each Backbone Router.

The Enhanced Address Registration Option (EARO) is intended to be used as a replacement to the ARO option within Neighbor Discovery NS and NA messages between a LLN node and its 6LoWPAN Router (6LR), as well as in Duplicate Address Request (DAR) and the Duplicate Address Confirmation (DAC) messages between 6LRs and 6LBRs in LLNs meshes such as 6TiSCH networks.

An NS message with an EARO option is a registration if and only if it also carries an SLLAO option. The AERO option also used in NS and NA messages between Backbone Routers over the backbone link to sort out the distributed registration state, and in that case, it does not carry the SLLAO option and is not confused with a registration.

The EARO extends the ARO and is recognized by the setting of the TID bit. A node that supports this specification MUST always use an EARO as a replacement to an ARO in its registration to a router. This is harmless since the TID bit and fields are reserved in [RFC6775] are ignored by a legacy router. A router that supports this specification answers to an ARO with an ARO and to an EARO with an EARO.

This specification changes the behavior of the peers in a registration flows. To enable backward compatibility, a node that registers to a router that is not known to support this specification MUST behave as prescribed by [RFC6775]. Once the router is known to support this specification, the node MUST obey this specification.

When using the EARO option, the address being registered is found in the Target Address field of the NS and NA messages. This differs from 6LoWPAN ND [RFC6775] which specifies that the address being registered is the source of the NS.

The reason for this change is to enable proxy-registrations on behalf of other nodes in Route-Over meshes, for instance to enable that a RPL root registers addresses on behalf LLN nodes that are deeper in a 6TiSCH mesh. In that case, the Registering Node MUST indicate its own address as source of the ND message and its MAC address in the Source Link-Layer Address Option (SLLAO), since it still expects to get the packets and route them down the mesh. But the Registered Address belongs to another node, the Registered Node, and that address is indicated in the Target Address field of the NS message.

One way of achieving all the above is for a node to first register an address that it owns in order to validate that the router supports this specification, placing the same address in the Source and Target Address fields of the NS message. The node may for instance register an address that is based on EUI-64. For such address, DAD is not required and using the SLLAO option in the NS is actually more amenable with older ND specifications such as ODAD [RFC4429].

Once that first registration is complete, the node knows from the setting of the TID in the response whether the router supports this specification. If this is verified, the node may register other addresses that it owns, or proxy-register addresses on behalf some another node, indicating those addresses being registered in the Target Address field of the NS messages, while using one of its own, already registered, addresses as source.

The format of the EARO option is as follows:

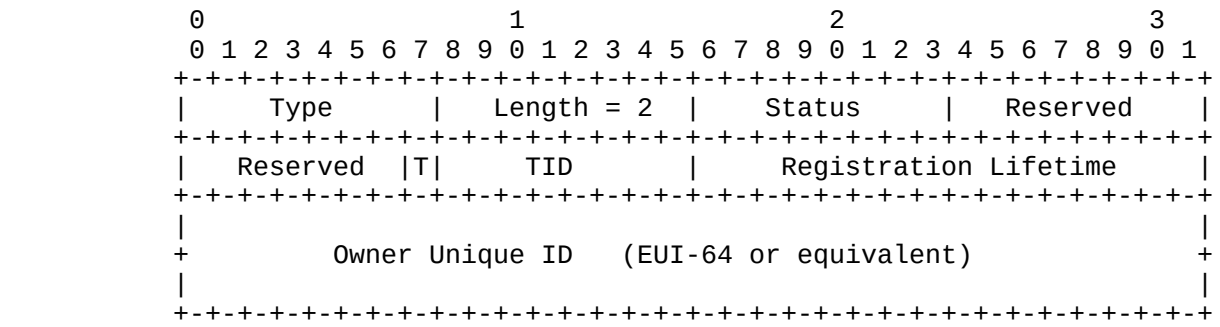



Figure 1: EARO

Option Fields

Type:

Length: 2

Status:

| Value | Description |
|---|---|
| 0..2 | See [RFC6775]. Note that a Status of 1 "Duplicate Address" applies to the Registered Address. If the Source Address differs from the Registered Address it conflicts with an existing registration, "Invalid Source Address" should be used instead. |
| 3 | Moved: The registration fails because it is not the freshest. |
| 4 | Removed: The binding state was removed. This may be placed in an asynchronous NS(ARO) message, or as the rejection of a proxy registration to a Backbone Router. |
| 5 | Proof requested |
| 6 | Invalid Source Address: The address used as source of the NS(ARO) conflicts with an existing registration, or is not usable on this link, e.g. it is not topologically correct. |
| 7 | Administrative Rejection: The address is reserved for another use by an administrative decision (e.g. placed in a DHCPv6 pool). The Registering Node is requested to form a different address and retry. |

Table 1

Reserved: This field is unused. It MUST be initialized to zero by the sender and MUST be ignored by the receiver.

T: One bit flag. Set if the next octet is a used as a TID.

TID: 1-byte integer; a transaction id that is maintained by the node and incremented with each transaction. it is recommended that the node maintains the TID in a persistent storage.

Registration Lifetime: 16-bit integer; expressed in minutes. 0 means that the registration has ended and the state should be removed.

Owner Unique Identifier (OUI): A globally unique identifier for the node associated. This can be the EUI-64 derived IID of an interface, or some provable ID obtained cryptographically.

New status values are introduced, their values to be confirmed by IANA:

Moved: This status indicates that the registration is rejected because another more recent registration was done, as indicated by a same OUI and a more recent TID. One possible cause is a stale registration that has progressed slowly in the network and was passed by a more recent one. It could also indicate a OUI collision.

Removed: This status is expected in asynchronous messages from a registrar (6LR, 6LBR, 6BBR) to indicate that the registration state is removed, for instance due to time out of a lifetime, or a movement. It is used for instance by a 6BBR in a NA(ARO) message to indicate that the ownership of the proxy state on the backbone was transfered to another 6BBR, which is indicative of a movement of the device. The receiver of the NA is the device that has performed a registration that is now stale and it should clean up its state.

## 6. Security Considerations

This specification expects that the link layer is sufficiently protected, either by means of physical or IP security for the Backbone Link or MAC sublayer cryptography. In particular, it is expected that the LLN MAC provides secure unicast to/from the Backbone Router and secure Broadcast from the Backbone Router in a way that prevents tempering with or replaying the RA messages.

The use of EUI-64 for forming the Interface ID in the link-local address prevents the usage of Secure ND ([RFC3971] and [RFC3972]) and address privacy techniques. This specification RECOMMENDS the use of additional protection against address theft such as provided by [I-D.sarikaya-6lo-ap-nd], which guarantees the ownership of the OUID.

When the ownership of the OUID cannot be assessed, this specification limits the cases where the OUID and the TID are multicasted, and obfuscates them in responses to attempts to take over an address.

The LLN nodes depend on the 6LBR and the 6BBR for their operation. A trust model must be put in place to ensure that the right devices are acting in these roles, so as to avoid threats such as black-holing, or bombing attack whereby an impersonated 6LBR would destroy state in the network by using the "Removed" status code.

## 7. IANA Considerations

This document requires the following additions:

Address Registration Option Status Values Registry

| Status | Description |
|---|---|
| 3 | Moved |
| 4 | Removed |
| 5 | Proof requested |
| 6 | Invalid Source Address |
| 7 | Administrative Rejection |

IANA is required to change the registry accordingly

Table 2: New ARO Status values

## 8. Acknowledgments

Kudos to Eric Levy-Abegnoli who designed the First Hop Security infrastructure at Cisco.

## 9. References

### 9.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <http://www.rfc-editor.org/info/rfc2119>.

[RFC4429] Moore, N., "Optimistic Duplicate Address Detection (DAD) for IPv6", RFC 4429, DOI 10.17487/RFC4429, April 2006, <http://www.rfc-editor.org/info/rfc4429>.

[RFC4861] Narten, T., Nordmark, E., Simpson, W., and H. Soliman, "Neighbor Discovery for IP version 6 (IPv6)", RFC 4861, DOI 10.17487/RFC4861, September 2007, <http://www.rfc-editor.org/info/rfc4861>.

[RFC4862] Thomson, S., Narten, T., and T. Jinmei, "IPv6 Stateless Address Autoconfiguration", RFC 4862, DOI 10.17487/RFC4862, September 2007, <http://www.rfc-editor.org/info/rfc4862>.

[RFC6550] Winter, T., Ed., Thubert, P., Ed., Brandt, A., Hui, J., Kelsey, R., Levis, P., Pister, K., Struik, R., Vasseur, JP., and R. Alexander, "RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks", RFC 6550, DOI 10.17487/RFC6550, March 2012, <http://www.rfc-editor.org/info/rfc6550>.

[RFC6775] Shelby, Z., Ed., Chakrabarti, S., Nordmark, E., and C. Bormann, "Neighbor Discovery Optimization for IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs)", RFC 6775, DOI 10.17487/RFC6775, November 2012, <http://www.rfc-editor.org/info/rfc6775>.

## 9.2. Informative References

[I-D.chakrabarti-nordmark-6man-efficient-nd]
Chakrabarti, S., Nordmark, E., Thubert, P., and M. Wasserman, "IPv6 Neighbor Discovery Optimizations for Wired and Wireless Networks", draft-chakrabarti-nordmark-6man-efficient-nd-07 (work in progress), February 2015.

[I-D.delcarpio-6lo-wlanah]
Vega, L., Robles, I., and R. Morabito, "IPv6 over 802.11ah", draft-delcarpio-6lo-wlanah-01 (work in progress), October 2015.

[I-D.ietf-6lo-6lobac]
Lynn, K., Martocci, J., Neilson, C., and S. Donaldson, "Transmission of IPv6 over MS/TP Networks", draft-ietf-6lo-6lobac-04 (work in progress), February 2016.

[I-D.ietf-6lo-backbone-router]
Thubert, P., "IPv6 Backbone Router", draft-ietf-6lo-backbone-router-01 (work in progress), March 2016.

[I-D.ietf-6lo-dect-ule]
Mariager, P., Petersen, J., Shelby, Z., Logt, M., and D. Barthel, "Transmission of IPv6 Packets over DECT Ultra Low Energy", draft-ietf-6lo-dect-ule-04 (work in progress), February 2016.

[I-D.ietf-6lo-nfc]
Hong, Y. and J. Youn, "Transmission of IPv6 Packets over Near Field Communication", draft-ietf-6lo-nfc-03 (work in progress), March 2016.

[I-D.ietf-6man-rs-refresh]
Nordmark, E., Yourtchenko, A., and S. Krishnan, "IPv6 Neighbor Discovery Optional RS/RA Refresh", draft-ietf-6man-rs-refresh-01 (work in progress), March 2016.

[I-D.ietf-6tisch-architecture]
Thubert, P., "An Architecture for IPv6 over the TSCH mode of IEEE 802.15.4", draft-ietf-6tisch-architecture-09 (work in progress), November 2015.

[I-D.ietf-6tisch-terminology]
Palattella, M., Thubert, P., Watteyne, T., and Q. Wang, "Terminology in IPv6 over the TSCH mode of IEEE 802.15.4e", draft-ietf-6tisch-terminology-07 (work in progress), March 2016.

[I-D.ietf-bier-architecture]
Wijnands, I., Rosen, E., Dolganow, A., Przygienda, A., and S. Aldrin, "Multicast using Bit Index Explicit Replication", draft-ietf-bier-architecture-03 (work in progress), January 2016.

[I-D.ietf-ipv6-multilink-subnets]
Thaler, D. and C. Huitema, "Multi-link Subnet Support in IPv6", draft-ietf-ipv6-multilink-subnets-00 (work in progress), July 2002.

[I-D.nordmark-6man-dad-approaches]
Nordmark, E., "Possible approaches to make DAD more robust and/or efficient", draft-nordmark-6man-dad-approaches-02 (work in progress), October 2015.

[I-D.popa-6lo-6loplc-ipv6-over-ieee19012-networks]
Popa, D. and J. Hui, "6LoPLC: Transmission of IPv6 Packets over IEEE 1901.2 Narrowband Powerline Communication Networks", draft-popa-6lo-6loplc-ipv6-over-ieee19012-networks-00 (work in progress), March 2014.

[I-D.sarikaya-6lo-ap-nd]
Sarikaya, B. and P. Thubert, "Address Protected Neighbor Discovery for Low-power and Lossy Networks", draft-sarikaya-6lo-ap-nd-02 (work in progress), March 2016.

[I-D.vyncke-6man-mcast-not-efficient]
Vyncke, E., Thubert, P., Levy-Abegnoli, E., and A. Yourtchenko, "Why Network-Layer Multicast is Not Always Efficient At Datalink Layer", draft-vyncke-6man-mcast-not-efficient-01 (work in progress), February 2014.

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

[RFC3971] Arkko, J., Ed., Kempf, J., Zill, B., and P. Nikander, "SEcure Neighbor Discovery (SEND)", RFC 3971, DOI 10.17487/RFC3971, March 2005, <http://www.rfc-editor.org/info/rfc3971>.

[RFC3972] Aura, T., "Cryptographically Generated Addresses (CGA)", RFC 3972, DOI 10.17487/RFC3972, March 2005, <http://www.rfc-editor.org/info/rfc3972>.

[RFC4919] Kushalnagar, N., Montenegro, G., and C. Schumacher, "IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs): Overview, Assumptions, Problem Statement, and Goals", RFC 4919, DOI 10.17487/RFC4919, August 2007, <http://www.rfc-editor.org/info/rfc4919>.

[RFC6282] Hui, J., Ed. and P. Thubert, "Compression Format for IPv6 Datagrams over IEEE 802.15.4-Based Networks", RFC 6282, DOI 10.17487/RFC6282, September 2011, <http://www.rfc-editor.org/info/rfc6282>.

[RFC7102] Vasseur, JP., "Terms Used in Routing for Low-Power and Lossy Networks", RFC 7102, DOI 10.17487/RFC7102, January 2014, <http://www.rfc-editor.org/info/rfc7102>.

[RFC7217] Gont, F., "A Method for Generating Semantically Opaque Interface Identifiers with IPv6 Stateless Address Autoconfiguration (SLAAC)", RFC 7217, DOI 10.17487/RFC7217, April 2014, <http://www.rfc-editor.org/info/rfc7217>.

[RFC7428] Brandt, A. and J. Buron, "Transmission of IPv6 Packets over ITU-T G.9959 Networks", RFC 7428, DOI 10.17487/RFC7428, February 2015, <http://www.rfc-editor.org/info/rfc7428>.

[RFC7559] Krishnan, S., Anipko, D., and D. Thaler, "Packet-Loss Resiliency for Router Solicitations", RFC 7559, DOI 10.17487/RFC7559, May 2015, <http://www.rfc-editor.org/info/rfc7559>.

[RFC7668] Nieminen, J., Savolainen, T., Isomaki, M., Patil, B., Shelby, Z., and C. Gomez, "IPv6 over BLUETOOTH(R) Low Energy", RFC 7668, DOI 10.17487/RFC7668, October 2015, <http://www.rfc-editor.org/info/rfc7668>.

[RFC7772] Yourtchenko, A. and L. Colitti, "Reducing Energy Consumption of Router Advertisements", BCP 202, RFC 7772, DOI 10.17487/RFC7772, February 2016, <http://www.rfc-editor.org/info/rfc7772>.

### 9.3. External Informative References

[IEEE80211]
IEEE standard for Information Technology, "IEEE Standard for Information technology-- Telecommunications and information exchange between systems Local and metropolitan area networks-- Specific requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications".

[IEEE802151]
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)".

[IEEE802154]
IEEE standard for Information Technology, "IEEE Standard for Local and metropolitan area networks-- Part 15.4: Low-Rate Wireless Personal Area Networks (LR-WPANs)".

## Appendix A. Requirements

This section lists requirements that were discussed at 6lo for an update to 6LoWPAN ND. This specification meets most of them, but those listed in Appendix A.5 which are deferred to a different specification such as [I-D.sarikaya-6lo-ap-nd].

### A.1. Requirements Related to Mobility

Due to the unstable nature of LLN links, even in a LLN of immobile nodes a 6LoWPAN Node may change its point of attachment to a 6LR, say 6LR-a, and may not be able to notify 6LR-a. Consequently, 6LR-a may still attract traffic that it cannot deliver any more. When links to a 6LR change state, there is thus a need to identify stale states in a 6LR and restore reachability in a timely fashion.

Req1.1: Upon a change of point of attachment, connectivity via a new 6LR MUST be restored timely without the need to de-register from the previous 6LR.

Req1.2: For that purpose, the protocol MUST enable to differentiate between multiple registrations from one 6LoWPAN Node and registrations from different 6LoWPAN Nodes claiming the same address.

Req1.3: Stale states MUST be cleaned up in 6LRs.

Req1.4: A 6LoWPAN Node SHOULD also be capable to register its Address to multiple 6LRs, and this, concurrently.

### A.2. Requirements Related to Routing Protocols

The point of attachment of a 6LoWPAN Node may be a 6LR in an LLN mesh. IPv6 routing in a LLN can be based on RPL, which is the routing protocol that was defined at the IETF for this particular purpose. Other routing protocols than RPL are also considered by Standard Defining Organizations (SDO) on the basis of the expected network characteristics. It is required that a 6LoWPAN Node attached via ND to a 6LR would need to participate in the selected routing protocol to obtain reachability via the 6LR.

Next to the 6LBR unicast address registered by ND, other addresses including multicast addresses are needed as well. For example a routing protocol often uses a multicast address to register changes to established paths. ND needs to register such a multicast address to enable routing concurrently with discovery.

Multicast is needed for groups. Groups MAY be formed by device type (e.g. routers, street lamps), location (Geography, RPL sub-tree), or both.

The Bit Index Explicit Replication (BIER) Architecture [I-D.ietf-bier-architecture] proposes an optimized technique to enable multicast in a LLN with a very limited requirement for routing state in the nodes.

Related requirements are:

Req2.1: The ND registration method SHOULD be extended in such a fashion that the 6LR MAY advertise the Address of a 6LoWPAN Node over the selected routing protocol and obtain reachability to that Address using the selected routing protocol.

Req2.2: Considering RPL, the Address Registration Option that is used in the ND registration SHOULD be extended to carry enough information to generate a DAO message as specified in [RFC6550] section 6.4, in particular the capability to compute a Path Sequence and, as an option, a RPLInstanceID.

Req2.3: Multicast operations SHOULD be supported and optimized, for instance using BIER or MPL. Whether ND is appropriate for the registration to the 6BBR is to be defined, considering the additional burden of supporting the Multicast Listener Discovery Version 2 [RFC3810] (MLDv2) for IPv6.

### A.3. Requirements Related to the Variety of Low-Power Link types

6LoWPAN ND [RFC6775] was defined with a focus on IEEE802.15.4 and in particular the capability to derive a unique Identifier from a globally unique MAC-64 address. At this point, the 6lo Working Group is extending the 6LoWPAN Header Compression (HC) [RFC6282] technique to other link types ITU-T G.9959 [RFC7428], Master-Slave/Token-Passing [I-D.ietf-6lo-6lobac], DECT Ultra Low Energy [I-D.ietf-6lo-dect-ule], Near Field Communication [I-D.ietf-6lo-nfc], IEEE802.11ah [I-D.delcarpio-6lo-wlanah], as well as IEEE1901.2 Narrowband Powerline Communication Networks [I-D.popa-6lo-6loplc-ipv6-over-ieee19012-networks] and BLUETOOTH(R) Low Energy [RFC7668].

Related requirements are:

Req3.1: The support of the registration mechanism SHOULD be extended to more LLN links than IEEE 802.15.4, matching at least the LLN links for which an "IPv6 over foo" specification exists, as well as Low-Power Wi-Fi.

Req3.2: As part of this extension, a mechanism to compute a unique Identifier should be provided, with the capability to form a Link-Local Address that SHOULD be unique at least within the LLN connected to a 6LBR discovered by ND in each node within the LLN.

Req3.3: The Address Registration Option used in the ND registration SHOULD be extended to carry the relevant forms of unique Identifier.

Req3.4: The Neighbour Discovery should specify the formation of a site-local address that follows the security recommendations from [RFC7217].

### A.4. Requirements Related to Proxy Operations

Duty-cycled devices may not be able to answer themselves to a lookup from a node that uses classical ND on a backbone and may need a proxy. Additionally, the duty-cycled device may need to rely on the 6LBR to perform registration to the 6BBR.

The ND registration method SHOULD defend the addresses of duty-cycled devices that are sleeping most of the time and not capable to defend their own Addresses.

Related requirements are:

Req4.1: The registration mechanism SHOULD enable a third party to proxy register an Address on behalf of a 6LoWPAN node that may be sleeping or located deeper in an LLN mesh.

Req4.2: The registration mechanism SHOULD be applicable to a duty-cycled device regardless of the link type, and enable a 6BBR to operate as a proxy to defend the registered Addresses on its behalf.

Req4.3: The registration mechanism SHOULD enable long sleep durations, in the order of multiple days to a month.

### A.5. Requirements Related to Security

In order to guarantee the operations of the 6LoWPAN ND flows, the spoofing of the 6LR, 6LBR and 6BBRs roles should be avoided. Once a node successfully registers an address, 6LoWPAN ND should provide energy-efficient means for the 6LBR to protect that ownership even when the node that registered the address is sleeping.

In particular, the 6LR and the 6LBR then should be able to verify whether a subsequent registration for a given Address comes from the original node.

In a LLN it makes sense to base security on layer-2 security. During bootstrap of the LLN, nodes join the network after authorization by a Joining Assistant (JA) or a Commissioning Tool (CT). After joining nodes communicate with each other via secured links. The keys for the layer-2 security are distributed by the JA/CT. The JA/CT can be part of the LLN or be outside the LLN. In both cases it is needed that packets are routed between JA/CT and the joining node.

Related requirements are:

Req5.1: 6LoWPAN ND security mechanisms SHOULD provide a mechanism for the 6LR, 6LBR and 6BBR to authenticate and authorize one another for their respective roles, as well as with the 6LoWPAN Node for the role of 6LR.

Req5.2: 6LoWPAN ND security mechanisms SHOULD provide a mechanism for the 6LR and the 6LBR to validate new registration of authorized nodes. Joining of unauthorized nodes MUST be impossible.

Req5.3: 6LoWPAN ND security mechanisms SHOULD lead to small packet sizes. In particular, the NS, NA, DAR and DAC messages for a re-registration flow SHOULD NOT exceed 80 octets so as to fit in a secured IEEE802.15.4 frame.

Req5.4: Recurrent 6LoWPAN ND security operations MUST NOT be computationally intensive on the LoWPAN Node CPU. When a Key hash calculation is employed, a mechanism lighter than SHA-1 SHOULD be preferred.

Req5.5: The number of Keys that the 6LoWPAN Node needs to manipulate SHOULD be minimized.

Req5.6: The 6LoWPAN ND security mechanisms SHOULD enable CCM* for use at both Layer 2 and Layer 3, and SHOULD enable the reuse of security code that has to be present on the device for upper layer security such as TLS.

Req5.7: Public key and signature sizes SHOULD be minimized while maintaining adequate confidentiality and data origin authentication for multiple types of applications with various degrees of criticality.

Req5.8: Routing of packets should continue when links pass from the unsecured to the secured state.

Req5.9: 6LoWPAN ND security mechanisms SHOULD provide a mechanism for the 6LR and the 6LBR to validate whether a new registration for a given address corresponds to the same 6LoWPAN Node that registered it initially, and, if not, determine the rightful owner, and deny or clean-up the registration that is duplicate.

### A.6. Requirements Related to Scalability

Use cases from Automatic Meter Reading (AMR, collection tree operations) and Advanced Metering Infrastructure (AMI, bi-directional communication to the meters) indicate the needs for a large number of

LLN nodes pertaining to a single RPL DODAG (e.g. 5000) and connected to the 6LBR over a large number of LLN hops (e.g. 15).

Related requirements are:

Req6.1: The registration mechanism SHOULD enable a single 6LBR to register multiple thousands of devices.

Req6.2: The timing of the registration operation should allow for a large latency such as found in LLNs with ten and more hops.

## Authors' Addresses

Pascal Thubert (editor)
Cisco Systems, Inc
Building D
45 Allee des Ormes - BP1200
MOUGINS - Sophia Antipolis 06254
FRANCE

Phone: +33 497 23 26 34
Email: pthubert@cisco.com

Erik Nordmark
Arista Networks
Santa Clara, CA
USA

Email: nordmark@arista.com

Samita Chakrabarti
Ericsson
San Jose, CA
USA

Email: samita.chakrabarti@ericsson.com