```
6lo                                                       S. Chakrabarti
Internet-Draft                                                  Ericsson
Updates: 4944, 6282 (if approved)                          G. Montenegro
Intended status: Standards Track                               Microsoft
Expires: January 7, 2016                                        R. Droms
                                                                   Cisco
                                                             J. Woodyatt
                                                                    Nest
                                                            July 6, 2015
```

IANA Registry for 6lowpan Additional Dispatch Bytes
draft-chairs-6lo-dispatch-iana-registry-00

Abstract

RFC4944 defines ESC dispatch type for additional dispatch bytes in the 6lowpan header. The value of ESC byte has been updated by RFC6282. However, the usage of ESC extension bytes has not been defined in RFC6282 and RFC4944. The purpose of this document is to define the usage of ESC extension bytes. It also records the initial values for extended dispatch values and requests corresponding IANA actions.

Status of this Memo

Copyright Notice

Table of Contents

1. Introduction

[RFC4944] section 5.1 defines the dispatch header and types. The ESC type is defined for using additional dispatch bytes in the 6lowpan header. RFC 6282 modifies the value of the ESC dispatch type and it is recorded in IANA registry [6LOWPAN-IANA]. However, the bytes and usage following the ESC byte are not defined in either [RFC4944] and [RFC6282]. However, in recent years with 6lowpan deployments, the implementations and Standards organizations have started using the ESC bytes and a co-ordination between the respective organizations and IETF/IANA are needed.

The following sections describe the ITU-T specification for ESC dispatch byte code points for the record and propose the use of ESC extension bytes in the future. The document also requests IANA actions for the first extension byte following the ESC byte.

2. Terminology

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

3. Usage of ESC dispatch bytes

The ESC byte [01 000000] is modified in RFC 6282[RFC6282] and [RFC4944] first introduces this dispatch header type for extension of dispatch bytes for different usage of 6lowpan applications.

For example, a dispatch header type (ex: LOWPAN_HC1, MESH etc.) might need some special handling of each packet for classification.

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

```
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 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|0 1| ESC       | Ext Type      | Extended Dispatch Payload
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 1: Frame Format with ESC Byte

ESC: The left most byte is the ESC dispatch type containing ’0100000’

Extension Type(ET): It is the first byte following the ESC byte. Extension type defines the payload for the additional dispatch bytes. The values are from 0 to 255. Value 0 and 255 are reserved for future use. These values are assigned by IANA. The extension types appear in the sequence [ESC][extension type], as opposed to the dispatch values which appear by themselves as [dispatch value] with no preceding ESC. Thus, extension types and dispatch values are orthogonal code spaces.

Extended Dispatch Payload(EDP): This part of frame format must be defined by the corresponding extension type. A specification is required to define each usage of extension type and its corresponding Extension Payload.

Note that section 5.1 in RFC4944 indicates that the Extension Type field may contain additional dispatch values (larger than 63). Note that the new dispatch type MUST NOT modify the behavior of existing dispatch types for the sake of interoperability.

3.1. Open Issues

Legacy node behavior: When a legacy 6lowpan node receives packets with ESC bytes or nodes receiving ESC bytes it does not understand, what should be its behavior? Two alternatives: 1) discard the 6lowpan packet 2) ignore the ESC bytes.

Sequence Of dispatch bytes and ESC bytes: TBD

3.2. Example: ITU-T G.9903 ESC type usage

[G3-PLC] provides native mesh under functionalities. The ESC dispatch type is used with the command frames specified in figure 9-12 and Table 9-35 in [G3-PLC] . The command ID values are 0x01 to 0x1F.

The frame format is defined as follows:

```
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 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|0 1| ESC       |  Command ID   | Command Payload
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 2: G.9903 Frame Format with ESC Byte

4. IANA Considerations

This document requests IANA to register the ’Extension Type’ values as per the policy ’Specification Required’[RFC5226] as specified in this document which follows the same policy as in the IANA section of [RFC4944]. For each Extension Type(except the Reserved values)the specification MUST define corresponding Extended Dispatch Payload frame bytes for the receiver implementation to read the ESC bytes with interoperability.

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

Value	Description	Reference
0	Reserved for future use	This document
1-31	Used by ITU-T G.9903 command ID	ITU-T G.9903 [G3-PLC]
32-254	Unassigned	This document
255	Reserved for future use	This document

Figure 3: Initial Values for IANA Registry

5. Security Considerations

There is no additional security threats due to the assignments of ESC byte usage described in this document. However, this document forbids defining any extended dispatch values or extension types that modifies the behavior of existing Dispatch types.

6. Acknowledgements

The authors would like to thank the members of the 6lo WG members for the comments in the mailing list. Many thanks to Carsten Bormann, Ralph Droms, Thierry Lys, Cedric Lavenu, Pascal Thubert for their discussions regarding resolving the bits allocation issues which led to this document.

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.

[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.

7.2. Informative References

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

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

[RFC5226] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 5226, May 2008.

Authors’ Addresses

Samita Chakrabarti
Ericsson
300 Holger Way
San Jose, CA
US

Phone: +1 408 750 5843
Email: samita.chakrabarti@ericsson.com

Gabriel Montenegro
Microsoft
Seattle
US

Email: gabriel.montenegro@microsoft.com

Ralph Droms
Cisco
USA

Email: rdroms@cisco.com

James Woodyatt
Nest
Mountain View, CA
USA

Email: jhw@netstlabs.com

6Lo Working Group   L. Del Carpio Vega
Internet-Draft   M. Robles
Intended status: Standards Track   R. Morabito
Expires: April 21, 2016   Ericsson
                                                        October 19, 2015

IPv6 over 802.11ah
draft-delcarpio-6lo-wlanah-01

Abstract

IEEE 802.11 is an established Wireless LAN (WLAN) technology which provides radio connectivity to a wide range of devices. The IEEE 802.11ah amendment defines a WLAN system operating at sub 1 GHz license-exempt bands designed to operate with low-rate/low-power consumption. This amendment supports large number of stations and extends the radio coverage to several hundreds of meters. This document describes how IPv6 is transported over 802.11ah using 6LoWPAN techniques.

Status of This Memo

This Internet-Draft will expire on April 21, 2016.

Copyright Notice

Table of Contents

1. Introduction

IEEE 802.11 [IEEE802.11], also known as Wi-Fi, is an established Wireless LAN (WLAN) technology operating in unlicensed Industrial, Scientific and Medical (ISM) bands. Its IEEE 802.11ah [IEEE802.11ah] amendment is tailored for Internet of Things (IoT) use-cases and at the moment of writing this draft it is in the final stages of IEEE standardization.

IEEE 802.11ah operates in the Sub-1 GHz spectrum which helps reducing the power consumption. It also supports a larger number of stations on a single Basic Service Set (BSS) and it provides power-saving mechanisms that allow radio stations to sleep in order to save power.

However, the system achieves lower throughput compared to 802.11n/ac amendments.

IEEE 802.11 specifies only the MAC and PHY layers of the radio technology. In other words, 802.11 does not specify a networking layer but it is compatible with commonly used internet protocol such as IPv4 and IPv6. As 802.11ah is a low-rate/low-power technology, the communication protocols used above MAC should also take power-efficiency into consideration. This motivates the introduction of 6LoWPAN techniques [RFC4944] [RFC6282] for efficient transport of IPv6 packets over IEEE 802.11ah radio networks.

This document specifies how to use 6LoWPAN techniques for 802.11ah.

2. Terminology and Language Requirements

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].

Terminology from 802.11ah:

Station (STA): defined in 802.11-2012 [IEEE802.11-2012] as a wireless station which is an addressable unit.

Sensor-STA: defined in 802.11ah as a device having low-power consumption requirements. This device might be a battery operated device.

Non-sensor STA: defined in 802.11ah as device which usually does not have low-power consumption requirements.

In this document, any type STA (sensor STA/non-sensor STA) is associated with a 6LoWPAN Node(6LN).

Access Point (AP): entity maintaining the WLAN Basic Service Set (BSS) and it is associated with the 6LoWPAN Border Router (6LBR). It is assumed that APs are connected to the power-line.

The terms 6LoWPAN Router (6LR) and 6LoWPAN Border Router (6LBR) are defined as in [RFC6775] and in this context 6LoWPAN Nodes (6LN) do not refer to a router (Access Point), just to a host (STA).

3. Overview of 802.11ah

The IEEE 802.11 technology uses the unlicensed spectrum in different ISM bands, using CSMA/CA techniques. Specifically 802.11ah is designed to operate in ISM band below Sub-1 Ghz with a basic bandwidth of 1Mhz/2Mhz (depending of configuration). The system is formed by an Access Point (AP) which maintains a Basic Service Set (BSS) and stations (STAs). STAs are connected to the AP in a star topology.

The 802.11ah is more energy efficient compared to other conventional 802.11 technologies because of it uses mechanisms which allow STAs to doze periodically and STAs request downlink data when switching to active mode i.e. Traffic Indication Map (TIM) operation, non-TIM operation, Target Wakeup Time (TWT)

An exemplary deployment of a 802.11ah BSS may include a large number of STAs associated to a BSS where STAs are sleeping (dozing) most of the time and they may monitor periodic beacon-frame transmissions containing Traffic Indication Maps (TIM). Data packets intended to STAs cannot be delivered when STAs sleep, thus the TIM indicates which STAs have downlink data buffered at the AP. After reading the TIM, STAs request their buffered data by transmitting a Power-Saving Poll (PS-Poll) frame to the AP. After the downlink data is delivered, STAs enter into sleep mode again. For uplink data delivery, STAs might transmit as soon as their data is available.

There might be STAs that do not monitor constantly the TIM and request downlink data sporadically after waking up.

3.1. Link Layer Topology of 802.11ah

The 802.11ah defines a star topology at L2 link connectivity, where the STAs are connected to the AP and any communication between STAs passes through the AP. It also includes L2 relays to extend the range of the system. As in other 802.11 amendments, the ad-hoc topology is also suported. Finally, the 802.11 standard does not define its own networking layer but is compatible with commonly used protocols e.g. IPv4, IPv6 via the Link Layer Control.

```
                      +---+
                      |STA|
                      +-+-+
         +---+          |
         |STA+------+   |
         +---+      |   |
                    +---+---+    +---+
                    |   AP  +----+STA|
                    ++-----++    +---+
          +----+     |      |
          |STA +-----+      |
          +----+          +-+--+
                          |STA |
                          +----+
```

Figure 1: Star Link Layer Topology

It is important to note that the communication link is unidirectional at any given point in time and that the medium is shared by CSMA/CA techniques which avoid that two or more STAs utilize the medium simultaneously.

3.2. Device Addressing and Frame Structure

The 802.11 physical transmission is composed by a preamble which is used to prepare a receiver for frame decoding, basic physical layer information, and the physical layer payload which encapsulates the MAC Protocol Data Unit (MPDU).

There can be different classes of MAC frames in 802.11, the MAC data frame is the only one carrying higher layer data. Other frames are control and management frames which are used to maintain MAC layer functions. In general in 802.11 MAC addresses use the EUI-48 bit address space.

A MAC data frame in 802.11 is composed by a MAC header, a MAC payload and a Frame Check Sequence (FCS) which are encoded in an MPDU. The MAC payload carries Link Layer Control PDUs which encapsulates, for example, IP packets. There are two protocol versions for MAC frame formats, the Protocol Version 0 (PV0) which is the default format of 802.11 and it is inherited to 802.11ah and the Protocol Version 1 (PV1) which has less overhead that PV0 and can be optionally supported by 802.11ah non-sensor STA and it is mandatory supported for 802.11ah sensor STA.

In 802.11ah, the maximum size of the MSDU (MAC payload) is given by the maximum size of a A-MSDU which is constrained by the maximum size of the A-MPDU of 7991 bytes. This maximum of the A-MPDU is independent of Protocol Version.

In addition, segmentation at 802.11 MAC layer level is supported if required.

3.3. Protocol Version 0

The elements of the MAC data frame with PV0 are defined in 802.11-2012, Section 8.2 [IEEE802.11-2012] and are depicted in the picture below.

```
+-------+--------+----+----+----+------+----+-----+----+-------+---+
+ Frame +Duration+ A1 + A2 + A3 + Seq. + A4 + QoS + HT + Frame +FCS+
+Control+   /ID  +    +    +    + Ctrl +    + Crl +Crl + Body  +   +
+-------+--------+----+----+----+------+----+-----+----+-------+---+
     2      2      6     6   6    2     6/0    2     4    0-7951  4
```

Figure 2: MAC frame PV0

Frame Control: contains information relevant in link layer such as the Protocol Version, frame type and subtype, Power Management, Fragmentation Information, among others.

A1, A2, A3: indicate the recipient, the transmitter and the BSSID which in infraestructure mode is the value of the STA contained in the AP (AP MAC address in practice). They follow 48-bits MAC address format.

A4, Sequence control, QoS control, HT control: The meaning of these field are out of scope of this draft. Please refer to 802.11-2012, Section 8.2.4 [IEEE802.11-2012] for further information.

Frame Body: is of variable-length field and contains the MAC payload for example L3 packets.

FCS: The Frame Check Sequence field is a 32-bit field containing a 32-bit CRC which is calculated over all the fields of the MAC header and the Frame Body field

3.4. Protocol Version 1

The MAC header for the PV1 format is at least formed by a Frame Control field and the address fields. Other fields are optional. Please refer to 802.11-2012, Section 8.8.1 [IEEE802.11ah] for further information.

```
       +---------------+-------+--------+---------------------+
       + Frame Control +   A1  +   A2   + Frame Body +  FCS   +
       +---------------+-------+--------+---------------------+
        Bytes:  2         6/2     2/6      0-7951         4
```

Figure 3: MAC frame PV1 of 802.11ah

Frame control: see above.

A1, A2: indicates the recipient and the transmitter respectively of the frame and it contains the 6-bytes MAC address or the Short ID (2-bytes) provided by the AP after association in a given BSS. Short ID includes the Association Identifier (AID) field which is used in TIM and power-saving mode.

Frame Body: The minimum length for non-data frames is 0 bytes. The maximum length of A-MSDU is constrained by the maximum size of the A-MPDU of 7991 bytes.

3.5. Link Layer Control

The Logical Link Control (LLC) layers works as the interface between higher layers, for example IP, and the 802.11 MAC. It supports higher layer protocol discrimination via the EtherType value utilizing the LLC SNAP or RFC1042.

```
           +----------------------------------------------------+
           | DSAP | SSAP | CTL    | OUI      | Ethertype  | SDU |
           | 0xAA | 0xAA | 0x03=UI| 00+00+00 |            |     |
           +----------------------------------------------------+
```

Figure 4: Format of LPD compatible with current 802.11 recommendations

Examples of EtherTypes are 0x0800 and 0x8DD, which are used to identify IPv4 and IPv6, respectively.

```
                           +-----------------------+
                           |     Upper Layer       |
                           +-----------------------+
                           |        802 LLC        |
                           +-----------------------+
                           | MAC Layer (802.11ah)  |
                           +-----------------------+
                           | PHY Layer (802.11ah)  |
                           +-----------------------+
```

Figure 5: WLAN Protocol Stack

3.6. Ad Hoc Mode and Extended Service Set

The standard allows to connect devices through ad-hoc mechanisms. In this mode the devices are connected using implementation specific protocols e.g. between two STAs or between two APs and the power-saving mechanism of 802.11ah cannot be used (as AP-STA hierarchy is required). The following figure describes STAs connected to AP through 802.11ah and connections between APs are not based on 802.11ah, but are implementation specific.

```
   +---+     +---+           +----+      +-----+
   |STA+-----+AP +-----------+AP  +------+STA  |
   +---+     +--++           +--+-+      +-----+
                |               |
                |               |
                |           +---+-+      +-----+
                +-----------+AP   +------+STA  |
                            +-----+      +-----+
```

Figure 6: WLAN Ad Hoc Mode

In an Extented Service Set(ESS), the connections between Base Service Station (BSS) happen through a distribution system. The distribution system (DS) maybe realised by a different technology or it can be composed by AP connections.

```
   +------------------+                         +------------------+
   | +---+     +---+  |                         | +----+    +----+ |
   | |STA+-----+AP +------------ DS -----------+AP  +----+STA | |
   | +---+     +---+  |                         | +----+    +----+ |
   +----------+-------+                         +------------------+
         BSS  |                                         | BSS
              +--------------->  ESS  <-----------------+
```

Figure 7: WLAN Protocol Stack

3.7. Relation with other 802.11 Versions

In principle, the 6Lo stack might be used for other 802.11 versions such as 802.11b, 802.11n and 802.11ac, due to these standards support LLC compatibility. LLC 6lo indentifier would be the same for all mentioned WiFi versions.

4. Uses Cases

[RFC7548] defines use cases for the management of constrained networks: Environmental Monitoring, Infrastructure Monitoring, Industrial Applications, Energy Management, Medical Applications, Building Automation, Home Automation, Transport Applications, Community Network Applications and Field Operations. These uses cases are apply as well to 802.11ah.

As a starting point in 802.11ah specification work, the Task Group AH proposed the following use-case categories [ReferenceUseCase802.11ah]:

- Sensor and Meters, where large number of sensor deliver data through 802.11ah connectivity

- Backhaul Sensor and meter data, where 802.11ah STA can be either directly integrated with a sensor or it will aggregate data from other tree of wireless sensors and then deliver 802.11ah connectivity

- Extended Range Wi-Fi, where the typical range of the Wi-Fi connection will extended due to the use of lower frequencies and other techniques.

5. 6LoWPAN over 802.11ah

IPv4 and IPv6 are compatible with 802.11ah via the LLC. However, 802.11ah technology presents a trade-off between energy consumption and link bitrate. Consequently, 6LoWPAN techniques are beneficial to reduce the overhead of transmissions, save energy and improve throughput. With 6LoWPAN, the nodes, i.e. 6LN, 6LBR, are co-located on the same devices with 802.11 features. The typical 802.11ah network uses a star topology where the 6LBR functionally is co-located with the AP. 6LNs are co-located with STAs and are connected to the 6LBR through 802.11ah links. As mesh topology at MAC level is not defined by the 802.11ah standard, 6LBR is the only router present in the network. Thus, there is no presence of 6LR.

```
              +---------+
              |+-------+|                      +---------+
              ||  6LN  ||  802.11ah            |+-------+|
              |+-------+|                      ||  6LN  ||
              |+-------++------------+---------|+-------+|
              ||  STA  ||            |         |+-------+|
              |+-------+|            |         ||  STA  ||
              +---------+            |         |+-------+|
                6LN-STA              |         +---------+
                               +-----+-----+
                               |+----+----+|
                               ||  6LBR   ||
                               |+---------+|
                 +---------+   |           |    +---------+
                 |+-------+|   |+---------++    ++-------+|
                 ||  6LN  ||   ||   AP    ||    ||  6LN  ||
                 |+-------+|   |+---------+|    |+-------+|
                 |+-------++---+----+------+    |         |
                 ||  STA  ||        |  6LBR-AP  |+-------+|
                 |+-------+|        |           ||  STA  ||
                 +--------+|        |           |+-------+|
                 +---------+        +-----------+---------+
```

Figure 8: Network Topology

There exists the possibility to have a 802.11ah relay node at L2 to extend the range of an AP. This however is an L2 feature and it is experienced as a single hop by the 6LoWPAN network. In case there is need to connect wirelessly several APs and ad hoc solution needs to be considered.

Devices in this kind of networks, not necessarily have constrained resources (memory, CPU, etc), but the radio link capacity is limited. It might be that APs are connected to mains power and STAs might be for example battery operated sensors. Therefore 6LoWPAN techniques might be good to support transmission of IPv6 packets over 802.11ah battery operated devices. Related to performance gain, a reduction in air-time is achieved if the stack is compressed. The communication 6LN-6LN is not supported directly using link-local addresses, it is done through the 6LBR using the shared prefix used on the subnet. This specification requires IPv6 header compression format specified in [RFC6282].

The Figure below shows the stack for PHY/MAC and IPv6 including 6LoWPAN

```
            +---------------------+
            |    Upper Layers     |
            +---------------------+
            |       IPv6          |
            +---------------------+
            |      6LoWPAN        |
            +---------------------+
            |      802 LLC        |
            +---------------------+
            | MAC Layer(802.11ah) |
            +---------------------+
            | PHY Layer(802.11ah) |
            +---------------------+
```

Figure 9: Protocol Stack with 6LoWPAN

6. Stateless Address Autoconfiguration

The IPv6 link local address follows Section 5.3 of [RFC4862] based on the 48-bit MAC device address.

To get the 64-bit Interface Identifier (IID) RFC 7136 [RFC7136] MUST be followed. Section 5 of this RFC states:

"For all unicast addresses, except those that start with the binary value 000, Interface IDs are required to be 64 bits long. If derived from an IEEE MAC-layer address, they must be constructed in Modified EUI-64 format."

```
              10 bits        54 bits            64 bits
           +----------+-----------------+----------------------+
           |1111111010|       0         | Interface Identifier |
           +----------+-----------------+----------------------+
```

Figure 10: IPv6 link local address

Following Appendix-A of RFC 4291 [RFC4291] the IID is formed inserting two octets, with hexadecimal values of 0xFF and 0xFE in the middle of the 48-bit MAC. The IID would be as follow where "a" is a bit of the 48 MAC address.

```
|0              1|1              3|3              4|4              6|
|0              5|6              1|2              7|8              3|
+----------------+----------------+----------------+----------------+
|aaaaaaaaaaaaaaaa|aaaaaaaa11111111|11111110aaaaaaaa|aaaaaaaaaaaaaaaa|
+----------------+----------------+----------------+----------------+
```

Figure 11: Modified EUI-64 format

For non-link-local addresses a 64-bit IID MAY be formed by utilizing the 48-bit MAC device address. Random IID can be generated for 6LN using alternative methods such as [I-D.ietf-6man-default-iids].

7. Neighbour Discovery in 802.11ah

Neighbour Discovery approach for 6LoWPAN [RFC6775] is applicable to 802.11ah topologies. Related to Host-initiated process, use of Address Registration Option (ARO), through the Neighbour Solicitation (NS) and Neighbour Advertisement (NA). Router Solicitation and Router Advertisement are applicable as well following [RFC6775].

As the topology is star, Multihop Distribution of prefix and 6LoWPAN header compression; and Multihop Duplicated Address Detection (DAD) mechanism are not applicable, since this technology does not cover multihop topology.

8. Header Compression

For header compression, the rules proposed in [RFC6282] are applicable. Section 3.1.1 mentions the base Encoding principle applicable to 802.11ah.

```
0                                       1
0   1   2   3   4   5   6   7   8   9   0   1   2   3   4   5
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
| 0 | 1 | 1 |  TF   |NH | HLIM  |CID|SAC|  SAM  | M |DAC|  DAM  |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
```

Figure 12: LOWPAN_IPHC base Encoding

TF: Traffic Class; Flow Label; For 802.11ah case would apply this field as defined in [RFC6282].

NH: Next Header; as defined in [RFC6282].

HLIM: Hop Limit; as star topology the common value would be HLIM=1.

CID: Context Identifier Extension; as defined in [RFC6282].

SAC: Source Address Compression; as defined in [RFC6282].

SAM: Source Address Mode; In this case, the combinations for 16-bits are not applicable to this technology since 802.11 uses 48-bits for addresses.

M: Multicast Compression; as defined in [RFC6282].

DAC: Destination Address Compression; as defined in [RFC6282].

DAM: Destination Address Mode. In this case, the combinations for 16-bits are not applicable to this technology since 802.11 uses 48-bits for addresses.

9. Fragmentation

802.11ah perform fragmentation at L2, thus the fragmentation at L3 would be not necessary.

10. Multicast at IP Level

802.11ah supports broadcast and multicast at link layer level, both can be used to carry multicast IP transmission depending on the system’s configuration. TBD: add an example.

11. Internet Connection

For Internet connection, the 6LBR acts as router and forwarding packets between 6LNs to and from Internet.

```
          +-----+
          | 6LN +--------+
          +-----+        |
                         |         +-----------+
                    +----+----+    |           |
                    |         |    |  Internet |
             +------+  6LBR   +----+           |
          +--+--+   |         |    |           |
          | 6LN |   +----+----+    +-----------+
          +-----+        |
                      +--+--+
                      | 6LN |
                      +-----+
```

Figure 13: Internet connection of 6Lo network

12. Management of the Network

TBD: how LightWeight Machine to Machine (LWM2M) or CoAP Management Interface (COMI) [I-D.vanderstok-core-comi] aspects can be applied to this technology, considering [RFC7547]

13. IANA Considerations

There are no IANA considerations related to this document.

14. Security Considerations

The security considerations defined in [RFC4944] and its update [RFC6282] can be assumed valid for the 802.11ah case as well. Indeed, the transmission of IPv6 over 802.11ah links meets all the requirements for security as for IEEE 802.15.4. The standard IEEE 802.11ah defines all those aspects related with Link Layer security. As well as for other existing WiFi solutions, 802.11ah Link Layer supports security mechanism such as WPA, WPS, 802.1X. To have a deeper understanding on how the Key Management processes are handled in 802.11ah, please refer to [TBD]

Implementations defined in [I-D.ietf-6man-default-iids], [RFC3972], [RFC4941], or [RFC5535], can be considered, for example, as methods to support non-link local addresses.

For what concerns privacy issues, the draft [I-D.thaler-6lo-privacy-considerations] introduces a series of recommendations which can be applied in order to overcome possible privacy threats in the particular case of technologies designed for IPv6 over networks of resource-constrained nodes.

15. Acknowledgements

This work is partially funded by the FP7 Marie Curie Initial Training Network (ITN) METRICS project (grant agreement No. 607728).

The authors are thankful to the members of IEEE Task Group AH for their valuable comments.

16. References

16.1. Normative References

[IEEE802.11ah]
Institute of Electrical and Electronics Engineers (IEEE), "Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications: Amendment- Sub 1 GHz License-Exempt Operation", January 2015.

[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>.

[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>.

16.2. Informative References

[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-6man-default-iids]
Gont, F., Cooper, A., Thaler, D., and S. LIU, "Recommendation on Stable IPv6 Interface Identifiers", draft-ietf-6man-default-iids-08 (work in progress), October 2015.

[I-D.thaler-6lo-privacy-considerations]
Thaler, D., "Privacy Considerations for IPv6 over Networks of Resource-Constrained Nodes", draft-thaler-6lo-privacy-considerations-01 (work in progress), October 2015.

[I-D.vanderstok-core-comi]
Stok, P., Bierman, A., Schoenwaelder, J., and A. Sehgal, "CoAP Management Interface", draft-vanderstok-core-comi-08 (work in progress), October 2015.

[IEEE802-2014]
Institute of Electrical and Electronics Engineers (IEEE), "IEEE Standard for Local and Metropolitan Area Networks: Overview and Architecture", 2014.

[IEEE802.11]
Institute of Electrical and Electronics Engineers (IEEE), "Wireless LAN", 2011.

[IEEE802.11-2012]
Institute of Electrical and Electronics Engineers (IEEE), "Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications", 2012.

[ReferenceUseCase802.11ah]
Institute of Electrical and Electronics Engineers (IEEE), "Potential compromise of 80211ah use case", 2012.

[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>.

[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>.

[RFC7547] Ersue, M., Ed., Romascanu, D., Schoenwaelder, J., and U. Herberg, "Management of Networks with Constrained Devices: Problem Statement and Requirements", RFC 7547, DOI 10.17487/RFC7547, May 2015, <http://www.rfc-editor.org/info/rfc7547>.

[RFC7548] Ersue, M., Ed., Romascanu, D., Schoenwaelder, J., and A. Sehgal, "Management of Networks with Constrained Devices: Use Cases", RFC 7548, DOI 10.17487/RFC7548, May 2015, <http://www.rfc-editor.org/info/rfc7548>.

Authors' Addresses

Luis Felipe Del Carpio Vega
Ericsson
Hirsalantie 11
Jorvas 02420
Finland

Email: felipe.del.carpio@ericsson.com

Maria Ines Robles
Ericsson
Hirsalantie 11
Jorvas 02420
Finland

Email: maria.ines.robles@ericsson.com

Roberto Morabito
Ericsson
Hirsalantie 11
Jorvas 02420
Finland

Email: roberto.morabito@ericsson.com

6Lo Working Group
Internet-Draft
Intended status: Standards Track
Expires: January 4, 2016

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

July 3, 2015

Transmission of IPv6 Packets over DECT Ultra Low Energy
draft-ietf-6lo-dect-ule-02

Abstract

DECT Ultra Low Energy 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

Copyright Notice

Table of Contents

1. Introduction

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 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 operates with 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 an elderly pendant (broche) 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 implements 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. Many of the 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 [I-D.ietf-6lo-btle].

1.1. 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].

1.2. Terms Used

PP: DECT Portable Part, typically the sensor node

FP: DECT Fixed Part, the gateway

LLME: Lower Layer Management Entity

RFPI: Radio Fixed Part Identity

IPEI: International Portable Equipment Identity

TPUI: Temporary Portable User Identity

PMID: Portable MAC Identity

PVC: Permanent Virtual Circuit

6LN: DECT Portable part having a role as defined in [RFC6775]

6LBR: DECT Fixed Part having a role as defined in [RFC6775]

2. DECT Ultra Low Energy

DECT ULE is a low power air interface technology that is designed to support both circuit switched for service, such as voice communication, and for 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 consists of the PHY layer operating at frequencies in the 1880 - 1920 MHz frequency band depending on the region and uses a symbol rate of 1.152 Mbps. Radio bearers are allocated by use of FDMA/TDMA/TDD technics.

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 as this is used for services like discovery, pairing, security features etc. All these features have been reused from DECT.

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

The current DECT ULE MAC layer standard supports low bandwidth data broadcast. However the usage of this broadcast service has not yet been standardized for higher layers. This document is not considering usage of this DECT ULE MAC broadcast service in current version.

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 Programmers 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 used application protocol is negotiated between the PP and FP when the PVC communication service is established. This draft defines 6LoWPAN as one of the possible protocols to negotiate.

```
+----------------------------------------+
|             Applications               |
+----------------------------------------+
| Generic Access     |     ULE Profile   |
|       Profile      |                   |
+----------------------------------------+
| DECT/Service API   | ULE Data API      |
+--------------------+-------------------+
| LLME  | NWK (MM,CC)|                   |
+--------------------+-------------------+
| DECT DLC           | DECT ULE DLC      |
+--------------------+-------------------+
|              MAC Layer                 |
+--------------------+-------------------+
|              Physical Layer            |
+--------------------+-------------------+
      (C-plane)             (U-plane)
```

Figure 1: DECT ULE Protocol Stack

The DECT ULE stack can be divided into control (C-plane) and user-data (U-plane) parts shown to the left and to the right in figure 1, respectively.

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 does only address this primary scenario.

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.

```
       [DECT ULE PP]-----\                 /-----[DECT ULE PP]
                          \               /
       [DECT ULE PP]-------+[DECT ULE FP]+-------[DECT ULE PP]
                          /               \
       [DECT ULE PP]-----/                 \-----[DECT ULE PP]
```

Figure 2: DECT ULE star topology

DECT ULE repeaters are not considered in 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 DECT globally unique for the PP and can be used to constitute the MAC address. However, it cannot be used to derive a globally unique IID.

When bound to a FP, a PP is assigned a 20 bit TPUI which is unique within the FP. This TPUI is used for addressing (layer 2) in messages between FP and PP.

Each DECT FP is assigned a RFPI during manufacturing. This identity has the size of 40 bits and is globally unique for a FP and can be used to constitute the MAC address. However, it cannot be used to derive a globally unique IID.

Alternatively 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. With such an approach, the FP and PP have to implement a mapping between used MAC-48 addresses and DECT identities.

2.4. MTU Considerations

Generally the DECT ULE FP and PP may be generating data that fits into a single MAC Layer packet (38 octets) for periodically transferred information, depending on application. IP data packets may be much larger and hence MTU size should be the size of the IP data packet. The DECT ULE DLC procedures supports segmentation and reassembly of any MTU size below 65536 octets, but most implementations do only support smaller values. The default MTU size in DECT ULE is 500 octets, but it SHALL be configured to fit the requirements from IPv6 data packets, hence [RFC4944] fragmentation/ reassembly is not required.

It is expected that the LOWPAN_IPHC packet will fulfill all the requirements for header compression without spending unnecessary overhead for mesh addressing.

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.

2.5. Additional Considerations

The DECT ULE standard allows PP to be registered (bind) to multiple FP and roaming between these FP. This draft does not consider the scenarios of PP roaming between multiple FP. The use of repeater functionality is also not considered in this draft.

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.

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 it’s primary configuration does not support the formation of multihop networks at the link layer. In consequence, the mesh header defined in [RFC4944] for mesh under routing MUST NOT be used in DECT ULE networks. In addition, a DECT ULE PP node MUST NOT play the role of a 6LoWPAN Router (6LR).

3.1. Protocol stack

In order to enable transmission of IPv6 packets over DECT ULE, a Permanent Virtual Circuit (PVC) has to be opened between FP and PP. This MUST be done by setting up a service call from PP to FP. 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-ATTRIBTES>> SHALL define the ULE Application Protocol Identifier to 0x06 and the MTU size to 1280 octets or larger. The FP MUST send a service-change-accept (resume) containing a valid paging descriptor. The PP MUST be pageable.

```
+-------------------+
|    UDP/TCP/other  |
+-------------------+
|       IPv6        |
+-------------------+
|6LoWPAN adapted to |
|    DECT ULE       |
+-------------------+
|  DECT ULE DLC     |
+-------------------+
|  DECT ULE MAC     |
+-------------------+
|  DECT ULE PHY     |
+-------------------+
```

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 implements fragmentation and reassembly functionality and [RFC4944] fragmentation and reassembly function MUST NOT be used. Since IPv6 requires MTU size of at least 1280 octets, the DECT ULE connection (PVC) MUST be configured with equivalent MTU size.

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 DECT ULE star topology, each branch of the star is considered to be an individual link and thus the PPs cannot directly hear one another and cannot talk to one another with link-local addresses. However, the FP acts as a 6LBR for communication between the PPs. 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 FP ensures address collisions do not occur.

3.2.1. Stateless address autoconfiguration

A DECT ULE 6LN performs stateless address autoconfiguration as per [RFC4862]. Following the guidance of [RFC7136], a 64-bit Interface identifier (IID) for a DECT ULE interface MAY be formed by utilizing a MAC-48 device address as defined in [RFC2464] "IPv6 over Ethernet" specification.

Alternatively, the DECT device addresses IPEI, RFPI or TPUI, MAY be used instead to derive the IID. These DECT devices addresses consisting of 40, 40 and 20 bits respectively, MUST be expanded with leading bits to form a 48 bit address. Least significant bit of this address is the last bit in network order. The expanded leading bits are all zeros except for 7th bit indicating not global unique. First bit is set to a one for addresses derived from the RFPI and 2nd bit is set to one when the address is derived from the PMID. For example from IPEI=01.23.45.67.89 is derived MAC address equal 02:01:23:45:67:89 and from PMID=0.01.23 is derived MAC address equal 42:00:00:00:01:23.

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

```
   10 bits       54 bits           64 bits
+----------+-----------------+----------------------+
|1111111010|      zeros      | Interface Identifier |
+----------+-----------------+----------------------+
```

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.

For non-link-local addresses a 64-bit IID MAY be formed by utilizing a MAC-48 device address. A 6LN can also use a randomly generated IID (see Section 3.2.2), 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 DHCPv6 [RFC3315]. The non-link-local addresses 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 is considered not to support mesh networks, hence only those aspects that apply to a 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. 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. The 6LN MUST register only one IPv6 address per available IPv6 prefix.

3.2.3. Unicast and Multicast address mapping

The DECT MAC layer broadcast service is considered inadequate for IP multicast.

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. 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 and ARO can be exploited in order to provide a mechanism for addess 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 node knows that the packet is destined for it even if the packet does not have destination IPv6 address. A node SHALL learn the IID of the other endpoint of each DECT ULE connection it participates in. By exploiting this information, a node that receives a PDU containing an IPv6 packet can infer the corresponding IPv6 source address. A node MUST maintain a Neighbor Cache, in which the entries include both the IID of the neighbor and the Device Address that identifies the neighbor. For the 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 elide the 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 elide the IPv6 destination address of the packet before forwarding it to the 6LN. 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 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.

A degenerate scenario can be imagined where a PP is acting as 6LBR and providing Internet connectivity for the FP. How the FP could then further provide Internet connectivity to other PP, possibly connected to the FP, is out of the scope of this document.

```
                          6LN
                           \            ____________
                            \          /            \
                    6LN ---- 6LBR --- |  Internet    |
                            /          \____________/
                           /
                          6LN

                    <-- DECT ULE -->
```

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.

```
                         6LN      6LN
                          \      /
                           \    /
                    6LN --- 6LBR --- 6LN
                           /    \
                          /      \
                         6LN      6LN

                    <------ DECT ULE ----->
```

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.

4. IANA Considerations

There are no IANA considerations related to this document.

5. Security Considerations

The secure transmission of speech over DECT will be based on the DSAA2 and DSC2 work developed by the DF Security group / 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) and during location registration procedure or when the permanent virtual circuit are established, the session security keys are generated. 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.

The IPv6 address configuration as described in Section 3.2.1 allows implementations the choice to support, for example, [I-D.ietf-6man-default-iids], [RFC3972], [RFC4941] or [RFC5535] for non-link-local addresses.

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 has 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.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC2464] Crawford, M., "Transmission of IPv6 Packets over Ethernet Networks", RFC 2464, December 1998.

[RFC3633] Troan, O. and R. Droms, "IPv6 Prefix Options for Dynamic Host Configuration Protocol (DHCP) version 6", RFC 3633, December 2003.

[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.

[RFC4862] Thomson, S., Narten, T., and T. Jinmei, "IPv6 Stateless Address Autoconfiguration", RFC 4862, September 2007.

[RFC4941] Narten, T., Draves, R., and S. Krishnan, "Privacy Extensions for Stateless Address Autoconfiguration in IPv6", RFC 4941, 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.

[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.

[RFC7136] Carpenter, B. and S. Jiang, "Significance of IPv6 Interface Identifiers", RFC 7136, February 2014.

[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.

[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.

8.2. Informative References

[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-14 (work in progress), June 2015.

[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-04 (work in progress), June 2015.

[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.

[RFC3610] Whiting, D., Housley, R., and N. Ferguson, "Counter with CBC-MAC (CCM)", RFC 3610, September 2003.

[RFC3972] Aura, T., "Cryptographically Generated Addresses (CGA)", RFC 3972, March 2005.

[RFC5535] Bagnulo, M., "Hash-Based Addresses (HBA)", RFC 5535, June 2009.

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
Sensinode
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: January 6, 2016

Y-G. Hong
Y-H. Choi
ETRI
J-S. Youn
DONG-EUI Univ
D-K. Kim
KNU
J-H. Choi
Samsung Electronics Co.,
July 5, 2015

Transmission of IPv6 Packets over Near Field Communication
draft-ietf-6lo-nfc-01

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

Copyright Notice

Table of Contents

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 techiques 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 of 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 of 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.

```
      |                                        |
      |                                        |  Application Layer
      |         Upper Layer Protocols          |   Transport Layer
      |                                        |    Network Layer
      |                                        |         |
      +----------------------------------------+ <------------------
      |           IPv6-LLCP Binding            |         |
      +----------------------------------------+        NFC
      |                                        |    Logical Link
      |      Logical Link Control Protocol     |       Layer
      |                 (LLCP)                 |         |
      +----------------------------------------+ <------------------
      |                                        |         |
      |               Activities               |         |
      |            Digital Protocol            |        NFC
      |                                        |      Physical
      +----------------------------------------+       Layer
      |                                        |         |
      |               RF Analog                |         |
      |                                        |         |
      +----------------------------------------+ <------------------
```

Figure 1: Protocol Stacks of NFC

The LLCP consists of Logical Link Control (LLC) and MAC Mapping. The MAC Mapping integrates an existing RF protocol into the LLCP architecture. The LLC contains three components, such as Link Management, Connection-oriented Transport, and Connection-less

Transport. The Link Management component is responsible for serializing all connection-oriented and connectionless LLC PDU (Protocol Data Unit) exchanges and for aggregation and disaggregation of small PDUs. This component also guarantees asynchronous balanced mode communication and provides link status supervision by performing the symmetry procedure. The Connection-oriented Transport component is responsible for maintaining all connection-oriented data exchanges including connection set-up and termination. The Connectionless Transport component is responsible for handling unacknowledged data exchanges.

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 MAC PDU Size and MTU

As mentioned in Section 3.2, an IPv6 packet SHALL be received at LLCP of NFC and transported to an Unnumbered Information Protocol Data Unit (UI PDU) and an Information Field in Protocol Data Unit (I PDU) of LLCP of the NFC-enabled peer device. The format of the UI PDU and I PDU SHALL be as shown in Figure 2 and Figure 3.

```
   0      0    1      1
   0      6    0      6
  +------+----+------+-------------------------------------------+
  |DDDDDD|1100|SSSSSS|             Service Data Unit             |
  +------+----+------+-------------------------------------------+
  | <-- 2 bytes ---> |                                           |
  | <------------------- 128 ˜ 2176 bytes ---------------------> |
  |                                                              |
```

Figure 2: Format of the UI PDU in NFC

```
      0      0    1      1    2    2
      0      6    0      6    0    4
     +------+----+------+----+----+---------------------------------+
     |DDDDDD|1100|SSSSSS|N(S)|N(R)|       Service Data Unit         |
     +------+----+------+----+----+---------------------------------+
     | <------- 3 bytes --------> |                                 |
     | <------------------- 128 ~ 2176 bytes ---------------------> |
     |                                                              |
```

Figure 3: 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. If no MIUX parameter is transmitted, the default MIU value of 128 SHALL be used. Otherwise, the MTU size in NFC LLCP SHALL calculate the MIU value as follows:

MIU = 128 + MIUX.

According to NFCForum-TS-LLCP_1.1 [3], format of the MIUX parameter TLV is as shown in Figure 4.

```
                    0        0        1    2          3
                    0        8        6    2          1
                   +--------+--------+----------------+
                   |  Type  | Length |     Value      |
                   +--------+--------+----+-----------+
                   |00000010|00000010|1011|    MIUX   |
                   +--------+--------+----+-----------+
                                          | <-------> |
                                          0x000 ~ 0x7FF
```

Figure 4: Format of the MIUX Parameter TLV

When the MIUX parameter is encoded as a TLV, the TLV Type field SHALL be 0x02 and the TLV Length field SHALL be 0x02. The MIUX parameter SHALL be encoded into the least significant 11 bits of the TLV Value field. The unused bits in the TLV Value field SHALL be set to zero by the sender and SHALL be ignored by the receiver. However, a maximun value of the TLV Value field can be 0x7FF, and a maximum size of the MTU in NFC LLCP SHALL calculate 2176 bytes.

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 4.5) and header compression (see Section 4.6).

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 Stacks

Figure 5 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.

```
      |                                        |     Transport &
      |         Upper Layer Protocols          |  Application Layer
      +----------------------------------------+ <------------------
      |                                        |         |
      |                 IPv6                   |         |
      |                                        |      Network
      +----------------------------------------+       Layer
      |   Adaptation Layer for IPv6 over NFC   |         |
      +----------------------------------------+ <------------------
      |            IPv6-LLCP Binding                     |
      |      Logical Link Control Protocol     |   NFC Link Layer
      |                 (LLCP)                 |         |
      +----------------------------------------+ <------------------
      |                                        |         |
      |               Activities               |        NFC
      |            Digital Protocol            |   Physical Layer
      |               RF Analog                |         |
      |                                        |         |
      +----------------------------------------+ <------------------
```

Figure 5: Protocol Stacks 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 does not have to conduct the FAR procedure. The NFC LLCP, by contrast, does not support the FAR functionality, so IPv6 over NFC needs to consider the FAR functionality, defined in RFC4944 [1]. However, MTU on NFC link can be configured in a connection procedure and extended enough to fit the MTU of IPv6 packet.

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 LLCP 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.

Following the guidance of RFC7136 [10], interface Identifiers of all unicast addresses for NFC-enabled devices are formed on the basis of 64 bits long and constructed in a modified EUI-64 format as shown in Figure 6.

```
   0                1                3                4          5    6
   0                6                2                8          8    3
  +----------------+----------------+----------------+----------+------+
  |0000000000000000|0000000011111111|1111111000000000|0000000000| SSAP |
  +----------------+----------------+----------------+----------+------+
```

Figure 6: Formation of IID from NFC-enabled device adddress

In addition, the "Universal/Local" bit in the case of NFC-enabled device address MUST be set to 0 RFC4291 [7].

4.4. IPv6 Link Local Address

Only if the NFC-enabled device address is known to be a public address the "Universal/Local" bit can be set to 1. 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 7.

```
        0          0                  0                          1
        0          1                  6                          2
        0          0                  4                          7
       +----------+------------------+----------------------------+
       |1111111010|       zeros      |    Interface Identifier    |
       +----------+------------------+----------------------------+
       |                                                          |
       | <---------------------- 128 bits ----------------------> |
       |                                                          |
```

Figure 7: 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]).

4.5. 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.

4.6. 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 8.

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

Figure 8: NFC short adress format

4.7. Fragmentation and Reassembly

NFC provides fragmentation and reassembly (FAR) for payloads from 128 bytes up to 2176 bytes as mention in Section 3.4. The MTU of a general IPv6 packet can fit into a sigle NFC link frame. Therefore, the FAR functionality as defined in RFC4944, which specifies the fragmentation methods for IPv6 datagrams on top of IEEE 802.15.4, is NOT REQUIRED in this document as the basis for IPv6 datagram FAR on top of NFC. The NFC link connection for IPv6 over NFC MUST be configured with an equivalent MIU size to fit the MTU of IPv6 Packet. However, the default configuration of MIUX value is 0x480 in order to fit the MTU (1280 bytes) of a IPv6 packet.

4.8. 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.

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

Figure 9: 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.

4.9. 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 10: 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 11 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 becomes 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.

```
                                         ************
         6LN ------------------- 6LBR -----* Internet *------- CN
          |  (dis. 10 cm or less)  |       ************         |
          |                        |                            |
          | <-------- NFC -------> | <----- IPv6 packet ------> |
          | (IPv6 over NFC packet) |                            |
```

Figure 11: 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 12.

```
         6LN --------------------- 6LR --------------------- 6LN
          |     (10 cm or less)      |     (10 cm or less)      |
          |                          |                          |
          | <--------- NFC --------> | <--------- NFC --------> |
          |   (IPv6 over NFC packet) |  (IPv6 over NFC packet)  |
```

Figure 12: 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 to required to protect from duplication through accident or forgery.

8. Acknowledgements

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

Michael Richardson, Suresh Krishnan, Pascal Thubert, Carsten Bormann, and Alexandru Petrescu have provided valuable feedback for this draft.

9. References

9.1. 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.

[10] Carpenter, B. and S. Jiang, "Significance of IPv6 Interface Identifiers", RFC 7136, February 2014.

9.2. Informative References

[11] "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

Network Working Group
Internet-Draft
Intended status: Experimental
Expires: December 30, 2015

Y. Ohba, Ed.
Toshiba

June 28, 2015

An Extension to Mesh Link Establishment (MLE) for Host Identity Protocol Diet Exchange (HIP DEX)

draft-ohba-mle-hip-dex-01

Abstract

This document defines an extension of MLE (Mesh Link Establishment) protocol to encapsulate HIP DEX key exchange protocol messages.

Status of This Memo

This Internet-Draft will expire on December 30, 2015.

Copyright Notice

Table of Contents

1. Introduction

HIP DEX (Host Identity Protocol Diet EXchange) [I-D.moskowitz-hip-dex] is a light-weight key exchange protocol designed for constrained devices. HIP DEX builds on the HIP Base EXchange (HIP BEX) [I-D.ietf-hip-rfc5201-bis] and inherits the transport-agnostic property of HIP BEX.

MLE (Mesh Link Establishment) [I-D.kelsey-intarea-mesh-link-establishment] is defined for establishing and configuring secure links in IEEE 802.15.4 mesh networks. MLE assumes that shared keys to secure link-layer frames and MLE messages exchanged between a pair of nodes are pre-configured between the nodes. Therefore, a key exchange protocol is required in order to dynamically configure the required shared keys. While such a key exchange protocol can be run outside MLE, sequentially running a key exchange protocol and MLE as separate protocols requires more message roundtrips. For example, running a HIP DEX 4-way handshake followed by an MLE 3-way handshake requires 3.5 message roundtrips.

In this document, an extension to the MLE protocol for encapsulating HIP DEX messages is defined in order to realize optimized key exchange and link establishment for IEEE 802.15.4 mesh networks.

1.1. Requirement Language

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

1.2. Acronyms

DEX-I1, DEX-R1, DEX-I2, DEX-R2: HIP DEX I1, R1, I2, R2 messages

ECDH: Elliptic Curve Diffie-Hellman

EI: HIP DEX Key Establishment Initiator

ER: HIP DEX Key Establishment Responder

LLFC: Link-Layer Frame Counter

MIC: MLE Message Integrity Code

MLFC: MLE Frame Counter

UI: HIP DEX Key Update Initiator

UR: HIP DEX Key Update Responder

1.3. Convention

In the figures of this document, MLE messages marked with ’*’ are those secured by the MLE protocol.

In the key material formats in this document, ’|’ denotes concatenation operator.

2. Overview

HIP DEX over MLE consists of two phases, i.e., Key Establishment Phase and Key Update Phase. In Key Establishment Phase, a HIP DEX 4-way handshake using I1, R1, I2 and R2 messages is conducted to establish a secure channel between an EI and an ER based on an ECDH shared secret and exchange session key materials over the secure channel.

In Key Update Phase, HIP DEX Update messages encrypting session key materials are exchanged between a UI and each UR using an MLE Update Request and Update exchange, followed by a multicast MLE Update message for triggering each UR to simultaneously activate new key materials and reset the associated link-layer frame counters. The UI and UR roles for a pair of nodes may be determined independently of the EI and ER roles that have been taken by the nodes.

All MLE messages used for the extension defined in this document SHOULD NOT be protected by link-layer so that a key exchange can be done regardless of the security state of the link-layer. A node that implements this specification MUST allow sending and receiving MLE messages not secured by the link-layer.

Secured 802.15.4 MAC frames and MLE messages that use keys established via HIP DEX MUST use a 5-octet Frame Counter so that the Frame Counter does not reach its maximum value throughout the lifetime of a node. An MLE Frame Counter is always carried in the Frame Counter field in the Aux Header of any secured MLE frame.

Other than the rules described in this document, the rules defined in [I-D.kelsey-intarea-mesh-link-establishment] are preserved.

3. Key Establishment Phase

A message exchange diagram for Key Establishment Phase is shown in Figure 1.

```
  (EI)  (ER)
     -->    Advertisement [HIP{DEX-I1}, Link Quality]

     <--    Advertisement [HIP{DEX-R1}, Link Quality]

     -->    Link Request  [HIP{DEX-I2}, Source Address, Mode,
                           Timeout, Challenge]*

     <--    Link Accept and Request
                          [HIP{DEX-R2}, LLFC, MLFC, Source Address, Mode,
                           Timeout, Response, Challenge]*

     -->    Link Accept   [LLFC, MLFC, Response]*
```

Figure 1: Key Establishment Phase

An EI sends an MLE Advertisement message containing a HIP TLV and a Link Quality TLV to an ER. The HIP TLV carries a DEX-I1 packet. How an EI discovers an ER is outside the scope of this document.

The ER receives the MLE Advertisement message containing a DEX-I1 packet from the EI and sends an MLE Advertisement message containing a HIP TLV and a Link Quality TLV to the EI. The HIP TLV carries a DEX-R1 packet. The DEX-R1 packet MUST contain mandatory R1 parameters specified in [I-D.moskowitz-hip-dex]. The DEX-R1 packet MAY contain optional R1 parameters specified in [I-D.moskowitz-hip-dex] and a CERT parameter defined in [RFC6253].

The EI receives the MLE Advertisement message from the ER and sends a secured MLE Link Request message containing HIP, Source Address, Mode, Timeout and Challenge TLVs to the ER. The HIP TLV carries a DEX-I2 packet. The DEX-I2 packet MUST contain mandatory I2 parameters specified in [I-D.moskowitz-hip-dex] including an ENCRYPTED_KEY parameter wrapping a session key material of the EI. The DEX-I2 packet MUST also contain an ENCRYPTED parameter wrapping group key materials of the EI. The DEX-I2 packet MAY contain optional I2 parameters specified in [I-D.moskowitz-hip-dex] and a CERT parameter defined in [RFC6253]. The MLE Link Request message is protected by the EI's group MLE key (see section Section 5.2) derived from the EI's group key materials.

The ER receives the MLE Link Request message from the EI and extracts the EI's session key material wrapped in the ENCRYPTED_KEY parameter and the EI's group key materials wrapped in the ENCRYPTED parameter. Then the ER sends a secured MLE Link Accept and Request message containing HIP, LLFC, MLFC, Source Address, Mode Timeout, Response and Challenge TLVs to the EI. The HIP TLV carries a DEX-R2 packet. The DEX-R2 packet MUST contain R2 parameters specified in [I-D.moskowitz-hip-dex] including an ENCRYPTED_KEY parameter wrapping a session key material of the ER. The DEX-R2 packet MUST also contain an ENCRYPTED parameter wrapping group key materials of the ER. The DEX-R2 packet MAY contain optional R2 parameters specified in [I-D.moskowitz-hip-dex]. Note that the MIC field of the MLE Link Request message is verified after the ER successfully extracts the EI's group key materials.

The EI receives the MLE Link Accept and Request message from the ER and extracts the ER's session key material wrapped in the ENCRYPTED_KEY parameter and the ER's group key materials wrapped in the ENCRYPTED parameter. Then the EI sends a secured MLE Link Accept message containing LLFC TLV, MLFC and Response TLVs to the ER. If a pair-wise key is used by the link-layer, the EI also creates a Pair-wise Key SA with the session key generated by the pair of session key materials of the EI and ER as specified in [I-D.moskowitz-hip-dex]. Note that the MIC field of the MLE Link Accept and Request message is verified after the EI successfully extracts the ER's group key materials.

The ER receives the MLE Link Accept message from the EI. If a pair-wise key is used by the link-layer, the EI creates a Pair-wise Key SA with the session key generated by the pair of session key materials of the EI and ER as specified in [I-D.moskowitz-hip-dex].

4. Key Update Phase

In Key Update Phase, group key materials are updated.

Since the 5-octet Frame Counter space is large enough considering the maximum bandwidth of 250Kbps in 802.15.4 [IEEE802154] to make an assumption that a Frame Counter does not reach its maximum value throughout the lifetime of a node, a mechanism for updating a pair-wise key is not defined in this document. Both link-layer Frame Counters and MLE Frame Counters are not reset in the Key Update Phase.

Updating a group key may happen when a node that shares the group key is revoked. A message exchange diagram for group key update is shown in Figure 2.

```
  (UI) (UR1)..(URn)
                   // Update 1st peer
    ---->          Update Request [HIP{DEX-UPDATE}, MLFC, Source Address]*
    <----          Update [HIP{DEX-UPDATE}, MLFC, Source Address]*
         ..                        ..

                   // Update n-th peer
    ----------->   Update Request [HIP{DEX-UPDATE}, MLFC, Source Address]*
    <-----------   Update [HIP{DEX-UPDATE}, MLFC, Source Address]*

                   // Key switch notification (multicast)
    ----> .. -->   Update [LLFC, MLFC]*
```

Figure 2: Group Key Update

First, a UI performs the following exchange for each UR:

- o The UI sends an MLE Update Request message containing HIP, MLFC, Source Address and MIC TLVs to a UR. The HIP TLV carries a DEX-UPDATE packet containing SEC, MAC and ENCRYPTED parameters. The ENCRYPTED parameter wraps new group key materials of the UI.

- o The UR receives the MLE Update Request message from the UI, extracts UI’s new group key materials from the ENCRYPTED parameter, activates the UI’s new group key materials for incoming frames, and sends an MLE Update message containing HIP, MLFC and Source Address TLVs, where the HIP TLV carries a DEX-UPDATE packet containing ACK and MAC parameters. Note that the MIC field of the MLE Update message is verified after the UR successfully extracts the UI’s new group key materials.

Once MLE Update Request and Update exchange is completed for all URs, the UI activates the UI’s new group key materials for outgoing frames by multicasting an MLE Update message containing LLFC and MLFC TLVs. The MLE Update message is protected by the UI’s group MLE key (see section Section 5.2) derived from the UI’s new group key materials.

When a UR receives the multicast MLE Update message, If the received message is valid, the UR deactivates the UI’s old group key materials for incoming frames.

A UR that did not receive the multicast MLE Update message may deactivate the UI’s old group key materials for incoming frames when it receives a valid MAC frame protected by the link-layer key derived from the UI’s new group key materials.

5. Key Materials

5.1. Pair-wise Key

The first 16 octets of the session key corresponding to the HIP DEX Pair-wise SA [I-D.moskowitz-hip-dex] is used as the pairwise link-layer key used for securing unicast link-layer frames with Key Identifier Mode 0x00.

An encrypted session key material is contained in an ENCRYPTED_KEY parameter of HIP when the session key is distributed during Key Establishment Phase.

5.2. Group Keys

Group key materials are created by a node and distributed to peer nodes.

The group key materials consist of a 1-octet key identifier (KeyId) and a 16-octet group master key (GroupMasterKey), and encoded as follows:

Group Key Materials = KeyId | GroupMasterKey

A 16-octet group link-layer key (GroupL2Key), and a 16-octet group MLE key (GroupMLEKey) are derived from GroupMasterKey as follows:

GroupL2Key = The first 16-octet of HMAC_SHA256(GroupMasterKey, KeyId).

GroupMLEKey = The last 16-octet of HMAC_SHA256(GroupMasterKey, KeyId).

A GroupL2Key is used for securing link-layer frames with Key Identifier Mode 0x03 sent by the node that created the group key material. GroupL2Key MUST be used for securing broadcast link-layer frames and MAY also be used for securing unicast link-layer frames.

A GroupMLEKey MUST be used for securing MLE messages with Key Identifier Mode 0x03 sent by the node that created the group key material.

The group key materials are contained in an GROUP_KEY_MATERIALS parameter of HIP, where the GROUP_KEY_MATERIALS parameter MUST be encrypted in an ENCRYPTED parameter of HIP.

6. MLE Security

As described in [I-D.kelsey-intarea-mesh-link-establishment], MLE security reuses that of IEEE 802.15.4, i.e., AES-CCM* [IEEE802154]. Since some of the MLE messages (i.e., MLE Link Accept and Request and MLE Accept messages carrying DEX-I2 and DEX-R2 packets, respectively, and unicast MLE Update Request and Update messages carrying a DEX-UPDATE packet) require to be sent unencrypted and only authentication is needed, MIC-64 (Security Level 2) or MIC-128 (Security Level 3) is used to secure MLE messages. MIC-64 is the default security level for securing MLE messages used in this document. GroupMLEKey (see section Section 5.2) with Key Identifier Mode 0x03 and a 5-octet Frame Counter MUST be used for any secured MLE message.

7. Certificate Revocation

Any MLE message used in this document MAY also contain a CRL (Certificate Revocation List) TLV in which CertificateList defined in [RFC5280] is encoded in the Value field. A node that receives a valid MLE message containing a CRL TLV revokes certificates specified in the TLV and deletes all pair-wise and group keys associated with the revoked certificates. A node MUST reject a CERT parameter for a revoked certificate in Key Establishment Phase.

How a CRL is propagated in a network depends on the network topology and is out of the scope of this document.

8. Security Considerations

The MLE extension defined in this document uses HIP DEX for key management of computation or memory constrained sensor/actuator devices, and thus it inherits all security considerations made for HIP DEX [I-D.moskowitz-hip-dex].

In order to mitigate security weakness caused by lack of Perfect Forward Secrecy (PFS) in HIP DEX, it is RECOMMENDED to use this MLE extension in conjunction with an additional mechanism to update public/private key pairs and renew HIP DEX SAs using new public/ private key pairs whenever necessary.

In both Key Establishment Phase and Key Update Phase, MLE messages are secured using a group key instead of a pairwise key in order to optimize message roundtrips since a group key establishment requires only a half roundtrip. As a result, a Denial of Service (DoS) attack from an insider sharing the group key is possible over MLE TLVs.

Due to integration of HIP DEX into MLE, secured MLE messages are authenticated but not encrypted because decryption can be done only after establishing a key. As a result, Source Address, Mode, Timeout, Challenge, Response LLFC and MLFC TLVs are sent in clear, and the cleartext information may be used by attackers for the DoS attack described above. Note that authentication of the MLE message carrying a DEX-I2, DEX-R2 or DEX-UPDATE packet is possible by validating MIC of the MLE message after extracting the authentication key (i.e., GroupMLEKey) from the HIP DEX packet.

9. IANA Considerations

9.1. MLE TLV Types

The following MLE TLV types are to be assigned by IANA based on the policy described in [I-D.kelsey-intarea-mesh-link-establishment]:

- HIP-DEX (Value: 9, Length: Variable, Meaning: HIP DEX packet, Reference: this document).

- CRL (Value: 10, Length: Variable, Meaning: Certificate Revocation List, Reference: this document).

9.2. HIP Parameter

The following HIP Parameter is assigned based on the policy described in [I-D.ietf-hip-rfc5201-bis]:

- GROUP_KEY_MATERIALS, (Value: 65530, Length: 33, Meaning: Group key materials for MLE and link-layer, Reference: this document).

10. Acknowledgments

The author would like to acknowledge the helpful comments of Randy Turner and Robert Cragie.

11. References

11.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC5280] Cooper, D., Santesson, S., Farrell, S., Boeyen, S., Housley, R., and W. Polk, "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile", RFC 5280, May 2008.

[RFC6253] Heer, T. and S. Varjonen, "Host Identity Protocol Certificates", RFC 6253, May 2011.

[I-D.moskowitz-hip-dex] Moskowitz, R. and R. Hummen, "HIP Diet EXchange (DEX)", draft-moskowitz-hip-dex-03 (work in progress), June 2015.

[I-D.ietf-hip-rfc5201-bis] Moskowitz, R., Heer, T., Jokela, P., and T. Henderson, "Host Identity Protocol Version 2 (HIPv2)", draft-ietf-hip-rfc5201-bis-20 (work in progress), October 2014.

[I-D.kelsey-intarea-mesh-link-establishment] Kelsey, R., "Mesh Link Establishment", draft-kelsey-intarea-mesh-link-establishment-06 (work in progress), May 2014.

11.2. External Informative References

[IEEE802154] IEEE standard for Information Technology, "IEEE std. 802.15.4, Part. 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks", June 2011.

Author’s Address

Yoshihiro Ohba (editor)
Toshiba Corporate Research and Development Center
1 Komukai-Toshiba-cho
Saiwai-ku, Kawasaki, Kanagawa 212-8582
Japan

Phone: +81 44 549 2127
Email: yoshihiro.ohba@toshiba.co.jp

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: August 20, 2015

D. Thaler
Microsoft

February 16, 2015

Enabling Security/Privacy Addressing On 6LoWPAN Technologies
draft-thaler-6lo-privacy-addrs-00

Abstract

It is commonly assumed today that 6LowPAN header compression is incompatible (or at least inefficient) with the notion of using addresses with sufficient entropy to mitigate various security and privacy threats. This draft explores ways one might dispel that notion, and discusses how security/privacy addressing might be used on 6LoWPAN technologies without additional overhead in data packets.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

RFC 6973 [RFC6973] discusses privacy considerations for Internet protocols, and Section 5.2 in particular covers a number of privacy-specific threats. In the context of IPv6 addresses, Section 3 of [I-D.ietf-6man-ipv6-address-generation-privacy] provides further elaboration on the applicability of the privacy threats. When interface identifiers (IIDs) are generated without sufficient entropy, devices and users become vulnerable to the various threats discussed there, including correlation of activities over time, location tracking, address scanning, and device-specific vulnerability exploitation.

Interfaces identifiers formed from IEEE identifiers can have insufficient entropy unless the IEEE identifier itself has sufficient entropy, and enough bits of entropy are carried over into the IPv6 address to sufficiently mitigate the threats. Typically "enough" bits of entropy means at least 46 bits (see Appendix for why); ideally all 64 bits of the IID should be used, although historically some bits have been excluded for reasons discussed in [RFC7421].

Furthermore, IEEE-identifier-based IIDs are also insufficient to prevent location tracking unless the IEEE identifier itself is different at each network location. This observation suggests that the privacy threats can be mitigated in either of two ways: either use an IPv6 address generation mechanism that is not IEEE-identifier-based, or else make sure the IEEE identifier contains at least 46 bits of entropy and is changed if a device moves to a different network. For this reason, [I-D.ietf-6man-default-iids] recommends using the address generation scheme in [RFC7217] by default, rather than IEEE-identifier-based addresses.

Furthermore, to mitigate the threat of correlation of activities over time, [RFC4941] specifies the notion of a "temporary" address to be used for sessions that should not be linkable to a more permanent identifier (such as a DNS name, user name, or stable hardware address). Such temporary addresses are appropriate for connections (typically locally-initiated outbound sessions) that an attacker cannot link to a stable identifier such as a user name or DNS name. Indeed, the default address selection rules [RFC6724] now prefer temporary addresses by default for outgoing connections. When temporary addresses are used, a new temporary address is periodically (default is 1 day in [RFC4941]) generated, which limits the threat of correlation of activies over time to that period. The address itself though may still be usable for existing long-lived connections (but not new connections) for some longer period (default is 1 week); this allows for not breaking application sessions, especially those that might be initiated shortly before a new temporary address is generated. This fact means that multiple temporary addresses can exist at the same time, one for new connections, and one or more (often up to 6, per the default periods) old ones for long-lived connections. This is in addition to any "stable" addresses that might be used for connections that are linkable to more permanent identifiers such as DNS names or user names. Whereas most threats could be mitigated if the IEEE identifier contains sufficient entropy and is different per-network, mitigating the threat of correlation of activities over time typically cannot be done using an IEEE-identifier-based-IID, since mitigating such a threat typically involves the ability to use multiple IPv6 addresses simultaneously whereas typically only one IEEE identifier can be used at a time.

Finally, allowing efficient use of addresses that are not IEEE-identifier-based also has additional security benefits not specific to privacy. For example, addresses such as Cryptographically Generated Addresses (CGAs) [RFC3972] and Hash-Based Addresses (HBAs) [RFC5535] can be used in security protocols such as Secure Neighbor Discovery (SeND) [RFC6496], IPsec, etc. Such techniques rely on having around 59 or more bits of entropy in the address to provide sufficient cryptographic protection.

RFC 6775 [RFC6775] already allows for the use of non-IEEE-identifier-based addresses, such as those provided by DHCPv6 [RFC3315]. There has been some concern, however, that such approaches necessarily interfere with efficient header compression for IPv6 (e.g., over IEEE 802.15.4-based networks [RFC6282]), as it is important to keep data packets small on 6LoWPAN networks.

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

The remainder of this document explores how one might use addresses with sufficient entropy on 6LoWPAN networks while avoiding extra overhead.

2. Terminology

This document uses the terminology defined in Section 3 of [RFC6973], including terms such as "(un)linkability" and "anonymity set".

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

3. Compression Details

The LOWPAN_IPHC encoding format specified in Section 3.1 of RFC 6282 [RFC6282] defines a method for deriving IIDs from the link-layer source and/or destination addresses in the encapsulation header. Unicast IPv6 addresses may be compressed to 64, 16, or 0 bits in the encoded IPv6 header.

3.1. Use of IEEE-Identifier-Based Addresses

As noted earlier, some threats could be mitigated using per-network "randomized" IEEE identifiers with 46 or more bits of entropy. A number of such proposals can be found at <https://mentor.ieee.org/privecsg/documents>, and Section 10.8 of [BTCorev4.1] specifies one for Bluetooth. Using IPv6 addresses derived from such IEEE identifiers would be roughly equivalent to those specified in [RFC7217].

Such addresses would be encoded as usual using the LOWPAN_IPHC encoding format. For example, if the source and destination addresses are both on-link and derived from the IEEE identifier in the encapsulating header:

o SAC (Source Address Compression) is set to 0 to indicate stateless compression.

o SAM (Source Address Mode) is set to 11 to indicate the address is fully elided and can be computed from the encapsulating header.

o DAC (Destination Address Compression) is set to 0 to indicate stateless compression.

o DAM (Destination Address Mode) is set to 11 to indicate the address is fully elided and can be computed from the encapsulating header.

3.2. Use of 16-Bit Short Addresses

An IPv6 address formed (per Section 6 of [RFC4944]) from an 16-bit identifier such as an IEEE 802.15.4 16-bit short address does not provide sufficient entropy to fully mitigate address scanning, as the size of the address scan search space depends on the entropy in the IID, and only 15 bits are available for unicast addresses. An adversary could also use statisical methods to determine the size of the L2 address space and thereby make some inference regarding the underlying technology being IEEE 802.15.4 on a given link. As such, this address generation mechanism SHOULD NOT be used on networks where privacy threats may be an issue, such as any networks that have Internet connectivity.

It might be possible to construct IPv6 addresses from 16-bit short addresses using an alternate mechanism that mitigates address scans, if all nodes on a given L2 network have a shared secret (such as the key needed to get on the layer-2 network) and generate the IID by using a one-way 64-bit hash of the shared secret together with the short address. The use of such a hash would result in the IIDs being spread out among the full range of IID address space.

"Temporary" addresses could possibly be generated in the same way by also including in the hash the Version Number from the Authoritative Border Router Option (ABDO) if any. This would allow changing temporary addresses whenever the Version Number is changed (even if the set of prefix or context information is unchanged). Such a scheme would likely require using the Context Identifier (CID) to distinguish between non-temporary addresses, "current" temporary addresses, and "past" temporary addresses based on a previous Version Number.

Specifying further details of such a scheme is left for future versions of this draft, if there is interest.

3.3. Use of Non-IEEE-Identifier-Based Addresses

Unicast addresses that are not IEEE-identifier based could be compressed to 0 bits as follows, using stateful context-based compression where the entire IPv6 address including the IID (as opposed to only the IPv6 prefix) are covered by context information. It is also worth pointing out that this same scheme would also allow compressing DHCPv6-assigned addresses even in networks where privacy is not a primary concern, thus potentially providing efficiency benefits in addition to privacy and security ones. Furthermore, unlike stateless compression, stateful context-based compression could also allow compressing addresses of nodes outside the local network (i.e., where the IEEE identifier in the encapsulating header is that of a router rather than the peer, and the peer’s address does not have a prefix in the local network) and hence can provide greater savings in such cases.

3.3.1. Source Address Compression

SAC (Source Address Compression) MUST be set to 1 to indicate stateful context-based compression.

SAM (Source Address Mode) MUST be set to 11 to indicate that the address is fully elided.

3.3.2. Destination Address Compression

DAC (Destination Address Compression) MUST be set to 1 to indicate stateful context-based compression.

DAM (Destination Address Mode) MUST be set to 11 to indicate that the address is fully elided.

3.3.3. Context Identifier

When non-IEEE-identifier-based addresses are used as described in this document, each address MUST be associated with a separate context. That is, the "prefix" associated with a context MUST be the full 128 bits of the IPv6 address.

LOWPAN_IPHC supports up to 16 source address contexts and 16 destination address contexts, allowing for simultaneous use of up to

16 source addresses and 16 destination addresses that are not IEEE-identifier-based. Context 0 is the default context if the CID (Context Identifier Extension) octet is absent, and other values require the CID to be present. As such, the address most commonly used (typically either the stable non-temporary address, or the currently preferred temporary address) could be assigned to context 0 so that the presence of the CID octet is minimized.

3.3.4. Context State

As specified in [RFC6775], context state is distributed by routers and is shared across a LoWPAN. This means that the use of CIDs described above would only support compression of 16 source and destination addresses across the entire LoWPAN. However, Section 8 of [RFC6775] explicitly allows for such context dissemination to be substituted by alternatives defined in other specifications. We now describe such a substitute that would allow header compression with up to 16 source addresses and 16 destination addresses *per node*.

First, a context entry is defined to be indexed by a { link-layer address, CID } tuple, rather than just a CID. Second, each node is responsible for generating and disseminating the CIDs for its own IPv6 addresses.

Thus, each Neighbor Cache Entry (NCE) in a router conceptually contains the CID of the neighbor's address, used when compressing packets sent to it.

3.3.5. Context Distribution

To disseminate CID information from a host to a router, the Address Registration Option (ARO) defined in Section 4.1 of [RFC6775] can be extended to include the CID by using 5 of the 24 Reserved bits (one for a flag to denote a CID is present, and 4 for the CID). For distribution in a multihop network, the Duplicate Address Request (DAR) and Duplicate Address Confirmation (DAC) messages can be similarly extended to include the CID in currently Reserved bits.

To disseminate CID information from a router to a host, Section 4.2 of [RFC6775] defines the 6LoWPAN Context Option (6CO) for use in Router Discovery. If a router sees that a host is sending packets without compressing a source or destination address, the router could send it an updated 6CO with a CID for that address as the context prefix, to allow compression of subsequent packets. Since each non-IEEE-identifier-based address requires its own context, the Context Length field MUST be set to 128 in the 6CO containing such context information. Note that the CID in a 6CO for another address within the 6LoWPAN is still generated by the router (since it is specific to the router’s link-layer address as used by the host to which the 6CO is sent); it is not the same value as the CID generated by the destination node itself, which CID is used by its router when forwarding a packet to it. Thus a router is responsible for updating CIDs in packets it forwards, just as it updates the link-layer source and destination addresses in the encapsulating header.

Specifying further details of such a scheme is left for future versions of this draft, if there is interest.

3.3.6. Negotiation

To negotiate using the substitute mechanisms above, rather than the default mechanisms specified in [RFC6775], the 6LoWPAN Capability Indication Option (6CIO) could be used as allowed for in Section 3.4 of [RFC7400] by assigning one of the "6LoWPAN capability Bits" for this purpose.

3.3.7. Discussion of Tradeoffs

This proposal decentralizes a portion of context generation and distribution to include simple nodes. In many 6LoWPAN scenarios, as much as possible is offloaded to router nodes precisely because end nodes are so limited. Until context info is learned for a given destination address, a node is not able to compress it. Compression would kick in after the context info is known. After context info is learned, the 4-bit CID must be stored for the destination address. As such, using this scheme requires a slight amount of overhead in the initial packet(s) but no additional overhead afterwards, and it requires no additional memory overhead initially, but a slight amount of additional memory overhead after context is learned.

In the rare case that a simple node needs to simultaneously communicate with more than 16 other non-IEEE-identifier-based destination addresses, at most 16 of them will be able to be compressed, and the others will have additional packet overhead.

4. IANA Considerations

The approach described in Section 3.3 would require IANA to allocate a bit in the "6LoWPAN capability Bits" subregistry for this purpose.

5. Security Considerations

This entire document is about security considerations and possible mitigations.

6. Acknowledgements

Thanks to Fernando Gont, Christian Huitema, and Gabriel Montenegro for discussion on the ideas described in this draft.

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.

[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.

[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.

[RFC7400] Bormann, C., "6LoWPAN-GHC: Generic Header Compression for IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs)", RFC 7400, November 2014.

7.2. Informative References

[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.

[RFC3972] Aura, T., "Cryptographically Generated Addresses (CGA)", RFC 3972, March 2005.

[RFC4429] Moore, N., "Optimistic Duplicate Address Detection (DAD) for IPv6", RFC 4429, April 2006.

[RFC4941] Narten, T., Draves, R., and S. Krishnan, "Privacy Extensions for Stateless Address Autoconfiguration in IPv6", RFC 4941, September 2007.

[RFC5535] Bagnulo, M., "Hash-Based Addresses (HBA)", RFC 5535, June 2009.

[RFC6496] Krishnan, S., Laganier, J., Bonola, M., and A. Garcia-Martinez, "Secure Proxy ND Support for SEcure Neighbor Discovery (SEND)", RFC 6496, February 2012.

[RFC6724] Thaler, D., Draves, R., Matsumoto, A., and T. Chown, "Default Address Selection for Internet Protocol Version 6 (IPv6)", RFC 6724, September 2012.

[RFC6973] Cooper, A., Tschofenig, H., Aboba, B., Peterson, J., Morris, J., Hansen, M., and R. Smith, "Privacy Considerations for Internet Protocols", RFC 6973, July 2013.

[RFC7136] Carpenter, B. and S. Jiang, "Significance of IPv6 Interface Identifiers", RFC 7136, February 2014.

[RFC7217] Gont, F., "A Method for Generating Semantically Opaque Interface Identifiers with IPv6 Stateless Address Autoconfiguration (SLAAC)", RFC 7217, April 2014.

[RFC7288] Thaler, D., "Reflections on Host Firewalls", RFC 7288, June 2014.

[RFC7421] Carpenter, B., Chown, T., Gont, F., Jiang, S., Petrescu, A., and A. Yourtchenko, "Analysis of the 64-bit Boundary in IPv6 Addressing", RFC 7421, January 2015.

[I-D.ietf-6man-ipv6-address-generation-privacy]
Cooper, A., Gont, F., and D. Thaler, "Privacy Considerations for IPv6 Address Generation Mechanisms", draft-ietf-6man-ipv6-address-generation-privacy-03 (work in progress), January 2015.

[I-D.ietf-6man-default-iids]
Gont, F., Cooper, A., Thaler, D., and W. Will, "Recommendation on Stable IPv6 Interface Identifiers", draft-ietf-6man-default-iids-02 (work in progress), January 2015.

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

Appendix A. Amount of Entropy Needed

In terms of privacy threats discussed in [I-D.ietf-6man-ipv6-address-generation-privacy], the one with the need for the most entropy is address scans. To mitigate address scans, one needs enough entropy to make the probability of a successful address probe be negligible. Typically this is measured in the length of time it would take to have a 50% probability of getting at least one hit. Address scans often rely on sending a packet such as a TCP SYN or ICMP Echo Request, and determining whether the reply is an ICMP unreachable errors (if no host exists) or TCP response or ICMP Echo Reply (if a host exists), or neither in which case nothing is known for certain.

Many privacy-sensitive devices support a "stealth mode" as discussed in Section 5 of [RFC7288] whereby they will not send a TCP RST or ICMP Echo Reply. In such cases, and when the device does not listen on a well-known TCP port known to the scanner, the effectiveness of an address scan is limited by the ability to get ICMP unreachable errors, since the attacker can only infer the presence of a host based on the absense of an ICMP unreachable error.

Generation of ICMP unreachable errors is typically rate limited to 2 per second (the default in routers such as Cisco routers running IOS 12.0 or later). Such a rate results in taking about a year to completely scan 26 bits of space. For a network with at most 2^16 devices on the same subnet, and the average lifetime of a device being 16 (2^4) years or less, this results in a need for at least 46 bits of entropy (16+26+4) so that a address scan would need to be sustained for longer than the lifetime of devices to have a 50% chance of getting a hit.

The actual math is as follows. Let 2^N be the number of devices on the subnet. Let 2^M be the size of the space to scan (i.e., M bits of entropy). Let S be the number of scan attempts. The formula is: P(at least one success) = 1 - (1 - 2^N/2^M)^S = 1/2. Assuming 2^M >> S, this simplifies to: S * 2^N/2^M = 1/2, giving S = 2^(M-N) / 2, or M = N + log_2(2S).

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

If, on the other hand, the devices being scanned for do not implement a "stealth mode", but respond with TCP RST or ICMP Echo Reply packets, then the address scan is not limited by the ICMP unreachable rate limit in routers, since the attacker can determine the presence of a host without them. In such cases, more bits of entropy would be needed to provide the same level of protection.

Author's Address

Dave Thaler
Microsoft
One Microsoft Way
Redmond, WA 98052
USA

Email: dthaler@microsoft.com

6lo
Internet-Draft
Updates: 4944 (if approved)
Intended status: Standards Track
Expires: January 3, 2016

P. Thubert, Ed.
Cisco
C. Bormann
Uni Bremen TZI
L. Toutain
IMT-TELECOM Bretagne
R. Cragie
ARM

July 02, 2015

A Routing Header Dispatch for 6LoWPAN
draft-thubert-6lo-routing-dispatch-05

Abstract

This specification provides a new 6LoWPAN dispatch type for use in Route-over and mixed Mesh-under and Route-over topologies, that reuses the encoding of the mesh type defined in RFC 4944 for pure Mesh-under topologies. This specification also defines a method to compress RPL Option (RFC6553) information and Routing Header type 3 (RFC6554), an efficient IP-in-IP technique and opens the way for further routing techniques. This extends 6LoWPAN Transmission of IPv6 Packets (RFC4944), and is applicable to new link-layer types where 6LoWPAN is being defined.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

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

Controlling the amount of data transmission is one possible venue to save energy. In a number of LLN standards, the frame size is limited to much smaller values than the IPv6 maximum transmission unit (MTU) of 1280 bytes. In particular, an LLN that relies on the classical Physical Layer (PHY) of IEEE 802.14.5 [IEEE802154] is limited to 127 bytes per frame. The need to compress IPv6 packets over IEEE 802.14.5 led to the 6LoWPAN Header Compression [RFC6282] work (6LoWPAN-HC).

Innovative Route-over techniques have been and are still being developed for routing inside a LLN. In a general fashion, such techniques require additional information in the packet to provide loop prevention and to indicate information such as flow identification, source routing information, etc.

For reasons such as security and the capability to send ICMP errors back to the source, an original packet must not be tampered with, and any information that must be inserted in or removed from an IPv6 packet must be placed in an extra IP-in-IP encapsulation. This is the case when the additional routing information is inserted by a router on the path of a packet, for instance a mesh root, as opposed to the source node. This is also the case when some routing information must be removed from a packet that will flow outside the LLN.

As an example, the Routing Protocol for Low Power and Lossy Networks [RFC6550] (RPL) is designed to optimize the routing operations in constrained LLNs. As part of this optimization, RPL requires the addition of RPL Packet Information (RPI) in every packet, as defined in Section 11.2 of [RFC6550].

The RPL Option for Carrying RPL Information in Data-Plane Datagrams [RFC6553] specification indicates how the RPI can be placed in a RPL Option for use in an IPv6 Hop-by-Hop header. This representation demands a total of 8 bytes when in most cases the actual RPI payload requires only 19 bits. Since the Hop-by-Hop header must not flow outside of the RPL domain, it must be removed from packets that leave the domain, and be inserted in packets entering the domain. In both cases, this operation implies an IP-in-IP encapsulation.

```
           ------+---------                            ^
                 |          Internet                   |
                 |                                     | Native IPv6
              +-----+                                  |
              |     | Border Router (RPL Root)    ^    |    ^
              |     |                             |    |    |
              +-----+                             |    |    | IPv6 in
                 |                                |    |    | IPv6
           o    o   o    o                        |    |    | + RPI
       o o   o  o   o  o  o o   o                 |    |    |  or RH3
      o o o  o o    o   o   o  o  o               |    |    |
      o   o    o  o     o  o    o  o  o           |    |    |
     o  o   o  o   o         o   o o              v    v    v
     o          o             o     o
                       LLN
```

Figure 1: IP-in-IP Encapsulation within the LLN

Additionally, in the case of the Non-Storing Mode of Operation (MOP), RPL requires a Routing Header type 3 (RH3) as defined in the IPv6 Routing Header for Source Routes with RPL [RFC6554] specification, for all packets that are routed down a RPL graph. With Non-Storing RPL, even if the source is a node in the same LLN, the packet must first reach up the graph to the root so that the root can insert the RH3 to go down the graph. In any fashion, whether the packet was originated in a node in the LLN or outside the LLN, and regardless of whether the packet stays within the LLN or not, as long as the source of the packet is not the root itself, the source-routing operation also implies an IP-in-IP encapsulation at the root to insert the RH3.

6TiSCH [I-D.ietf-6tisch-architecture] specifies the operation of IPv6 over the TimeSlotted Channel Hopping [I-D.ietf-6tisch-tsch] (TSCH) mode of operation of IEEE 802.14.5. The architecture requires the use of both RPL and the 6lo adaptation layer framework ([RFC4944], [RFC6282]) over IEEE 802.14.5. Because it inherits the constraints on the frame size from the MAC layer, 6TiSCH cannot afford to spend 8 bytes per packet on the RPI. Hence the requirement for a 6LoWPAN header compression of the RPI.

The type of information that needs to be present in a packet inside the LLN but not outside of the LLN varies with the routing operation, but there is overall a need for an extensible compression technique that would simplify the IP-in-IP encapsulation, when needed, and optimally compress existing routing artifacts found in LLNs.

This specification extends 6LoWPAN [RFC4944] and in particular reuses the Mesh Header formats that are defined for the Mesh-under use cases so as to carry routing information for Route-over use cases. The specification includes the formats necessary for RPL and is extensible for additional formats.

2. Terminology

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

The Terminology used in this document is consistent with and incorporates that described in ‘Terminology in Low power And Lossy Networks’ [RFC7102] and [RFC6550].

The terms Route-over and Mesh-under are defined in [RFC6775].

Other terms in use in LLNs are found in [RFC7228].

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

3. Updating RFC 4944

This draft proposes 3 ways to adapt 6LoWPAN while maintaining backward compatibility with IPv6 over IEEE 802.15.4 [RFC4944].

Option 1 considers that a network where this specification applies is physically separate from a network where the Mesh Header defined in [RFC4944] is used. With that assumption, the Mesh Header dispatch space can be reused. A variation is proposed whereby the NALP pattern 00xxxxxx is reused instead of the Mesh Header pattern.

Option 2 defines a new Separator Dispatch value that indicates that no Mesh Header is present in the remainder of the packet. If the 10xxxxxx pattern is found in the packet after this new Separator Dispatch, then this specification applies. It is suggested that the new Separator Dispatch would also enable to reuse patterns 00xxxxxx and 11xxxxxx in the future.

Option 3 uses values in pattern 11xxxxxx that are free to this date, avoiding patterns 11000xxx and 11100xxx that are used for the Fragmentation Header as defined in [RFC4944].

3.1. Reusing Mesh Header (or NALP) Dispatch Space

Section 5.1 of the IPv6 over IEEE 802.15.4 [RFC4944] specification defines various Dispatch Types and Headers, and in particular a Mesh Header that corresponds to a pattern 10xxxxxx and effectively consumes one third of the whole 6LoWPAN dispatch space for Mesh-under specific applications.

This specification reuses the Dispatch space for Route-over and mixed operations. This means that a device that use the Mesh Header as specified in [RFC4944] should not be placed in a same network as a device which operates per this update. This is generally not a problem since a network is classically either Mesh-under OR Route-over.

A new implementation of Mesh-under MAY support both types of encoding, and if so, it SHOULD provide a management toggle to enable either mode and it SHOULD use this specification as the default mode.

A dispatch space of equivalent size to the Mesh Header was reserved in [RFC4944] for external specifications Not A LowPan (NALP), hoping that such specification could coexist harmlessly on a same network as early 6LoWPAN.

It is unclear that this disposition was useful at some point and that NALP was effectively used in a network where 6LoWPAN is deployed. A variation of the suggestion above would be, to use pattern 10xxxxxx instead of pattern 10xxxxxx If deemed necessary, it would be possible to move NALP to some other (smaller) dispatch space.

3.2. Add A New Dispatch

The suggestion here is not to use the Escape Dispatch, which is not entirely defined at this point, but to block one other dispatch value (say 11111111) to indicate that from that point on, the parsing of the packet should use this specification if the pattern 10xxxxxx is found.

The expectation is that if there is a Mesh Header, it is placed early in the packet and from there this specification will apply to any other appearance of the 10xxxxxx pattern. On the other hand, if there is no mesh header, there is a need to indicate so with this new dispatch value, and then any appearance of the 10xxxxxx pattern will be parsed per this specification.

It must be noted that the NALP space is really reserved for the first dispatch in the 6LoWPAN packet. Once a packet is identified as a 6LoWPAN packet by a first dispatch, the NALP range could be used.

Finally, the specification indicates that Fragments Headers must always preceed Routing header.

As a result, the 11111111 pattern could be considered a delimiminator between a portion of the frame that is formatted per [RFC4944] on the left, and a portion from which the space for Mesh Header, Fragment Header and NALP can be reused, on the right. This specification would reuse the Mesh Header or the NALP as discuused above, so the text in this specification is not impacted.

3.3. Use Free Space the the FRAG range

With the third proposal, the 6LoRH uses free bit patterns that are defined in [RFC4944] in the 11 xxxxxx range, avoiding FRAG1 of 11 000xxx and FRAGN of 11 100xxx.

The third bit, which differentiates FRAG1 from FRAGN in their particular ranges, indicates Elective vs. Critical; the fourth bit is always set to ensure that the 6LoRH does not collision with FRAG1 or FRAGN. The net result is one bit less than in the other proposals for the encoding space in the 6LoRH, which means only 4 bits to encode the length in the Elective format, as discussed below, and only 4 bits TSE.

The resulting formats and consequences are detailed in the relevant sections.

4. Placement of 6LoRH

One or more 6LoRHs MAY be placed in a 6LoWPAN packet and MUST always be placed before the LOWPAN_IPHC [RFC6282]. A 6LoRH MUST always be placed after Fragmentation Header and Mesh Header [RFC6282].

5. General Format

The 6LoWPAN Routing Header (6LoRH) may contain source routing information such as a compressed form of RH3, or other sorts of routing information such as the RPL RPI, source and/or destination address, and is extensible for future uses, with the given example of BIER bitmap encoding in Section 10.

There are two forms for 6LoRH:

Elective (6LoRHE)

Critical (6LoRHC)

This specification proposes several alternatives for the 6LoRH encoding:

Reuse Mesh Header Space in route over mode

Same as above, signaled by an initial escape byte

a more complex encoding using other coding space that is still free in the 6lo adaptation layer framework

The layout of the Elective and Critical forms depends on the encoding of the 6LoRH itself.

With the Mesh Header reuse proposal, the 6LoRH reuses the bit patterns that are defined in [RFC4944] for the Mesh Header, specifically the Dispatch Value Bit Pattern of 10xxxxxx.

With the Escaped Mesh Header reuse, the 6LoRH also reuses the bit patterns that are defined in [RFC4944] for the Mesh Header, but an ESC dispatch, with a value of 11111111, must be placed before the first 6LoRH. The ESC indicates that the parsing of the pattern of 10xxxxxx will now be performed following this specification. There is no need to place an ESC before each 6LoRH, since the ESC influences the parsing of the rest of the packet.

5.1. Elective Format

With the first and second proposals, the 6LoRHE uses the Dispatch Value Bit Pattern of 101xxxxx.
A 6LoRHE may be ignored and skipped in parsing.
If it is ignored, the 6LoRHE is forwarded with no change inside the LLN.

```
       0                   1
       0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-       ...        -+
      |1|0|1| Length  |      Type     |                    |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-       ...        -+
                                     <--    Length    -->
```

Figure 2: Elective 6LoWPAN Routing Header

With the third proposal 6LoRHE uses the Dispatch Value Bit Pattern of 1111xxxx.

```
       0                   1
       0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-       ...        -+
      |1|1|1|1| Length|      Type     |                    |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-       ...        -+
                                      <--    Length    -->
```

Figure 3: Elective 6LoWPAN Routing Header

Length:
Length of the 6LoRHE expressed in bytes, excluding the first 2 bytes. This is done to enable a node to skip a 6LoRH that it does not support and/or cannot parse, for instance if the Type is not known.

Type:
Type of the 6LoRHE

5.2. Critical Format

With the first and second proposals, the 6LoRHC uses the Dispatch Value Bit Pattern of 100xxxxx.
A node which does not support the 6LoRHC Type MUST silently discard the packet (note that there is no provision for the exchange of error messages; such a situation should be avoided by judicious use of administrative control and/or capability indications).

```
     0                   1
     0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-              ...             -+
    |1|0|0|   TSE   |      Type     |                                |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-              ...             -+
                                  <-- Length implied by Type/TSE -->
```

Figure 4: Critical 6LoWPAN Routing Header

With the third proposal 6LoRHE uses the Dispatch Value Bit Pattern of 1101xxxx.

```
     0                   1
     0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-              ...             -+
    |1|1|0|1|  TSE  |      Type     |                                |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-              ...             -+
                                  <-- Length implied by Type/TSE -->
```

TSE:

Type Specific Extension. The meaning depends on the Type, which must be known in all of the nodes. The interpretation of the TSE depends on the Type field that follows. For instance, it may be used to transport control bits, the number of elements in an array, or the length of the remainder of the 6LoRHC expressed in a unit other than bytes.

Type:
Type of the 6LoRHC

6. The Routing Header type 3 (RH3) 6LoRH

The Routing Header type 3 (RH3) 6LoRH (RH3-6LoRH) is a Critical 6LoWPAN Routing Header that provides a compressed form for the RH3, as defined in [RFC6554] for use by RPL routers. Routers that need to forward a packet with a RH3-6LoRH are expected to be RPL routers and expected to support this specification. If a non-RPL router receives a packet with a RPI-6LoRH, this means that there was a routing error and the packet should be dropped so the Type cannot be ignored.

```
  0                   1
  0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-    -+-    -+ ... +-    -+
 |1|0|0|  Size   |6LoRH Type 0..4| Hop1 | Hop2 |     | HopN |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-    -+-    -+ ... +-    -+
```

Size indicates the number of compressed addresses

Figure 5: The RH3-6LoRH

The values for the RH3-6LoRH Type are an enumeration, 0 to 4. The form of compression is indicated by the Type as follows:

Type	Size Unit
0	1
1	2
2	4
3	8
4	16

Figure 6: The RH3-6LoRH Types

In the case of a RH3-6LoRH, the TSE field is used as a Size, which encodes the number of hops minus 1; so a Size of 0 means one hop, and the maximum that can be encoded is 32 hops. (If more than 32 hops need to be expressed, a sequence of RH3-6LoRH can be employed.)

The next Hop is indicated in the first entry of the first RH3-6LoRH. Upon reception, the entry is checked whether it refers to the processing router itself. If it so, the entry is removed from the RH3-6LoRH and the Size is decremented. If the Size is now zero, the whole RH3-6LoRH is removed. If there is no more RH3-6LoRH, the processing node is the last router on the way, which may or may not be collocated with the final destination.

The last hop in the last RH3-6LoRH is the last router prior to the destination in the LLN. So even when there is a RH3-6LoRH in the frame, the address of the final destination is in the LoWPAN_IPHC [RFC6282].

If some bits of the first address in the RH3-6LoRH can be derived from the final destination is in the LoWPAN_IPHC, then that address may be compressed, otherwise is is expressed in full. Next addresses only need to express the delta from the previous address.

All addresses in a RH3-6LoRH are compressed in a same fashion, down to the same number of bytes per address. In order to get different forms of compression, multiple consecutive RH3-6LoRH must be used.

7. The RPL Packet Information 6LoRH

[RFC6550], Section 11.2, specifies the RPL Packet Information (RPI) as a set of fields that are to be added to the IP packets for the purpose of Instance Identification, as well as Loop Avoidance and Detection.

In particular, the SenderRank, which is the scalar metric computed by an specialized Objective Function such as [RFC6552], indicates the Rank of the sender and is modified at each hop. The SenderRank allows to validate that the packet progresses in the expected direction, either upwards or downwards, along the DODAG.

RPL defines the RPL Option for Carrying RPL Information in Data-Plane Datagrams [RFC6553] to transport the RPI, which is carried in an IPv6 Hop-by-Hop Options Header [RFC2460], typically consuming eight bytes per packet.

With [RFC6553], the RPL option is encoded as six Octets; it must be placed in a Hop-by-Hop header that consumes two additional octets for a total of eight. In order to limit its range to the inside the RPL domain, the Hop-by-Hop header must be added to (or removed from) packets that cross the border of the RPL domain.

The 8-bytes overhead is detrimental to the LLN operation, in particular with regards to bandwidth and battery constraints. These bytes may cause a containing frame to grow above maximum frame size, leading to Layer 2 or 6LoWPAN [RFC4944] fragmentation, which in turn cause even more energy spending and issues discussed in the LLN Fragment Forwarding and Recovery [I-D.thubert-6lo-forwarding-fragments].

An additional overhead comes from the need, in certain cases, to add an IP-in-IP encapsulation to carry the Hop-by-Hop header. This is needed when the router that inserts the Hop-by-Hop header is not the source of the packet, so that an error can be returned to the router. This is also the case when a packet originated by a RPL node must be stripped from the Hop-by-Hop header to be routed outside the RPL domain.

This specification defines an IPinIP-6LoRH in Section 8 for that purpose, but it must be noted that stripping a 6LoRH does not require a manipulation of the packet in the LOWPAN_IPHC, and thus, if the source address in the LOWPAN_IPHC is the node that inserted the IPinIP-6LoRH then this alone does not mandate an IPinIP-6LoRH.

As a result, a RPL packet may bear only a RPI-6LoRH and no IPinIP-6LoRH. In that case, the source and destination of the packet are located in the LOWPAN_IPHC.

As with [RFC6553], the fields in the RPI include an ’O’, an ’R’, and an ’F’ bit, an 8-bit RPLInstanceID (with some internal structure), and a 16-bit SenderRank.

The remainder of this section defines the RPI-6LoRH, a Critical 6LoWPAN Routing Header that is designed to transport the RPI in 6LoWPAN LLNs.

7.1. Compressing the RPLInstanceID

RPL Instances are discussed in [RFC6550], Section 5. A number of simple use cases will not require more than one instance, and in such a case, the instance is expected to be the global Instance 0. A global RPLInstanceID is encoded in a RPLInstanceID field as follows:

```
 0 1 2 3 4 5 6 7
+-+-+-+-+-+-+-+-+
|0|     ID      |  Global RPLInstanceID in 0..127
+-+-+-+-+-+-+-+-+
```

Figure 7: RPLInstanceID Field Format for Global Instances

For the particular case of the global Instance 0, the RPLInstanceID field is all zeros. This specification allows to elide a RPLInstanceID field that is all zeros, and defines a I flag that, when set, signals that the field is elided.

7.2. Compressing the SenderRank

The SenderRank is the result of the DAGRank operation on the rank of the sender; here the DAGRank operation is defined in [RFC6550], Section 3.5.1, as:

```
DAGRank(rank) = floor(rank/MinHopRankIncrease)
```

If MinHopRankIncrease is set to a multiple of 256, the least significant 8 bits of the SenderRank will be all zeroes; by eliding those, the SenderRank can be compressed into a single byte. This idea is used in [RFC6550] by defining DEFAULT_MIN_HOP_RANK_INCREASE as 256 and in [RFC6552] that defaults MinHopRankIncrease to DEFAULT_MIN_HOP_RANK_INCREASE.

This specification allows to encode the SenderRank as either one or two bytes, and defines a K flag that, when set, signals that a single byte is used.

7.3. The Overall RPI-6LoRH encoding

The RPI-6LoRH provides a compressed form for the RPL RPI. Routers that need to forward a packet with a RPI-6LoRH are expected to be RPL routers and expected to support this specification. If a non-RPL router receives a packet with a RPI-6LoRH, this means that there was a routing error and the packet should be dropped so the Type cannot be ignored.

Since the I flag is not set, the TSE field does not need to be a length expressed in bytes. The field is fully reused for control bits so as to encode the O, R and F flags from the RPI, and the I and K flags that indicate the compression that is taking place.

The Type for the RPI-6LoRH is 5 with the first proposal and in the range 5-8 with the third proposal.

The RPI-6LoRH is immediately followed by the RPLInstanceID field, unless that field is fully elided, and then the SenderRank, which is either compressed into one byte or fully in-lined as the whole 2 bytes. The I and K flags in the RPI-6LoRH indicate whether the RPLInstanceID is elided and/or the SenderRank is compressed and depending on these bits, the Length of the RPI-6LoRH may vary as described hereafter.

```
       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
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+  ...  -+-+-+
      |1|0|0|O|R|F|I|K| 6LoRH Type=5  |   Compressed fields  |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+  ...  -+-+-+
```

Figure 8: The Generic RPI-6LoRH Format

O, R, and F bits:
      The O, R, and F bits as defined in [RFC6550], Section 11.2.

I bit:
      If it is set, the Instance ID is elided and the RPLInstanceID is the Global RPLInstanceID 0. If it is not set, the octet immediately following the type field contains the RPLInstanceID as specified in [RFC6550] section 5.1.

K bit:
      If it is set, the SenderRank is be compressed into one octet, and the lowest significant octet is elided. If it is not set, the SenderRank, is fully inlined as 2 octets.

In Figure 9, the RPLInstanceID is the Global RPLInstanceID 0, and the MinHopRankIncrease is a multiple of 256 so the least significant byte is all zeros and can be elided:

```
       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
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |1|0|0|O|R|F|1|1| 6LoRH Type=5  | SenderRank    |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
                I=1, K=1
```

Figure 9: The most compressed RPI-6LoRH

In Figure 10, the RPLInstanceID is the Global RPLInstanceID 0, but both bytes of the SenderRank are significant so it can not be compressed:

```
       0                   1                   2                   3
       0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |1|0|0|O|R|F|1|0| 6LoRH Type=5  |        SenderRank             |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
                I=1, K=0
```

Figure 10: Eliding the RPLInstanceID

In Figure 11, the RPLInstanceID is not the Global RPLInstanceID 0, and the MinHopRankIncrease is a multiple of 256:

```
    0                   1                   2                   3
    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |1|0|0|O|R|F|0|1| 6LoRH Type=5  | RPLInstanceID |  SenderRank   |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
             I=0, K=1
```

Figure 11: Compressing SenderRank

In Figure 12, the RPLInstanceID is not the Global RPLInstanceID 0, and both bytes of the SenderRank are significant:

```
    0                   1                   2                   3
    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |1|0|0|O|R|F|0|0| 6LoRH Type=5  | RPLInstanceID |    Sender-...
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     ...-Rank      |
   +-+-+-+-+-+-+-+-+
             I=0, K=0
```

Figure 12: Least compressed form of RPI-6LoRH

A typical packet in RPL non-storing mode going down the RPL graph requires an IPinIP encapsulating the RH3, whereas the RPI is usually omitted, unless it is important to indicate the RPLInstanceID. To match this structure, an optimized IPinIP 6LoRH is defined in Section 8.

With the third approach, the format becomes:

```
    0                   1
    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
   |1|1|0|1|O|R|F|r|6LoRH Type 5-8 | ...
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
              r = reserved
```

Figure 13: Third encoding

And the types include the setting of I and K as follows:

Type	I	K
5	0	0
6	0	1
7	1	0
8	1	1

Figure 14: The RPI-6LoRH Types

8. The IP-in-IP 6LoRH

The IP-in-IP 6LoRH (IPinIP-6LoRH) is an Elective 6LoWPAN Routing Header that provides a compressed form for the encapsulating IPv6 Header in the case of an IP-in-IP encapsulation.

An IPinIP encapsulation is used to insert a field such as a Routing Header or an RPI at a router that is not the source of the packet. In order to send an error back regarding the inserted field, the address of the router that performs the insertion must be provided.

The encapsulation can also enable a router down the path removing a field such as the RPI, but this can be done in the compressed form by removing the RPI-6LoRH, so an IPinIP-6LoRH encapsulation is not required for that sole purpose.

This field is not critical for routing so the Type can be ignored, and the TSE field contains the Length in bytes.

```
     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
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-       ...      -+
    |1|0|1| Length  | 6LoRH Type 9  |  Hop Limit    | Encaps. Address  |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-       ...      -+
```

Figure 15: The IPinIP-6LoRH

The Length of an IPinIP-6LoRH is expressed in bytes and MUST be at least 1, to indicate a Hop Limit (HL), that is decremented at each hop. When the HL reaches 0, the packet is dropped per [RFC2460]

If the Length of an IPinIP-6LoRH is exactly 1, then the Encapsulator Address is elided, which means that the Encapsulator is a well-known router, for instance the root in a RPL graph.

If the Length of an IPinIP-6LoRH is strictly more than 1, then an Encapsulator Address is placed in a compressed form after the Hop Limit field. The value of the Length indicates which compression is performed on the Encapsulator Address. For instance, a Size of 3 indicates that the Encapsulator Address is compressed to 2 bytes.

When it cannot be elided, the destination IP address of the IP-in-IP header is transported in a RH3-6LoRH as the first address of the list.

With RPL, the destination address in the IP-in-IP header is implicitly the root in the RPL graph for packets going upwards, and the destination address in the IPHC for packets going downwards. If the implicit value is correct, the destination IP address of the IP-in-IP encapsulation can be elided.

If the final destination of the packet is a leaf that does not support this specification, then the chain of 6LoRH must be stripped by the RPL/6LR router to which the leaf is attached. In that example, the destination IP address of the IP-in-IP header cannot be elided.

In the special case where the 6LoRH is used to route 6LoWPAN fragments, the destination address is not accessible in the IPHC on all fragments and can be elided only for the first fragment and for packets going upwards.

9. The Mesh Header 6LoRH

The Mesh Header 6LoRH (MH-6LoRH) is an Elective 6LoWPAN Routing Header that provides an alternate form for the Mesh Addressing Type and Header defined in [RFC4944] with the same semantics.

The MH-6LoRH is introduced as replacement for use in potentially mixed Route_Over and Mesh-under environments. LLN nodes that need to forward a packet with a MH-6LoRH are expected to support this specification. If a router that supports only Route-over receives a packet with a MH-6LoRH, this means that there was a routing error and the packet should be dropped, so the Type cannot be ignored.

The HopsLft field defined in [RFC4944] is encoded in the TSE, so this specification doubles the potential number of hops vs. [RFC4944] in the first proposal and is conserved in the third proposal.

The HopsLft value of 0x1F is reserved and signifies an 8-bit Deep Hops Left field immediately following the Type, and allows a source node to specify a hop limit greater than 30 hops.

```
    0                   1
    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-              ...             -+
   |1|0|0| HopsLft |6LoRHType 11-14| originator address, final address |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-              ...             -+
```

Figure 16: The MH-6LoRH

The V and F flags defined in [RFC4944] are encoded in the MH-6LoRH Type as follows:

Type	V	F
11	0	0
12	0	1
13	1	0
14	1	1

Figure 17: The MH-6LoRH Types

10. The BIER 6LoRH

(Note that the current contents of this section is a proof of concept only; the details for this encoding need to be developed in parallel with defining the semantics of a constrained version of BIER.)

The Bit Index Explicit Replication (BIER) 6LoRH (BIER-6LoRH) is an Elective 6LoWPAN Routing Header that provides a variable-size container for a BIER Bitmap. BIER can be used to route downwards a RPL graph towards one or more LLN node, as discussed in the BIER Architecture [I-D.wijnands-bier-architecture] specification. The capability to parse the BIER Bitmap is necessary to forward the packet so the Type cannot be ignored.

```
       0                   1
       0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ ... -+-+-+-    ...      -+
      |1|0|0|  Size   |6LoRHType 15-19| Control Fields |    bitmap     |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ ... -+-+-+-    ...      -+
```

Figure 18: The BIER-6LoRH

The Type for a BIER-6LoRH indicates the size of words used to build the bitmap and whether the bitmap is operated as an uncompressed bit-by-bit mapping, or as a Bloom filter.

In the bit-by-bit case, each bit is mapped in an unequivocal fashion with a single addressable resource in the network. This may rapidly lead to large bitmaps, and BIER allows to divide a network into groups that partition the network so that a given bitmap is locally significant to one group only. This specification allows to encode a 1-byte Group ID in the BIER-6LoRH Control Fields.

A Bloom Filter can be seen as a compression technique for the bitmap. A Bloom Filter may generate false positives, which, in the case of BIER, result in undue forwarding of a packet down a path where no listener exists.

As an example, the Constrained-Cast [I-D.bergmann-bier-ccast] specification employs Bloom Filters as a compact representation of a match or non-match for elements in a large set.

In the case of a Bloom Filter, a number of Hash functions must be run to obtain a multi-bit signature of an encoded element. This specification allows to signal an Identifier of the Hash functions being used to generate a certain bitmap, so as to enable a migration scenario where Hash functions are renewed. A Hash ID is signaled as a 1-byte value, and, depending on the Type, there may be up to 2 or up to 8 Hash IDs passed in the BIER-6LoRH Control Fields associated with a Bloom Filter bitmap, as follows:

Type	encoding	Control Fields	Word Size
15	bit-by-bit	none	32 bits
16	Bloom filter	2* 1-byte HashID	32 bits
17	bit-by-bit	none	128 bits
18	Bloom filter	8* 1-byte HashID	128 bits
19	bit-by-bit	1-byte GroupID	128 bits

Figure 19: The BIER-6LoRH Types

In order to address a potentially large number of devices, the bitmap may grow very large. Yet, the maximum frame size for a given MAC layer may limit the number of bits that can be dedicated to routing. The Size indicates the number of words in the bitmap minus one, so a size of 0 means one word, a Size of 1 means 64 2 words, up to a size of 31 which means 32 words.

11. Security Considerations

The security considerations of [RFC4944], [RFC6282], and [RFC6553] apply.

Using a compressed format as opposed to the full in-line format is logically equivalent and does not create an opening for a new threat when compared to [RFC6550], [RFC6553] and [RFC6554].

12. IANA Considerations

This document creates a IANA registry for the 6LoWPAN Routing Header Type, and assigns the following values:

0..4 : RH3-6LoRH [RFCthis]

5 : RPI-6LoRH [RFCthis]

9 : IPinIP-6LoRH [RFCthis]

11..14 : MH-6LoRH [RFCthis]

15..19 : BIER-6LoRH [RFCthis]

13. Acknowledgments

The authors wish to thank Martin Turon, James Woodyatt and Ralph Droms for constructive reviews to the design in the 6lo Working Group. The overall discussion involved participants to the 6MAN, 6TiSCH and ROLL WGs, thank you all. Special thanks to the chairs of the ROLL WG, Michael Richardson and Ines Robles, and Brian Haberman, Internet Area A-D, and Adrian Farrel, Routing Area A-D, for driving this complex effort across Working Groups and Areas.

14. References

14.1. Normative References

[IEEE802154]
IEEE standard for Information Technology, "IEEE std. 802.15.4, Part. 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks", 2015.

[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.

[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.

[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.

[RFC6552] Thubert, P., "Objective Function Zero for the Routing Protocol for Low-Power and Lossy Networks (RPL)", RFC 6552, March 2012.

[RFC6553] Hui, J. and JP. Vasseur, "The Routing Protocol for Low-Power and Lossy Networks (RPL) Option for Carrying RPL Information in Data-Plane Datagrams", RFC 6553, March 2012.

[RFC6554] Hui, J., Vasseur, JP., Culler, D., and V. Manral, "An IPv6 Routing Header for Source Routes with the Routing Protocol for Low-Power and Lossy Networks (RPL)", RFC 6554, March 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.

14.2. Informative References

[I-D.bergmann-bier-ccast]
Bergmann, O., Bormann, C., and S. Gerdes, "Constrained-Cast: Source-Routed Multicast for RPL", draft-bergmann-bier-ccast-00 (work in progress), November 2014.

[I-D.ietf-6tisch-architecture]
Thubert, P., "An Architecture for IPv6 over the TSCH mode of IEEE 802.15.4", draft-ietf-6tisch-architecture-08 (work in progress), May 2015.

[I-D.ietf-6tisch-tsch]
Watteyne, T., Palattella, M., and L. Grieco, "Using IEEE802.15.4e TSCH in an IoT context: Overview, Problem Statement and Goals", draft-ietf-6tisch-tsch-06 (work in progress), March 2015.

[I-D.thubert-6lo-forwarding-fragments]
Thubert, P. and J. Hui, "LLN Fragment Forwarding and Recovery", draft-thubert-6lo-forwarding-fragments-02 (work in progress), November 2014.

[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.

[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.

Authors’ Addresses

Pascal Thubert (editor)
Cisco Systems
Village d’Entreprises Green Side
400, Avenue de Roumanille
Batiment T3
Biot - Sophia Antipolis 06410
FRANCE

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

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

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

Laurent Toutain
Institut MINES TELECOM; TELECOM Bretagne
2 rue de la Chataigneraie
CS 17607
Cesson-Sevigne Cedex 35576
France

Email: Laurent.Toutain@telecom-bretagne.eu

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

Email: robert.cragie@gridmerge.com

```
individual submission                                          R. Turner
Internet-Draft                                                Landis+Gyr
Intended status: Standards Track                               June 2015
Expires: November 23, 2015
```

DHCPv6 Option for Configuration of 6LoWPAN Compression Contexts
draft-turner-dhcp-6co-00

Abstract

This document specifies a DHCPv6 option to configure hosts on a 6LoWPAN with IPv6 address compression information as required by stateful compression methods specified in RFC 6282. The option provides up to 16 prefixes that can be associated with specific instances of IPv6 address compression used in 6LoWPANs. Each prefix can be a variable length of bits, and includes a validity lifetime as well.

Status of This Memo

Copyright Notice

1. Introduction

RFC 6282 describes a procedure for the compression of IPv6 addresses in IP headers, and this same technique can be used to compress arbitrary IPv6 addresses. The compression algorithms contain default, stateless methods, as well as "stateful" address compression based on a set of compression "contexts". However, the method by which 6LoWPAN nodes acquire these contexts is out of scope of RFC 6282. RFC 6775 introduced a method by which router advertisements can include "6LowPAN Context Options" that communicate context information to devices on the 6LoWPAN network. RFC 6775 also introduced a way to limit the rate of multicast router advertisements to make these router advertisements more friendly to constrained LoWPANs. These mechanisms allow a stateless auto-configuration option for LoWPANs. DHCPv6 is both a stateful method for address configuration, as well as stateless alternative for configuration of 6LoWPAN devices. DHCPv6 provides extensibility through the support of TLV options in the protocol. This document specifies such a DHCPv6 option for configuring 6LoWPAN compression contexts.

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] [RFC2119].

2. Terminology

This document primarily uses the terminology described in [RFC6550], [RFC3315] and [RFC6282]. The terminology and concepts described in these documents will assist in the reading of this document.

3. DHCP Option format for 6LowPAN Compression Contexts

```
     0                   1                   2                   3
    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |         OPTION_V6_6CO         |         Option Length         |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |Context Length | Res |C| CID   |        Valid Lifetime         |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   .                                                               .
   .                       Context Prefix                          .
   .                                                               .
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 1: 6LoWPAN Context Option Format

Figure 1

The description of the option fields is provided below:

o OPTION_V6_6CO: The option-code per RFC 3315 (TBA by IANA)

o Option Length: 16-bit unsigned length (in bytes) of the entire option contents, including the type and length fields

o Context Length: 8-bit unsigned integer. The number (0 - 128) of valid leading bits in the "Context Prefix" field.

o Res: This field is currently unused. It MUST be initialized to zero by the server and MUST be ignored by the DHCPv6 client.

o CID: 4-bit Context Identifier for this prefix information. The CID is used by context-based header compression as specified in [RFC6282].

o Valid Lifetime: 16-bit unsigned integer. The length of time in units of 60 seconds (relative to the time the option is received) that the context is valid for the purpose of header compression or decompression. A value of zero indicates that no specific validity lifetime is specified (prefix validity does not expire).

o Context Prefix: The IPv6 prefix or address corresponding to the CID field. This field is padded with zeros in order to make the option a multiple of 8-bytes

There is one option per IPv6 context prefix, with each prefix option containing a "CID" that provides the context identifier (or index) in the range 0 to 15. This index is referred to by subsequent compressed IPv6 addresses to indicate which stateful prefix should be used to either compress or decompress a particular IPv6 address.

4. DHCPv6 Client Behavior

Clients will utilize the OPTION_ORO (Option Request Option), specifying the OPTION_V6_6CO option to be returned by the server, in addition to any other required configuration parameters. Because of the constrained nature of 6LoWPAN networks, clients are advised to utilize the DHCPv6 Rapid Commit [RFC3315] option when requesting DHCPv6 configuration.

5. DHCPv6 Server Behavior

Servers that support OPTION_V6_6CO are expected to be aware of the existence of constrained networks that use the server during configuration. Therefore servers SHOULD support the abbreviated "Rapid Commit" packet exchange specified in [RFC3315].

6. Security Considerations

Any type of mis-configuration of the option described in this document may cause re-routing of packets on a 6LoWPAN network, due to the compression context being blindly trusted by DHCPv6 clients requesting this option. The trust relationship necessary to create a trusted binding of compression contexts and clients on the network should be established by means other than that specified in this document. This trust relationship should be binding for all such configuration information transmitted from a DHCPv6 server to clients requesting options. DHCPv6 traffic is traditionally communicated "in the clear" on most networks, and in these scenarios where traffic is neither encrypted nor integrity protected, man-in-the-middle attacks are possible. However, in many 6LoWPAN deployment scenarios, these networks include protection at layer-2 (for example, 802.15.4 encryption), including a "secure join" mechanism that protects these networks from introducing unauthorized traffic onto the network ("rougue nodes"). In these types of networks, man-in-the-middle attacks are less likely.

7. IANA Considerations

IANA is requested to assign one option code for OPTION_V6_6CO from the "DHCP Option Codes" table of the Dynamic Host Configuration Protocol for IPv6 (DHCPv6) Registry.

8. References

[RFC6550] Winter, T., "RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks", RFC 6550, March 2012.

[RFC3315] Bound, J., "DHCP for IPv6", RFC 3315, July 2003.

[RFC6282] Hui, J. and P. Thubert, "Compression Format for IPv6 Datagrams over 802.15.4-Based Networks", RFC 6282, September 2011.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", RFC 2119, March 1997.

Author's Address

Randy Turner
Landis+Gyr
30000 Mill Creek Ave
Suite 100
Alpharetta, GA 30022
US

Phone: +1 678 258 1292
Email: randy.turner@landisgyr.com
URI: http://www.landisgyr.com/