

6Lo Working Group
Internet-Draft
Intended status: Informational
Expires: May 31, 2015

Y-G. Hong
Y-H. Choi
ETRI
J-S. Youn
DONG-EUI Univ
D-K. Kim
KNU
J-H. Choi
Samsung Electronics Co.,
November 27, 2014

Transmission of IPv6 Packets over Near Field Communication
draft-hong-6lo-ipv6-over-nfc-03

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. 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 <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 31, 2015.

Copyright Notice

Copyright (c) 2014 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	2
2. Conventions and Terminology	4
3. Overview of Near Field Communication Technology	4
3.1. Peer-to-peer Mode for IPv6 over NFC	4
3.2. Protocol Stacks in IPv6 over NFC	5
3.3. NFC-enabled Device Addressing	6
3.4. NFC Packet Size and MTU	6
4. Specification of IPv6 over NFC	7
4.1. Protocol Stack	7
4.2. Link Model	8
4.3. Stateless Address Autoconfiguration	8
4.4. Neighbor Discovery	9
4.5. Header Compression	9
4.6. Fragmentation and Reassembly	10
4.7. Unicast Address Mapping	10
4.8. Multicast Address Mapping	11
5. Internet Connectivity Scenarios	11
5.1. NFC-enabled Device Connected to the Internet	11
5.2. Isolated NFC-enabled Device Network	12
6. IANA Considerations	12
7. Security Considerations	12
8. References	12
8.1. Normative References	12
8.2. Informative References	13
Authors' Addresses	13

1. Introduction

NFC is a set of short-range wireless technologies, typically requiring a distance of 10 cm or less. NFC operates at 13.56 MHz on ISO/IEC 18000-3 air interface and at rates ranging from 106 kbit/s to

424 kbit/s. 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 that do not require batteries. NFC peer-to-peer communication is possible, provided both devices are powered. NFC builds upon RFID systems by allowing two-way communication between endpoints, where earlier systems such as contactless smart cards were one-way only. It has been used in devices such as mobile phones, running Android operating system, named with a feature called "Android Beam". In addition, it is expected for the other mobile phones, running the other operating systems (e.g., iOS, etc.) to be equipped with NFC technology in the near future.

Considering the potential for exponential growth in the number of heterogeneous air interface technologies, NFC would be widely used as one of the other air interface technologies, such as Bluetooth Low Energy (BT-LE), Wi-Fi, and so on. Each of the heterogeneous air interface technologies has its own characteristics, which cannot be covered by the other technologies, so various kinds of air interface technologies would be existing together. Therefore, it is required for them to communicate each other. NFC also has the strongest point (e.g., secure communication distance of 10 cm) to prevent the third party from attacking privacy.

When the number of devices and things having different air interface technologies communicate each other, IPv6 is an ideal internet protocols owing to its large address space. Also, NFC would be one of the endpoints using IPv6. Therefore, This document describes how IPv6 is transmitted over NFC using 6LoWPAN techniques with following scopes.

- o Overview of NFC technologies;
- o Specifications for IPv6 over NFC;
 - * Neighbor Discovery;
 - * Addressing and Configuration;
 - * Header Compression;
 - * Fragmentation & Reassembly for a IPv6 datagram;

RFC4944 [1] specifies the transmission of IPv6 over IEEE 802.15.4. The NFC link also has similar characteristics to that of IEEE 802.15.4. Many of the mechanisms defined in the RFC4944 [1] can be

applied to the transmission of IPv6 on NFC links. This document specifies the details of IPv6 transmission over NFC links.

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 [2].

3. Overview of Near Field Communication Technology

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.

3.1. Peer-to-peer Mode for IPv6 over NFC

NFC-enabled devices are unique in that they can support three modes of operation: card emulation, peer-to-peer, and reader/writer. Peer-to-peer mode enables two NFC-enabled devices to communicate with each other to exchange information and share files, so that users of NFC-enabled devices can quickly share contact information and other files with a touch. Therefore, a NFC-enabled device can securely send IPv6 packets to any corresponding node on the Internet when a NFC-enabled gateway is linked to the Internet.

3.2. Protocol Stacks in IPv6 over NFC

The IP protocol can use the services provided by Logical Link Control Protocol (LLCP) in the NFC stack to provide reliable, two-way transport of information between the peer devices. Figure 1 depicts the NFC P2P protocol stack with IPv6 bindings to the LLCP.

For data communication in IPv6 over NFC, an IPv6 packet SHALL be received at LLCP of NFC and transported to an Information Field in Protocol Data Unit (I PDU) of LLCP of the NFC-enabled peer device. Since LLCP does not support fragmentation and reassembly, Upper Layers SHOULD support fragmentation and reassembly. For IPv6 addressing or address configuration, LLCP SHALL provide related information, such as link layer addresses, to its upper layer. LLCP to IPv6 protocol Binding SHALL transfer the SSAP and DSAP value to the IPv6 over NFC protocol. SSAP stands for Source Service Access Point, which is 6-bit value meaning a kind of Logical Link Control (LLC) address, while DSAP means a LLC address of destination NFC-enabled device.

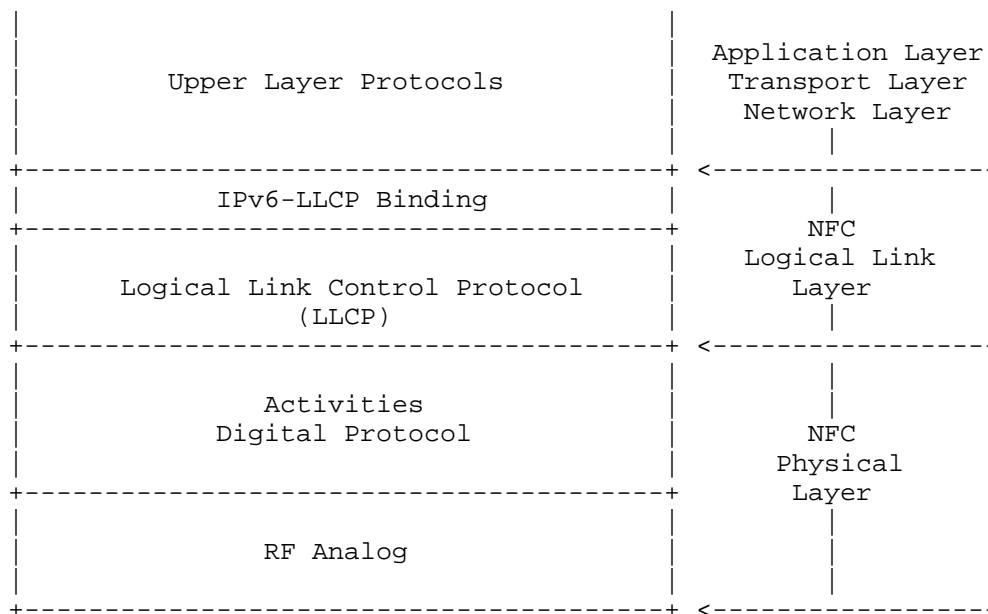


Figure 1: Protocol Stack of NFC

3.3. NFC-enabled Device Addressing

NFC-enabled devices are identified by 6-bit LLC address. In other words, Any address SHALL be usable as both an SSAP and a DSAP address. According to NFCForum-TS-LLCP_1.1 [3], address values between 0 and 31 (00h - 1Fh) SHALL be reserved for well-known service access points for Service Discovery Protocol (SDP). Address values between 32 and 63 (20h - 3Fh) inclusively, SHALL be assigned by the local LLC as the result of an upper layer service request.

3.4. NFC Packet Size and MTU

As mentioned in Section 3.2, an IPv6 packet SHALL be received at LLCP of NFC and transported to an Information Field in Protocol Data Unit (I PDU) of LLCP of the NFC-enabled peer device. The format of the I PDU SHALL be as shown in Figure 2.

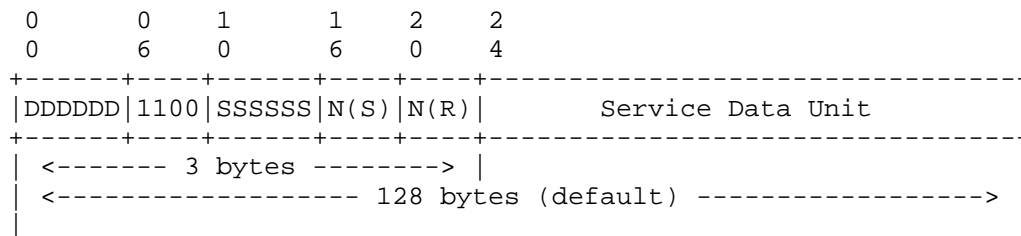


Figure 2: Format of the I PDU in NFC

The I PDU sequence field SHALL contain two sequence numbers: The send sequence number N(S) and the receive sequence number N(R). The send sequence number N(S) SHALL indicate the sequence number associated with this I PDU. The receive sequence number N(R) value SHALL indicate that I PDUs numbered up through N(R) - 1 have been received correctly by the sender of this I PDU and successfully passed to the senders SAP identified in the SSAP field. These I PDUs SHALL be considered as acknowledged.

The information field of an I PDU SHALL contain a single service data unit. The maximum number of octets in the information field SHALL be determined by the Maximum Information Unit (MIU) for the data link connection. The default value of the MIU for I PDUs SHALL be 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 MIUX extension parameter within the information field.

4. Specification of IPv6 over NFC

NFC technology sets also has considerations and requirements owing to low power consumption and allowed protocol overhead. 6LoWPAN standards RFC4944 [1], RFC6775 [4], and RFC6282 [5] provide useful functionality for reducing overhead which can be applied to BT-LE. This functionality comprises of link-local IPv6 addresses and stateless IPv6 address auto-configuration (see Section 4.3), Neighbor Discovery (see Section 4.4) and header compression (see Section 4.5).

One of the differences between IEEE 802.15.4 and NFC is that the former supports both star and mesh topology (and requires a routing protocol), whereas NFC can support direct peer-to-peer connection and simple mesh-like topology depending on NFC application scenarios because of very short RF distance of 10 cm or less.

4.1. Protocol Stack

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

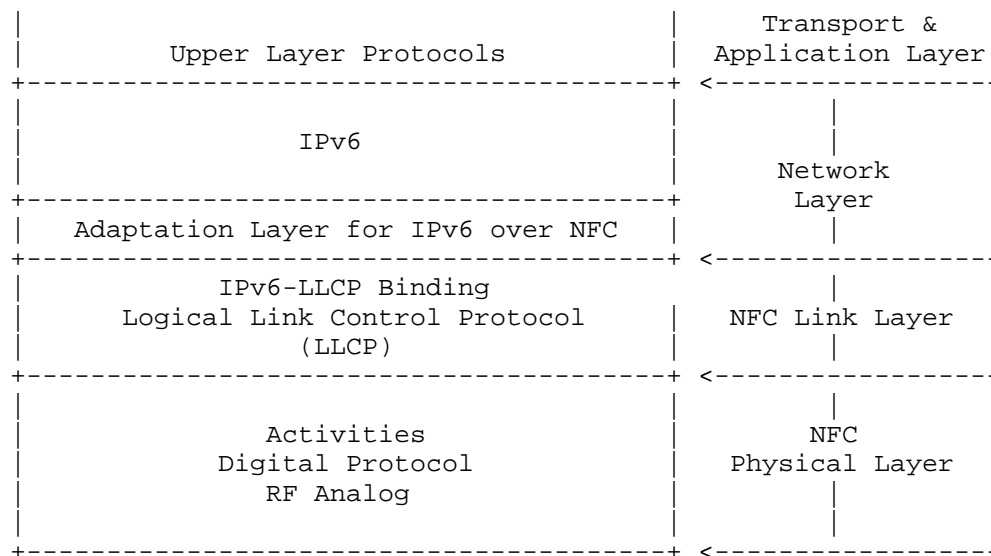


Figure 3: Protocol Stack for IPv6 over NFC

Adaptation layer for IPv6 over NFC SHALL support neighbor discovery, address auto-configuration, header compression, and fragmentation & reassembly.

4.2. Link Model

In the case of BT-LE, Logical Link Control and Adaptation Protocol (L2CAP) supports fragmentation and reassembly (FAR) functionality; therefore, adaptation layer for IPv6 over BT-LE do not have to conduct the FAR procedure. However, NFC link layer is similar to IEEE 802.15.4. Adaptation layer for IPv6 over NFC SHOULD support FAR functionality. Therefore, fragmentation functionality as defined in RFC4944 [1] SHALL be used in NFC-enabled device networks.

The NFC link between two communicating devices is considered to be a point-to-point link only. Unlike in BT-LE, NFC link does not consider star topology and mesh network topology but peer-to-peer topology and simple multi-hop topology. Due to this characteristics, 6LoWPAN functionality, such as addressing and auto-configuration, and header compression, is specialized into NFC.

4.3. Stateless Address Autoconfiguration

A NFC-enabled device (i.e., 6LN) performs stateless address autoconfiguration as per RFC4862 [6]. A 64-bit Interface identifier (IID) for a NFC interface MAY be formed by utilizing the 6-bit NFC LLC address (i.e., SSAP or DSAP) (see Section 3.3). In the viewpoint of address configuration, such an IID MAY guarantee a stable IPv6 address 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.

In the case of NFC-enabled device address, the "Universal/Local" bit MUST be set to 0 RFC4291 [7]. Only if the NFC-enabled device address is known to be a public address the "Universal/Local" bit can be set to 1. As defined in RFC4291, the IPv6 link-local address for a NFC-enabled device is formed by appending the IID, to the prefix FE80::/64, as depicted in Figure 4.

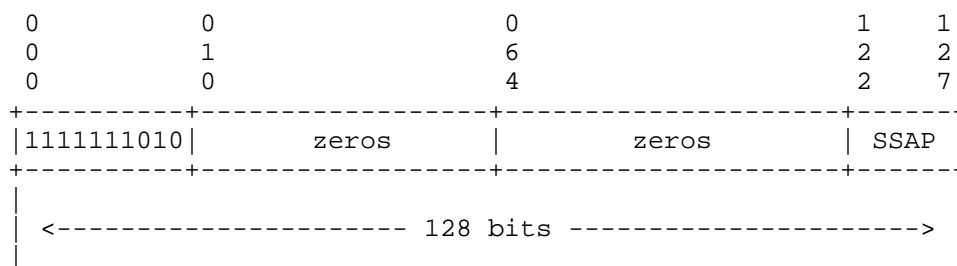


Figure 4: IPv6 link-local address in NFC

The tool for a 6LBR to obtain an IPv6 prefix for numbering the NFC network is can be accomplished via DHCPv6 Prefix Delegation (RFC3633 [8]).

4.4. Neighbor Discovery

Neighbor Discovery Optimization for 6LoWPANs (RFC6775 [4]) describes the neighbor discovery approach in several 6LoWPAN topologies, such as mesh topology. NFC does not consider complicated mesh topology but simple multi-hop network topology or directly connected peer-to-peer network. Therefore, the following aspects of RFC6775 are applicable to NFC:

1. In a case that a NFC-enabled device (6LN) is directly connected to 6LBR, A NFC 6LN MUST register its address with the 6LBR by sending a Neighbor Solicitation (NS) message with the Address Registration Option (ARO) and process the Neighbor Advertisement (NA) accordingly. In addition, DHCPv6 is used to assigned an address, Duplicate Address Detection (DAD) is not required.
2. For sending Router Solicitations and processing Router Advertisements the NFC 6LNs MUST follow Sections 5.3 and 5.4 of the RFC6775.

4.5. Header Compression

Header compression as defined in RFC6282 [5] , 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 RFC6282 encoding formats.

If a 16-bit address is required as a short address of IEEE 802.15.4, it MUST be formed by padding the 6-bit NFC link-layer (node) address to the left with zeros as shown in Figure 5.

```

      0                               1
    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
| Padding(all zeros)| NFC Addr. |
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+

```

Figure 5: NFC short adress format

4.6. Fragmentation and Reassembly

Fragmentation and reassembly (FAR) as defined in RFC4944, which specifies the fragmentation methods for IPv6 datagrams on top of IEEE 802.15.4, is REQUIRED in this document as the basis for IPv6 datagram FAR on top of NFC. All headers MUST be compressed according to RFC4944 encoding formats, but the default MTU of NFC is 128 bytes. This MUST be considered.

4.7. Unicast Address Mapping

The address resolution procedure for mapping IPv6 non-multicast addresses into NFC link-layer addresses follows the general description in Section 7.2 of RFC4861 [9], unless otherwise specified.

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

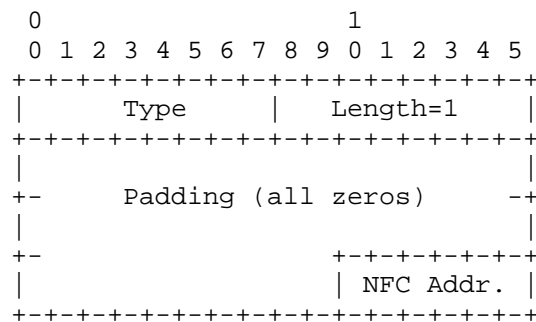


Figure 6: Unicast address mapping

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 octets. 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.

4.8. Multicast Address Mapping

All IPv6 multicast packets MUST be sent to NFC Destination Address, 0x3F (broadcast) and filtered at the IPv6 layer. When represented as a 16-bit address in a compressed header, it MUST be formed by padding on the left with a zero. In addition, the NFC Destination Address, 0x3F, MUST not be used as a unicast NFC address of SSAP or DSAP.

```

      0                               1
    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5
+---+---+---+---+---+---+---+---+
| Padding(all zeros) | 1 1 1 1 1 1 |
+---+---+---+---+---+---+---+---+

```

Figure 7: Multicast address mapping

5. Internet Connectivity Scenarios

As two typical scenarios, the NFC network can be isolated and connected to the Internet.

5.1. NFC-enabled Device Connected to the Internet

One of the key applications by using adaptation technology of IPv6 over NFC is the most securely transmitting IPv6 packets because RF distance between 6LN and 6LBR SHOULD be within 10 cm. If any third party wants to hack into the RF between them, it MUST come to nearly touch them. Applications can choose which kinds of air interfaces (e.g., BT-LE, Wi-Fi, NFC, etc.) to send data depending characteristics of data. NFC SHALL be the best solution for secured and private information.

Figure 8 illustrates an example of NFC-enabled device network connected to the Internet. Distance between 6LN and 6LBR SHOULD be 10 cm or less. If there is any of close laptop computers to a user, it SHALL become the 6LBR. Additionally, When the user mounts a NFC-enabled air interface adapter (e.g., portable small NFC dongle) on the close laptop PC, the user's NFC-enabled device (6LN) can communicate the laptop PC (6LBR) within 10 cm distance.

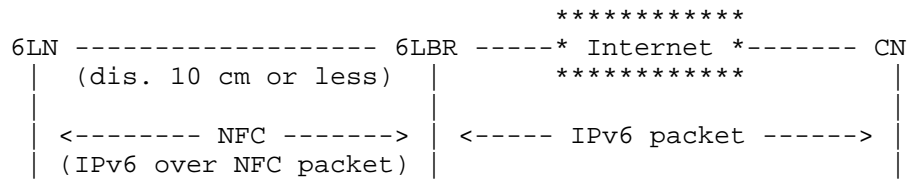


Figure 8: NFC-enabled device network connected to the Internet

5.2. Isolated NFC-enabled Device Network

In some scenarios, the NFC-enabled device network may transiently be a simple isolated network as shown in the Figure 9.

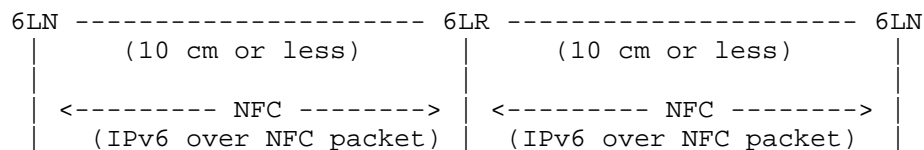


Figure 9: Isolated NFC-enabled device network

In mobile phone markets, applications are designed and made by user developers. They may image interesting applications, where three or more mobile phones touch or attach each other to accomplish outstanding performance. For instance, three or more mobile phones can play multi-channel sound of music together. In addition, attached three or more mobile phones can make an extended banner to show longer sentences in a concert hall.

6. IANA Considerations

There are no IANA considerations related to this document.

7. Security Considerations

The method of deriving Interface Identifiers from 6-bit NFC Link layer addresses is intended to preserve global uniqueness when it is possible. Therefore, it is required to protect from duplication through accident or forgery.

8. References

8.1. Normative References

- [1] Montenegro, G., Kushalnagar, N., Hui, J., and D. Culler, "Transmission of IPv6 Packets over IEEE 802.15.4 Networks", RFC 4944, September 2007.

- [2] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [3] "Logical Link Control Protocol version 1.1", NFC Forum Technical Specification , June 2011.
- [4] Shelby, Z., Chakrabarti, S., Nordmark, E., and C. Bormann, "Neighbor Discovery Optimization for IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs)", RFC 6775, November 2012.
- [5] Hui, J. and P. Thubert, "Compression Format for IPv6 Datagrams over IEEE 802.15.4-Based Networks", RFC 6282, September 2011.
- [6] Thomson, S., Narten, T., and T. Jinmei, "IPv6 Stateless Address Autoconfiguration", RFC 4862, September 2007.
- [7] Hinden, R. and S. Deering, "IP Version 6 Addressing Architecture", RFC 4291, February 2006.
- [8] Troan, O. and R. Droms, "IPv6 Prefix Options for Dynamic Host Configuration Protocol (DHCP) version 6", RFC 3633, December 2003.
- [9] Narten, T., Nordmark, E., Simpson, W., and H. Soliman, "Neighbor Discovery for IP version 6 (IPv6)", RFC 4861, September 2007.

8.2. Informative References

- [10] "Near Field Communication - Interface and Protocol (NFCIP-1) 3rd Ed.", ECMA-340 , June 2013.

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

Younghwan Choi
ETRI
218 Gajeongno, Yuseong
Daejeon 305-700
Korea

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

Joo-Sang Youn
DONG-EUI University
176 Eomgwangno Busan_jin_gu
Busan 614-714
Korea

Phone: +82 51 890 1993
Email: joosang.youn@gmail.com

Dongkyun Kim
Kyungpook National University
80 Daehak-ro, Buk-gu
Daegu 702-701
Korea

Phone: +82 53 950 7571
Email: dongkyun@knu.ac.kr

JinHyouk Choi
Samsung Electronics Co.,
129 Samsung-ro, Youngdong-gu
Suwon 447-712
Korea

Phone: +82 2 2254 0114
Email: jinchoe@samsung.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

Copyright (c) 2017 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents

carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	2
2. Profile for IPv6 over MS/TP	5
3. Addressing Modes	6
4. Maximum Transmission Unit (MTU)	7
5. LoBAC Adaptation Layer	7
6. Stateless Address Autoconfiguration	8
7. IPv6 Link Local Address	9
8. Unicast Address Mapping	9
9. Multicast Address Mapping	10
10. Header Compression	10
11. IANA Considerations	10
12. Security Considerations	11
13. Acknowledgments	11
14. References	11
Appendix A. Abstract MAC Interface	14
Appendix B. Consistent Overhead Byte Stuffing [COBS]	17
Appendix C. Encoded CRC-32K [CRC32K]	20
Appendix D. Example 6LoBAC Frame Decode	22
Authors' Addresses	27

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 specification (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:

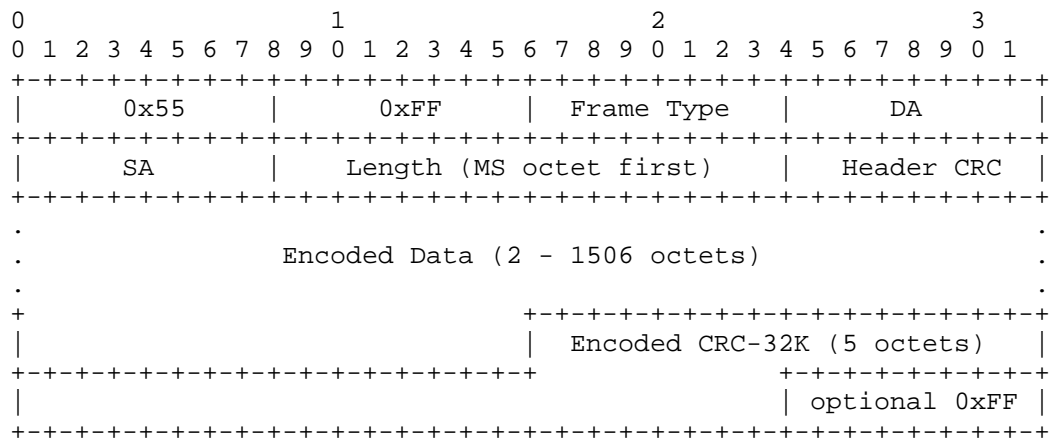


Figure 1: MS/TP COBS-Encoded Frame Format

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:

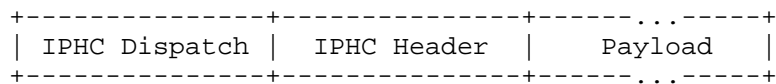


Figure 2: A LoBAC Encapsulated LOWPAN_IPHC Compressed IPv6 Datagram

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:

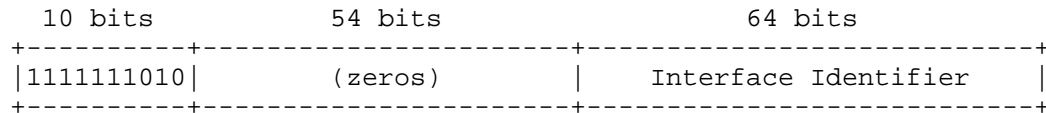
0	1 1	3 3	4 4	6
0	5 6	1 2	7 8	3
+-----+-----+-----+-----+-----+				
0000000000000000	0000000011111111	1111111000000000	0000000001001111	
+-----+-----+-----+-----+-----+				

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

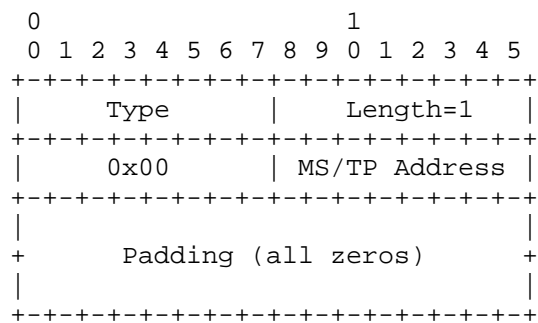
The IPv6 link-local address [RFC4291] for an MS/TP interface is formed by appending the Interface Identifier, as defined above, to the prefix FE80::/64.



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:

```

      0                               1
    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5
+---+---+---+---+---+---+---+---+
|           0x00           | 0xFF |
+---+---+---+---+---+---+---+---+

```

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:

```

      0                               1
    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5
+---+---+---+---+---+---+---+---+
|           0x00           | MS/TP address |
+---+---+---+---+---+---+---+---+

```

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/StdAddenda/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 is 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 IJKLMNOPQRSTUVWXYZ
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 ijklmnopqrstuvwxyz
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 IJKLMNOPQRSTUVWXYZ
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 ijklmnopqrstuvwxyz
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 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  @ABCDEFGHJKLMNO
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  `abcdefghijklmnopqrstuvwxyz{|}~.
70 71 72 73 74 75 76 77 78 79 7a 7b 7c 7d 7e 7f  .....
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 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  @ABCDEFGHJKLMNO
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  `abcdefghijklmnopqrstuvwxyz{|}~.
70 71 72 73 74 75 76 77 78 79 7a 7b 7c 7d 7e 7f  .....
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@deltaccontrols.com

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

Email: stuart.donaldson@honeywell.com

6Lo Working Group
Internet-Draft
Intended status: Standards Track
Expires: February 5, 2016

J. Nieminen
T. Savolainen
M. Isomaki
Nokia
B. Patil
AT&T
Z. Shelby
Arm
C. Gomez
Universitat Politecnica de Catalunya/i2CAT
August 4, 2015

IPv6 over BLUETOOTH(R) Low Energy
draft-ietf-6lo-btle-17

Abstract

Bluetooth Smart is the brand name for the Bluetooth low energy feature in the Bluetooth specification defined by the Bluetooth Special Interest Group. The standard Bluetooth radio has been widely implemented and available in mobile phones, notebook computers, audio headsets and many other devices. The low power version of Bluetooth is a specification that enables the use of this air interface with devices such as sensors, smart meters, appliances, etc. The low power variant of Bluetooth has been standardized since revision 4.0 of the Bluetooth specifications, although version 4.1 or newer is required for IPv6. This document describes how IPv6 is transported over Bluetooth low energy using IPv6 over Low-power Wireless Personal Area Network (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 <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 February 5, 2016.

Copyright Notice

Copyright (c) 2015 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	2
1.1. Terminology and Requirements Language	3
2. Bluetooth Low Energy	3
2.1. Bluetooth LE stack	4
2.2. Link layer roles and topology	5
2.3. Bluetooth LE device addressing	6
2.4. Bluetooth LE packet sizes and MTU	6
3. Specification of IPv6 over Bluetooth Low Energy	7
3.1. Protocol stack	7
3.2. Link model	8
3.2.1. IPv6 subnet model and Internet connectivity	9
3.2.2. Stateless address autoconfiguration	10
3.2.3. Neighbor discovery	12
3.2.4. Header compression	13
3.2.4.1. Remote destination example	14
3.2.4.2. Example of registration of multiple-addresses	15
3.2.5. Unicast and Multicast address mapping	16
4. IANA Considerations	16
5. Security Considerations	16
6. Additional contributors	17
7. Acknowledgements	17
8. References	18
8.1. Normative References	18
8.2. Informative References	19
Authors' Addresses	20

1. Introduction

Bluetooth Smart is the brand name for the Bluetooth low energy feature (hereinafter, Bluetooth LE) in the Bluetooth specification defined by the Bluetooth Special Interest Group. Bluetooth LE is a

radio technology targeted for devices that operate with very low capacity (e.g., coin cell) batteries or minimalistic power sources, which means that low power consumption is essential. Bluetooth LE is especially attractive technology for Internet of Things applications, such as health monitors, environmental sensing, proximity applications and many others.

Considering the potential for the exponential growth in the number of sensors and Internet connected devices, IPv6 is an ideal protocol for communication with such devices due to the large address space it provides. In addition, IPv6 provides tools for stateless address autoconfiguration, which is particularly suitable for sensor network applications and nodes which have very limited processing power or lack a full-fledged operating system.

This document describes how IPv6 is transported over Bluetooth LE connections using IPv6 over Low power Wireless Personal Area Networks (6LoWPAN) techniques. RFCs 4944, 6282, and 6775 [RFC4944][RFC6282][RFC6775] developed for 6LoWPAN specify the transmission of IPv6 over IEEE 802.15.4 [fifteendotfour]. The Bluetooth LE link in many respects has similar characteristics to that of IEEE 802.15.4 and many of the mechanisms defined for the IPv6 over IEEE 802.15.4 can be applied to the transmission of IPv6 on Bluetooth LE links. This document specifies the details of IPv6 transmission over Bluetooth LE links.

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 2.2) can both be either 6LN or 6LBR.

2. Bluetooth Low Energy

Bluetooth LE is designed for transferring small amounts of data infrequently at modest data rates with a very small energy expenditure per bit. Bluetooth Special Interest Group (Bluetooth SIG) has introduced two trademarks, Bluetooth Smart for single-mode devices (a device that only supports Bluetooth LE) and Bluetooth Smart Ready for dual-mode devices (devices that support both Bluetooth and Bluetooth LE; note that Bluetooth and Bluetooth LE are different, non-interoperable radio technologies). In the rest of the

document, the term Bluetooth LE is used regardless of whether this technology is supported by a single-mode or dual-mode device.

Bluetooth LE was introduced in Bluetooth 4.0, enhanced in Bluetooth 4.1 [BTCorev4.1], and developed even further in successive versions. Bluetooth SIG has also published the Internet Protocol Support Profile (IPSP) [IPSP], which includes the Internet Protocol Support Service (IPSS). The IPSP enables discovery of IP-enabled devices and establishment of a link layer connection for transporting IPv6 packets. IPv6 over Bluetooth LE is dependent on both Bluetooth 4.1 and IPSP 1.0 or more recent versions of either specification to provide necessary capabilities.

Devices such as mobile phones, notebooks, tablets and other handheld computing devices that incorporate chipsets implementing Bluetooth 4.1 or later will also have the low-energy functionality of Bluetooth. Bluetooth LE is also expected to 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.

2.1. Bluetooth LE stack

The lower layer of the Bluetooth LE stack consists of the Physical (PHY), the Link Layer (LL), and a test interface called the Direct Test Mode (DTM). The Physical Layer transmits and receives the actual packets. The Link Layer is responsible for providing medium access, connection establishment, error control and flow control. The Direct Test Mode is only used for testing purposes. The upper layer consists of the Logical Link Control and Adaptation Protocol (L2CAP), Attribute Protocol (ATT), Security Manager (SM), Generic Attribute Profile (GATT) and Generic Access Profile (GAP) as shown in Figure 1. The Host Controller Interface (HCI) separates the lower layers, often implemented in the Bluetooth controller, from higher layers, often implemented in the host stack. GATT and Bluetooth LE profiles together enable the creation of applications in a standardized way without using IP. L2CAP provides multiplexing capability by multiplexing the data channels from the above layers. L2CAP also provides fragmentation and reassembly for large data packets. The Security Manager defines a protocol and mechanisms for pairing, key distribution and a security toolbox for the Bluetooth LE device.

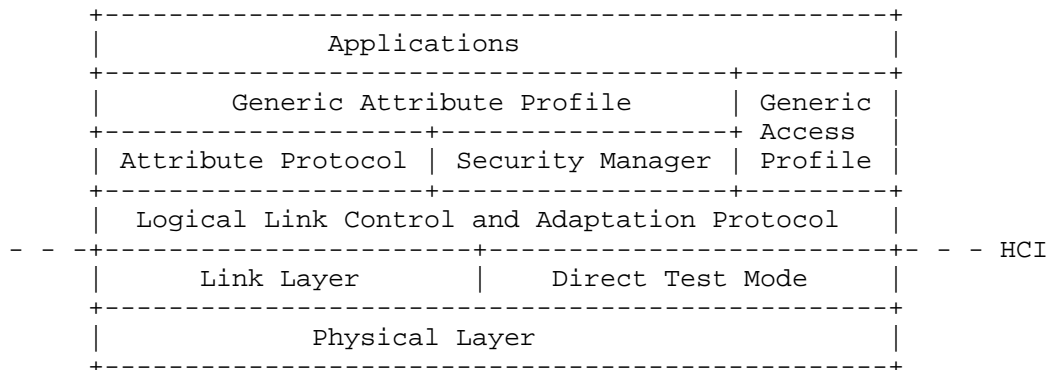


Figure 1: Bluetooth LE Protocol Stack

As shown in Section 3.1, IPv6 over Bluetooth LE requires an adapted 6LoWPAN layer which runs on top of Bluetooth LE L2CAP.

2.2. Link layer roles and topology

Bluetooth LE defines two 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 from now on. A peripheral is commonly connected to a single central, but with versions of Bluetooth from 4.1 onwards it can also connect to multiple centrals at the same time. In this document for IPv6 networking purposes the Bluetooth LE network (i.e., a Bluetooth LE piconet) follows a star topology shown in the Figure 2, where a router typically implements the Bluetooth LE central role and the rest of nodes implement the Bluetooth LE peripheral role. In the future mesh networking and/or parallel connectivity to multiple centrals at a time may be defined for IPv6 over Bluetooth LE.

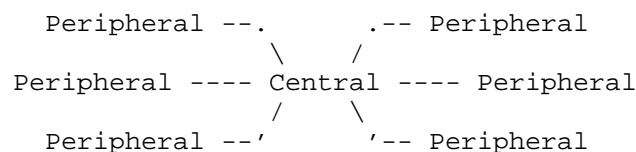


Figure 2: Bluetooth LE Star Topology

In Bluetooth LE, direct wireless communication only takes place between a central and a peripheral. This means that inherently the Bluetooth LE star represents a hub and spokes link model.

Nevertheless, two peripherals may communicate through the central by using IP routing functionality per this specification.

2.3. Bluetooth LE device addressing

Every Bluetooth LE device is identified by a 48-bit device address. The Bluetooth specification describes the device address of a Bluetooth LE device as: "Devices are identified using a device address. Device addresses may be either a public device address or a random device address." [BTCorev4.1]. The public device addresses are based on the IEEE 802-2001 standard [IEEE802-2001]. Random device addresses and Bluetooth LE privacy feature are described in Bluetooth Generic Access Profile specification sections 10.8 and 10.7, respectively [BTCorev4.1]. There are two types of random device addresses: static and private addresses. The private addresses are further divided into two sub-types: resolvable or non-resolvable addresses, which are explained in depth in the referenced Bluetooth specification. Once a static address is initialized, it does not change until the device is power cycled. The static address can be initialized to a new value after each power cycle, but that is not mandatory. Recommended time interval before randomizing new private address is 15 minutes, as determined by timer T_GAP(private_addr_int) at Bluetooth Generic Access Profile Table 17.1. The selection of which device address types are used is implementation and deployment specific. In random addresses first 46 bits are randomized and last 2 bits indicate the random address type. Bluetooth LE does not support device address collision avoidance or detection. However, these 48 bit random device addresses have a very small probability of being in conflict within a typical deployment.

2.4. Bluetooth LE packet sizes and MTU

The optimal MTU defined for L2CAP fixed channels over Bluetooth LE is 27 octets including the L2CAP header of 4 octets. The default MTU for Bluetooth LE is hence defined to be 27 octets. Therefore, excluding the L2CAP header of 4 octets, a protocol data unit (PDU) size of 23 octets is available for upper layers. In order to be able to transmit IPv6 packets of 1280 octets or larger, a link layer fragmentation and reassembly solution is provided by the L2CAP layer. The IPSP defines means for negotiating up a link layer connection that provides an MTU of 1280 octets or higher for the IPv6 layer [IPSP]. The link layer MTU is negotiated separately for each direction. Implementations that require an equal link layer MTU for the two directions SHALL use the smallest of the possibly different MTU values.

3. Specification of IPv6 over Bluetooth Low Energy

Bluetooth LE technology sets strict requirements for low power consumption and thus limits the allowed protocol overhead. 6LoWPAN standards [RFC6775], and [RFC6282] provide useful functionality for reducing overhead, which are applied to Bluetooth LE. This functionality is comprised of link-local IPv6 addresses and stateless IPv6 address autoconfiguration (see Section 3.2.2), Neighbor Discovery (see Section 3.2.3), and header compression (see Section 3.2.4). Fragmentation features from 6LoWPAN standards are not used due to Bluetooth LE's link layer fragmentation support (see Section 2.4).

A significant difference between IEEE 802.15.4 and Bluetooth LE is that the former supports both star and mesh topologies (and requires a routing protocol), whereas Bluetooth LE does not currently support the formation of multihop networks at the link layer. However, inter-peripheral communication through the central is enabled by using IP routing functionality per this specification.

In Bluetooth LE a central node is assumed to be less resource constrained than a peripheral node. Hence, in the primary deployment scenario central and peripheral will act as 6LoWPAN Border Router (6LBR) and a 6LoWPAN Node (6LN), respectively.

Before any IP-layer communications can take place over Bluetooth LE, Bluetooth LE enabled nodes such as 6LNs and 6LBRs have to find each other and establish a suitable link layer connection. The discovery and Bluetooth LE connection setup procedures are documented by the Bluetooth SIG in the IPSP specification [IPSP].

In the rare case of Bluetooth LE random device address conflict, a 6LBR can detect multiple 6LNs with the same Bluetooth LE device address, as well as a 6LN with the same Bluetooth LE address as the 6LBR. The 6LBR MUST ignore 6LNs with the same device address the 6LBR has, and the 6LBR MUST have at most one connection for a given Bluetooth LE device address at any given moment. This will avoid addressing conflicts within a Bluetooth LE network.

3.1. Protocol stack

Figure 3 illustrates how the IPv6 stack works in parallel to the GATT stack on top of Bluetooth LE L2CAP layer. The GATT stack is needed herein for discovering nodes supporting the Internet Protocol Support Service. UDP and TCP are provided as examples of transport protocols, but the stack can be used by any other upper layer protocol capable of running atop of IPv6.

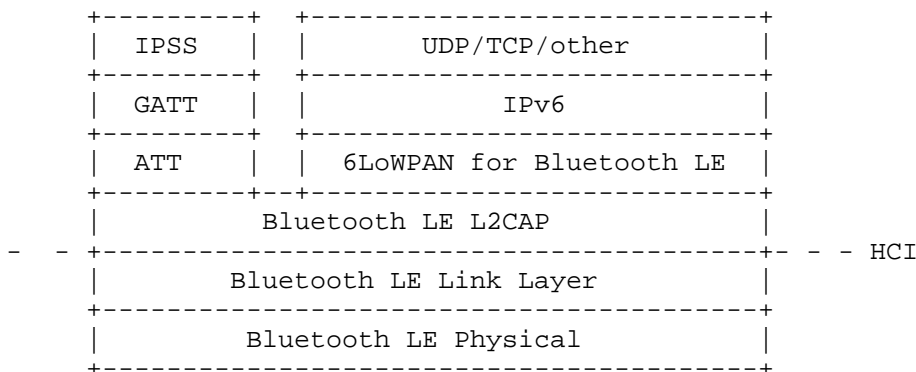


Figure 3: IPv6 and IPSS on the Bluetooth LE Stack

3.2. Link model

The distinct concepts of the IPv6 link (layer 3) and the physical link (combination of PHY and MAC) need to be clear and their relationship has to be well understood in order to specify the addressing scheme for transmitting IPv6 packets over the Bluetooth LE link. RFC 4861 [RFC4861] defines a link as "a communication facility or medium over which nodes can communicate at the link layer, i.e., the layer immediately below IPv6."

In the case of Bluetooth LE, the 6LoWPAN layer is adapted to support transmission of IPv6 packets over Bluetooth LE. The IPSP defines all steps required for setting up the Bluetooth LE connection over which 6LoWPAN can function [IPSP], including handling the link layer fragmentation required on Bluetooth LE, as described in Section 2.4. Even though MTUs larger than 1280 octets can be supported, use of a 1280 octet MTU is RECOMMENDED in order to avoid need for Path MTU discovery procedures.

While Bluetooth LE protocols, such as L2CAP, utilize little-endian byte ordering, IPv6 packets MUST be transmitted in big endian order (network byte order).

Per this specification, the IPv6 header compression format specified in RFC 6282 MUST be used [RFC6282]. The IPv6 payload length can be derived from the L2CAP header length and the possibly elided IPv6 address can be reconstructed from the link layer address, used at the time of Bluetooth LE connection establishment, from the HCI Connection Handle during connection, compression context if any, and from address registration information (see Section 3.2.3).

Bluetooth LE connections used to build a star topology are point-to-point in nature, as Bluetooth broadcast features are not used for IPv6 over Bluetooth LE (except for discovery of nodes supporting IPSS). After the peripheral and central have connected at the Bluetooth LE level, the link can be considered up and IPv6 address configuration and transmission can begin.

3.2.1. IPv6 subnet model and Internet connectivity

In the Bluetooth LE piconet model (see Section 2.2) peripherals each have a separate link to the central and the central acts as an IPv6 router rather than a link layer switch. As discussed in [RFC4903], conventional usage of IPv6 anticipates IPv6 subnets spanning a single link at the link layer. 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 the Bluetooth LE case, 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].

Hence a multilink model has been chosen, as further illustrated in Figure 4. Because of this, link-local multicast communications can happen only within a single Bluetooth LE connection, and thus 6LN-to-6LN communications using link-local addresses are not possible. 6LNs connected to the same 6LBR have to communicate with each other by using the shared prefix used on the subnet. The 6LBR ensures address collisions do not occur (see Section 3.2.3) and forwards packets sent by one 6LN to another.

In a typical scenario, the Bluetooth LE network is connected to the Internet as shown in the Figure 4. In this scenario, the Bluetooth LE star is deployed as one subnet, using one /64 IPv6 prefix, with each spoke representing individual link. The 6LBR is acting as router and forwarding packets between 6LNs and to and from Internet.

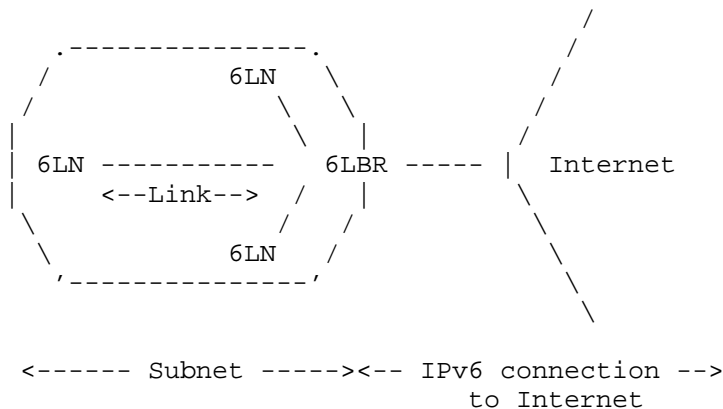


Figure 4: Bluetooth LE network connected to the Internet

In some scenarios, the Bluetooth LE network may transiently or permanently be an isolated network as shown in the Figure 5. In this case the whole star consist of a single subnet with multiple links, where 6LBR is at central routing packets between 6LNs. In simplest case the isolated network has one 6LBR and one 6LN.

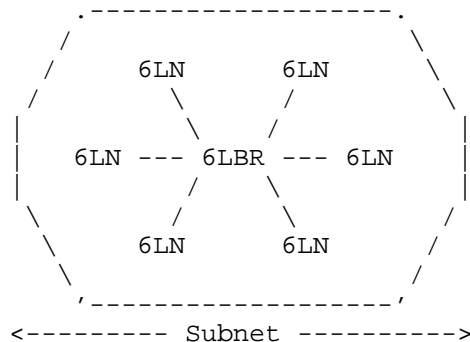


Figure 5: Isolated Bluetooth LE network

3.2.2. Stateless address autoconfiguration

At network interface initialization, both 6LN and 6LBR SHALL generate and assign to the Bluetooth LE network interface IPv6 link-local addresses [RFC4862] based on the 48-bit Bluetooth device addresses (see Section 2.3) that were used for establishing the underlying Bluetooth LE connection. A 6LN and a 6LBR are RECOMMENDED to use private Bluetooth device addresses. A 6LN SHOULD pick a different

Bluetooth device address for every Bluetooth LE connection with a 6LBR, and a 6LBR SHOULD periodically change its random Bluetooth device address. Following the guidance of [RFC7136], a 64-bit Interface Identifier (IID) is formed from the 48-bit Bluetooth device address by inserting two octets, with hexadecimal values of 0xFF and 0xFE in the middle of the 48-bit Bluetooth device address as shown in Figure 6. In the Figure letter 'b' represents a bit from the Bluetooth device address, copied as is without any changes on any bit. This means that no bit in the IID indicates whether the underlying Bluetooth device address is public or random.

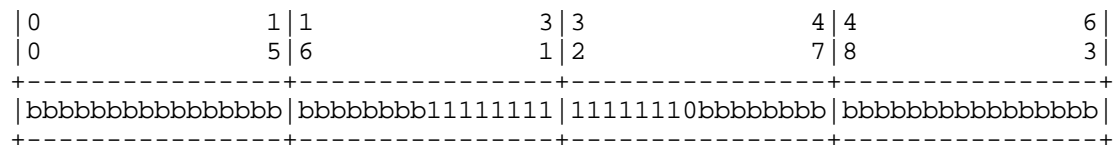


Figure 6: Formation of IID from Bluetooth device address

The IID is then prepended with the prefix fe80::/64, as described in RFC 4291 [RFC4291] and as depicted in Figure 7. The same link-local address SHALL be used for the lifetime of the Bluetooth LE L2CAP channel. (After a Bluetooth LE logical link has been established, it is referenced with a Connection Handle in HCI. Thus possibly changing device addresses do not impact data flows within existing L2CAP channels. Hence there is no need to change IPv6 link-local addresses even if devices change their random device addresses during L2CAP channel lifetime).

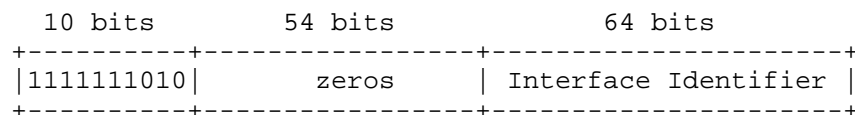


Figure 7: IPv6 link-local address in Bluetooth LE

A 6LN MUST join the all-nodes multicast address. There is no need for 6LN to join the solicited-node multicast address, since 6LBR will know device addresses and hence link-local addresses of all connected 6LNs. The 6LBR will ensure no two devices with the same Bluetooth LE device address are connected at the same time. Detection of duplicate link-local addresses is performed by the process on the 6LBR responsible for the discovery of IP-enabled Bluetooth LE nodes and for starting Bluetooth LE connection establishment procedures.

This approach increases the complexity of 6LBR, but reduces power consumption on both 6LN and 6LBR in the link establishment phase by reducing the number of mandatory packet transmissions.

After link-local address configuration, the 6LN sends Router Solicitation messages as described in [RFC4861] Section 6.3.7.

For non-link-local addresses, 6LNs SHOULD NOT be configured to embed the Bluetooth device address in the IID by default. Alternative schemes such as Cryptographically Generated Addresses (CGA) [RFC3972], privacy extensions [RFC4941], Hash-Based Addresses (HBA, [RFC5535]), DHCPv6 [RFC3315], or static, semantically opaque addresses [RFC7217] SHOULD be used by default. In situations where the Bluetooth device address is known to be a private device address and/or the header compression benefits of embedding the device address in the IID are required to support deployment constraints, 6LNs MAY form a 64-bit IID by utilizing the 48-bit Bluetooth device address. The non-link-local addresses that a 6LN generates MUST be registered with the 6LBR as described in Section 3.2.3.

The tool for a 6LBR to obtain an IPv6 prefix for numbering the Bluetooth LE network is out of scope of this document, but can be, for example, accomplished via DHCPv6 Prefix Delegation [RFC3633] or by using Unique Local IPv6 Unicast Addresses (ULA) [RFC4193]. Due to the link model of the Bluetooth LE (see Section 3.2.1) the 6LBR MUST set the "on-link" flag (L) to zero in the Prefix Information Option in Neighbor Discovery messages [RFC4861] (see Section 3.2.3). This will cause 6LNs to always send packets to the 6LBR, including the case when the destination is another 6LN using the same prefix.

3.2.3. 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. Bluetooth LE does not support mesh networks and hence only those aspects that apply to a star topology are considered.

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 the 6LBR 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. 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 (see Section 3.2.4).

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].

3.2.4. 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.

The Bluetooth LE's star topology structure and ARO can be exploited in order to provide a mechanism for address compression. The following text describes the principles of IPv6 address compression on top of Bluetooth LE.

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].

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.

When a 6LN is sending a packet to a 6LBR, it MUST fully elide the source address if it is a link-local address. For other packets to or through a 6LBR with a non-link-local source address that the 6LN has registered with ARO to the 6LBR 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. If a source non-link-local address is not the latest registered, then the 64-bits of the IID SHALL be fully carried in-line (SAM=01) or if the first 48-bits of the IID match with the latest registered address, then the last 16-bits of the IID SHALL be carried in-line (SAM=10). That is, if SAC=0 and SAM=11 the 6LN MUST be using the link-local IPv6 address derived from Bluetooth LE device address, and if SAC=1 and SAM=11 the 6LN MUST have registered the source IPv6 address with the prefix related to the compression context and the 6LN MUST be referring to the latest registered address related to the compression context. The IPv6 address MUST be considered to be registered only after the

6LBR has sent a Neighbor Advertisement with an ARO having its status field set to success. The destination IPv6 address MUST be fully elided if the destination address is 6LBR's link-local-address based on the 6LBR's Bluetooth device address (DAC=0, DAM=11). The destination IPv6 address MUST be fully or partially elided if context has been set up for the destination address. For example, DAC=0 and DAM=01 when destination prefix is link-local, and DAC=1 and DAM=01 if compression context has been configured for the destination prefix used.

When a 6LBR is transmitting packets to a 6LN, it MUST fully elide the source IID if the source IPv6 address is the link-local address based on the 6LBR's Bluetooth device address (SAC=0, SAM=11), and it MUST elide the source prefix or address if a compression context related to the IPv6 source address has been set up. The 6LBR also MUST fully elide the destination IPv6 address if it is the link-local-address based on the 6LN's Bluetooth device address (DAC=0, DAM=11), or 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).

3.2.4.1. Remote destination example

When a 6LN transmits an IPv6 packet to a remote destination using global Unicast IPv6 addresses, if a context is defined for the 6LN's global IPv6 address, the 6LN has to indicate this context in the corresponding source fields of the compressed IPv6 header as per Section 3.1 of RFC 6282 [RFC6282], and has to elide the full IPv6 source address previously registered with ARO (if using the latest registered address, otherwise part or all of the IID may have to be transmitted in-line). For this, the 6LN MUST use the following settings in the IPv6 compressed header: SAC=1 and SAM=11. The CID may be set 0 or 1, depending on which context is used. In this case, the 6LBR can infer the elided IPv6 source address since 1) the 6LBR has previously assigned the prefix to the 6LNs; and 2) the 6LBR maintains a Neighbor Cache that relates the Device Address and the IID the device has registered with ARO. If a context is defined for the IPv6 destination address, the 6LN has to also indicate this context in the corresponding destination fields of the compressed IPv6 header, and elide the prefix of or the full destination IPv6 address. For this, the 6LN MUST set the DAM field of the compressed IPv6 header as DAM=01 (if the context covers a 64-bit prefix) or as DAM=11 (if the context covers a full, 128-bit address). DAC MUST be set to 1. Note that when a context is defined for the IPv6

destination address, the 6LBR can infer the elided destination prefix by using the context.

When a 6LBR receives an IPv6 packet sent by a remote node outside the Bluetooth LE network, and the destination of the packet is a 6LN, if a context is defined for the prefix of the 6LN's global IPv6 address, the 6LBR has to indicate this context in the corresponding destination fields of the compressed IPv6 header. The 6LBR has to elide the IPv6 destination address of the packet before forwarding it, if the IPv6 destination address is inferable by the 6LN. For this, the 6LBR will set the DAM field of the IPv6 compressed header as DAM=11 (if the address is the latest 6LN has registered). DAC needs to be set to 1. If a context is defined for the IPv6 source address, the 6LBR needs to indicate this context in the source fields of the compressed IPv6 header, and elide that prefix as well. For this, the 6LBR needs to set the SAM field of the IPv6 compressed header as SAM=01 (if the context covers a 64-bit prefix) or SAM=11 (if the context covers a full, 128-bit address). SAC is to be set to 1.

3.2.4.2. Example of registration of multiple-addresses

As described above, a 6LN can register multiple non-link-local addresses that map to a same compression context. From the multiple address registered, only the latest address can be fully elided (SAM=11, DAM=11), and the IIDs of previously registered addresses have to be transmitted fully in-line (SAM=01, DAM=01) or in the best case can be partially elided (SAM=10, DAM=10). This is illustrated in an example below.

1) A 6LN registers first address 2001:db8::1111:2222:3333:4444 to a 6LBR. At this point the address can be fully elided using SAC=1/SAM=11 or DAC=1/DAM=11.

2) The 6LN registers second address 2001:db8::1111:2222:3333:5555 to the 6LBR. As the second address is now the latest registered, it can be fully elided using SAC=1/SAM=11 or DAC=1/DAM=11. The first address can now be partially elided using SAC=1/SAM=10 or DAC=1/DAM=10, as the first 112 bits of the address are the same between the first and the second registered addresses.

3) Expiration of registration time for the first or the second address has no impact on the compression. Hence even if the most recently registered address expires, the first address can only be partially elided (SAC=1/SAM=10, DAC=1/DAM=10). The 6LN can register a new address, or re-register an expired address, to become able to again fully elide an address.

3.2.5. Unicast and Multicast address mapping

The Bluetooth LE link layer does not support multicast. Hence traffic is always unicast between two Bluetooth LE nodes. 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 may not be energy-efficient and particular care must be taken if the central is battery-powered. To further conserve power, the 6LBR MUST keep track of multicast listeners at Bluetooth LE 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 6LBR. Hence, when a 6LN needs to transmit an IPv6 multicast packet, the 6LN will unicast the corresponding Bluetooth LE packet to the 6LBR.

4. IANA Considerations

There are no IANA considerations related to this document.

5. Security Considerations

The transmission of IPv6 over Bluetooth LE links has similar requirements and concerns for security as for IEEE 802.15.4. Bluetooth LE Link Layer security considerations are covered by the IPSP [IPSP].

Bluetooth LE Link Layer supports encryption and authentication by using the Counter with CBC-MAC (CCM) mechanism [RFC3610] and a 128-bit AES block cipher. Upper layer security mechanisms may exploit this functionality when it is available. (Note: CCM does not consume octets from the maximum per-packet L2CAP data size, since the link layer data unit has a specific field for them when they are used.)

Key management in Bluetooth LE is provided by the Security Manager Protocol (SMP), as defined in [BTCorev4.1].

The Direct Test Mode offers two setup alternatives: with and without accessible HCI. In designs with accessible HCI, the so called upper tester communicates through the HCI (which may be supported by Universal Asynchronous Receiver Transmitter (UART), Universal Serial Bus (USB) and Secure Digital transports), with the Physical and Link Layers of the Bluetooth LE device under test. In designs without accessible HCI, the upper tester communicates with the device under test through a two-wire UART interface. The Bluetooth specification

does not provide security mechanisms for the communication between the upper tester and the device under test in either case. Nevertheless, an attacker needs to physically connect a device (via one of the wired HCI types) to the device under test to be able to interact with the latter.

The IPv6 link-local address configuration described in Section 3.2.2 only reveals information about the 6LN to the 6LBR that the 6LBR already knows from the link layer connection. This means that a device using Bluetooth privacy features reveals the same information in its IPv6 link-local addresses as in its device addresses. Respectively, device not using privacy at Bluetooth level will not have privacy at IPv6 link-local address either. For non-link local addresses implementations have a choice to support, for example, [I-D.ietf-6man-default-iids], [RFC3315], [RFC3972], [RFC4941], [RFC5535], or [RFC7217].

A malicious 6LN may attempt to perform a denial of service attack on the Bluetooth LE network, for example, by flooding packets. This sort of attack is mitigated by the fact that link-local multicast is not bridged between Bluetooth LE links and by 6LBR being able to rate limit packets sent by each 6LN by making smart use of Bluetooth LE L2CAP credit-based flow control mechanism.

6. Additional contributors

Kanji Kerai, Jari Mutikainen, David Canfeng-Chen and Minjun Xi from Nokia have contributed significantly to this document.

7. Acknowledgements

The Bluetooth, Bluetooth Smart and Bluetooth Smart Ready marks are registered trademarks owned by Bluetooth SIG, Inc.

Carsten Bormann, Samita Chakrabarti, Niclas Comstedt, Alissa Cooper, Elwyn Davies, Brian Haberman, Marcel De Kogel, Jouni Korhonen, Chris Lonvick, Erik Nordmark, Erik Rivard, Dave Thaler, Pascal Thubert, Xavi Vilajosana and Victor Zhodzishsky have provided valuable feedback for this draft.

Authors would like to give special acknowledgements for Krishna Shingala, Frank Berntsen, and Bluetooth SIG's Internet Working Group for providing significant feedback and improvement proposals for this document.

8. References

8.1. 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>>.
- [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>>.
- [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>>.

8.2. Informative References

- [fifteendotfour]
IEEE Computer Society, "IEEE Std. 802.15.4-2011 IEEE Standard for Local and metropolitan area networks--Part 15.4: Low-Rate Wireless Personal Area Networks (LR-WPANs)", June 2011.
- [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-05 (work in progress), July 2015.
- [IEEE802-2001]
Institute of Electrical and Electronics Engineers (IEEE), "IEEE 802-2001 Standard for Local and Metropolitan Area Networks: Overview and Architecture", 2002.
- [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>>.
- [RFC3610] Whiting, D., Housley, R., and N. Ferguson, "Counter with CBC-MAC (CCM)", RFC 3610, DOI 10.17487/RFC3610, September 2003, <<http://www.rfc-editor.org/info/rfc3610>>.
- [RFC3633] Troan, O. and R. Droms, "IPv6 Prefix Options for Dynamic Host Configuration Protocol (DHCP) version 6", RFC 3633, DOI 10.17487/RFC3633, December 2003, <<http://www.rfc-editor.org/info/rfc3633>>.
- [RFC3972] Aura, T., "Cryptographically Generated Addresses (CGA)", RFC 3972, DOI 10.17487/RFC3972, March 2005, <<http://www.rfc-editor.org/info/rfc3972>>.
- [RFC4193] Hinden, R. and B. Haberman, "Unique Local IPv6 Unicast Addresses", RFC 4193, DOI 10.17487/RFC4193, October 2005, <<http://www.rfc-editor.org/info/rfc4193>>.
- [RFC4903] Thaler, D., "Multi-Link Subnet Issues", RFC 4903, DOI 10.17487/RFC4903, June 2007, <<http://www.rfc-editor.org/info/rfc4903>>.

- [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>>.
- [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>>.

Authors' Addresses

Johanna Nieminen
Nokia

Email: johannamaria.nieminen@gmail.com

Teemu Savolainen
Nokia
Visiokatu 3
Tampere 33720
Finland

Email: teemu.savolainen@nokia.com

Markus Isomaki
Nokia
Otaniementie 19
Espoo 02150
Finland

Email: markus.isomaki@nokia.com

Basavaraj Patil
AT&T
1410 E. Renner Road
Richardson, TX 75082
USA

Email: basavaraj.patil@att.com

Zach Shelby
Arm
Hallituskatu 13-17D
Oulu 90100
Finland

Email: zach.shelby@arm.com

Carles Gomez
Universitat Politecnica de Catalunya/i2CAT
C/Esteve Terradas, 7
Castelldefels 08860
Spain

Email: carlesgo@entel.upc.edu

6Lo Working Group
Internet-Draft
Intended status: Standards Track
Expires: June 18, 2017

P. Mariager
J. Petersen, Ed.
RTX A/S
Z. Shelby
ARM
M. Van de Logt
Gigaset Communications GmbH
D. Barthel
Orange Labs
December 15, 2016

Transmission of IPv6 Packets over DECT Ultra Low Energy
draft-ietf-6lo-dect-ule-09

Abstract

Digital Enhanced Cordless Telecommunications (DECT) Ultra Low Energy (ULE) is a low power air interface technology that is defined by the DECT Forum and specified by ETSI.

The DECT air interface technology has been used world-wide in communication devices for more than 20 years, primarily carrying voice for cordless telephony but has also been deployed for data centric services.

The DECT Ultra Low Energy is a recent addition to the DECT interface primarily intended for low-bandwidth, low-power applications such as sensor devices, smart meters, home automation etc. As the DECT Ultra Low Energy interface inherits many of the capabilities from DECT, it benefits from long range, interference free operation, world wide reserved frequency band, low silicon prices and maturity. There is an added value in the ability to communicate with IPv6 over DECT ULE such as for Internet of Things applications.

This document describes how IPv6 is transported over DECT ULE 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 <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 June 18, 2017.

Copyright Notice

Copyright (c) 2016 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	3
1.1. Requirements Notation	4
1.2. Terms Used	4
2. DECT Ultra Low Energy	6
2.1. The DECT ULE Protocol Stack	6
2.2. Link Layer Roles and Topology	7
2.3. Addressing Model	8
2.4. MTU Considerations	9
2.5. Additional Considerations	9
3. Specification of IPv6 over DECT ULE	9
3.1. Protocol Stack	10
3.2. Link Model	10
3.3. Subnets and Internet Connectivity Scenarios	15
4. IANA Considerations	17
5. Security Considerations	17
6. ETSI Considerations	18
7. Acknowledgements	18
8. References	18
8.1. Normative References	18
8.2. Informative References	20
Authors' Addresses	21

1. Introduction

Digital Enhanced Cordless Telecommunications (DECT) is a standard series [EN300.175-part1-7] specified by ETSI and CAT-iq (Cordless Advanced Technology - internet and quality) is a set of product certification and interoperability profiles [CAT-iq] defined by DECT Forum. DECT Ultra Low Energy (DECT ULE or just ULE) is an air interface technology building on the key fundamentals of traditional DECT / CAT-iq but with specific changes to significantly reduce the power consumption at the expense of data throughput. DECT ULE devices with requirements on power consumption as specified by ETSI in [TS102.939-1] and [TS102.939-2], will operate on special power optimized silicon, but can connect to a DECT Gateway supporting traditional DECT / CAT-iq for cordless telephony and data as well as the ULE extensions.

DECT terminology has two major role definitions: The Portable Part (PP) is the power constrained device, while the Fixed Part (FP) is the Gateway or base station. This FP may be connected to the Internet. An example of a use case for DECT ULE is a home security sensor transmitting small amounts of data (few bytes) at periodic intervals through the FP, but is able to wake up upon an external event (burglar) and communicate with the FP. Another example incorporating both DECT ULE as well as traditional CAT-iq telephony is a pendant (brooch) for the elderly which can transmit periodic status messages to a care provider using very little battery, but in the event of urgency, the elderly person can establish a voice connection through the pendant to an alarm service. It is expected that DECT ULE will be integrated into many residential gateways, as many of these already implement DECT CAT-iq for cordless telephony. DECT ULE can be added as a software option for the FP.

It is desirable to consider IPv6 for DECT ULE devices due to the large address space and well-known infrastructure. This document describes how IPv6 is used on DECT ULE links to optimize power while maintaining the many benefits of IPv6 transmission. [RFC4944], [RFC6282] and [RFC6775] specify the transmission of IPv6 over IEEE 802.15.4. DECT ULE has many characteristics similar to those of IEEE 802.15.4, but also differences. A subset of mechanisms defined for transmission of IPv6 over IEEE 802.15.4 can be applied to the transmission of IPv6 on DECT ULE links.

This document specifies how to map IPv6 over DECT ULE inspired by [RFC4944], [RFC6282], [RFC6775] and [RFC7668].

1.1. Requirements Notation

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

1.2. Terms Used

6CO	6LoWPAN Context Option [RFC6775]
6BBR	6LoWPAN Backbone Router
6LBR	6LoWPAN Border Router as defined in [RFC6775]. The DECT Fixed Part is having this role
6LN	6LoWPAN Node as defined in [RFC6775]. The DECT Portable part is having this role
6LoWPAN	IPv6 over Low-Power Wireless Personal Area Network
AES128	Advanced Encryption Standard with key size of 128 bits
API	Application Programming Interface
ARO	Address Registration Option [RFC6775]
CAT-iq	Cordless Advanced Technology - internet and quality
CID	Context Identifier [RFC6775]
DAC	Destination Address Compression
DAD	Duplicate Address Detection [RFC4862]
DAM	Destination Address Mode
DHCPv6	Dynamic Host Configuration Protocol for IPv6 [RFC3315]
DLC	Data Link Control
DSAA2	DECT Standard Authentication Algorithm #2
DSC	DECT Standard Cipher
DSC2	DECT Standard Cipher #2
FDMA	Frequency Division Multiplex
FP	DECT Fixed Part, the gateway
GAP	Generic Access Profile
IID	Interface Identifier
IPEI	International Portable Equipment Identity; (DECT identity)
MAC-48	48 bit global unique MAC address managed by IEEE
MAC	Media Access Control
MTU	Maximum Transmission Unit
NBMA	Non-broadcast multi-access
ND	Neighbor Discovery [RFC4861] [RFC6775]
PDU	Protocol Data Unit
PHY	Physical Layer
PMID	Portable MAC Identity; (DECT identity)
PP	DECT Portable Part, typically the sensor node (6LN)
PVC	Permanent Virtual Circuit
RFPI	Radio Fixed Part Identity; (DECT identity)
SAC	Source Address Compression
SAM	Source Address Mode
TDD	Time Division Duplex
TDMA	Time Division Multiplex
TPUI	Temporary Portable User Identity; (DECT identity)
UAK	User Authentication Key, DECT master security key
ULA	Unique Local Address [RFC4193]

2. DECT Ultra Low Energy

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. This draft is only addressing the packet mode data service of DECT ULE.

2.1. The DECT ULE Protocol Stack

The DECT ULE protocol stack contains a PHY layer operating at frequencies in the 1880 - 1920 MHz frequency band depending on the region and uses a symbol rate of 1.152 Mbaud. 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 FP defining the network with a number of PP attached. The MAC layer supports both traditional DECT circuit mode operation as this is used for services like discovery, pairing, security features etc, and it supports new ULE packet mode operation. The circuit mode 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, this document is not considering usage of the DECT ULE MAC layer broadcast service for IPv6 over DECT ULE.

In general, communication sessions can be initiated from both FP and PP side. Depending on power down modes employed in the PP, latency may occur when initiating sessions from FP side. MAC layer communication can take place using either connection oriented packet transfer with low overhead for short sessions or take place using connection oriented bearers including media reservation. The MAC layer autonomously selects the radio spectrum positions that are available within the band and can rearrange these to avoid interference. The MAC layer has built-in retransmission procedures in order to improve transmission reliability.

The DECT ULE device will typically incorporate an application programming interface (API) as well as common elements known as Generic Access Profile (GAP) for enrolling into the network. The

DECT ULE stack establishes a permanent virtual circuit (PVC) for the application layers and provides support for a range of different application protocols. The application protocol is negotiated between the PP and FP when the PVC communication service is established. [TS102.939-1] defines this negotiation and specifies an Application Protocol Identifier of 0x06 for 6LowPAN. This document defines the behavior of that Application Protocol.

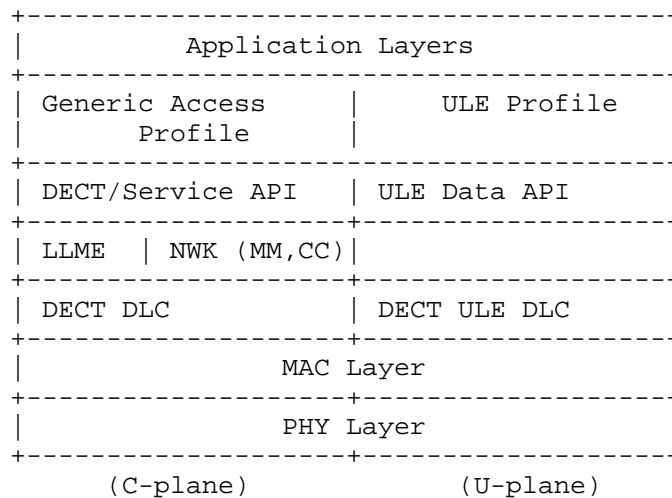


Figure 1: DECT ULE Protocol Stack

Figure 1 above shows the DECT ULE Stack divided into the Control-plane and User-data plane, to left and to the right, respectively. The shown entities in the Stack are the (PHY) Physical Layer, (MAC) Media Access Control Layer, (DLC) Data Link Control Layer, (NWK) Network Layer with subcomponents: (LLME) Lower Layer Management Entity, (MM) Mobility Management and (CC) Call Control. Above there are the typically (API) Application Programmers Interface and application profile specific layers.

2.2. Link Layer Roles and Topology

A FP is assumed to be less constrained than a PP. Hence, in the primary scenario FP and PP will act as 6LBR and a 6LN, respectively. This document only addresses this primary scenario and all other scenarios with different roles of FP and PP are out of scope.

In DECT ULE, at link layer the communication only takes place between a FP and a PP. A FP is able to handle multiple simultaneous

connections with a number of PP. Hence, in a DECT ULE network using IPv6, a radio hop is equivalent to an IPv6 link and vice versa (see Section 3.3).

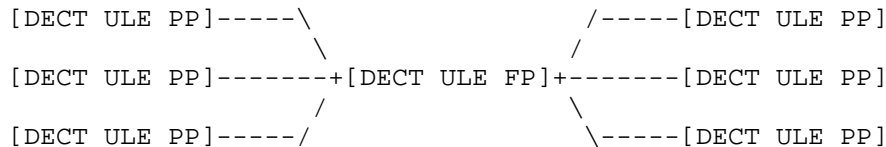


Figure 2: DECT ULE star topology

A significant difference between IEEE 802.15.4 and DECT ULE is that the former supports both star and mesh topology (and requires a routing protocol), whereas DECT ULE in its primary configuration does not support the formation of multihop networks at the link layer. In consequence, the mesh header defined in [RFC4944] is not used in DECT ULE networks.

DECT ULE repeaters are considered to operate transparently in the DECT protocol domain and are outside the scope of this document.

2.3. Addressing Model

Each DECT PP is assigned an IPEI during manufacturing. This identity has the size of 40 bits and is globally unique within DECT addressing space and can be used to constitute the MAC address used to derive the IID for link-local address.

During a DECT location registration procedure, the FP assigns a 20 bit TPUI to a PP. The FP creates a unique mapping between the assigned TPUI and the IPEI of each PP. This TPUI is used for addressing (layer 2) in messages between FP and PP. Although the TPUI is temporary by definition, many implementations assign the same value repeatedly to any given PP, hence it seems not suitable for construction of IID, see [I-D.ietf-6lo-privacy-considerations].

Each DECT FP is assigned a RFPI during manufacturing. This identity has the size of 40 bits and is globally unique within DECT addressing space and can be used to constitute the MAC address used to derive the IID for link-local address.

Optionally each DECT PP and DECT FP can be assigned a unique (IEEE) MAC-48 address additionally to the DECT identities to be used by the

6LoWPAN. During the address registration of non-link-local addresses as specified by this document, the FP and PP can use such MAC-48 to construct the IID. However, as these addresses are considered as being permanent, such scheme is NOT RECOMMENDED as per [I-D.ietf-6lo-privacy-considerations].

2.4. MTU Considerations

Ideally the DECT ULE FP and PP may generate data that fits into a single MAC Layer packets (38 octets) for periodically transferred information, depending on application. However, IP packets may be much larger. The DECT ULE DLC procedures natively support segmentation and reassembly and provide any MTU size below 65536 octets. The default MTU size defined in DECT ULE [TS102.939-1] is 500 octets. In order to support complete IPv6 packets, the DLC layer of DECT ULE SHALL per this specification be configured with a MTU size of 1280 octets, hence [RFC4944] fragmentation/reassembly is not required.

It is important to realize that the usage of larger packets will be at the expense of battery life, as a large packet inside the DECT ULE stack will be fragmented into several or many MAC layer packets, each consuming power to transmit / receive. The increased MTU size does not change the MAC layer packet and PDU size.

2.5. Additional Considerations

The DECT ULE standard allows PP to be DECT-registered (bound) to multiple FP and to roam between them. These FP and their 6LBR functionalities can either operate individually or connected through a Backbone Router as per [I-D.ietf-6lo-backbone-router].

3. Specification of IPv6 over DECT ULE

Before any IP-layer communications can take place over DECT ULE, DECT ULE enabled nodes such as 6LNs and 6LBRs have to find each other and establish a suitable link-layer connection. The obtain-access-rights registration and location registration procedures are documented by ETSI in the specifications [EN300.175-part1-7], [TS102.939-1] and [TS102.939-2].

DECT ULE technology sets strict requirements for low power consumption and thus limits the allowed protocol overhead. 6LoWPAN standards [RFC4944], [RFC6775], and [RFC6282] provide useful functionality for reducing overhead which can be applied to DECT ULE. This functionality comprises link-local IPv6 addresses and stateless IPv6 address autoconfiguration, Neighbor Discovery and header compression.

The ULE 6LoWPAN adaptation layer can run directly on this U-plane DLC layer. Figure 3 illustrates IPv6 over DECT ULE stack.

Because DECT ULE in its primary configuration does not support the formation of multihop networks at the link layer, the mesh header defined in [RFC4944] for mesh under routing MUST NOT be used. In addition, the role of a 6LoWPAN Router (6LR) is not defined per this specification.

3.1. Protocol Stack

In order to enable data transmission over DECT ULE, a Permanent Virtual Circuit (PVC) has to be configured and opened between FP and PP. This is done by setting up a DECT service call between PP and FP. In DECT protocol domain the PP SHALL specify the <<IWU-ATTRIBUTES>> in a service-change (other) message before sending a service-change (resume) message as defined in [TS102.939-1]. The <<IWU-ATTRIBUTES>> SHALL define the ULE Application Protocol Identifier to 0x06 and the MTU size to 1280 octets or larger. The FP sends a service-change-accept (resume) that MUST contain a valid paging descriptor. The PP MUST listen to paging messages from the FP according to the information in the received paging descriptor. Following this, transmission of IPv6 packets can start.

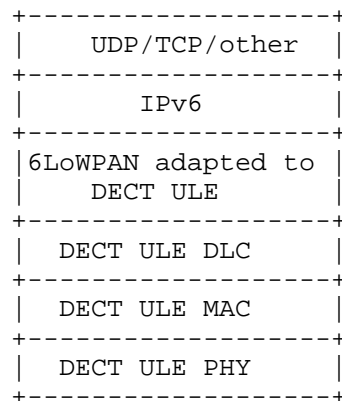


Figure 3: IPv6 over DECT ULE Stack

3.2. Link Model

The general model is that IPv6 is layer 3 and DECT ULE MAC+DLC is layer 2. The DECT ULE already implements fragmentation and

reassembly functionality, hence [RFC4944] fragmentation and reassembly function MUST NOT be used.

After the FP and PPs have connected at the DECT ULE level, the link can be considered up and IPv6 address configuration and transmission can begin. The 6LBR ensures address collisions do not occur.

Per this specification, the IPv6 header compression format specified in [RFC6282] MUST be used. The IPv6 payload length can be derived from the ULE DLC packet length and the possibly elided IPv6 address can be reconstructed from the link-layer address, used at the time of DECT ULE connection establishment, from the ULE MAC packet address, compression context if any, and from address registration information (see Section 3.2.2).

Due to the DECT ULE star topology (see Section 2.2), each PP has a separate link to the FP, and thus the PPs cannot directly hear one another and cannot talk to one another. As discussed in [RFC4903], conventional usage of IPv6 anticipates IPv6 subnets spanning a single link at the link layer. In order avoid the complexity of implementing separate subnet for each DECT ULE link, a Multi-Link Subnet model [RFC4903] has been chosen, specifically Non-broadcast multi-access (NBMA) at layer 2. Because of this, link-local multicast communications can happen only within a single DECT ULE connection; thus, 6LN-to-6LN communications using link-local addresses are not possible. 6LNs connected to the same 6LBR have to communicate with each other by using the shared prefix used on the subnet. The 6LBR forwards packets sent by one 6LN to another.

3.2.1. Stateless Address Autoconfiguration

At network interface initialization, both 6LN and 6LBR SHALL generate and assign to the DECT ULE network interface IPv6 link-local addresses [RFC4862] based on the DECT device addresses (see Section 2.3) that were used for establishing the underlying DECT ULE connection.

The DECT device addresses IPEI and RFPI MUST be used to derive the IPv6 link-local 64 bit Interface Identifiers (IID) for 6LN and 6LBR, respectively.

The rule for deriving IID from DECT device addresses is as follows: The DECT device addresses that are consisting of 40 bits each, MUST be expanded with leading zero bits to form 48 bit intermediate addresses. Most significant bit in this newly formed 48-bit intermediate address is set to one for addresses derived from the RFPI and set to zero for addresses derived from the IPEI. From these intermediate 48 bit addresses are derived 64 bit IIDs following the

guidance in Appendix A of [RFC4291]. However, because DECT and IEEE address spaces are different, this intermediate address cannot be considered as unique within IEEE address space. In the derived IIDs the U/L bit (7th bit) will be zero, indicating that derived IID's are not globally unique, see [RFC7136]. For example from RFPI=11.22.33.44.55 the derived IID is 80:11:22:ff:fe:33:44:55 and from IPEI=01.23.45.67.89 the derived IID is 00:01:23:ff:fe:45:67:89.

Globally uniqueness of IID in link-local addresses are not required as they should never be leaked outside the subnet domain.

As defined in [RFC4291], the IPv6 link-local address is formed by appending the IID, to the prefix FE80::/64, as shown in Figure 4.

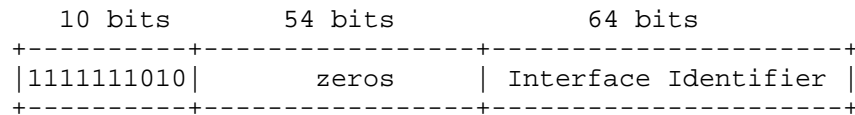


Figure 4: IPv6 link-local address in DECT ULE

A 6LN MUST join the all-nodes multicast address.

After link-local address configuration, 6LN sends Router Solicitation messages as described in [RFC4861] Section 6.3.7 and [RFC6775] Section 5.3.

For non-link-local addresses, 6LNs SHOULD NOT be configured to use IIDs derived from a MAC-48 device address or DECT device addresses. Alternative schemes such as Cryptographically Generated Addresses (CGAs) [RFC3972], privacy extensions [RFC4941], Hash-Based Addresses (HBAs) [RFC5535], DHCPv6 [RFC3315], or static, semantically opaque addresses [RFC7217] SHOULD be used by default. See also [I-D.ietf-6lo-privacy-considerations] for guidance of needed entropy in IIDs and recommended lifetime of used IIDs. When generated IID's are not globally unique, Duplicate Address Detection (DAD) [RFC4862] MUST be used. In situations where deployment constraints require the device's address to be embedded in the IID, the 6LN MAY form a 64-bit IID by utilizing the MAC-48 device address or DECT device addresses. The non-link-local addresses that a 6LN generates MUST be registered with 6LBR as described in Section 3.2.2.

The means for a 6LBR to obtain an IPv6 prefix for numbering the DECT ULE network is out of scope of this document, but can be, for example, accomplished via DHCPv6 Prefix Delegation [RFC3633] or by using Unique Local IPv6 Unicast Addresses (ULA) [RFC4193]. Due to

the link model of the DECT ULE the 6LBR MUST set the "on-link" flag (L) to zero in the Prefix Information Option [RFC4861]. This will cause 6LNs to always send packets to the 6LBR, including the case when the destination is another 6LN using the same prefix.

3.2.2. 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. As DECT ULE does not support mesh networks, only those aspects of [RFC6775] that apply to star topology are considered.

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

1. For sending Router Solicitations and processing Router Advertisements the DECT ULE 6LNs MUST, respectively, follow Sections 5.3 and 5.4 of the [RFC6775].
2. A DECT ULE 6LN MUST NOT register its link-local address. Because the IIDs used in link-local addresses are derived from DECT addresses, there will always exist a unique mapping between link-local and layer-2 addresses.
3. A DECT ULE 6LN MUST register its non-link-local addresses with the 6LBR 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.

3.2.3. Unicast and Multicast Address Mapping

The DECT MAC layer broadcast service is considered inadequate for IP multicast, because it does not support the MTU size required by IPv6.

Hence traffic is always unicast between two DECT ULE nodes. 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 may not be energy-efficient and particular care should be taken if the FP is battery-powered. To further conserve power, the 6LBR MUST keep track of multicast listeners at DECT-ULE link level granularity and it MUST NOT forward multicast packets to 6LNs that have not registered for multicast groups the packets belong to. In the opposite direction, a 6LN can only transmit data to or through the 6LBR. Hence, when a 6LN

needs to transmit an IPv6 multicast packet, the 6LN will unicast the corresponding DECT ULE packet to the 6LBR. The 6LBR will then forward the multicast packet to other 6LNs.

3.2.4. 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 DECT ULE. All headers MUST be compressed according to [RFC6282] encoding formats. The DECT ULE's star topology structure, ARO and 6CO can be exploited in order to provide a mechanism for address compression. The following text describes the principles of IPv6 address compression on top of DECT ULE.

3.2.4.1. Link-local Header Compression

In a link-local communication terminated at 6LN and 6LBR, both the IPv6 source and destination addresses MUST be elided, since the used IIDs map uniquely into the DECT link end point addresses. A 6LN or 6LBR that receives a PDU containing an IPv6 packet can infer the corresponding IPv6 source address. For the unicast type of communication considered in this paragraph, the following settings MUST be used in the IPv6 compressed header: CID=0, SAC=0, SAM=11, DAC=0, DAM=11.

3.2.4.2. Non-link-local Header Compression

To enable efficient header compression, the 6LBR MUST include 6LoWPAN Context Option (6CO) [RFC6775] for all prefixes the 6LBR advertises in Router Advertisements for use in stateless address autoconfiguration.

When a 6LN transmits an IPv6 packet to a destination using global Unicast IPv6 addresses, if a context is defined for the prefix of the 6LNs global IPv6 address, the 6LN MUST indicate this context in the corresponding source fields of the compressed IPv6 header as per Section 3.1 of [RFC6282], and MUST fully elide the latest registered IPv6 source address. For this, the 6LN MUST use the following settings in the IPv6 compressed header: CID=1, SAC=1, SAM=11. In this case, the 6LBR can infer the elided IPv6 source address since 1) the 6LBR has previously assigned the prefix to the 6LNs; and 2) the 6LBR maintains a Neighbor Cache that relates the Device Address and the IID of the corresponding PP. If a context is defined for the IPv6 destination address, the 6LN MUST also indicate this context in the corresponding destination fields of the compressed IPv6 header, and MUST elide the prefix of the destination IPv6 address. For this, the 6LN MUST set the DAM field of the compressed IPv6 header as

CID=1, DAC=1 and DAM=01 or DAM=11. Note that when a context is defined for the IPv6 destination address, the 6LBR can infer the elided destination prefix by using the context.

When a 6LBR receives a IPv6 packet having a global Unicast IPv6 address, and the destination of the packet is a 6LN, if a context is defined for the prefix of the 6LN's global IPv6 address, the 6LBR MUST indicate this context in the corresponding destination fields of the compressed IPv6 header, and MUST fully elide the IPv6 destination address of the packet if the destination address is the latest registered by the 6LN for the indicated context. For this, the 6LBR MUST set the DAM field of the IPv6 compressed header as DAM=11. CID and DAC MUST be set to CID=1 and DAC=1. If a context is defined for the prefix of the IPv6 source address, the 6LBR MUST indicate this context in the source fields of the compressed IPv6 header, and MUST elide that prefix as well. For this, the 6LBR MUST set the SAM field of the IPv6 compressed header as CID=1, SAC=1 and SAM=01 or SAM=11.

3.3. Subnets and Internet Connectivity Scenarios

In the DECT ULE star topology (see Section 2.2), PP each have a separate link to the FP and the FP acts as an IPv6 router rather than a link-layer switch. A Multi-Link Subnet model [RFC4903] has been chosen, specifically Non-broadcast multi-access (NBMA) at layer 2 as further illustrated in Figure 5. The 6LBR forwards packets sent by one 6LN to another. In a typical scenario, the DECT ULE network is connected to the Internet as shown in the Figure 5. In this scenario, the DECT ULE network is deployed as one subnet, using one /64 IPv6 prefix. The 6LBR is acting as router and forwarding packets between 6LNs and to and from Internet.

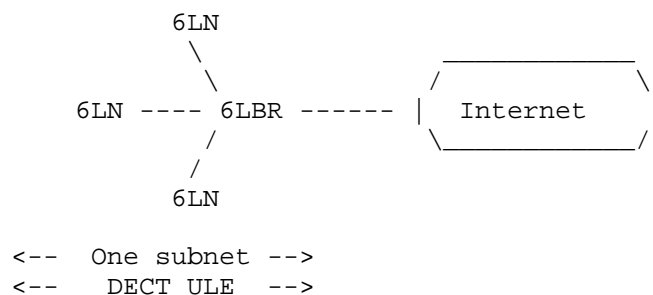


Figure 5: DECT ULE network connected to the Internet

In some scenarios, the DECT ULE network may transiently or permanently be an isolated network as shown in the Figure 6. In this case the whole DECT ULE network consists of a single subnet with multiple links, where 6LBR is routing packets between 6LNs.

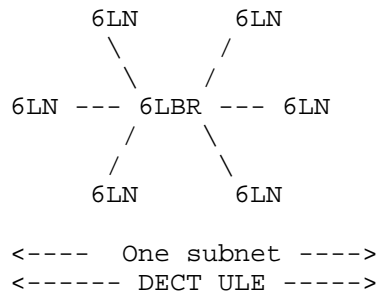


Figure 6: Isolated DECT ULE network

In the isolated network scenario, communications between 6LN and 6LBR can use IPv6 link-local methodology, but for communications between different PP, the FP has to act as 6LBR, number the network with ULA prefix [RFC4193], and route packets between PP.

In other more advanced systems scenarios with multiple FP and 6LBR, each DECT ULE FP constitutes a wireless cell. The network can be configured as a Multi-Link Subnet, in which the 6LN can operate within the same /64 subnet prefix in multiple cells as shown in the Figure 7. The FPs in such a scenario should behave as Backbone Routers (6BBR) as defined in [I-D.ietf-6lo-backbone-router].

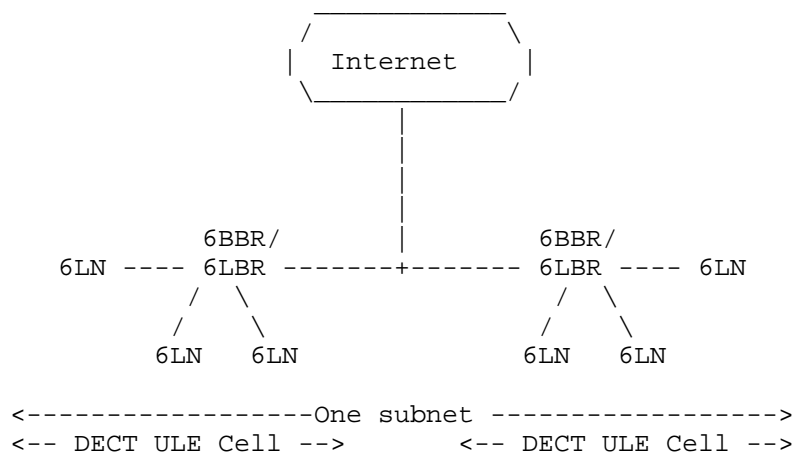


Figure 7: Multiple DECT ULE cells in a single Multi-Link subnet

4. IANA Considerations

There are no IANA considerations related to this document.

5. Security Considerations

The secure transmission of circuit mode services in DECT is based on the DSAA2 and DSC/DSC2 specifications developed by ETSI TC DECT and the ETSI SAGE Security expert group.

DECT ULE communications are secured at the link-layer (DLC) by encryption and per-message authentication through CCM mode (Counter with CBC-MAC) similar to [RFC3610]. The underlying algorithm for providing encryption and authentication is AES128.

The DECT ULE pairing procedure generates a master authentication key (UAK). During location registration procedure or when the permanent virtual circuit are established, the session security keys are generated. Both the master authentication key and session security keys are generated by use of the DSAA2 algorithm [EN300.175-part1-7], which is using AES128 as underlying algorithm. Session security keys may be renewed regularly. The generated security keys (UAK and session security keys) are individual for each FP-PP binding, hence all PP in a system have different security keys. DECT ULE PPs do not use any shared encryption key.

Even though DECT ULE offers link-layer security, it is still recommended to use secure transport or application protocols above 6LoWPAN.

From privacy point of view, the IPv6 link-local address configuration described in Section 3.2.1 only reveals information about the 6LN to the 6LBR that the 6LBR already knows from the link-layer connection. For non-link-local IPv6 addresses, by default a 6LN SHOULD use a randomly generated IID, for example, as discussed in [I-D.ietf-6man-default-iids], or use alternative schemes such as Cryptographically Generated Addresses (CGA) [RFC3972], privacy extensions [RFC4941], Hash-Based Addresses (HBA, [RFC5535]), or static, semantically opaque addresses [RFC7217].

6. ETSI Considerations

ETSI is standardizing a list of known application layer protocols that can use the DECT ULE permanent virtual circuit packet data service. Each protocol is identified by a unique known identifier, which is exchanged in the service-change procedure as defined in [TS102.939-1]. The IPv6/6LoWPAN as described in this document is considered as an application layer protocol on top of DECT ULE. In order to provide interoperability between 6LoWPAN / DECT ULE devices a common protocol identifier for 6LoWPAN is standardized by ETSI.

The ETSI DECT ULE Application Protocol Identifier is specified to 0x06 for 6LoWPAN [TS102.939-1].

7. Acknowledgements

We are grateful to the members of the IETF 6lo working group; this document borrows liberally from their work.

Ralph Droms, Samita Chakrabarti, Kerry Lynn, Suresh Krishnan, Pascal Thubert, Tatuya Jinmei, Dale Worley and Robert Sparks have provided valuable feedback for this draft.

8. References

8.1. Normative References

[EN300.175-part1-7]
ETSI, "Digital Enhanced Cordless Telecommunications (DECT); Common Interface (CI);", March 2015, <https://www.etsi.org/deliver/etsi_en/300100_300199/30017501/02.06.01_60/en_30017501v020601p.pdf>.

- [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>>.
- [RFC3633] Troan, O. and R. Droms, "IPv6 Prefix Options for Dynamic Host Configuration Protocol (DHCP) version 6", RFC 3633, DOI 10.17487/RFC3633, December 2003, <<http://www.rfc-editor.org/info/rfc3633>>.
- [RFC4193] Hinden, R. and B. Haberman, "Unique Local IPv6 Unicast Addresses", RFC 4193, DOI 10.17487/RFC4193, October 2005, <<http://www.rfc-editor.org/info/rfc4193>>.
- [RFC4291] Hinden, R. and S. Deering, "IP Version 6 Addressing Architecture", RFC 4291, DOI 10.17487/RFC4291, February 2006, <<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>>.
- [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>>.

[TS102.939-1]
ETSI, "Digital Enhanced Cordless Telecommunications (DECT); Ultra Low Energy (ULE); Machine to Machine Communications; Part 1: Home Automation Network (phase 1)", March 2015, <https://www.etsi.org/deliver/etsi_ts/102900_102999/10293901/01.02.01_60/ts_10293901v010201p.pdf>.

[TS102.939-2]
ETSI, "Digital Enhanced Cordless Telecommunications (DECT); Ultra Low Energy (ULE); Machine to Machine Communications; Part 2: Home Automation Network (phase 2)", March 2015, <https://www.etsi.org/deliver/etsi_ts/102900_102999/10293902/01.01.01_60/ts_10293902v010101p.pdf>.

8.2. Informative References

[CAT-iq] DECT Forum, "Cordless Advanced Technology - internet and quality", January 2016, <http://www.dect.org/userfiles/Public/DF_CAT-iq%20at%20a%20Glance.pdf>.

[I-D.ietf-6lo-backbone-router]
Thubert, P., "IPv6 Backbone Router", draft-ietf-6lo-backbone-router-02 (work in progress), September 2016.

[I-D.ietf-6lo-privacy-considerations]
Thaler, D., "Privacy Considerations for IPv6 Adaptation Layer Mechanisms", draft-ietf-6lo-privacy-considerations-04 (work in progress), October 2016.

[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.

[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>>.

- [RFC3610] Whiting, D., Housley, R., and N. Ferguson, "Counter with CBC-MAC (CCM)", RFC 3610, DOI 10.17487/RFC3610, September 2003, <<http://www.rfc-editor.org/info/rfc3610>>.
- [RFC3972] Aura, T., "Cryptographically Generated Addresses (CGA)", RFC 3972, DOI 10.17487/RFC3972, March 2005, <<http://www.rfc-editor.org/info/rfc3972>>.
- [RFC4903] Thaler, D., "Multi-Link Subnet Issues", RFC 4903, DOI 10.17487/RFC4903, June 2007, <<http://www.rfc-editor.org/info/rfc4903>>.
- [RFC5535] Bagnulo, M., "Hash-Based Addresses (HBA)", RFC 5535, DOI 10.17487/RFC5535, June 2009, <<http://www.rfc-editor.org/info/rfc5535>>.
- [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>>.
- [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>>.

Authors' Addresses

Peter B. Mariager
RTX A/S
Stroemmen 6
DK-9400 Noerresundby
Denmark

Email: pm@rtx.dk

Jens Toftgaard Petersen (editor)
RTX A/S
Stroemmen 6
DK-9400 Noerresundby
Denmark

Email: jtp@rtx.dk

Zach Shelby
ARM
150 Rose Orchard
San Jose, CA 95134
USA

Email: zach.shelby@arm.com

Marco van de Logt
Gigaset Communications GmbH
Frankenstrasse 2
D-46395 Bocholt
Germany

Email: marco.van-de-logt@gigaset.com

Dominique Barthel
Orange Labs
28 chemin du Vieux Chene
38243 Meylan
France

Email: dominique.barthel@orange.com

6Lo Working Group
Internet-Draft
Intended status: Standards Track
Expires: March 23, 2015

C. Bormann
Universitaet Bremen TZI
September 19, 2014

6LoWPAN Generic Compression of Headers and Header-like Payloads (GHC)
draft-ietf-6lo-ghc-05

Abstract

This short specification provides a simple addition to 6LoWPAN Header Compression that enables the compression of generic headers and header-like payloads, without a need to define a new header compression scheme for each new such header or header-like payload.

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 March 23, 2015.

Copyright Notice

Copyright (c) 2014 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	2
1.1. The Header Compression Coupling Problem	2
1.2. Compression Approach	3
1.3. Terminology	3
1.4. Notation	4
2. 6LoWPAN-GHC	5
3. Integrating 6LoWPAN-GHC into 6LoWPAN-HC	6
3.1. Compressing payloads (UDP and ICMPv6)	6
3.2. Compressing extension headers	6
3.3. Indicating GHC capability	7
3.4. Using the 6CIO Option	8
4. IANA considerations	9
5. Security considerations	10
6. Acknowledgements	11
7. References	13
7.1. Normative References	13
7.2. Informative References	13
Appendix A. Examples	14
Author's Address	24

1. Introduction

1.1. The Header Compression Coupling Problem

6LoWPAN-HC [RFC6282] defines a scheme for header compression in 6LoWPAN [RFC4944] packets. As with most header compression schemes, a new specification is needed for every new kind of header that needs to be compressed. In addition, [RFC6282] does not define an extensibility scheme like the ROHC profiles defined in ROHC [RFC3095] [RFC5795]. This leads to the difficult situation that 6LoWPAN-HC tended to be reopened and reexamined each time a new header receives consideration (or an old header is changed and reconsidered) in the 6LoWPAN/roll/CoRE cluster of IETF working groups. While [RFC6282] finally got completed, the underlying problem remains unsolved.

The purpose of the present contribution is to plug into [RFC6282] as is, using its NHC (next header compression) concept. We add a slightly less efficient, but vastly more general form of compression for headers of any kind and even for header-like payloads such as those exhibited by routing protocols, DHCP, etc.: Generic Header Compression (GHC). The objective is an extremely simple specification that can be defined on a single page and implemented in a small number of lines of code, as opposed to a general data compression scheme such as that defined in [RFC1951].

1.2. Compression Approach

The basic approach of GHC's compression function is to define a bytecode for LZ77-style compression [LZ77]. The bytecode is a series of simple instructions for the decompressor to reconstitute the uncompressed payload. These instructions include:

- o appending bytes to the reconstituted payload that are literally given with the instruction in the compressed data
- o appending a given number of zero bytes to the reconstituted payload
- o appending bytes to the reconstituted payload by copying a contiguous sequence from the payload being reconstituted ("backreferencing")
- o an ancillary instruction for setting up parameters for the backreferencing instruction in "decompression variables"
- o a stop code (optional, see Section 3.2)

The buffer for the reconstituted payload ("destination buffer") is prefixed by a predefined dictionary that can be used in the backreferencing as if it were a prefix of the payload. This predefined dictionary is built from the IPv6 addresses of the packet being reconstituted, followed by a static component, the "static dictionary".

As usual, this specification defines the decompressor operation in detail, but leaves the detailed operation of the compressor open to implementation. The compressor can be implemented as with a classical LZ77 compressor, or it can be a simple protocol encoder that just makes use of known compression opportunities.

1.3. 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 RFC 2119 [RFC2119].

The term "byte" is used in its now customary sense as a synonym for "octet".

Terms from [RFC7228] are used in Section 5.

1.4. Notation

This specification uses a trivial notation for code bytes and the bitfields in them, the meaning of which should be mostly obvious. More formally, the meaning of the notation is:

Potential values for the code bytes themselves are expressed by templates that represent 8-bit most-significant-bit-first binary numbers (without any special prefix), where 0 stands for 0, 1 for 1, and variable segments in these code byte templates are indicated by sequences of the same letter such as kkkkkkkk or ssss, the length of which indicates the length of the variable segment in bits.

In the notation of values derived from the code bytes, 0b is used as a prefix for expressing binary numbers in most-significant-bit first notation (akin to the use of 0x for most-significant-digit-first hexadecimal numbers in the C programming language). Where the above-mentioned sequences of letters are then referenced in such a binary number in the text, the intention is that the value from these bitfields in the actual code byte be inserted.

Example: The code byte template

101nssss

stands for a byte that starts (most-significant-bit-first) with the bits 1, 0, and 1, and continues with five variable bits, the first of which is referenced as "n" and the next four are referenced as "ssss". Based on this code byte template, a reference to

0b0ssss000

means a binary number composed from a zero bit, the four bits that are in the "ssss" field (for 101nssss, the four least significant bits) in the actual byte encountered, kept in the same order, and three more zero bits.

2. 6LoWPAN-GHC

The format of a GHC-compressed header or payload is a simple bytecode. A compressed header consists of a sequence of pieces, each of which begins with a code byte, which may be followed by zero or more bytes as its argument. Some code bytes cause bytes to be laid out in the destination buffer, some simply modify some decompression variables.

At the start of decompressing a header or payload within a L2 packet (= fragment), the decompression variables "sa" and "na" are initialized as zero.

The code bytes are defined as follows (Table 1):

code byte	Action	Argument
0kkkkkkk	Append k = 0b0kkkkkkk bytes of data in the bytecode argument (k < 96)	k bytes of data
1000nnnn	Append 0b0000nnnn+2 bytes of zeroes	
10010000	STOP code (end of compressed data, see Section 3.2)	
101nssss	Set up extended arguments for a backreference: sa += 0b0ssss000, na += 0b0000n000	
11nnnnkkk	Backreference: n = na+0b00000nnn+2; s = 0b00000kkk+sa+n; append n bytes from previously output bytes, starting s bytes to the left of the current output pointer; set sa = 0, na = 0	

Table 1: Bytecodes for generic header compression

Note that the following bit combinations are reserved at this time: 011xxxxx, and 1001nnnn (where 0b0000nnnn > 0).

For the purposes of the backreferences, the expansion buffer is initialized with a predefined dictionary, at the end of which the reconstituted payload begins. This dictionary is composed of the source and destination IPv6 addresses of the packet being reconstituted, followed by a 16-byte static dictionary (Figure 1).

These 48 dictionary bytes are therefore available for backreferencing, but not copied into the final reconstituted payload.

```
16 fe fd 17 fe fd 00 01 00 00 00 00 00 01 00 00
```

Figure 1: The 16 bytes of static dictionary (in hex)

3. Integrating 6LoWPAN-GHC into 6LoWPAN-HC

6LoWPAN-GHC plugs in as an NHC format for 6LoWPAN-HC [RFC6282].

3.1. Compressing payloads (UDP and ICMPv6)

GHC is by definition generic and can be applied to different kinds of packets. Many of the examples given in Appendix A are for ICMPv6 packets; a single NHC value suffices to define an NHC format for ICMPv6 based on GHC (see below).

In addition it is useful to include an NHC format for UDP, as many headerlike payloads (e.g., DHCPv6, DTLS) are carried in UDP. [RFC6282] already defines an NHC format for UDP (11110CPP). GHC uses an analogous NHC byte formatted as shown in Figure 2. The difference to the existing UDP NHC specification is that for 0b11010cpp NHC bytes, the UDP payload is not supplied literally but compressed by 6LoWPAN-GHC.

0	1	2	3	4	5	6	7
1	1	0	1	0	C	P	

Figure 2: NHC byte for UDP GHC (to be allocated by IANA)

To stay in the same general numbering space, we use 0b11011111 as the NHC byte for ICMPv6 GHC (Figure 3).

0	1	2	3	4	5	6	7
1	1	0	1	1	1	1	1

Figure 3: NHC byte for ICMPv6 GHC (to be allocated by IANA)

3.2. Compressing extension headers

Compression of specific extension headers is added in a similar way (Figure 4) (however, probably only EID 0 to 3 need to be assigned). As there is no easy way to extract the length field from the GHC-

encoded header before decoding, this would make detecting the end of the extension header somewhat complex. The easiest (and most efficient) approach is to completely elide the length field (in the same way NHC already elides the next header field in certain cases) and reconstruct it only on decompression. To serve as a terminator for the extension header, the reserved bytecode 0b10010000 has been assigned as a stop marker. Note that the stop marker is only needed for extension headers, not for the final payloads discussed in the previous subsection, the decompression of which is automatically stopped by the end of the packet.

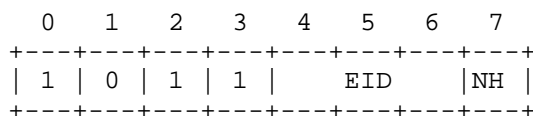


Figure 4: NHC byte for extension header GHC

3.3. Indicating GHC capability

The 6LoWPAN baseline includes just [RFC4944], [RFC6282], [RFC6775] (see [I-D.bormann-6lo-6lowpan-roadmap]). To enable the use of GHC towards a neighbor, a 6LoWPAN node needs to know that the neighbor implements it. While this can also simply be administratively required, a transition strategy as well as a way to support mixed networks is required.

One way to know a neighbor does implement GHC is receiving a packet from that neighbor with GHC in it ("implicit capability detection"). However, there needs to be a way to bootstrap this, as nobody ever would start sending packets with GHC otherwise.

To minimize the impact on [RFC6775], we define an ND option 6LoWPAN Capability Indication (6CIO), as illustrated in Figure 5. (For the fields marked by an underscore in Figure 5, see Section 3.4.)

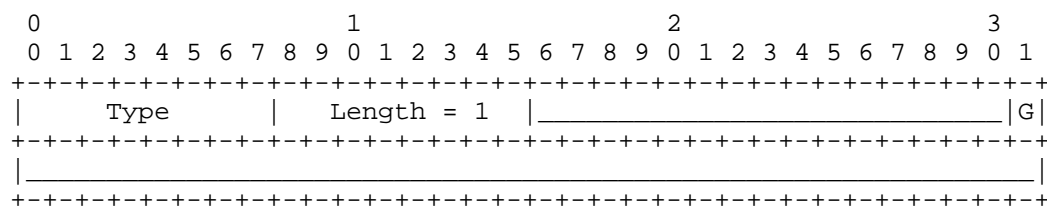


Figure 5: 6LoWPAN Capability Indication Option (6CIO)

The G bit indicates whether the node sending the option is GHC capable.

Once a node receives either an explicit or an implicit indication of GHC capability from another node, it may send GHC-compressed packets to that node. Where that capability has not been recently confirmed, similar to the way PLPMTUD [RFC4821] finds out about changes in the network, a node SHOULD make use of NUD (neighbor unreachability detection) failures to switch back to basic 6LoWPAN header compression [RFC6282].

3.4. Using the 6CIO Option

The 6CIO option will typically only be ever sent in 6LoWPAN-ND RS packets (which cannot itself be GHC compressed unless the host desires to limit itself to talking to GHC capable routers). The resulting 6LoWPAN-ND RA can then already make use of GHC and thus indicate GHC capability implicitly, which in turn allows both nodes to use GHC in the 6LoWPAN-ND NS/NA exchange.

6CIO can also be used for future options that need to be negotiated between 6LoWPAN peers; an IANA registry is used to assign the flags. Bits marked by underscores in Figure 5 are unassigned and available for future assignment. They MUST be sent as zero and MUST be ignored on reception until assigned by IANA. Length values larger than 1 MUST be accepted by implementations in order to enable future extensions; the additional bits in the option are then deemed unassigned in the same way. For the purposes of the IANA registry, the bits are numbered in most-significant-bit-first order from the 16th bit of the option onward: the 16th bit is flag number 0, the 31st bit (the G bit) is flag number 15, up to the 63rd bit for flag number 47. (Additional bits may also be used by a follow-on version of this document if some bit combinations that have been left unassigned here are then used in an upward compatible manner.)

Flag numbers 0 to 7 are reserved for experiments. They MUST NOT be used for actual deployments.

Where the use of this option by other specifications or by experiments is envisioned, the following items have to be kept in mind:

- o The option can be used in any ND packet.
- o Specific bits are set in the option to indicate that a capability is present in the sender. (There may be other ways to infer this information, as is the case in this specification.) Bit combinations may be used as desired. The absence of the capability `_indication_` is signaled by setting these bits to zero; this does not necessarily mean that the capability is absent.

- o The intention is not to modify the semantics of the specific ND packet carrying the option, but to provide the general capability indication described above.
- o Specifications have to be designed such that receivers that do not receive or do not process such a capability indication can still interoperate (presumably without exploiting the indicated capability).
- o The option is meant to be used sparsely, i.e. once a sender has reason to believe the capability indication has been received, there no longer is a need to continue sending it.

4. IANA considerations

[This section to be removed/replaced by the RFC Editor.]

In the IANA registry for the "LOWPAN_NHC Header Type" (in the "IPv6 Low Power Personal Area Network Parameters"), IANA is requested to add the assignments in Figure 6.

10110IIN: Extension header GHC	[RFCthis]
11010CPP: UDP GHC	[RFCthis]
11011111: ICMPv6 GHC	[RFCthis]

Figure 6: IANA assignments for the NHC byte

IANA is requested to allocate an ND option number for the "6LoWPAN Capability Indication Option (6CIO)" ND option format in the Registry "IPv6 Neighbor Discovery Option Formats" [RFC4861].

IANA is requested to create a subregistry for "6LoWPAN capability bits" within the "Internet Control Message Protocol version 6 (ICMPv6) Parameters". The bits are assigned by giving their numbers as small non-negative integers as defined in section Section 3.4, preferably in the range 0..47. The policy is "IETF Review" or "IESG Approval" [RFC5226]. The initial content of the registry is as in Figure 7:

0..7: reserved for experiments	[RFCthis]
8..14: unassigned	
15: GHC capable bit (G bit)	[RFCthis]
16..47: unassigned	

Figure 7: IANA assignments for the 6LoWPAN capability bits

5. Security considerations

The security considerations of [RFC4944] and [RFC6282] apply. As usual in protocols with packet parsing/construction, care must be taken in implementations to avoid buffer overflows and in particular (with respect to the back-referencing) out-of-area references during decompression.

One additional consideration is that an attacker may send a forged packet that makes a second node believe a third victim node is GHC-capable. If it is not, this may prevent packets sent by the second node from reaching the third node (at least until robustness features such as those discussed in Section 3.3 kick in).

No mitigation is proposed (or known) for this attack, except that a victim node that does implement GHC is not vulnerable. However, with unsecured ND, a number of attacks with similar outcomes are already possible, so there is little incentive to make use of this additional attack. With secured ND, 6CIO is also secured; nodes relying on secured ND therefore should use 6CIO bidirectionally (and limit the implicit capability detection to secured ND packets carrying GHC) instead of basing their neighbor capability assumptions on receiving any kind of unprotected packet.

As with any LZ77 scheme, decompression bombs (compressed packets crafted to expand so much that the decompressor is overloaded) are a problem. An attacker cannot send a GHC decompressor into a tight loop for too long, because the MTU will be reached quickly. Some amplification of an attack from inside the compressed link is possible, though. Using a constrained node in a constrained network as a DoS attack source is usually not very useful, though, except maybe against other nodes in that constrained network. The worst case for an attack to the outside is a not-so-constrained device using a (typically not-so-constrained) edge router to attack by forwarding out of its Ethernet interface. The worst-case amplification of GHC is 17, so an MTU-size packet can be generated from a 6LoWPAN packet of 76 bytes. The 6LoWPAN network is still constrained, so the amplification at the edge router turns an entire 250 kbit/s 802.15.4 network (assuming a theoretical upper bound of 225 kbit/s throughput to a single-hop adjacent node) into a 3.8 Mbit/s attacker.

The amplification may be more important inside the 6LoWPAN, if there is a way to obtain reflection (otherwise the packet is likely to simply stay compressed on the way and do little damage), e.g., by pinging a node using a decompression bomb, somehow keeping that node from re-compressing the ping response (which would probably require something more complex than simple runs of zeroes, so the worst-case

amplification is likely closer to 9). Or, if there are nodes that do not support GHC, those can be attacked via a router that is then forced to decompress.

All these attacks are mitigated by some form of network access control.

In a 6LoWPAN stack, sensitive information will normally be protected by transport or application (or even IP) layer security, which are all above the adaptation layer, leaving no sensitive information to compress at the GHC level. However, a 6LoWPAN deployment that entirely depends on MAC layer security may be vulnerable to attacks that exploit redundancy information disclosed by compression to recover information about secret values. The attacker would need to be in radio range to observe the compressed packets. Since compression is stateless, the attacker would need to entice the party sending the secret value to also send some value controlled (or at least usefully varying and knowable) by the attacker in (what becomes the first adaptation layer fragment of) the same packet. This attack is fully mitigated by not exposing secret values to the adaptation layer, or by not using GHC in deployments where this is done.

6. Acknowledgements

Colin O'Flynn has repeatedly insisted that some form of compression for ICMPv6 and ND packets might be beneficial. He actually wrote his own draft, [I-D.oflynn-6lowpan-icmphc], which compresses better, but addresses basic ICMPv6/ND only and needs a much longer spec (around 17 pages of detailed spec, as compared to the single page of core spec here). This motivated the author to try something simple, yet general. Special thanks go to Colin for indicating that he indeed considers his draft superseded by the present one.

The examples given are based on pcap files that Colin O'Flynn, Owen Kirby, Olaf Bergmann and others provided.

Using these pcap files as a corpus, the static dictionary was developed, and the bit allocations validated, based on research by Sebastian Dominik.

Erik Nordmark provided input that helped shaping the 6CIO option. Thomas Bjorklund proposed simplifying the predefined dictionary.

Yoshihiro Ohba insisted on clarifying the notation used for the definition of the bytecodes and their bitfields. Ulrich Herberg provided some additional review and suggested expanding the introductory material, and with Hannes Tschofenig and Brian Haberman

he helped come up with the IANA policy for the "6LoWPAN capability bits" assignments in the 6CIO option.

The IESG reviewers Richard Barnes and Stephen Farrell have contributed issues to the security considerations section; they and Barry Leiba, as well as GEN-ART reviewer Vijay K. Gurbani also have provided editorial improvements.

7. References

7.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC4861] Narten, T., Nordmark, E., Simpson, W., and H. Soliman, "Neighbor Discovery for IP version 6 (IPv6)", RFC 4861, September 2007.
- [RFC4944] Montenegro, G., Kushalnagar, N., Hui, J., and D. Culler, "Transmission of IPv6 Packets over IEEE 802.15.4 Networks", RFC 4944, September 2007.
- [RFC5226] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 5226, May 2008.
- [RFC6282] Hui, J. and P. Thubert, "Compression Format for IPv6 Datagrams over IEEE 802.15.4-Based Networks", RFC 6282, September 2011.
- [RFC6775] Shelby, Z., Chakrabarti, S., Nordmark, E., and C. Bormann, "Neighbor Discovery Optimization for IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs)", RFC 6775, November 2012.

7.2. Informative References

- [I-D.bormann-6lo-6lowpan-roadmap] Bormann, C., "6LoWPAN Roadmap and Implementation Guide", draft-bormann-6lo-6lowpan-roadmap-00 (work in progress), October 2013.
- [I-D.oflynn-6lowpan-icmphc] O'Flynn, C., "ICMPv6/ND Compression for 6LoWPAN Networks", draft-oflynn-6lowpan-icmphc-00 (work in progress), July 2010.
- [LZ77] Ziv, J. and A. Lempel, "A Universal Algorithm for Sequential Data Compression", IEEE Transactions on Information Theory, Vol. 23, No. 3, pp. 337-343, May 1977.
- [RFC1951] Deutsch, P., "DEFLATE Compressed Data Format Specification version 1.3", RFC 1951, May 1996.

- [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, July 2001.
- [RFC4821] Mathis, M. and J. Heffner, "Packetization Layer Path MTU Discovery", RFC 4821, March 2007.
- [RFC5795] Sandlund, K., Pelletier, G., and L-E. Jonsson, "The RObust Header Compression (ROHC) Framework", RFC 5795, March 2010.
- [RFC7228] Bormann, C., Ersue, M., and A. Keranen, "Terminology for Constrained-Node Networks", RFC 7228, May 2014.

Appendix A. Examples

This section demonstrates some relatively realistic examples derived from actual PCAP dumps taken at previous interops.

Figure 8 shows an RPL DODAG Information Solicitation, a quite short RPL message that obviously cannot be improved much.

```

IP header:
 60 00 00 00 00 08 3a ff fe 80 00 00 00 00 00 00
 02 1c da ff fe 00 20 24 ff 02 00 00 00 00 00 00
 00 00 00 00 00 00 00 00 1a
Payload:
 9b 00 6b de 00 00 00 00
Dictionary:
 fe 80 00 00 00 00 00 00 02 1c da ff fe 00 20 24
 ff 02 00 00 00 00 00 00 00 00 00 00 00 00 00 1a
 16 fe fd 17 fe fd 00 01 00 00 00 00 00 01 00 00
copy: 04 9b 00 6b de
4 nulls: 82
Compressed:
 04 9b 00 6b de 82
Was 8 bytes; compressed to 6 bytes, compression factor 1.33

```

Figure 8: A simple RPL example

Figure 9 shows an RPL DODAG Information Object, a longer RPL control message that is improved a bit more. Note that the compressed output exposes an inefficiency in the simple-minded compressor used to generate it; this does not devalue the example since constrained nodes are quite likely to make use of simple-minded compressors.

```

IP header:
 60 00 00 00 00 5c 3a ff fe 80 00 00 00 00 00 00
 02 1c da ff fe 00 30 23 ff 02 00 00 00 00 00 00
 00 00 00 00 00 00 00 00 1a
Payload:
 9b 01 7a 5f 00 f0 01 00 88 00 00 00 20 02 0d b8
 00 00 00 00 00 00 00 00 ff fe 00 fa ce 04 0e 00 14
 09 ff 00 00 01 00 00 00 00 00 00 00 08 1e 80 20
 ff ff ff ff ff ff ff ff 00 00 00 00 20 02 0d b8
 00 00 00 00 00 00 00 00 ff fe 00 fa ce 03 0e 40 00
 ff ff ff ff 20 02 0d b8 00 00 00 00
Dictionary:
 fe 80 00 00 00 00 00 00 02 1c da ff fe 00 30 23
 ff 02 00 00 00 00 00 00 00 00 00 00 00 00 00 1a
 16 fe fd 17 fe fd 00 01 00 00 00 00 00 01 00 00
copy: 06 9b 01 7a 5f 00 f0
ref(9): 01 00 -> ref 11nnnkkk 0 7: c7
copy: 01 88
3 nulls: 81
copy: 04 20 02 0d b8
7 nulls: 85
ref(60): ff fe 00 -> ref 10lnssss 0 7/11nnnkkk 1 1: a7 c9
copy: 08 fa ce 04 0e 00 14 09 ff
ref(39): 00 00 01 00 00 -> ref 10lnssss 0 4/11nnnkkk 3 2: a4 da
5 nulls: 83
copy: 06 08 1e 80 20 ff ff
ref(2): ff ff -> ref 11nnnkkk 0 0: c0
ref(4): ff ff ff ff -> ref 11nnnkkk 2 0: d0
4 nulls: 82
ref(48): 20 02 0d b8 00 00 00 00 00 00 00 ff fe 00 fa ce
-> ref 10lnssss 1 4/11nnnkkk 6 0: b4 f0
copy: 03 03 0e 40
ref(9): 00 ff -> ref 11nnnkkk 0 7: c7
ref(28): ff ff ff -> ref 10lnssss 0 3/11nnnkkk 1 1: a3 c9
ref(24): 20 02 0d b8 00 00 00 00
-> ref 10lnssss 0 2/11nnnkkk 6 0: a2 f0
Compressed:
 06 9b 01 7a 5f 00 f0 c7 01 88 81 04 20 02 0d b8
 85 a7 c9 08 fa ce 04 0e 00 14 09 ff a4 da 83 06
 08 1e 80 20 ff ff c0 d0 82 b4 f0 03 03 0e 40 c7
 a3 c9 a2 f0
Was 92 bytes; compressed to 52 bytes, compression factor 1.77

```

Figure 9: A longer RPL example

Similarly, Figure 10 shows an RPL DAO message. One of the embedded addresses is copied right out of the pseudo-header, the other one is effectively converted from global to local by providing the prefix FE80 literally, inserting a number of nulls, and copying (some of) the IID part again out of the pseudo-header. Note that a simple implementation would probably emit fewer nulls and copy the entire IID; there are multiple ways to encode this 50-byte payload into 27 bytes.

IP header:

```
60 00 00 00 00 32 3a ff 20 02 0d b8 00 00 00 00
00 00 00 ff fe 00 33 44 20 02 0d b8 00 00 00 00
00 00 00 ff fe 00 11 22
```

Payload:

```
9b 02 58 7d 01 80 00 f1 05 12 00 80 20 02 0d b8
00 00 00 00 00 00 00 00 ff fe 00 33 44 06 14 00 80
f1 00 fe 80 00 00 00 00 00 00 00 00 00 ff fe 00
11 22
```

Dictionary:

```
20 02 0d b8 00 00 00 00 00 00 00 00 ff fe 00 33 44
20 02 0d b8 00 00 00 00 00 00 00 00 ff fe 00 11 22
16 fe fd 17 fe fd 00 01 00 00 00 00 00 01 00 00
```

copy: 0c 9b 02 58 7d 01 80 00 f1 05 12 00 80

ref(60): 20 02 0d b8 00 00 00 00 00 00 00 00 ff fe 00 33 44

-> ref 10lnssss 1 5/1lnnnkkk 6 4: b5 f4

copy: 08 06 14 00 80 f1 00 fe 80

9 nulls: 87

ref(66): ff fe 00 11 22 -> ref 10lnssss 0 7/1lnnnkkk 3 5: a7 dd

Compressed:

```
0c 9b 02 58 7d 01 80 00 f1 05 12 00 80 b5 f4 08
06 14 00 80 f1 00 fe 80 87 a7 dd
```

Was 50 bytes; compressed to 27 bytes, compression factor 1.85

Figure 10: An RPL DAO message

Figure 11 shows the effect of compressing a simple ND neighbor solicitation.

```

IP header:
 60 00 00 00 00 30 3a ff 20 02 0d b8 00 00 00 00
 00 00 00 ff fe 00 3b d3 fe 80 00 00 00 00 00 00
 02 1c da ff fe 00 30 23
Payload:
 87 00 a7 68 00 00 00 00 fe 80 00 00 00 00 00 00
 02 1c da ff fe 00 30 23 01 01 3b d3 00 00 00 00
 1f 02 00 00 00 00 00 06 00 1c da ff fe 00 20 24
Dictionary:
 20 02 0d b8 00 00 00 00 00 00 00 ff fe 00 3b d3
 fe 80 00 00 00 00 00 00 02 1c da ff fe 00 30 23
 16 fe fd 17 fe fd 00 01 00 00 00 00 00 01 00 00
copy: 04 87 00 a7 68
4 nulls: 82
ref(40): fe 80 00 00 00 00 00 00 02 1c da ff fe 00 30 23
-> ref 10lnssss 1 3/1lnnnkkk 6 0: b3 f0
copy: 04 01 01 3b d3
4 nulls: 82
copy: 02 1f 02
5 nulls: 83
copy: 02 06 00
ref(24): 1c da ff fe 00 -> ref 10lnssss 0 2/1lnnnkkk 3 3: a2 db
copy: 02 20 24
Compressed:
 04 87 00 a7 68 82 b3 f0 04 01 01 3b d3 82 02 1f
 02 83 02 06 00 a2 db 02 20 24
Was 48 bytes; compressed to 26 bytes, compression factor 1.85

```

Figure 11: An ND neighbor solicitation

Figure 12 shows the compression of an ND neighbor advertisement.

IP header:

```
60 00 00 00 00 30 3a fe fe 80 00 00 00 00 00 00
02 1c da ff fe 00 30 23 20 02 0d b8 00 00 00 00
00 00 00 ff fe 00 3b d3
```

Payload:

```
88 00 26 6c c0 00 00 00 fe 80 00 00 00 00 00 00
02 1c da ff fe 00 30 23 02 01 fa ce 00 00 00 00
1f 02 00 00 00 00 00 06 00 1c da ff fe 00 20 24
```

Dictionary:

```
fe 80 00 00 00 00 00 00 02 1c da ff fe 00 30 23
20 02 0d b8 00 00 00 00 00 00 ff fe 00 3b d3
16 fe fd 17 fe fd 00 01 00 00 00 00 00 01 00 00
```

copy: 05 88 00 26 6c c0

3 nulls: 81

ref(56): fe 80 00 00 00 00 00 00 02 1c da ff fe 00 30 23

-> ref 10lnssss 1 5/1lnnnkkk 6 0: b5 f0

copy: 04 02 01 fa ce

4 nulls: 82

copy: 02 1f 02

5 nulls: 83

copy: 02 06 00

ref(24): 1c da ff fe 00 -> ref 10lnssss 0 2/1lnnnkkk 3 3: a2 db

copy: 02 20 24

Compressed:

```
05 88 00 26 6c c0 81 b5 f0 04 02 01 fa ce 82 02
1f 02 83 02 06 00 a2 db 02 20 24
```

Was 48 bytes; compressed to 27 bytes, compression factor 1.78

Figure 12: An ND neighbor advertisement

Figure 13 shows the compression of an ND router solicitation. Note that the relatively good compression is not caused by the many zero bytes in the link-layer address of this particular capture (which are unlikely to occur in practice): 7 of these 8 bytes are copied from the pseudo-header (the 8th byte cannot be copied as the universal/local bit needs to be inverted).

IP header:

```
60 00 00 00 00 18 3a ff fe 80 00 00 00 00 00 00
ae de 48 00 00 00 00 01 ff 02 00 00 00 00 00 00
00 00 00 00 00 00 00 02
```

Payload:

```
85 00 90 65 00 00 00 00 01 02 ac de 48 00 00 00
00 01 00 00 00 00 00 00
```

Dictionary:

```
fe 80 00 00 00 00 00 00 ae de 48 00 00 00 00 01
ff 02 00 00 00 00 00 00 00 00 00 00 00 00 02
16 fe fd 17 fe fd 00 01 00 00 00 00 00 01 00 00
```

copy: 04 85 00 90 65

ref(11): 00 00 00 00 01 -> ref 11nnnkkk 3 6:de

copy: 02 02 ac

ref(50): de 48 00 00 00 00 01

-> ref 10lnssss 0 5/11nnnkkk 5 3: a5 eb

6 nulls: 84

Compressed:

```
04 85 00 90 65 de 02 02 ac a5 eb 84
```

Was 24 bytes; compressed to 12 bytes, compression factor 2.00

Figure 13: An ND router solicitation

Figure 14 shows the compression of an ND router advertisement. The indefinite lifetime is compressed to four bytes by backreferencing; this could be improved (at the cost of minor additional decompressor complexity) by including some simple runlength mechanism.

```

IP header:
 60 00 00 00 00 60 3a ff fe 80 00 00 00 00 00 00
 10 34 00 ff fe 00 11 22 fe 80 00 00 00 00 00 00
 ae de 48 00 00 00 00 01
Payload:
 86 00 55 c9 40 00 0f a0 1c 5a 38 17 00 00 07 d0
 01 01 11 22 00 00 00 00 03 04 40 40 ff ff ff ff
 ff ff ff ff 00 00 00 00 20 02 0d b8 00 00 00 00
 00 00 00 00 00 00 00 00 20 02 40 10 00 00 03 e8
 20 02 0d b8 00 00 00 00 21 03 00 01 00 00 00 00
 20 02 0d b8 00 00 00 00 00 00 00 00 ff fe 00 11 22
Dictionary:
 fe 80 00 00 00 00 00 00 10 34 00 ff fe 00 11 22
 fe 80 00 00 00 00 00 00 ae de 48 00 00 00 00 01
 16 fe fd 17 fe fd 00 01 00 00 00 00 00 01 00 00
copy: 0c 86 00 55 c9 40 00 0f a0 1c 5a 38 17
2 nulls: 80
copy: 06 07 d0 01 01 11 22
4 nulls: 82
copy: 06 03 04 40 40 ff ff
ref(2): ff ff -> ref 11nnnkkk 0 0: c0
ref(4): ff ff ff ff -> ref 11nnnkkk 2 0: d0
4 nulls: 82
copy: 04 20 02 0d b8
12 nulls: 8a
copy: 04 20 02 40 10
ref(38): 00 00 03 -> ref 10lnssss 0 4/11nnnkkk 1 3: a4 cb
copy: 01 e8
ref(24): 20 02 0d b8 00 00 00 00
-> ref 10lnssss 0 2/11nnnkkk 6 0: a2 f0
copy: 02 21 03
ref(84): 00 01 00 00 00 00
-> ref 10lnssss 0 9/11nnnkkk 4 6: a9 e6
ref(40): 20 02 0d b8 00 00 00 00 00 00 00
-> ref 10lnssss 1 3/11nnnkkk 1 5: b3 cd
ref(128): ff fe 00 11 22
-> ref 10lnssss 0 15/11nnnkkk 3 3: af db
Compressed:
0c 86 00 55 c9 40 00 0f a0 1c 5a 38 17 80 06 07
d0 01 01 11 22 82 06 03 04 40 40 ff ff c0 d0 82
04 20 02 0d b8 8a 04 20 02 40 10 a4 cb 01 e8 a2
f0 02 21 03 a9 e6 b3 cd af db
Was 96 bytes; compressed to 58 bytes, compression factor 1.66

```

Figure 14: An ND router advertisement

Figure 15 shows the compression of a DTLS application data packet with a net payload of 13 bytes of cleartext, and 8 bytes of

authenticator (note that the IP header is not relevant for this example and has been set to 0). This makes good use of the static dictionary, and is quite effective crunching out the redundancy in the TLS_PSK_WITH_AES_128_CCM_8 header, leading to a net reduction by 15 bytes.

IP header:

```
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00
```

Payload:

```
17 fe fd 00 01 00 00 00 00 01 00 1d 00 01 00
00 00 00 00 01 09 b2 0e 82 c1 6e b6 96 c5 1f 36
8d 17 61 e2 b5 d4 22 d4 ed 2b
```

Dictionary:

```
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
16 fe fd 17 fe fd 00 01 00 00 00 00 01 00 00
```

ref(13): 17 fe fd 00 01 00 00 00 00 01 00

-> ref 10lnssss 1 0/1lnnnkkk 2 1: b0 d1

copy: 01 1d

ref(10): 00 01 00 00 00 00 00 01 -> ref 1lnnnkkk 6 2: f2

copy: 15 09 b2 0e 82 c1 6e b6 96 c5 1f 36 8d 17 61 e2

copy: b5 d4 22 d4 ed 2b

Compressed:

```
b0 d1 01 1d f2 15 09 b2 0e 82 c1 6e b6 96 c5 1f
36 8d 17 61 e2 b5 d4 22 d4 ed 2b
```

Was 42 bytes; compressed to 27 bytes, compression factor 1.56

Figure 15: A DTLS application data packet

Figure 16 shows that the compression is slightly worse in a subsequent packet (containing 6 bytes of cleartext and 8 bytes of authenticator, yielding a net compression of 13 bytes). The total overhead does stay at a quite acceptable 8 bytes.

IP header:

```
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00
```

Payload:

```
17 fe fd 00 01 00 00 00 00 00 05 00 16 00 01 00
00 00 00 00 05 ae a0 15 56 67 92 4d ff 8a 24 e4
cb 35 b9
```

Dictionary:

```
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
16 fe fd 17 fe fd 00 01 00 00 00 00 00 01 00 00
```

ref(13): 17 fe fd 00 01 00 00 00 00

-> ref 10lnssss 1 0/1lnnnkkk 0 3: b0 c3

copy: 03 05 00 16

ref(10): 00 01 00 00 00 00 00 05 -> ref 1lnnnkkk 6 2: f2

copy: 0e ae a0 15 56 67 92 4d ff 8a 24 e4 cb 35 b9

Compressed:

```
b0 c3 03 05 00 16 f2 0e ae a0 15 56 67 92 4d ff
8a 24 e4 cb 35 b9
```

Was 35 bytes; compressed to 22 bytes, compression factor 1.59

Figure 16: Another DTLS application data packet

Figure 17 shows the compression of a DTLS handshake message, here a client hello. There is little that can be compressed about the 32 bytes of randomness. Still, the net reduction is by 14 bytes.

IP header:

```
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00
```

Payload:

```
16 fe fd 00 00 00 00 00 00 00 00 00 00 36 01 00 00
2a 00 00 00 00 00 00 00 2a fe fd 51 52 ed 79 a4
20 c9 62 56 11 47 c9 39 ee 6c c0 a4 fe c6 89 2f
32 26 9a 16 4e 31 7e 9f 20 92 92 00 00 00 02 c0
a8 01 00
```

Dictionary:

```
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
16 fe fd 17 fe fd 00 01 00 00 00 00 00 01 00 00
```

ref(16): 16 fe fd -> ref 10lnssss 0 1/1lnnnkkk 1 5: a1 cd

9 nulls: 87

copy: 01 36

ref(16): 01 00 00 -> ref 10lnssss 0 1/1lnnnkkk 1 5: a1 cd

copy: 01 2a

7 nulls: 85

copy: 23 2a fe fd 51 52 ed 79 a4 20 c9 62 56 11 47 c9

copy: 39 ee 6c c0 a4 fe c6 89 2f 32 26 9a 16 4e 31 7e

copy: 9f 20 92 92

3 nulls: 81

copy: 05 02 c0 a8 01 00

Compressed:

```
a1 cd 87 01 36 a1 cd 01 2a 85 23 2a fe fd 51 52
ed 79 a4 20 c9 62 56 11 47 c9 39 ee 6c c0 a4 fe
c6 89 2f 32 26 9a 16 4e 31 7e 9f 20 92 92 81 05
02 c0 a8 01 00
```

Was 67 bytes; compressed to 53 bytes, compression factor 1.26

Figure 17: A DTLS handshake packet (client hello)

Author's Address

Carsten Bormann
Universitaet Bremen TZI
Postfach 330440
D-28359 Bremen
Germany

Phone: +49-421-218-63921
Email: cabo@tzi.org

Internet Engineering Task Force
Internet-Draft
Intended status: Standards Track
Expires: March 9, 2015

J. Schoenwaelder
A. Sehgal
Jacobs University
T. Tsou
Huawei Technologies (USA)
C. Zhou
Huawei Technologies
September 5, 2014

Definition of Managed Objects for IPv6 over Low-Power Wireless Personal
Area Networks (6LoWPANs)
draft-ietf-6lo-lowpan-mib-04

Abstract

This document defines a portion of the Management Information Base (MIB) for use with network management protocols in the Internet community. In particular, it defines objects for managing IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs).

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 March 9, 2015.

Copyright Notice

Copyright (c) 2014 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect

to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	2
2. The Internet-Standard Management Framework	2
3. Conventions	3
4. Overview	3
5. Relationship to Other MIB Modules	7
6. Definitions	7
7. Security Considerations	24
8. IANA Considerations	25
9. Acknowledgements	25
10. References	25
10.1. Normative References	25
10.2. Informative References	26

1. Introduction

This document defines a portion of the Management Information Base (MIB) for use with network management protocols. In particular it defines objects for managing IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs) [RFC4944].

While a MIB module provides a direct binding for accessing data via the Simple Network Management Protocol (SNMP) [RFC3410], supporting SNMP may not always be affordable on constrained devices. Other protocols to access data modeled in MIB modules are possible and proposals have been made recently to provide bindings to the Constrained Application Protocol (CoAP) [RFC7252].

2. The Internet-Standard Management Framework

For a detailed overview of the documents that describe the current Internet-Standard Management Framework, please refer to section 7 of RFC 3410 [RFC3410].

Managed objects are accessed via a virtual information store, termed the Management Information Base or MIB. MIB objects are generally accessed through the Simple Network Management Protocol (SNMP). Objects in the MIB are defined using the mechanisms defined in the Structure of Management Information (SMI). This document specifies a MIB module that is compliant to the SMIV2, which is described in STD 58, RFC 2578 [RFC2578], STD 58, RFC 2579 [RFC2579] and STD 58, RFC 2580 [RFC2580].

3. Conventions

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 RFC 2119 [RFC2119].

4. Overview

The left part of Figure 1 provides an overview of the IETF protocols designed for constrained devices. The right part lists the MIB modules providing monitoring and troubleshooting support ([RFC4113], [RFC4292], [RFC4293], [RFC2863]). The LOWPAN-MIB defined in this document fills a hole by providing monitoring and troubleshooting support for the 6LoWPAN layer.

Protocol Layer	MIB Modules
+-----+	+-----+
CoAP [RFC7252]	UDP-MIB [RFC4113]
+-----+	+-----+
UDP [RFC0768]	IP-MIB [RFC4293]
+-----+	+-----+
IPv6 [RFC2460]	IP-FORWARD-MIB [RFC4292]
ICMPv6 [RFC4443]	+-----+
+-----+	LOWPAN-MIB [RFCXXXX]
6LoWPAN [RFC4944]	+-----+
+-----+	IF-MIB [RFC2863]
IEEE 802.15.4, ...	+-----+
+-----+	

/* RFC Ed.: replace XXXX above with RFC number
and remove this note */

Figure 1: Protocol Layers and MIB Modules

The LOWPAN-MIB module is primarily a collection of counters that reflect how 6LoWPAN datagrams are processed by the 6LoWPAN layer. The objects are defined twice, once to report the global statistics as seen by the 6LoWPAN layer and once to report per interface 6LoWPAN layer statistics. The per interface statistics are optional to implement. The object identifier registration tree has the following structure:

/* RFC Ed.: replace XXXX below with IANA assigned OID number
and remove this note */

```

---- lowpanMIB(1.3.6.1.2.1.XXXX)
+---- lowpanNotifications(0)
+---- lowpanObjects(1)
|   +---- lowpanStats(1)
|   |   +--r- lowpanReasmTimeout(1)           Unsigned32
|   |   +--r- lowpanInReceives(2)           Counter32
|   |   +--r- lowpanInHdrErrors(3)          Counter32
|   |   +--r- lowpanInMeshReceives(4)       Counter32
|   |   +--r- lowpanInMeshForwds(5)         Counter32
|   |   +--r- lowpanInMeshDelivers(6)       Counter32
|   |   +--r- lowpanInReasmReqds(7)         Counter32
|   |   +--r- lowpanInReasmFails(8)         Counter32
|   |   +--r- lowpanInReasmOKs(9)          Counter32
|   |   +--r- lowpanInCompReqds(10)         Counter32
|   |   +--r- lowpanInCompFails(11)         Counter32
|   |   +--r- lowpanInCompOKs(12)          Counter32
|   |   +--r- lowpanInDiscards(13)         Counter32
|   |   +--r- lowpanInDelivers(14)         Counter32
|   |   +--r- lowpanOutRequests(15)         Counter32
|   |   +--r- lowpanOutCompReqds(16)        Counter32
|   |   +--r- lowpanOutCompFails(17)        Counter32
|   |   +--r- lowpanOutCompOKs(18)         Counter32
|   |   +--r- lowpanOutFragReqds(19)        Counter32
|   |   +--r- lowpanOutFragFails(20)        Counter32
|   |   +--r- lowpanOutFragOKs(21)         Counter32
|   |   +--r- lowpanOutFragCreates(22)      Counter32
|   |   +--r- lowpanOutMeshHopLimitExceeds(23) Counter32
|   |   +--r- lowpanOutMeshNoRoutes(24)     Counter32
|   |   +--r- lowpanOutMeshRequests(25)     Counter32
|   |   +--r- lowpanOutMeshForwds(26)      Counter32
|   |   +--r- lowpanOutMeshTransmits(27)    Counter32
|   |   +--r- lowpanOutDiscards(28)        Counter32
|   |   +--r- lowpanOutTransmits(29)        Counter32
|   +---- lowpanIfStatsTable(2)
|   |   +---- lowpanIfStatsEntry(1) [ifIndex]
|   |   |   +--r- lowpanIfReasmTimeout(1)   Unsigned32
|   |   |   +--r- lowpanIfInReceives(2)     Counter32
|   |   |   +--r- lowpanIfInHdrErrors(3)     Counter32
|   |   |   +--r- lowpanIfInMeshReceives(4)  Counter32
|   |   |   +--r- lowpanIfInMeshForwds(5)    Counter32
|   |   |   +--r- lowpanIfInMeshDelivers(6)  Counter32
|   |   |   +--r- lowpanIfInReasmReqds(7)    Counter32
|   |   |   +--r- lowpanIfInReasmFails(8)    Counter32
|   |   |   +--r- lowpanIfInReasmOKs(9)      Counter32
|   |   |   +--r- lowpanIfInCompReqds(10)    Counter32
|   |   |   +--r- lowpanIfInCompFails(11)    Counter32
|   |   |   +--r- lowpanIfInCompOKs(12)      Counter32
|   |   |   +--r- lowpanIfInDiscards(13)     Counter32

```



```

|      +--r- lowpanIfInDelivers(14)          Counter32
|      +--r- lowpanIfOutRequests(15)         Counter32
|      +--r- lowpanIfOutCompReqds(16)        Counter32
|      +--r- lowpanIfOutCompFails(17)        Counter32
|      +--r- lowpanIfOutCompOKs(18)          Counter32
|      +--r- lowpanIfOutFragReqds(19)        Counter32
|      +--r- lowpanIfOutFragFails(20)        Counter32
|      +--r- lowpanIfOutFragOKs(21)          Counter32
|      +--r- lowpanIfOutFragCreates(22)      Counter32
|      +--r- lowpanIfOutMeshHopLimitExceeds(23) Counter32
|      +--r- lowpanIfOutMeshNoRoutes(24)     Counter32
|      +--r- lowpanIfOutMeshRequests(25)     Counter32
|      +--r- lowpanIfOutMeshForwds(26)       Counter32
|      +--r- lowpanIfOutMeshTransmits(27)    Counter32
|      +--r- lowpanIfOutDiscards(28)         Counter32
|      +--r- lowpanIfOutTransmits(29)        Counter32
+----- lowpanConformance(2)
+----- lowpanGroups(1)
|   +----- lowpanStatsGroup(1)
|   +----- lowpanStatsMeshGroup(2)
|   +----- lowpanIfStatsGroup(3)
|   +----- lowpanIfStatsMeshGroup(4)
+----- lowpanCompliances(2)
|   +----- lowpanCompliance(1)

```

The counters defined in the LOWPAN-MIB module provide information about the 6LoWPAN datagrams received and transmitted and how they are processed in the 6LoWPAN layer. For link-layers that use the 6LoWPAN dispatch byte as defined in [RFC4944] (e.g., IEEE 802.15.4), a 6LoWPAN datagram is a datagram with a dispatch byte matching the bit patterns 01xxxxxx, 10xxxxxx, or 11xxxxxx. Datagrams with a dispatch byte matching the bit pattern 00xxxxxx (NALP - not a LoWPAN frame) are not considered to be 6LoWPAN datagram by this specification. Other radio technologies may use different mechanisms to identify 6LoWPAN datagrams (e.g., the BLUETOOTH Low Energy Logical Link Control and Adaptation Protocol uses Channel Identifiers [I-D.ietf-6lo-btle]).

The Case Diagram [CASE] in Figure 2 illustrates the conceptual relationships between the counters. Implementations may choose to implement the processing of 6LoWPAN datagrams in a different order.

The generic InDiscards and OutDiscards counters can be incremented anytime when 6LoWPAN datagrams are discarded due to reasons not covered by the other more specific counters. For example, an implementation discarding 6LoWPAN datagrams while all buffers are used for ongoing packet reassemblies will increment the relevant InDiscards counters for each discarded 6LoWPAN datagram.

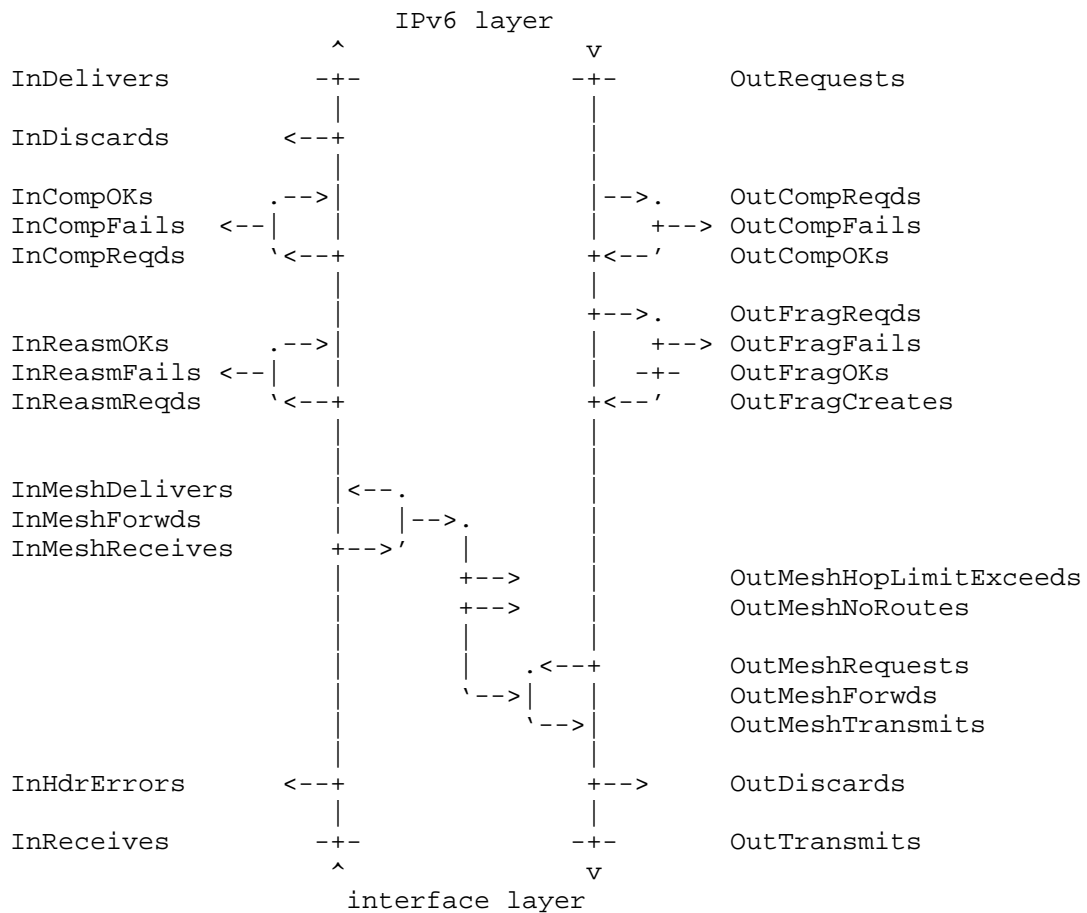


Figure 2: Conceptual Relationship between LOWPAN-MIB Counters

The fragmentation related counters have been modeled after the fragmentation related counters of the IP-MIB [RFC4293]. The discard counters have been placed at the end of the input and output chains but they can be bumped any time if a datagram is discarded for a reason not covered by the other counters.

The compression related counters provide insights into compression requests and in particular also compression related failures. Note that the diagram is conceptual in the sense that compression happens after reassembly for incoming 6LoWPAN datagrams and compression happens before fragmentation for outgoing 6LoWPAN datagrams. Implementations may choose to implement things slightly differently. For example, implementations may decompress FRAG1 fragments as soon as they are received, not waiting for reassembly to complete.

The mesh header processing related counters do not have an explicit discard counter. Implementations that do not support mesh forwarding MUST count the number of received 6LoWPAN datagrams with a MESH header (lowpanInMeshReceives) but they MUST NOT increment the lowpanInMeshReceives and lowpanInMeshDelivers counters if these 6LoWPAN datagrams are dropped.

5. Relationship to Other MIB Modules

The MIB module imports definitions from SNMPv2-SMI [RFC2578], SNMPv2-CONF [RFC2580], and IF-MIB [RFC2863].

6. Definitions

```
LOWPAN-MIB DEFINITIONS ::= BEGIN
```

```
IMPORTS
```

```
    MODULE-IDENTITY, OBJECT-TYPE, Unsigned32, Counter32, mib-2
        FROM SNMPv2-SMI                                -- RFC 2578
    OBJECT-GROUP, MODULE-COMPLIANCE
        FROM SNMPv2-CONF                                -- RFC 2580
    ifIndex FROM IF-MIB;                                -- RFC 2863
```

```
lowpanMIB      MODULE-IDENTITY
```

```
    LAST-UPDATED      "201409050000Z"
```

```
    ORGANIZATION
```

```
        "IETF IPv6 over Networks of Resource-constrained Nodes
         Working Group"
```

```
    CONTACT-INFO
```

```
        "WG Email: 6lo@ietf.org
```

```
        WG Web:   http://tools.ietf.org/wg/6lo/
```

```
        Juergen Schoenwaelder
```

```
        Jacobs University Bremen
```

```
        Email: j.schoenwaelder@jacobs-university.de
```

```
        Anuj Sehgal
```

```
        Jacobs University Bremen
```

```
        Email: s.anuj@jacobs-university.de
```

```
        Tina Tsou
```

```
        Huawei Technologies
```

```
        Email: tina.tsou.zouting@huawei.com
```

```
        Cathy Zhou
```

```
        Huawei Technologies
```

```
        Email: cathyzhou@huawei.com"
```

DESCRIPTION

"The MIB module for monitoring nodes implementing the IPv6 over Low-Power Wireless Personal Area Networks (6LoWPAN) protocol.

Copyright (c) 2014 IETF Trust and the persons identified as authors of the code. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, is permitted pursuant to, and subject to the license terms contained in, the Simplified BSD License set forth in Section 4.c of the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>)."

REVISION "201409050000Z"

DESCRIPTION

"Initial version, published as RFC XXXX."

-- RFC Ed.: replace XXXX with RFC number and remove this note

::= { mib-2 YYYY }

-- RFC Ed.: replace YYYY with IANA assigned number

-- object definitions

lowpanNotifications OBJECT IDENTIFIER ::= { lowpanMIB 0 }
lowpanObjects OBJECT IDENTIFIER ::= { lowpanMIB 1 }
lowpanConformance OBJECT IDENTIFIER ::= { lowpanMIB 2 }

lowpanStats OBJECT IDENTIFIER ::= { lowpanObjects 1 }

lowpanReasmTimeout OBJECT-TYPE

SYNTAX Unsigned32

UNITS "seconds"

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The maximum number of seconds that received fragments are held while they are awaiting reassembly at this entity."

::= { lowpanStats 1 }

lowpanInReceives OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The total number of 6LoWPAN datagrams received, including those received in error."

```
 ::= { lowpanStats 2 }

lowpanInHdrErrors OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION
        "The number of received 6LoWPAN datagrams discarded due to
         errors in their headers, including unknown dispatch values."
    ::= { lowpanStats 3 }

lowpanInMeshReceives OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION
        "The number of received 6LoWPAN datagrams with a MESH
         header."
    ::= { lowpanStats 4 }

lowpanInMeshForwds OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION
        "The number of received 6LoWPAN datagrams requiring MESH
         forwarding."
    ::= { lowpanStats 5 }

lowpanInMeshDelivers OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION
        "The number of received 6LoWPAN datagrams with a MESH header
         delivered to the local system."
    ::= { lowpanStats 6 }

lowpanInReasmReqds OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION
        "The number of received 6LoWPAN fragments that needed to
         be reassembled. This includes both FRAG1 and FRAGN 6LoWPAN
         datagrams."
    ::= { lowpanStats 7 }
```

lowpanInReasmFails OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The number of failures detected by the re-assembly algorithm (e.g., timeouts). Note that this is not necessarily a count of discarded 6LoWPAN fragments since implementations can lose track of the number of fragments by combining them as received."

::= { lowpanStats 8 }

lowpanInReasmOKs OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The number of IPv6 packets successfully reassembled."

::= { lowpanStats 9 }

lowpanInCompReqds OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The number of 6LoWPAN datagrams requiring header decompression."

::= { lowpanStats 10 }

lowpanInCompFails OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The number of 6LoWPAN datagrams where header decompression failed (e.g., because the necessary context information was not available)."

::= { lowpanStats 11 }

lowpanInCompOKs OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The number of 6LoWPAN datagrams where header decompression was successful."

::= { lowpanStats 12 }

lowpanInDiscards OBJECT-TYPE
SYNTAX Counter32
MAX-ACCESS read-only
STATUS current
DESCRIPTION
 "The number of received 6LoWPAN datagrams for which no
 problems were encountered to prevent their continued
 processing, but were discarded (e.g., for lack of buffer
 space). Note that this counter does not include any
 datagrams discarded due to a reassembly failure or a
 compression failure."
 ::= { lowpanStats 13 }

lowpanInDelivers OBJECT-TYPE
SYNTAX Counter32
MAX-ACCESS read-only
STATUS current
DESCRIPTION
 "The total number of IPv6 packets successfully delivered
 to the IPv6 layer."
 ::= { lowpanStats 14 }

lowpanOutRequests OBJECT-TYPE
SYNTAX Counter32
MAX-ACCESS read-only
STATUS current
DESCRIPTION
 "The total number of IPv6 packets supplied by the IPv6
 layer."
 ::= { lowpanStats 15 }

lowpanOutCompReqds OBJECT-TYPE
SYNTAX Counter32
MAX-ACCESS read-only
STATUS current
DESCRIPTION
 "The total number of IPv6 packets for which header
 compression was attempted."
 ::= { lowpanStats 16 }

lowpanOutCompFails OBJECT-TYPE
SYNTAX Counter32
MAX-ACCESS read-only
STATUS current
DESCRIPTION
 "The total number of IPv6 packets for which header
 compression failed."
 ::= { lowpanStats 17 }

```
lowpanOutCompOKs OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION
        "The total number of IPv6 packets for which header
         compression was successful."
    ::= { lowpanStats 18 }

lowpanOutFragReqds OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION
        "The number of IPv6 packets that required fragmentation
         in order to be transmitted."
    ::= { lowpanStats 19 }

lowpanOutFragFails OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION
        "The number of IPv6 packets that have been discarded because
         fragmentation failed."
    ::= { lowpanStats 20 }

lowpanOutFragOKs OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION
        "The number of IPv6 packets that have been successfully
         fragmented."
    ::= { lowpanStats 21 }

lowpanOutFragCreates OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION
        "The number of 6LoWPAN fragments that have been
         generated as a result of fragmentation. This includes
         both FRAG1 and FRAGN 6LoWPAN datagrams."
    ::= { lowpanStats 22 }

lowpanOutMeshHopLimitExceeds OBJECT-TYPE
    SYNTAX      Counter32
```



```
MAX-ACCESS    read-only
STATUS        current
DESCRIPTION
    "The number of 6LoWPAN datagrams with a MESH header that
    were dropped because the hop limit has been exceeded."
 ::= { lowpanStats 23 }

lowpanOutMeshNoRoutes OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The number of 6LoWPAN datagrams with a MESH header that
        were dropped because there was no forwarding information
        available."
    ::= { lowpanStats 24 }

lowpanOutMeshRequests OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The number of 6LoWPAN datagrams requiring MESH header
        encapsulation."
    ::= { lowpanStats 25 }

lowpanOutMeshForwds OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The number of 6LoWPAN datagrams with a MESH header for
        which suitable forwarding information was available."
    ::= { lowpanStats 26 }

lowpanOutMeshTransmits OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The number of 6LoWPAN datagrams with a MESH header
        created."
    ::= { lowpanStats 27 }

lowpanOutDiscards OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS  read-only
    STATUS      current
```

DESCRIPTION

"The number of IPv6 packets for which no problem was encountered to prevent their transmission to their destination, but were discarded (e.g., for lack of buffer space)."

::= { lowpanStats 28 }

lowpanOutTransmits OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The total number of 6LoWPAN datagram that this entity supplied to the lower layers for transmission."

::= { lowpanStats 29 }

lowpanIfStatsTable OBJECT-TYPE

SYNTAX SEQUENCE OF LowpanIfStatsEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"A table providing per interface statistics."

::= { lowpanObjects 2 }

lowpanIfStatsEntry OBJECT-TYPE

SYNTAX LowpanIfStatsEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"An entry providing statistics for a specific interface."

INDEX { ifIndex }

::= { lowpanIfStatsTable 1 }

LowpanIfStatsEntry ::= SEQUENCE {

lowpanIfReasmTimeout	Unsigned32,
lowpanIfInReceives	Counter32,
lowpanIfInHdrErrors	Counter32,
lowpanIfInMeshReceives	Counter32,
lowpanIfInMeshForwds	Counter32,
lowpanIfInMeshDelivers	Counter32,
lowpanIfInReasmReqds	Counter32,
lowpanIfInReasmFails	Counter32,
lowpanIfInReasmOKs	Counter32,
lowpanIfInCompReqds	Counter32,
lowpanIfInCompFails	Counter32,
lowpanIfInCompOKs	Counter32,
lowpanIfInDiscards	Counter32,
lowpanIfInDelivers	Counter32,

```
    lowpanIfOutRequests          Counter32,
    lowpanIfOutCompReqds         Counter32,
    lowpanIfOutCompFails         Counter32,
    lowpanIfOutCompOKs           Counter32,
    lowpanIfOutFragReqds         Counter32,
    lowpanIfOutFragFails         Counter32,
    lowpanIfOutFragOKs           Counter32,
    lowpanIfOutFragCreates        Counter32,
    lowpanIfOutMeshHopLimitExceeds Counter32,
    lowpanIfOutMeshNoRoutes       Counter32,
    lowpanIfOutMeshRequests       Counter32,
    lowpanIfOutMeshForwds         Counter32,
    lowpanIfOutMeshTransmits      Counter32,
    lowpanIfOutDiscards           Counter32,
    lowpanIfOutTransmits          Counter32
}

lowpanIfReasmTimeout OBJECT-TYPE
    SYNTAX      Unsigned32
    UNITS        "seconds"
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION
        "The maximum number of seconds that received fragments are
         held while they are awaiting reassembly at this interface."
    ::= { lowpanIfStatsEntry 1 }

lowpanIfInReceives OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION
        "The total number of 6LoWPAN datagrams received on this
         interface, including those received in error."
    ::= { lowpanIfStatsEntry 2 }

lowpanIfInHdrErrors OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION
        "The number of 6LoWPAN datagrams received on this
         interface that were discarded due to errors in
         their headers, including unknown dispatch values."
    ::= { lowpanIfStatsEntry 3 }

lowpanIfInMeshReceives OBJECT-TYPE
    SYNTAX      Counter32
```

```
MAX-ACCESS    read-only
STATUS        current
DESCRIPTION
    "The number of 6LoWPAN datagrams reveived on this
    interface with a MESH header."
 ::= { lowpanIfStatsEntry 4 }

lowpanIfInMeshForwds OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The number of 6LoWPAN datagrams received on this
        interface requiring MESH forwarding."
    ::= { lowpanIfStatsEntry 5 }

lowpanIfInMeshDelivers OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The number of 6LoWPAN datagrams received on this
        interface with a MESH header delivered to the local
        system."
    ::= { lowpanIfStatsEntry 6 }

lowpanIfInReasmReqds OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The number of 6LoWPAN fragments received on this
        interface that needed to be reassembled. This
        includes both FRAG1 and FRAGN 6LoWPAN datagrams."
    ::= { lowpanIfStatsEntry 7 }

lowpanIfInReasmFails OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The number of failures detected by the re-assembly
        algorithm (e.g., timeouts) for datagrams received
        on this interface. Note that this is not necessarily
        a count of discarded 6LoWPAN fragments since
        implementations can lose track of the number
        of fragments by combining them as received."
    ::= { lowpanIfStatsEntry 8 }
```

lowpanIfInReasmOKs OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The number of IPv6 packets successfully reassembled from fragments received on this interface."

::= { lowpanIfStatsEntry 9 }

lowpanIfInCompReqds OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The number of 6LoWPAN datagrams received on this interface requiring header decompression."

::= { lowpanIfStatsEntry 10 }

lowpanIfInCompFails OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The number of 6LoWPAN datagrams received on this interface where header decompression failed (e.g., because the necessary context information was not available)."

::= { lowpanIfStatsEntry 11 }

lowpanIfInCompOKs OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The number of 6LoWPAN datagrams received on this interface where header decompression was successful."

::= { lowpanIfStatsEntry 12 }

lowpanIfInDiscards OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The number of 6LoWPAN datagrams received on this interface for which no problems were encountered to prevent their continued processing, but were discarded (e.g., for lack of buffer space). Note that this counter does not include any datagrams discarded due

to a reassembly failure or a compression failure."
 ::= { lowpanIfStatsEntry 13 }

lowpanIfInDelivers OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The total number of IPv6 packets received on this interface that were successfully delivered to the IPv6 layer."

::= { lowpanIfStatsEntry 14 }

lowpanIfOutRequests OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The total number of IPv6 packets supplied by the IPv6 layer to be sent over this interface."

::= { lowpanIfStatsEntry 15 }

lowpanIfOutCompReqds OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The total number of IPv6 packets to be sent over this interface for which header compression was attempted."

::= { lowpanIfStatsEntry 16 }

lowpanIfOutCompFails OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The total number of IPv6 packets to be sent over this interface for which header compression failed."

::= { lowpanIfStatsEntry 17 }

lowpanIfOutCompOKs OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The total number of IPv6 packets to be sent over this interface for which header compression was

```
        successful."
 ::= { lowpanIfStatsEntry 18 }

lowpanIfOutFragReqds OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION
        "The number of IPv6 packets to be sent over this
        interface that required fragmentation in order
        to be transmitted."
 ::= { lowpanIfStatsEntry 19 }

lowpanIfOutFragFails OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION
        "The number of IPv6 packets to be sent over this
        interface that have been discarded because
        fragmentation failed."
 ::= { lowpanIfStatsEntry 20 }

lowpanIfOutFragOKs OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION
        "The number of IPv6 packets to be sent over this
        interface that have been successfully fragmented."
 ::= { lowpanIfStatsEntry 21 }

lowpanIfOutFragCreates OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION
        "The number of 6LoWPAN fragments that have been
        generated on this interface as a result of
        fragmentation. This includes both FRAG1 and FRAGN
        6LoWPAN datagrams."
 ::= { lowpanIfStatsEntry 22 }

lowpanIfOutMeshHopLimitExceeds OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION
```

```
        "The number of 6LoWPAN datagrams to be sent on this
        interface with a MESH header that were dropped
        because the hop limit has been exceeded."
 ::= { lowpanIfStatsEntry 23 }

lowpanIfOutMeshNoRoutes OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS   read-only
    STATUS      current
    DESCRIPTION
        "The number of 6LoWPAN datagrams to be sent on this
        interface with a MESH header that were dropped
        because there was no forwarding information available."
 ::= { lowpanIfStatsEntry 24 }

lowpanIfOutMeshRequests OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS   read-only
    STATUS      current
    DESCRIPTION
        "The number of 6LoWPAN datagrams to be sent on this
        interface requiring MESH header encapsulation."
 ::= { lowpanIfStatsEntry 25 }

lowpanIfOutMeshForwds OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS   read-only
    STATUS      current
    DESCRIPTION
        "The number of 6LoWPAN datagrams to be sent on this
        interface with a MESH header for which suitable
        forwarding information was available."
 ::= { lowpanIfStatsEntry 26 }

lowpanIfOutMeshTransmits OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS   read-only
    STATUS      current
    DESCRIPTION
        "The number of 6LoWPAN datagrams to be send on this
        interface with a MESH header created."
 ::= { lowpanIfStatsEntry 27 }

lowpanIfOutDiscards OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS   read-only
    STATUS      current
    DESCRIPTION
```



```
        "The number of IPv6 packets to be sent over this
        interface for which no problem was encountered to
        prevent their transmission to their destination, but
        were discarded (e.g., for lack of buffer space)."
```

```
 ::= { lowpanIfStatsEntry 28 }
```

```
lowpanIfOutTransmits OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The total number of 6LoWPAN datagrams to be sent on
        this interface that this entity supplied to the lower
        layers for transmission."
    ::= { lowpanIfStatsEntry 29 }
```

```
-- conformance definitions
```

```
lowpanGroups          OBJECT IDENTIFIER ::= { lowpanConformance 1 }
lowpanCompliances     OBJECT IDENTIFIER ::= { lowpanConformance 2 }
```

```
lowpanCompliance MODULE-COMPLIANCE
    STATUS      current
    DESCRIPTION
        "Compliance statement for systems that implement 6LoWPAN."
    MODULE      -- this module
    MANDATORY-GROUPS {
        lowpanStatsGroup
    }
    GROUP        lowpanStatsMeshGroup
    DESCRIPTION
        "This group is mandatory for implementations that process
        or forward 6LoWPAN datagrams with mesh headers."
    GROUP        lowpanIfStatsGroup
    DESCRIPTION
        "This group is mandatory for implementations that expose
        per interface statistics."
    GROUP        lowpanIfStatsMeshGroup
    DESCRIPTION
        "This group is mandatory for implementations that expose
        per interface statistics and that process or forward
        6LoWPAN datagrams with mesh headers."
    ::= { lowpanCompliances 1 }
```

```
lowpanStatsGroup OBJECT-GROUP
    OBJECTS {
        lowpanReasmTimeout,
        lowpanInReceives,
```

```
        lowpanInHdrErrors,
        lowpanInMeshReceives,
        lowpanInReasmReqds,
        lowpanInReasmFails,
        lowpanInReasmOKs,
        lowpanInCompReqds,
        lowpanInCompFails,
        lowpanInCompOKs,
        lowpanInDiscards,
        lowpanInDelivers,
        lowpanOutRequests,
        lowpanOutCompReqds,
        lowpanOutCompFails,
        lowpanOutCompOKs,
        lowpanOutFragReqds,
        lowpanOutFragFails,
        lowpanOutFragOKs,
        lowpanOutFragCreates,
        lowpanOutDiscards,
        lowpanOutTransmits
    }
    STATUS          current
    DESCRIPTION
        "A collection of objects providing information and
        statistics about the processing of 6LoWPAN datagrams,
        excluding counters covering the processing of datagrams
        with a mesh headers."
    ::= { lowpanGroups 1 }

lowpanStatsMeshGroup OBJECT-GROUP
    OBJECTS {
        lowpanInMeshForwds,
        lowpanInMeshDelivers,
        lowpanOutMeshHopLimitExceeds,
        lowpanOutMeshNoRoutes,
        lowpanOutMeshRequests,
        lowpanOutMeshForwds,
        lowpanOutMeshTransmits
    }
    STATUS          current
    DESCRIPTION
        "A collection of objects providing information and
        statistics about the processing of 6LoWPAN datagrams
        with a 6LoWPAN mesh header."
    ::= { lowpanGroups 2 }

lowpanIfStatsGroup OBJECT-GROUP
    OBJECTS {
```

```
        lowpanIfReasmTimeout,
        lowpanIfInReceives,
        lowpanIfInHdrErrors,
        lowpanIfInMeshReceives,
        lowpanIfInReasmReqds,
        lowpanIfInReasmFails,
        lowpanIfInReasmOKs,
        lowpanIfInCompReqds,
        lowpanIfInCompFails,
        lowpanIfInCompOKs,
        lowpanIfInDiscards,
        lowpanIfInDelivers,
        lowpanIfOutRequests,
        lowpanIfOutCompReqds,
        lowpanIfOutCompFails,
        lowpanIfOutCompOKs,
        lowpanIfOutFragReqds,
        lowpanIfOutFragFails,
        lowpanIfOutFragOKs,
        lowpanIfOutFragCreates,
        lowpanIfOutDiscards,
        lowpanIfOutTransmits
    }
    STATUS          current
    DESCRIPTION
        "A collection of objects providing per interface
        information and statistics about the processing
        of 6LoWPAN datagrams, excluding counters covering
        the processing of datagrams with a mesh headers."
    ::= { lowpanGroups 3 }

lowpanIfStatsMeshGroup OBJECT-GROUP
    OBJECTS {
        lowpanIfInMeshForwds,
        lowpanIfInMeshDelivers,
        lowpanIfOutMeshHopLimitExceeds,
        lowpanIfOutMeshNoRoutes,
        lowpanIfOutMeshRequests,
        lowpanIfOutMeshForwds,
        lowpanIfOutMeshTransmits
    }
    STATUS          current
    DESCRIPTION
        "A collection of objects providing per interface
        information and statistics about the processing
        of 6LoWPAN datagrams with a 6LoWPAN mesh header."
    ::= { lowpanGroups 4 }
```

END

7. Security Considerations

There are no management objects defined in this MIB module that have a MAX-ACCESS clause of read-write and/or read-create. So, if this MIB module is implemented correctly, then there is no risk that an intruder can alter or create any management objects of this MIB module via direct SNMP SET operations.

Some of the readable objects in this MIB module (i.e., objects with a MAX-ACCESS other than not-accessible) may be considered sensitive or vulnerable in some network environments. It is thus important to control even GET and/or NOTIFY access to these objects and possibly to even encrypt the values of these objects when sending them over the network via SNMP.

The read-only counters provide insights into the amount of 6LoWPAN traffic a node is receiving or transmitting. This might provide information whether a device is regularly exchanging information with other devices or whether a device is mostly not participating in any communication (e.g., the device might be "easier" to take away unnoticed). The reassembly counters could be used to direct denial of service attacks on the reassembly mechanism.

SNMP versions prior to SNMPv3 did not include adequate security. Even if the network itself is secure (for example by using IPsec), even then, there is no control as to who on the secure network is allowed to access and GET/SET (read/change/create/delete) the objects in this MIB module.

It is RECOMMENDED that implementers consider the security features as provided by the SNMPv3 framework (see [RFC3410], section 8), including full support for the SNMPv3 cryptographic mechanisms (for authentication and privacy).

Further, deployment of SNMP versions prior to SNMPv3 is NOT RECOMMENDED. Instead, it is RECOMMENDED to deploy SNMPv3 and to enable cryptographic security. It is then a customer/operator responsibility to ensure that the SNMP entity giving access to an instance of this MIB module is properly configured to give access to the objects only to those principals (users) that have legitimate rights to indeed GET or SET (change/create/delete) them.

8. IANA Considerations

IANA and RFC Ed.: IANA is requested to assign a value for "YYYY" under the 'mib-2' subtree and to record the assignment in the SMI Numbers registry. When the assignment has been made, the RFC Editor is asked to replace "YYYY" (here and in the MIB module) with the assigned value and to remove this note.

The MIB module in this document uses the following IANA-assigned OBJECT IDENTIFIER values recorded in the SMI Numbers registry:

Descriptor	OBJECT IDENTIFIER value
-----	-----
lowpanMIB	{ mib-2 YYYY }

9. Acknowledgements

This specification borrows heavily from the IP-MIB defined in [RFC4293].

Juergen Schoenwaelder and Anuj Sehgal were partly funded by Flamingo, a Network of Excellence project (ICT-318488) supported by the European Commission under its Seventh Framework Programme.

10. References

10.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC2578] McCloghrie, K., Ed., Perkins, D., Ed., and J. Schoenwaelder, Ed., "Structure of Management Information Version 2 (SMIv2)", STD 58, RFC 2578, April 1999.
- [RFC2579] McCloghrie, K., Ed., Perkins, D., Ed., and J. Schoenwaelder, Ed., "Textual Conventions for SMIv2", STD 58, RFC 2579, April 1999.
- [RFC2580] McCloghrie, K., Perkins, D., and J. Schoenwaelder, "Conformance Statements for SMIv2", STD 58, RFC 2580, April 1999.
- [RFC4944] Montenegro, G., Kushalnagar, N., Hui, J., and D. Culler, "Transmission of IPv6 Packets over IEEE 802.15.4 Networks", RFC 4944, September 2007.

- [RFC2863] McCloghrie, K. and F. Kastenholz, "The Interfaces Group MIB", RFC 2863, June 2000.

10.2. Informative References

- [RFC0768] Postel, J., "User Datagram Protocol", STD 6, RFC 768, August 1980.
- [RFC2460] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", RFC 2460, December 1998.
- [RFC3410] Case, J., Mundy, R., Partain, D., and B. Stewart, "Introduction and Applicability Statements for Internet-Standard Management Framework", RFC 3410, December 2002.
- [RFC4113] Fenner, B. and J. Flick, "Management Information Base for the User Datagram Protocol (UDP)", RFC 4113, June 2005.
- [RFC4292] Haberman, B., "IP Forwarding Table MIB", RFC 4292, April 2006.
- [RFC4293] Routhier, S., "Management Information Base for the Internet Protocol (IP)", RFC 4293, April 2006.
- [RFC4443] Conta, A., Deering, S., and M. Gupta, "Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification", RFC 4443, March 2006.
- [RFC7252] Shelby, Z., Hartke, K., and C. Bormann, "The Constrained Application Protocol (CoAP)", RFC 7252, June 2014.
- [I-D.ietf-6lo-btle]
Nieminen, J., Savolainen, T., Isomaki, M., Patil, B., Shelby, Z., and C. Gomez, "Transmission of IPv6 Packets over BLUETOOTH(R) Low Energy", draft-ietf-6lo-btle-03 (work in progress), September 2014.
- [CASE] Case, J. and C. Partridge, "Case Diagrams: A First Step to Diagrammed Management Information Bases", Computer Communications Review 19(1), January 1989.

Authors' Addresses

Juergen Schoenwaelder
Jacobs University
Campus Ring 1
Bremen 28759
Germany

EMail: j.schoenwaelder@jacobs-university.de

Anuj Sehgal
Jacobs University
Campus Ring 1
Bremen 28759
Germany

EMail: s.anuj@jacobs-university.de

Tina Tsou
Huawei Technologies (USA)
2330 Central Expressway
Santa Clara CA 95050
USA

EMail: tina.tsou.zouting@huawei.com

Cathy Zhou
Huawei Technologies
Bantian, Longgang District
Shenzhen 518129
P.R. China

EMail: cathyzhou@huawei.com

IPv6 over Networks of Resource-constrained Nodes (6lo) WG A. Brandt
Internet-Draft J. Buron
Intended status: Standards Track Sigma Designs
Expires: May 3, 2015 October 30, 2014

Transmission of IPv6 packets over ITU-T G.9959 Networks
draft-ietf-6lo-lowpanz-08

Abstract

This document describes the frame format for transmission of IPv6 packets and a method of forming IPv6 link-local addresses and statelessly autoconfigured IPv6 addresses on ITU-T G.9959 networks.

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].

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, 2015.

Copyright Notice

Copyright (c) 2014 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect

to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	3
1.1. Terms used	3
2. G.9959 parameters to use for IPv6 transport	5
2.1. Addressing mode	5
2.2. IPv6 Multicast support	6
2.3. G.9959 MAC PDU size and IPv6 MTU	6
2.4. Transmission status indications	7
2.5. Transmission security	7
3. 6LoWPAN Adaptation Layer and Frame Format	7
3.1. Dispatch Header	8
4. 6LoWPAN addressing	9
4.1. Stateless Address Autoconfiguration of routable IPv6 addresses	9
4.2. IPv6 Link Local Address	9
4.3. Unicast Address Mapping	10
4.4. On the use of Neighbor Discovery technologies	10
4.4.1. Prefix and CID management (Route-over)	11
4.4.2. Prefix and CID management (Mesh-under)	11
5. Header Compression	12
6. IANA Considerations	13
7. Security Considerations	13
8. Privacy Considerations	13
9. Acknowledgements	14
10. References	14
10.1. Normative References	14
10.2. Informative References	15
Appendix A. G.9959 6LoWPAN datagram example	16
Appendix B. Change Log	20
B.1. Changes since -00	20
B.2. Changes since -01	20
B.3. Changes since -02	21
B.4. Changes since -03	21
B.5. Changes since -04	22
B.6. Changes since -05	22
B.7. Changes since -06	22
B.8. Changes since -07	22
Authors' Addresses	23

1. Introduction

The ITU-T G.9959 recommendation [G.9959] targets low-power Personal Area Networks (PANs). This document defines the frame format for transmission of IPv6 [RFC2460] packets as well as the formation of IPv6 link-local addresses and statelessly autoconfigured IPv6 addresses on G.9959 networks.

The general approach is to adapt elements of [RFC4944] to G.9959 networks. G.9959 provides a Segmentation and Reassembly (SAR) layer for transmission of datagrams larger than the G.9959 MAC PDU.

[RFC6775] updates [RFC4944] by specifying 6LoWPAN optimizations for IPv6 Neighbor Discovery (ND) (originally defined by [RFC4861]). This document limits the use of [RFC6775] to prefix and Context ID assignment. An IID may be constructed from a G.9959 link-layer address, leading to a "link-layer-derived IPv6 address". If using that method, Duplicate Address Detection (DAD) is not needed. Alternatively, IPv6 addresses may be assigned centrally via DHCP, leading to a "non-link-layer-derived IPv6 address". Address registration is only needed in certain cases.

In addition to IPv6 application communication, the frame format defined in this document may be used by IPv6 routing protocols such as RPL [RFC6550] or P2P-RPL [RFC6997] to implement IPv6 routing over G.9959 networks.

The encapsulation frame defined by this specification may optionally be transported via mesh routing below the 6LoWPAN layer. Mesh-under and route-over routing protocol specifications are out of scope of this document.

1.1. Terms used

6LoWPAN: IPv6-based Low-power Personal Area Network

ABR: Authoritative 6LBR ([RFC6775])

Ack: Acknowledgement

AES: Advanced Encryption Scheme

CID: Context Identifier ([RFC6775])

DAD: Duplicate Address Detection ([RFC6775])

DHCPv6: Dynamic Host Configuration Protocol for IPv6 ([RFC3315])

EUI-64: Extended Unique Identifier ([EUI64])

G.9959: Short range, narrow-band digital radiocommunication transceiver ([G.9959])

GHC: Generic Header Compression ([RFC_TBD_GHC])

HomeID: G.9959 Link-Layer Network Identifier

IID: Interface IDentifier

Link-layer-derived address: IPv6 Address constructed on basis of link layer address information

MAC: Media Access Control

Mesh-under: Forwarding via mesh routing below the 6LoWPAN layer

MTU: Maximum Transmission Unit

ND: Neighbor discovery ([RFC4861], [RFC6775])

NodeID: G.9959 Link-Layer Node Identifier

Non-link-layer-derived address: IPv6 Address assigned by a managed process, e.g. DHCPv6.

NVM: Non-volatile Memory

P2P-RPL: Reactive Discovery of Point-to-Point Routes in Low-Power and Lossy Networks ([RFC6997])

PAN: Personal Area Network

PDU: Protocol Data Unit

PHY: Physical Layer

RA: Router Advertisement ([RFC4861], [RFC6775])

Route-over: Forwarding via IP routing above the 6LoWPAN layer

RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks ([RFC6550])

SAR: G.9959 Segmentation And Reassembly

ULA: Unique Local Address [RFC4193]

2. G.9959 parameters to use for IPv6 transport

This chapter outlines properties applying to the PHY and MAC of G.9959 and how to use these for IPv6 transport.

2.1. Addressing mode

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 which is identified by one or more IPv6 prefixes.

An IPv6 host MUST construct its link-local IPv6 address from the link-layer-derived IID in order to facilitate IP header compression as described in [RFC6282].

A node interface MAY support the M flag of the RA message for the construction of routable IPv6 addresses. A cost optimized node implementation may save memory by skipping support for the M flag. The M flag MUST be interpreted as defined in Figure 1.

M Flag support	M flag value	Required node behavior
No	(ignore)	Node MUST use link-layer-derived addressing
Yes	0	Node MUST use link-layer-derived addressing
	1	Node MUST use DHCPv6 based addressing and Node MUST comply fully with [RFC6775]

Figure 1: RA M flag support and interpretation

A node that uses DHCPv6 based addressing MUST comply fully with the text of [RFC6775].

If DHCPv6 based addressing is used, the DHCPv6 client must use a DUID of type DUID-UUID, as described in [RFC6355]. The UUID used in the DUID-UUID must be generated as specified in [RFC4122], section 4.5, starting at the second paragraph in that section (the 47-bit random number-based UUID). The DUID must be stored persistently by the node as specified in section 3 of [RFC6355].

A word of caution: since HomeIDs and NodeIDs are handed out by a network controller function during inclusion, identifier validity and

uniqueness is limited by the lifetime of the network membership. This can be cut short by a mishap occurring to the network controller. Having a single point of failure at the network controller suggests that high-reliability network deployments may benefit from a redundant network controller function.

This warning applies to link-layer-derived addressing as well as to non-link-layer-derived addressing deployments.

2.2. IPv6 Multicast support

[RFC3819] recommends that IP subnetworks support (subnet-wide) multicast. G.9959 supports direct-range IPv6 multicast while subnet-wide multicast is not supported natively by G.9959. Subnet-wide multicast may be provided by an IP routing protocol or a mesh routing protocol operating below the 6LoWPAN layer. Routing protocol specifications are out of scope of this document.

IPv6 multicast packets MUST be carried via G.9959 broadcast.

As per [G.9959], this is accomplished as follows:

1. The destination HomeID of the G.9959 MAC PDU MUST be the HomeID of the network
2. The destination NodeID of the G.9959 MAC PDU MUST be the broadcast NodeID (0xff)

G.9959 broadcast MAC PDUs are only intercepted by nodes within the network identified by the HomeID.

2.3. G.9959 MAC PDU size and IPv6 MTU

IPv6 packets MUST be transmitted using G.9959 transmission profile R3 or higher.

[RFC2460] specifies that any link that cannot convey a 1280-octet packet in one piece, must provide link-specific fragmentation and reassembly at a layer below IPv6.

G.9959 provides Segmentation And Reassembly for payloads up to 1350 octets. IPv6 Header Compression [RFC6282] improves the chances that a short IPv6 packet can fit into a single G.9959 frame. Therefore, Section 3 specifies that [RFC6282] MUST be supported. With the mandatory link-layer security enabled, a G.9959 R3 MAC PDU may accommodate 6LoWPAN datagrams of up to 130 octets without triggering G.9959 Segmentation and Reassembly (SAR). Longer 6LoWPAN datagrams will lead to the transmission of multiple G.9959 PDUs.

2.4. Transmission status indications

The G.9959 MAC layer provides native acknowledgement and retransmission of MAC PDUs. The G.9959 SAR layer does the same for larger datagrams. A mesh routing layer may provide a similar feature for routed communication. An IPv6 routing stack communicating over G.9959 may utilize link-layer status indications such as delivery confirmation and Ack timeout from the MAC layer.

2.5. Transmission security

Implementations claiming conformance with this document MUST enable G.9959 shared network key security.

The shared network key is intended to address security requirements in the home at the normal security requirements level. For applications with high or very high requirements on confidentiality and/or integrity, additional application layer security measures for end-to-end authentication and encryption may need to be applied. (The availability of the network relies on the security properties of the network key in any case)

3. 6LoWPAN Adaptation Layer and Frame Format

The 6LoWPAN encapsulation formats defined in this chapter are carried as payload in the G.9959 MAC PDU. IPv6 header compression [RFC6282] MUST be supported by implementations of this specification. Further, implementations MAY support Generic Header Compression (GHC) [RFC_TBD_GHC]. A node implementing [RFC_TBD_GHC] MUST probe its peers for GHC support before applying GHC compression.

All 6LoWPAN datagrams transported over G.9959 are prefixed by a 6LoWPAN encapsulation header stack. The 6LoWPAN payload follows this encapsulation header stack. Each header in the header stack contains a header type followed by zero or more header fields. An IPv6 header stack may contain, in the following order, addressing, hop-by-hop options, routing, fragmentation, destination options, and finally payload [RFC2460]. The 6LoWPAN header format is structured the same way. Currently only one payload option is defined for the G.9959 6LoWPAN header format.

The definition of 6LoWPAN headers consists of the dispatch value, the definition of the header fields that follow, and their ordering constraints relative to all other headers. Although the header stack structure provides a mechanism to address future demands on the 6LoWPAN adaptation layer, it is not intended to provide general purpose extensibility.

An example of a complete G.9959 6LoWPAN datagram can be found in Appendix A.

3.1. Dispatch Header

The dispatch header is shown below:

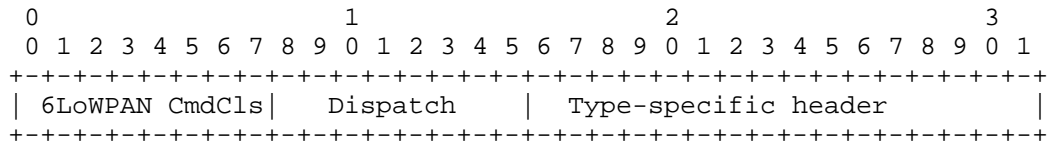


Figure 2: Dispatch Type and Header

6LoWPAN CmdCls: 6LoWPAN Command Class identifier. This field **MUST** carry the value 0x4F [G.9959]. The value is assigned by the ITU-T and specifies that the following bits are a 6LoWPAN encapsulated datagram. 6LoWPAN protocols **MUST** ignore the G.9959 frame if the 6LoWPAN Command Class identifier deviates from 0x4F.

Dispatch: Identifies the header type immediately following the Dispatch Header.

Type-specific header: A header determined by the Dispatch Header.

The dispatch value may be treated as an unstructured namespace. Only a few symbols are required to represent current 6LoWPAN functionality. Although some additional savings could be achieved by encoding additional functionality into the dispatch byte, these measures would tend to constrain the ability to address future alternatives.

Dispatch values used in this specification are compatible with the dispatch values defined by [RFC4944] and [RFC6282].

Pattern	Header Type	Reference
01 1xxxxx	6LoWPAN_IPHC - Compressed IPv6 Addresses	[RFC6282]

All other Dispatch values are unassigned in this document.

Figure 3: Dispatch values

6LoWPAN_IPHC: IPv6 Header Compression. Refer to [RFC6282].

4. 6LoWPAN addressing

IPv6 addresses may be autoconfigured from IIDs which may again be constructed from link-layer address information to save memory in devices and to facilitate efficient IP header compression as per [RFC6282]. Link-layer-derived addresses have a static nature and may involuntarily expose private usage data on public networks. Refer to Section 8.

A NodeID is mapped into an IEEE EUI-64 identifier as follows:

IID = 0000:00ff:fe00:YYXX

Figure 4: Constructing a compressible IID

where XX carries the G.9959 NodeID and YY is a one byte value chosen by the individual node. The default YY value MUST be zero. A node MAY use other values of YY than zero to form additional IIDs in order to instantiate multiple IPv6 interfaces. The YY value MUST be ignored when computing the corresponding NodeID (the XX value) from an IID.

The method of constructing IIDs from the link-layer address obviously does not support addresses assigned or constructed by other means. A node MUST NOT compute the NodeID from the IID if the first 6 bytes of the IID do not comply with the format defined in Figure 4. In that case, the address resolution mechanisms of RFC 6775 apply.

4.1. Stateless Address Autoconfiguration of routable IPv6 addresses

The IID defined above MUST be used whether autoconfiguring a ULA IPv6 address [RFC4193] or a globally routable IPv6 address [RFC3587] in G.9959 subnets.

4.2. IPv6 Link Local Address

The IPv6 link-local address [RFC4291] for a G.9959 interface is formed by appending the IID defined above to the IPv6 link local prefix FE80::/64.

The "Universal/Local" (U/L) bit MUST be set to zero in keeping with the fact that this is not a globally unique value [EUI64].

The resulting link local address is formed as follows:

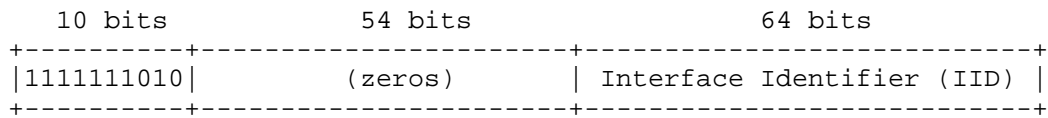


Figure 5: IPv6 Link Local Address

4.3. Unicast Address Mapping

The address resolution procedure for mapping IPv6 unicast addresses into G.9959 link-layer addresses follows the general description in Section 7.2 of [RFC4861]. The Source/Target Link-layer Address option MUST have the following form when the link layer is G.9959.

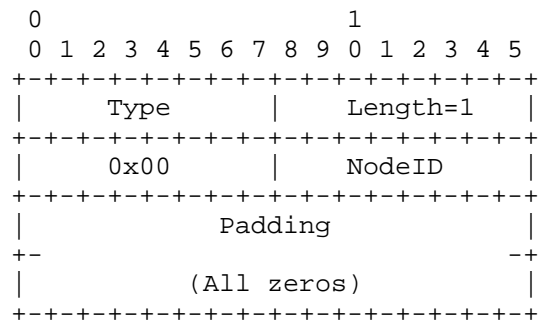


Figure 6: IPv6 Unicast Address Mapping

Option fields:

Type: The value 1 signifies the Source Link-layer address. The value 2 signifies the Destination 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 always 1 for G.9959 NodeIDs.

NodeID: This is the G.9959 NodeID the actual interface currently responds to. The link-layer address may change if the interface joins another network at a later time.

4.4. On the use of Neighbor Discovery technologies

[RFC4861] specifies how IPv6 nodes may resolve link layer addresses from IPv6 addresses via the use of link-local IPv6 multicast. [RFC6775] is an optimization of [RFC4861], specifically targeting

6LoWPAN networks. [RFC6775] defines how a 6LoWPAN node may register IPv6 addresses with an authoritative border router (ABR). Mesh-under networks MUST NOT use [RFC6775] address registration. However, [RFC6775] address registration MUST be used if the first 6 bytes of the IID do not comply with the format defined in Figure 3.

4.4.1. Prefix and CID management (Route-over)

In route-over environments, IPv6 hosts MUST use [RFC6775] address registration. A node implementation for route-over operation MAY use RFC6775 mechanisms for obtaining IPv6 prefixes and corresponding header compression context information [RFC6282]. RFC6775 Route-over requirements apply with no modifications.

4.4.2. Prefix and CID management (Mesh-under)

An implementation for mesh-under operation MUST use [RFC6775] mechanisms for managing IPv6 prefixes and corresponding header compression context information [RFC6282]. [RFC6775] Duplicate Address Detection (DAD) MUST NOT be used, since the link-layer inclusion process of G.9959 ensures that a NodeID is unique for a given HomeID.

With this exception and the specific redefinition of the RA Router Lifetime value 0xFFFF (refer to Section 4.4.2.3), the text of the following subsections is in compliance with [RFC6775].

4.4.2.1. Prefix assignment considerations

As stated by [RFC6775], an ABR is responsible for managing prefix(es). Global routable prefixes may change over time. It is RECOMMENDED that a ULA prefix is assigned to the 6LoWPAN subnet to facilitate stable site-local application associations based on IPv6 addresses. A node MAY support the M flag of the RA message. This influences the way IPv6 addresses are assigned. Refer to Section 2.1 for details.

4.4.2.2. Robust and efficient CID management

The 6LoWPAN Context Option (6CO) is used according to [RFC6775] in an RA to disseminate Context IDs (CID) to use for compressing prefixes. One or more prefixes and corresponding Context IDs MUST be assigned during initial node inclusion.

When updating context information, a CID may have its lifetime set to zero to obsolete it. The CID MUST NOT be reused immediately; rather the next vacant CID should be assigned. Header compression based on CIDs MUST NOT be used for RA messages carrying Context Information.

An expired CID and the associated prefix MUST NOT be reset but rather retained in receive-only mode if there is no other current need for the CID value. This will allow an ABR to detect if a sleeping node without clock uses an expired CID and in response, the ABR MUST return an RA with fresh Context Information to the originator.

4.4.2.3. Infinite prefix lifetime support for island-mode networks

Nodes MUST renew the prefix and CID according to the lifetime signaled by the ABR. [RFC6775] specifies that the maximum value of the RA Router Lifetime field MAY be up to 0xFFFF. This document further specifies that the value 0xFFFF MUST be interpreted as infinite lifetime. This value MUST NOT be used by ABRs. Its use is only intended for a sleeping network controller; for instance a battery powered remote control being master for a small island-mode network of light modules.

5. Header Compression

IPv6 header compression [RFC6282] MUST be implemented and [RFC_TBD_GHC] compression for higher layers MAY be implemented. This section will simply identify substitutions that should be made when interpreting the text of [RFC6282] and [RFC_TBD_GHC].

In general the following substitutions should be made:

- o Replace "802.15.4" with "G.9959"
- o Replace "802.15.4 short address" with "<Interface><G.9959 NodeID>"
- o Replace "802.15.4 PAN ID" with "G.9959 HomeID"

When a 16-bit address is called for (i.e., an IEEE 802.15.4 "short address") it MUST be formed by prepending an Interface label byte to the G.9959 NodeID:

```

      0                               1
      0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5
+---+---+---+---+---+---+---+---+---+
|   Interface   |   NodeID   |
+---+---+---+---+---+---+---+---+

```

A transmitting node may be sending to an IPv6 destination address which can be reconstructed from the link-layer destination address. If the Interface number is zero (the default value), all IPv6 address bytes may be elided. Likewise, the Interface number of a fully elided IPv6 address (i.e. SAM/DAM=11) may be reconstructed to the value zero by a receiving node.

64 bit 802.15.4 address details do not apply.

6. IANA Considerations

This document makes no request of IANA.

Note to RFC Editor: this section may be removed on publication as an RFC.

7. Security Considerations

The method of derivation of Interface Identifiers from 8-bit NodeIDs preserves uniqueness within the network. However, there is no protection from duplication through forgery. Neighbor Discovery in G.9959 links may be susceptible to threats as detailed in [RFC3756]. G.9959 networks may feature mesh routing. This implies additional threats due to ad hoc routing as per [KW03]. G.9959 provides capability for link-layer security. G.9959 nodes MUST use link-layer security with a shared key. Doing so will alleviate the majority of threats stated above. A sizeable portion of G.9959 devices is expected to always communicate within their PAN (i.e., within their subnet, in IPv6 terms). In response to cost and power consumption considerations, these devices will typically implement the minimum set of features necessary. Accordingly, security for such devices may rely on the mechanisms defined at the link layer by G.9959. G.9959 relies on the Advanced Encryption Standard (AES) for authentication and encryption of G.9959 frames and further employs challenge-response handshaking to prevent replay attacks.

It is also expected that some G.9959 devices (e.g. billing and/or safety critical products) will implement coordination or integration functions. These may communicate regularly with IPv6 peers outside the subnet. Such IPv6 devices are expected to secure their end-to-end communications with standard security mechanisms (e.g., IPsec, TLS, etc).

8. Privacy Considerations

IP addresses may be used to track devices on the Internet, which in turn can be linked to individuals and their activities. Depending on the application and the actual use pattern, this may be undesirable. To impede tracking, globally unique and non-changing characteristics of IP addresses should be avoided, e.g. by frequently changing the global prefix and avoiding unique link-layer-derived IIDs in addresses.

Some link layers use a 48-bit or a 64-bit link layer address which uniquely identifies the node on a global scale regardless of global

prefix changes. The risk of exposing a G.9959 device from its link-layer-derived IID is limited because of the short 8-bit link layer address.

While intended for central address management, DHCPv6 address assignment also decouples the IPv6 address from the link layer address. Addresses may be made dynamic by the use of a short DHCP lease period and an assignment policy which makes the DHCP server hand out a fresh IP address every time. For enhanced privacy, the DHCP assigned addresses should be logged only for the duration of the lease provided the implementation also allows logging for longer durations as per the operational policies.

It should be noted that privacy and frequently changing address assignment comes at a cost. Non-link-layer-derived IIDs require the use of address registration and further, non-link-layer-derived IIDs cannot be compressed, which leads to longer datagrams and increased link layer segmentation. Finally, frequent prefix changes necessitate more Context Identifier updates, which not only leads to increased traffic but also may affect the battery lifetime of sleeping nodes.

9. Acknowledgements

Thanks to the authors of RFC 4944 and RFC 6282 and members of the IETF 6LoWPAN working group; this document borrows extensively from their work. Thanks to Erez Ben-Tovim, Erik Nordmark, Kerry Lynn, Michael Richardson, Tommas Jess Christensen for useful comments. Thanks to Carsten Bormann for extensive feedback which improved this document significantly. Thanks to Brian Haberman for pointing out unclear details.

10. References

10.1. Normative References

- [G.9959] "G.9959 (02/12) + G.9959 Amendment 1 (10/13): Short range, narrow-band digital radiocommunication transceivers", February 2012.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC2460] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", RFC 2460, December 1998.

- [RFC4122] Leach, P., Mealling, M., and R. Salz, "A Universally Unique Identifier (UUID) URN Namespace", RFC 4122, July 2005.
- [RFC4193] Hinden, R. and B. Haberman, "Unique Local IPv6 Unicast Addresses", RFC 4193, October 2005.
- [RFC4291] Hinden, R. and S. Deering, "IP Version 6 Addressing Architecture", RFC 4291, February 2006.
- [RFC4861] Narten, T., Nordmark, E., Simpson, W., and H. Soliman, "Neighbor Discovery for IP version 6 (IPv6)", RFC 4861, September 2007.
- [RFC4944] Montenegro, G., Kushalnagar, N., Hui, J., and D. Culler, "Transmission of IPv6 Packets over IEEE 802.15.4 Networks", RFC 4944, September 2007.
- [RFC6282] Hui, J. and P. Thubert, "Compression Format for IPv6 Datagrams over IEEE 802.15.4-Based Networks", RFC 6282, September 2011.
- [RFC6355] Narten, T. and J. Johnson, "Definition of the UUID-Based DHCPv6 Unique Identifier (DUID-UUID)", RFC 6355, August 2011.
- [RFC6775] Shelby, Z., Chakrabarti, S., Nordmark, E., and C. Bormann, "Neighbor Discovery Optimization for IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs)", RFC 6775, November 2012.
- [RFC_TBD_GHC] "draft-ietf-6lo-ghc: 6LoWPAN Generic Compression of Headers and Header-like Payloads", September 2014.

10.2. Informative References

- [EUI64] IEEE, "GUIDELINES FOR 64-BIT GLOBAL IDENTIFIER (EUI-64) REGISTRATION AUTHORITY", IEEE Std <http://standards.ieee.org/regauth/oui/tutorials/EUI64.html>, November 2012.
- [KW03] Elsevier's AdHoc Networks Journal, "Secure Routing in Sensor Networks: Attacks and Countermeasures", Special Issue on Sensor Network Applications and Protocols vol 1, issues 2-3", , September 2003.

- [RFC3315] Droms, R., Bound, J., Volz, B., Lemon, T., Perkins, C., and M. Carney, "Dynamic Host Configuration Protocol for IPv6 (DHCPv6)", RFC 3315, July 2003.
- [RFC3587] Hinden, R., Deering, S., and E. Nordmark, "IPv6 Global Unicast Address Format", RFC 3587, August 2003.
- [RFC3756] Nikander, P., Kempf, J., and E. Nordmark, "IPv6 Neighbor Discovery (ND) Trust Models and Threats", RFC 3756, May 2004.
- [RFC3819] Karn, P., Bormann, C., Fairhurst, G., Grossman, D., Ludwig, R., Mahdavi, J., Montenegro, G., Touch, J., and L. Wood, "Advice for Internet Subnetwork Designers", BCP 89, RFC 3819, July 2004.
- [RFC6550] Winter, T., Thubert, P., 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, March 2012.
- [RFC6997] Goyal, M., Baccelli, E., Philipp, M., Brandt, A., and J. Martocci, "Reactive Discovery of Point-to-Point Routes in Low-Power and Lossy Networks", RFC 6997, August 2013.

Appendix A. G.9959 6LoWPAN datagram example

This example outlines each individual bit of a sample IPv6 UDP packet arriving to a G.9959 node from a host in the Internet via a PAN border router.

In the G.9959 PAN, the complete frame has the following fields.

G.9959:

```
+-----+-----+-----+-----+-----+-----+...
|HomeID|SrcNodeID|FrmControl|Len|SeqNo|DestNodeID|
+-----+-----+-----+-----+-----+-----+...
```

6LoWPAN:

```
...+-----+-----+-----+-----+-----+-----+...
|6LoWPAN CmdCls|6LoWPAN_IPHC Hdr|Compressed IPv6 headers|
...+-----+-----+-----+-----+-----+-----+...
```

6LoWPAN, TCP/UDP, App payload:

```
...+-----+-----+-----+-----+-----+-----+
|Uncompressed IPv6 headers|TCP/UDP/ICMP|App payload|
...+-----+-----+-----+-----+-----+-----+
```

The frame comes from the source IPv6 address
2001:0db8:ac10:ef01::ff:fe00:1206. The source prefix
2001:0db8:ac10:ef01/64 is identified by the IPHC CID = 3.

The frame is delivered in direct range from the gateway which has
source NodeID = 1. The Interface Identifier (IID) ff:fe00:1206 is
recognised as a link-layer-derived address and is compressed to the
16 bit value 0x1206.

The frame is sent to the destination IPv6 address
2001:0db8:27ef:42ca::ff:fe00:0004. The destination prefix
2001:0db8:27ef:42ca/64 is identified by the IPHC CID = 2.

The Interface Identifier (IID) ff:fe00:0004 is recognised as a link-
layer-derived address.

Thanks to the link-layer-derived addressing rules, the sender knows
that this is to be sent to G.9959 NodeID = 4; targeting the IPv6
interface instance number 0 (the default).

To reach the 6LoWPAN stack of the G.9959 node, (skipping the G.9959
header fields) the first octet must be the 6LoWPAN Command Class
(0x4F).


```

0
0 1 2 3 4 5 6 7 8
+---+---+---+---+---+---+---+---+...
|           0x4F           |
+---+---+---+---+---+---+---+---+...

```

The Dispatch header bits '011' advertises a compressed IPv6 header.

```

0                               1
0 1 2 3 4 5 6 7 8 9 0
+---+---+---+---+---+---+---+---+...
|           0x4F           | 0 1 1
+---+---+---+---+---+---+---+---+...

```

The following bits encode the first IPv6 header fields:

TF = '11' : Traffic Class and Flow Label are elided.
 NH = '1' : Next Header is elided
 HLIM = '10' : Hop limit is 64

```

0                               1
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5
+---+---+---+---+---+---+---+---+...
|           0x4F           | 0 1 1 1 1 1 1 0 |
+---+---+---+---+---+---+---+---+...

```

CID = '1' : CI data follows the DAM field
 SAC = '1' : Src addr uses stateful, context-based compression
 SAM = '10' : Use src CID and 16 bits for link-layer-derived addr
 M = '0' : Dest addr is not a multicast addr
 DAC = '1' : Dest addr uses stateful, context-based compression
 DAM = '11' : Use dest CID and dest NodeID to link-layer-derived addr

```

0                               1                               2
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+...
|           0x4F           | 0 1 1 1 1 1 1 0 | 1 1 1 0 0 1 1 1 |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+...

```

Address compression context identifiers:

SCI = 0x3

DCI = 0x2

```

      2          3
    4 5 6 7 8 9 0 1
...+--+--+--+--+--+...
  | 0x3 | 0x2 |
...+--+--+--+--+--+...
```

IPv6 header fields:

(skipping "version" field)

(skipping "Traffic Class")

(skipping "flow label")

(skipping "payload length")

IPv6 header address fields:

SrcIP = 0x1206 : Use SCI and 16 LS bits of link-layer-derived address

(skipping DestIP) - completely reconstructed from Dest NodeID and DCI

```

      2          3          4
    4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7
...+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+...
  | 0x3 | 0x2 | 0x12 | 0x06 |
...+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+...
```

Next header encoding for the UDP header:

Dispatch = '11110': Next Header dispatch code for UDP header

C = '0' : 16 bit checksum carried inline

P = '00' : Both src port and dest Port are carried in-line.

```

      4    5
    8 9 0 1 2 3 4 5
...+--+--+--+--+--+...
  | 1 1 1 1 0 | 0 | 0 |
...+--+--+--+--+--+...
```

UDP header fields:

src Port = 0x1234

dest port = 0x5678

```

      5           6           7           8
      6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7
...+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+...
  |      0x12      |      0x34      |      0x56      |      0x78      |
...+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+...

```

(skipping "length")

checksum = (actual checksum value depends on
the actual UDP payload)

```

              1
              0
      8 9      0
      8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3
...+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+...
  |      (UDP checksum)      |
...+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+...

```

Add your own UDP payload here...

Appendix B. Change Log

B.1. Changes since -00

- o Clarified that mesh-under routing may take place below the 6LoWPAN layer but that specific mesh-under routing protocols are not within the scope of this doc.
- o Clarified that RFC6282 IPv6 Header Compression MUST be supported.
- o Clarified the text of section 5.4 on the use of RFC6775 address registration in mesh-under networks.
- o Split 5.4.2 into multiple paragraphs.

B.2. Changes since -01

- o Added this Change Log
- o Editorial nits.

- o Made IPv6 Header Compression mandatory. Therefore, the Dispatch value "01 000001 - Uncompressed IPv6 Addresses" was removed from figure 2.
- o Changed SHOULD to MUST: An IPv6 host SHOULD construct its link-local IPv6 address and routable IPv6 addresses from the NodeID in order to facilitate IP header compression as described in [RFC6282].
- o Changed SHOULD NOT to MUST NOT: Mesh-under networks MUST NOT use [RFC6775] address registration.
- o Changed SHOULD NOT to MUST NOT: [RFC6775] Duplicate Address Detection (DAD) MUST NOT be used.
- o Changed SHOULD NOT to MUST NOT: The CID MUST NOT be reused immediately;
- o Changed SHOULD NOT to MUST NOT: An expired CID and the associated prefix MUST NOT be reset but rather retained in receive-only mode
- o Changed LBR -> ABR
- o Changed SHOULD to MUST: , the ABR MUST return an RA with fresh Context Information to the originator.
- o Changed SHOULD NOT to MUST NOT: This value MUST NOT be used by ABRs. Its use is only intended for a sleeping network controller.

B.3. Changes since -02

- o Editorial nits.
- o Moved text to the right section so that it does not prohibit DAD for Route-Over deployments.
- o Introduced RA M flag support so that nodes may be instructed to use DHCPv6 for centralized address assignment.
- o Added example appendix: Complete G.9959 6LoWPAN datagram composition with CID-based header compression.

B.4. Changes since -03

- o Corrected error in 6LoWPAN datagram example appendix: 64 hop limit in comment => also 64 hop limit in actual frame format.
- o Added section "Privacy Considerations"

B.5. Changes since -04

- o Text on RA M flag support was replaced with a table to improve clarity.
- o Added further details to text on achieving privacy addressing with DHCP.

B.6. Changes since -05

- o Term ABR now points to Authoritative 6LBR as defined by RFC6775.
- o Removed sentence "The G.9959 network controller function SHOULD be integrated in the ABR." as this was an implementation guideline with no relevance to network performance.
- o Clarifying that network controller function redundancy is relevant for network deployers; not user-level application designers.
- o Clarified that RFC2460 specifies that link layer must provide fragmentation if 1280 octet packets cannot be carried in one piece by the link layer.
- o Clarified that the 6LoWPAN CmdCls identifier value is assigned by the ITU-T.
- o Added reference to Privacy Considerations section from 6LoWPAN Addressing section.
- o Introducing optional GHC header compression.

B.7. Changes since -06

- o Added a note to section 5, that the mapping of 802.15.4 terms to similar G.9959 terms applies not only to RFC6282 but also to GHC.

B.8. Changes since -07

- o Added a note to the Privacy considerations section on avoiding DHCP logging.
- o Added requirements for forming a UUID if DHCPv6 address assignment is used.

Authors' Addresses

Anders Brandt
Sigma Designs
Emdrupvej 26A, 1.
Copenhagen O 2100
Denmark

Email: anders_brandt@sigmadesigns.com

Jakob Buron
Sigma Designs
Emdrupvej 26A, 1.
Copenhagen O 2100
Denmark

Email: jakob_buron@sigmadesigns.com

Internet Engineering Task Force
Internet-Draft
Intended status: Informational
Expires: September 2, 2015

M. Ersue, Ed.
Nokia Networks
D. Romascanu
Avaya
J. Schoenwaelder
Jacobs University Bremen
U. Herberg
March 1, 2015

Management of Networks with Constrained Devices: Problem Statement and
Requirements
draft-ietf-opsawg-coman-probstate-reqs-05

Abstract

This document provides a problem statement, deployment and management topology options as well as requirements addressing the different use cases of the management of networks where constrained devices are involved.

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 2, 2015.

Copyright Notice

Copyright (c) 2015 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect

to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	3
1.1. Overview	3
1.2. Terminology	4
1.3. Network Types and Characteristics in Focus	5
1.4. Constrained Device Deployment Options	9
1.5. Management Topology Options	9
1.6. Managing the Constrainedness of a Device or Network	10
1.7. Configuration and Monitoring Functionality Levels	13
2. Problem Statement	14
3. Requirements on the Management of Networks with Constrained Devices	16
3.1. Management Architecture/System	17
3.2. Management Protocols and Data Models	21
3.3. Configuration Management	24
3.4. Monitoring Functionality	26
3.5. Self-management	31
3.6. Security and Access Control	32
3.7. Energy Management	34
3.8. Software Distribution	36
3.9. Traffic Management	36
3.10. Transport Layer	37
3.11. Implementation Requirements	39
4. IANA Considerations	40
5. Security Considerations	40
6. Acknowledgments	41
7. Informative References	41
Appendix A. Change Log	42
A.1. draft-ietf-opsawg-coman-probstate-reqs-04 - draft-ietf-opsawg-coman-probstate-reqs-05	42
A.2. draft-ietf-opsawg-coman-probstate-reqs-03 - draft-ietf-opsawg-coman-probstate-reqs-04	42
A.3. draft-ietf-opsawg-coman-probstate-reqs-02 - draft-ietf-opsawg-coman-probstate-reqs-03	42
A.4. draft-ietf-opsawg-coman-probstate-reqs-01 - draft-ietf-opsawg-coman-probstate-reqs-02	43
A.5. draft-ietf-opsawg-coman-probstate-reqs-00 - draft-ietf-opsawg-coman-probstate-reqs-01	43
A.6. draft-ersue-constrained-mgmt-03 - draft-ietf-opsawg-coman-probstate-reqs-00	44
A.7. draft-ersue-constrained-mgmt-02-03	44
A.8. draft-ersue-constrained-mgmt-01-02	45

A.9. draft-ersue-constrained-mgmt-00-01	46
Authors' Addresses	46

1. Introduction

1.1. Overview

Constrained devices, aka. sensor, smart object, or smart device, with limited CPU, memory, and power resources, can constitute a network. Such a network of constrained devices itself may be constrained or challenged, e.g., with unreliable or lossy channels, wireless technologies with limited bandwidth and a dynamic topology, needing the service of a gateway or proxy to connect to the Internet. In other scenarios, the constrained devices can be connected to a non-constrained network using off-the-shelf protocol stacks.

Constrained devices might be in charge of gathering information in diverse settings including natural ecosystems, buildings, and factories, and send the information to one or more server stations. Constrained devices may also work under severe resource constraints such as limited battery and computing power, little memory and insufficient wireless bandwidth, and communication capabilities. A central entity, e.g., a base station or controlling server, might have more computational and communication resources and can act as a gateway between the constrained devices and the application logic in the core network.

Today diverse size of constrained devices with different resources and capabilities are being connected. Mobile personal gadgets, building-automation devices, cellular phones, Machine-to-machine (M2M) devices, etc. benefit from interacting with other "things" in the near or somewhere in the Internet. With this the Internet of Things (IoT) becomes a reality, build up of uniquely identifiable objects (things). And over the next decade, this could grow to trillions of constrained devices and will greatly increase the Internet's size and scope.

Network management is characterized by monitoring network status, detecting faults, and inferring their causes, setting network parameters, and carrying out actions to remove faults, maintain normal operation, and improve network efficiency and application performance. The traditional network monitoring application periodically collects information from a set of elements that are needed to manage, processes the data, and presents them to the network management users. Constrained devices, however, often have limited power, low transmission range, and might be unreliable. They might also need to work in hostile environments with advanced security requirements or need to be used in harsh environments for a

long time without supervision. Due to such constraints, the management of a network with constrained devices faces different type of challenges compared to the management of a traditional IP network.

The IETF has already done substantial standardization work to enable the communication in IP networks and to manage such networks as well as the manifold type of nodes in these networks [RFC6632]. However, the IETF so far has not developed any specific technologies for the management of constrained devices and the networks comprised by constrained devices. IP-based sensors or constrained devices in such an environment, i.e., devices with very limited memory, CPU, and energy resources, use nowadays application-layer protocols in an ad-hoc manner to do simple resource management and monitoring.

This document provides a problem statement and lists requirements for the different use cases of management of a network with constrained devices. Section 1.3 and Section 1.5 describe different topology options for the networking and management of constrained devices. Section 2 provides a problem statement on the issue of the management of networked constrained devices. Section 3 lists requirements on the management of applications and networks with constrained devices. Note that the requirements listed in Section 3 have been separated from the context in which they may appear. Depending on the concrete circumstances, an implementer may decide to address a certain relevant subset of the requirements.

The use cases in the context of networks with constrained devices can be found in the companion document [COM-USE]. This informational document provides a list of objectives for discussions and does not aim to be a strict requirements document for all use cases. In fact, there likely is not a single solution that works equally well for all the use cases.

1.2. Terminology

Concerning constrained devices and networks this document generally builds on the terminology defined in [RFC7228], where the terms Constrained Device, Constrained Network, etc. are defined.

The following terms are additionally used throughout this documentation:

AMI: (Advanced Metering Infrastructure) A system including hardware, software, and networking technologies that measures, collects, and analyzes energy usage, and communicates with a hierarchically deployed network of metering devices, either on request or on a schedule.

C0: Class 0 constrained device as defined in Section 3. of [RFC7228].

C1: Class 1 constrained device as defined in Section 3. of [RFC7228].

C2: Class 2 constrained device as defined in Section 3. of [RFC7228].

Network of Constrained Devices: A network to which constrained devices are connected that may or may not be a Constrained Network (see [RFC7228] for the definition of the term Constrained Network).

M2M: (Machine to Machine) stands for the automatic data transfer between devices of different kind. In M2M scenarios a device (such as a sensor or meter) captures an event, which is relayed through a network (wireless, wired or hybrid) to an application.

MANET: Mobile Ad-hoc Networks [RFC2501], a self-configuring and infrastructureless network of mobile devices connected by wireless technologies.

Smart Grid: An electrical grid that uses communication technologies to gather and act on information in an automated fashion to improve the efficiency, reliability and sustainability of the production and distribution of electricity.

Smart Meter: An electrical meter in the context of a Smart Grid.

For a detailed discussion on the constrained networks as well as classes of constrained devices and their capabilities please see [RFC7228].

1.3. Network Types and Characteristics in Focus

In this document we differentiate following types of networks concerning their transport and communication technologies:

(Note that a network in general can involve constrained and non-constrained devices.)

1. Wireline non-constrained networks, e.g., an Ethernet-LAN with constrained and non-constrained devices involved.
2. A combination of wireline and wireless networks, possibly with a multi-hop connectivity between constrained devices, utilizing dynamic routing in both the wireless and wireline portions of the

network. Such networks usually support highly distributed applications with many nodes (e.g., environmental monitoring) and tend to deal with large-scale multipoint-to-point systems. Wireless Mesh Networks (WMN), as a specific variant, use off-the-shelf radio technology such as Wi-Fi, WiMax, and cellular 3G/4G. WMNs are reliable based on the redundancy they offer and have often a more planned deployment to provide dynamic and cost effective connectivity over a certain geographic area.

3. A combination of wireline and wireless networks with point-to-point or point-to-multipoint communication generally with single-hop connectivity to constrained devices, utilizing static routing over the wireless network. Such networks support short-range, point-to-point, low-data-rate, source-to-sink type of applications such as RFID systems, light switches, fire and smoke detectors, and home appliances. This type of networks also support confined short-range spaces such as a home, a factory, a building, or the human body. IEEE 802.15.1 (Bluetooth) and IEEE 802.15.4 are well-known examples of applicable standards for such networks. By using 6LowPAN (IPv6 over Low-Power Wireless Personal Area Networks) [RFC4919] and RPL (IPv6 Routing Protocol for Low-Power and Lossy Networks) [RFC6550] on top of IEEE 802.15.4, multi-hop connectivity and dynamic routing can be achieved. With RPL the IETF has specified a proactive route-over architecture where routing and forwarding is implemented at the network layer. The protocol provides a mechanism whereby multipoint-to-point, point-to-multipoint and point-to-point traffic are supported.
4. Self-configuring infrastructureless networks of mobile devices (e.g., Mobile Adhoc networks, MANET) are a particular type of network connected by wireless technologies. Infrastructureless networks are mostly based on point-to-point communications of devices moving independently in any direction and changing the links to other devices frequently. Such devices do act as a router to forward traffic unrelated to their own use.

Wireline non-constrained networks with constrained and non-constrained devices are mainly used for specific applications like Building Automation or Infrastructure Monitoring. Wireline and wireless networks with multi-hop or point-to-multipoint connectivity are used e.g., for environmental monitoring as well as transport and mobile applications.

Furthermore different network characteristics are determined by multiple dimensions: dynamicity of the topology, bandwidth, and loss rate. In the following, each dimension is explained, and networks in scope for this document are outlined:

Network Topology:

The topology of a network can be represented as a graph, with edges (i.e., links) and vertices (routers and hosts). Examples of different topologies include "star" topologies (with one central node and multiple nodes in one hop distance), tree structures (with each node having exactly one parent), directed acyclic graphs (with each node having one or more parents), clustered topologies (where one or more "cluster heads" are responsible for a certain area of the network), mesh topologies (fully distributed), etc.

Management protocols may take advantage of specific network topologies, for example by distributing large-scale management tasks amongst multiple distributed network management stations (e.g., in case of a mesh topology), or by using a hierarchical management approach (e.g., in case of a tree or clustered topology). These different management topology options are described in Section 1.6.

Note that in certain network deployments, such as community ad hoc networks (see the use case "Community Network Applications" in [COM-USE]), the topology is not pre-planned, and thus may be unknown for management purposes. In other use cases, such as industrial applications (see the use case "Industrial Applications" in [COM-USE]), the topology may be designed in advance and therefore taken advantage of when managing the network.

Dynamicity of the network topology:

The dynamicity of the network topology determines the rate of change of the graph as a function of time. Such changes can occur due to different factors, such as mobility of nodes (e.g., in MANETs or cellular networks), duty cycles (for low-power devices enabling their network interface only periodically to transmit or receive packets), or unstable links (in particular wireless links with strongly fluctuating link quality).

Examples of different levels of dynamicity of the topology are Ethernets (with typically a very static topology) on the one side, and low-power and lossy networks (LLNs) on the other side. LLNs nodes are often duty-cycled and operate on unreliable wireless links and are potentially mobile (e.g., for sensor networks).

The more dynamic the topology is, the more have routing, transport and application layer protocols to cope with interrupted connectivity and/or longer delays. For example, management protocols (with a given underlying transport protocol) that expect continuous session flows without changes of routes during a communication flow, may fail to operate.

Networks with a very low dynamicity (e.g., Ethernet) with no or infrequent topology changes (e.g., less than once every 30 minutes), are in-scope of this document if they are used with constrained devices (see e.g., the use case "Building Automation" in [COM-USE]).

Traffic flows:

The traffic flow in a network determines from which sources data traffic is sent to which destinations in the network. Several different traffic flows are defined in [RFC7102], including "point-to-point" (P2P), "multipoint-to-point" (MP2P), and "point-to-multipoint" (P2MP) flows as:

- o P2P: Point-To-Point. This refers to traffic exchanged between two nodes (regardless of the number of hops between the two nodes).
- o P2MP: Point-to-Multipoint traffic refers to traffic between one node and a set of nodes. This is similar to the P2MP concept in Multicast or MPLS Traffic Engineering.
- o MP2P: Multipoint-to-Point is used to describe a particular traffic pattern (e.g., MP2P flows collecting information from many nodes flowing inwards towards a collecting sink).

If one of these traffic patterns is predominant in a network, protocols (routing, transport, application) may be optimized for the specific traffic flow. For example, in a network with a tree topology and MP2P traffic, collection tree protocols are efficient to send data from the leaves of the tree to the root of the tree, via each node's parent.

Bandwidth:

The bandwidth of the network is the amount of data that can be sent per unit of time between two communication end-points. It is usually determined by the link with the minimum bandwidth on the path from the source to the destination of data packets. The bandwidth in networks can range from a few Kilobytes per second (such as on some 802.15.4 link layers) to many Gigabytes per second (e.g., on fiber optics).

For management purposes, the management protocol typically requires to send information between the network management station and the clients, for monitoring or control purposes. If the available bandwidth is insufficient for the management protocol, packets will be buffered and eventually dropped, and thus management is not possible with such a protocol.

Networks without bandwidth limitation (e.g., Ethernet) are in-scope of this document if they are used with constrained devices (see the use case "Building Automation" in [COM-USE]).

Loss rate:

The loss rate (or bit error rate) is the number of bit errors divided by the total number of bits transmitted. For wired networks, loss rates are typically extremely low, e.g., around 10^{-12} or 10^{-13} for the latest 10Gbit Ethernet. For wireless networks, such as 802.15.4, the bit error rate can be as high as 10^{-1} to 1 in case of interferences. Even when using a reliable transport protocol, management operations can fail if the loss rate is too high, unless they are specifically designed to cope with these situations.

1.4. Constrained Device Deployment Options

We differentiate following deployment options for the constrained devices:

- o A network of constrained devices that communicate with each other,
- o Constrained devices, which are connected directly to an IP network,
- o A network of constrained devices which communicate with a gateway or proxy with more communication capabilities acting possibly as a representative of the device to entities in the non-constrained network
- o Constrained devices, which are connected to the Internet or an IP network via a gateway/proxy
- o A hierarchy of constrained devices, e.g., a network of C0 devices connected to one or more C1 devices - connected to one or more C2 devices - connected to one or more gateways - connected to some application servers or NMS system
- o The possibility of device grouping (possibly in a dynamic manner) such as that the grouped devices can act as one logical device at the edge of the network and one device in this group can act as the managing entity

1.5. Management Topology Options

We differentiate following options for the management of networks of constrained devices:

- o A network of constrained devices managed by one central manager. A logically centralized management might be implemented in a hierarchical fashion for scalability and robustness reasons. The manager and the management application logic might have a gateway/proxy in between or might be on different nodes in different networks, e.g., management application running on a cloud server.
- o Distributed management, where a network of constrained devices is managed by more than one manager. Each manager controls a subnetwork and may communicate directly with other manager stations in a cooperative fashion. The distributed management may be weakly distributed, where functions are broken down and assigned to many managers dynamically, or strongly distributed, where almost all managed things have embedded management functionality and explicit management disappears, which usually comes with the price that the strongly distributed management logic now needs to be managed.
- o Hierarchical management, where a hierarchy of networks with constrained devices are managed by the managers at their corresponding hierarchy level. I.e., each manager is responsible for managing the nodes in its sub-network. It passes information from its sub-network to its higher-level manager, and disseminates management functions received from the higher-level manager to its sub-network. Hierarchical management is essentially a scalability mechanism, logically the decision-making may be still centralized.

1.6. Managing the Constrainedness of a Device or Network

The capabilities of a constrained device or network and the constrainedness thereof influence and have an impact on the requirements for the management of such network or devices.

Note that the list below gives examples and does not claim completeness.

A constrained device:

- o might only support an unreliable (e.g. lossy) radio link, i.e., the client and server of a management protocol need to gracefully handle incomplete command exchanges or missing commands.
- o might only be able to go online from time-to-time, where it is reachable, i.e., a command might be necessary to repeat after a longer timeout or the timeout value with which one endpoint waits on a response needs to be sufficiently high.

- o might only be able to support a limited operating time (e.g., based on the available battery), or may behave as 'sleepy endpoints' setting their network links to a disconnected state during long periods of time i.e., the devices need to economize their energy usage with suitable mechanisms and the managing entity needs to monitor and control the energy status of the constrained devices it manages.
- o might only be able to support one simple communication protocol, i.e., the management protocol needs to be possible to downscale from constrained (C2) to very constrained (C0) devices with modular implementation and a very basic version with just a few simple commands.
- o might only be able to support a communication protocol, which is not IP-based.
- o might only be able to support limited or no user and/or transport security, i.e., the management system needs to support a less-costly and simple but sufficiently secure authentication mechanism.
- o might not be able to support compression and decompression of exchanged data based on limited CPU power, i.e., an intermediary entity which is capable of data compression should be able to communicate with both, devices that support data compression (e.g., C2) and devices that do not support data compression (e.g., C1 and C0).
- o might only be able to support a simple encryption, i.e., it would be beneficial if the devices use cryptographic algorithms that are supported in hardware and the encryption used is efficient in terms of memory and CPU usage.
- o might only be able to communicate with one single managing entity and cannot support the parallel access of many managing entities.
- o might depend on a self-configuration feature, i.e., the managing entity might not know all devices in a network and the device needs to be able to initiate connection setup for the device configuration.
- o might depend on self- or neighbor-monitoring feature, i.e., the managing entity might not be able to monitor all devices in a network continuously.
- o might only be able to communicate with its neighbors, i.e., the device should be able to get its configuration from a neighbor.

- o might only be able to support parsing of data models with limited size, i.e., the device data models need to be compact containing the most necessary data and if possible parsable as a stream.
- o might only be able to support a limited or no failure detection, i.e., the managing entity needs to handle the situation, where a failure does not get detected or gets detected late gracefully e.g., with asking repeatedly.
- o might only be able to support the reporting of just one or a limited set failure types.
- o might only be able to support a limited set of notifications, possible only an "I-am-alive" message.
- o might only be able to support a soft-reset from failure recovery.
- o might possibly generate a large amount of redundant reporting data, i.e., the intermediary management entity (see [RFC7252]) should be able to filter and aggregate redundant data.

A network of constrained devices:

- o might only support an unreliable (e.g. lossy) radio link, i.e., the client and server of a management protocol need to repeat commands as necessary or gracefully ignore incomplete commands.
- o might be necessary to manage based on multicast communication, i.e., the managing entity needs to be prepared to configure many devices at once based on the same data model.
- o might have a very large topology supporting 10,000 or more nodes for some applications and as such node naming is a specific issue for constrained networks.
- o needs to support self-organization, i.e., given the large number of nodes and their potential placement in hostile locations and frequently changing topology, manual configuration of nodes is typically not feasible. As such, the network would benefit from the ability to reconfigure itself so that it can continue to operate properly and support reliable connectivity.
- o might need a management solution that is energy-efficient, using as little wireless bandwidth as possible since communication is highly energy demanding.

- o needs to support localization schemes to determine the location of devices since the devices might be moving and location information is important for some applications.
- o needs a management solution that is scalable as the network may consist of thousands of nodes and may need to be extended continuously.
- o needs to provide fault tolerance. Faults in network operation including hardware and software errors or failures detected by the transport protocol should be handled smoothly. In such a case it should be possible to run the protocol possibly at a reduced level but avoiding to fail completely. E.g., self-monitoring mechanisms or graceful degradation of features can be used to provide fault tolerance.
- o might require new management capabilities: for example, network coverage information and a constrained device power-distribution-map.
- o might require a new management function for data management, since the type and amount of data collected in constrained networks is different from those of the traditional networks.
- o might also need energy-efficient key management.

1.7. Configuration and Monitoring Functionality Levels

Devices often differ significantly on the level of configuration management support they provide. This document classifies the configuration management functionality as follows:

CL0: Devices are pre-configured and allow no runtime configuration changes. Configuration parameters are often hard coded and compiled directly into the firmware image.

CL1: Devices have explicit configuration objects. However, changes require a restart of the device to take effect.

CL2: Devices allow management systems to replace the entire configuration (or pre-determined subsets) in bulk. Configuration changes take effect by soft-restarts of the system (or subsystems).

CL3: Devices allow management systems to modify configuration objects without bulk replacements and changes take effect immediately.

CL4: Devices support multiple configuration datastores and they might distinguish between the currently running and the next startup configuration.

CL5: Devices support configuration datastore locking and device-local configuration change transactions, i.e., either all configuration changes are applied or none of them.

CL6: Devices support configuration change transactions across devices.

This document defines a classification of devices with regards to different levels of monitoring support. In general a device may be in several of the levels listed below:

ML0: Devices push pre-defined monitoring data.

ML1: Devices allow management systems to pull pre-defined monitoring data.

ML2: Devices allow management systems to pull user-defined filtered subsets of monitoring data.

ML3: Devices are able to locally process monitoring data in order to detect threshold crossings or to aggregate data.

At the time of this writing, constrained devices often implement a combination of one of CL0-CL2 with one of ML0-ML1.

2. Problem Statement

The terminology for the "Internet of Things" is still nascent, and depending on the network type or layer in focus diverse technologies and terms are in use. Common to all these considerations is the "Things" or "Objects" are supposed to have physical or virtual identities using interfaces to communicate. In this context, we need to differentiate between the Constrained and Smart Devices identified by an IP address compared to virtual entities such as Smart Objects, which can be identified as a resource or a virtual object by using a unique identifier. Furthermore, the smart devices usually have a limited memory and CPU power as well as aim to be self-configuring and easy to deploy.

However, the constraints of the network nodes require a rethinking of the protocol characteristics concerning power consumption, performance, bandwidth consumption, memory, and CPU usage. As such, there is a demand for protocol simplification, energy-efficient communication, less CPU usage and smaller memory footprint.

On the application layer the IETF is already developing protocols like the Constrained Application Protocol (CoAP) [RFC7252] enabling the communication of constrained devices and networks e.g., for smart energy applications or home automation environments. The deployment of such an environment involves in fact many, in some scenarios up to million constrained devices (e.g., smart meters), which produce a large amount of data. This data needs to be collected, filtered, and pre-processed for further use in diverse services.

Considering the high number of nodes to deploy, one has to think about the manageability aspects of the smart devices and plan for easy deployment, configuration, and management of the networks of constrained devices as well as the devices themselves. Consequently, seamless monitoring and self-configuration of such network nodes becomes more and more imperative. Self-configuration and self-management is already a reality in the standards of some of the bodies such as 3GPP. To introduce self-configuration of smart devices successfully a device-initiated connection establishment is often required.

A simple and efficient application layer protocol, such as CoAP, is essential to address the issue of efficient object-to-object communication and information exchange. Such an information exchange should be done based on interoperable data models to enable the exchange and interpretation of diverse application and management related data.

In an ideal world, we would have only one network management protocol for monitoring, configuration, and exchanging management data, independently of the type of the network (e.g., Smart Grid, wireless access, or core network). Furthermore, it would be desirable to derive the basic data models for constrained devices from the core models used today to enable reuse of functionality and end-to-end information exchange. However, the current management protocols seem to be too heavyweight compared to the capabilities the constrained devices have and are not applicable directly for the use in a network of constrained devices. Furthermore, the data models addressing the requirements of such smart devices need yet to be designed.

The IETF so far has not developed any specific technologies for the management of constrained devices and the networks comprised by constrained devices. IP-based sensors or constrained devices in such an environment, i.e., devices with very limited memory and CPU resources, use today, e.g., application-layer protocols to do simple resource management and monitoring. This might be sufficient for some basic cases, however, there is a need to reconsider the network management mechanisms based on the new, changed, as well as reduced requirements coming from smart devices and the network of such

constrained devices. Albeit it is questionable whether we can take the same comprehensive approach we use in an IP network also for the management of constrained devices. Hence, the management of a network with constrained devices is necessary to design in a simplified and less complex manner.

As Section 1.6 highlights, there are diverse characteristics of constrained devices or networks, which stem from their constrainedness and therefore have an impact on the requirements for the management of such a network with constrained devices. The use cases discussed in [COM-USE] show that the requirements on constrained networks are manifold and need to be analyzed from different angles, e.g., concerning the design of the management architecture, the selection of the appropriate protocol features as well as the specific issues which are new in the context of constrained devices. Examples of such issues are e.g., the careful management of the scarce energy resources, the necessity for self-organization and self-management of such devices but also the implementation considerations to enable the use of common communication technologies on a constrained hardware in an efficient manner. For an exhaustive list of issues and requirements that need to be addressed for the management of a network with constrained devices please see Section 1.6 and Section 3.

3. Requirements on the Management of Networks with Constrained Devices

This section describes the requirements categorized by management areas listed in subsections.

Note that the requirements listed in this section have been separated from the context in which they may appear. This document in general does not recommend the realization of any subset of the described requirements. As such this document avoids selecting any of the requirements as mandatory to implement. A device might be able to provide only a particular selected set of requirements and might not be capable to provide all requirements in this document. On the other hand a device vendor might select a specific relevant subset of the requirements to implement.

The following template is used for the definition of the requirements.

Req-ID: An ID composed by two numbers: section number indicating the topic area and a unique three-digit number per section

Title: The title of the requirement.

Description: The rationale and description of the requirement.

Source: The origin of the requirement and the matching use case or application. For the discussion of referred use cases for constrained management please see [COM-USE].

Requirement Type: Functional Requirement, Non-Functional Requirement. A functional requirement is related to a function or component. As such functional requirements may be technical details, or specific functionality that define what a system is supposed to accomplish. Non-functional requirements (also known as design constraints or quality requirements) impose implementation-related considerations such as performance requirements, security, or reliability.

Device type: The device types by which this requirement can be supported: C0, C1 and/or C2.

Priority: The priority of the requirement showing its importance for a particular type of device: High, Medium, and Low. The priority of a requirement can be High e.g., for a C2 device but Low for a C1 or C0 device as the realization of complex features in a C1 device is in many cases not possible.

3.1. Management Architecture/System

Req-ID: 1.001

Title: Support multiple device classes within a single network.

Description: Larger networks usually consist of devices belonging to different device classes (e.g., constrained mesh endpoints and less constrained routers) communicating with each other. Hence, the management architecture must be applicable to networks that have a mix of different device classes. See Section 3. of [RFC7228] for the definition of Constrained Device Classes.

Source: All use cases.

Requirement Type: Non-Functional Requirement

Device type: C1 and/or C2

Priority: High

Req-ID: 1.002

Title: Management scalability.

Description: The management architecture must be able to scale with the number of devices involved and operate efficiently in any network size and topology. This implies that e.g., the managing entity is able to handle large amounts of device monitoring data and the management protocol is not sensitive to the decrease of the time between two client requests. To achieve good scalability, caching techniques, in-network data aggregation techniques, hierarchical management models may be used.

Source: General requirement for all use cases to enable large scale networks.

Requirement Type: Non-Functional Requirement

Device type: C0, C1, and C2

Priority: High

Req-ID: 1.003

Title: Hierarchical management

Description: Provide a means of hierarchical management, i.e., provide intermediary management entities on different levels, which can take over the responsibility for the management of a sub-hierarchy of the network of constraint devices. The intermediary management entity can e.g., support management data aggregation to handle e.g., high-frequent monitoring data or provide a caching mechanism for the uplink and downlink communication. Hierarchical management contributes to management scalability.

Source: Use cases where a large amount of devices are deployed with a hierarchical topology.

Requirement Type: Non-Functional Requirement

Device type: Managing and intermediary entities.

Priority: Medium

Req-ID: 1.004

Title: Minimize state maintained on constrained devices.

Description: The amount of state that needs to be maintained on constrained devices should be minimized. This is important in order to save memory (especially relevant for C0 and C1 devices) and in order to allow devices to restart for example to apply configuration changes or to recover from extended periods of inactivity.

Note: One way to achieve this is to adopt a RESTful architecture that minimizes the amount of state maintained by managed constrained devices and that makes resources of a device addressable via URIs.

Source: Basic requirement which concerns all use cases.

Requirement Type: Functional Requirement

Device type: C0, C1, and C2

Priority: High

Req-ID: 1.005

Title: Automatic re-synchronization with eventual consistency.

Description: To support large scale networks, where some constrained devices may be offline at any point in time, it is necessary to distribute configuration parameters in a way that allows temporary inconsistencies but eventually converges, after a sufficiently long period of time without further changes, towards global consistency.

Source: Use cases with large scale networks with many devices.

Requirement Type: Functional Requirement

Device type: C0, C1, and C2

Priority: High

Req-ID: 1.006

Title: Support for lossy links and unreachable devices

Description: Some constrained devices will only be able to support lossy and unreliable links characterized by a limited data rate, a high latency, and a high transmission error rate. Furthermore, constrained devices often duty cycle their radio or the whole device in order to save energy. Some classes of devices labeled as 'sleepy endpoints' set their network links to a disconnected state during long periods of time. In all cases the management system must not assume that constrained devices are always reachable.

Source: Basic requirement for networks of constrained devices with unreliable links and constrained devices that sleep to save energy.

Requirement Type: Non-Functional Requirement

Device type: C0, C1, and C2

Priority: High

Req-ID: 1.007

Title: Network-wide configuration

Description: Provide means by which the behavior of the network can be specified at a level of abstraction (network-wide configuration) higher than a set of configuration information specific to individual devices. It is useful to derive the device specific configuration from the network-wide configuration. Such a repository can be used to configure pre-defined device or protocol parameters for the whole network. Furthermore, such a network-wide view can be used to monitor and manage a group of routers or a whole network. E.g., monitoring the performance of a network requires additional information other than what can be acquired from a single router using a management protocol.

Note: The identification of the relevant subset of the policies to be provisioned is according to the capabilities of each device and can be obtained from a pre-configured data-repository.

Source: In general all use cases of network and device configuration based on a network view in a top-down manner.

Requirement Type: Non-Functional Requirement

Device type: C0, C1, and C2

Priority: Medium

Req-ID: 1.008

Title: Distributed management

Description: Provide a means of simple distributed management, where a network of constrained devices can be managed or monitored by more than one manager. Since the connectivity to a server cannot be guaranteed at all times, a distributed approach may provide a higher reliability, at the cost of increased complexity. This requirement implies the handling of data consistency in case of concurrent read and write access to the device datastore. It might also happen that no management (configuration) server is accessible and the only reachable node is a peer device. In this case the device should be able to obtain its configuration from peer devices.

Source: Use cases where the count of devices to manage is high.

Requirement Type: Non-Functional Requirement

Device type: C1 and C2

Priority: Medium

3.2. Management Protocols and Data Models

Req-ID: 2.001

Title: Modular implementation of management protocols

Description: Management protocols should be specified to allow for modular implementations, i.e., it should be possible to implement only a basic set of protocol primitives on highly constrained devices while devices with additional resources may provide more support for additional protocol primitives. See Section 1.7 for a discussion on the level of configuration management and monitoring support constrained devices may provide.

Source: Basic requirement interesting for all use cases.

Requirement Type: Non-Functional Requirement

Device type: C0, C1, and C2

Priority: High

Req-ID: 2.002

Title: Compact encoding of management data

Description: The encoding of management data should be compact and space efficient, enabling small message sizes.

Source: General requirement to save memory for the receiver buffer and on-air bandwidth.

Requirement Type: Functional Requirement

Device type: C0, C1, and C2

Priority: High

Req-ID: 2.003

Title: Compression of management data or complete messages

Description: Management data exchanges can be further optimized by applying data compression techniques or delta encoding techniques. Compression typically requires additional code size and some additional buffers and/or the maintenance of some additional state information. For C0 devices compression may not be feasible.

Source: Use cases where it is beneficial to reduce transmission time and bandwidth, e.g., mobile applications which require to save on-air bandwidth.

Requirement Type: Functional Requirement

Device type: C1 and C2

Priority: Medium

Req-ID: 2.004

Title: Mapping of management protocol interactions

Description: It is desirable to have a lossless automated mapping between the management protocol used to manage constrained devices and the management protocols used to manage regular devices. In the ideal case, the same core management protocol can be used with certain restrictions taking into account the resource limitations of constrained devices. However, for very resource constrained devices, this goal might not be achievable.

Source: Use cases where high-frequent interaction with the management system of a non-constrained network is required.

Requirement Type: Functional Requirement

Device type: C1 and C2

Priority: Medium

Req-ID: 2.005

Title: Consistency of data models with the underlying information model

Description: The data models used by the management protocol must be consistent with the information model used to define data models for non-constrained networks. This is essential to facilitate the integration of the management of constrained networks with the management of non-constrained networks. Using an underlying information model for future data model design enables furthermore top-down model design and model reuse as well as data interoperability (i.e., exchange of management information between the constrained and non-constrained networks). This is a strong requirement, even despite the fact that the underlying information models are often not explicitly documented in the IETF.

Source: General requirement to support data interoperability, consistency and model reuse.

Requirement Type: Non-Functional Requirement

Device type: C0, C1, and C2

Priority: High

Req-ID: 2.006

Title: Lossless mapping of management data models.

Description: It is desirable to have a lossless automated mapping between the management data models used to manage regular devices and the management data models used for managing constrained devices. In the ideal case, the same core data models can be used with certain restrictions taking into account the resource limitations of constrained devices. However, for very resource constrained devices, this goal might not be achievable.

Source: Use cases where consistent data exchange with the management system of a non-constrained network is required.

Requirement Type: Functional Requirement

Device type: C2

Priority: Medium

Req-ID: 2.007

Title: Protocol extensibility

Description: Provide means of extensibility for the management protocol, i.e., by adding new protocol messages or mechanisms that can deal with changing requirements on a supported message and data types effectively, without causing interoperability problems or having to replace/update large amount of deployed devices.

Source: Basic requirement useful for all use cases.

Requirement Type: Functional Requirement

Device type: C0, C1, and C2

Priority: High

3.3. Configuration Management

Req-ID: 3.001

Title: Self-configuration capability

Description: Automatic configuration and re-configuration of devices without manual intervention. Compared to the traditional management of devices where the management application is the

central entity configuring the devices, in the auto-configuration scenario the device is the active part and initiates the configuration process. Self-configuration can be initiated during the initial configuration or for subsequent configurations, where the configuration data needs to be refreshed. Self-configuration should be also supported during the initialization phase or in the event of failures, where prior knowledge of the network topology is not available or the topology of the network is uncertain.

Source: In general all use cases requiring easy deployment and plug&play behavior as well as easy maintenance of many constrained devices.

Requirement Type: Functional Requirement

Device type: C0, C1, and C2

Priority: High for device categories C0 and C1, Medium for C2.

Req-ID: 3.002

Title: Capability discovery

Description: Enable the discovery of supported optional management capabilities of a device and their exposure via at least one protocol and/or data model.

Source: Use cases where the device interaction with other devices or applications is a function of the level of support for its capabilities.

Requirement Type: Functional Requirement

Device type: C1 and C2

Priority: Medium

Req-ID: 3.003

Title: Asynchronous transaction support

Description: Provide configuration management with asynchronous (event-driven) transaction support. Configuration operations must

support a transactional model, with asynchronous indications that the transaction was completed.

Source: Use cases that require transaction-oriented processing because of reliability or distributed architecture functional requirements.

Requirement Type: Functional Requirement

Device type: C1 and C2

Priority: Medium

Req-ID: 3.004

Title: Network reconfiguration

Description: Provide a means of iterative network reconfiguration in order to recover the network from node and communication failures. The network reconfiguration can be failure-driven and self-initiated (automatic reconfiguration). The network reconfiguration can be also performed on the whole hierarchical structure of a network (network topology).

Source: Practically all use cases, as network connectivity is a basic requirement.

Requirement Type: Functional Requirement

Device type: C0, C1, and C2

Priority: Medium

3.4. Monitoring Functionality

Req-ID: 4.001

Title: Device status monitoring

Description: Provide a monitoring function to collect and expose information about device status and exposing it via at least one management interface. The device monitoring might make use of the hierarchical management through the intermediary entities and the caching mechanism. The device monitoring might also make use of neighbor-monitoring (fault detection in local network) to support fast fault detection and recovery, e.g., in a scenario where a

managing entity is unreachable and a neighbor can take over the monitoring responsibility.

Source: All use cases

Requirement Type: Functional Requirement

Device type: C0, C1, and C2

Priority: High, Medium for neighbor-monitoring.

Req-ID: 4.002

Title: Energy status monitoring

Description: Provide a monitoring function to collect and expose information about device energy parameters and usage (e.g., battery level and average power consumption).

Source: Use case Energy Management

Requirement Type: Functional Requirement

Device type: C0, C1, and C2

Priority: High for energy reporting devices, Low for others.

Req-ID: 4.003

Title: Monitoring of current and estimated device availability

Description: Provide a monitoring function to collect and expose information about current device availability (energy, memory, computing power, forwarding plane utilization, queue buffers, etc.) and estimation of remaining available resources.

Source: All use cases. Note that monitoring energy resources (like battery status) may be required on all kinds of devices.

Requirement Type: Functional Requirement

Device type: C0, C1, and C2

Priority: Medium

Req-ID: 4.004

Title: Network status monitoring

Description: Provide a monitoring function to collect, analyze and expose information related to the status of a network or network segments connected to the interface of the device.

Source: All use cases.

Requirement Type: Functional Requirement

Device type: C1 and C2

Priority: Low, based on the realization complexity.

Req-ID: 4.005

Title: Self-monitoring

Description: Provide self-monitoring (local fault detection) feature for fast fault detection and recovery.

Source: Use cases where the devices cannot be monitored centrally in appropriate manner, e.g., self-healing is required.

Requirement Type: Functional Requirement

Device type: C1 and C2

Priority: High for C2, Medium for C1

Req-ID: 4.006

Title: Performance monitoring

Description: The device will provide a monitoring function to collect and expose information about the basic performance parameter of the device. The performance management functionality might make use of the hierarchical management through the intermediary devices.

Source: Use cases Building automation, and Transport applications

Requirement Type: Functional Requirement

Device type: C1 and C2

Priority: Low

Req-ID: 4.007

Title: Fault detection monitoring

Description: The device will provide fault detection monitoring. The system collects information about network states in order to identify whether faults have occurred. In some cases the detection of the faults might be based on the processing and analysis of the parameters retrieved from the network or other devices. In case of C0 devices the monitoring might be limited to the check whether the device is alive or not.

Source: Use cases Environmental Monitoring, Building Automation, Energy Management, Infrastructure Monitoring

Requirement Type: Functional Requirement

Device type: C0, C1 and C2

Priority: Medium

Req-ID: 4.008

Title: Passive and reactive monitoring

Description: The device will provide passive and reactive monitoring capabilities. The system or manager collects information about device components and network states (passive monitoring) and may perform postmortem analysis of collected data. In case events of interest have occurred the system or manager can adaptively react (reactive monitoring), e.g., reconfigure the network. Typically actions (re-actions) will be executed or sent as commands by the management applications.

Source: Diverse use cases relevant for device status and network state monitoring

Requirement Type: Functional Requirement

Device type: C2

Priority: Medium

Req-ID: 4.009

Title: Recovery

Description: Provide local, central and hierarchical recovery mechanisms (recovery is in some cases achieved by recovering the whole network of constrained devices).

Source: Use cases Industrial applications, Home and Building Automation, Mobile Applications that involve different forms of clustering or area managers.

Requirement Type: Functional Requirement

Device type: C2

Priority: Medium

Req-ID: 4.010

Title: Network topology discovery

Description: Provide a network topology discovery capability (e.g., use of topology extraction algorithms to retrieve the network state) and a monitoring function to collect and expose information about the network topology.

Source: Use cases Community Network Applications and Mobile Applications

Requirement Type: Functional Requirement

Device type: C1 and C2

Priority: Low, based on the realization complexity.

Req-ID: 4.011

Title: Notifications

Description: The device will provide the capability of sending notifications on critical events and faults.

Source: All use cases.

Requirement Type: Functional Requirement

Device type: C0, C1, and C2

Priority: Medium for C2, Low for C0 and C1

Req-ID: 4.012

Title: Logging

Description: The device will provide the capability of building, keeping, and allowing retrieval of logs of events (including but not limited to critical faults and alarms).

Source: Use cases Industrial Applications, Building Automation, Infrastructure monitoring

Requirement Type: Functional Requirement

Device type: C2

Priority: High for some medical or industrial applications, Medium otherwise

3.5. Self-management

Req-ID: 5.001

Title: Self-management - Self-healing

Description: Enable event-driven and/or periodic self-management functionality in a device. The device should be able to react in case of a failure e.g., by initiating a fully or partly reset and initiate a self-configuration or management data update as necessary. A device might be further able to check for failures cyclically or schedule-controlled to trigger self-management as necessary. It is a matter of device design and subject for

discussion how much self-management a C1 device can support. A minimal failure detection and self-management logic is assumed to be generally useful for the self-healing of a device.

Source: The requirement generally relates to all use cases in this document.

Requirement Type: Functional Requirement

Device type: C1 and C2

Priority: High for C2, Medium for C1

3.6. Security and Access Control

Req-ID: 6.001

Title: Authentication of management system and devices.

Description: Systems having a management role must be properly authenticated to the device such that the device can exercise proper access control and in particular distinguish rightful management systems from rogue systems. On the other hand managed devices must authenticate themselves to systems having a management role such that management systems can protect themselves from rogue devices. In certain application scenarios, it is possible that a large number of devices need to be (re)started at about the same time. Protocols and authentication systems should be designed such that a large number of devices (re)starting simultaneously does not negatively impact the device authentication process.

Source: Basic security requirement for all use cases.

Requirement Type: Functional Requirement

Device type: C0, C1, and C2

Priority: High, Medium for the (re)start of a large number of devices

Req-ID: 6.002

Title: Support suitable security bootstrapping mechanisms

Description: Mechanisms should be supported that simplify the bootstrapping of device that is the discovery of newly deployed devices in order to provide them with appropriate access control permissions.

Source: Basic security requirement for all use cases.

Requirement Type: Functional Requirement

Device type: C0, C1, and C2

Priority: High

Req-ID: 6.003

Title: Access control on management system and devices

Description: Systems acting in a management role must provide an access control mechanism that allows the security administrator to restrict which devices can access the managing system (e.g., using an access control white list of known devices). On the other hand managed constrained devices must provide an access control mechanism that allows the security administrator to restrict how systems in a management role can access the device (e.g., no-access, read-only access, and read-write access).

Source: Basic security requirement for use cases where access control is essential.

Requirement Type: Functional Requirement

Device type: C0, C1, and C2

Priority: High

Req-ID: 6.004

Title: Select cryptographic algorithms that are efficient in both code space and execution time.

Description: Cryptographic algorithms have a major impact in terms of both code size and overall execution time. It is therefore necessary to select mandatory to implement cryptographic algorithms that are reasonable to implement with the available

code space and that have a small impact at runtime. Furthermore some wireless technologies (e.g., IEEE 802.15.4) require the support of certain cryptographic algorithms. It might be useful to choose algorithms that are likely to be supported in wireless chipsets for certain wireless technologies.

Source: Generic requirement to reduce the footprint and CPU usage of a constrained device.

Requirement Type: Non-Functional Requirement

Device type: C0, C1, and C2

Priority: High, Medium for hardware-supported algorithms.

3.7. Energy Management

Req-ID: 7.001

Title: Management of energy resources

Description: Enable managing power resources in the network, e.g., reduce the sampling rate of nodes with critical battery and reduce node transmission power, put nodes to sleep, put single interfaces to sleep, reject a management job based on available energy, criteria e.g., importance levels pre-defined by the management application, etc. (e.g., a task marked as essential can be executed even if the energy level is low). The device may further implement standard data models for energy management and expose it through a management protocol interface, e.g., EMAN MIB modules and extensions (work ongoing). It might be necessary to use a subset of EMAN MIBs for C1 and C2 devices.

Source: Use case Energy Management

Requirement Type: Functional Requirement

Device type: C0, C1, and C2

Priority: Medium for the use case Energy Management, Low otherwise.

Req-ID: 7.002

Title: Support of energy-optimized communication protocols

Description: Use of an optimized communication protocol to minimize energy usage for the device (radio) receiver/transmitter, on-air bandwidth (protocol efficiency), reduced amount of data communication between nodes (implies data aggregation and filtering but also a compact format for the transferred data).

Source: Use cases Energy Management and Mobile Applications.

Requirement Type: Non-Functional Requirement

Device type: C2

Priority: Medium

Req-ID: 7.003

Title: Support for layer 2 energy-aware protocols

Description: The device will support layer 2 energy management protocols (e.g., energy-efficient Ethernet IEEE 802.3az) and be able to report on these.

Source: Use case Energy Management

Requirement Type: Non-Functional Requirement

Device type: C0, C1, and C2

Priority: Medium

Req-ID: 7.004

Title: Dying gasp

Description: When energy resources draw below the red line level, the device will send a dying gasp notification and perform if still possible a graceful shutdown including conservation of critical device configuration and status information.

Source: Use case Energy Management

Requirement Type: Functional Requirement

Device type: C0, C1, and C2

Priority: Medium

3.8. Software Distribution

Req-ID: 8.001

Title: Group-based provisioning

Description: Support group-based provisioning, i.e., firmware update and configuration management, of a large set of constrained devices with eventual consistency and coordinated reload times. The device should accept group-based configuration management based on bulk commands, which aim similar configurations of a large set of constrained devices of the same type in a given group, and which may share a common data model. Activation of configuration may be based on pre-loaded sets of default values.

Source: All use cases

Requirement Type: Non-Functional Requirement

Device type: C0, C1, and C2

Priority: Medium

3.9. Traffic Management

Req-ID: 9.001

Title: Congestion avoidance

Description: Support congestion control principles as defined in [RFC2914], e.g., the ability to avoid congestion by modifying the device's reporting rate for periodical data (which is usually redundant) based on the importance and reliability level of the management data. This functionality is usually controlled by the managing entity, where the managing entity marks the data as important or relevant for reliability. However, reducing a device's reporting rate can also be initiated by a device if it is able to detect congestion or has insufficient buffer memory.

Source: Use cases with high reporting rate and traffic e.g., AMI or M2M.

Requirement Type: Non-Functional Requirement

Device type: C1 and C2

Priority: Medium

Req-ID: 9.002

Title: Reroute traffic

Description: Provide the ability for network nodes to redirect traffic from overloaded intermediary nodes in a network to another path in order to prevent congestion on a central server and in the primary network.

Source: Use cases with high reporting rate and traffic e.g., AMI or M2M.

Requirement Type: Non-Functional Requirement

Device type: Intermediary entity in the network.

Priority: Medium

Req-ID: 9.003

Title: Traffic Shaping.

Description: Provide the ability to apply traffic shaping policies to incoming and outgoing links on an overloaded intermediary node as necessary in order to reduce the amount of traffic in the network.

Source: Use cases with high reporting rate and traffic e.g., AMI or M2M.

Requirement Type: Non-Functional Requirement

Device type: Intermediary entity in the network.

Priority: Medium

3.10. Transport Layer

Req-ID: 10.001

Title: Scalable transport layer

Description: Enable the use of a scalable transport layer, i.e., not sensitive to a high rate of incoming client requests, which is useful for applications requiring frequent access to device data.

Source: Applications with high frequent access to the device data.

Requirement Type: Non-Functional Requirement

Device type: C0, C1 and C2

Priority: Medium

Req-ID: 10.002

Title: Reliable unicast transport of messages

Description: Diverse applications need a reliable transport of messages. The reliability might be achieved based on a transport protocol such as TCP or can be supported based on message repetition if an acknowledgment is missing.

Source: Generally applications benefit from the reliability of the message transport.

Requirement Type: Functional Requirement

Device type: C0, C1, and C2

Priority: High

Req-ID: 10.003

Title: Best-effort multicast

Description: Provide best-effort multicast of messages, which is generally useful when devices need to discover a service provided by a server or many devices need to be configured by a managing entity at once based on the same data model.

Source: Use cases where a device needs to discover services as well as use cases with high amount of devices to manage, which are hierarchically deployed, e.g., AMI or M2M.

Requirement Type: Functional Requirement

Device type: C0, C1, and C2

Priority: Medium

Req-ID: 10.004

Title: Secure message transport

Description: Enable secure message transport providing authentication, data integrity, confidentiality by using existing transport layer technologies with small footprint such as TLS/DTLS.

Source: All use cases.

Requirement Type: Non-Functional Requirements

Device type: C1 and C2

Priority: High

3.11. Implementation Requirements

Req-ID: 11.001

Title: Avoid complex application layer transactions requiring large application layer messages.

Description: Complex application layer transactions tend to require large memory buffers that are typically not available on C0 or C1 devices and only by limiting functionality on C2 devices. Furthermore, the failure of a single large transaction requires repeating the whole transaction. On constrained devices, it is often more desirable to split a large transaction into a sequence of smaller transactions that require less resources and allow to make progress using a sequence of smaller steps.

Source: Basic requirement which concerns all use cases with memory constrained devices.

Requirement Type: Non-Functional Requirement

Device type: C0, C1, and C2

Priority: High

Req-ID: 11.002

Title: Avoid reassembly of messages at multiple layers in the protocol stack.

Description: Reassembly of messages at multiple layers in the protocol stack requires buffers at multiple layers, which leads to inefficient use of memory resources. This can be avoided by making sure the application layer, the security layer, the transport layer, the IPv6 layer and any adaptation layers are aware of the limitations of each other such that unnecessary fragmentation and reassembly can be avoided. In addition, message size constraints must be announced to protocol peers such that they can adapt and avoid sending messages that can't be processed due to resource constraints on the receiving device.

Source: Basic requirement which concerns all use cases with memory constrained devices.

Requirement Type: Non-Functional Requirement

Device type: C0, C1, and C2

Priority: High

4. IANA Considerations

This document does not introduce any new code-points or namespaces for registration with IANA.

Note to RFC Editor: this section may be removed on publication as an RFC.

5. Security Considerations

This document discusses the problem statement and requirements on networks of constrained devices. Section 1.6 mentions a number of limitations that could prevent the implementation of strong cryptographic algorithms. Requirements for security and access control are listed in Section 3.6.

Constrained devices might be deployed often in unsafe environments, where attackers can gain physical access to the devices. As a consequence, it is crucial that devices are robust and tamper resistant, have no backdoors, do not provide services that are not essential for the primary function, and properly protect any security

credentials that may be stored on the device (e.g., by using hardware protection mechanisms). Furthermore, it is important that any credentials leaking from a single device do not simplify the attack on other (similar) devices. In particular, security credentials should never be shared.

Since constrained devices often have limited computational resources, care should be taken in choosing efficient but cryptographically strong cryptographic algorithms. Designers of constrained devices that have a long expected lifetime need to ensure that cryptographic algorithms can be updated once devices have been deployed. The ability to perform secure firmware and software updates is an important management requirement.

Constrained devices might also generate sensitive data or require the processing of sensitive data. It is therefore an important requirement to properly protect access to the data in order to protect the privacy of humans using Internet-enabled devices. For certain types of data, protection during the transmission over the network may not be sufficient and methods should be investigated that provide protection of data while it is cached or stored (e.g., when using a store-and-forward transport mechanism).

6. Acknowledgments

Following persons reviewed and provided valuable comments to different versions of this document:

Dominique Barthel, Andy Bierman, Carsten Bormann, Zhen Cao, Benoit Claise, Hui Deng, Bert Greevenbosch, Joel M. Halpern, Ulrich Herberg, James Nguyen, Anuj Sehgal, Zach Shelby, Peter van der Stok, Thomas Watteyne, and Bert Wijnen.

The editors would like to thank the reviewers and the participants on the Coman and OPSAWG mailing lists for their valuable contributions and comments.

7. Informative References

- [RFC2914] Floyd, S., "Congestion Control Principles", BCP 41, RFC 2914, September 2000.
- [RFC2501] Corson, M. and J. Macker, "Mobile Ad hoc Networking (MANET): Routing Protocol Performance Issues and Evaluation Considerations", RFC 2501, January 1999.
- [RFC6632] Ersue, M. and B. Claise, "An Overview of the IETF Network Management Standards", RFC 6632, June 2012.

- [RFC7102] Vasseur, JP., "Terms Used in Routing for Low-Power and Lossy Networks", RFC 7102, January 2014.
- [RFC7228] Bormann, C., Ersue, M., and A. Keranen, "Terminology for Constrained-Node Networks", RFC 7228, May 2014.
- [RFC7252] Shelby, Z., Hartke, K., and C. Bormann, "The Constrained Application Protocol (CoAP)", RFC 7252, June 2014.
- [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, August 2007.
- [RFC6550] Winter, T., Thubert, P., 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, March 2012.
- [COM-USE] Ersue, M., Romascanu, D., and J. Schoenwaelder, "Constrained Management: Use Cases", draft-ietf-opsawg-coman-use-cases (work in progress), July 2014.

Appendix A. Change Log

- A.1. draft-ietf-opsawg-coman-probstate-reqs-04 - draft-ietf-opsawg-coman-probstate-reqs-05
 - o Extended Abstract and Overview sections to clarify the type of requirements the draft describes.
 - o Extended security highlighting the devices should make sure credentials are properly protected.
- A.2. draft-ietf-opsawg-coman-probstate-reqs-03 - draft-ietf-opsawg-coman-probstate-reqs-04
 - o Changed in section 1.3 "10⁻⁰" to "1".
 - o Clarified in section 3 how the Requirements ID is composed.
- A.3. draft-ietf-opsawg-coman-probstate-reqs-02 - draft-ietf-opsawg-coman-probstate-reqs-03
 - o General bug fixing.
 - o Stated in the abstract and introduction section that the requirements listed in the document are potential requirements.

- o Added text in section 1.3 to highlight that with the usage of 6LoWPAN and RPL multi-hop connectivity and dynamic routing can be achieved.
- A.4. draft-ietf-opsawg-coman-probstate-reqs-01 - draft-ietf-opsawg-coman-probstate-reqs-02
- o General bug fixing.
 - o Resolved the use of the term profile of requirements.
 - o Changed requirement title from Redirect traffic to Reroute traffic and the description accordingly.
 - o Changed requirement title from Traffic delay schemes to Traffic Shaping and the description accordingly.
 - o Extended Security Considerations section.
 - o Deleted empty section on Normative References.
- A.5. draft-ietf-opsawg-coman-probstate-reqs-00 - draft-ietf-opsawg-coman-probstate-reqs-01
- o General bug fixing.
 - o Added Section 1.7. on Configuration and Monitoring Functionality Levels.
 - o Changed diverse occurrences of "networks" to "networks with/of constrained devices".
 - o Introduced the term "Self-configuring infrastructureless networks" instead of MANET as it is a superset.
 - o Introduced the term 'sleepy endpoints'.
 - o Changed requirement IDs to be independent of section number.
 - o Introduced notes for parts of the requirements text if it is focusing on implementation or solution.
 - o Extended Security Considerations section.
 - o Deleted Appendix A and B on other SDO's work and related projects as they provided dynamic information and couldn't be kept up-to-date.

- A.6. draft-ersue-constrained-mgmt-03 - draft-ietf-opsawg-coman-probstate-reqs-00
- o Reduced the terminology section for terminology addressed in the LWIG terminology draft. Referenced the LWIG terminology draft.
 - o Checked and aligned all terminology against the LWIG terminology draft.
 - o Moved section 1.4. Constrained Device Deployment Options and section 3. Use Cases to the companion document [COM-USE].
 - o Renamed Section 1.3. Class of Networks in Focus to "Network Types in Focus" and removed abbreviations C0, C1 and C2 for network classes as they have not been used.
 - o Changed requirement priority classes to be High, Medium and Low.
 - o Changed requirement types to be Functional and Non-Functional and added text to explain the requirement types.
 - o Reformulation of some text parts for more clarity.
- A.7. draft-ersue-constrained-mgmt-02-03
- o Extended the terminology section and removed some of the terminology addressed in the new LWIG terminology draft. Referenced the LWIG terminology draft.
 - o Moved Section 1.3. on Constrained Device Classes to the new LWIG terminology draft.
 - o Class of networks considering the different type of radio and communication technologies in use and dimensions extended.
 - o Extended the Problem Statement in Section 2. following the requirements listed in Section 4.
 - o Following requirements, which belong together and can be realized with similar or same kind of solutions, have been merged.
 - * Distributed Management and Peer Configuration,
 - * Device status monitoring and Neighbor-monitoring,
 - * Passive Monitoring and Reactive Monitoring,

- * Event-driven self-management - Self-healing and Periodic self-management,
 - * Authentication of management systems and Authentication of managed devices,
 - * Access control on devices and Access control on management systems,
 - * Management of Energy Resources and Data models for energy management,
 - * Software distribution (group-based firmware update) and Group-based provisioning.
- o Deleted the empty section on the gaps in network management standards, as it will be written in a separate draft.
 - o Added links to mentioned external pages.
 - o Added text on OMA M2M Device Classification in appendix.

A.8. draft-ersue-constrained-mgmt-01-02

- o Extended the terminology section.
- o Added additional text for the use cases concerning deployment type, network topology in use, network size, network capabilities, radio technology, etc.
- o Added examples for device classes in a use case.
- o Added additional text provided by Cao Zhen (China Mobile) for Mobile Applications and by Peter van der Stok for Building Automation.
- o Added the new use cases 'Advanced Metering Infrastructure' and 'MANET Concept of Operations in Military'.
- o Added the section 'Managing the Constrainedness of a Device or Network' discussing the needs of very constrained devices.
- o Added a note that the requirements in Section 3 need to be seen as standalone requirements and the current document does not recommend any profile of requirements.
- o Added Section 3 on the detailed requirements on constrained management matched to management tasks like fault, monitoring,

configuration management, Security and Access Control, Energy Management, etc.

- o Solved nits and added references.
- o Added Appendix A on the related development in other bodies.
- o Added Appendix B on the work in related research projects.

A.9. draft-ersue-constrained-mgmt-00-01

- o Splitted the section on 'Networks of Constrained Devices' into the sections 'Network Topology Options' and 'Management Topology Options'.
- o Added the use case 'Community Network Applications' and 'Mobile Applications'.
- o Provided a Contributors section.
- o Extended the section on 'Medical Applications'.
- o Solved nits and added references.

Authors' Addresses

Mehmet Ersue (editor)
Nokia Networks

Email: mehmet.ersue@nsn.com

Dan Romascanu
Avaya

Email: dromasca@avaya.com

Juergen Schoenwaelder
Jacobs University Bremen

Email: j.schoenwaelder@jacobs-university.de

Ulrich Herberg

Email: ulrich@herberg.name

Internet Engineering Task Force
Internet-Draft
Intended status: Informational
Expires: September 2, 2015

M. Ersue, Ed.
Nokia Networks
D. Romascanu
Avaya
J. Schoenwaelder
A. Sehgal
Jacobs University Bremen
March 1, 2015

Management of Networks with Constrained Devices: Use Cases
draft-ietf-opsawg-coman-use-cases-05

Abstract

This document discusses use cases concerning the management of networks, where constrained devices are involved. A problem statement, deployment options and the requirements on the networks with constrained devices can be found in the companion document on "Management of Networks with Constrained Devices: Problem Statement and Requirements".

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 2, 2015.

Copyright Notice

Copyright (c) 2015 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents

carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	3
2. Access Technologies	4
2.1. Constrained Access Technologies	4
2.2. Cellular Access Technologies	5
3. Device Lifecycle	6
3.1. Manufacturing and Initial Testing	6
3.2. Installation and Configuration	6
3.3. Operation and Maintenance	7
3.4. Recommissioning and Decommissioning	7
4. Use Cases	8
4.1. Environmental Monitoring	8
4.2. Infrastructure Monitoring	9
4.3. Industrial Applications	10
4.4. Energy Management	12
4.5. Medical Applications	14
4.6. Building Automation	15
4.7. Home Automation	17
4.8. Transport Applications	18
4.9. Community Network Applications	20
4.10. Field Operations	22
5. IANA Considerations	23
6. Security Considerations	24
7. Contributors	24
8. Acknowledgments	24
9. Informative References	24
Appendix A. Change Log	25
A.1. draft-ietf-opsawg-coman-use-cases-04 - draft-ietf-opsawg-coman-use-cases-05	25
A.2. draft-ietf-opsawg-coman-use-cases-03 - draft-ietf-opsawg-coman-use-cases-04	26
A.3. draft-ietf-opsawg-coman-use-cases-02 - draft-ietf-opsawg-coman-use-cases-03	26
A.4. draft-ietf-opsawg-coman-use-cases-01 - draft-ietf-opsawg-coman-use-cases-02	26
A.5. draft-ietf-opsawg-coman-use-cases-00 - draft-ietf-opsawg-coman-use-cases-01	28
A.6. draft-ersue-constrained-mgmt-03 - draft-ersue-opsawg-coman-use-cases-00	28
A.7. draft-ersue-constrained-mgmt-02-03	28
A.8. draft-ersue-constrained-mgmt-01-02	29

A.9. draft-ersue-constrained-mgmt-00-01	30
Authors' Addresses	30

1. Introduction

Small devices with limited CPU, memory, and power resources, so called constrained devices (aka. sensor, smart object, or smart device) can be connected to a network. Such a network of constrained devices itself may be constrained or challenged, e.g., with unreliable or lossy channels, wireless technologies with limited bandwidth and a dynamic topology, needing the service of a gateway or proxy to connect to the Internet. In other scenarios, the constrained devices can be connected to a non-constrained network using off-the-shelf protocol stacks. Constrained devices might be in charge of gathering information in diverse settings including natural ecosystems, buildings, and factories and send the information to one or more server stations.

Network management is characterized by monitoring network status, detecting faults, and inferring their causes, setting network parameters, and carrying out actions to remove faults, maintain normal operation, and improve network efficiency and application performance. The traditional network management application periodically collects information from a set of elements that are needed to manage, processes the data, and presents them to the network management users. Constrained devices, however, often have limited power, low transmission range, and might be unreliable. Such unreliability might arise from device itself (e.g., battery exhausted) or from the channel being constrained (i.e., low-capacity and high-latency). They might also need to work in hostile environments with advanced security requirements or need to be used in harsh environments for a long time without supervision. Due to such constraints, the management of a network with constrained devices offers different type of challenges compared to the management of a traditional IP network.

This document aims to understand use cases for the management of a network, where constrained devices are involved. The document lists and discusses diverse use cases for the management from the network as well as from the application point of view. The list of discussed use cases is not an exhaustive one since other scenarios, currently unknown to the authors, are possible. The application scenarios discussed aim to show where networks of constrained devices are expected to be deployed. For each application scenario, we first briefly describe the characteristics followed by a discussion on how network management can be provided, who is likely going to be responsible for it, and on which time-scale management operations are likely to be carried out.

A problem statement, deployment and management topology options as well as the requirements on the networks with constrained devices can be found in the companion document [COM-REQ].

This documents builds on the terminology defined in [RFC7228] and [COM-REQ]. [RFC7228] is a base document for the terminology concerning constrained devices and constrained networks. Some use cases specific to IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs) can be found in [RFC6568].

2. Access Technologies

Besides the management requirements imposed by the different use cases, the access technologies used by constrained devices can impose restrictions and requirements upon the Network Management System (NMS) and protocol of choice.

It is possible that some networks of constrained devices might utilize traditional non-constrained access technologies for network access, e.g., local area networks with plenty of capacity. In such scenarios, the constrainedness of the device presents special management restrictions and requirements rather than the access technology utilized.

However, in other situations constrained or cellular access technologies might be used for network access, thereby causing management restrictions and requirements to arise as a result of the underlying access technologies.

A discussion regarding the impact of cellular and constrained access technologies is provided in this section since they impose some special requirements on the management of constrained networks. On the other hand, fixed line networks (e.g., power line communications) are not discussed here since tend to be quite static and do not typically impose any special requirements on the management of the network.

2.1. Constrained Access Technologies

Due to resource restrictions, embedded devices deployed as sensors and actuators in the various use cases utilize low-power low data-rate wireless access technologies such as IEEE 802.15.4, DECT ULE or Bluetooth Low-Energy (BT-LE) for network connectivity.

In such scenarios, it is important for the NMS to be aware of the restrictions imposed by these access technologies to efficiently manage these constrained devices. Specifically, such low-power low data-rate access technologies typically have small frame sizes. So

it would be important for the NMS and management protocol of choice to craft packets in a way that avoids fragmentation and reassembly of packets since this can use valuable memory on constrained devices.

Devices using such access technologies might operate via a gateway that translates between these access technologies and more traditional Internet protocols. A hierarchical approach to device management in such a situation might be useful, wherein the gateway device is in-charge of devices connected to it, while the NMS conducts management operations only to the gateway.

2.2. Cellular Access Technologies

Machine to machine (M2M) services are increasingly provided by mobile service providers as numerous devices, home appliances, utility meters, cars, video surveillance cameras, and health monitors, are connected with mobile broadband technologies. Different applications, e.g., in a home appliance or in-car network, use Bluetooth, Wi-Fi or ZigBee locally and connect to a cellular module acting as a gateway between the constrained environment and the mobile cellular network.

Such a gateway might provide different options for the connectivity of mobile networks and constrained devices:

- o a smart phone with 3G/4G and WLAN radio might use BT-LE to connect to the devices in a home area network,
- o a femtocell might be combined with home gateway functionality acting as a low-power cellular base station connecting smart devices to the application server of a mobile service provider,
- o an embedded cellular module with LTE radio connecting the devices in the car network with the server running the telematics service,
- o an M2M gateway connected to the mobile operator network supporting diverse IoT connectivity technologies including ZigBee and CoAP over 6LoWPAN over IEEE 802.15.4.

Common to all scenarios above is that they are embedded in a service and connected to a network provided by a mobile service provider. Usually there is a hierarchical deployment and management topology in place where different parts of the network are managed by different management entities and the count of devices to manage is high (e.g. many thousands). In general, the network is comprised by manifold type and size of devices matching to different device classes. As such, the managing entity needs to be prepared to manage devices with diverse capabilities using different communication or management

protocols. In case the devices are directly connected to a gateway they most likely are managed by a management entity integrated with the gateway, which itself is part of the Network Management System (NMS) run by the mobile operator. Smart phones or embedded modules connected to a gateway might be themselves in charge to manage the devices on their level. The initial and subsequent configuration of such a device is mainly based on self-configuration and is triggered by the device itself.

The gateway might be in charge of filtering and aggregating the data received from the device as the information sent by the device might be mostly redundant.

3. Device Lifecycle

Since constrained devices deployed in a network might go through multiple phases in their lifetime, it is possible for different managers of networks and/or devices to exist during different parts of the device lifetimes. An in-depth discussion regarding the possible device lifecycles can be found in [IOT-SEC].

3.1. Manufacturing and Initial Testing

Typically, the lifecycle of a device begins at the manufacturing stage. During this phase the manufacturer of the device is responsible for the management and configuration of the devices. It is also possible that a certain use case might utilize multiple types of constrained devices (e.g., temperature sensors, lighting controllers, etc.) and these could be manufactured by different entities. As such, during the manufacturing stage different managers can exist for different devices. Similarly, during the initial testing phase, where device quality assurance tasks might be performed, the manufacturer remains responsible for the management of devices and networks that might comprise them.

3.2. Installation and Configuration

The responsibility of managing the devices must be transferred to the installer during the installation phase. There must exist procedures for transferring management responsibility between the manufacturer and installer. The installer may be the customer or an intermediary contracted to setup the devices and their networks. It is important that the NMS utilized allows devices originating at different vendors to be managed, ensuring interoperability between them and the configuration of trust relationships between them as well.

It is possible that the installation and configuration responsibilities might lie with different entities. For example, the

installer of a device might only be responsible for cabling a network, physically installing the devices and ensuring initial network connectivity between them (e.g., configuring IP addresses). Following such an installation, the customer or a sub-contractor might actually configure the operation of the device. As such, during installation and configuration multiple parties might be responsible for managing a device and appropriate methods must be available to ensure that this management responsibility is transferred suitably.

3.3. Operation and Maintenance

At the outset of the operation phase, the operational responsibility of a device and network should be passed on to the customer. It is possible that the customer, however, might contract the maintenance of the devices and network to a sub-contractor. In this case, the NMS and management protocol should allow for configuring different levels of access to the devices. Since different maintenance vendors might be used for devices that perform different functions (e.g., HVAC, lighting, etc.) it should also be possible to restrict management access to devices based on the currently responsible manager.

3.4. Recommissioning and Decommissioning

The owner of a device might choose to replace, repurpose or even decommission it. In each of these cases, either the customer or the contracted maintenance agency must ensure that appropriate steps are taken to meet the end goal.

In case the devices needs to be replaced, the manager of the network (customer or contractor responsible) must detach the device from the network, remove all appropriate configuration and discard the device. A new device must then be configured to replace it. The NMS should allow for transferring configuration from and replacing an existing device. The management responsibility of the operation/maintenance manager would end once the device is removed from the network. During the installation of the new replacement device, the same responsibilities would apply as those during the Installation and Configuration phases.

The device being replaced may not have yet reached end-of-life, and as such, instead of being discarded it may be installed in a new location. In this case, the management responsibilities are once again resting in the hands of the entities responsible for the Installation and Configuration phases at the new location.

If a device is repurposed, then it is possible that the management responsibility for this device changes as well. For example, a device might be moved from one building to another. In this case, the managers responsible for devices and networks in each building could be different. As such, the NMS must not only allow for changing configuration but also transferring management responsibilities.

In case a device is decommissioned, the management responsibility typically ends at that point.

4. Use Cases

4.1. Environmental Monitoring

Environmental monitoring applications are characterized by the deployment of a number of sensors to monitor emissions, water quality, or even the movements and habits of wildlife. Other applications in this category include earthquake or tsunami early-warning systems. The sensors often span a large geographic area, they can be mobile, and they are often difficult to replace. Furthermore, the sensors are usually not protected against tampering.

Management of environmental monitoring applications is largely concerned with the monitoring whether the system is still functional and the roll-out of new constrained devices in case the system loses too much of its structure. The constrained devices themselves need to be able to establish connectivity (auto-configuration) and they need to be able to deal with events such as losing neighbors or being moved to other locations.

Management responsibility typically rests with the organization running the environmental monitoring application. Since these monitoring applications must be designed to tolerate a number of failures, the time scale for detecting and recording failures is for some of these applications likely measured in hours and repairs might easily take days. In fact, in some scenarios it might be more cost- and time-effective to not repair such devices at all. However, for certain environmental monitoring applications, much tighter time scales may exist and might be enforced by regulations (e.g., monitoring of nuclear radiation).

Since many applications of environmental monitoring sensors are likely to be in areas that are important to safety (flood monitoring, nuclear radiation monitoring, etc.) it is important for management protocols and network management systems (NMS) to ensure appropriate security protections. These protections include not only access control, integrity and availability of data, but also provide

appropriate mechanisms that can deal with situations that might be categorized as emergencies or when tampering with sensors/data might be detected.

4.2. Infrastructure Monitoring

Infrastructure monitoring is concerned with the monitoring of infrastructures such as bridges, railway tracks, or (offshore) windmills. The primary goal is usually to detect any events or changes of the structural conditions that can impact the risk and safety of the infrastructure being monitored. Another secondary goal is to schedule repair and maintenance activities in a cost effective manner.

The infrastructure to monitor might be in a factory or spread over a wider area but difficult to access. As such, the network in use might be based on a combination of fixed and wireless technologies, which use robust networking equipment and support reliable communication via application layer transactions. It is likely that constrained devices in such a network are mainly C2 devices [RFC7228] and have to be controlled centrally by an application running on a server. In case such a distributed network is widely spread, the wireless devices might use diverse long-distance wireless technologies such as WiMAX, or 3G/LTE. In cases, where an in-building network is involved, the network can be based on Ethernet or wireless technologies suitable for in-building usage.

The management of infrastructure monitoring applications is primarily concerned with the monitoring of the functioning of the system. Infrastructure monitoring devices are typically rolled out and installed by dedicated experts and changes are rare since the infrastructure itself changes rarely. However, monitoring devices are often deployed in unsupervised environments and hence special attention must be given to protecting the devices from being modified.

Management responsibility typically rests with the organization owning the infrastructure or responsible for its operation. The time scale for detecting and recording failures is likely measured in hours and repairs might easily take days. However, certain events (e.g., natural disasters) may require that status information be obtained much more quickly and that replacements of failed sensors can be rolled out quickly (or redundant sensors are activated quickly). In case the devices are difficult to access, a self-healing feature on the device might become necessary. Since infrastructure monitoring is closely related to ensuring safety, management protocols and systems must provide appropriate security

protections to ensure confidentiality, integrity and availability of data.

4.3. Industrial Applications

Industrial Applications and smart manufacturing refer to tasks such as networked control and monitoring of manufacturing equipment, asset and situation management, or manufacturing process control. For the management of a factory it is becoming essential to implement smart capabilities. From an engineering standpoint, industrial applications are intelligent systems enabling rapid manufacturing of new products, dynamic response to product demands, and real-time optimization of manufacturing production and supply chain networks. Potential industrial applications (e.g., for smart factories and smart manufacturing) are:

- o Digital control systems with embedded, automated process controls, operator tools, as well as service information systems optimizing plant operations and safety.
- o Asset management using predictive maintenance tools, statistical evaluation, and measurements maximizing plant reliability.
- o Smart sensors detecting anomalies to avoid abnormal or catastrophic events.
- o Smart systems integrated within the industrial energy management system and externally with the smart grid enabling real-time energy optimization.

Management of Industrial Applications and smart manufacturing may in some situations involve Building Automation tasks such as control of energy, HVAC (heating, ventilation, and air conditioning), lighting, or access control. Interacting with management systems from other application areas might be important in some cases (e.g., environmental monitoring for electric energy production, energy management for dynamically scaling manufacturing, vehicular networks for mobile asset tracking). Management of constrained devices and networks may not only refer to the management of their network connectivity. Since the capabilities of constrained devices are limited, it is quite possible that a management system would even be required to configure, monitor and operate the primary functions that a constrained device is utilized for, besides managing its network connectivity.

Sensor networks are an essential technology used for smart manufacturing. Measurements, automated controls, plant optimization, health and safety management, and other functions are provided by a

large number of networked sectors. Data interoperability and seamless exchange of product, process, and project data are enabled through interoperable data systems used by collaborating divisions or business systems. Intelligent automation and learning systems are vital to smart manufacturing but must be effectively integrated with the decision environment. The NMS utilized must ensure timely delivery of sensor data to the control unit so it may take appropriate decisions. Similarly, relaying of commands must also be monitored and managed to ensure optimal functioning. Wireless sensor networks (WSN) have been developed for machinery Condition-based Maintenance (CBM) as they offer significant cost savings and enable new functionalities. Inaccessible locations, rotating machinery, hazardous areas, and mobile assets can be reached with wireless sensors. WSNs can provide today wireless link reliability, real-time capabilities, and quality-of-service and enable industrial and related wireless sense and control applications.

Management of industrial and factory applications is largely focused on monitoring whether the system is still functional, real-time continuous performance monitoring, and optimization as necessary. The factory network might be part of a campus network or connected to the Internet. The constrained devices in such a network need to be able to establish configuration themselves (auto-configuration) and might need to deal with error conditions as much as possible locally. Access control has to be provided with multi-level administrative access and security. Support and diagnostics can be provided through remote monitoring access centralized outside of the factory.

Factory automation tasks require that continuous monitoring be used to optimize production. Groups of manufacturing and monitoring devices could be defined to establish relationships between them. To ensure timely optimization of processes, commands from the NMS must arrive at all destination within an appropriate duration. This duration could change based on the manufacturing task being performed. Installation and operation of factory networks have different requirements. During the installation phase many networks, usually distributed along different parts of the factory/assembly line, co-exist without a connection to a common backbone. A specialized installation tool is typically used to configure the functions of different types of devices, in different factory location, in a secure manner. At the end of the installation phase, interoperability between these stand-alone networks and devices must be enabled. During the operation phase, these stand-alone networks are connected to a common backbone so that they may retrieve control information from and send commands to appropriate devices.

Management responsibility is typically owned by the organization running the industrial application. Since the monitoring

applications must handle a potentially large number of failures, the time scale for detecting and recording failures is for some of these applications likely measured in minutes. However, for certain industrial applications, much tighter time scales may exist, e.g. in real-time, which might be enforced by the manufacturing process or the use of critical material. Management protocols and NMSs must ensure appropriate access control since different users of industrial control systems will have varying levels of permissions. E.g., while supervisors might be allowed to change production parameters, they should not be allowed to modify the functional configuration of devices like a technician should. It is also important to ensure integrity and availability of data since malfunctions can potentially become safety issues. This also implies that management systems must be able to react to situations that may pose dangers to worker safety.

4.4. Energy Management

The EMAN working group developed an energy management framework [RFC7326] for devices and device components within or connected to communication networks. This document observes that one of the challenges of energy management is that a power distribution network is responsible for the supply of energy to various devices and components, while a separate communication network is typically used to monitor and control the power distribution network. Devices in the context of energy management can be monitored for parameters like power, energy, demand and power quality. If a device contains batteries, they can be also monitored and managed.

Energy devices differ in complexity and may include basic sensors or switches, specialized electrical meters, or power distribution units (PDU), and subsystems inside the network devices (routers, network switches) or home or industrial appliances. The operators of an Energy Management System are either the utility providers or customers that aim to control and reduce the energy consumption and the associated costs. The topology in use differs and the deployment can cover areas from small surfaces (individual homes) to large geographical areas. The EMAN requirements document [RFC6988] discusses the requirements for energy management concerning monitoring and control functions.

It is assumed that energy management will apply to a large range of devices of all classes and networks topologies. Specific resource monitoring like battery utilization and availability may be specific to devices with lower physical resources (device classes C0 or C1 [RFC7228]).

Energy management is especially relevant to the Smart Grid. A Smart Grid is an electrical grid that uses data networks to gather and to act on energy and power-related information in an automated fashion with the goal to improve the efficiency, reliability, economics, and sustainability of the production and distribution of electricity.

Smart Metering is a good example of Smart Grid based energy management applications. Different types of possibly wireless small meters produce all together a large amount of data, which is collected by a central entity and processed by an application server, which may be located within the customer's residence or off-site in a data-center. The communication infrastructure can be provided by a mobile network operator as the meters in urban areas will have most likely a cellular or WiMAX radio. In case the application server is located within the residence, such meters are more likely to use Wi-Fi protocols to interconnect with an existing network.

An Advanced Metering Infrastructure (AMI) network is another example of the Smart Grid that enables an electric utility to retrieve frequent electric usage data from each electric meter installed at a customer's home or business. Unlike Smart Metering, in which case the customer or their agents install appliance level meters, an AMI infrastructure is typically managed by the utility providers and could also include other distribution automation devices like transformers and reclosers. Meters in AMI networks typically contain constrained devices that connect to mesh networks with a low-bandwidth radio. Usage data and outage notifications can be sent by these meters to the utility's headend systems, via aggregation points of higher-end router devices that bridge the constrained network to a less constrained network via cellular, WiMAX, or Ethernet. Unlike meters, these higher-end devices might be installed on utility poles owned and operated by a separate entity.

It thereby becomes important for a management application to not only be able to work with diverse types of devices, but also over multiple links that might be operated and managed by separate entities, each having divergent policies for their own devices and network segments. During management operations, like firmware updates, it is important that the management system performs robustly in order to avoid accidental outages of critical power systems that could be part of AMI networks. In fact, since AMI networks must also report on outages, the management system might have to manage the energy properties of battery operated AMI devices themselves as well.

A management system for home based Smart Metering solutions is likely to have devices laid out in a simple topology. However, AMI networks installations could have thousands of nodes per router, i.e., higher-end device, which organize themselves in an ad-hoc manner. As such,

a management system for AMI networks will need to discover and operate over complex topologies as well. In some situations, it is possible that the management system might also have to setup and manage the topology of nodes, especially critical routers. Encryption key management and sharing in both types of networks is also likely to be important for providing confidentiality for all data traffic. In AMI networks the key may be obtained by a meter only after an end-to-end authentication process based on certificates. Smart Metering solution could adopt a similar approach or the security may be implied due to the encrypted Wi-Fi networks they become part of.

The management of such a network requires end-to-end management of and information exchange through different types of networks. However, as of today there is no integrated energy management approach and no common information model available. Specific energy management applications or network islands use their own management mechanisms.

4.5. Medical Applications

Constrained devices can be seen as an enabling technology for advanced and possibly remote health monitoring and emergency notification systems, ranging from blood pressure and heart rate monitors to advanced devices capable of monitoring implanted technologies, such as pacemakers or advanced hearing aids. Medical sensors may not only be attached to human bodies, they might also exist in the infrastructure used by humans such as bathrooms or kitchens. Medical applications will also be used to ensure treatments are being applied properly and they might guide people losing orientation. Fitness and wellness applications, such as connected scales or wearable heart monitors, encourage consumers to exercise and empower self-monitoring of key fitness indicators. Different applications use Bluetooth, Wi-Fi or ZigBee connections to access the patient's smartphone or home cellular connection to access the Internet.

Constrained devices that are part of medical applications are managed either by the users of those devices or by an organization providing medical (monitoring) services for physicians. In the first case, management must be automatic and/or easy to install and setup by average people. In the second case, it can be expected that devices be controlled by specially trained people. In both cases, however, it is crucial to protect the safety and privacy of the people to which medical devices are attached. Security precautions to protect access (authentication, encryption, integrity protections, etc.) to such devices may be critical to safeguarding the individual. The level of access granted to different users also may need to be

regulated. For example, an authorized surgeon or doctor must be allowed to configure all necessary options on the devices, however, a nurse or technician may only be allowed to retrieve data that can assist in diagnosis. Even though the data collected by a heart beat monitor might be protected, the pure fact that someone carries such a device may need protection. As such, certain medical appliances may not want to participate in discovery and self-configuration protocols in order to remain invisible.

Many medical devices are likely to be used (and relied upon) to provide data to physicians in critical situations since the biggest market is likely elderly and handicapped people. Timely delivery of data can be quite important in certain applications like patient mobility monitoring in old-age homes. Data must reach the physician and/or emergency services within specified limits of time in order to be useful. As such, fault detection of the communication network or the constrained devices becomes a crucial function of the management system that must be carried out with high reliability and, depending on the medical appliance and its application, within seconds.

4.6. Building Automation

Building automation comprises the distributed systems designed and deployed to monitor and control the mechanical, electrical and electronic systems inside buildings with various destinations (e.g., public and private, industrial, institutions, or residential). Advanced Building Automation Systems (BAS) may be deployed concentrating the various functions of safety, environmental control, occupancy, security. More and more the deployment of the various functional systems is connected to the same communication infrastructure (possibly Internet Protocol based), which may involve wired or wireless communications networks inside the building.

Building automation requires the deployment of a large number (10-100.000) of sensors that monitor the status of devices, and parameters inside the building and controllers with different specialized functionality for areas within the building or the totality of the building. Inter-node distances between neighboring nodes vary between 1 to 20 meters. The NMS must, as a result, be able to manage and monitor a large number of devices, which may be organized in multi-hop meshed networks. Distances between the nodes, and the use of constrained protocols, means that networks of nodes might be segmented. The management of such network segments and nodes in these segments should be possible. Contrary to home automation, in building management the devices are expected to be managed assets and known to a set of commissioning tools and a data storage, such that every connected device has a known origin. This requires the management system to be able to discover devices on the

network and ensure that the expected list of devices is currently matched. Management here includes verifying the presence of the expected devices and detecting the presence of unwanted devices.

Examples of functions performed by controllers in building automation are regulating the quality, humidity, and temperature of the air inside the building and lighting. Other systems may report the status of the machinery inside the building like elevators, or inside the rooms like projectors in meeting rooms. Security cameras and sensors may be deployed and operated on separate dedicated infrastructures connected to the common backbone. The deployment area of a BAS is typically inside one building (or part of it) or several buildings geographically grouped in a campus. A building network can be composed of network segments, where a network segment covers a floor, an area on the floor, or a given functionality (e.g., security cameras). It is possible that the management tasks of different types of some devices might be separated from others (e.g., security cameras might operate and be managed via a separate network to the HVAC in a building).

Some of the sensors in Building Automation Systems (for example fire alarms or security systems) register, record and transfer critical alarm information and therefore must be resilient to events like loss of power or security attacks. A management system must be able to deal with unintentional segmentation of networks due to power loss or channel unavailability. It must also be able to detect security events. Due to specific operating conditions required from certain devices, there might be a need to certify components and subsystems operating in such constrained conditions based on specific requirements. Also in some environments, the malfunctioning of a control system (like temperature control) needs to be reported in the shortest possible time. Complex control systems can misbehave, and their critical status reporting and safety algorithms need to be basic and robust and perform even in critical conditions. Providing this monitoring, configuration and notification service is an important task of the management system used in building automation.

Building automation solutions are deployed in some cases in newly designed buildings, in other cases it might be over existing infrastructures. In the first case, there is a broader range of possible solutions, which can be planned for the infrastructure of the building. In the second case the solution needs to be deployed over an existing infrastructure taking into account factors like existing wiring, distance limitations, the propagation of radio signals over walls and floors, thereby making deployment difficult. As a result, some of the existing WLAN solutions (e.g., IEEE 802.11 or IEEE 802.15) may be deployed. In mission-critical or security sensitive environments and in cases where link failures happen often,

topologies that allow for reconfiguration of the network and connection continuity may be required. Some of the sensors deployed in building automation may be very simple constrained devices for which C0 or C1 [RFC7228] may be assumed.

For lighting applications, groups of lights must be defined and managed. Commands to a group of light must arrive within 200 ms at all destinations. The installation and operation of a building network has different requirements. During the installation, many stand-alone networks of a few to 100 nodes co-exist without a connection to the backbone. During this phase, the nodes are identified with a network identifier related to their physical location. Devices are accessed from an installation tool to connect them to the network in a secure fashion. During installation, the setting of parameters of common values to enable interoperability may be required. During operation, the networks are connected to the backbone while maintaining the network identifier to physical location relation. Network parameters like address and name are stored in DNS. The names can assist in determining the physical location of the device.

It is also important for a building automation NMS to take safety and security into account. Ensuring privacy and confidentiality of data, such that unauthorized parties do not get access to it, is likely to be important since users' individual behaviors could be potentially understood via their settings. Appropriate security considerations for authorization and access control to the NMS is also important since different users are likely to have varied levels of operational permissions in the system. E.g., while end users should be able to control lighting systems, HVACs, etc., only qualified technicians should be able to configure parameters that change the fundamental operation of a device. It is also important for devices and the NMS to be able to detect and report any tampering they might detect, since these could lead to potential user safety concerns, e.g., if sensors controlling air quality are tampered with such that the levels of Carbon Monoxide become life threatening. This implies that a NMS should also be able to deal with and appropriately prioritize situations that might potentially lead to safety concerns.

4.7. Home Automation

Home automation includes the control of lighting, heating, ventilation, air conditioning, appliances, entertainment and home security devices to improve convenience, comfort, energy efficiency, and safety. It can be seen as a residential extension of building automation. However, unlike a building automation system, the infrastructure in a home is operated in a considerably more ad-hoc manner. While in some installations it is likely that there is no

centralized management system, akin to a Building Automation System (BAS), available, in other situations outsourced and cloud based systems responsible for managing devices in the home might be used.

Home automation networks need a certain amount of configuration (associating switches or sensors to actuators) that is either provided by electricians deploying home automation solutions, by third party home automation service providers (e.g., small specialized companies or home automation device manufacturers) or by residents by using the application user interface provided by home automation devices to configure (parts of) the home automation solution. Similarly, failures may be reported via suitable interfaces to residents or they might be recorded and made available to services providers in charge of the maintenance of the home automation infrastructure.

The management responsibility lies either with the residents or it may be outsourced to electricians and/or third parties providing management of home automation solutions as a service. A varying combination of electricians, service providers or the residents may be responsible for different aspects of managing the infrastructure. The time scale for failure detection and resolution is in many cases likely counted in hours to days.

4.8. Transport Applications

Transport application is a generic term for the integrated application of communications, control, and information processing in a transportation system. Transport telematics or vehicle telematics are used as a term for the group of technologies that support transportation systems. Transport applications running on such a transportation system cover all modes of the transport and consider all elements of the transportation system, i.e. the vehicle, the infrastructure, and the driver or user, interacting together dynamically. Examples for transport applications are inter and intra vehicular communication, smart traffic control, smart parking, electronic toll collection systems, logistic and fleet management, vehicle control, and safety and road assistance.

As a distributed system, transport applications require an end-to-end management of different types of networks. It is likely that constrained devices in a network (e.g. a moving in-car network) have to be controlled by an application running on an application server in the network of a service provider. Such a highly distributed network including cellular devices on vehicles is assumed to include a wireless access network using diverse long distance wireless technologies such as WiMAX, 3G/LTE or satellite communication, e.g. based on an embedded hardware module. As a result, the management of

constrained devices in the transport system might be necessary to plan top-down and might need to use data models obliged from and defined on the application layer. The assumed device classes in use are mainly C2 [RFC7228] devices. In cases, where an in-vehicle network is involved, C1 devices [RFC7228] with limited capabilities and a short-distance constrained radio network, e.g. IEEE 802.15.4 might be used additionally.

All Transport Applications will require an IT infrastructure to run on top of, e.g., in public transport scenarios like trains, bus or metro network infrastructure might be provided, maintained and operated by third parties like mobile network or satellite network operators. However, the management responsibility of the transport application typically rests within the organization running the transport application (in the public transport scenario, this would typically be the public transport operator). Different aspects of the infrastructure might also be managed by different entities. For example, the in-car devices are likely to be installed and managed by the manufacturer, while the public works might be responsible for the on-road vehicular communication infrastructure used by these devices. The back-end infrastructure is also likely to be maintained by third party operators. As such, the NMS must be able to deal with different network segments, each being operated and controlled by separate entities, and enable appropriate access control and security as well.

Depending on the type of application domain (vehicular or stationary) and service being provided, it would be important for the NMS to be able to function with different architectures, since different manufacturers might have their own proprietary systems relying on a specific Management Topology Option, as described in [COM-REQ]. Moreover, constituents of the network can be either private, belonging to individuals or private companies, or owned by public institutions leading to different legal and organization requirements. Across the entire infrastructure, a variety of constrained devices are likely to be used, and must be individually managed. The NMS must be able to either work directly with different types of devices, or have the ability to interoperate with multiple different systems.

The challenges in the management of vehicles in a mobile transport application are manifold. The up-to-date position of each node in the network should be reported to the corresponding management entities, since the nodes could be moving within or roaming between different networks. Secondly, a variety of troubleshooting information, including sensitive location information, needs to be reported to the management system in order to provide accurate service to the customer. Management systems dealing with mobile

nodes could possibly exploit specific patterns in the mobility of the nodes. These patterns emerge due to repetitive vehicular usage in scenarios like people commuting to work, logistics supply vehicles transporting shipments between warehouses, etc. The NMS must also be able to handle partitioned networks, which would arise due to the dynamic nature of traffic resulting in large inter-vehicle gaps in sparsely populated scenarios. Since mobile nodes might roam in remote networks, the NMS should be able to provide operating configuration updates regardless of node location.

The constrained devices in a moving transport network might be initially configured in a factory and a reconfiguration might be needed only rarely. New devices might be integrated in an ad-hoc manner based on self-management and -configuration capabilities. Monitoring and data exchange might be necessary to do via a gateway entity connected to the back-end transport infrastructure. The devices and entities in the transport infrastructure need to be monitored more frequently and can be able to communicate with a higher data rate. The connectivity of such entities does not necessarily need to be wireless. The time scale for detecting and recording failures in a moving transport network is likely measured in hours and repairs might easily take days. It is likely that a self-healing feature would be used locally. On the other hand, failures in fixed transport application infrastructure (e.g., traffic-lights, digital signage displays) is likely to be measured in minutes so as to avoid untoward traffic incidents. As such, the NMS must be able to deal with differing timeliness requirements based on the type of devices.

Since transport applications of the constrained devices and networks deal with automotive vehicles, malfunctions and misuse can potentially lead to safety concerns as well. As such, besides access control, privacy of user data and timeliness management systems should also be able to detect situations that are potentially hazardous to safety. Some of these situations could be automatically mitigated, e.g., traffic lights with incorrect timing, but others might require human intervention, e.g., failed traffic lights. The management system should take appropriate actions in these situations. Maintaining data confidentiality and integrity is also an important security aspect of a management system since tampering (or malfunction) can also lead to potentially dangerous situations.

4.9. Community Network Applications

Community networks are comprised of constrained routers in a multi-hop mesh topology, communicating over a lossy, and often wireless channels. While the routers are mostly non-mobile, the topology may be very dynamic because of fluctuations in link quality of the

(wireless) channel caused by, e.g., obstacles, or other nearby radio transmissions. Depending on the routers that are used in the community network, the resources of the routers (memory, CPU) may be more or less constrained - available resources may range from only a few kilobytes of RAM to several megabytes or more, and CPUs may be small and embedded, or more powerful general-purpose processors. Examples of such community networks are the FunkFeuer network (Vienna, Austria), FreiFunk (Berlin, Germany), Seattle Wireless (Seattle, USA), and AWMN (Athens, Greece). These community networks are public and non-regulated, allowing their users to connect to each other and - through an uplink to an ISP - to the Internet. No fee, other than the initial purchase of a wireless router, is charged for these services. Applications of these community networks can be diverse, e.g., location based services, free Internet access, file sharing between users, distributed chat services, social networking, video sharing, etc.

As an example of a community network, the FunkFeuer network comprises several hundred routers, many of which have several radio interfaces (with omnidirectional and some directed antennas). The routers of the network are small-sized wireless routers, such as the Linksys WRT54GL, available in 2011 for less than 50 Euros. These routers, with 16 MB of RAM and 264 MHz of CPU power, are mounted on the rooftops of the users. When new users want to connect to the network, they acquire a wireless router, install the appropriate firmware and routing protocol, and mount the router on the rooftop. IP addresses for the router are assigned manually from a list of addresses (because of the lack of auto-configuration standards for mesh networks in the IETF).

While the routers are non-mobile, fluctuations in link quality require an ad hoc routing protocol that allows for quick convergence to reflect the effective topology of the network (such as NHDP [RFC6130] and OLSRv2 [RFC7181] developed in the MANET WG). Usually, no human interaction is required for these protocols, as all variable parameters required by the routing protocol are either negotiated in the control traffic exchange, or are only of local importance to each router (i.e. do not influence interoperability). However, external management and monitoring of an ad hoc routing protocol may be desirable to optimize parameters of the routing protocol. Such an optimization may lead to a more stable perceived topology and to a lower control traffic overhead, and therefore to a higher delivery success ratio of data packets, a lower end-to-end delay, and less unnecessary bandwidth and energy usage.

Different use cases for the management of community networks are possible:

- o One single Network Management Station, e.g. a border gateway providing connectivity to the Internet, requires managing or monitoring routers in the community network, in order to investigate problems (monitoring) or to improve performance by changing parameters (managing). As the topology of the network is dynamic, constant connectivity of each router towards the management station cannot be guaranteed. Current network management protocols, such as SNMP and NETCONF, may be used (e.g., using interfaces such as the NHDp-MIB [RFC6779]). However, when routers in the community network are constrained, existing protocols may require too many resources in terms of memory and CPU; and more importantly, the bandwidth requirements may exceed the available channel capacity in wireless mesh networks. Moreover, management and monitoring may be unfeasible if the connection between the network management station and the routers is frequently interrupted.
- o Distributed network monitoring, in which more than one management station monitors or manages other routers. Because connectivity to a server cannot be guaranteed at all times, a distributed approach may provide a higher reliability, at the cost of increased complexity. Currently, no IETF standard exists for distributed monitoring and management.
- o Monitoring and management of a whole network or a group of routers. Monitoring the performance of a community network may require more information than what can be acquired from a single router using a network management protocol. Statistics, such as topology changes over time, data throughput along certain routing paths, congestion etc., are of interest for a group of routers (or the routing domain) as a whole. As of 2014, no IETF standard allows for monitoring or managing whole networks, instead of single routers.

4.10. Field Operations

The challenges of configuration and monitoring of networks operated in the field by rescue and security agencies can be different from the other use cases since the requirements and operating conditions of such networks are quite different.

With technology advancements, field networks operated nowadays are becoming large and can consist of varieties of different types of equipment that run different protocols and tools that obviously increase complexity of these mission-critical networks. In many scenarios, configurations are, most likely, manually performed. Furthermore, some legacy and even modern devices do not even support IP networking. A majority of protocols and tools developed by

vendors that are being used are proprietary, which makes integration more difficult.

The main reason for this disjoint operation scenario is that most equipment is developed with specific task requirements in mind, rather than interoperability of the varied equipment types. For example, the operating conditions experienced by high altitude security equipment is significantly different from that used in desert conditions. Similarly, search and rescue operations equipment used in case of fire rescue has different requirements than flood relief equipment. Furthermore, inter-operation of equipment with telecommunication equipment was not an expected outcome or in some scenarios this may not even be desirable.

Currently, field networks operate with a fixed Network Operations Center (NOC) that physically manages the configuration and evaluation of all field devices. Once configured, the devices might be deployed in fixed or mobile scenarios. Any configuration changes required would need to be appropriately encrypted and authenticated to prevent unauthorized access.

Hierarchical management of devices is a common requirement in such scenarios since local managers or operators may need to respond to changing conditions within their purview. The level of configuration management available at each hierarchy must also be closely governed.

Since many field operation devices are used in hostile environments, a high failure and disconnection rate should be tolerated by the NMS, which must also be able to deal with multiple gateways and disjoint management protocols.

Multi-national field operations involving search, rescue and security are becoming increasingly common, requiring inter-operation of a diverse set of equipment designed with different operating conditions in mind. Furthermore, different intra- and inter-governmental agencies are likely to have a different set of standards, best practices, rules and regulation, and implementation approaches that may contradict or conflict with each other. The NMS should be able to detect these and handle them in an acceptable manner, which may require human intervention.

5. IANA Considerations

This document does not introduce any new code-points or namespaces for registration with IANA.

Note to RFC Editor: this section may be removed on publication as an RFC.

6. Security Considerations

This document discusses use cases for management of networks with constrained devices. The security considerations described throughout the companion document [COM-REQ] apply here as well.

7. Contributors

Following persons made significant contributions to and reviewed this document:

- o Ulrich Herberg contributed the Section 4.9 on Community Network Applications.
- o Peter van der Stok contributed to Section 4.6 on Building Automation.
- o Zhen Cao contributed to Section 2.2 Cellular Access Technologies.
- o Gilman Tolle contributed the Section 4.4 on Automated Metering Infrastructure.
- o James Nguyen and Ulrich Herberg contributed to Section 4.10 on Military operations.

8. Acknowledgments

Following persons reviewed and provided valuable comments to different versions of this document:

Dominique Barthel, Carsten Bormann, Zhen Cao, Benoit Claise, Bert Greevenbosch, Ulrich Herberg, Ted Lemon, Kathleen Moriarty, James Nguyen, Zach Shelby, Peter van der Stok, and Martin Thomson.

The editors would like to thank the reviewers and the participants on the Coman maillist for their valuable contributions and comments.

9. Informative References

- [RFC6130] Clausen, T., Dearlove, C., and J. Dean, "Mobile Ad Hoc Network (MANET) Neighborhood Discovery Protocol (NHDP)", RFC 6130, April 2011.
- [RFC6568] Kim, E., Kaspar, D., and JP. Vasseur, "Design and Application Spaces for IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs)", RFC 6568, April 2012.

- [RFC6779] Herberg, U., Cole, R., and I. Chakeres, "Definition of Managed Objects for the Neighborhood Discovery Protocol", RFC 6779, October 2012.
- [RFC6988] Quittek, J., Chandramouli, M., Winter, R., Dietz, T., and B. Claise, "Requirements for Energy Management", RFC 6988, September 2013.
- [RFC7181] Clausen, T., Dearlove, C., Jacquet, P., and U. Herberg, "The Optimized Link State Routing Protocol Version 2", RFC 7181, April 2014.
- [RFC7228] Bormann, C., Ersue, M., and A. Keranen, "Terminology for Constrained-Node Networks", RFC 7228, May 2014.
- [RFC7326] Parello, J., Claise, B., Schoening, B., and J. Quittek, "Energy Management Framework", RFC 7326, September 2014.
- [COM-REQ] Ersue, M., Romascanu, D., and J. Schoenwaelder, "Management of Networks with Constrained Devices: Problem Statement and Requirements", draft-ietf-opsawg-coman-probstate-reqs (work in progress), February 2014.
- [IOT-SEC] Garcia-Morchon, O., Kumar, S., Keoh, S., Hummen, R., and R. Struik, "Security Considerations in the IP-based Internet of Things", draft-garcia-core-security-06 (work in progress), September 2013.

Appendix A. Change Log

- A.1. draft-ietf-opsawg-coman-use-cases-04 - draft-ietf-opsawg-coman-use-cases-05
 - o Added text regarding security and safety considerations to the Environmental Monitoring, Infrastructure Monitoring, Industrial Applications, Medical Applications, Building Automation and Transport Applications section.
 - o Adopted text as per comments received from Kathleen Moriarty during IESG review.
 - o Added security related text to use cases for addressing concerns raised by Ted Lemon during the IESG review.

- A.2. draft-ietf-opsawg-coman-use-cases-03 - draft-ietf-opsawg-coman-use-cases-04
- o Resolved Gen-ART review comments received from Martin Thomson.
 - o Deleted company name for the list of contributors.
 - o Added Martin Thomson to Acknowledgments section.
- A.3. draft-ietf-opsawg-coman-use-cases-02 - draft-ietf-opsawg-coman-use-cases-03
- o Updated references to take into account RFCs that have now been published
 - o Added text to the access technologies section explaining why fixed line technologies (e.g., powerline communications) have not been discussed.
 - o Created a new section, Device Lifecycle, discussing the impact of different device lifecycle stages on the management of constrained networks.
 - o Homogenized usage of device classes to form C0, C1 and C2.
 - o Ensured consistency in usage of Wi-Fi, ZigBee and other terminologies.
 - o Added text clarifying the management aspects of the Building Automation and Industrial Automation use cases.
 - o Clarified the meaning of unreliability in context of constrained devices and networks.
 - o Added information regarding the configuration and operation of factory automation use case, based on the type of information provided in the building automation use case.
 - o Fixed editorial issues discovered by reviewers.
- A.4. draft-ietf-opsawg-coman-use-cases-01 - draft-ietf-opsawg-coman-use-cases-02
- o Renamed Mobile Access Technologies section to Cellular Access Technologies
 - o Changed references to mobile access technologies to now read cellular access technologies.

- o Added text to the introduction to point out that the list of use cases is not exhaustive since others unknown to the authors might exist.
- o Updated references to take into account RFCs that have been now published.
- o Updated Environmental Monitoring section to make it clear that in some scenarios it may not be prudent to repair devices.
- o Added clarification in Infrastructure Monitoring section that reliable communication is achieved via application layer transactions
- o Removed reference to Energy Devices from Energy Management section, instead labeling them as devices within the context of energy management.
- o Reduced descriptive content in Energy Management section.
- o Rewrote text in Energy Management section to highlight management characteristics of Smart Meter and AMI networks.
- o Added text regarding timely delivery of information, and related management system characteristic, to the Medical Applications section
- o Changed subnets to network segment in Building Automation section.
- o Changed structure to infrastructure in Building Automation section, and added text to highlight associated deployment difficulties.
- o Removed Trickle timer as example of common values to be set in Building Automation section.
- o Added text regarding the possible availability of outsourced and cloud based management systems for Home Automation.
- o Added text to Transport Applications section to highlight the requirement of IT infrastructure for such applications to function on top of.
- o Merged the Transport Applications and Vehicular Networks section together. Following changes to the Vehicular Networks section were merged back into Transport Applications

- * Replaced wireless last hops with wireless access to vehicles in Vehicular Networks.
 - * Expanded proprietary systems to "systems relying on a specific Management Topology Option, as described in [COM-REQ]." within Vehicular Networks section.
 - * Added text regarding mobility patterns to Vehicular Networks.
 - o Changed the Military Operations use case to Field Operations and edited the text to be suitable to such scenarios.
- A.5. draft-ietf-opsawg-coman-use-cases-00 - draft-ietf-opsawg-coman-use-cases-01
- o Reordered some use cases to improve the flow.
 - o Added "Vehicular Networks".
 - o Shortened the Military Operations use case.
 - o Started adding substance to the security considerations section.
- A.6. draft-ersue-constrained-mgmt-03 - draft-ersue-opsawg-coman-use-cases-00
- o Reduced the terminology section for terminology addressed in the LWIG and Coman Requirements drafts. Referenced the other drafts.
 - o Checked and aligned all terminology against the LWIG terminology draft.
 - o Spent some effort to resolve the intersection between the Industrial Application, Home Automation and Building Automation use cases.
 - o Moved section section 3. Use Cases from the companion document [COM-REQ] to this draft.
 - o Reformulation of some text parts for more clarity.
- A.7. draft-ersue-constrained-mgmt-02-03
- o Extended the terminology section and removed some of the terminology addressed in the new LWIG terminology draft. Referenced the LWIG terminology draft.

- o Moved Section 1.3. on Constrained Device Classes to the new LWIG terminology draft.
- o Class of networks considering the different type of radio and communication technologies in use and dimensions extended.
- o Extended the Problem Statement in Section 2. following the requirements listed in Section 4.
- o Following requirements, which belong together and can be realized with similar or same kind of solutions, have been merged.
 - * Distributed Management and Peer Configuration,
 - * Device status monitoring and Neighbor-monitoring,
 - * Passive Monitoring and Reactive Monitoring,
 - * Event-driven self-management - Self-healing and Periodic self-management,
 - * Authentication of management systems and Authentication of managed devices,
 - * Access control on devices and Access control on management systems,
 - * Management of Energy Resources and Data models for energy management,
 - * Software distribution (group-based firmware update) and Group-based provisioning.
- o Deleted the empty section on the gaps in network management standards, as it will be written in a separate draft.
- o Added links to mentioned external pages.
- o Added text on OMA M2M Device Classification in appendix.

A.8. draft-ersue-constrained-mgmt-01-02

- o Extended the terminology section.
- o Added additional text for the use cases concerning deployment type, network topology in use, network size, network capabilities, radio technology, etc.

- o Added examples for device classes in a use case.
- o Added additional text provided by Cao Zhen (China Mobile) for Mobile Applications and by Peter van der Stok for Building Automation.
- o Added the new use cases 'Advanced Metering Infrastructure' and 'MANET Concept of Operations in Military'.
- o Added the section 'Managing the Constrainedness of a Device or Network' discussing the needs of very constrained devices.
- o Added a note that the requirements in [COM-REQ] need to be seen as standalone requirements and the current document does not recommend any profile of requirements.
- o Added a section in [COM-REQ] for the detailed requirements on constrained management matched to management tasks like fault, monitoring, configuration management, Security and Access Control, Energy Management, etc.
- o Solved nits and added references.
- o Added Appendix A on the related development in other bodies.
- o Added Appendix B on the work in related research projects.

A.9. draft-ersue-constrained-mgmt-00-01

- o Splitted the section on 'Networks of Constrained Devices' into the sections 'Network Topology Options' and 'Management Topology Options'.
- o Added the use case 'Community Network Applications' and 'Mobile Applications'.
- o Provided a Contributors section.
- o Extended the section on 'Medical Applications'.
- o Solved nits and added references.

Authors' Addresses

Mehmet Ersue (editor)
Nokia Networks

Email: mehmet.ersue@nsn.com

Dan Romascanu
Avaya

Email: dromasca@avaya.com

Juergen Schoenwaelder
Jacobs University Bremen

Email: j.schoenwaelder@jacobs-university.de

Anuj Sehgal
Jacobs University Bremen

Email: s.anuj@jacobs-university.de

ipsecme
Internet-Draft
Intended status: Standards Track
Expires: 14 November 2022

D. Migault
Ericsson
T. Guggemos
LMU Munich
C. Bormann
Universitaet Bremen TZI
D. Schinazi
Google LLC
13 May 2022

ESP Header Compression and Diet-ESP
draft-mglt-ipsecme-diet-esp-08

Abstract

With the use of encrypted ESP for secure IP communication, the compression of IP payload is only possible with complex frameworks, such as RObust Header Compression (ROHC). Such frameworks are too complex for numerous use cases and especially for IoT scenarios, which makes IPsec not being used here, although it offers architectural benefits.

ESP Header Compression (EHC) defines a flexible framework to compress communications protected with IPsec/ESP. Compression and decompression is defined by EHC Rules orchestrated by EHC Strategies. The necessary state is hold within the IPsec Security Association and can be negotiated during key agreement, e.g. with IKEv2.

The document specifies the necessary parameters of the EHC Context to allow compression of ESP and the most common included protocols, such as IPv4, IPv6, UDP and TCP and the corresponding EHC Rules. It also defines the Diet-ESP EHC Strategy which compresses up to 32 bytes per packet for traditional IPv6 VPN and up to 66 bytes for IPv6 VPN sent over a single TCP or UDP session.

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 14 November 2022.

Copyright Notice

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

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Revised BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Revised BSD License.

Table of Contents

1. Requirements notation	3
2. Introduction	3
3. Terminology	4
4. Protocol Overview	5
5. IPsec Compression Mode	7
6. EHC Context	7
6.1. EHC Context Parameters for ESP	8
6.2. EHC Context Parameters for Inner IP	8
6.3. EHC Context Parameters for Transport Protocol	10
7. EHC Rules	12
7.1. EHC Rules for ESP	14
7.2. EHC Rules for inner IPv4	16
7.3. EHC Rules for inner IPv6	19
7.4. EHC Rules for UDP	21
7.5. EHC Rules for UDP-Lite	22
7.6. EHC Rules for TCP	22
8. Diet-ESP EHC Strategy	24
8.1. Outbound Packet Processing	28
8.2. Inbound Packet Processing	30
9. IANA Considerations	32
10. Security Considerations	32
11. Privacy Considerations	34
12. Acknowledgment	34
13. References	34
13.1. Normative References	34

13.2. Informational References	35
Appendix A. Illustrative Examples	36
A.1. Single UDP Session IoT VPN	36
A.2. Single TCP session IoT VPN	39
A.3. Traditional VPN	43
Authors' Addresses	52

1. Requirements notation

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

2. Introduction

IPsec/ESP [RFC4303] secures communications either using end-to-end security or by building a VPN, where the traffic is carried to a secure domain via a security gateway.

IPsec/ESP was not designed to minimize its associated networking overhead. In fact, bandwidth optimization often adds computational overhead that may negatively impact large infrastructures in which bandwidth usage is not a constraint. On the other hand, in IoT communications, sending extra bytes can significantly impact the battery life of devices and thus the life time of the device. The document describes a framework that optimizes the networking overhead associated to IPsec/ESP for these devices.

Most compression mechanisms work with dynamic compression contexts. Some mechanisms, such as ROHC [RFC5858], agree and dynamically change the context over a dedicated channel. Others, such as 6LoWPAN, send the context together with the actual protocol information in a separate compression header. Those mechanism fail when it comes to compress encrypted payloads as appearing in ESP. This is found to be a major reason, why IPsec and in particular ESP is not widely developed in environments where bandwidth saving is a critical task, such as in IoT scenarios.

ESP Header Compression (EHC) chooses another form of context agreement, which is similar to the one defined by Static Context Header Compression (SCHC). It works with a static compression context agreed for a specific Security Association. The context itself can be negotiated during the key agreement, which allows only minimal the changes to the actual ESP implementation.

EHC itself is defined as a framework that specifically compresses ESP protected communications. EHC is highly flexible to address any use case where compression is necessary. EHC takes advantage of the negotiation between the communication endpoint to agree on the cryptographic parameters, which in some cases already includes parameters that remain constant during the communications (like layer 4 ports, or IP addresses) and can thus be used as part of the compression context. Only additional, EHC specific parameters need to be agreed for the purpose of compression. In addition EHC Rules define how fields may be compressed and decompressed given the provided parameters. Finally, EHC defines EHC Strategy which defines how a set of EHC Rule is coordinated.

This document specifies EHC Context parameters for the most common Layer 3 and 4 protocols and the associated EHC Rules. Additionally, an EHC Strategy called Diet-ESP is defined, which compresses up to 32 bytes per packet for traditional VPN and up to 66 bytes for VPN set over a single TCP or UDP session. Its main purpose is a maximum level of compression with a minimum of additional agreement. This is achieved by defining a default usage of existing IPsec SA parameters wherever possible.

3. Terminology

This document uses the following terminology:

- EHC ESP Header Compression
- IoT Internet of Things
- IP If not stated otherwise, IP means IPv6.
- LSB Least Significant Bytes
- MSB Most Significant Bytes
- SAD IPsec Security Association Database
- SA IPsec Security Association
- SPD IPsec Security Policy Database
- TS IPsec Traffic Selector
- SPI ESP Security Parameter Index
- SN ESP Sequence Number
- PAD ESP Padding
- PL ESP Pad Length
- NH Next Header
- IV Initialization Vector
- IIV Implicit Initialization Vector
- ICV Integrity Check Value
- VPN Virtual Private Network

4. Protocol Overview

ESP Header Compression (EHC) compresses IPsec ESP packets, thus reducing the size of the packet sent on the wire, while carrying an equivalent level of information with an equivalent level of security.

EHC is able to compress any protocol encapsulated in ESP and ESP itself. Concerned fields include those of the ESP protocol, as well as other protocols in the ESP payload such as the IP header when the tunnel mode is used, but also upper layer protocols, such as the UDP or the TCP header. Non ESP fields may be compressed by ESP under certain circumstances, but EHC is not intended to provide a generic way outside of ESP to compress these protocols. Compression of the unprotected IP header and the unencrypted ESP header may be performed by mechanism such as 6LoWPAN [RFC4944], SCHC [RFC8724], ROHC [RFC5795] or 6LoWPAN-GHC [RFC7400].

EHC is based on a static compression context, EHC Rules coordinated by an EHC Strategy:

EHC Context: Stores the information of a specific header field which can be compressed by EHC. This can be specific header values such as IP addresses or L4 ports do not have to be send on the wire at all, or compression information for fields which can be partially compressed, such as sequence numbers.

EHC Rules: Defines how the information of the EHC Context is used to compress a specific field. It defines compression functions, such as "elided", "least significant byte" and others, being applied on the header field.

EHC Strategy: Is applied to efficiently coordinate EHC Context and EHC Strategy. The EHC Strategy "Diet-ESP" defined in this document utilizes the information in the IPsec SA to pre-define the EHC Context without explicitly exchanging the EHC Context.

As depicted in Figure 1, the EHC Strategy - Diet-ESP in our case - and the EHC Context are agreed upon between the two peers, e.g. during key exchange. The EHC Rules are to be implemented on the peers and do not require further agreement.

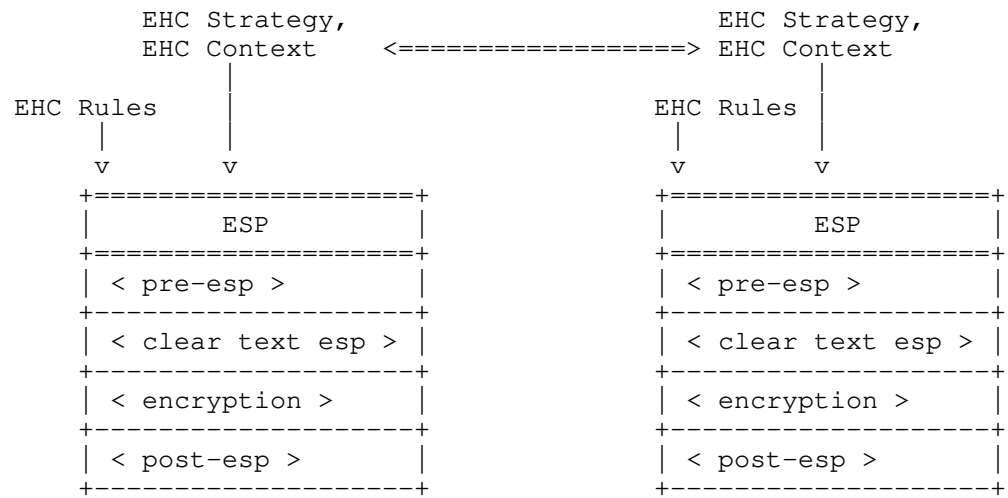


Figure 1: ESP Header Compression Overview

In Figure 1, the ESP stack is represented by various sub layers describing the packet processing inside the ESP:

- pre-esp: represents treatment performed to a non ESP packet, i.e. before ESP encapsulation or decapsulation is being performed. Any compression of protocols not specific to but encrypted by ESP, such as L4 and higher protocols, is performed here.
- clear text esp: designates the ESP encapsulation / decapsulation processing performed on an non encrypted ESP packet. This layer includes compression for fields which are included during the ESP encapsulation. A typical example is the later encrypted Tunnel IP header and the fields of the ESP trailer.
- encryption: designates the encryption/decryption phase This layer could include compression of encryption information (e.g. Initialization Vector, etc.), but this is currently out of scope of this document.
- post-esp the processing performed on an ESP encrypted packet. This layer includes compression of the ESP header.

EHC Rules may be processed at any of these layers and thus impact differently the standard ESP. More specifically, EHC Rules performed at the "pre-esp" or "post-esp" layer do not require the current ESP stack to be updated and can simply be appended to the current ESP stack. On the other hand, EHC Rules at the "clear text esp" may require modification of the current ESP stack.

The set of EHC rules described in this document as well as the EHC Strategies may be extended in the future. Nothing prevents such EHC Rules and Strategies to be updated.

5. IPsec Compression Mode

Signalling the compression of a certain ESP packet is crucial for correct decompression at the sender. Situation where decompression may fail unforeseen are various, such as IP fragmentation, UDP options [I-D.ietf-tsvwg-udp-options] just to name a few.

With EHC, the agreement of the level or occurrence of compression is left the negotiation protocol (e.g. IKEv2). In order to achieve per-packet signalization of the compression level, this document proposes new IPsec modes "Compressed Transport" and "Compressed Tunnel", which are meant to be agreed during the negotiation of the EHC Context and EHC Strategy. This can lead to multiple SAs, and thus, multiple SPIs for different levels of compression agreed with the EHC Context. The receiver can detect the level of compression of an incoming packet by looking up the used EHC Context and EHC strategy in the corresponding SA.

If the sender detects that the de-compression can not be guaranteed with a given EHC Context and EHC Strategy, it MUST NOT apply compression. If an SA with IPsec Mode "Tunnel"/"Transport" is available, the sender SHOULD send the packet uncompressed, rather than discard the packet. When there is no uncompressed SA available, the packet MUST be dropped.

6. EHC Context

The EHC Context provides the necessary information so the two peers can proceed to the appropriated compression and decompression defined by the EHC Strategy.

The EHC Context is defined on a per-SA basis. A context can be defined for any protocol encapsulated with ESP and for ESP itself. For each header field, a context attribute is provided to the EHC Context in order to allow compression and decompression. Most power of EHC lies in the fact, that the attributes for some protocols are already available in the IPsec SA (e.g. IP addresses in the Traffic Selector). Such attributes are designated by "Yes" in the "In SA" column. All others need to be negotiated separately in order to allow EHC to work properly.

As this document is limited to the Diet-ESP strategy, the EHC Context in this section used by the Diet-ESP Strategy to activate specific EHC Rules as well as to execute the EHC Rules by providing the necessary parameters..

6.1. EHC Context Parameters for ESP

Context Attribute	In SA	Possible Values
ipsec_mode	Yes	"Tunnel", "Transport"
outer_version	Yes	"IPv4", "IPv6"
esp_spi	Yes	ESP SPI
esp_spi_lsb	No	0, 1, 2, 3, 4
esp_sn	Yes	ESP Sequence Number
esp_sn_lsb	No	0, 1, 2, 3, 4
esp_sn_gen	No	"Time", "Incremental"
esp_align	No	8, 16, 24, 32
esp_encr	Yes	ESP Encryption Algorithm

Table 1

6.2. EHC Context Parameters for Inner IP

Parameters associated to the Inner IP addresses are only specified when the SA has been configured with the tunnel mode. As a result when ipsec_mode is set to "Transport" the parameters below MUST NOT be considered and are considered as "Undefined"

Context Attribute	In SA	Possible Values
ip_version	Yes	"IPv4", "IPv6"

Table 2

6.2.1. EHC Context Parameters for inner IPv6

Context Attribute	In SA	Possible Values
ip6_tcfl_comp	No	"Outer", "Value", "UnComp"
ip6_tc	No	IPv6 Traffic Class
ip6_fl	No	IPv6 Flow Label
ip6_hl_comp	No	"Outer", "Value", "UnComp"
ip6_hl	No	Hop Limit Value
ip6_src	Yes	IPv6 Source Address
ip6_dst	Yes	IPv6 Destination Address

Table 3

ip6_tcfl_comp indicates how Traffic Class and Flow Label fields of the inner IP Header are expected to be compressed. "Outer" indicates Traffic Class and Flow Label are read from the outer IP header, "Value" indicates these values are provided by the Diet-ESP Context, while "Uncompress" indicates that no compression occurs and these values are read in the inner IP inner header.

ip6_hl_comp indicates how Hop Limit field of the inner IP Header is expected to be compressed. (see ip6_tcfl_comp).

ip6_dst designates the Destination IPv6 Address of the inner IP header. The IP address is provided by the TS, and can be defined as a range of IP addresses. Compression is only considered when ip6_dst indicates a single IP Address. When the TS defines more than a single IP address ip6_dst is considered as "Unspecified" and its value MUST NOT be considered for compression.

6.2.2. EHC Context Parameters for inner IPv4

Context Attribute	In SA	Possible Values
ip4_options	No	"Options", "No_Options"
ip4_id	No	IPv4 Identification

ip4_id_lsb	No	0,1,2	
+-----+-----+-----+			
ip4_ttl_comp	No	"Outer", "Value", "UnComp"	
+-----+-----+-----+			
ip4_ttl	No	IPv4 Time To Live	
+-----+-----+-----+			
ip4_src	Yes	IPv4 Source Address	
+-----+-----+-----+			
ip4_dst	Yes	IPv4 Destination Address	
+-----+-----+-----+			
ip4_frag_enable	No	"True", "False"	
+-----+-----+-----+			

Table 4

ip4_options specifies if the IPv4 header contains any options. If set to "No_Options", the first 8 bit of the IPv4 header (being the IP version and IP header length) are compressed. If set to "Options" this bits are sent uncompressed.

ip4_ttl indicates how the Time To Live field of the inner IP Header is expected to be compressed. (see ip6_hl_comp).

6.3. EHC Context Parameters for Transport Protocol

The following parameters are provided by the SA but the SA may specify single value or a range of values. When the SA specifies a range of values, these parameters MUST NOT be considered and are considered as Unspecified.

Context Attribute	In SA	Possible Values
14_proto	Yes	IPv6/ESP Next Header, IPv4 Protocol
14_src	Yes	UDP/UDP-Lite/TCP Source Port
14_dst	Yes	UDP/UDP-Lite/TCP Destination Port

Table 5

6.3.1. EHC Context Parameters for UDP

For UDP, there are no additional parameters necessary than the ones in Section 6.3

6.3.2. EHC Context Parameters for UDP-Lite

Context Attribute	In SA	Possible Values
udplite_coverage	No	8-6535, "Length", "uncompressed"

Table 6

`udplite_coverage`: For UDP-Lite, the checksum can have different coverages, which is defined by the "Checksum Coverage" field which replaces the "Length" field of UDP. This context field defines the coverage in advance by either a specific value (8-16535), the actual length of the UDP-Lite payload ("Length" or 0) or as uncompressed. Note that `udplite_coverage` is indicated on a packet basis and cannot be greater than the UDP length. In this case `udplite_coverage` is negotiated for all packets and the actual coverage for a given UDP packet is derived as the minimum value between `udplite_coverage` and the length of the UDP packet.

6.3.3. EHC Context Parameters for TCP

Context Attribute	In SA	Possible Values
<code>tcp_sn</code>	No	TCP Sequence Number
<code>tcp_ack</code>	No	TCP Acknowledgment Number
<code>tcp_lsb</code>	No	0, 1, 2, 3, 4
<code>tcp_options</code>	No	"True", "False"
<code>tcp_urgent</code>	No	"True", "False"

Table 7

`tcp_sn` holds the current Sequence Number of the TCP session.

`tcp_ack` holds the current Acknowledgement Number of the TCP session.

`tcp_lsb` holds the number of lsb of `tcp_sn` and `tcp_ack` sent on the wire.

`tcp_options` says if options are enabled in the current TCP session. If `tcp_options` is set to "False" the Options field in TCP can be elided.

`tcp_urgent` says if the urgent pointer is enabled in the current TCP session. If `tcp_urgent` is set to "False" the Urgent Pointer field in TCP can be elided.

7. EHC Rules

This section describes the EHC Rules involved in Diet-ESP. The EHC Rules defined by Diet-ESP may be used in the future by EHC Strategies other than Diet-ESP, so they are described in an independent way.

A EHC Rule defines the compression and decompression of one or more fields and EHC Rules are represented this way:

EHC Rule	Field	Action	Parameters
EHC_RULE_NAME	f1	a1	p1_1, ... p1_n
	~	...	~
	fm	am	pm_1, ... pm_n

Figure 2: EHC Rules

The EHC Rule is designated by a name (`EHC_RULE_NAME`) and the concerned Fields (`f1, ..., fm`). Each field compression and decompression is represented by an Action (`a1, ..., am`). The Parameters indicate the necessary parameters for the action to perform both the compression and the decompression.

The table below provides a high level description of the Actions used by Diet-ESP. As these Action may take different arguments and may operate differently for each field a complete description is provided in the next sections as part of the EHC Rule description.

Function	Compression	Decompression
send-value	No	No
elided	Not send	Get from EHC Context
lsb(_lsb_size)	Sent LSB	Get from EHC Context
lower	Not send	Get from lower layer
checksum	Not send	Compute checksum.
padding(_align)	Compute padding	Get padding

Table 8

- a. send-value designates an action that does not perform any compression or decompression of a field.
- b. elided designates an action where both peers have a local value of the field. The compression of the field consists in removing the field, and the decompression consists in retrieving the field value from a known local value. The local value may be stored in a EHC Context or defined by the EHC Rule (like a zero value for example).
- c. lsb designates an action where both peers have a local value of the field, but the compression consists in sending only the LSB bytes instead of the whole field. The decompression consists in retrieving the field from the LSB sent as well as some other additional local values.
- d. lower designates an action where the compression consists in not sending the field. The decompression consists in retrieving the field from the lower layers of the packet. A typical example is when both IP and UDP carry the length of the payload, then the length of the UDP payload can be inferred from the one of the IP layer.
- e. checksum designates an action where the compression consists in not sending a checksum field. The decompression consists in re-computing the checksum. ESP provides an integrity-check based on signature of the ESP payload (ICV). This makes removing checksum possible, without harming the checksum mechanism.
- f. padding designates an action that computes the padding of the ESP packet. The function is specific to the ESP.

For all actions, the function can be performed only when the appropriated parameters and fields are provided. When a field or a parameters does not have an appropriated value its value is

designated as "Unspecified". Specifically some fields such as inner IP addresses, ports or transport protocols are agreed during the SA negotiation and are specified by the SA. Their value in the SA may take various values that are not appropriated to enable a compression. For example, when these fields are defined as a range of values, or by selectors such as OPAQUE or ANY these fields cannot be retrieved from a local value. Instead, when they are defined as a "Single" value (i.e a single IP address, or a single port number or a single transport protocol number) compression and decompression can be performed. These SA related fields are considered as "Unspecified" when not limited to a "Single" value.

When a field or a parameter is "Unspecified", the EHC Rule MUST NOT be activated. This is the purpose of the EHC Strategy to avoid ending in such case. In any case, when one of these condition is not met, the EHC Rule MUST NOT perform any compression or decompression action and the packet MUST be discarded. When possible, an error SHOULD be raised and logged.

7.1. EHC Rules for ESP

This section describes the EHC Rules for ESP which are summed up in the table below.

EHC Rule	Field	Action	Parameters
ESP_SPI	SPI	lsb	esp_spi_lsb, esp_spi
ESP_SN	Sequence Number	lsb	esp_sn_lsb, esp_sn_gen, esp_sn
ESP_NH	Next Header	elided	l4_proto, ipsec_mode
ESP_PAD	Pad Length, Padding	padding	esp_align, esp_encr

Table 9

ESP_SPI designates the EHC Rule compressing / decompressing the SPI. ESP_SPI is performed in the "post-esp" phase. The SPI is compressed using "lsb". The sending peer only places the LSB bytes of the SPI and the receiving peer retrieve the SPI from the LSB bytes carried in the packets as well as from the SPI value stored in the SA. The SPI MUST be retrieved as its full value is included in the signature check. The two peers MUST agree on the number of LSB bytes to be sent: "esp_spi_lsb". Upon agreeing on "esp_spi_lsb", the receiving peer MUST NOT agree on a value not carrying sufficient information to retrieve the full SPI.

ESP_SN designates the EHC Rule compressing / decompressing the ESP Sequence Number. ESP_SN is performed in the "post-esp" phase. ESP_SN is only activated if the SN ("esp_sn"), the LSB significant bytes ("esp_sn_lsb") and the method used to generate the SN ("esp_sn_gen") are defined. The Sequence Number is compressed using "lsb". Similarly to the SPI, the Sequence Number MUST be retrieved in order to complete the signature check of the ESP packet. Unlike the SPI, the Sequence Number is not agreed by the peers, but is changing for every packet. As a result, in order to retrieve the Sequence Number from the LSB "esp_sn_lsb", the peers MUST agree on generating Sequence Number in a similar way. This is negotiated with "esp_sn_gen" and the receiver MUST ensure that "esp_sn_lsb" is big enough to absorb minor packet losses or time differences between the peers.

ESP_NH designates the EHC Rule compressing / decompressing the ESP Next Header. ESP_NH is performed in the "clear text esp" phase. ESP_NH is only activated if the Next Header is specified. The Next Header can be specified as IP (IPv4 or IPv6) when the IPsec tunnel mode is used ("ipsec_mode" set to "Tunnel") or when the transport mode ("ipsec_mode" set to "Transport") is used when the Traffic Selector defines a "Single" Protocol ID ("l4_proto"). The Next Header, is compressed using "elided". The Next Header indicates the Header in the Payload Data. When the Tunnel mode is chosen, the type of the header is known to be an IP header. Similarly, the TS may also hold transport layer protocol, which specifies the Next Header value for Transport mode. The Next Header value is only there to provide sufficient information for decapsulating ESP. In other words decompressing this fields would occur in the "clear text esp" phase and striped but directly removed again by the ESP stack. For these reasons, implementation may simply omit decompressing this field.

ESP_PAD designates the EHC Rule compressing / decompressing the Pad Length and Padding fields. ESP_PAD is performed in the "clear text esp" phase. Pad Length and Padding define the padding. The purpose of padding is to respect a 32 bit alignment for ESP or block sizes of the used cryptographic suite. As the ESP trailer is encrypted,

Padding and Pad Length MUST to be performed by ESP and not by the encryption algorithm. Thus, ESP_PAD always needs to respect the cipher alignment ("esp_encr"), if applicable. Compression may be performed especially when device support alignment smaller than 32 bit. Such alignment is designated as "esp_align" and the padding bytes are the necessary bytes so the ESP packet has a length that is a multiple of "esp_align".

When "esp_align" is set to an 8-bit alignment padding bytes are not necessary, and Padding as well as Pad Length are removed. For values that are different from 8-bit alignment, padding bytes needs to be computed according to the ESP packet length why ESP_PAD MUST be the last action of "clear text esp". The resulting number of padding byte is then expressed in Padding and Pad Length fields with Pad Length set to padding bytes number - 1 and Padding is generated as described in [RFC4303].

Combining the Pad Length and Padding fields could potentially add an overhead on fixed size padding. In fact some applications may only send the same type of fixed size data, in which case the Pad Length would not be necessary to be specified. However, the only corner case Pad Length fields would actually add an overhead is when padding is expected to be of zero size. In this case, specifying an 8-bit alignment solve this issue.

7.2. EHC Rules for inner IPv4

All IPv4 EHC Rules MUST be performed during the "clear text esp" phase. The EHC Rules are only defined for compressing the inner IPv4 header and thus can only be used when the SA is using the Tunnel mode.

EHC Rule	Field	Action	Parameters
IP4_OPT_DIS	Version	elided	ip_version
	Header Length	elided	
IP4_LENGTH	Total Length	lower	
IP4_ID	Identification	lsb	ip4_id, ip4_id_lsb
IP4_FRAG_DIS	Flags	elided	
	Fragment Offset	elided	
IP4_TTL_OUTER	Time To Live	elided	ip4_ttl
IP4_TTL_VALUE	Time To Live	elided	ip4_ttl
IP4_PROT	Protocol	elided	l4_proto
IP4_CHECK	Header Checksum	checksum	
IP4_SRC	Source Address	elided	ip4_src
IP4_DST	Dest. Address	elided	ip4_dst

Table 10

IP4_OPT_DIS designates that the IPv4 header does not include any options and indicates if the first byte of the IPv4 header - consisting of IP version and IPv4 Header Length, are compressed. The Version "ip_version" is defined by the SA and is thus compressed using "elided". If the header does not contain any options, it is compressed with "elided" and decompressed to "20", the default length of the IPv4 header. If the header does contains some options, the length is not compressed.

IP4_LENGTH designates the EHC Rule compressing / decompressing the Total Length Field of the inner IPv4 header. The Total Length is compressed by the sender and not sent. The receiver decompresses it by recomputing the Total Length from the outer IP header. The outer IP header can be IPv4 or IPv6 and IP4_LENGTH MUST support both versions if both versions are supported by the device. Note that the length of the inner IP payload may also be subject to updates if decompression of the upper layers occurs.

IP4_ID designates the EHC Rule compressing / decompressing the Identification Field. IP4_ID is only activated if the ID ("ip4_id"), the LSB significant bytes ("ip4_id_lsb") are defined. Upon agreeing on "ip4_id_lsb", the receiving peer MUST NOT agree on a value not carrying sufficient information to retrieve the full IP Identification. Note also that unlike the ESP SN, the IPv4 Identification is not part of the SA. As a result, when the ID is compressed, its value MUST be stored in the EHC Context. The reserved attribute for that is "ip4_id"

IP4_FRAG_DIS designates that the inner IPv4 header does not support fragmentation. If activated, IP4_FRAG_DIS indicates compression of Flags and Fragment Offset field in the IPv4 header which consists of 2 bytes. Both fields are compressed with "elided" and decompressed with their default value according to [RFC0791], which is 0b010 for Flags and 0 for Fragment Offset.

IP4_TTL_OUTER designates an EHC Rule compressing / decompressing the Time To Live field of the inner IP header. If the outer IP header is an IPv6 header, the Hop Limit is used for decompression. The Time To Live field is compressed / decompressed using "lower", thus the field is not sent. The receiver decompresses it by reading its value from the outer IP header (TTL in case of IPv4 or HL in case of IPv6).

IP4_TTL_VALUE designates an EHC Rule compressing / decompressing the Time To Live field of the inner IP header. IP4_TTL_VALUE is only activated when the Hop Limit ("ip4_ttl") has been agreed. Time To Live is compressed / decompressed using the "elided" method.

IP4_PROTO designates the EHC Rule compressing / decompressing the Protocol field of the inner IPv4 header. IP4_PROTO is only activated if the Protocol is specified, that is when the Traffic Selectors defines a "Single" Protocol ID ("l4_proto"). When the Protocol ID identified by the SA has a "Single" value, the Protocol is compressed and decompressed using the "elided" method.

IP4_CHECK designates the EHC rule compressing / decompressing the Header Checksum field of the inner IPv4 header. The IPv4 header checksum is not sent by the sender and the receiver computes from the decompressed inner IPv4 header. IP4_CHECK MUST compute the checksum and not fill the checksum field with zeros. As a result, IP4_CHECK is the last decompressing EHC Rule to be performed on the decompressed IPv4 header.

IP4_SRC compresses the source IP address of the inner IPv4 header. IP4_SRC_IP is only be activated when the Traffic Selectors agreed by the SA defines a "Single" source IP address ("ip4_src"). The Source IP address is compressed / decompressed using the "elided" method.

IP4_DST works in a similar way as IP4_SRC_IP but for the destination IP address ("ip4_dst")

7.3. EHC Rules for inner IPv6

All IPv6 EHC Rules MUST be performed during the "clear text esp" phase. The EHC Rules are only defined for compressing the inner IPv6 header and thus can only be used when the SA is using the Tunnel mode.

EHC Rule	Field	Action	Parameters
IP6_OUTER	Version	elided	ip_version
	Traffic Class	lower	
	Flow Label	lower	
IP6_VALUE	Version	elided	ip_version
	Traffic Class	elided	ip6_tc
	Flow Label	elided	ip6_fl
IP6_LENGTH	Payload Length	lower	
IP6_NH	Next Header	elided	l4_proto
IP6_HL_OUTER	Hop Limit	lower	
IP6_HL_VALUE	Hop Limit	elided	ip6_hl
IP6_SRC	Source Address	elided	ip6_src
IP6_DST	Dest. Address	elided	ip6_dst

Table 11

IP6_OUTER designates an EHC Rule for compressing / decompressing the first 32 bits of the inner IPv6 header formed by the Version, Traffic Class and Flow Label. IP6_OUTER only proceeds to compression when both the outer and inner IP header are IPv6 header. When the outer IP header is an IPv4, the compression is bypassed. Bypassing enables to proceed to compression of IPv4 and IPv6 traffic in a VPN use case with a single SA. The Version "ip_version" is defined by the SA and is thus compressed using "elided". The other parameters Traffic

Class and Flow Label are compressed using "lower". More specifically, the fields are not sent. The receiver decompresses them by reading their value from the outer IPv6 header.

IP6_VALUE designates an EHC Rule for compressing / decompressing the first 32 bits of the inner IPv6 header formed by the Version, Traffic Class and Flow Label. IP6_VALUE is only activated if the Version of the inner IP header agreed by the SA is set to "Version 6" ("ip_version" set to "Version 6") and the specific values of the Traffic Class ("ip6_tc") and the Flow Label ("ip6_fl") are specified. With IP6_VALUE all fields are compressed and decompressed using "elided". Version is provided by the SA ("ip_version") while other fields are explicitly provided (ip6_tc, ip6_fl).

IP6_LENGTH designates the EHC Rule compressing / decompressing the Payload Length Field of the inner IPv6 header. The Payload Length is compressed by the sender and is not sent. The receiver decompress it by recomputing the Payload Length from the outer IP header. The IP header can be IPv4 or IPv6 and IP6_LENGTH MUST support both versions if both versions are supported by the device. Note that the length of the inner IP payload may also be subject to updates if decompression of the upper layers occurs.

IP6_NH designates the EHC Rule compressing / decompressing the Next Header field of the inner IPv6 header. IP6_NH is only activated if the Next Header is specified, that is when the Traffic Selectors defines a "Single" Protocol ID ("l4_proto"). When the Protocol ID identified by the SA has a "Single" value, the Next Header is compressed and decompressed using the "elided" method.

IP6_HL_OUTER designates an EHC Rule compressing / decompressing the Hop Limit field of the inner IP header. If the outer IP header is an IPv4 header, the Time To Live is used for decompression. The Hop Limit field is compressed / decompressed using the "lower". More specifically, the fields are not sent. The receiver decompresses them by reading their value from the outer IPv6 header.

IP6_HL_VALUE designates an EHC Rule compressing / decompressing the Hop Limit field of the inner IP header. IP6_HL_VALUE is only activated when the Hop Limit ("ip6_hl") has been agreed. The Hop Limit is compressed / decompressed using the "elided" method.

IP6_SRC compresses the source IP address of the inner IP header. IP6_SRC_IP is only be activated when the Traffic Selectors agreed by the SA defines a "Single" source IP address ("ip6_src"). The Source IP address is compressed / decompressed using the "elided" method.

IP6_DST works in a similar way as IP6_SRC_IP but for the destination IP address ("ip6_dst")

7.4. EHC Rules for UDP

All UDP EHC Rules MUST be performed during the "pre-esp" phase. The EHC Rules are only defined when the Traffic Selectors agreed during the SA negotiation results in "Single" Protocol ID ("l4_proto") which is set to UDP (17).

EHC Rule	Field	Action	Parameters
UDP_SRC	Source Port	elided	l4_source
UDP_DST	Dest. Port	elided	l4_dest
UDP_LENGTH	Length	lower	
UDP_CHECK	UDP Checksum	checksum	

Table 12

UDP_SRC designates the EHC Rule that compresses / decompresses the UDP Source Port. UDP_SRC is only activated when the Source Port agreed by the SA negotiation ("l4_src") is "Single". The Source Port is then compressed / decompressed using the "elided" method.

UDP_DST works in a similar way as UDP_SRC but for the Destination Port ("l4_dst").

UDP_LENGTH designates the EHC Rule compressing / decompressing the Length Field of the UDP header. The length is compressed by the sender and is not sent. The receiver decompresses it by recomputing the Length from the IP address header. The IP address can be IPv4 or IPv6 and UDP_LENGTH MUST support both versions if both versions are supported by the device.

UDP_CHECK designates the EHC Rule compressing / decompressing the UDP Checksum. The UDP Checksum is not sent by the sender and the receiver computes from the decompressed UDP payload. UDP_CHECK MUST compute the checksum and not fill the checksum field with zeros. As a result, UDP_CHECK is the last decompressing EHC Rule to be performed on the decompressed UDP Payload.

7.5. EHC Rules for UDP-Lite

All UDP-lite EHC Rules MUST be performed during the "pre-esp" phase. The EHC Rules are only defined when the Traffic Selectors agreed during the SA negotiation results in a "Single" Protocol ID ("l4_proto") which is set to UDPLite (136).

EHC Rule	Field	Action	Parameters
UDP-LITE_SRC	Source Port	elided	l4_source
UDP-LITE_DST	Dest. Port	elided	l4_dest
UDP-LITE_COVERAGE	Checksum Coverage	elided	udplite_coverage
UDP-LITE_CHECK	UDP-Lite Checksum	checksum	

Table 13

UDP-LITE_SRC works similarly to UDP_SRC

UDP-LITE_DST works similarly to UDP_DST

UDP-LITE_COVERAGE designates the EHC Rule compressing / decompressing the UDP-Lite Coverage field. UDP-LITE_COVERAGE is only activated when the Coverage ("udplite_coverage") has been agreed with a valid value. The Coverage is compressed / decompressed using the "elided" method.

UDP-LITE_CHECK designates the EHC Rule compressing / decompressing the UDP-Lite checksum. UDP-LITE_CHECK is only activated if the Coverage is defined either elided or sent. UDP-LITE_CHECK computes the checksum using "checksum" according to the uncompressed UDP packet and the value of the Coverage.

7.6. EHC Rules for TCP

All TCP EHC Rules MUST be performed during the "pre-esp" phase. The EHC Rules are only defined when the Traffic Selectors agreed during the SA negotiation results in a "Single" Protocol ID ("l4_proto") which is set to TCP (6).

EHC Rule	Field	Action	Parameters
TCP_SRC	Source Port	elided	l4_source
TCP_DST	Dest. Port	elided	l4_dest
TCP_SN	Sequence Number	lsb	tcp_sn, tcp_lsb
TCP_ACK	Acknowledgment Number	lsb	tcp_ack, tcp_lsb
TCP_OPTIONS	Data Offset	elided	tcp_options
	Reserved Bits	elided	
TCP_CHECK	TCP Checksum	checksum	
TCP_URGENT	TCP Urgent Field	elided	tcp_urgent

Table 14

TCP_SRC works similarly to UDP_SRC.

TCP_DST works similarly to UDP_DST.

TCP_SN designates the EHC Rule compressing / decompressing the TCP Sequence Number. TCP_SN is only activated if the SN ("tcp_sn") and the LSB significant bytes ("tcp_lsb") are defined. The TCP SN is compressed using "lsb". The sending peer only places the LSB bytes of the TCP SN ("tcp_sn") and the receiving peer retrieve the TCP SN from the LSB bytes carried in the packets as well as from the TCP SN value stored in EHC Context ("tcp_sn"). The two peers MUST agree on the number of LSB bytes to be sent: "tcp_lsb". Upon agreeing on "tcp_lsb", the receiving peer MUST NOT agree on a value not carrying sufficient information to retrieve the full TCP SN. Note also that unlike the ESP SN, the TCP SN is not part of the SA. As a result, when the SN is compressed, the value of the TCP SN MUST be stored in the EHC Context. The reserved attribute for that is "tcp_sn"

TCP_ACK designates the EHC Rule compressing / decompressing the TCP Acknowledgment Number and works similarly to TCP SN. Note that "tcp_lsb" is agreed for both TCP SN and TCP Acknowledgment. Similarly the value of the complete TCP Acknowledgment Number MUST be stored in the "tcp_ack" attribute of the EHC Context.

TCP_OPTIONS designates the EHC Rule compressing / decompressing TCP options related fields such as Data Offset and Reserved Bits. TCP_OPTION can only be activated when the TCP Option ("tcp_options") is defined. When "tcp_options" is set to "False" and indicates there are no TCP Options, the Data Offsets and Reserved Bits are compressed / decompressed using the "elided" method with Data Offset and Reserved Bits set to zero.

TCP_CHECK designates the EHC Rule compressing / decompressing the TCP Checksum. TCP_CHECK works similarly as UDP_CHECK.

TCP_URGENT designates the EHC Rule compressing / decompressing the urgent related information. When "tcp_urgent" is set to "False" and indicates there are no TCP Urgent related information, the Urgent Pointer is then "elided" and filled with zeros.

8. Diet-ESP EHC Strategy

From the attributes of the EHC Context, Diet-ESP defined as an EHC Strategy, which EHC Rules to apply. The EHC Strategy is defined for outbound packets which compresses the packet as well as for inbound packet where the decompression occurs.

Diet-ESP results from a compromise between compression efficiency, ease to configure Diet-ESP and the various use cases considered. In order to achieve a great simplicity,

- * Diet-ESP favors compression methods that required fewer configuration: For IPv6, ip6_tcfl_comp and ip6_hl_com to "Outer" so that ip6_tc, ip6_fl and ip6_hl can be derived from the packet. Similarly, ip4_ttl_comp has is set to "Outer" so ip4_tll can be derived from the packet.
- * Diet-ESP limits compression method to those foreseen as the most commonly used. As such, esp_sn_gen has been set to "Incremental" as this is the most common method used to generate SN. The other method would be "Time".
- * Diet-ESP limits compression to the most foreseen scenarios. IPv4 compression has been limited in favor of IPv6 as constraint devices have largely adopted IPv6, and the gain versus the complexity to deploy IPv4 inner IP addresses has not been proved. As a result some compressions for IPv4 are not considered by DIet-ESP. This involved compression of the IPv4 options by setting ip4_options to "No_Options". Similarly IPv4 ID compression has not been enabled by setting ip4_id and ip4_id_lsb to "Unspecified".
- * Diet-ESP negotiated values shared by different rules such as tcp_lsb which is shared for TCP ACK as well as for the TCP SN.

- * Diet-ESP defines a logic to set the necessary parameters from those agreed by the standard ESP agreement, which limits the setting of parameters.

The following tables shows, which EHC Rules are activated by default for the supported protocols ESP, IPv4, IPv6, UDP, UDP-Lite and TCP when using the Diet-ESP strategy and which ones are activated due to certain circumstances or explicit negotiation

ESP:

EHC Rule	Activated if	Parameter	Value
ESP_SPI	Diet-ESP	esp_spi_lsb	Negotiated
		esp_spi	In SA
ESP_SN	Diet-ESP	esp_sn_lsb	Negotiated
		esp_sn_gen	Negotiated
		esp_sn	In SA
ESP_NH	Diet-ESP	ipsec_mode	In SA
		l4_proto	In SA
ESP_PAD	Diet-ESP	esp_align	Negotiated
		esp_encr	In SA

Table 15

IPv4:

EHC Rule	Activated if	Parameter	Value
IP4_OPT_DIS	ip_version==4	ip_version	In SA
IP4_LENGTH	ip_version==4	None	
IP4_FRAG_DIS	ip_version==4	None	
IP4_TTL_OUTER	ip_version==4	None	

IP4_TTL_OUTER	ip_version==4	l4_proto	In SA
IP4_CHECK	ip_version==4	None	
IP4_SRC	ip_version==4	ip4_src	In SA
IP4_DST	ip_version==4	ip4_dst	In SA

Table 16

IPv6:

EHC Rule	Activated if	Parameter	Value
IP6_OUTER	ip_version==6	ip_version	In SA
IP6_LENGTH	ip_version==6	None	
IP6_NH	ip_version==6	l4_proto	In SA
IP6_HL_OUTER	ip_version==6	None	
IP6_SRC	ip_version==6	ip6_src	In SA
IP6_DST	ip_version==6	ip6_dst	In SA

Table 17

UDP:

EHC Rule	Activated if	Parameter	Value
UDP_SRC	l4_proto==17	l4_source	In SA
UDP_DST	l4_proto==17	l4_dest	In SA
UDP_LENGTH	l4_proto==17	None	
UDP_CHECK	l4_proto==17	None	

Table 18

UDP-Lite:

EHC Rule	Activated if	Parameter	Value
UDP_LITE_SRC	14_proto==136	14_source	In SA
UDP_LITE_DST	14_proto==136	14_dest	In SA
UDP_LITE_COVERAGE	14_proto==136	udplite_coverage	Negotiated
UDP_LITE_CHECK	14_proto==136	None	

Table 19

TCP:

EHC Rule	Activated if	Parameter	Value
TCP_SRC	14_proto==6	14_source	In SA
TCP_DST	14_proto==6	14_dest	In SA
TCP_SN	14_proto==6	tcp_sn	In SA
		tcp_lsb	Negotiated
TCP_ACK	14_proto==6	tcp_ack	In SA
		tcp_lsb	Negotiated
TCP_OPTIONS	14_proto==6	tcp_options	Negotiated
TCP_CHECK	14_proto==6	None	
TCP_URGENT	14_proto==6	tcp_urgent	Negotiated

Table 20

Thus, the parameters that the two peers needs to agree on are:

- * esp_sn_lsb
- * esp_spi_lsb
- * esp_align
- * udplite_coverage
- * tcp_lsb
- * tcp_options

* tcp_urgent

Implementation may differ from the description below. However, the outcome MUST remain the same.

8.1. Outbound Packet Processing

Diet-ESP compression is defined as follows:

1. In phase "pre-esp": Match the inbound packet with the SA and determine if the Diet-ESP EHC Strategy has been activated. If the Diet-ESP EHC Strategy has been activated proceed to next step, otherwise skip all steps associated to Diet-ESP and proceed to the standard ESP as defined in [RFC4303]
2. In phase "pre-esp": If "l4_proto" designates a "Single" Protocol ID (UDP, TCP or UDP-Lite), proceed to the compression of the specific layer. Otherwise, the transport layer is not compressed.
3. In phase "clear text esp": If "ipsec_mode" is set to "Tunnel" mode, determine "ip_version" the IP version of the inner IP addresses and proceed to the appropriated inner IP address compression.
4. In phase "clear text esp" and "post-esp": Proceed to the ESP compression.

UDP compression is defined as below:

1. If "l4_src" designates a "Single" Source Port, apply UDP_SRC to compress the Source Port.
2. If "l4_dst" designates a "Single" Destination Port, apply UDP_DST to compress the Destination Port.
3. Apply UDP_CHECK to compress the Checksum.
4. Apply UDP_LENGTH to compress the Length.

UDP-lite compression is defined as below:

1. If "l4_src" designates a "Single" Source Port, apply the UDP-LITE_SRC to compress the Source Port.
2. If "l4_dst" designates a "Single" Destination Port, apply the UDP-LITE_DST, to compress the Destination Port.
3. If "udplite_coverage" is specified, apply the UDP-LITE_COVERAGE, to compress the Coverage.
4. Apply UDP-LITE_CHECK to compress the Checksum.

TCP compression is defined as below:

1. If "l4_src" designates a "Single" Source Port than apply the TCP_SRC to compress the Source Port.

2. If "l4_dst" designates a "Single" Destination Port than apply the TCP_DST to compress the Destination Port.
3. If "tcp_lsb" is lower than 4, then "tcp_sn" "tcp_ack" attributes of the Diet-ESP Context are updated with the value provided from the packet before applying the TCP_SN and the TCP_ACK EHC Rules.
4. If "tcp_options" is set to "False" apply the TCP_OPTIONS EHC Rule.
5. If "tcp_urgent" is set to "False" apply the TCP_URGENT EHC Rule.
6. Apply TCP_CHECK to compress the Checksum.

Inner IPv6 Header compression is defined as below:

1. If "ip6_src" designates a "Single" Source IP address, apply the IP6_SRC to compress the IPv6 Source Address.
2. If "ip6_dst" designates a "Single" Destination IP address, apply the IP6_DST to decompress the IPv6 Destination Address.
3. Apply IP6_HL_OUTER to compress the Hop Limit.
4. If "l4_proto" designates a "Single" Protocol ID (UDP, TCP or UDP-Lite), apply IP6_NH to compress the Next Header.
5. Apply, IP6_LENGTH to compress the Length.
6. Apply IP6_OUTER to compress Version, Traffic Class and Flow Label.

Inner IPv4 Header compression is defined as below:

1. Apply, IP4_LENGTH to compress the Length.
2. Apply IP4_TTL_OUTER to compress Time To Live.
3. Apply, IP4_CHECK to compress the IPv4 header checksum.
4. If "ip4_src" designates a "Single" Source IP address, apply the IP4_SRC to compress the IPv4 Source Address.
5. If "ip4_dst" designates a "Single" Destination IP address, apply the IP4_DST to decompress the IPv4 Destination Address.

ESP compression is defined as below:

1. In phase "clear text esp": If "ipsec_mode" is set to "Tunnel" or "l4_proto" is set to a "Single value - eventually different from TCP, UDP or UDP-Lite, apply ESP_NH, to compress the Next Header.
2. In phase "clear text esp": If "esp_encr" specify an encryption algorithm that does not provide padding, then apply ESP_PAD to compress the Pad Length and Padding.
3. Proceed to the ESP encryption as defined in [RFC4303].
4. In phase "post-esp": If "esp_sn_lsb" is different from 4, then apply ESP_SN. To compress the ESP SN.
5. In phase "post-esp": If "esp_spi_lsb" is different from 4, then apply ESP_SPI to compress the SPI.

8.2. Inbound Packet Processing

Diet-ESP decompression is defined as follows:

1. Match the inbound packet with the SA and determine if the Diet-ESP EHC Strategy has been activated. When Diet-ESP is activated this means that the "esp_spi_lsb" are sufficient to index the SA and proceed to next step, otherwise skip all steps associated to Diet-ESP and proceed to the standard ESP as defined in [RFC4303]
2. In phase "clear text esp" and "post-esp": Proceed to the ESP decompression.
3. In phase "clear text esp": If "ipsec_mode" is set to "Tunnel" mode, determine "ip_version" the IP version of the inner IP addresses and proceed to the appropriated inner IP address decompression, except for the computation of the checksums and length.
4. In phase "pre-esp": If "l4_proto" designates a "Single" Protocol ID (UDP, TCP or UDP-Lite), proceed to the decompression of the specific layer, except for the computation of the checksums and length replaced by zero fields.
5. In phase "pre-esp": Proceed to the decompression of the checksums and length.

ESP decompression is defined as follows:

1. In phase "post-esp": If "esp_spi_lsb" is different from 4, then apply ESP_SPI to decompress the SPI.
2. In phase "post-esp": If "esp_sn_lsb" is different from 4, then apply ESP_SN. To decompress the ESP SN.
3. Proceed to the ESP signature validation and decryption as defined in [RFC4303].
4. In phase "clear text esp": If "ipsec_mode" is set to "Tunnel" or "l4_proto" is set to a "Single" value - eventually different from TCP, UDP or UDP-Lite, apply ESP_NH, to decompress the Next Header.
5. In phase "clear text esp": If "esp_encr" specify an encryption algorithm that does not provide padding, then apply ESP_PAD to compress the Pad Length and Padding.
6. Extract the ESP Data Payload and apply decompression EHC Rule to the ESP Data Payload.

UDP decompression is defined as follows:

1. If "l4_src" designates a "Single" Source Port, apply UDP_SRC to decompress the Source Port.
2. If "l4_dst" designates a "Single" Destination Port, apply UDP_DST to decompress the Destination Port.

3. Apply UDP_LENGTH to compress the Length. The length value is computed from the length provided by the lower layer, with the additional added bytes during the UDP decompression including the length size.
4. Apply UDP_CHECK to decompress the Checksum.
5. Update the Length of the lower layers:
 1. If "ipsec_mode" is set to "Transport" mode, update the Length of the outer IP header (IPv4 or IPv6). The Length is incremented by the number of bytes generated by the decompression of the transport layer.
 2. If "ipsec_mode" is set to "Tunnel" mode, update the Length of the inner IP address (IPv4 or IPv6) as well as the outer IP header (IPv4 or IPv6). The Length is incremented by the number of bytes generated by the decompression of the transport layer.

UDP-Lite decompression is defined as follows:

1. If "l4_src" designates a "Single" Source Port, apply the UDP-LITE_SRC to decompress the Source Port.
2. If "l4_dst" designates a "Single" Destination Port, apply the UDP-LITE_DST, to decompress the Destination Port.
3. If "udplite_coverage" is specified, apply the UDP-LITE_COVERAGE, to decompress the Coverage.
4. Apply UDP-LITE_CHECK to compress the Checksum.
5. Update the Length of the lower layers as defined in UDP.

TCP decompression is defined as follows:

1. If "l4_src" designates a "Single" Source Port than apply the TCP_SRC to decompress the Source Port.
2. If "l4_dst" designates a "Single" Destination Port than apply the TCP_DST to decompress the Destination Port.
3. If "tcp_lsb" is lower than 4, apply TCP_SN and the TCP_ACK to decompress the TCP Sequence Number and the TCP Acknowledgment Number.
4. If "tcp_options" is set to "False" apply TCP_OPTIONS to decompress Data Offset and Reserved Bits.
5. If "tcp_urgent" is set to "False" apply the TCP_URGENT to decompress the Urgent Pointer.
6. Apply TCP_CHECK to decompress the Checksum.

Inner IPv6 decompression is defined as follows:

1. Apply IP6_OUTER to decompress Version, Traffic Class and Flow Label.
2. Set the Length to zero.

3. If "l4_proto" designates a "Single" Protocol ID (UDP, TCP or UDP-Lite), apply IP6_NH to decompress the Next Header.
4. Hop Limit is decompressed with IP6_HL_OUTER (with "ip6_hl_comp" set to "Outer").
5. If the "ip6_src" designates a "Single" Source IP address, apply the IP6_SRC to decompress the IPv6 Source Address.
6. If the "ip6_dst" designates a "Single" Destination IP address then apply the IP6_DST to decompress the IPv6 Destination Address.
7. Apply, IP6_LENGTH to provide the replace the zero length value by its appropriated value. The Length value considers the length provided by the lower layers to which are added the additional bytes due to the decompression, minus the length of the inner IP6 Header.

Inner IPv4 decompression is defined as follows:

1. Apply, IP4_LENGTH to provide the replace the zero length value by its appropriated value. The Length value considers the length provided by the lower layers to which are added the additional bytes due to the decompression, minus the length of the inner IPv4 Header. The value computed from the lower layer will have to be overwritten in case further decompression occurs.
2. Apply IP4_TTL_OUTER to decompress Time To Live.
3. If "l4_proto" designates a "Single" Protocol ID (UDP, TCP or UDP-Lite), apply IP4_PROT to decompress the Protocol Field.
4. If "ip4_src" designates a "Single" Source IP address, apply the IP4_SRC to decompress the IPv4 Source Address.
5. If "ip4_dst" designates a "Single" Destination IP address then apply the IP4_DST to decompress the IPv4 Destination Address.
6. Apply IP4_CHECK to decompress the checksum of the IPv4 header

9. IANA Considerations

There are no IANA consideration for this document.

10. 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 EHC, the value of the SPI is not reduced, but compressed why the SPI value may not be fully provided between the compressor and the de-

compressor. On the other hand, its uncompressed value is provided to the ESP-processing 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 an 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.

Sequence Number (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 EHC 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 de-synchronized 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. If TIME is used as the method to generate the SN, the receiver MUST ensure that the `esp_sn_lsb` is big enough to resist time differences between the nodes. 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.

11. Privacy Considerations

Security Parameter Index (SPI):

Until Diet-ESP is 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 TIME 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. Another possibility may also consists in compressing the SN to the low order bytes to reduce the information related to the age or the number of packets being exchanged.

12. Acknowledgment

We would like to thank Orange and Universitee Pierre et Marie Curie for initiating the work on Diet-ESP. We Would like to 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 would like to thank Bob Moskowitz for not copyrighting Diet HIP. The "Diet" terminology is from him.

We would like to thank those we received many useful feed backs among others: Dominique Bartel, Anna Minaburo, Suresh Krishnan, Samita Chakrabarti, Michael Richarson, Tero Kivinen.

13. References

13.1. Normative References

- [RFC0791] Postel, J., "Internet Protocol", STD 5, RFC 791, DOI 10.17487/RFC0791, September 1981, <<https://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, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC4303] Kent, S., "IP Encapsulating Security Payload (ESP)", RFC 4303, DOI 10.17487/RFC4303, December 2005, <<https://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, <<https://www.rfc-editor.org/info/rfc4309>>.
- [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>>.
- [RFC5795] Sandlund, K., Pelletier, G., and L-E. Jonsson, "The RObust Header Compression (ROHC) Framework", RFC 5795, DOI 10.17487/RFC5795, March 2010, <<https://www.rfc-editor.org/info/rfc5795>>.
- [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, <<https://www.rfc-editor.org/info/rfc5858>>.
- [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>>.

13.2. Informational References

- [RFC8724] Minaburo, A., Toutain, L., Gomez, C., Barthel, D., and JC. Zuniga, "SCHC: Generic Framework for Static Context Header Compression and Fragmentation", RFC 8724, DOI 10.17487/RFC8724, April 2020, <<https://www.rfc-editor.org/info/rfc8724>>.
- [RFC8750] Migault, D., Guggemos, T., and Y. Nir, "Implicit Initialization Vector (IV) for Counter-Based Ciphers in Encapsulating Security Payload (ESP)", RFC 8750, DOI 10.17487/RFC8750, March 2020, <<https://www.rfc-editor.org/info/rfc8750>>.
- [I-D.ietf-tsvwg-udp-options] Touch, J., "Transport Options for UDP", Work in Progress, Internet-Draft, draft-ietf-tsvwg-udp-options-18, 26 March 2022, <<https://www.ietf.org/archive/id/draft-ietf-tsvwg-udp-options-18.txt>>.
- [fit-iot-lab] "Future Internet of Things (FIT) IoT-LAB", <<https://www.ietf-lab.info>>.

Appendix A. Illustrative Examples

A.1. Single UDP Session IoT VPN

This section considers a IoT IPv6 probe hosting a UDP application. The probe is dedicated to a single application and establishes a single UDP session. As a result, inner IP addresses and UDP Ports have a "Single" value and can be easily compressed. The probes sets an IPsec VPN using IPv6 addresses in order to connect its secure domain - typically a Home Gateway. The use of IPv6 for inner and outer IP addresses, enables to infer inner IP fields from the outer IP address. The probes encrypts with AES-CCM_8 [RFC4309]. AES-CCM does not have padding, so the padding is performed by ESP. The probes uses an 8 bit alignment which enables to fully compress the ESP Trailer. In addition, as the probe SA is indexed using the outer IP addresses (or eventually the radio identifiers) which enables to fully compress the SPI. As the probe provides information every hour, the Sequence Number using time can be derived from the received time, which enables to fully compress the SN.

Figure 3 represents the original UDP packet and Figure 4 represents the corresponding packet compressed with Diet-ESP. The compression with Diet-ESP results in a reduction of 61 bytes overhead. With IPv4 inner IP addressed Diet-ESP results in an 45 byte overhead reduction.

Further compression may be done for example by using an implicit IV [RFC8750] and by compressing the outer IP addresses (not represented on the figure). In addition, application data may also be compressed with mechanisms outside of the scope of Diet-ESP.

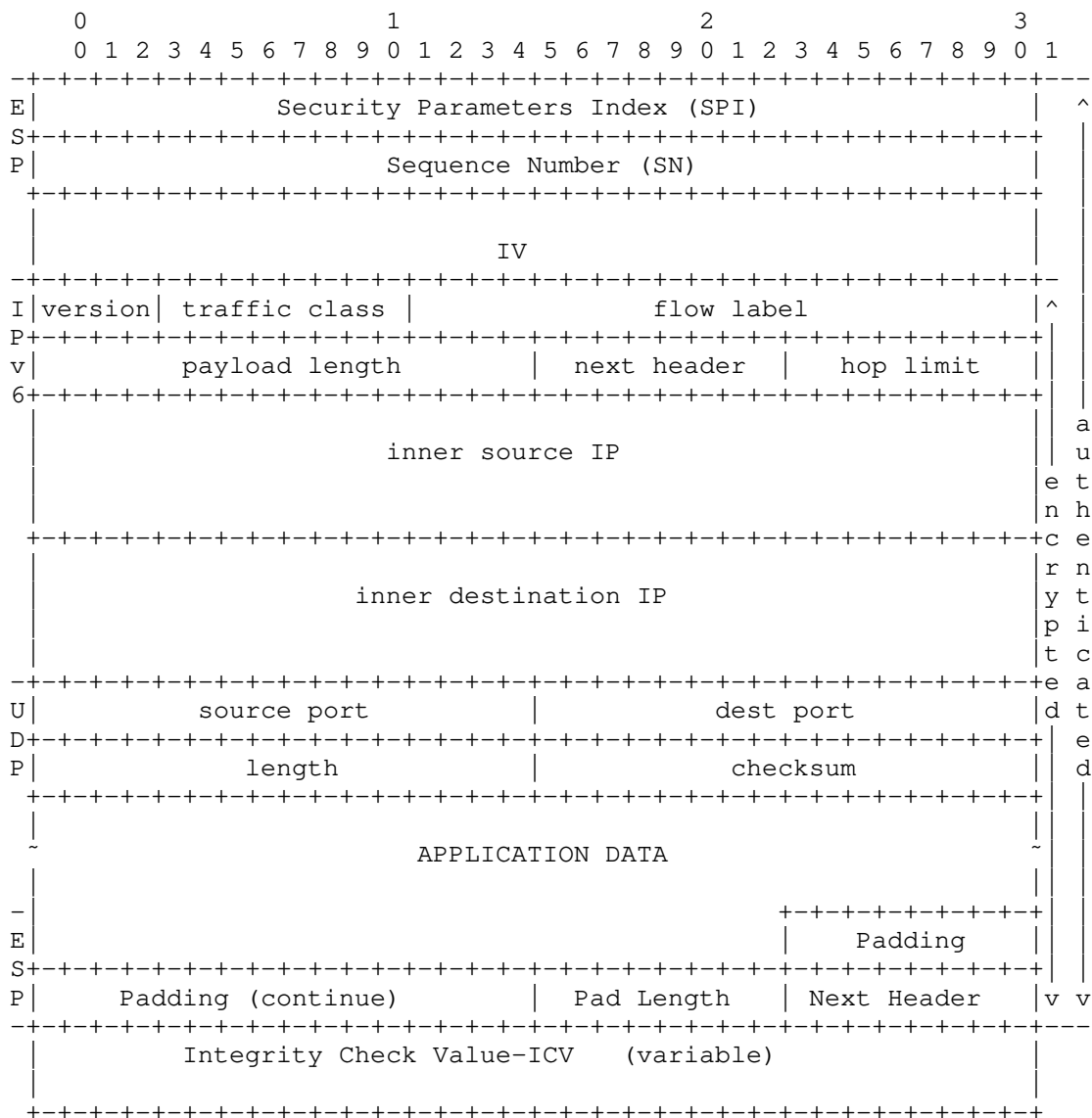


Figure 3: Standard ESP VPN Packet Description

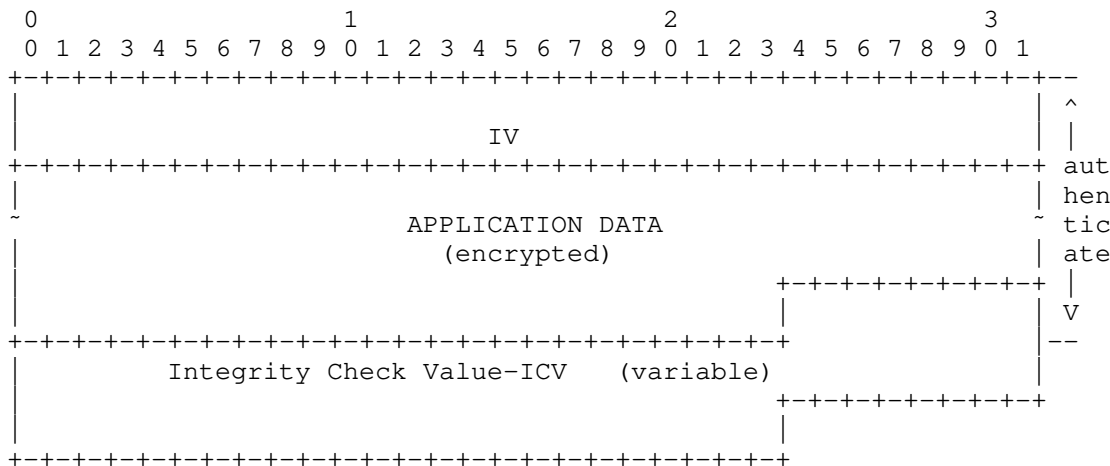


Figure 4: Diet-ESP Single UDP Session IoT VPN Packet Description

The following table illustrates the activated rules and the attributes of the Diet-ESP Context that needs an explicit agreement to achieve the compression. All other attributes used by the rules are part of the SA agreement. Parameters of not activated rules are left "Unspecified".

EHC Rule	Context Attribute	Value
ESP_SPI	esp_spi_lsb	0
ESP_SN	esp_sn_lsb	0
	esp_sn_gen	
ESP_NH		
ESP_PAD	esp_align	8
IP6_OUTER	ip6_tcfl_comp	
	ip6_hl_comp	
IP6_LENGTH		
IP6_NH		
IP6_HL_OUTER		
IP6_SRC		
IP6_DST		
UDP_SRC		
UDP_DST		
UDP_LENGTH		
UDP_CHECK		

Table 21

A.2. Single TCP session IoT VPN

This section considers the same probe as described in Appendix A.1 but instead of using UDP as a transport layer, the probe uses TCP. In this case TCP is used with no options, no urgent pointers and the SN and ACK Number are compressed to 2 bytes as the throughput is expected to be low.

Figure 5 represents the original TCP packet and Figure 6 represents the corresponding packet compressed with Diet-ESP. The compression with Diet-ESP results in a reduction of 66 bytes overhead. With IPv4 inner address Diet-ESP results in a 50 byte overhead reduction.

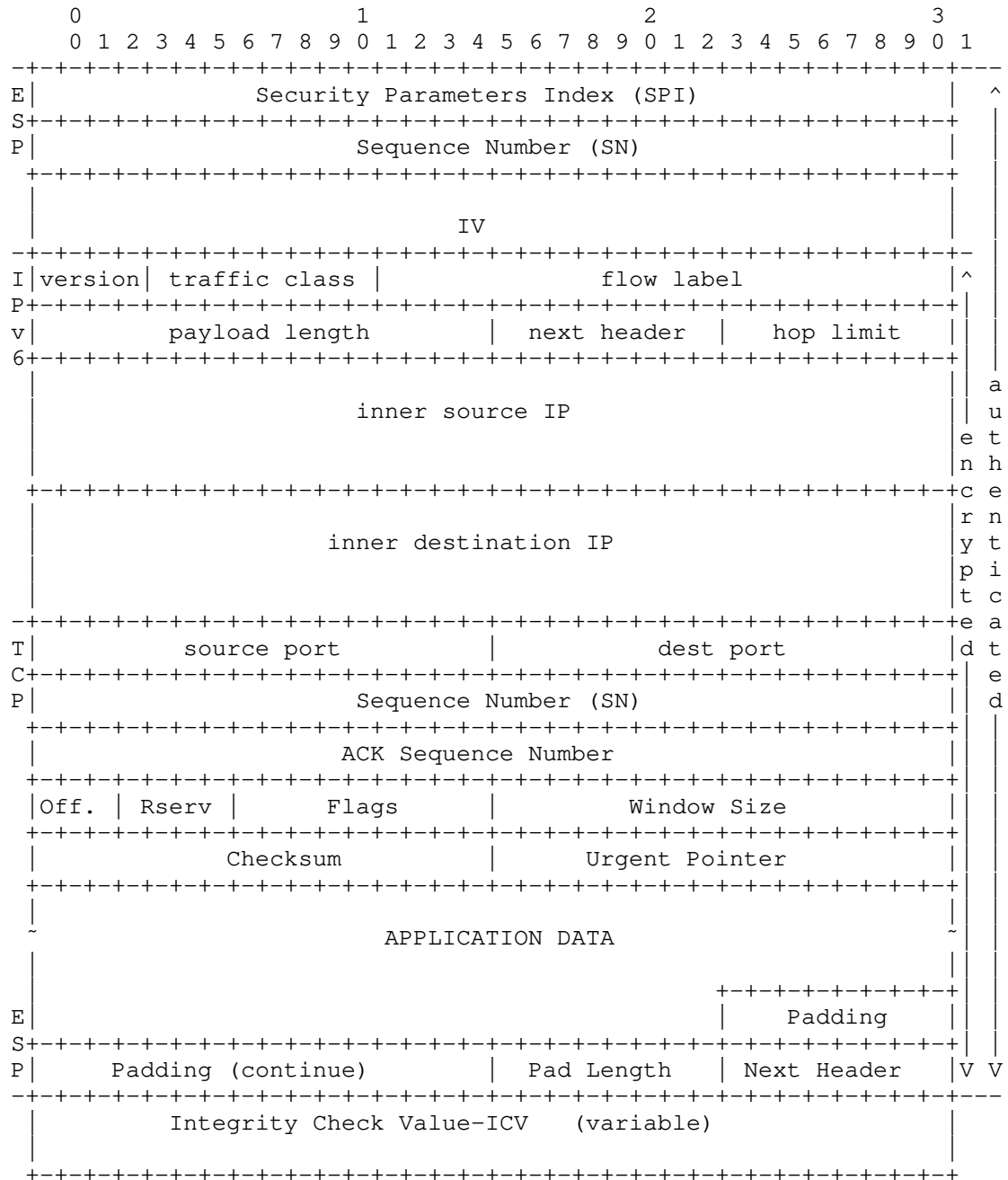


Figure 5: Standard IoT Single TCP Session VPN Packet Description

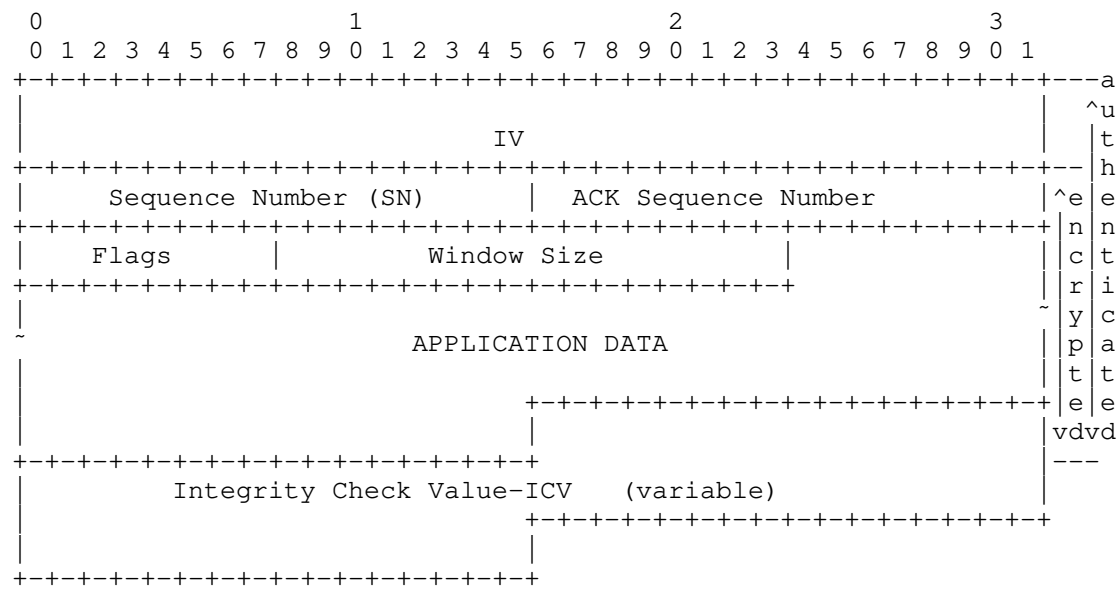


Figure 6: Diet-ESP Single TCP Session IoT VPN Packet Description

The following table illustrates the activated rules and the attributes of the Diet-ESP Context that needs an explicit agreement to achieve the compression. All other attributes used by the rules are part of the SA agreement. Parameters of not activated rules are left "Unspecified". Note for simplicity, tcp_sn and tcp_ack are negotiated to start with 0, but it could be any other value as well.

EHC Rule	Context Attribute	Value
ESP_SPI	esp_spi_lsb	0
ESP_SN	esp_sn_lsb	0
	esp_sn_gen	
ESP_NH		
ESP_PAD	esp_align	8
IP6_OUTER	ip6_tcfl_comp	
	ip6_hl_comp	

IP6_LENGTH		
IP6_NH		
IP6_HL_OUTER		
IP6_SRC		
IP6_DST		
TCP_SRC		
TCP_DST		
TCP_SN	tcp_lsb	2
	tcp_sn	0
TCP_ACK	tcp_lsb	2
	tcp_ack	0
TCP_OPTIONS	tcp_options	"False"
TCP_CHECK		
TCP_URGENT	tcp_urgent	"False"

Table 22

A.3. Traditional VPN

This section illustrates the case of an company VPN. The VPN is typically set by a remote host that forwards all its traffic to the security gateway. As transport protocols are "Unspecified", compression is limited to ESP and the inner IP header. For the inner IP header, the Destination IP address is "Unspecified" so the compression of the inner IP address excludes the Destination IP address. Similarly, the inner IP Next Header cannot be compressed as the transport layer is not specified. For ESP, the security gateway may only have a sufficiently low number of remote users with relatively low throughput in which case SPI and SN can be compressed to 2 bytes. As throughput remains relatively low, the alignment may also set to 8 bits.

A.3.1. IPv6 in IPv6

Figure 7 represents the original TCP packet with IPv6 inner IP addresses and Figure 8 represents the corresponding packet compressed with Diet-ESP. The compression with Diet-ESP results in a reduction of 32 bytes.

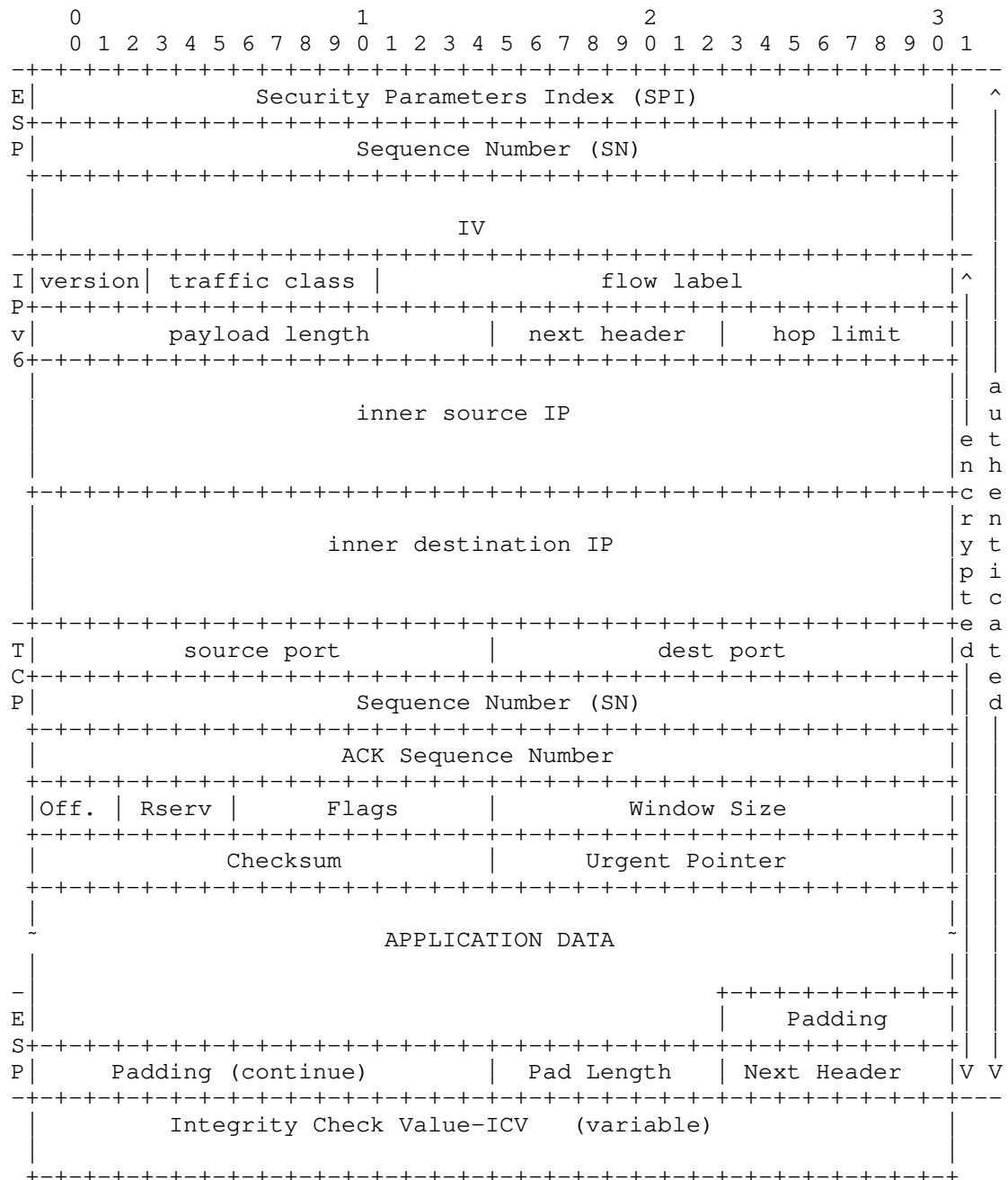


Figure 7: Standard ESP VPN Packet Description

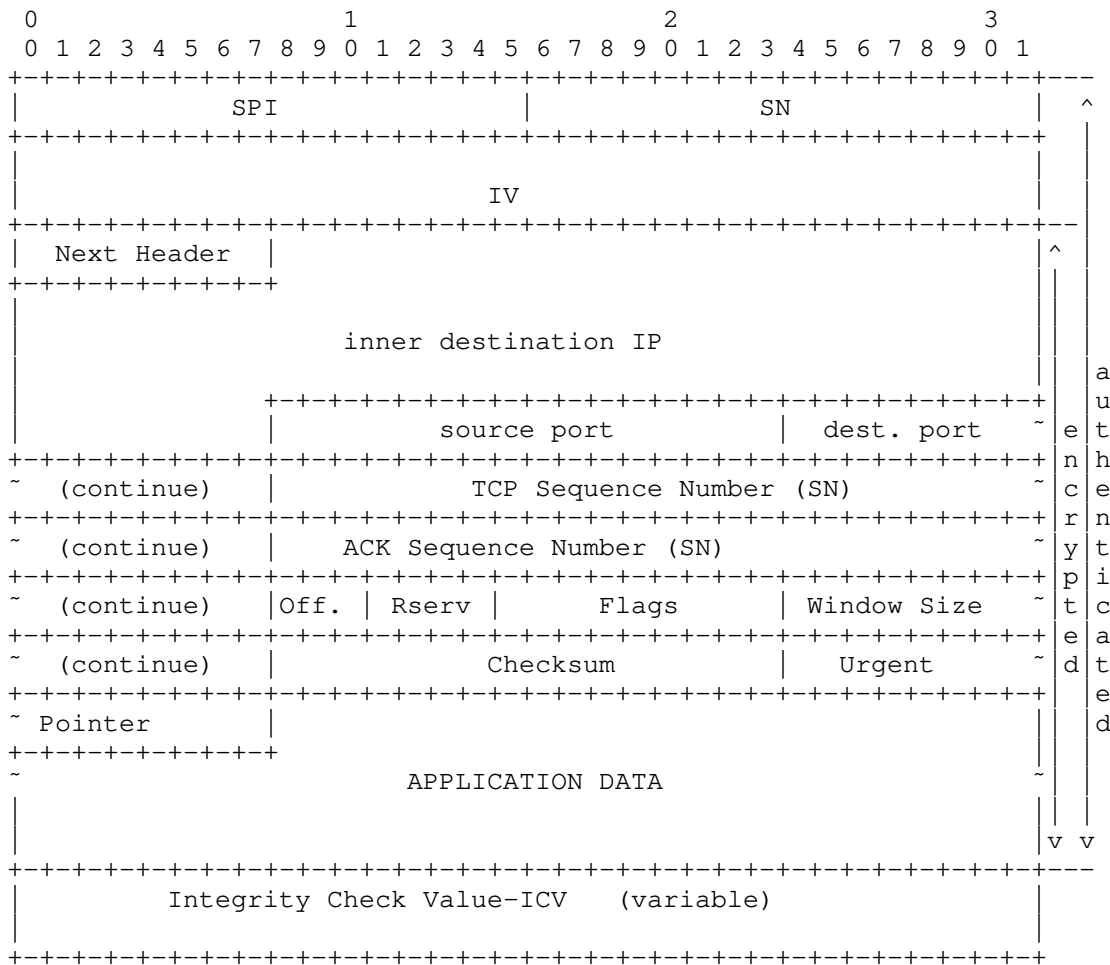


Figure 8: Diet-ESP VPN Packet Description

The following table illustrates the activated rules and the attributes of the Diet-ESP Context that needs an explicit agreement to achieve the compression. All other attributes used by the rules are part of the SA agreement. Parameters of not activated rules are left "Unspecified".

EHC Rule	Context Attribute	Value
ESP_SPI	esp_spi_lsb	2
ESP_SN	esp_sn_lsb	2
	esp_sn_gen	
ESP_NH		
ESP_PAD	esp_align	8
IP6_OUTER	ip6_tcfl_comp	
IP6_LENGTH		
IP6_HL_OUTER	ip6_hl_comp	
IP6_SRC		

Table 23

A.3.2. IPv6 in IPv4

If the compressed inner IP header is an IPv6, but the outer IP header is an IPv4 header, the activated rules differ, as IP6_OUTER cannot be used. Instead, ip6_tcfl_comp and ip6_hl_comp are set to "Value". The resulting ESP packet is the same as in Figure 8.

EHC Rule	Context Attribute	Value
ESP_SPI	esp_spi_lsb	2
ESP_SN	esp_sn_lsb	2
	esp_sn_gen	
ESP_NH		
ESP_PAD	esp_align	8
IP6_OUTER	ip6_tcfl_comp	
IP6_LENGTH		
IP6_HL_OUTER	ip6_hl_comp	
IP6_SRC		

Table 24

A.3.3. IPv4 in IPv4

Figure 9 represents the original TCP packet with IPv4 inner IP addresses and Figure 10 represents the corresponding packet compressed with Diet-ESP. The compression with Diet-ESP results in a reduction of 24 bytes.

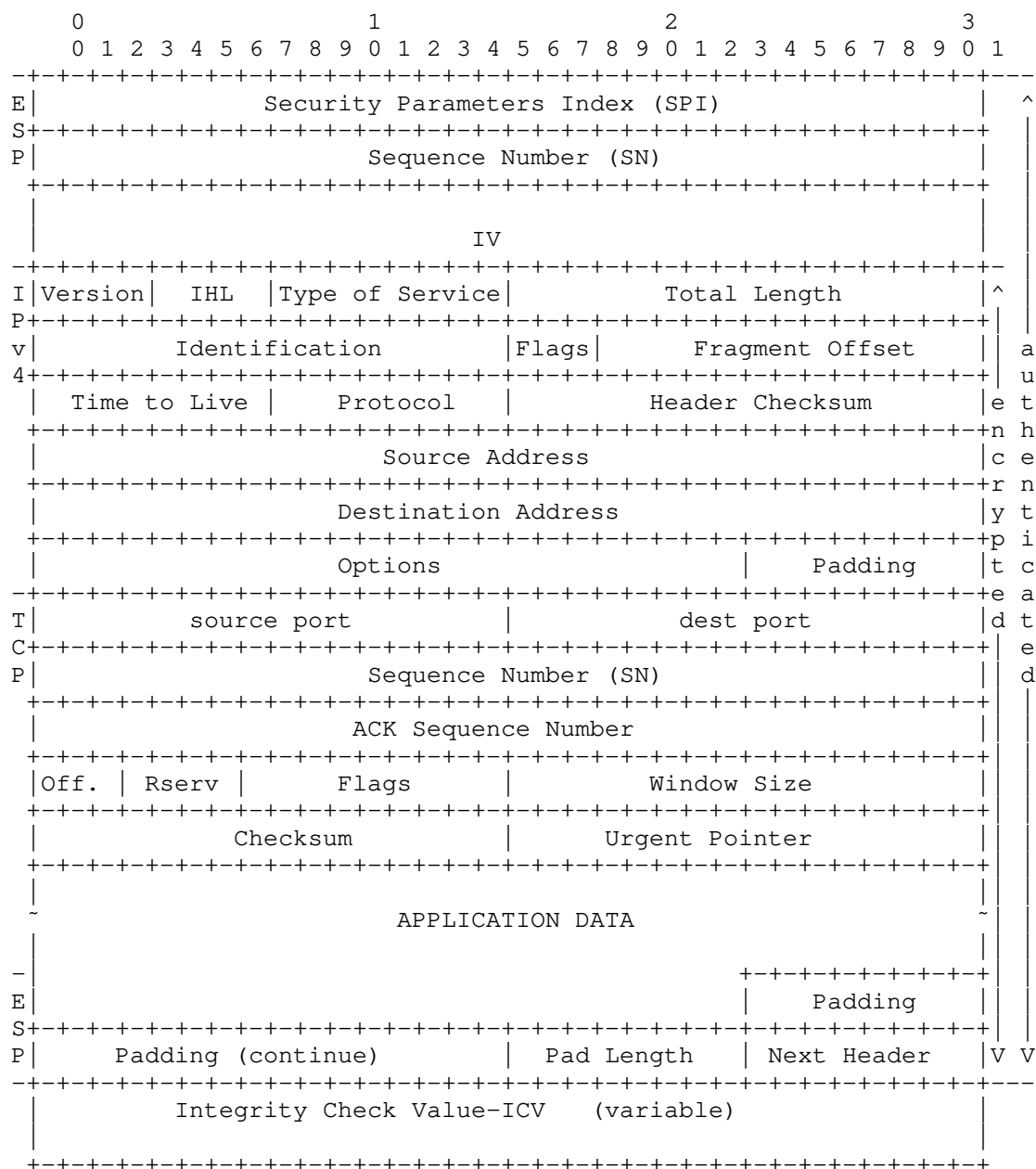


Figure 9: Standard ESP VPN Packet Description

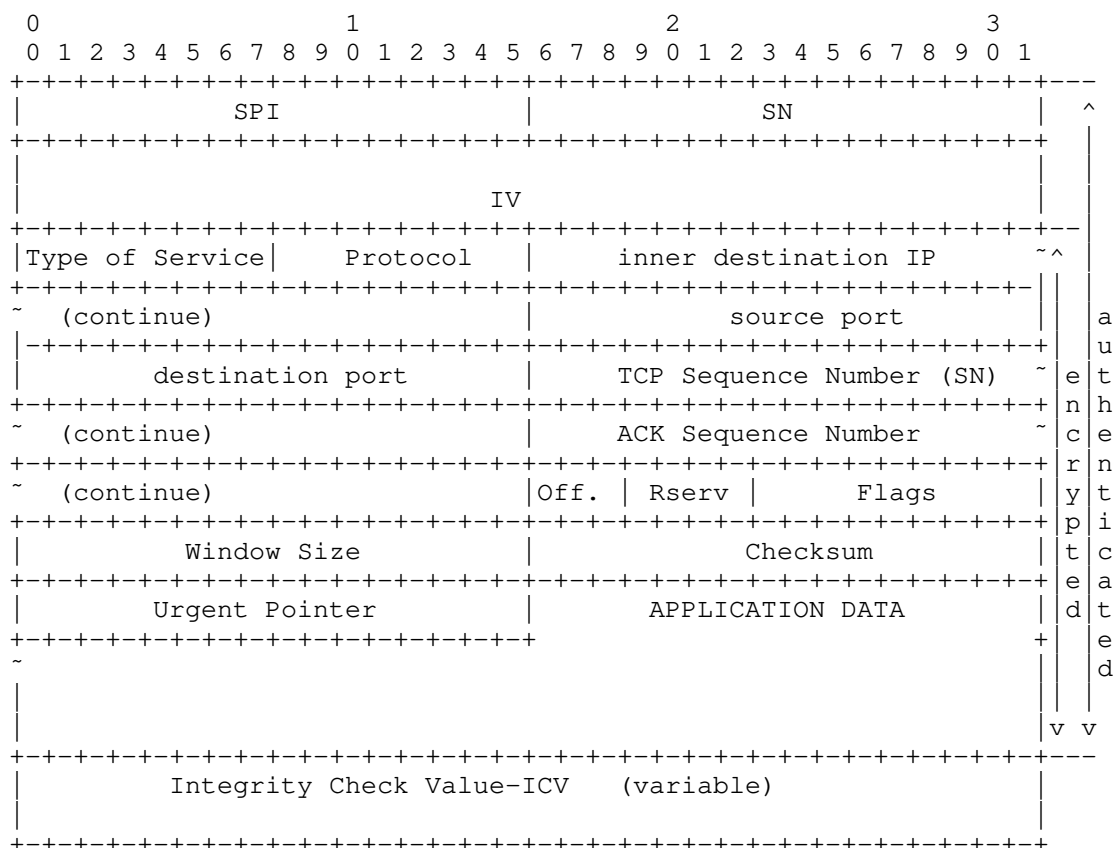


Figure 10: Diet-ESP VPN Packet Description

The following table illustrates the activated rules and the attributes of the Diet-ESP Context that needs an explicit agreement to achieve the compression. All other attributes used by the rules are part of the SA agreement. Parameters of not activated rules are left "Unspecified".

EHC Rule	Context Attribute	Value
ESP_SPI	esp_spi_lsb	2
ESP_SN	esp_sn_lsb	2
	esp_sn_gen	"Incremental"
ESP_NH		
ESP_PAD	esp_align	8
IP4_OPT_DIS		
IP4_LENGTH		
IP4_FRAG_DIS		
IP4_TTL_OUTER		
IP4_CHECK		
IP4_SRC		

Table 25

A.3.4. IPv4 in IPv6

If the compressed inner IP header is an IPv4, but the outer IP header is an IPv6 header, the activated rules differ, as IP4_TTL_OUTER cannot be used. Instead, IP4_TTL_VALUE is used. The resulting ESP packet is the same as in Figure 10.

EHC Rule	Context Attribute	Value
ESP_SPI	esp_spi_lsb	2
ESP_SN	esp_sn_lsb	2
	esp_sn_gen	"Incremental"
ESP_NH		
ESP_PAD	esp_align	8
IP4_OPT_DIS		
IP4_LENGTH		
IP4_FRAG_DIS		
IP4_CHECK		
IP4_SRC		

Table 26

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 Munchen
Germany
Email: guggemos@nm.ifi.lmu.de
URI: <http://www.nm.ifi.lmu.de/~guggemos>

Carsten Bormann
Universitaet Bremen TZI
Postfach 330440
Bremen D-28359
Germany
Phone: +49-421-218-63921
Email: cabo@tzi.org

David Schinazi
Google LLC
1600 Amphitheatre Parkway
Mountain View, California 94043
United States of America
Email: dschinazi.ietf@gmail.com

IPSECME
Internet-Draft
Intended status: Standards Track
Expires: January 3, 2015

D. Migault, Ed.
Orange
T. Guggemos, Ed.
Orange / LMU Munich
July 2, 2014

Diet-ESP: Generating compressed IV and SN
draft-mglt-ipsecme-diet-esp-iv-generation-00.txt

Abstract

Diet-ESP describes how to compress the various ESP fields, thanks to the Diet-ESP Context. This document describes how the IV fields that belong to the encrypted payload can be compressed.

The document describes the extensions of the the Diet-ESP Context as well as the compression protocol.

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 3, 2015.

Copyright Notice

Copyright (c) 2014 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of

the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Requirements notation	2
2. Introduction	2
3. Terminology	3
4. Diet-ESP context extension	4
5. Pseudo Random Function	4
6. Protocol Description	4
7. IANA Considerations	5
8. Security Considerations	5
9. Acknowledgment	5
10. References	5
10.1. Normative References	5
10.2. Informational References	6
10.3. URIs	6
Appendix A. Document Change Log	6
Authors' Addresses	6

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

Diet-ESP [I-D.mglt-ipsecme-diet-esp] describes how to compress ESP fields. Fields are compressed according to a Diet-ESP Context. Diet-ESP has been described as a specific ROHC [RFC5795] framework that has no IR, IR-DYN nor any feed back ROHC message. It works in the Unidirectional mode of operation (U mode), and all necessary parameters are transmitted via the Diet-ESP Context that is negotiated between the two peers. As a result Diet-ESP defines a very specific and simplified ROHC framework which makes possible to implement Diet-ESP without implementing the whole ROHC.

In fact, Diet-ESP avoids ROHC complexity as a lot of parameters have already been negotiated with IKEv2 [RFC5996].

The Initialization Vector (IV) is defined as a input for AES encryption and decryption. In order to provide the appropriated IV value AES-CBC [RFC3602] and AES-CTR [RFC3686] sends the IV in each IP packet as shown in figure Figure 1. In fact the output of AES-CTR and AES-CBC outputs a payload where the encrypted data is appended to the IV.

The IV MUST have to properties 1) they MUST be unpredictable by someone observing the network, then 2) the IV MUST be unique. The size of the IV differs depending on the encryption algorithm. AES-CTR has an 8 byte IV and AES-CBC a 16 byte IV.

This document defines a way to avoid sending the IV in each packet. Instead peers agree on a suite of pseudo random bytes. This makes the IV predictable by both peers only, and random to the rest of the world. As the IV can be derived by both peers, it may be removed completely from each IP packet. Another way is to only provide the LSB of the generated IV so receiver can better identify the appropriated IV used for decryption.

Note that the ICV of standard ESP [RFC4303] and Diet-ESP ICV includes the whole IV. As a result, the IV MUST be restored prior to the ICV check.

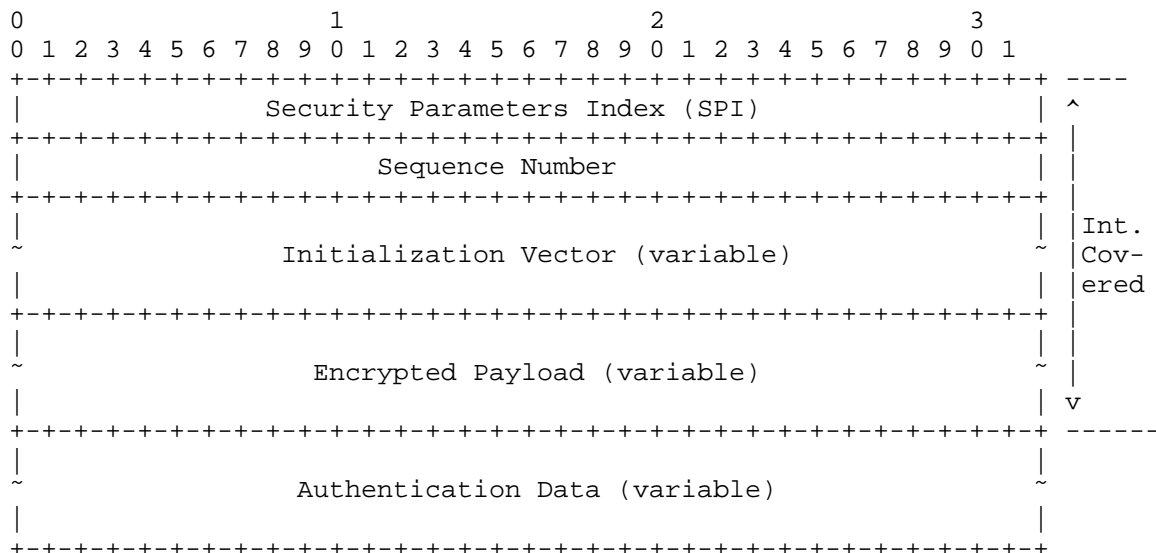


Figure 1: The IV in the ESP payload.

Section 4 describes the new parameters for the Diet-ESP Context. Section 5 describes how the Pseudo Random Function is derived, and Section 6 describes the protocol.

3. Terminology

- IoT: Internet of Things
- IV: Initialization Vector

- ICV: Integrity Check Value
- PRF: Pseudo Random Function

4. Diet-ESP context extension

To enable the compression of the IV, the Diet-ESP context defined in [I-D.mglt-ipsecme-diet-esp] is extended with the values:

IV_COMPRESSION:

Defines if the IV is generated and compresses.

IV_PRFT (optional):

Defines the Pseudo Random Function Transform used for the Pseudo Random Function. Available IDs are defined in [1] Section Transform Type 2 - Pseudo random Function Transform IDs. Section 2.13 [RFC5996] defines how the PRF is derived. By default PRF_AES128_CBC is the Pseudo Random Function Transform considered.

IV_LSB:

Defines the number of Least Significant Bytes of the IV carried by the payload.

5. Pseudo Random Function

The Pseudo Random Function (PRF) is defined from the Pseudo Random Function Transform in Section 2.13 [RFC5996]. Unless specified otherwise PRF_AES128_XCBC [RFC4434] is the default Pseudo Random Function Transform.

The PRF "prf+" described in Section 2.13 [RFC5996] takes two arguments designated as K and S. In this document K is the encryption key and S is the authentication key appended to the string "IV random generation". The string results in non null S value even if no integrity algorithms are negotiated.

6. Protocol Description

IV generation and compression is performed only and only if IV_COMPRESSION is set. Otherwise, the IV is embedded into the packet and sent on the wire as described in [RFC4303].

When IV_COMPRESSION is set, the PRD is defined as described in Section 5. On the sending part, the ICV or Diet-ESP ICV is computed, the IV is compressed to its LSB, before it is sent on the wire. On the receiver part, the IV is decompressed prior to the ICV check, then decryption is performed with the decompressed IV.

7. IANA Considerations

There are no IANA consideration for this document.

8. Security Considerations

9. Acknowledgment

The current draft represents the work of Tobias Guggemos while his internship at Orange [GUGG14] .

Diet-ESP is a joint work between Orange and Ludwig-Maximilians-Universitaet Munich. We thank Daniel Palomares and Carsten Bormann for their useful remarks, comments and guidance.

10. References

10.1. Normative References

- [I-D.mglt-ipsecme-diet-esp]
Migault, D., Guggemos, T., and D. Palomares, "Diet-ESP: a flexible and compressed format for IPsec/ESP", draft-mglt-ipsecme-diet-esp-00 (work in progress), March 2014.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC3602] Frankel, S., Glenn, R., and S. Kelly, "The AES-CBC Cipher Algorithm and Its Use with IPsec", RFC 3602, September 2003.
- [RFC3686] Housley, R., "Using Advanced Encryption Standard (AES) Counter Mode With IPsec Encapsulating Security Payload (ESP)", RFC 3686, January 2004.
- [RFC4303] Kent, S., "IP Encapsulating Security Payload (ESP)", RFC 4303, December 2005.
- [RFC4434] Hoffman, P., "The AES-XCBC-PRF-128 Algorithm for the Internet Key Exchange Protocol (IKE)", RFC 4434, February 2006.
- [RFC5795] Sandlund, K., Pelletier, G., and L-E. Jonsson, "The RObust Header Compression (ROHC) Framework", RFC 5795, March 2010.

- [RFC5996] Kaufman, C., Hoffman, P., Nir, Y., and P. Eronen,
"Internet Key Exchange Protocol Version 2 (IKEv2)", RFC
5996, September 2010.

10.2. Informational References

- [GUGG14] Guggemos, TG., "Diet-ESP: Applying IP-Layer Security in
Constrained Environments (Masterthesis)", September 2014.

10.3. URIs

- [1] [http://www.iana.org/assignments/ikev2-parameters/
ikev2-parameters.xhtml#ikev2-parameters-6](http://www.iana.org/assignments/ikev2-parameters/ikev2-parameters.xhtml#ikev2-parameters-6)

Appendix A. Document Change Log

[draft-mglt-ipsecme-diet-esp-IV-generation-00.txt]: First version
published.

Authors' Addresses

Daniel Migault (editor)
Orange
38 rue du General Leclerc
92794 Issy-les-Moulineaux Cedex 9
France

Phone: +33 1 45 29 60 52
Email: daniel.migault@orange.com

Tobias Guggemos (editor)
Orange / LMU Munich
Am Osteroesch 9
87637 Seeg, Bavaria
Germany

Email: tobias.guggemos@gmail.com

IPSECME
Internet-Draft
Intended status: Standards Track
Expires: January 3, 2015

D. Migault, Ed.
Orange
T. Guggemos, Ed.
Orange / LMU Munich
July 2, 2014

Diet-IPsec: ESP Payload Compression of IPv6 / UDP / TCP / UDP-Lite
draft-mglt-ipsecme-diet-esp-payload-compression-00.txt

Abstract

ESP is a IPsec protocol that takes as input a Clear Text Data and outputs an encrypted ESP packet according to IPsec rules and parameters stored in different IPsec databases.

Diet-ESP compresses the ESP fields. However, Diet-ESP does not consider compression of the Clear Text Data. Instead, if compression of the Clear Text Data is expected protocols like ROHcoverIPsec can be used.

ROHcoverIPsec remains complex to implement in IoT devices, as states, and negotiations are involved between the compressors and decompressors of the two IoT devices. Most of this complexity can be avoided by considering the parameters that have been negotiated by IPsec.

This document describes an extension of the Diet-ESP Context that enables the compression of the Clear Text Data, without implementing the complex ROHcoverIPsec framework. As opposed to ROHcoverIPsec the compression is not generic and as such all communication will not benefit from this compression. However, we believe this extension addresses most of IoT communications.

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 3, 2015.

Copyright Notice

Copyright (c) 2014 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Requirements notation	2
2. Introduction	3
3. Terminology	4
4. Diet-ESP Context Extension	4
5. Protocol Overview	5
6. IP Layer Compression	6
7. UDP Transport Layer Compression	8
8. UDP-Lite Transport Layer Compression	9
9. TCP Transport Layer Compression	10
10. IANA Considerations	11
11. Security Considerations	11
12. Acknowledgment	11
13. References	11
13.1. Normative References	11
13.2. Informational References	12
Appendix A. Interaction with ROHC profiles	12
Appendix B. Document Change Log	13
Authors' Addresses	13

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

Diet-ESP [I-D.mglt-ipsecme-diet-esp] describes how to compress ESP fields. Fields are compressed according to a Diet-ESP Context. Diet-ESP has been described as a specific ROHC [RFC5795] framework that has no IR, IR-DYN nor any feed back ROHC message. It works in the Unidirectional mode of operation (U mode), and all necessary parameters are transmitted via the Diet-ESP Context that is negotiated between the two peers. As a result Diet-ESP defines a very specific and simplified ROHC framework which makes possible to implement Diet-ESP without implementing the whole ROHC.

In fact, Diet-ESP avoids ROHC complexity as a lot of parameters have already been negotiated with IKEv2 [RFC5996].

This document describes the Diet-ESP Payload Compression Extension. It does not consider the compression of the ESP fields. Instead, it goes one step further and describes how to compress the Clear Text Data or ESP Payload before it is encrypted by Diet-ESP. The Clear Text Data is generally constituted by an IP packet with IP -- if IPsec tunnel mode is used --, transport and application layers. Similarly to Diet-ESP, compression takes advantage of the IPsec parameters -- like IP addresses, transport layer parameters -- that have been negotiated in order to establish the Security Association -- via IKEv2 for example. In addition, similarly to Diet-ESP, the compression is described using the ROHC terminology, but uses a very specific and simplified ROHC framework of Diet-ESP. This makes possible compression of the Clear Text Data without implementing a whole ROHC framework for ROHCoverIPsec [RFC5856].

[I-D.mglt-ipsecme-diet-esp] clarifies the interactions of Diet-ESP with ROHC and 6LoWPAN. The Diet-ESP extension explained in this document replaces ROHCoverIPsec and 6LoWPANoverIPsec, protocols which offers similar functionality without using the IPsec databases. The Diet-ESP Payload Compression Extension uses the IPsec databases to avoid complex dialogues between compressors and decompressors.

The Diet-ESP Payload Compression Extension can be described as follows:

- 1. Definition of Diet-ESP parameters: COMPRESS_ESP_PAYLOAD, CHECKSUM_LSB and SEQUENCE_NUMBER_LSB. COMPRESS_ESP_PAYLOAD indicates the peers expect the Clear Text Data to be compressed, CHECKSUM_LSB and SEQUENCE_NUMBER_LSB are additional parameters to perform the compression.
- 2. Definition of a Diet-ESP Payload Compression algorithm.

The remaining of the document is as follows. Section 4 describes the new parameters for the Diet-ESP Context. Section 5 describes the protocol. Section 6, Section 7, Section 7 and Section 9 describe the compression of the IP layer and the transport layer (UDP, UDP-Lite).

3. Terminology

Diet-ESP Context: Like defined in Diet-ESP document.

SPD: Security Policy Database

SAD: Security Association Database

TS: Traffic Selector of a Security Association.

LSB: Least Significant Byte

MSB: Most Significant Byte

4. Diet-ESP Context Extension

This section describes the additional parameters of the Diet-ESP Context to implement the ESP Payload Compression extension.

Context Field Name	Overview
COMPRESS_ESP_PAYLOAD	Defines the use of the Traffic Selector for (de-)compression.
CHECKSUM_LSB	LSB of the UDP, UDP-Lite or TCP checksum
SEQUENCE_NUMBER_LSB	LSB of the TCP Sequence Number.

Table 1: Diet-ESP Context.

COMPRESS_ESP_PAYLOAD:

Defines if the ESP Payload MUST be compressed or not. Note that as detailed later, compression of the ESP Payload requires that IP addresses, or protocols are unique in the Security Association Databases. If not the compression does not compress does not output a compressed ESP Payload.

CHECKSUM_LSB:

If an inner header provides a checksum this can be compressed by the LSB mechanism. How the checksum is compressed is specified by the related profiles, e.g. UDP Section 7, UDP-Lite Section 8 and TCP Section 9.

SEQUENCE_NUMBER_LSB:

If an inner header provides a Sequence Number, one MAY choose to use the SN stored in the SA for compression. Therefore the context provides the LSB of the Sequence Number which is used by all profiles, defining the Sequence Number as compressed with LSB, e.g. TCP Section 9.

5. Protocol Overview

The Diet-ESP Payload Compression is described by the pseudo code in Figure 1. The Clear Text Data is compressed only if COMPRESS_ESP_PAYLOAD is set. Otherwise, it is left unchanged. When COMPRESS_ESP_PAYLOAD is set, compression is performed on the IP and transport layer if and only if two conditions are met. First the layer must exist. This means for example that the IP layer is compressed only for the tunnel mode. Then, the layer can be compressed if and only if the values are uniquely derived from the IPsec databases. More specifically, if a SPD match occurs with at least two different values, then the compression do not occurs.

As a result, the IP layer can be compressed only if the IP address appears as a Traffic Selector. If the Traffic Selector is defined as a subnetwork, a SPD match occurs with more then one IP address, and then no compression occurs. Similarly, the transport layer is compressed if and only if it appears as a Traffic Selector. If a SPD match occurs with different transport protocol then the compression of the transport layer does not occurs.

The Diet-ESP Payload Compression is straight forward, but may at some point not fits all the needs. At some point using alternative compression as those proposed by ROHcoverIPsec may be preferred. In these cases, Diet-ESP Payload Compression MUST NOT be performed and COMPRESS_ESP_PAYLOAD MUST be unset.

```

if COMPRESS_ESP_PAYLOAD is set :
    proceed to Diet-ESP Payload Compression
else:
    clear_text_data is left unchanged.

def diet_esp_payload_compression(clear_text_data, \
                                CHECKSUM_LSB,\
                                SEQUENCE_NUMBER_LSB):
    if clear_text_data has IP layer and \ ## i.e. IPsec mode Tunnel mode
        IP address is a Traffic Selector:  ## subnets are not considered
        compress the IP layer
    if clear_text_data has transport layer and \
        transport layer is a Traffic Selector:
        compress transport layer

```

Figure 1: Diet-ESP Payload Compression Pseudo Code

Roughly speaking Diet-ESP is able to remove all header fields which have unique values inside the Security Association Database. Most probably they are stored in the Traffic Selector, which defines the traffic which has to be secured with IPsec. Table 2 shows some header fields which can be adopted from the Traffic Selector. The table provides the ROHC class of these values, as we use the ROHC terminology to describe the compression algorithms.

Field	Protocol	ROHC class
IP version	IP/IPv6	STATIC-KNOWN
Source Address	IP/IPv6	STATIC-DEF
Destination Address	IP/IPv6	STATIC-DEF
Next Header	IP/IPv6	STATIC
Source PORT	UDP/TCP	STATIC-DEF
Destination PORT	UDP/TCP	STATIC-DEF

Table 2: This values are carried in the Security Association.

6. IP Layer Compression

This section describes how the compression of the IP layer is performed. The compression of this layer mostly occurs when the peers have negotiated the IPsec tunnel mode.

The basic idea for IP layer compression is to remove the IP layer before Diet-ESP encrypts the Clear Text Data. Similarly, for incoming packet, Diet-ESP decrypts the ESP packet, and restores the IP layer by reading the IP address in the IPsec SAD. However, the IP

address is not sufficient to restore the complete IP header as other fields must be considered. To appropriately describes the compression of the IP layer, this section uses the ROHC terminology and describes the associated profile.

The IP header is classified as shown in Table 3

Field	Class	Compression Method	Diet-ESP ROHC class	Data origin
Version	STATIC	removed	STATIC	TS
Traffic Class	CHANGING	removed	INFERRED	outer IP
Flow Label	STATIC-DEF	removed	STATIC-DEF	outer IP
Payload Length	INFERRED	removed	INFERRED	outer IP
Next Header	STATIC	removed	STATIC	TS
Hop Limit	RACH	removed	INFERRED	outer IP
Source Address	STATIC-DEF	removed	STATIC-DEF	TS
Destination Address	STATIC-DEF	removed	STATIC-DEF	TS

Table 3: Header classification for IPv6.

Version:

The IP version is specified in the SA and can be copied to the ROHC context, before the first packet is sent/received.

Traffic Class:

Traffic Class can be read from the outer IP header. Therefore the classification is changed to INFERRED.

Flow Label:

Flow Label can be read from the outer IP header. Therefore the classification is changed to INFERRED.

Next Header

The Next Header is stored in the protocol of the Traffic Selector and is fixed. It can be copied to the ROHC context, before the first packet is sent/received.

Hop Limit

The Hop Limit can be read from the outer IP header. Therefore the classification is changed to INFERRED.

Source Address:

The Source Address is fixed in the SA and can be copied to the ROHC context, before the first packet is sent/received.

Destination Address:

The Destination Address is fixed in the SA and can be copied to the ROHC context, before the first packet is sent/received.

7. UDP Transport Layer Compression

This section shows the compression of ESP payload for all ROHC profiles including an UDP header.

The UDP header is classified as shown in Table 4

Field	Class	Compr. Method	Diet-ESP ROHC class	Data origin
Source Port	STATIC-DEF	removed	STATIC-DEF	TS
Destination Port	STATIC-DEF	removed	STATIC-DEF	TS
Length	INFERRED	removed	INFERRED	IP payload length
Checksum	IRREGULAR	LSB	INFERRED	calc.

Table 4: Header classification for UDP.

Source Port:

The Source Port is fixed in the SA and can be copied to the ROHC context, before the first packet is sent/received.

Destination Port:

The Destination Port is fixed in the SA and can be copied to the ROHC context, before the first packet is sent/received.

Length:

The length of the UDP header can be calculated like: IP header - IP header length. Therefore there is no need to send it on the wire and it is defined as INFERRED.

Checksum:

The checksum can be calculated by Diet-ESP and proved by comparing the LSB sent on the wire. The number of bytes sent on the wire can be 0, 1 and 2 stored in CHECKSUM_LSB. If 0 LSB is chosen, the checksum MUST be decompressed with the value 0. If the UDP implementation of the sender chose to disable the UDP checksum by

setting the checksum to 0 Diet-ESP SHOULD be used with
CHECKSUM_LSB = 0.

8. UDP-Lite Transport Layer Compression

This section shows the compression of ESP payload for all ROHC profiles including an UDP-Lite header.

The UDP header is classified as shown in Table 5

Field	Class	Compression Method	Diet-ESP ROHC class	Data origin
Source Port	STATIC-DEF	removed	STATIC-DEF	TS
Destination Port	STATIC-DEF	removed	STATIC-DEF	TS
Checksum	IRREGULAR	LSB	IRREGULAR	calc.
Coverage				
Checksum	IRREGULAR	LSB	INFERRED	calc.

Table 5: Header classification for UDP-Lite.

Source Port:

The Source Port is fixed in the SA and can be copied to the ROHC context, before the first packet is sent/received.

Destination Port:

The Destination Port is fixed in the SA and can be copied to the ROHC context, before the first packet is sent/received.

Checksum Coverage:

The Checksum specifies the number of octets carried by the UDP-Lite checksum. It can have the same value as the UDP length (0 or UDP length) or any value between 8 and UDP length. This field is compressed with CHECKSUM_LSB of 0, 1 or 2 bytes. If 0 or 1 LSB is chosen, the field MUST be decompressed with the UDP length. If 2 LSB is chosen, the checksum has to carry this behaviour.

Checksum:

The checksum can be calculated by Diet-ESP and proved by comparing the LSB sent on the wire. The number of bytes sent on the wire can be 0, 1 and 2 stored in CHECKSUM_LSB. If 0 LSB is chosen, the checksum MUST be decompressed with the value 0. If an UDP-lite implementation of the sender chose to disable the UDP checksum by setting the checksum to 0 Diet-ESP SHOULD be used with CHECKSUM_LSB = 0.

9. TCP Transport Layer Compression

This section shows the compression of ESP payload for all ROHC profiles including a TCP header. The ROHC context is partly filled while the Diet-ESP context exchange, wherefore some values can be removed. Since TCP is not stateless only fields with the compression methods 'removed' and 'LSB' are allowed to be compressed, the other fields MUST be sent on the wire uncompressed.

The UDP header is classified as shown in Table 6

Field	Class	Compression Method	Diet-ESP ROHC class	Data origin
Source Port	STATIC-DEF	removed	STATIC-DEF	TS
Destination Port	STATIC-DEF	removed	STATIC-DEF	TS
Sequence Number	CHANGING	LSB	CHANGING	ESP SN
Acknowledgement Num	INFERRED	N/A	INFERRED	
Data Offset	CHANGING	N/A	CHANGING	
Reserved	CHANGING	N/A	CHANGING	
CWR flag	CHANGING	N/A	CHANGING	
ECE flag	CHANGING	N/A	CHANGING	
URG flag	CHANGING	N/A	CHANGING	
ACK flag	CHANGING	N/A	CHANGING	
PSH flag	CHANGING	N/A	CHANGING	
RST flag	CHANGING	N/A	CHANGING	
SYN flag	CHANGING	N/A	CHANGING	
FIN flag	CHANGING	N/A	CHANGING	
Window	CHANGING	N/A	CHANGING	
Checksum	IRREGULAR	LSB	INFERRED	calc.
Urgent Pointer	CHANGING	N/A	CHANGING	
Options	CHANGING	N/A	CHANGING	

Table 6: Header classification for TCP.

Source Port:

The Source Port is fixed in the SA and can be copied to the ROHC context, before the first packet is sent/received.

Destination Port:

The Destination Port is fixed in the SA and can be copied to the ROHC context, before the first packet is sent/received.

Sequence Number:

The Sequence Number can be compressed with a LSB by using the SN stored in the SA for the remaining MSB not sent on the wire.

Checksum:

The checksum can be calculated by Diet-ESP and proved by comparing the LSB sent on the wire. The number of bytes sent on the wire can be 0, 1 and 2 stored in CHECKSUM_LSB. If 0 LSB is chosen, the checksum MUST be decompressed with the value 0. If an UDP-lite implementation of the sender chose to disable the UDP checksum by setting the checksum to 0 Diet-ESP SHOULD be used with CHECKSUM_LSB = 0.

10. IANA Considerations

There are no IANA consideration for this document.

11. Security Considerations

12. Acknowledgment

The current draft represents the work of Tobias Guggemos while his internship at Orange [GUGG14] .

Diet-ESP is a joint work between Orange and Ludwig-Maximilians-Universitaet Munich. We thank Daniel Palomares and Carsten Bormann for their useful remarks, comments and guidance.

13. References

13.1. Normative References

[I-D.mglt-ipsecme-diet-esp]

Migault, D., Guggemos, T., and D. Palomares, "Diet-ESP: a flexible and compressed format for IPsec/ESP", draft-mglt-ipsecme-diet-esp-00 (work in progress), March 2014.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[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, July 2001.

- [RFC3843] Jonsson, L-E. and G. Pelletier, "RObust Header Compression (ROHC): A Compression Profile for IP", RFC 3843, June 2004.
- [RFC4019] Pelletier, G., "RObust Header Compression (ROHC): Profiles for User Datagram Protocol (UDP) Lite", RFC 4019, April 2005.
- [RFC5225] Pelletier, G. and K. Sandlund, "RObust Header Compression Version 2 (ROHCv2): Profiles for RTP, UDP, IP, ESP and UDP-Lite", RFC 5225, April 2008.
- [RFC5795] Sandlund, K., Pelletier, G., and L-E. Jonsson, "The RObust Header Compression (ROHC) Framework", RFC 5795, March 2010.
- [RFC5996] Kaufman, C., Hoffman, P., Nir, Y., and P. Eronen, "Internet Key Exchange Protocol Version 2 (IKEv2)", RFC 5996, September 2010.
- [RFC6846] Pelletier, G., Sandlund, K., Jonsson, L-E., and M. West, "RObust Header Compression (ROHC): A Profile for TCP/IP (ROHC-TCP)", RFC 6846, January 2013.

13.2. Informational References

- [GUGG14] Guggemos, TG., "Diet-ESP: Applying IP-Layer Security in Constrained Environments (Masterthesis)", September 2014.
- [RFC5856] Ertekin, E., Jasani, R., Christou, C., and C. Bormann, "Integration of Robust Header Compression over IPsec Security Associations", RFC 5856, May 2010.

Appendix A. Interaction with ROHC profiles

Each ROHC profile defines compression rules for a set of protocol headers. Table 7 clarifies how ROHC profiles can be mapped to Diet-ESP payload compression.

Profile Number	ROHC version	Protocol	RFC	Diet-ESP compression
0x0000	ROHC	uncompressed IP	[RFC3095]	no compression
0x0001	ROHC	RTP/UDP/IP	[RFC3095]	not used
0x1001	ROHCv2	RTP/UDP/IP	[RFC5225]	not used
0x0002	ROHC	UDP/IP	[RFC3095]	UDP and IP in Tunnel Mode
0x1002	ROHCv2	UDP/IP	[RFC5225]	UDP and IP in Tunnel Mode
0x0003	ROHC	ESP/IP	[RFC3095]	not used
0x1003	ROHCv2	ESP/IP	[RFC5225]	not used
0x0004	ROHC	IP	[RFC3843]	IP in Tunnel Mode
0x1004	ROHCv2	IP	[RFC5225]	IP in Tunnel Mode
0x0006	ROHC	TCP/IP	[RFC6846]	TCP and IP in Tunnel Mode
0x0007	ROHC	RTP/UDP-Lite/IP	[RFC4019]	not used
0x1007	ROHCv2	RTP/UDP-Lite/IP	[RFC5225]	not used
0x0008	ROHC	UDP-Lite/IP	[RFC4019]	UDP-Lite and IP in Tunnel Mode
0x1008	ROHCv2	UDP-Lite/IP	[RFC5225]	UDP-Lite and IP in Tunnel Mode

Table 7: Overview over currently existing ROHC profiles.

Appendix B. Document Change Log

00-First version published

Authors' Addresses

Daniel Migault (editor)
 Orange
 38 rue du General Leclerc
 92794 Issy-les-Moulineaux Cedex 9
 France

Phone: +33 1 45 29 60 52
 Email: daniel.migault@orange.com

Tobias Guggemos (editor)
Orange / LMU Munich
Am Osteroesch 9
87637 Seeg, Bavaria
Germany

Email: tobias.guggemos@gmail.com

IPSECME
Internet-Draft
Intended status: Standards Track
Expires: January 3, 2015

D. Migault, Ed.
Orange
T. Guggemos, Ed.
Orange / LMU Munich
July 2, 2014

Diet-IPsec: Requirements for new IPsec/ESP protocols according to IoT
use cases
draft-mglt-ipsecme-diet-esp-requirements-00.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 3, 2015.

Copyright Notice

Copyright (c) 2014 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Requirements notation	2
2. Introduction	2
3. Terminology	3
4. Byte-Alignment	3
5. Crypto-Suites	3
6. Compression	4
7. Flexibility	4
8. Code Complexity	5
9. Usability	5
10. Compatibility with IP compression Protocols	5
11. Compatibility with Standard ESP	6
12. IANA Considerations	6
13. Security Considerations	6
14. Acknowledgment	6
15. Normative References	6
Appendix A. Power Consumption Example	7
Appendix B. Document Change Log	8
Authors' Addresses	8

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.

A common disadvantage of IPsec is that it is mostly implemented in the kernel, whereas application are in the user space. 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. 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.

R1: Diet-ESP SHOULD support Byte-Alignment that are different from 32 bits or 64 bits to prevent unnecessary padding.

R2: Each peer SHOULD be able to advertise and negotiate the Byte-Alignment, used for Diet-ESP. This could be done for example during the IKEv2 exchange.

5. 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.

- R3: Diet-ESP MUST support AES-CCM and MUST be able to take advantage of AES-CCM hardware acceleration. Diet-ESP MAY support other modes.

6. 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.

- R4: Diet-ESP SHOULD be able to compress/remove all static ESP fields (SPI, Next Header) as well as the other fields SN, PADDING, Pad Length or ICV.
- R5: Diet-ESP SHOULD also allow compression mechanisms before the IPsec/ESP processing.
- R6: 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.

7. 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 be foreseen. In fact Diet-ESP and ESP differs in the following point: ESP has been designed so that any ESP secured communication on 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.

- R7: The developer SHOULD be able to specify the maximum level of compression.

- R8: Diet-ESP SHOULD be able to compress any field independent from another.
- R9: Diet-ESP SHOULD be able to define different compression method, when appropriated.
- R10: Each peer SHOULD be able to announce and negotiate the different compressed fields as well as the used method.

8. Code Complexity

IoT devices have limited space for memory and storage.

- R11: Diet-ESP SHOULD be able to be implemented with minimal complexity. More especially, Diet-ESP SHOULD consider small implementation that implement only a subset of all Diet-ESP capabilities without requiring involving standard ESP, specific compressors and de-compressors.

9. Usability

Application Developer usually do not want to take care about the underlying protocols and security. Standard ESP addresses the goal by providing a framework that secures communication in any circumstances. Although application developers for IoT are expected to pay more attention to the device security and system requirements, we do not expect them to be security aware developers. As a result, some default parameters that provides a standard secure framework for most cases should be provided. This is of course performed at the expense of some optimization, but it makes possible for application developers to have "standard" security and standard Diet-ESP compression by setting a single bit "DIET-ESP secure". More advanced developers will be able to tune the security parameters for their needs.

- R12: Diet-ESP SHOULD provide default configurations, which can be easily set up by a developer.

10. Compatibility with IP compression Protocols

There are different protocols providing IP layer compression for constraint devices like IoT (6LoWPAN [RFC6282]) or Mobile Devices (ROHC).

- R13: Diet-ESP SHOULD be able to interact with IP compression protocols. More especially, this means that a Diet-ESP packet SHOULD be able to be sent in a ROHC or a 6LowPAN packet. Diet-ESP document should explicitly detail how this can be achieved.

R14: Diet-ESP SHOULD also detail how compression of layers above IP with ROHC or 6LoWPAN is compatible with Diet-ESP.

11. 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.

R15: Diet-ESP SHOULD be able to interact with Standard ESP implementations on a single platform.

R16: Diet-ESP SHOULD be able to communicate with Standard ESP.

12. IANA Considerations

There are no IANA consideration for this document.

13. Security Considerations

14. Acknowledgment

The current draft represents the work of Tobias Guggemos while his internship at Orange [GUGG14].

Diet-ESP is a joint work between Orange and Ludwig-Maximilians-Universitaet Munich. We thank Daniel Palomares and Carsten Bormann for their useful remarks, comments and guidance.

15. Normative References

- [GUGG14] Guggemos, TG., "Diet-ESP: Applying IP-Layer Security in Constrained Environments (Masterthesis)", September 2014.
- [RFC0791] Postel, J., "Internet Protocol", STD 5, RFC 791, September 1981.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC2460] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", RFC 2460, December 1998.

- [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, July 2001.
- [RFC4301] Kent, S. and K. Seo, "Security Architecture for the Internet Protocol", RFC 4301, December 2005.
- [RFC4303] Kent, S., "IP Encapsulating Security Payload (ESP)", RFC 4303, December 2005.
- [RFC5225] Pelletier, G. and K. Sandlund, "RObust Header Compression Version 2 (ROHCv2): Profiles for RTP, UDP, IP, ESP and UDP-Lite", RFC 5225, April 2008.
- [RFC6282] Hui, J. and P. Thubert, "Compression Format for IPv6 Datagrams over IEEE 802.15.4-Based Networks", RFC 6282, September 2011.

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-ipsecme-diet-ipsec-requirements-00.txt]: First version published.

Authors' Addresses

Daniel Migault (editor)
Orange
38 rue du General Leclerc
92794 Issy-les-Moulineaux Cedex 9
France

Phone: +33 1 45 29 60 52
Email: daniel.migault@orange.com

Tobias Guggemos (editor)
Orange / LMU Munich
Am Osteroesch 9
87637 Seeg, Bavaria
Germany

Email: tobias.guggemos@gmail.com

6Lo Working Group
-Draft
4 and RFC 6282 (if approved)
ds Track

D. Popa, Ed. Internet
Itron, Inc. Updates: RFC 494
J.H. Hui Intended status: Standard
Cisco Expires: September 30, 2014

March 31, 2014 6LoPLC: Transmission of IPv6 Packets
over IEEE 1901.2 Narrowband Powerline Communication Networks
draft-popa-6lo-6loplc-ipv6-over-ieee19012-networks-00.txt Abstract This
document updates [RFC 4944], "Transmission of IPv6 Packets over IEEE 802.15.4
Networks", and [RFC 6282], "Compression Format for IPv6 Datagrams over IEEE 80
2.15.4-Based Networks", and specifies the 6LoPLC technology: the transmission
of IPv6 packets over IEEE 1901.2 narrowband powerline communication networks. S
tatus of this Memo This Internet-Draft is submitted in full conformance with t
he 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, repl
aced, 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 30, 2014. Copyright Not
ice Copyright (c) 2014 IETF Trust and the persons identified as the document
authors. All rights reserved. This document is subject to BCP 78 and the IET
F Trust's Legal Provisions Relating to IETF Documents ([http://trustee.ietf.org](http://trustee.ietf.org/license-info)
/ license-info) in effect on the date of publication of this document. Pleas
e review these documents carefully, as they describe your rights and restricti
ons with respect to this document. Code Components extracted from this docume
nt must include Simplified BSD License text as described in Section 4.e of the
Trust Legal Provisions and are provided without warranty as described in the
Simplified BSD License. Popa & Hui Expires September 30, 2014

[Page 1]

Internet-Draft 6LoPLC March 2014 1. Introduction 6LoWPAN technology defines the transport of IPv6 packets over IEEE
802.15.4-2006 low power and lossy networks (LLNs). Because the 802.15.4-2006 w
ireless links do not support the IPv6 requirement for a link MTU of at least 1
280 octets, 6LoWPAN adaptation layer defines header compression and fragmentat
ion of IPv6 packets. A link in a LLN is characterized as lossy, low-power, low
bit-rate, and short range. The LLN nodes have resources constrained in terms
of processing power, memory capabilities, and communication bandwidth, due
to a combination of factors including regulations on spectrum use, form factor
and cost considerations. Recently, IEEE Standard Association published the IE
EE 1901.2 PHY and MAC standard for narrowband powerline communications (NB-PLC
) . When used in LLNs, apart from using powerline communications instead of w
ireless communications, the devices implementing IEEE 1901.2 standard share th
e same constraints as their wireless counterparts. 1.1. Applicability This doc
ument updates [RFC4944] and [RFC6282] and specifies 6LoPLC: the transmission o
f IPv6 packets over IEEE 1901.2 NB-PLC networks. The term 6LoPLC is used to ma
ke a clear difference between the 6LoWPAN technology, known in the industry as
a mechanism to transmit IPv6 packets over 802.15.4-2006 wireless networks, an
d the use of 6LoWPAN technology for the transmission of IPv6 packets over IEEE
1901.2 networks. This document specifies a set of behaviors between devices
in 1901.2 networks, which apply to both mesh and star topologies. An imple
mentation that adheres to this document MUST implement these behaviors. 1.2. I
EEE 1901.2 Technology This section describes those features from IEEE 1901.2 s
tandard that are relevant to the transmission of IPv6 packets over 1901.2 ne
tworks. For further details on IEEE 1901.2 technology, the reader is invited
to refer to [IEEE1901.2]. IEEE 1901.2 standard defines a Narrowband PLC PHY an
d MAC technology for indoor and outdoor communications (e.g., smart grid netwo
rks, home area networks). IEEE 1901.2 MAC frame format endorses the IEEE 802
.15.4-2006 MAC frame format [IEEE802.15.4], with a few exceptions described be
low. Popa & Hui Expires September 30, 2014 [Page 2]

Internet-Draft 6LoPLC March 2014 2. The IEEE 1901.2 MAC frame format is obtained by prepending a Segment Control
Field to the IEEE 802.15.4-2006 MAC frame. One function of the Segment Con
trol Field is to carry inline information for the MAC sub-layer fragmentati
on and reassembly process. Note that the complete format and use of Segmen
t Control Field are not relevant to the transmission of IPv6 packets over
IEEE 1901.2 networks. o IEEE 1901.2 MAC frame format endorses only the IE

IEEE 802.15.4-2006 short and extended MAC addresses with a length of 16 and 64 bits, respectively. o IEEE 1901.2 MAC frame format endorses the concept of Information Elements, as defined in IEEE 802.15.4e-2012 [IEEE802.15.4e]. Note that the format and use of Information Elements are not relevant

to the transmission of IPv6 packets over IEEE 1901.2 networks. The maximum size of a 1901.2 MAC frame payload is 1280 bytes, while the maximum size of a 1901.2 PHY frame payload is 512 bytes. The PHY frame payload size can vary from frame to frame, as a function of the modulation used to transmit the frame and the strength of the Forward Error Correction scheme. To cope with the mismatch between the size of the PHY frame payload and the size of the MAC frame, the IEEE 1901.2 standard specifies a mandatory MAC sub-layer fragmentation and reassembly process. This process fragments an upper layer datagram into multiple fragments and provides a reliable one-hop transfer of the resulting fragments.2. Transmission of IPv6 Packets over IEEE 1901.2 Networks The transmission of IPv6 packets over low-power and lossy networks relies on two mechanisms defined at 6LOWPAN adaptation layer. The first mechanism defines a set of procedures for IPv6 and UDP header compression (as specified in [RFC4944] and updated in [RFC6282]). The second mechanism defines a scheme for one-hop fragmentation and reassembly of IPv6 packets (as specified in [RFC4944]).2.1. 6LOWPAN Header compression Because IEEE 1901.2 fundamentally supports the IEEE 802.15.4-2006 MAC frame format and addressing scheme, IEEE 1901.2 devices implementing this specification MUST support the 6LOWPAN header compression schemes specified in [RFC6282]. Note that header compression mechanisms defined in [RFC6282] completely replace the header compression mechanisms defined in [RFC4944].2.2. 6LOWPAN FragmentationPopa & Hui Expires September 30, 2014

[Page 3]

Internet-Draft 6LoPLC March 2014 The use of fragmentation and reassembly consumes resources in terms of buffering and processing power. Also, fragmentation and reassembly consumes link capacity because, for each fragment that is transmitted, additional headers are required to properly manage the transmission, retransmission and reassembly of the fragments. As such, in the context of LLNs, where HW resources are constrained and network capacity is scarce, the fragmentation and reassembly should be avoided whenever possible. Because IEEE 1901.2 fundamentally supports a MAC payload of 1280 bytes and provides its own MAC sub-layer fragmentation mechanism, the use of 6LOWPAN fragmentation scheme defined in [RFC4944], when transmitting IPv6 packets over IEEE 1901.2 networks, is NOT RECOMMENDED.3. IANA Considerations No IANA considerations.4. Security Considerations This document has no security considerations beyond those in [RFC4291].5. Acknowledgements

The authors would like to acknowledge the review, feedback, and comments of Matthew Gillmore, Samita Chakrabarti, and Ulrich Herberg.6. References6.1. Normative References [RFC4291] Hinden, R. and S. Deering, "IP Version 6 Addressing Architecture", RFC 4291, February 2006. [RFC4944] Montenegro, G., Kushalnagar, N., Hui, J. and D. Culler, "Transmission of IPv6 Packets over IEEE 802.15.4 Networks", RFC 4944, September 2007. [RFC6282] Hui, J. and P. Thubert, "Compression Format for IPv6 Data Grams over IEEE 802.15.4-Based Networks", RFC 6282, September 2011.6.2. Informative References [IEEE1901.2] IEEE SA, "IEEE Standard for Low-Frequency (less than 500 kHz) Narrowband Power Line Communications for Smart Grid Applications", December 2013. [IEEE802.15.4]Popa & Hui Expires September 30, 2014 [Page 4]

Internet-Draft 6LoPLC March 2014 IEEE SA, "IEEE Standard for Information technology-- Local area and metropolitan area networks-- Specific requirements-- Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low Rate Wireless Personal Area Networks (WPANs)", September 2006. [IEEE802.15.4e] IEEE SA, "IEEE Standard for Local area and metropolitan area networks--Part 15.4: Low-Rate Wireless Personal Area Networks (LR-WPANs) Amendment 1: MAC sublayer", April 2012.

Authors' Addresses Daniel Popa, editor Itron, Inc. 52, rue Camille Desmoulins Issy les Moulineaux, 92130 FR Email: daniel.popa@itron.com Jonathan W. Hui Cisco 170 West Tasman Drive San Jose, California 95134 USA Phone: +408 424 1547 Email: jonhui@cisco.comPopa & Hui Expires September 30, 2014 [Page 5]

6lo
Internet-Draft
Intended status: Standards Track
Expires: September 10, 2015

B. Sarikaya, Ed.
Huawei USA
F. Xia
Huawei Technologies Co., Ltd.
P. Thubert, Ed.
Cisco
March 9, 2015

Lightweight and Secure Neighbor Discovery for Low-power and Lossy
Networks
draft-sarikaya-6lo-cga-nd-02

Abstract

This document defines a lightweight and secure version of 6LoWPAN Neighbor Discovery for application in low-power and lossy networks. Cryptographically Generated Address and digital signatures are calculated using Elliptic Curve Cryptography, so that the cryptographic operations are suitable for low power devices. An optimal version of this protocol is also specified which supports faster CGA calculation and multi-hop operation. A node computes a Cryptographically Generated Address to be used as a Unique Interface ID, and associate all its Registered Addresses with that Unique Interface ID in place of the EUI-64 that is used in RFC 6775 to uniquely identify the interface of the Registered Address. Once an address is registered with a cryptographic unique ID, only the owner of that ID can modify the state in the 6LR and 6LBR regarding the Registered Address.

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 10, 2015.

Copyright Notice

Copyright (c) 2015 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	2
2. Terminology	4
3. Requirements	4
4. New and Modified Options	5
4.1. Modified Address Registration Option	5
4.2. CGA Parameters and Digital Signature Option	6
4.3. Digital Signature Option	8
4.4. Calculation of the Digital Signature and CGA Using ECC	10
5. Protocol Interactions	10
6. Optimizations	11
6.1. Overview	11
6.2. Protocol Operations	14
6.3. Multihop Operation	15
7. Security Considerations	16
8. IANA considerations	16
9. Acknowledgements	16
10. References	16
10.1. Normative References	16
10.2. Informative references	18
Authors' Addresses	18

1. Introduction

Neighbor discovery for IPv6 [RFC4861] and stateless address autoconfiguration [RFC4862], together referred to as neighbor discovery protocols (NDP), are defined for regular hosts operating with wired/wireless links. These protocols are not suitable and require optimizations for resource constrained, low power hosts operating with lossy wireless links. Neighbor Discovery optimizations for 6LoWPAN networks include simple optimizations such as a host address registration feature using the address registration

option (ARO) which is sent in unicast Neighbor Solicitation (NS) and Neighbor Advertisement (NA) messages [RFC6775]. With 6LoWPAN ND [RFC6775], the ARO option includes a EUI-64 address to uniquely identify the interface of the Registered Address on the registering device, so as to correlate further registrations for a same address and avoid address duplication. The EUI-64 address is not secured and its ownership cannot be verified. It results that any device claiming the same EUI-64 address may take over a registration and attract the traffic for that address.

Neighbor Discovery Protocols (NDP) are not secure especially when physical security on the link is not assured and vulnerable to attacks defined in [RFC3756]. Secure neighbor discovery protocol (SEND) is defined to secure NDP [RFC3971]. Cryptographically Generated Addresses (CGA) are used in SEND [RFC3972]. SEND mandates the use of the RSA signature algorithm which is computationally heavy and not suitable to use for low-power and resource constrained nodes. The use of an RSA public key and signature leads to long message sizes not suitable to use in low-bit rate, short range, asymmetric and non-transitive links such as IEEE 802.15.4.

In this document, we extend 6LoWPAN ND with CGA; but as opposed to SEND, the cryptographic address is not necessarily used as Interface ID (IID) in an IPv6 address but as a correlator associated to the registration of the IPv6 address. This approach is made possible with 6LoWPAN ND [RFC6775], where the 6LR and the 6LBR maintain a state for each Registered Address. If a CGA is associated with an original 6LoWPAN ND registration and stored in the registration state, then it can be used to validate that any update to the registration state is made by the owner of that CGA.

To achieve this, this specification replaces the EUI-64 address, that is used in 6LoWPAN ND to avoid address duplication, with a CGA address whose ownership can be verified; it also provides new means for the 6LR to validate ownership of the CGA address by the registering device. A node generates one 64-bit CGA address and uses it as Unique Interface ID in the registration of (one or more of) its addresses with the 6LR, which it attaches to and uses as default router. The 6LR validates ownership of the CGA address typically upon creation or update of a registration state, for instance following an apparent movement from a point of attachment to another. The ARO option is modified to indicate that the Unique Interface ID is CGA-based, and through the DAR/DAC exchange, the 6LBR is kept aware that this is the case and whether the 6LR has verified the claim.

CGA generation is based on elliptic curve cryptography (ECC) and signature is calculated using elliptic curve digital signature

algorithm (ECDSA) known to be lightweight, leading to much smaller packet sizes. The resulting protocol is called Lightweight Secure Neighbor Discovery Protocol (LSEND).

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 [RFC3971], [RFC3972], "neighbor Discovery for IP version 6" [RFC4861], "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] where the 6LoWPAN Router (6LR) and the 6LoWPAN Border Router (6LBR) are introduced, and [I-D.chakrabarti-nordmark-6man-efficient-nd], which proposes an evolution of [RFC6775] for a larger applicability.

The draft also conforms to the terms and models described in [RFC5889] and uses the vocabulary and the concepts defined in [RFC4291] for the IPv6 Architecture.

3. Requirements

In this section we state requirements of a secure neighbor discovery protocol for low-power and lossy networks.

The protocol MUST be based on the Neighbor Discovery Optimization for Low-power and Lossy Networks protocol defined in [RFC6775] due to the host-initiated interactions to allow for sleeping hosts, elimination of multicast-based address resolution for hosts, etc.

New options to be added to Neighbor Solicitation messages MUST lead to small packet sizes. Smaller packet sizes facilitate low-power transmission by resource constrained nodes on lossy links.

CGA generation, signature and key hash calculation MUST avoid the use of SHA-1 which is known to have security flaws. In this document, we use SHA-2 instead of SHA-1 and thus avoid SHA-1's flaws.

Public key and signature sizes MUST be minimized and signature calculation MUST be lightweight. In this document we adopt ECC and ECDSA with the P-256 curve in order to meet this requirement.

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 a 6lo "IPv6 over foo" specification exists, as well as Low-Power Wi-Fi.

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.

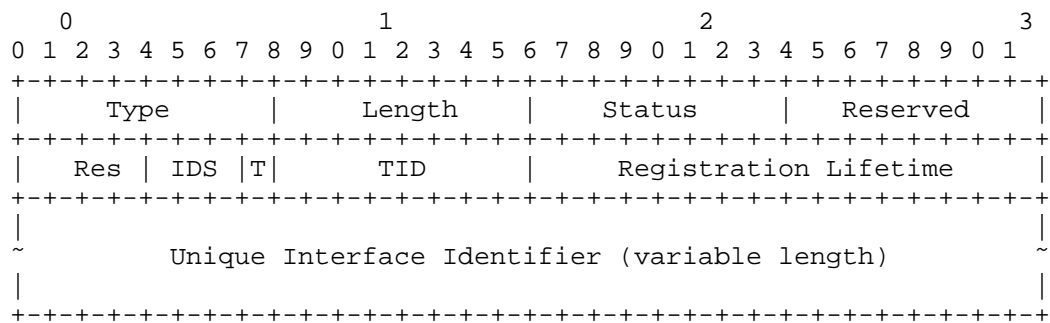
The Address Registration Option used in the ND registration SHOULD be extended to carry the relevant forms of Unique Interface Identifier.

The Neighbour Discovery should specify the formation of a site-local address that follows the security recommendations from [RFC7217].

4. New and Modified Options

4.1. Modified Address Registration Option

The ARO option is modified to transport a CGA-based Unique Interface ID.



Track Forwarding, Transport Mode

Fields:

Type: 33 [RFC6775]

Length: 8-bit unsigned integer. Defined in [RFC6775]. The length of the option (including the type and length fields) in units of 8 bytes. The value 0 is invalid.

Status: 8-bit unsigned integer. Extended from [RFC6775]. Indicates the status of a registration in the NA response. MUST be set to 0 in NS messages. A new status for req-proof of to-be-defined-by-iana (4

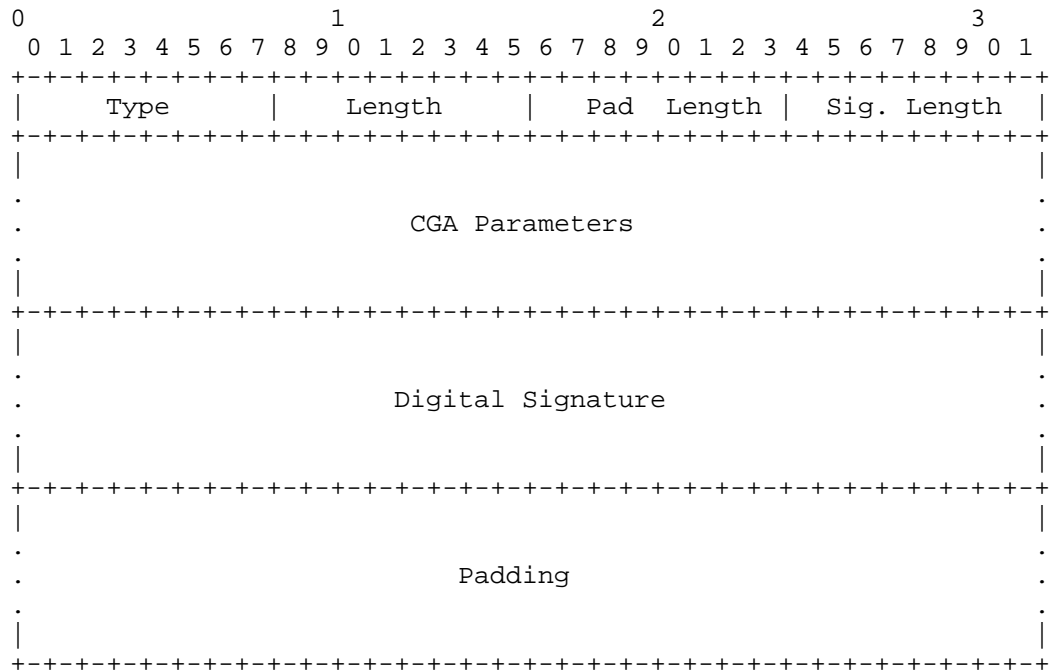
suggested) indicates that the cryptographic material that proves the CGA ownership is requested in a new NS.

- Reserved: 8 bits. This field is unused. It MUST be initialized to zero by the sender and MUST be ignored by the receiver.
- Res: 4 bits. This field is unused. It MUST be initialized to zero by the sender and MUST be ignored by the receiver.
- IDS: Identifier name Space. Indicates the name space for the Unique Interface Identifier. IDS of 0 means EUI-64 UID. A new IDS to be assigned by IANA (a value of 2 is suggested) is defined for CGA-based Unique Interface ID.
- T bit: 1 bit flag. Set if the TID octet is valid.
- TID: 8-bit integer. It is a transaction id maintained by the host and used by the 6LR to indicate the registration that is being validated
- Registration Lifetime: 16-bit unsigned integer. Defined in [RFC6775]. The amount of time in a unit of 60 seconds that the router should retain the Neighbor Cache entry for the sender of the NS that includes this option. A value of zero means to remove the registration.
- Unique Interface Identifier: 8 bytes. May be CGA-based with this specification.

4.2. CGA Parameters and Digital Signature Option

This option contains both CGA parameters and the digital signature.

A summary of the CGA Parameters and Digital Signature Option format is shown below.



Type

TBA1 for CGA Parameters and Digital Signature

Length

The length of the option (including the Type, Length, Pad Length, Signature Length, CGA Parameters, Digital Signature and Padding fields) in units of 8 octets.

Pad Length

The length of the Padding field.

Sig Length

The length of the Digital Signature field.

CGA Parameters

The CGA Parameters field is variable-length containing the CGA Parameters data structure described in Section 4 of [RFC3972].

Digital Signature

The Digital Signature field is a variable length field containing a Elliptic Curve Digital Signature Algorithm (ECDSA) signature (with SHA-256 and P-256 curve of [FIPS-186-3]). Digital signature is constructed as explained in Section 4.4.

Padding

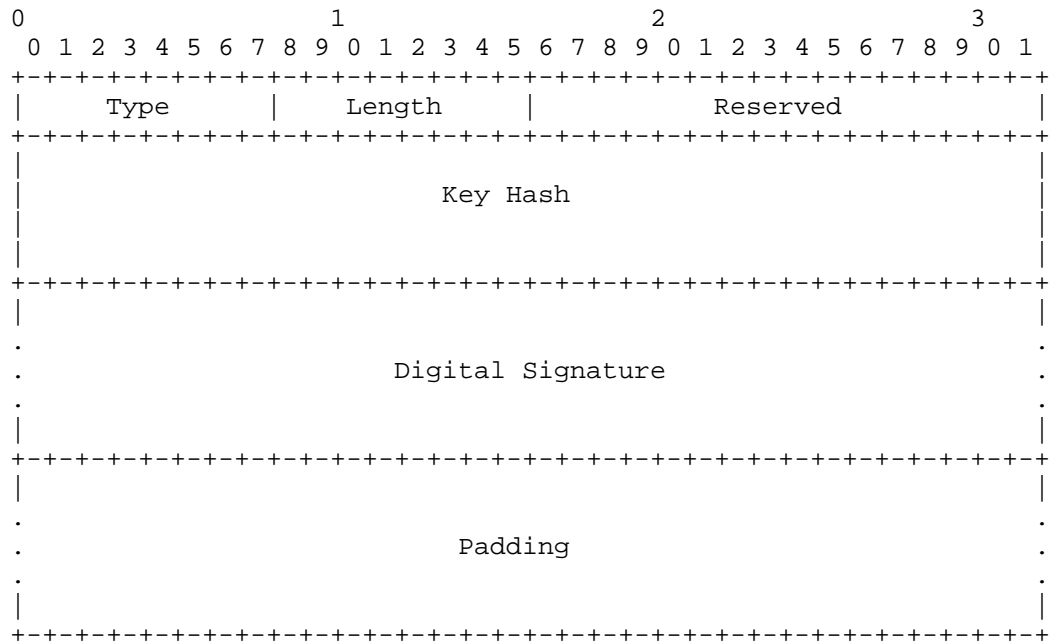
The Padding field contains a variable-length field making the CGA Parameters and Digital Signature Option length a multiple of 8.

4.3. Digital Signature Option

This option contains the digital signature.

A summary of the Digital Signature Option format is shown below. Note that this option has the same format as RSA Signature Option defined in [RFC3971]. The differences are that Digital Signature field carries an ECDSA signature not an RSA signature, and in calculating Key Hash field SHA-2 is used instead of SHA-1.

In the sequence of octets to be signed using the sender's private key includes 128-bit CGA Message Type tag. In LSEND, CGA Message Type tag of 0xE8C47FB7FD2BB85DAB2D31A0F2808B4 MUST be used.



Type

TBA2 for Digital Signature

Length

The length of the option (including the Type, Length, Reserved, Key Hash, Digital Signature and Padding fields) in units of 8 octets.

Key Hash

The Key Hash field is a 128-bit field containing the most significant (leftmost) 128 bits of a SHA-2 hash of the public key used for constructing the signature. This is the same as in [RFC3971] except for SHA-1 which has been replaced by SHA-2.

Digital Signature

Same as in Section 4.2.

Padding

The Padding field contains a variable-length field containing as many bytes long as remain after the end of the signature.

4.4. Calculation of the Digital Signature and CGA Using ECC

Due to the use of Elliptic Curve Cryptography, the following modifications are needed to [RFC3971] and [RFC3972].

The digital signature is constructed by using the sender's private key over the same sequence of octets specified in Section 5.2 of [RFC3971] up to all neighbor discovery protocol options preceding the Digital Signature option containing the ECC-based signature. The signature value is computed using the ECDSA signature algorithm as defined in [SEC1] and hash function SHA-256.

Public Key is the most important parameter in CGA Parameters defined in Section 4.2. Public Key MUST be DER-encoded ASN.1 structure of the type SubjectPublicKeyInfo formatted as ECC Public Key. The AlgorithmIdentifier, contained in ASN.1 structure of type SubjectPublicKeyInfo, MUST be the (unrestricted) id-ecPublicKey algorithm identifier, which is OID 1.2.840.10045.2.1, and the subjectPublicKey MUST be formatted as an ECC Public Key, specified in Section 2.2 of [RFC5480].

Note that the ECC key lengths are determined by the namedCurves parameter stored in ECPParameters field of the AlgorithmIdentifier. The named curve to use is secp256r1 corresponding to P-256 which is OID 1.2.840.10045.3.1.7 [SEC2].

ECC Public Key could be in uncompressed form or in compressed form where the first octet of the OCTET STRING is 0x04 and 0x02 or 0x03, respectively. Point compression using secp256r1 reduces the key size by 32 octets. In LSEND, point compression MUST be supported.

5. Protocol Interactions

Lightweight Secure Neighbor Discovery for Low-power and Lossy Networks (LSEND for LLN) modifies Neighbor Discovery Optimization for Low-power and Lossy Networks [RFC6775] as explained in this section. Protocol interactions are shown in Figure 1.

6LoWPAN Nodes (6LN, or simply "nodes") receive RAs from adjacent 6LRs and generate their own cryptographically generated addresses using elliptic curve cryptography as explained in Section 4.4. The node sends a neighbor solicitation (NS) message with the address registration option (ARO) to 6LR. Such a NS is called an address registration NS.

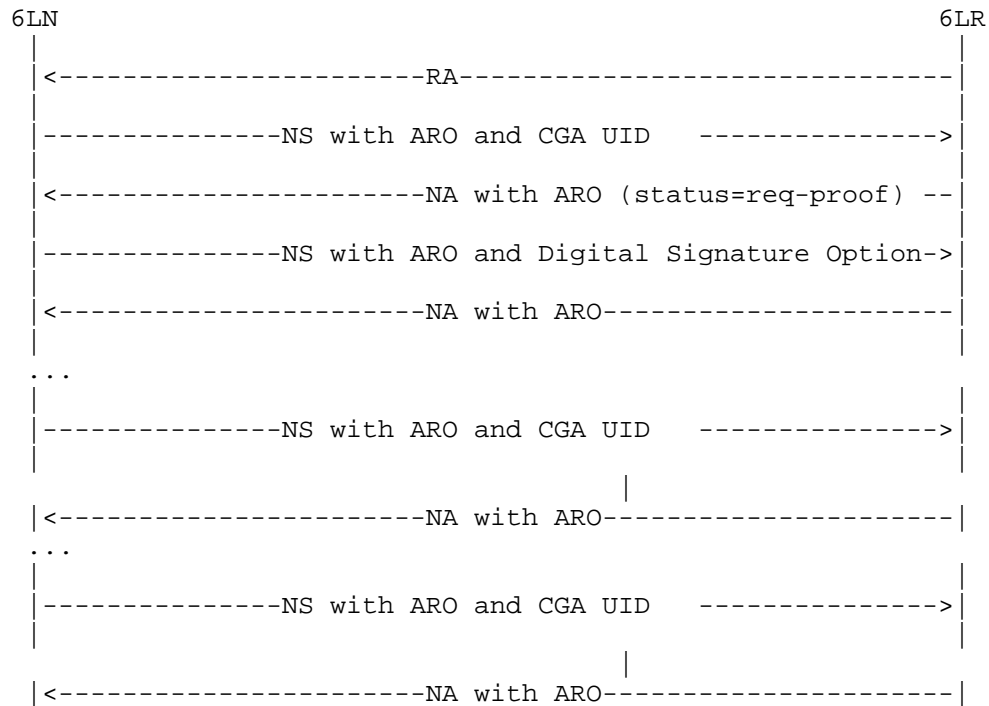


Figure 1: Lightweight SEND for LLN Protocol

6. Optimizations

In this section we present optimizations to the base LSEND defined above. We use EUI-64 identifier instead of source address in CGA calculations. We also extend LSEND operation to 6LoWPAN multihop network.

6.1. Overview

The scope of the present work is a 6LoWPAN Low Power Lossy Network (LLN), typically a stub network connected to a larger IP network via a Border Router called a 6LBR per [RFC6775].

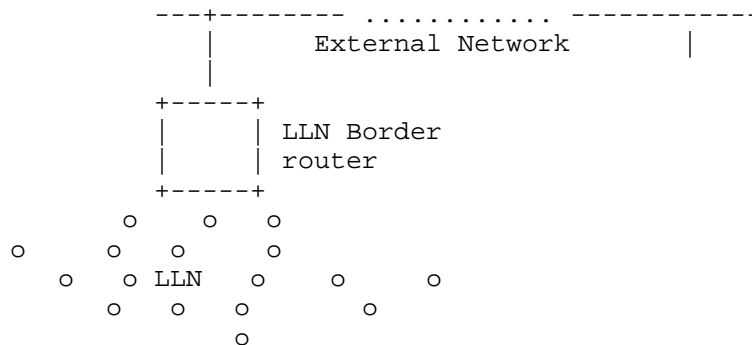


Figure 2: Basic Configuration

The 6LBR maintains a registration state for all devices in the attached LLN, and, in conjunction with the first-hop router (the 6LR), is in position to validate uniqueness and grant ownership of an IPv6 address before it can be used in the LLN. This is a fundamental difference with a classical network that relies on IPv6 address auto-configuration [RFC4862], where there is no guarantee of ownership from the network, and any IPv6 Neighbor Discovery packet must be individually secured [RFC3971].

In a route-over mesh network, the 6LR is directly connected to the host device; this specification expects that peer-wise Layer-2 security is deployed so that all the packets from a particular host are identified as such by the 6LR. The 6LR may be multiple hops away from the 6LBR. Packets are routed between the 6LR and the 6LBR via other 6LRs; this specification expects that a chain of trust is established so that a packet that was validated by the first 6LR can be safely routed by the next 6LRs and 6LBR.

The [I-D.ietf-6tisch-architecture] suggests to use RPL [RFC6550] as the routing protocol between the 6LRs and the 6LBR, and to leverage [I-D.chakrabarti-nordmark-6man-efficient-nd] to extend the LLN in a larger multilink subnet [RFC4903]. In that model, a registration flow happens as shown in Figure 3:

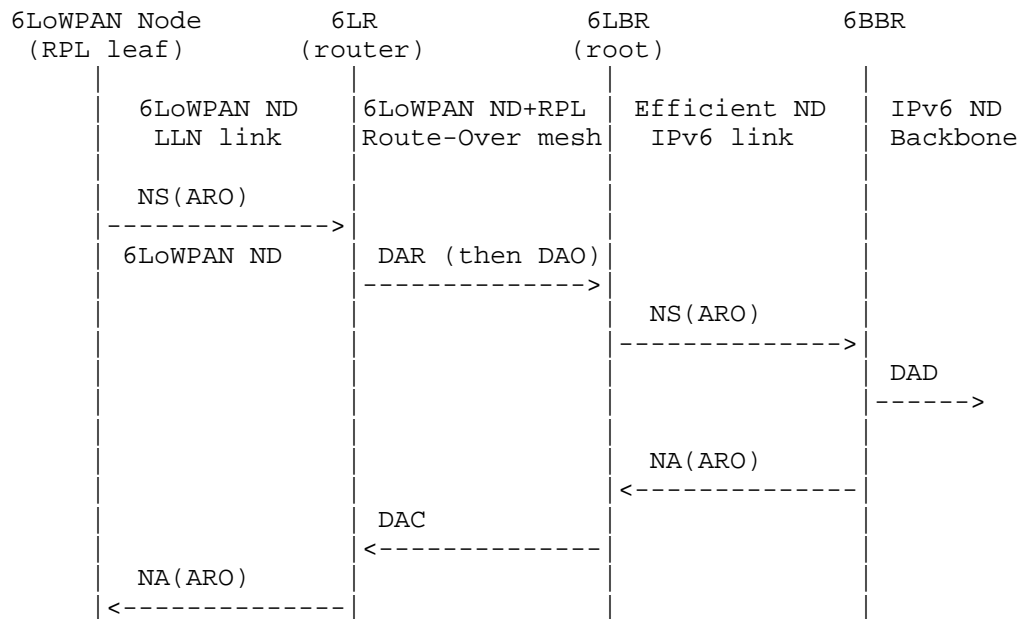


Figure 3: (Re-)Registration Flow over Multi-Link Subnet

A new device that joins the network auto-configures and address and performs an initial registration to an on-link 6LR with an NS message that carries a new Address Registration Option (ARO) [RFC6775]. The 6LR validates with address with the central 6LBR using a DAR/DAC exchange, and the 6LR confirms (or infirms) the address ownership with an NA message that also carries an Address Registration Option.

The registration mechanism in [RFC6775] was created for the original purpose of Duplicate Address Detection (DAD), whereby use of an address would be granted as long as the address is not already present in the subnet. But [RFC6775] does not require that the 6LR use the registration for source address validation (SAVI).

In order to validate address ownership, that mechanism enables the 6LBR to correlate further claims for a registered address with the device to which it is granted, based on a Unique Interface IDentifier (UID) that is derived from the MAC address of the device (EUI-64).

The limit of the mechanism in [RFC6775] is that it does not enable to prove the UID itself, so any node connected to the subnet and aware of the address/UID mapping may effectively fake the same UID and steal an address.

This draft uses a Cryptographically Generated Address (CGA) [RFC3972] as an alternate UID for the registration. Proof of ownership of the UID is passed with the first registration to a given 6LR, and enforced at the 6LR, which validates the proof. With this new operation, the 6LR allows only packets from a connected host if the connected host owns the registration of the source address of the packet.

If a chain of trust is present between the 6LR and the 6LBR, then there is no need to propagate the proof of ownership to the 6LBR. All the 6LBR need to know is that this particular UID is based on CGA, so as to enforce that any update via a different 6LR is also based on CGA.

6.2. Protocol Operations

Digital signature and CGA are calculated over EUI-64 or interface id of the node. It is only done initially at once not repeated with every message the node sends. The calculation does not change even if the node has a new address since EUI-64 does not change. This means that this CGA can be used to claim multiple targets. The calculation is ECC based as described in Section 4.4.

Protocol interactions are as defined in Section 5. The address registration NS message contains CGA Parameters and Digital Signature Option defined in Section 4.2. The node MUST set the Extended Unique Interface IDentifier (EUI-64) field [Guide] in ARO to the cryptographically generated address. The Subnet Prefix field of CGA Parameters MUST be set to the 64-bit prefix in the RA message received from 6LBR. Source address MUST be set to the prefix concatenated with the node's cryptographically generated address. The Public Key field of CGA Parameters MUST be set to the node's ECC Public Key.

CGA calculated may need to be modified before it is used as EUI-64. The b2 bit or U/L or "u" bit MUST be set to zero for globally unique and b1 bit or I/G or "g" bit MUST be set to zero for unicast before using it in IPv6 address as the interface identifier. In LSEND, senders and receivers ignore any differences in the three leftmost bits and in bits 6 and 7 (i.e., the "u" and "g" bits) in the interface identifiers [RFC3972].

The Target Address field in NS message is set to the prefix concatenated with the node's cryptographically generated address. This address does not need duplicate address detection as EUI-64 is globally unique. So a host cannot steal an address that is already registered unless it has the key for the EUI-64. The same EUI-64 can thus be used to protect multiple addresses e.g. when the node

receives a different prefix. The node adds CGA Parameters (including Public Key) and Digital Signature Option defined in Section 4.2 into NS message. The node sends the address registration option (ARO) which is set to the CGA calculated.

Protocol interactions given in Figure 1 are modified a bit in that Digital Signature option with the public key and ARO are passed to and stored by the 6LR/6LBR on the first NS and not sent again in the next NS.

The 6LR/6LBR ensures first-come/first-serve by storing the ARO and the cryptographic material correlated to the target being registered. Then, if the node is the first to claim any address it likes, then it becomes owner of that address and the address is bound to the CGA in the 6LR/6LBR registry. This procedure avoids the constrained device to compute multiple keys for multiple addresses. The registration process allows the node to tie all the addresses to the same EUI-64 and have the 6LR/6LBR enforce first come first serve after that.

6.3. Multihop Operation

In multihop 6LoWPAN, 6LBR sends RAs with prefixes downstream and it is the 6LR that receives and relays them to the nodes. 6LR and 6LBR communicate with the ICMPv6 Duplicate Address Request (DAR) and the Duplicate Address Confirmation (DAC) messages. The DAR and DAC use the same message format as NS and NA with different ICMPv6 type values.

In LSEND we extend DAR/DAC messages to carry CGA Parameters and Digital Signature Option defined in Section 4.2.

In a multihop 6LoWPAN, the node exchanges the messages shown in Figure 3. 6LBR must be aware of who owns an address (EUI-64) to defend the first user if there is an attacker on another 6LR. Because of this the content that the source signs and the signature needs to be propagated to the 6LBR in DAR message. For this purpose we need the DAR message sent by 6LR to 6LBR MUST contain CGA Parameters and Digital Signature Option carrying the CGA that the node calculates and its public key. DAR message also contains ARO.

It is possible that occasionally, 6LR may miss the node's CGA (that it received in ARO) or the crypto information (that it received in CGA Parameters and Digital Signature Option). 6LR should be able to ask for it again. This is done by restarting the exchanges shown in Figure 1. The result enables 6LR to refresh CGA and public key information that was lost. 6LR MUST send DAR message with CGA Parameters and Digital Signature Option and ARO to 6LBR. 6LBR as a

reply forms a DAC message with the information copied from the DAR and the Status field is set to zero. With this exchange, the 6LBR can (re)validate and store the CGA and crypto information to make sure that the 6LR is not a fake.

7. Security Considerations

The same considerations regarding the threats to the Local Link Not Covered (as in [RFC3971]) apply.

The threats discussed in Section 9.2 of [RFC3971] are countered by the protocol described in this document as well.

As to the attacks to the protocol itself, denial of service attacks that involve producing a very high number of packets are deemed unlikely because of the assumptions on the node capabilities in low-power and lossy networks.

8. IANA considerations

This document defines two new options to be used in neighbor discovery protocol messages and new type values for CGA Parameters and Digital Signature Option (TBA1) and Digital Signature Option (TBA2) need to be assigned by IANA.

This document defines 0xE8C47FB7FD2BB885DAB2D31A0F2808B4 for LSEND CGA Message Type Tag.

9. Acknowledgements

TBD.

10. References

10.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC3756] Nikander, P., Kempf, J., and E. Nordmark, "IPv6 Neighbor Discovery (ND) Trust Models and Threats", RFC 3756, May 2004.
- [RFC3971] Arkko, J., Kempf, J., Zill, B., and P. Nikander, "SEcure Neighbor Discovery (SEND)", RFC 3971, March 2005.
- [RFC3972] Aura, T., "Cryptographically Generated Addresses (CGA)", RFC 3972, March 2005.

- [RFC4291] Hinden, R. and S. Deering, "IP Version 6 Addressing Architecture", RFC 4291, February 2006.
- [RFC4861] Narten, T., Nordmark, E., Simpson, W., and H. Soliman, "Neighbor Discovery for IP version 6 (IPv6)", RFC 4861, September 2007.
- [RFC4862] Thomson, S., Narten, T., and T. Jinmei, "IPv6 Stateless Address Autoconfiguration", RFC 4862, September 2007.
- [RFC4903] Thaler, D., "Multi-Link Subnet Issues", RFC 4903, June 2007.
- [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, August 2007.
- [RFC5480] Turner, S., Brown, D., Yiu, K., Housley, R., and T. Polk, "Elliptic Curve Cryptography Subject Public Key Information", RFC 5480, March 2009.
- [RFC5889] Baccelli, E. and M. Townsley, "IP Addressing Model in Ad Hoc Networks", RFC 5889, September 2010.
- [RFC6550] Winter, T., Thubert, P., 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, March 2012.
- [RFC6775] Shelby, Z., Chakrabarti, S., Nordmark, E., and C. Bormann, "Neighbor Discovery Optimization for IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs)", RFC 6775, November 2012.
- [RFC7217] Gont, F., "A Method for Generating Semantically Opaque Interface Identifiers with IPv6 Stateless Address Autoconfiguration (SLAAC)", RFC 7217, April 2014.
- [SEC1] "Standards for Efficient Cryptography Group. SEC 1: Elliptic Curve Cryptography Version 2.0", May 2009.
- [Guide] "Guidelines for 64-bit global Identifier (EUI-64TM)", November 2012, <<http://standards.ieee.org/develop/regauth/tut/eui64.pdf>>.

[ANSIX9.62]

"American National Standards Institute (ANSI), ANS
X9.62-2005: The Elliptic Curve Digital Signature Algorithm
(ECDSA)", November 2005.

10.2. Informative references

[SEC2] "Standards for Efficient Cryptography Group. SEC 2:
Recommended Elliptic Curve Domain Parameters Version 2.0",
January 2010.

[FIPS-186-3]

"National Institute of Standards and Technology, "Digital
Signature Standard", June 2009.

[NIST-ST]

"National Institute of Standards and Technology, "NIST
Comments on Cryptanalytic Attacks on SHA-1", January
2009,
<<http://csrc.nist.gov/groups/ST/hash/statement.html>>.

[I-D.rafiee-6man-ssas]

Rafiee, H. and C. Meinel, "A Simple Secure Addressing
Scheme for IPv6 AutoConfiguration (SSAS)", draft-rafiee-
6man-ssas-11 (work in progress), September 2014.

[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.ietf-6tisch-architecture]

Thubert, P., Watteyne, T., Struik, R., and M. Richardson,
"An Architecture for IPv6 over the TSCH mode of IEEE
802.15.4e", draft-ietf-6tisch-architecture-06 (work in
progress), March 2015.

Authors' Addresses

Behcet Sarikaya (editor)
Huawei USA
5340 Legacy Dr. Building 3
Plano, TX 75024

Email: sarikaya@ieee.org

Frank Xia
Huawei Technologies Co., Ltd.
101 Software Avenue, Yuhua District
Nanjing, Jiangsu 210012, China

Phone: ++86-25-56625443
Email: xiayangsong@huawei.com

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

6Lo
Internet-Draft
Intended status: Informational
Expires: October 6, 2016

P. Thubert, Ed.
cisco
P. van der Stok
consultant
April 4, 2016

Requirements for an update to 6LoWPAN ND
draft-thubert-6lo-rfc6775-update-reqs-07

Abstract

Work presented at the ROLL, 6lo, 6TiSCH and 6MAN Working Groups suggest that enhancements to the 6LoWPAN ND mechanism are now needed. This document elaborates on those requirements and suggests approaches to serve them.

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 October 6, 2016.

Copyright Notice

Copyright (c) 2016 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	2
2. Terminology	3
3. Overview	4
4. Requirements	6
4.1. Requirements Related to Mobility	6
4.2. Requirements Related to Routing Protocols	7
4.3. Requirements Related to the Variety of Low-Power Link types	8
4.4. Requirements Related to Proxy Operations	8
4.5. Requirements Related to Security	9
4.6. Requirements Related to Scalability	10
5. Security Considerations	11
6. IANA Considerations	11
7. Acknowledgments	11
8. References	11
8.1. Normative References	11
8.2. Informative References	13
Appendix A. Suggested Changes to Protocol Elements	14
A.1. ND Neighbor Solicitation (NS)	14
A.2. ND Router Advertisement (RA)	14
A.3. RPL DODAG Information Object (DIO)	15
A.4. ND Enhanced Address Registration Option (EARO)	15
Authors' Addresses	16

1. Introduction

A number of use cases, including the Industrial Internet, require a large scale deployment of sensors that can not be realized with wires and is only feasible over wireless Low power and Lossy Network (LLN) technologies. When simpler hub-and-spoke topologies are not sufficient for the expected throughput and density, mesh networks are deployed, which implies the routing of packets over the mesh, operated at either Layer-2 or Layer-3.

For routing over a mesh at layer-3, the IETF has designed the IPv6 Routing Protocol over LLN (RPL) [RFC6550].

To assign routable addresses, DHCPv6 is still a viable option in LLNs. However, the IETF standard that supports address assignment specifically for LLNs is 6LoWPAN ND, the Neighbor Discovery Optimization for Low-power and Lossy Networks [RFC6775]. 6LoWPAN ND was designed as a stand-alone mechanism separately from its IETF routing counterpart, the IPv6 Routing Protocol for Low power and Lossy Networks [RFC6550] (RPL), and the interaction between the 2 protocols was not defined.

The 6TiSCH WG is now considering an architecture [I-D.ietf-6tisch-architecture] whereby a 6LoWPAN ND host could connect to the Internet via a RPL Network, but this requires additions to the 6LoWPAN ND protocol to support mobility and reachability in a secured and manageable environment.

At the same time, new work at 6MAN on Efficiency aware IPv6 Neighbor Discovery Optimizations [I-D.chakrabarti-nordmark-6man-efficient-nd] suggests that 6LoWPAN ND can be extended to other types of networks on top of the Low power and Lossy Networks (LLNs) for which it was already defined. The value of such extension is especially apparent in the case of mobile wireless devices, to reduce the multicast operations that are related to classical ND ([RFC4861], [RFC4862]) and plague the wireless medium. In this context also, there is a need for additions to 6LoWPAN ND.

The Optimistic Duplicate Address Detection [RFC4429] (ODAD) specification details how an address can be used before a Duplicate Address Detection (DAD) is complete, and insists that an address that is TENTATIVE should not be associated to a Source Link-Layer Address Option in a Neighbor Solicitation message. Applying this rule to 6LoWPAN ND implies another change to its specification.

In [I-D.richardson-6tisch--security-6top], the 6tisch working group considers the use of layer-2 security. It develops a network bootstrap protocol that provides secure link connections at the same rate that nodes are discovered. This approach needs the presence of a routing protocol to route packets from a joining node to a security providing node (e.g. a PCE or commissioning tool).

This document suggests a limited evolution to [RFC6775] so as to allow operation of a 6LoWPAN ND node while a routing protocol (in first instance RPL) is present and operational. It also suggests a more generalized use of the information in the ARO option of the ND messages outside the strict LLN domain, for instance over a converged backbone.

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 "Transmission of IPv6 Packets over IEEE 802.15.4 Networks" [RFC4944].

Additionally, this document uses terminology from 6TiSCH [I-D.ietf-6tisch-terminology] and ROLL [RFC7102].

3. Overview

This document is mostly motivated by the work ongoing in the 6TiSCH working group. The 6TiSCH architecture [I-D.ietf-6tisch-architecture] draft explains the network architecture of a 6TiSCH network. This architecture is used for the remainder of this document.

The scope of the 6TiSCH Architecture is a Backbone Link that federates multiple LLNs (mesh) as a single IPv6 Multi-Link Subnet. Each LLN in the subnet is anchored at a Backbone Router (6BBR). The Backbone Routers interconnect the LLNs over the Backbone Link and emulate that the LLN nodes are present on the Backbone thus creating a so-called: Multi-Link Subnet. An LLN node can move freely from an LLN anchored at a Backbone Router to another LLN anchored at the same or a different Backbone Router inside the Multi-Link Subnet and conserve its addresses.

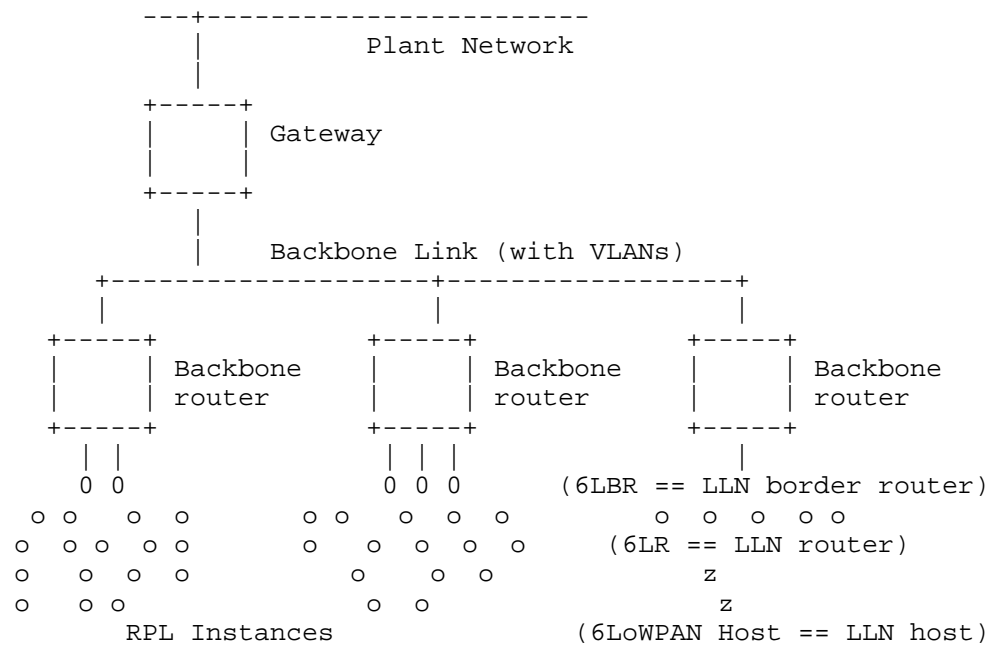


Figure 1: 6TiSCH architecture

The 6LBR is the border router that is placed between the LLN and nodes outside the LLN. The 6LBR is logically separated from the 6BBR that is used to connect the LLN to the backbone. The 6LBR can use Efficient ND as the interface to register an LLN node in its topology to the 6BBR for whatever operation the 6BBR performs, such as ND proxy operations, or injection in a routing protocol. It results that, as illustrated in Figure 2, the periodic signaling could start at the leaf node with 6LoWPAN ND, then would be routed to the 6LBR, and then with Efficient-ND to the 6BBR. Efficient ND being an adaptation of 6LoWPAN ND, it makes sense to keep those two homogeneous in the way they use the source and the target addresses in the Neighbor Solicitation (NS) messages for registration, as well as in the options that they use for that process.

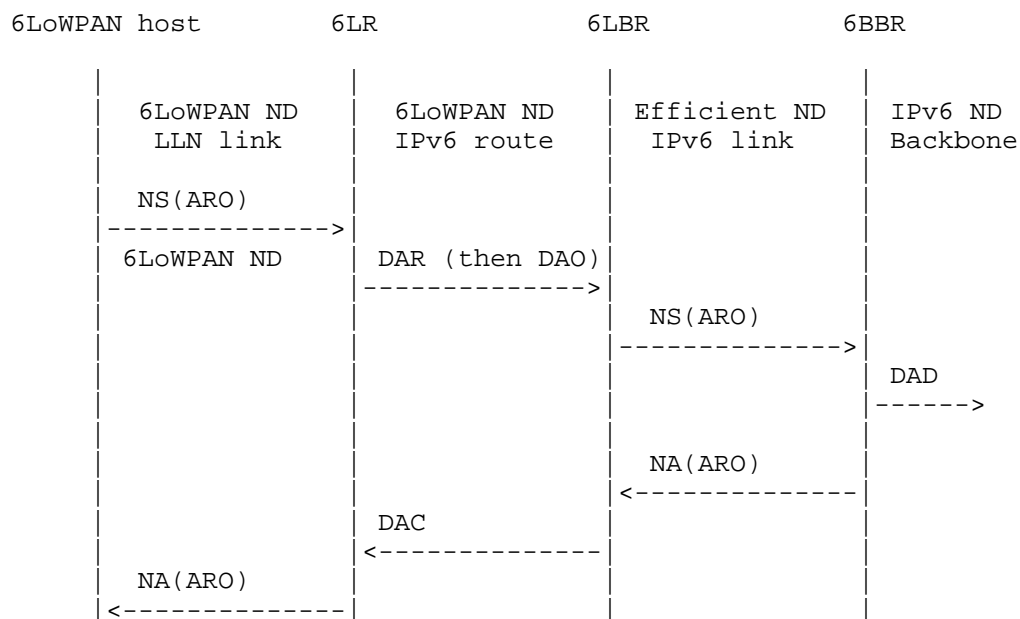


Figure 2: (Re-)Registration Flow over Multi-Link Subnet

As the network builds up, a LoWPAN host starts as a leaf to join the LLN, and may later turn into a 6LR, so as to accept other nodes to recursively join the LLN.

Section 5 of the 6TiSCH architecture [I-D.ietf-6tisch-architecture] provides more information on the need to update the protocols that sustain the requirements in the next section.

4. Requirements

4.1. 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.

4.2. 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.wijnands-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 DAOSequence 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.

4.3. 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 [I-D.brandt-6man-lowpanz], Master-Slave/Token-Passing [I-D.ietf-6lo-6lobac], DECT Ultra Low Energy [I-D.ietf-6lo-dect-ule], Near Field Communication [I-D.hong-6lo-ipv6-over-nfc], as well as IEEE1901.2 Narrowband Powerline Communication Networks [I-D.popa-6lo-6loplc-ipv6-over-ieee19012-networks] and BLUETOOTH(R) Low Energy [I-D.ietf-6lo-btle].

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].

4.4. 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.

4.5. 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.

4.6. 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.

5. Security Considerations

This specification expects that the link layer is sufficiently protected, either by means of 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 tampering with or replaying the RA messages. Still, Section 4.5 has a requirement for a mutual authentication and authorization for a role for 6LRs, 6LBRs and 6BBRs.

This documents also suggests in Appendix A.4 that a 6LoWPAN Node could form a single Unique Interface ID (CUID) based on cryptographic techniques similar to CGA. The CUID would be used as Unique Interface Identifier in the ARO option and new Secure ND procedures would be proposed to use it as opposed to the source IPv6 address to secure the binding between an Address and its owning Node, and enforce First/Come-First/Serve at the 6LBR.

6. IANA Considerations

This draft does not require an IANA action.

7. Acknowledgments

The author wishes acknowledge the contributions by Samita Chakrabarti, Erik Normark, JP Vasseur, Eric Levy-Abegnoli, Patrick Wetterwald, Thomas Watteyne, and Behcet Sarikaya.

8. References

8.1. 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>>.
- [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>>.
- [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>>.
- [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>>.
- [RFC6217] Ritter, T., "Regional Broadcast Using an Atmospheric Link Layer", RFC 6217, DOI 10.17487/RFC6217, April 2011, <<http://www.rfc-editor.org/info/rfc6217>>.
- [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>>.
- [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>>.
- [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>>.

8.2. Informative References

- [I-D.brandt-6man-lowpanz]
Brandt, A. and J. Buron, "Transmission of IPv6 packets over ITU-T G.9959 Networks", draft-brandt-6man-lowpanz-02 (work in progress), June 2013.
- [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.hong-6lo-ipv6-over-nfc]
Hong, Y. and J. Youn, "Transmission of IPv6 Packets over Near Field Communication", draft-hong-6lo-ipv6-over-nfc-03 (work in progress), November 2014.
- [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-btle]
Nieminen, J., Savolainen, T., Isomaki, M., Patil, B., Shelby, Z., and C. Gomez, "IPv6 over BLUETOOTH(R) Low Energy", draft-ietf-6lo-btle-17 (work in progress), August 2015.
- [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-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.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.richardson-6tisch--security-6top]

Richardson, M., "6tisch secure join using 6top", draft-
richardson-6tisch--security-6top-05 (work in progress),
November 2015.

[I-D.wijnands-bier-architecture]

Wijnands, I., Rosen, E., Dolganow, A., Przygienda, T., and
S. Aldrin, "Multicast using Bit Index Explicit
Replication", draft-wijnands-bier-architecture-05 (work in
progress), March 2015.

Appendix A. Suggested Changes to Protocol Elements

A.1. ND Neighbor Solicitation (NS)

The NS message used for registration should use a source address that respects the rules in [RFC6775], [RFC4861], and [RFC4429] for DAD. The SLLA Option may be present but only if the address passed DAD, and it is used to allow the 6LR to respond as opposed to as a registration mechanism.

The address that is being registered is the target address in the NS message and the TLLA Option must be present.

A.2. ND Router Advertisement (RA)

[I-D.chakrabarti-nordmark-6man-efficient-nd] adds an 'E' bit in the Router Advertisement flag, as well as a new Registrar Address Option (RAO). These fields are probably pertinent to LLNs inclusion into a revised 6LoWPAN ND should be studied. If the new 6LoWPAN flows require a change of behaviour (e.g. registering the Target of the NS message) then the RA must indicate that the router supports the new capability, and the NS must indicate that the Target is registered as opposed to the Source in an unequivocal fashion.

There is some amount of duplication between the options in the RPL DIO [RFC6550] and the options in the ND RA messages. At the same time, there are a number of options, including the 6LoWPAN Context Option (6CO) [RFC6775], the MTU and the SLLA Options [RFC4861] that can only be found in the RA messages. Considering that these options are useful for a joining node, the recommendation would be to associate the RA messages to the join beacon, and make them rare when the network is stable. On the other hand, the DIO message is to be used as the propagated heartbeat of the RPL network and provide the sense of time and liveliness.

RAs should also be issued and the information therein propagated when a change occurs in the information therein, such as a router or a prefix lifetime.

A.3. RPL DODAG Information Object (DIO)

If the RPL root serves as 6LBR, it makes sense to add at least a bit of information in the DIO to signal so. A Registrar Address Option (RAO) may also be considered for addition.

A.4. ND Enhanced Address Registration Option (EARO)

The ARO option contains a Unique ID that is supposed to identify the device across multiple registrations. It is envisioned that the device could form a single CGA-based Unique Interface ID (CUID) to securely bind all of its addresses. The CUID would be used as Unique Interface Identifier in the ARO option and to form a Link-Local address that would be deemed unique regardless of the Link type. Provided that the relevant cryptographic material is passed to the 6LBR upon the first registration or on-demand at a later time, the 6LBR can validate that a Node is effectively the owner of a CUID, and ensure that the ownership of an Address stays with the CUID that registered it first.

This option is designed to be used with standard NS and NA messages between backbone Routers as well as between nodes and 6LRs over the LLN and between the 6LBR and the 6BBR over whatever IP link they use to communicate.

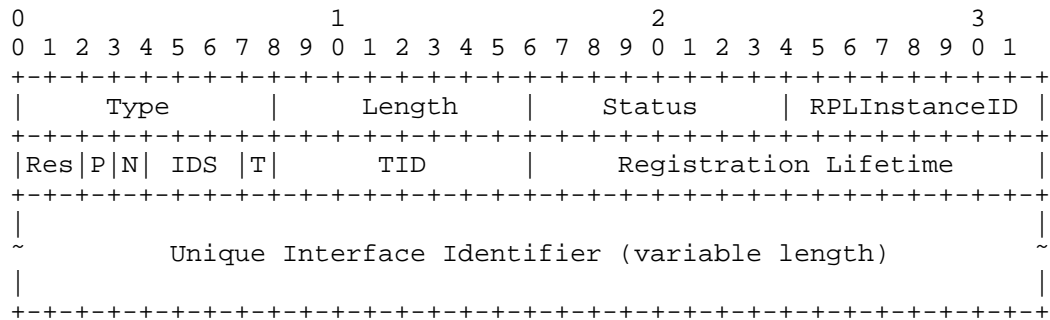


Figure 3: EARO

The representation above is based on [I-D.chakrabarti-nordmark-6man-efficient-nd]. Only the proposed changes from that specification are discussed below but the expectation is that 6LoWPAN ND and Efficient ND converge on the ARO format.

Status: 8-bit integer. A new value of 3 is suggested to indicate a rejection due to an obsolete TID, typically an indication of a movement.

RPLInstanceID: 8-bit integer. This field is set to 0 when unused. Otherwise it contains the RPLInstanceID for which this address is registered, as specified in RPL [RFC6550], and discussed in particular in section 3.1.2.

P: One bit flag. When the bit is set, the address being registered is Target of the NS as opposed to the Source, for instance to enable ND proxy operation.

N: One bit flag. Set if the device moved. If not set, the 6BBR will refrain from sending gratuitous NA(0) or other form of distributed ND cache clean-up over the backbone. For instance, the flag should be reset after the DAD operation upon address formation.

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

Peter van der Stok
consultant

Phone: +31-492474673 (Netherlands), +33-966015248 (France)
Email: consultancy@vanderstok.org
URI: www.vanderstok.org