6LoWPAN Working Group
Internet-Draft
Intended status: Standards Track
Expires: September 15, 2011

C. Bormann
Universitaet Bremen TZI

March 14, 2011

6LoWPAN Generic Compression of Headers and Header-like Payloads
draft-bormann-6lowpan-ghc-02

Abstract

This short I-D provides a complete design for a simple addition to 6LoWPAN Header Compression that enables the compression of generic headers and header-like payloads.

Status of this Memo

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

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

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

This Internet-Draft will expire on September 15, 2011.

Table of Contents

1. Introduction

1.1. The Header Compression Coupling Problem

[I-D.ietf-6lowpan-hc] defines a scheme for header compression in 6LoWPAN [RFC4944] packets. As with most header compression schemes, a new specification is needed for every new kind of header that needs to be compressed. In addition, [I-D.ietf-6lowpan-hc] does not define an extensibility scheme like the ROHC profiles defined in ROHC [RFC3095] [RFC5795]. This leads to the difficult situation that [I-D.ietf-6lowpan-hc] tends to be reopened and reexamined each time a new header receives consideration (or an old header is changed and reconsidered) in the 6lowpan/roll/core cluster of IETF working groups. At this rate, [I-D.ietf-6lowpan-hc] will never get completed (fortunately, by now it has passed WGLC, but the underlying problem remains unsolved).

The purpose of the present contribution is to plug into [I-D.ietf-6lowpan-hc] as is, using its NHC (next header compression) concept. We add a slightly less efficient, but vastly more general form of compression for headers of any kind and even for header-like payloads such as those exhibited by routing protocols, DHCP, etc. The objective is to arrive at something that can be defined on a single page and implemented in a couple of lines of code, as opposed to a general data compression scheme such as that defined in [RFC1951].

1.2. 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 RFC 2119 [RFC2119].

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

2. 6LoWPAN-GHC

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

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

The code bytes are defined as follows:

code byte	Action	Argument
0kkkkkkk	Append k = 0b0kkkkkkk bytes of data in the bytecode argument (k < 96)	k bytes of data
0110iiii	Append all bytes (possibly filling an incomplete byte with zero bits) from Context i	
0111iiii	Append 8 bytes from Context i; i.e., the context value truncated/extended to 8 bytes, and then append 0000 00FF FE00 (i.e., 14 bytes total)	
1000nnnn	Append 0b0000nnnn+2 bytes of zeroes	
10010000	STOP code (end of compressed data)	
1001nnnn	Enter nibblecode (Section 2.1)	
101nssss	sa += 0b0ssss000, na += 0b0000n000	
11nnnkkk	n = na+0b00000nnn+2; s = 0b00000kkk+sa+n; append n bytes from previously output bytes, starting s bytes to the left of the current output pointer; set sa = 0, na = 0	

For the purposes of the backreferences, the expansion buffer is initialized with the pseudo-header as defined in [RFC2460], at the end of which the target buffer begins. These pseudo-header bytes are therefore available for backreferencing, but not copied into the final result.

2.1. Nibblecode

(It is to be decided whether the mechanism described in this section is worth its additional complexity. To make this decision, it would be useful to obtain more packet captures, in particular those that do include ASCII data - the packet-capture-based examples in Section 3 currently do not include nibblecode.)

Some headers/header-like structures, such as those used in CoAP or DNS, may use ASCII data. There is very little redundancy by repetition in these (DNS actually has its own compression mechanism for repetition), so the backreferencing mechanism provided in the bytecode is not very effective.

Efficient stateless compression for small amounts of ASCII data of this kind is pretty much confined to Huffman (or, for even more complexity, arithmetic) coding. The complexity can be reduced significantly by moving to n-ary Huffman coding, i.e., optimizing not to the bit level, but to a larger level of granularity. Informal experiments by the author show that a 16ary Huffman coding is close to optimal at least for a small corpus of URI data. In other words, basing the encoding on nibbles (4-bit half-bytes) is both nearly optimal and relatively inexpensive to implement.

The actual letter frequencies that will occur in more general 6LoWPAN ASCII data are hard to predict. As a first indication, the author has analyzed an HTTP-based URI corpus and found the following lower case letters to be the ASCII characters that occur with highest frequency: aeinorst - it is therefore most useful to compress these.

In the encoding proposed, each byte representing one of these eight highly-compressed characters is represented by a single 4-bit nibble from the range 0x8 to 0xF. Bytes representing printable ASCII characters, more specifically bytes from 0x20 to 0x7F, are represented by both of their nibbles. Bytes from 0x00 to 0x1F and from 0x80 to 0xFF are represented by a 0x1 nibble followed by both nibbles of the byte. An 0x0 nibble terminates the nibblecode sequence and returns to bytecode on the next byte boundary.

The first nibble of the nibblecode is transmitted right in the "enter nibblecode" bytecode (0x9x - note that since it is never useful to immediately return to bytecode, the bytecode 0x90 is allocated for a different purpose). All other nibbles of the nibblecode are transmitted as a sequence of bytes in most-significant-nibble-first order; any unused nibble in the last byte of a nibblecode sequence is set to 0x0.

The encoding is summarized in Figure 1.

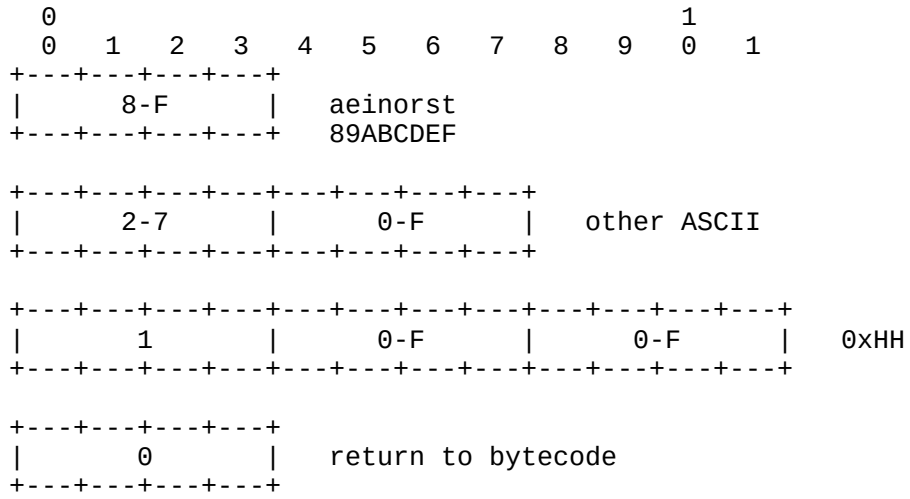


Figure 1: A nibble-based encoding

As an example for what level of compression can be expected, the 121 bytes of ASCII text shown in Figure 2 (taken from [I-D.ietf-core-link-format]) are compressed into 183 nibbles of nibblecode, which (including delimiter and padding overhead) need 93 bytes, resulting in a net compression factor of 1.30. (Note that RFC 4944/6LoWPAN-HC supports compression only in the first of a sequence of adaptation layer fragments; 93 bytes may not all fit into the first fragment, so any remaining payload would be sent without the benefit of compression.)

```
 <http://www.example.com/sensors/temp123>;anchor="/sensors/temp"
 ;rel=describedby,
 </t>;anchor="/sensors/temp";rel=alternate
```

Figure 2: Example input text (line-wrapped)

3. Examples

This section demonstrates some relatively realistic examples derived from actual PCAP dumps taken at previous interops. Unfortunately, for these dumps, no context information was available, so the relatively powerful effect of context-based compression is not shown. (TBD: Add a couple DHCP examples.)

Figure 3 shows an RPL DODAG Information Solicitation, a quite short RPL message that obviously cannot be improved much.

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

Figure 3: A simple RPL example

Figure 4 shows an RPL DODAG Information Object, a longer RPL control message that is improved a bit more (but would likely benefit additionally from a context reference). Note that the compressed output exposes an inefficiency in the simple-minded compressor used to generate it; this does not devalue the example since constrained nodes are quite likely to make use of simple-minded compressors.

```
   IP header:
    60 00 00 00 00 5c 3a ff fe 80 00 00 00 00 00 00
    02 1c da ff fe 00 30 23 ff 02 00 00 00 00 00 00
    00 00 00 00 00 00 00 1a
   Payload:
    9b 01 7a 5f 00 f0 01 00 88 00 00 00 20 02 0d b8
    00 00 00 00 00 00 00 ff fe 00 fa ce 04 0e 00 14
    09 ff 00 00 01 00 00 00 00 00 00 00 08 1e 80 20
    ff ff ff ff ff ff ff ff 00 00 00 00 20 02 0d b8
    00 00 00 00 00 00 00 ff fe 00 fa ce 03 0e 40 00
    ff ff ff ff 20 02 0d b8 00 00 00 00
   Pseudoheader:
    fe 80 00 00 00 00 00 00 02 1c da ff fe 00 30 23
    ff 02 00 00 00 00 00 00 00 00 00 00 00 00 00 1a
    00 00 00 5c 00 00 00 3a
   copy: 09 9b 01 7a 5f 00 f0 01 00 88
   3 nulls: 81
   copy: 04 20 02 0d b8
   7 nulls: 85
   ref(52): ff fe 00 -> ref 101nssss 0 6/11nnnkkk 1 1: a6 c9
   copy: 08 fa ce 04 0e 00 14 09 ff
   2 nulls: 80
   copy: 01 01
   7 nulls: 85
   copy: 06 08 1e 80 20 ff ff
   ref(2): ff ff -> ref 11nnnkkk 0 0: c0
   ref(4): ff ff ff ff -> ref 11nnnkkk 2 0: d0
   4 nulls: 82
   ref(48): 20 02 0d b8 00 00 00 00 00 00 00 ff fe 00 fa ce
    -> ref 101nssss 1 4/11nnnkkk 6 0: b4 f0
   copy: 03 03 0e 40
   ref(9): 00 ff -> ref 11nnnkkk 0 7: c7
   ref(28): ff ff ff -> ref 101nssss 0 3/11nnnkkk 1 1: a3 c9
   ref(24): 20 02 0d b8 00 00 00 00
    -> ref 101nssss 0 2/11nnnkkk 6 0: a2 f0
   Compressed:
    09 9b 01 7a 5f 00 f0 01 00 88 81 04 20 02 0d b8
    85 a6 c9 08 fa ce 04 0e 00 14 09 ff 80 01 01 85
    06 08 1e 80 20 ff ff c0 d0 82 b4 f0 03 03 0e 40
    c7 a3 c9 a2 f0
   Was 92 bytes; compressed to 53 bytes, compression factor 1.74
```

Figure 4: A longer RPL example

Similarly, Figure 5 shows an RPL DAO message. One of the embedded addresses is copied right out of the pseudoheader, the other one is effectively converted from global to local by providing the prefix FE80 literally, inserting a number of nulls, and copying (some of) the IID part again out of the pseudoheader. Note that a simple implementation would probably emit fewer nulls and copy the entire IID; there are multiple ways to encode this 50-byte payload into 27 bytes.

```
IP header:
 60 00 00 00 00 32 3a ff 20 02 0d b8 00 00 00 00
 00 00 00 ff fe 00 33 44 20 02 0d b8 00 00 00 00
 00 00 00 ff fe 00 11 22
Payload:
 9b 02 58 7d 01 80 00 f1 05 12 00 80 20 02 0d b8
 00 00 00 00 00 00 00 ff fe 00 33 44 06 14 00 80
 f1 00 fe 80 00 00 00 00 00 00 00 00 00 ff fe 00
 11 22
Pseudoheader:
 20 02 0d b8 00 00 00 00 00 00 00 ff fe 00 33 44
 20 02 0d b8 00 00 00 00 00 00 00 ff fe 00 11 22
 00 00 00 32 00 00 00 3a
copy: 0c 9b 02 58 7d 01 80 00 f1 05 12 00 80
ref(52): 20 02 0d b8 00 00 00 00 00 00 00 ff fe 00 33 44
 -> ref 101nssss 1 4/11nnnkkk 6 4: b4 f4
copy: 08 06 14 00 80 f1 00 fe 80
9 nulls: 87
ref(58): ff fe 00 11 22 -> ref 101nssss 0 6/11nnnkkk 3 5: a6 dd
Compressed:
 0c 9b 02 58 7d 01 80 00 f1 05 12 00 80 b4 f4 08
 06 14 00 80 f1 00 fe 80 87 a6 dd
Was 50 bytes; compressed to 27 bytes, compression factor 1.85
```

Figure 5: An RPL DAO message

Figure 6 shows the effect of compressing a simple ND neighbor solicitation (again, no context-based compression).

```
IP header:
 60 00 00 00 00 30 3a ff 20 02 0d b8 00 00 00 00
 00 00 00 ff fe 00 3b d3 fe 80 00 00 00 00 00 00
 02 1c da ff fe 00 30 23
Payload:
 87 00 a7 68 00 00 00 00 fe 80 00 00 00 00 00 00
 02 1c da ff fe 00 30 23 01 01 3b d3 00 00 00 00
 1f 02 00 00 00 00 00 06 00 1c da ff fe 00 20 24
Pseudoheader:
 20 02 0d b8 00 00 00 00 00 00 00 ff fe 00 3b d3
 fe 80 00 00 00 00 00 00 02 1c da ff fe 00 30 23
 00 00 00 30 00 00 00 3a
copy: 04 87 00 a7 68
4 nulls: 82
ref(32): fe 80 00 00 00 00 00 00 02 1c da ff fe 00 30 23
 -> ref 101nssss 1 2/11nnnkkk 6 0: b2 f0
copy: 04 01 01 3b d3
4 nulls: 82
copy: 02 1f 02
5 nulls: 83
copy: 02 06 00
ref(24): 1c da ff fe 00 -> ref 101nssss 0 2/11nnnkkk 3 3: a2 db
copy: 02 20 24
Compressed:
 04 87 00 a7 68 82 b2 f0 04 01 01 3b d3 82 02 1f
 02 83 02 06 00 a2 db 02 20 24
Was 48 bytes; compressed to 26 bytes, compression factor 1.85
```

Figure 6: An ND neighbor solicitation

Figure 7 shows the compression of an ND neighbor advertisement.

```
IP header:
 60 00 00 00 00 30 3a fe fe 80 00 00 00 00 00 00
 02 1c da ff fe 00 30 23 20 02 0d b8 00 00 00 00
 00 00 00 ff fe 00 3b d3
Payload:
 88 00 26 6c c0 00 00 00 fe 80 00 00 00 00 00 00
 02 1c da ff fe 00 30 23 02 01 fa ce 00 00 00 00
 1f 02 00 00 00 00 00 06 00 1c da ff fe 00 20 24
Pseudoheader:
 fe 80 00 00 00 00 00 00 02 1c da ff fe 00 30 23
 20 02 0d b8 00 00 00 00 00 00 00 ff fe 00 3b d3
 00 00 00 30 00 00 00 3a
copy: 05 88 00 26 6c c0
3 nulls: 81
ref(48): fe 80 00 00 00 00 00 00 02 1c da ff fe 00 30 23
 -> ref 101nssss 1 4/11nnnkkk 6 0: b4 f0
copy: 04 02 01 fa ce
4 nulls: 82
copy: 02 1f 02
5 nulls: 83
copy: 02 06 00
ref(24): 1c da ff fe 00 -> ref 101nssss 0 2/11nnnkkk 3 3: a2 db
copy: 02 20 24
Compressed:
 05 88 00 26 6c c0 81 b4 f0 04 02 01 fa ce 82 02
 1f 02 83 02 06 00 a2 db 02 20 24
Was 48 bytes; compressed to 27 bytes, compression factor 1.78
```

Figure 7: An ND neighbor advertisement

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

```
IP header:
 60 00 00 00 00 18 3a ff fe 80 00 00 00 00 00 00
 ae de 48 00 00 00 00 01 ff 02 00 00 00 00 00 00
 00 00 00 00 00 00 00 02
Payload:
 85 00 90 65 00 00 00 00 01 02 ac de 48 00 00 00
 00 01 00 00 00 00 00 00
Pseudoheader:
 fe 80 00 00 00 00 00 00 ae de 48 00 00 00 00 01
 ff 02 00 00 00 00 00 00 00 00 00 00 00 00 00 02
 00 00 00 18 00 00 00 3a
copy: 04 85 00 90 65
ref(33): 00 00 00 00 01 -> ref 101nssss 0 3/11nnnkkk 3 4: a3 dc
copy: 02 02 ac
ref(42): de 48 00 00 00 00 01
 -> ref 101nssss 0 4/11nnnkkk 5 3: a4 eb
6 nulls: 84
Compressed:
 04 85 00 90 65 a3 dc 02 02 ac a4 eb 84
Was 24 bytes; compressed to 13 bytes, compression factor 1.85
```

Figure 8

Figure 9 shows the compression of an ND router advertisement. The indefinite lifetime is compressed to four bytes by backreferencing; this could be improved (at the cost of minor additional decompressor complexity) by including some simple runlength mechanism.

```
   IP header:
    60 00 00 00 00 60 3a ff fe 80 00 00 00 00 00 00
    10 34 00 ff fe 00 11 22 fe 80 00 00 00 00 00 00
    ae de 48 00 00 00 00 01
   Payload:
    86 00 55 c9 40 00 0f a0 1c 5a 38 17 00 00 07 d0
    01 01 11 22 00 00 00 00 03 04 40 40 ff ff ff ff
    ff ff ff ff 00 00 00 00 20 02 0d b8 00 00 00 00
    00 00 00 00 00 00 00 00 20 02 40 10 00 00 03 e8
    20 02 0d b8 00 00 00 00 21 03 00 01 00 00 00 00
    20 02 0d b8 00 00 00 00 00 00 00 ff fe 00 11 22
   Pseudoheader:
    fe 80 00 00 00 00 00 00 10 34 00 ff fe 00 11 22
    fe 80 00 00 00 00 00 00 ae de 48 00 00 00 00 01
    00 00 00 60 00 00 00 3a
   copy: 0c 86 00 55 c9 40 00 0f a0 1c 5a 38 17
   2 nulls: 80
   copy: 06 07 d0 01 01 11 22
   4 nulls: 82
   copy: 06 03 04 40 40 ff ff
   ref(2): ff ff -> ref 11nnnkkk 0 0: c0
   ref(4): ff ff ff ff -> ref 11nnnkkk 2 0: d0
   4 nulls: 82
   copy: 04 20 02 0d b8
   12 nulls: 8a
   copy: 04 20 02 40 10
   ref(38): 00 00 03 -> ref 101nssss 0 4/11nnnkkk 1 3: a4 cb
   copy: 01 e8
   ref(24): 20 02 0d b8 00 00 00 00
    -> ref 101nssss 0 2/11nnnkkk 6 0: a2 f0
   copy: 02 21 03
   ref(84): 00 01 00 00 00 -> ref 101nssss 0 9/11nnnkkk 3 7: a9 df
   ref(40): 00 20 02 0d b8 00 00 00 00 00 00 00
    -> ref 101nssss 1 3/11nnnkkk 2 4: b3 d4
   ref(120): ff fe 00 11 22
    -> ref 101nssss 0 14/11nnnkkk 3 3: ae db
   Compressed:
    0c 86 00 55 c9 40 00 0f a0 1c 5a 38 17 80 06 07
    d0 01 01 11 22 82 06 03 04 40 40 ff ff c0 d0 82
    04 20 02 0d b8 8a 04 20 02 40 10 a4 cb 01 e8 a2
    f0 02 21 03 a9 df b3 d4 ae db
   Was 96 bytes; compressed to 58 bytes, compression factor 1.66
```

Figure 9: An ND router advertisement

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

6LoWPAN-GHC is intended to plug in as an NHC format for 6LoWPAN-HC [I-D.ietf-6lowpan-hc]. This section shows how this can be done (without supplying the detailed normative text yet, although it could be implemented from this page).

GHC is by definition generic and can be applied to different kinds of packets. All the examples given above are for ICMPv6 packets; it is trivial to define an NHC format for ICMPv6 based on GHC.

In addition it may be useful to include an NHC format for UDP, as many headerlike payloads (e.g., DHCPv6) are carried in UDP. [I-D.ietf-6lowpan-hc] already defines an NHC format for UDP (11110CPP). What remains to be done is to define an analogous NHC byte formatted, e.g. as shown in Figure 10, and simply reference the existing specification, indicating that for 0b11010cpp NHC bytes, the UDP payload is not supplied literally but compressed by 6LoWPAN-GHC.

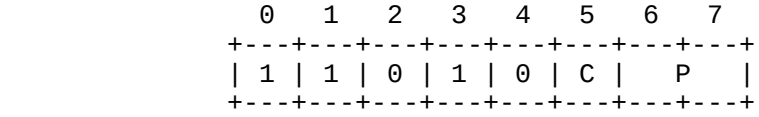


Figure 10: A possible NHC byte for UDP GHC

To stay in the same general numbering space, we propose 0b11011111 as the NHC byte for ICMPv6 GHC.

4.1. Compressing extension headers

If the compression of specific extension headers is considered desirable, this can be added in a similar way, e.g. as in Figure 11 (however, probably only EID 0 to 3 need to be assigned). As there is no easy way to extract the length field from the GHC-encoded header before decoding, this would make detecting the end of the extension header somewhat complex. The easiest (and most efficient) approach is to completely elide the length field (in the same way NHC already elides the next header field in certain cases) and reconstruct it only on decompression. Instead, the reserved bytecode 0b10010000 would be assigned as a stop marker.

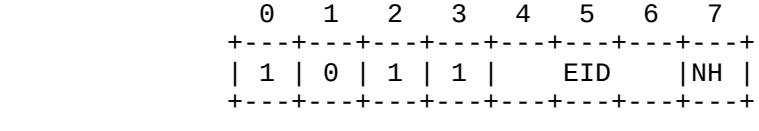


Figure 11: A possible NHC byte for extension header GHC

4.2. Indicating GHC capability

The 6LoWPAN baseline includes just [RFC4944], [I-D.ietf-6lowpan-hc], [I-D.ietf-6lowpan-nd] (see [I-D.bormann-6lowpan-roadmap]). To enable the use of GHC, 6LoWPAN nodes need to know that their neighbors implement it. While this can simply be administratively required, a transition strategy as well as a way to support mixed networks is required.

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

To minimize the impact on [I-D.ietf-6lowpan-nd], we propose adding an ND option 6LoWPAN Capability Indication (6CIO), as illustrated in Figure 12.

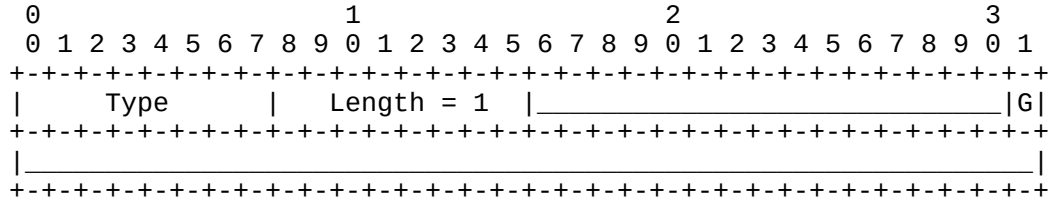


Figure 12: 6LoWPAN Capability Indication (6CIO)

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

The 6CIO option will typically only be sent in 6LoWPAN-ND RS packets; the resulting 6LoWPAN-ND RA can already make use of GHC and thus indicate GHC capability implicitly, which in turn allows the nodes to use GHC in the 6LoWPAN-ND NS/NA exchange.

6CIO can also be used for future options that need to be negotiated between 6LoWPAN peers; an IANA registry will administrate the flags. (Bits marked by underscores in Figure 12 are reserved for future allocation, i.e., they MUST be sent as zero and MUST be ignored on reception until allocated. Length values larger than 1 MUST be supported for future extensions; the additional bits in the option are then reserved in the same way. For the purposes of the IANA registry, the bits are numbered in msb-first order from the 16th bit of the option onwards, i.e., the G bit is flag number 15.)

5. IANA considerations

In the IANA registry for the 6LOWPAN_NHC header type, IANA would need to add the assignments in Figure 13.

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

Figure 13: IANA assignments for the NHC byte

*) if the functionality of Section 4.1 is made part of this document.

An IANA registry is needed for 6LoWPAN capability flags. (Policy TBD.)

IANA needs to allocate an ND option number for 6CIO.

6. Security considerations

The security considerations of [RFC4944] and [I-D.ietf-6lowpan-hc] apply. As usual in protocols with packet parsing/construction, care must be taken in implementations to avoid buffer overflows and in particular (with respect to the back-referencing) out-of-area references during decompression.

One additional consideration is that an attacker may send a forged packet that makes a second node believe a third victim node is GHC-capable. If it is not, this may prevent packets sent by the second node from reaching the third node.

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

7. Acknowledgements

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

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

8. References

8.1. Normative References

[I-D.ietf-6lowpan-hc]
Hui, J. and P. Thubert, "Compression Format for IPv6 Datagrams in Low Power and Lossy Networks (6LoWPAN)", draft-ietf-6lowpan-hc-15 (work in progress), February 2011.

[I-D.ietf-6lowpan-nd]
Shelby, Z., Chakrabarti, S., and E. Nordmark, "Neighbor Discovery Optimization for Low-power and Lossy Networks", draft-ietf-6lowpan-nd-15 (work in progress), December 2010.

[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.

8.2. Informative References

[I-D.bormann-6lowpan-roadmap]
Bormann, C., "6LoWPAN Roadmap and Implementation Guide", draft-bormann-6lowpan-roadmap-00 (work in progress), March 2011.

[I-D.ietf-core-link-format]
Shelby, Z., "CoRE Link Format", draft-ietf-core-link-format-02 (work in progress), December 2010.

[I-D.oflynn-6lowpan-icmphc]
O'Flynn, C., "ICMPv6/ND Compression for 6LoWPAN Networks", draft-oflynn-6lowpan-icmphc-00 (work in progress), July 2010.

[RFC1951] Deutsch, P., "DEFLATE Compressed Data Format Specification version 1.3", RFC 1951, May 1996.

[RFC3095] Bormann, C., Burmeister, C., Degermark, M., Fukushima, H., Hannu, H., Jonsson, L-E., Hakenberg, R., Koren, T., Le,

K., Liu, Z., Martensson, A., Miyazaki, A., Svanbro, K., Wiebke, T., Yoshimura, T., and H. Zheng, "RObust Header Compression (ROHC): Framework and four profiles: RTP, UDP, ESP, and uncompressed", RFC 3095, July 2001.

[RFC5795] Sandlund, K., Pelletier, G., and L-E. Jonsson, "The RObust Header Compression (ROHC) Framework", RFC 5795, March 2010.

Author's Address

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

Phone: +49-421-218-63921
Fax: +49-421-218-7000
Email: cabo@tzi.org

6LoWPAN Working Group
Internet-Draft
Intended status: Informational
Expires: September 8, 2011

C. Bormann
Universitaet Bremen TZI

March 7, 2011

6LoWPAN Roadmap and Implementation Guide
draft-bormann-6lowpan-roadmap-00

Abstract

6LoWPAN is defined in RFC 4944 in conjunction with a number of specifications that are currently nearing completion. The entirety of these specifications may be hard to understand, pose specific implementation problems, or be simply inconsistent.

The present guide aims to provide a roadmap to these documents as well as provide specific advice how to use these specifications in combination. In certain cases, it may provide clarifications or even corrections to the specifications referenced.

This guide is intended as a continued work-in-progress, i.e. a long-lived Internet-Draft, to be updated whenever new information becomes available and new consensus on how to handle issues is formed. Similar to the ROHC implementation guide, RFC 4815, it might be published as an RFC at some future time later in the acceptance curve of the specifications.

This document does not describe a new protocol or attempts to set a new standard of any kind -- it mostly describes good practice in using the existing specifications, but it may also document emerging consensus where a correction needs to be made.

The current version -00 of this document is just an initial draft that is intended to spark the collection of relevant information.

time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on September 8, 2011.

Table of Contents

1. Introduction

(To be written - for now please read the Abstract.)

1.1. Terminology

This document is a guide. However, it might evolve to make specific recommendations on how to use standards-track specification. Therefore: 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. They indicate requirement levels for compliant 6LoWPAN implementations [RFC2119]. Note that these keywords are not only used where a correction or clarification is intended; the latter are explicitly identified as such.

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

2. 6LoWPAN

What is a 6LoWPAN?

The term, originally just the name of the IETF WG that created the specifications, nowadays refers to a specific way of building IP-connected wireless networks for embedded use cases. The 6LoWPAN core specifications are:

- [RFC4944], as updated by
- [I-D.ietf-6lowpan-hc] and
- [I-D.ietf-6lowpan-nd].

While [RFC4944] defines 6LoWPAN specifically for IEEE 802.15.4 networks, 6LoWPAN concepts have been applied to other PHY/MAC layers.

6LoWPANs MAY use additional protocols, such as [I-D.ietf-roll-rpl] for routing, or [I-D.ietf-core-coap] for application data transfer. However, the "6LoWPANness" of a network is caused by adherence to the core specifications.

3. 6LoWPAN MTU

IPv6 defines a minimal value for the "Minimum Transmission Unit", MTU, of 1280 bytes. This means that every IPv6 network must be able to transfer a packet of at least 1280 bytes of IPv6 headers and data without requiring fragmentation.

A common Internet MTU is 1500 bytes (motivated by the Ethernet MTU). The gap between 1280 and 1500 allows tunneling protocols to insert headers on the way from the source of a packet to a destination without breaking the overall MTU of the path. As various tunneling protocols do indeed insert bytes, it is unwise to simply assume an end-to-end MTU of 1500 bytes even with the current dominance of Ethernet. Path MTU discovery [RFC1981] [RFC4821] has been defined to enable transport protocols to find an MTU value better than 1280 bytes, but still reliably within the MTU of the path being used. Path MTU discovery places, however, additional strain on constrained nodes, which therefore may want to stick with an MTU of 1280 bytes for all IPv6 applications.

6LoWPAN was designed as a stub network, not requiring any tunneling. As IEEE 802.15.4 packets are rather small (127 bytes maximum at the physical layer, minus MAC/security and adaptation layer overhead), 1280 bytes was already considered a somewhat large packet size. Therefore, the 6LoWPAN network MTU was simply set at the minimum size allowable by IPv6, 1280 bytes, although the 6LoWPAN fragmentation mechanism is able to support packets with total lengths (including the initial IPv6 header) of up to 2047 bytes.

As a more recent development, some modes of operation of the RPL protocol [I-D.ietf-roll-rpl] do indeed operate by tunneling data packets between RPL routers. Maintaining the MTU visible to applications at 1280 therefore requires making a larger MTU available to the tunnels.

6LoWPAN routers that employ RPL therefore MUST support a more appropriate MTU between routers that make use of tunneling between them. [The specific MTU value is TBD, to be chosen between 1280 and 2047 based on RPL considerations that need to be added to this document.]

4. PAN identifiers in IPv6 addresses

[RFC4944] incorporates PAN identifiers in IPv6 addresses created from 16-bit MAC addresses, in a somewhat awkward way (one of the 16 bits needs to be cleared to enable the U/L bit.).

As the use of PAN identifiers in 6LoWPAN networks has since become less and less meaningful, [I-D.ietf-6lowpan-hc] provides specific support only for interface IDs of the form 0000:00ff:fe00:XXXX, i.e. PAN identifiers of zero. (Other forms can be supported by creating sufficiently long pieces of compression context information for each non-zero PAN identifier; however there is a limited number of context elements and each consumes space in all nodes of a 6LoWPAN.)

It is therefore RECOMMENDED to employ a PAN identifier of zero with 6LoWPAN.

5. IANA Considerations

This document has no actions for IANA.

6. Security Considerations

(None so far; this section will certainly grow as additional security considerations beyond those listed in the base specifications become known.)

7. Acknowledgements

(The concept for this document is borrowed from [RFC4815], which was invented by Lars-Erik Jonsson. Thanks!)

8. References

8.1. Normative References

[I-D.ietf-6lowpan-hc]
Hui, J. and P. Thubert, "Compression Format for IPv6 Datagrams in Low Power and Lossy Networks (6LoWPAN)", draft-ietf-6lowpan-hc-15 (work in progress), February 2011.

[I-D.ietf-6lowpan-nd]
Shelby, Z., Chakrabarti, S., and E. Nordmark, "Neighbor Discovery Optimization for Low-power and Lossy Networks", draft-ietf-6lowpan-nd-15 (work in progress), December 2010.

[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.

8.2. Informative References

[I-D.ietf-core-coap]
Shelby, Z., Hartke, K., Bormann, C., and B. Frank, "Constrained Application Protocol (CoAP)", draft-ietf-core-coap-04 (work in progress), January 2011.

[I-D.ietf-roll-rpl]
Winter, T., Thubert, P., Brandt, A., Clausen, T., Hui, J., Kelsey, R., Levis, P., Pister, K., Struik, R., and J. Vasseur, "RPL: IPv6 Routing Protocol for Low power and Lossy Networks", draft-ietf-roll-rpl-18 (work in progress), February 2011.

[RFC1981] McCann, J., Deering, S., and J. Mogul, "Path MTU Discovery for IP version 6", RFC 1981, August 1996.

[RFC4815] Jonsson, L-E., Sandlund, K., Pelletier, G., and P. Kremer, "RObust Header Compression (ROHC): Corrections and Clarifications to RFC 3095", RFC 4815, February 2007.

[RFC4821] Mathis, M. and J. Heffner, "Packetization Layer Path MTU Discovery", RFC 4821, March 2007.

Author's Address

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

Phone: +49-421-218-63921
Fax: +49-421-218-7000
Email: cabo@tzi.org

6Lowpan Working Group
Internet-Draft
Intended status: Informational
Expires: September 7, 2011

Z. Cao
China Mobile

March 6, 2011

Considerations for Lightweight IP Gateways
draft-cao-lwig-gateway-00

Abstract

This document discusses several considerations of the gateway that connects the IPv6 smart devices with the non-ready IPv6 Internet.

Status of this Memo

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

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

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

This Internet-Draft will expire on September 7, 2011.

Copyright Notice

Table of Contents

1. Introduction

The ultimately goal of enabling IP stack on small devices is to connect them to the global Internet. Many efforts are dedicated to compressing IPv6 header for smart devices [RFC4944] [I-D.ietf-6lowpan-hc] so that the smart objects network is IPv6 ready. However the connection from the gateway to the outside network is still evolving to the IPv6; many parts of the network is still IPv4, especially for home users. And many Internet application servers are not IPv6 ready. The IPv6 smart device could not connect to the IPv4 service platform without intermediate boxes.

In this situation, it is important to discuss how to connect the IPv6 ready smart objects network to the non IPv6 ready global Internet. This document introduces several identified problems and some considerations on solutions to these problems.

1.1. Conventions used in this document

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

2. Network Architecture and Scenarios

Figure 1 depicts the secenario of the interconnection between the smart objects network and the Internet. Several important components within this architecture is analyzed below:

1. Node: the smart device. In current IETF efforts, the Node is IPv6 ready and IPv6 only. Numbers of the Nodes constitute the smart objects network, which is IPv6 ready.
2. SNG: Smart Network Gateway. The SNG interfaces with the smart objects network and the operators’s access network. The SNG should support the wireless technolgies connecting with the Node and the lightweight IPv6 implementation as well. The upper connection from the SNG to the access network depends on the capability provided by the operator.
3. ONG: Operator Network Gateway. The ONG may not be visible to users. It is used to apply charging, security and QoS policies. In certain scenarios, the ONG is used to manage an IPv6-in-IPv4 tunnel between the SNG and itself.
4. Server: the applicatoin server. The server may be IPv4 or IPv6, or dual-stack. It collects information from the smart network and share/push these information to users Internet wide.

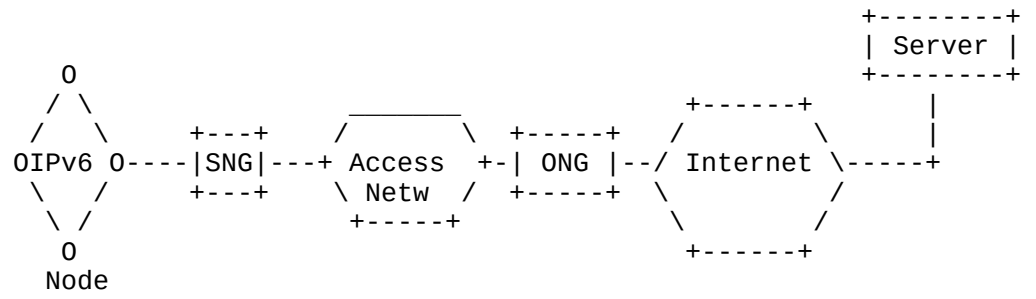


Figure 1: Smart Network Connecting the Internet

3. Solution Considerations

When both the access network and the application servers are IPv6 enabled, the solution to this problem is trivial as far as we can see. The communication between the Node and the Server is end-to-end.

When the server is not IPv6 ready or part of the network is not IPv6 ready, several solutions need discussion at the current point. This section discusses several considerations on the solutions.

3.1. Aggregated Smart Network Gateways

In this sense, the connection between the Node and Server is not end to end. Rather, the SNG aggregates the information collected from the smart devices and sends the aggregated message to the service platform. As long as the SNG is enabled with Server the same IP family, the rest of the work is trivial.

Most existing applications follow this non end-to-end architecture. But in this architecture, the SNG should be implemented with service logical and its scalability is challenged.

3.2. Tunneling IPv6

When the server is IPv6 ready but part of the network is not IPv6 ready, tunneling the IPv6 within the IPv4 packets is a direct solution.

For example as shown in Figure 2, the access network is IPv4 only and the Internet and Server is dual stack. The SNG and ONG should establish an IPv6-in-IPv4 tunnel. The SNG encapsulates the IPv6 packets within the IPv4 header to the ONG and ONG de-capsulates the IPv6 packet and sends to the service platform. Softwire tunnels

[I-D.ietf-softwire-dual-stack-lite] may be used in this scenario.

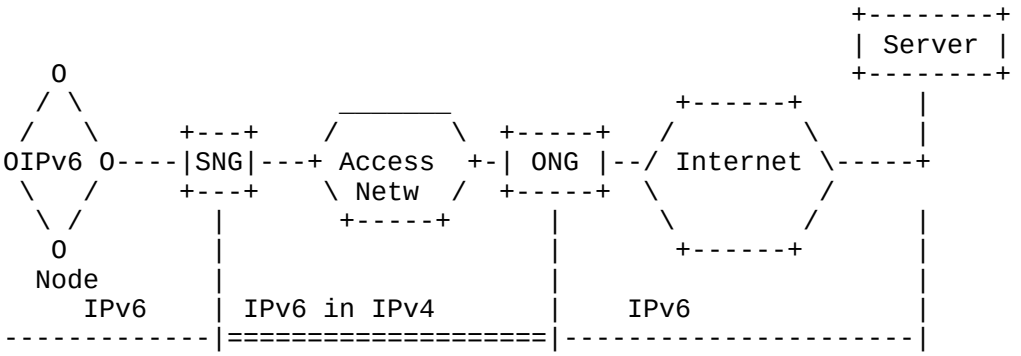


Figure 2: Tunneling Solution

3.3. IP Family Translation

If the service platform is IPv4 only, the need of IPv6 to IPv4 translation is indispensable.

In Figure 3, the SNG does not translate the IPv6 directly. Rather, SNG tunnels the IPv6 packets to the ONG within the IPv4, and the ONG decapsulates and translates the IPv6 to IPv4, using stateless or stateful translation [I-D.ietf-behave-v6v4-xlate-stateful] [I-D.ietf-behave-v6v4-xlate].

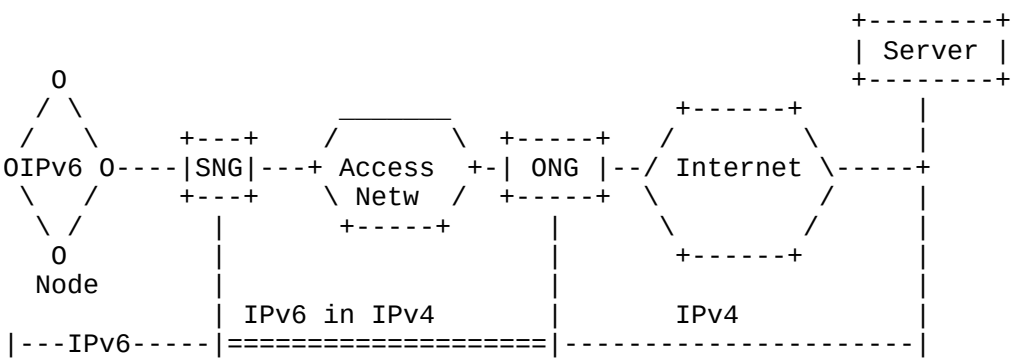


Figure 3: Translation on ONG

In Figure 4, different from the above scenario, the SNG translates the IPv6 to IPv4 directly, using stateless or stateful translation [I-D.ietf-behave-v6v4-xlate-stateful] [I-D.ietf-behave-v6v4-xlate].

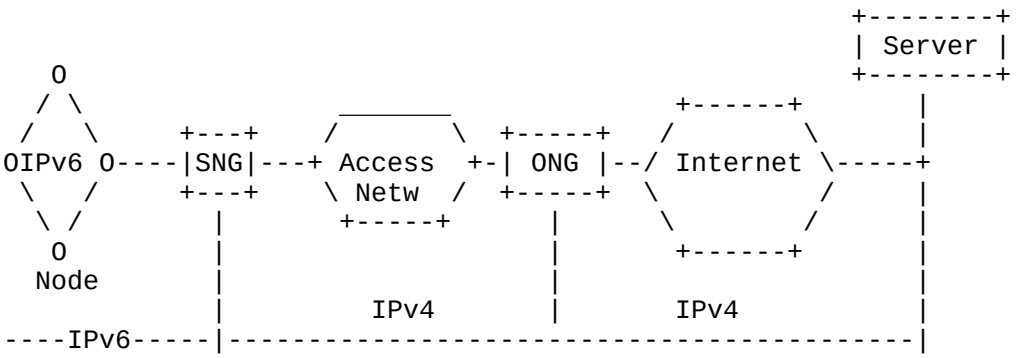


Figure 4: Translation on SNG

4. Security Considerations

TBD.

5. IANA Considerations

This document does not require any IANA actions.

6. Normative References

[I-D.ietf-6lowpan-hc]
Hui, J. and P. Thubert, "Compression Format for IPv6 Datagrams in Low Power and Lossy Networks (6LoWPAN)", draft-ietf-6lowpan-hc-15 (work in progress), February 2011.

[I-D.ietf-behave-v6v4-xlate]
Li, X., Bao, C., and F. Baker, "IP/ICMP Translation Algorithm", draft-ietf-behave-v6v4-xlate-23 (work in progress), September 2010.

[I-D.ietf-behave-v6v4-xlate-stateful]
Bagnulo, M., Matthews, P., and I. Beijnum, "Stateful NAT64: Network Address and Protocol Translation from IPv6 Clients to IPv4 Servers", draft-ietf-behave-v6v4-xlate-stateful-12 (work in progress), July 2010.

[I-D.ietf-softwire-dual-stack-lite]
Durand, A., Droms, R., Woodyatt, J., and Y. Lee, "Dual-

Stack Lite Broadband Deployments Following IPv4 Exhaustion", draft-ietf-softwire-dual-stack-lite-07 (work in progress), March 2011.

[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.

Author's Address

Zhen Cao
China Mobile
Unit2, 28 Xuanwumenxi Ave,Xuanwu District
Beijing 100053
China

Email: zehn.cao@gmail.com

Internet Engineering Task Force
Internet-Draft
Intended status: Informational
Expires: September 9, 2011

A. Cardenas
Fujitsu Laboratories
S. Cespedes
University of Waterloo
T. Iwao
Fujitsu Limited

March 8, 2011

Depth-First Forwarding in Unreliable Networks
draft-cardenas-dff-00

Abstract

Routing protocols are generally composed of two independent phases, the control plane and the data forwarding plane. The control plane is responsible for route discovery and maintenance. The data forwarding plane performs a table lookup operation to set the packet on the right path. In unreliable networks, the routing process incurs a large control overhead when is constantly repairing routes, detecting loops, and finding alternate paths due to frequent link failures.

This document describes the Depth-First Forwarding (DFF) protocol; a data forwarding mechanism that can be used to minimize the burden and control overhead of a control plane used in unreliable networks. DFF offers reliability and low control overhead by supporting in the data plane loop detection, updates to the routing tables, and rerouting of data packets through alternate paths. DFF can be integrated with different types of control plane mechanisms and can be used in mesh-under and route-over specifications. In this draft, we describe a sample DFF implementation as a 6LoWPAN mesh-under data forwarding protocol.

Status of this Memo

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

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

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

This Internet-Draft will expire on September 9, 2011.

Table of Contents

1. Introduction

Networks with dynamic links present a challenge for typical routing protocols because the reliability of links may be different at the time when the route was discovered, and at the time when data is forwarded.

In these unreliable networks, the control overhead for detecting routing errors and for fixing paths happens often, so it is important to avoid expensive control plane mechanisms that might overreact in the presense of instability. Because a lightweight control plane mechanism cannot guarantee the construction and maintenance of error-free routes, a data forwarding protocol designed for these conditions should be able to detect errors and find backup paths to survive link failures.

This document describes Depth-First Forwarding (DFF), a data forwarding mechanism that can detect loops, update the routing tables, and reroute data packets via alternate paths. DFF is compatible with light-weight control plane mechanisms supporting routing tables that maintain more than one possible next hop for each final destination.

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. Terminology

Readers are expected to be familiar with all the terms and concepts that are discussed in "Transmission of IPv6 Packets over IEEE 802.15.4 Networks" [RFC4944].

Other terms used:

Final Destination: This is the final destination of the data packet within the mesh network.

Local destination: The local destination of the data packet refers to the next-hop neighbor to which the packet is forwarded on its way to the final destination.

Originator: This is the source node that created the 6lowpan data packet.

2. Protocol Overview

DFF is a data forwarding strategy responsible for loop detection, choosing alternate next hops, and updating the cost metrics in the routing tables to reflect information gathered by forwarding data packets.

DFF is intended to work in a network where nodes maintain proactively a routing table with multiple candidate next hops for each final destination. An example of a control plane satisfying these conditions is described in the Appendix.

DFF provides an advantage in networks where the reliability of links changes rapidly. It assumes that the control plane mechanism cannot guarantee up to date routing tables, nor the absence of loops. Therefore, whenever a data packet is forwarded, DFF can keep a data packet identifier to detect loops, update routing tables if a loop is detected, and use alternate paths to reroute the packet around the failed path.

DFF achieves this functionality by implementing a distributed depth-first search over the network graph as defined by the routing table. If the routing tables are up to date, the search will only involve the default route. However, if the routing table is not up to date and forwarding of a data packet results in a loop, or if the link layer fails to successfully transmit the packet to the next hop, the data packet is then sent to an alternate next hop neighbor. A distributed depth-first search mechanism is implemented in order to keep track of the nodes that have participated in the forwarding of the data packet.

Although DFF can be used without a control plane by performing a blind (i.e., without a routing table) depth-first search of the network, this configuration will incurr in increased latency because data packets are forwarded by intermediate nodes to a random next-hop neighbor. Therefore, it is recommended to implement DFF in combination with a proactive control plane protocol, in order to efficiently guide the depth-first search by using information stored in the routing tables.

3. Hop-by-Hop Implementation Options

While DFF can be used in a route-over or mesh-under protocol, this document provides a sample implementation of a mesh-under forwarding solution for 6LoWPAN networks; therefore, all addresses referenced in this document are either 16-bit short or EUI-64 link layer addresses.

DFF requires the use of hop-by-hop options, and this document describes how these hop-by-hop options can be implemented by allocating a new dispatch byte from the reserved values for mesh forwarding in [RFC4944].

To avoid the request of a new dispatch byte, the appendix describes a way to implement DFF by overloading the fragmentation header in [RFC4944]. This implementation has the advantage of using headers already defined by the standard; however, the implementation by overloading the fragmentation header only allows rerouting a packet on loop detection and not when a link fails. The reason behind this loss of functionality is that rerouting when a link fails requires the use of a duplicate flag in the header of the packet. This is a hop-by-hop option that can be dynamically updated by intermediate nodes, and the fragmentation header of [RFC4944] cannot be changed by intermediate nodes.

Similarly, a route over implementation of DFF would need to obtain new fields in the hop-by-hop options of IPv6 packets.

4. Depth-First Forwarding Operation

The operation procedure described in this section relies on the existence of a routing table in every node. This table SHOULD be filled by a proactive control plane that stores multiple candidate next hops for final destinations. An example of a proactive distance vector control plane that could be integrated to DFF is provided in the Appendix.

In order for an Originator to send packets based on depth-first forwarding, it encapsulates the data packet using the standard mesh header defined in [RFC4944] and the DFF mesh header (Figure 1). The DFF mesh header is employed to detect loops and reroute packets in the forwarding path. The Originator then checks the routing table to select the next hop with the lowest cost to reach the destination. Before forwarding the packet, an entry is created in the loop detection table (Figure 2), where information such as the Originator’s address, data packet identifier, previous hop (it points to the node itself when the node is the Originator of packet), and the selected next hop are stored.

Upon reception of a data packet at an intermediate node (which might be the Originator if there is a loop in the path), the node checks if an entry with the same (Originator,DID) exists in the loop detection table. If there is no such entry, intermediate nodes SHOULD create a new entry in a similar way to that described for the originator of the data packet; however, in the Previous hop field, they store the

address of the router from which the packet was received. After the entry has been created in the loop detection table, the node forwards the packet to the selected candidate next hop.

For those cases in which an entry already exists in the loop detection table, the node checks which one was the last attemped node, and poisons the routing table entry that uses that particular node to reach the destination. By poisoning failed paths, DFF updates the routing table based on the results from the data plane. Then, in order to reroute the data packet, the node selects a new next hop among the list of candidates stored in the routing table. The selected node MUST not be registered as a previous attempt in the list of attempted neighbours in the loop detection table. I also MUST be a different node from that registered in the Previous Hop field. In this way, DFF effectively makes data forwarding with loops a depth-first search guided by the routing table stored in each node.

If the node has attempted all the candidate next hops, then the packet is returned to Previous Hop. If the Previous hop address is the same address of the current node, that means the node is the originator of the packet. If in addition, the node has already attempted all next-hop options, this means that routing has failed; therefore, the originator must drop the packet and delete the entry in the loop table.

In addition to rerouting packets when a loop is detected, nodes reroute packets when the link layer fails to receive ACK from the neighbor they sent the data packet to. As soon as the link layer gives up on the transmission, DFF proceeds to reroute the packet through a different candidate. In this case, nodes set a duplicate detection flag in the DFF mesh header, identifying whether or not the packet is a potential duplicate. Duplicate packets can appear in the network when the link layer reports a failed transmission due to a failed reception of ACKs from the recipient of the packet. This situation may appear on links that are lossy only in one direction.

Duplicate packets do not alter the depth-first search logic: if a packet with a duplicate flag is received by a node who has already sent a packet with the same (Originator,DID) to Next Hop n (Last Next Hop attempted), it assumes that this resulted in a loop, and the node then attempts to reroute the packet to Next Hop n+1 (if available), or to send it back to Previous Hop if no other candidate next hop are available. This, however, may be a false loop detection, therefore the node does not poison entries in the routing table whenever the forwarded packet has the duplicate flag activated.

For packets encapsulated according to [RFC4944] that do not include a DFF mesh header, the DFF node processes them with a simple forwarding

mechanism that selects the next hop with the lowest cost to reach the final destination. In this case, the node does node create any entries in the loop detection table, and it does not attempt to reroute such packets through alternate paths. This forwarding option allows for the coexistence of DFF nodes with nodes that do not follow the message formats defined in this document (Figure 1). A 6lowpan mesh header [RFC4944] is still required for the operation of this basic forwarding mechanism.

5. Message Formats

This document assumes that multi-hop forwarding occurs in the adaptation layer following the message format of [RFC4944]. [RFC4944] indicates that hop-by-hop processing headers with additional mesh routing capabilities may be expressed by defining additional headers that precede fragmentation or addressing headers. Hence, all data packets to be forwarded using DFF MUST be preceded by the standard mesh (L2) addressing header defined in [RFC4944], and MAY be preceded by a header that identifies the data forwarding mechanism (in this case DFF).

After these two headers, other LoWPAN headers such as hop-by-hop options, header compression or fragmentation can also be included before the actual payload. (Figure 1) shows the mesh headers of a data frame to be forwarded with DFF.

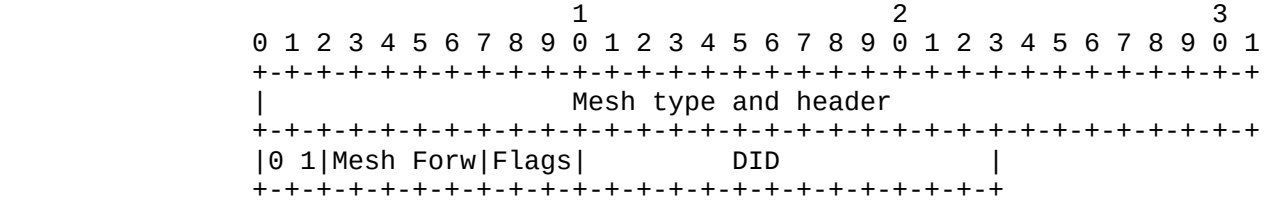


Figure 1: Mesh Header for DFF data frames

Field definitions are as follows:

Mesh type and header: The mesh (L2) addressing header and its associated dispatch byte as defined in [RFC4944].

Mesh Forw: is a 6-bit identifier that allows for the use of different mesh forwarding mechanisms. As specified in [RFC4944], additional mesh forwarding mechanisms should use the reserved dispatch byte values following LOWPAN_BCO; therefore, 01 SHOULD precede Mesh Forw. A possible value to use as a mesh forwarding identifier based on the reserved ranges defined in [RFC4944] is

010001. In this case the dispatch byte would be 01010001.

Flags: Bits in this field are used by DFF to set control flags. 0xx means that this current packet is not a duplicate, and 1xx is used to identify a potential duplicate. The last two bits are reserved to define possible new options for data forwarding.

DID: This is Data Frame Identifier. It is a sequence number generated by the Originator. The originator address concatenated with the DID sequence number form an identifier of previously seen data packets.

6. Data Structures

The loop detection option is based on the idea of storing the DID and originator-ID of a data packet, so that if a packet containing the same DID identifier and originator is received, DFF detects it as a loop.

After the loop is detected, DFF follows a distributed depth-first search for the destination through the candidate next hops kept in the routing table. In order to do a Depth-First search, nodes need to keep a list of their children (i.e., the candidate next hops that have been used to forward the packet), and the previous hop (the node who sent the data packet for the first time to the current router).

A Loop Detection Table (Figure 2) needs to be kept by the nodes to support the loop detection functionality. The candidate next hop field does not need to be pre-stored, it can be filled dynamically as soon as the node attempts to send the packet to a next-hop neighbor.

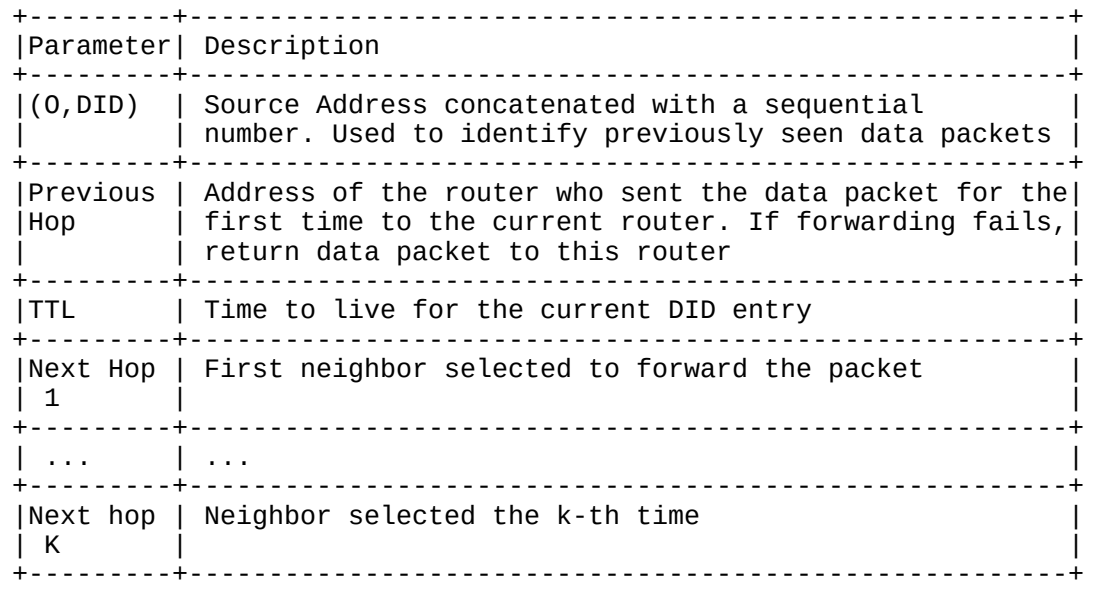


Figure 2: Basic Elements of a Loop Detection Table

7. Acknowledgements

Ganesh Venkatesh, and Geoff Mulligan provided useful discussions which helped shape this document.

8. IANA Considerations

This memo includes the request of a new dispatch byte to identify DFF headers. In the Appendix there is an implementation that avoids the use of new dispatch bytes.

9. Security Considerations

The security of a mesh forwarding protocol depends on the integrity, authentication, and confidentiality of the messages. The security mechanisms for protecting the network can be provided by link-layer technologies. Further details are presented in the Security Considerations section of [RFC4944].

10. Appendix A: Example Implementation of a Control Plane for DFF

There are many route discovery protocols compatible with DFF. The final selection of which control plane to use depends on the particular constraints and requirements of the network. For example, if nodes have tight memory constraints and the network is large, managing the size of the routing table is important. Therefore, a control plane that builds a network with a routing table that grows at a slower rate than the size of the network--e.g., via hierarchical routing, or clustering--is important. If minimizing the routing stretch of the network is a priority, then the control plane needs to keep larger routing tables.

The only condition for a control plane in order to leverage an implementation of DFF is that, nodes should maintain a number of alternate routes, which are being advertised by multiple neighbors and which can be used immediately if the selected route were to fail, or if a loop is detected through a previous route.

While a number of routing protocols satisfy the above constraint, they tend to include extra overhead for preventing loops or dealing with routing inconsistencies or failures. One of the primary goals of DFF is to avoid the use of these extra control messages. This appendix presents a basic control plane compatible with DFF.

TBD.

11. Appendix B: Implementing DFF without requesting new dispatch bytes

DFF can be implemented as a full standard conforming to [RFC4944] without requesting any new dispatch bytes. In this way, nodes implementing DFF can interoperate with other nodes that only implement headers defined in [RFC4944].

A possible way to avoid the DFF mesh header is by overloading the datagram_tag and datagram_offset fields of the fragmentation header defined in [RFC4944].

Because each source maintains a sequence number for the datagram_tag, and the datagram_offset can be used to differentiate between fragmented packets with the same value in datagram_tag, the DID value required by DFF can be generated by the concatenation of the datagram_tag and datagram_offset values of a fragmented data frame.

Nonetheless, an implementation of DFF that avoids the request of a new dispatch byte will prevent the use of flags, and without the existence of a duplicate flag, duplicate packets will not be

detected. Therefore, it is RECOMMENDED that nodes that implement DFF by using the datagram_tag and datagram_offset fields for storing the DID value, do not reroute on link-layer ACK failures, but only on loop detections. In this case, all previously seen (Originator,DID) values can be assumed to correspond to loop detections, and the routing table cost to reach the final destination via the last attempted neighbor can be safely poisoned, without the risk of poisoning valid routes taken by duplicate packets.

This implementation of DFF assumes the existence of fragmentation headers within the LoWPAN encapsulation. This works well if data packets are fragmented, but if the entire payload datagram fits within a single 802.15.4 frame, then [RFC4944] states that the LoWPAN encapsulation should not contain a fragmentation header. However, the use of a fragmentation header for a packet that does not need to be fragmented should, in principle, not affect the operation of nodes implementing [RFC4944]. Therefore, even if a packet does not need to be fragmented, the originator node can append the fragmentation header so DFF nodes can use it for extracting the DID identifier.

The control plane used to populate the routing tables can also avoid the need to request a new dispatch byte by encapsulating routing updates in UDP packets.

12. 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.

Authors' Addresses

Alvaro A. Cardenas
Fujitsu Laboratories
1240 E. Arques Avenue, M/S 345
Sunnyvale, CA 94085
US

Phone: +1 408 530-4516
Email: alvaro.cardenas-mora@us.fujitsu.com

Sandra Cespedes
University of Waterloo
200 University Ave. W.
Waterloo, ON N2L 3G1
Canada

Phone: +1 (519) 8884567 x37448
Email: slcesped@bbcr.uwaterloo.ca

Tadashige Iwao
Fujitsu Limited
Fujitsu Kyushu R and D Center, 2-1, Momochihama 2-chome, Sawara-ku.
Fukuoka,
JP

Phone: +81-92-821-8030
Email: smartnetpro-iwao_std@ml.css.fujitsu.com

IPv6 over Low Power WPAN (6lowpan)
Internet-Draft
Intended status: Informational
Expires: September 16, 2011

S. Park
Samsung Electronics
K. Kim
Ajou University
W. Haddad (Ed.)
S. Chakrabarti
Ericsson
J. Laganier
Juniper

March 15, 2011

IPv6 over Low Power WPAN Security Analysis
draft-daniel-6lowpan-security-analysis-05

Abstract

This document discusses possible threats and security options for IPv6-over-IEEE802.15.4 networks. Its goal is to raise awareness about security issues in IPv6 lowPan networks.

Status of this Memo

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

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

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

This Internet-Draft will expire on September 16, 2011.

Copyright Notice

Table of Contents

1. Introduction

IEEE 802.15.4 [ieee802.15.4] specification defines Physical and MAC layers targeted for the Low Rate Wireless Personal Area Networks (LR-WPAN) using short distance applications with low power and low cost communication networks, particularly for the short range applications such as Wireless Sensors Network (WSN). In an IEEE 802.15.4 compliant WPAN, a central controller device, i.e., the PAN coordinator, builds a WPAN with other devices within a small physical space known as the personal operating system. IEEE 802.15.4 is designed to support a variety of applications in personal area networks; many of these applications are security sensitive. The principal goal of the 6lowpan working group is to design IPv6 transmission over IEEE 802.15.4.

In fact, some of the IEEE 802.15.4 optional features actually reduce security and implementation would be limited for their extensions. The applications range from defense systems to building monitoring, fire-safety, patient monitoring, etc. If the network is not secured, an intruder can inject incorrect messages to trigger false situations.

IEEE 802.15.4 working group is trying to improving the link-layer security specification. However, this document will focus on discussing different security threats from the 6lowpan perspective and discuss different options for applying existing security methods to overcome/alleviate these threats. The main goal is to provide a trust model using both link-layer and IP layer security packages whenever possible.

Designing a new security protocol for 6lowpan network is out of scope of this document. However, the document states desired security requirements, which can be fed into the appropriate IETF security working group in order to design appropriate security protocols.

2. 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 [RFC2119].

3. Terminology

This document uses terminology specific to IPv6 and DHCPv6 as defined in the "Terminology" section of the DHCPv6 specification [RFC3315].

4. Overview

As described in [RFC4919], unlike regular IP network, 6lowpan has some special characteristics such as small packet size, low bandwidth, large number of devices, etc. 6lowpan devices are generally assumed to be resource-limited with respect to computation power, storage, memory and especially battery life. One common feature, which is worthy to remember is the disproportionately high cost of transmitting information as compared to performing local computation. For example, a Berkeley mote spends approximately 800 instructions as it does in sending a single bit [Madden]. It thus become a main design criteria for 6lowpan to reduce the number of bits forwarded by intermediate nodes, in order to extend the entire network's lifetime as recharging may not be practical in some deployment scenarios.

IEEE 802.15.4 nodes can operate in either secure mode or non-secure mode. Two security modes are defined in the specification in order to achieve different security objectives:

- Access Control List (ACL) mode which provides limited security services and requires each device to maintain its own ACL. This mode allows receiving frames only from nodes that are present in the devices's ACL, i.e., considered as trusted nodes. Frames from non-registered devices are filtered. However, cryptographic protection is not provided in this mode.

- Secure mode provides all the security services according to the defined security suite. It provides confidentiality of the frame along with the message integrity, access control, and sequential freshness.

However, the specification is not clear about key management methods, state of ACL table in the event of power loss and support of group keying in which case, network shared common key may be an answer for the link layer security but is vulnerable to replay attacks launched from stolen devices. Yet, in most common cases, network shared keying can be the simple answer to the link layer security as it is easily configurable among large number of devices.

The security aspect, however, seems a bit tradeoff in the 6lowpan since security is always a costly function. This is particularly true to low rate WPAN. Obviously, adding security makes the issue even more challenging. For instance, when putting IPv6 on top of 6lowpan, it may seem possible to use IP security protocol [RFC4301] and turn off the security mechanism defined by IEEE 802.15.4. But on the other hand, IPsec is relatively mature for services at IP or upper layers. Furthermore, due to their inherent properties and/or

constraints mentioned earlier, 6lowpan poses unique challenges to which, traditional security techniques cannot be applied directly. For example, public key cryptography primitives are typically avoided (as being too expensive) as are relatively heavyweight conventional encryption methods.

Consequently, it becomes questionable whether the 6lowpan devices can support IPsec as it is. This document explains in the following sections some of the difficulties resulting from adopting IPSec. However, Layer 2 security must be used for all associated operations such as MAC sub-layer association, beaconing, orphaning, etc.

While IPsec is mandatory with IPv6, considering the power constraints and limited processing capabilities of IEEE802.15.4 capable devices, IPsec is computationally expensive; Internet key exchange (IKEv2) messaging described in [RFC5996] will not work well in 6lowpans as we want to minimize the amount of signaling in these networks. Thus, 6lowpan may need to define its own keying management method(s) that requires minimum overhead in packet-size and in number of signaling messages exchange. IPsec will provide authentication and confidentiality between end-nodes and across multiple lowpan-links, and may be useful only when two nodes want to apply security to all exchanged messages. However, in most cases, the security may be requested at the application layer as needed, while other messages can flow in the network without security overhead.

Attacks against 6lowpands can be classified into external attacks and internal ones. In an external attack, the attacker is not an authorized entity of the 6lowpan. External attacks can be further divided into two categories: passive and active. Passive attacks involve mainly eavesdropping on network’s radio frequency range in an attempt to discover sensitive information. Among active attacks against 6lowpans, denial-of service (DoS) attack at the physical layer can produce devastating consequences. To this end, the attacker can broadcast a powerful signal within the WPAN zone, i.e., jamming, and paralyzes part(s) or even the entire network.

An attacker may also disable a 6lowpan node (e.g., by smashing it!) or capture one, extracts the key(s) and uses it for eavesdropping purposes and/or to directly intervene at some point in time, by injecting false but valid data in order to disturb the overall system, e.g., trigger an undesired chain of events. Consequently, a challenging issue facing 6lowpans is to provide resiliency against node capture attack.

Data collection and dissemination being their ultimate goals, 6lowpans also highlights privacy concerns. In fact, as devices are in general, getting smaller (i.e., easier to conceal) and cheaper

(i.e., easier to obtain), an obvious risk is that 6lowpan technology might be used for privacy violation purposes, e.g., employers might spy on their employees, neighbors might spy on each other.

Possible threats in 6lopwan include intrusion, sink-hole and replay attacks. As in traditional networks, routing mechanisms in 6lowpan present another window from which, an attacker might disrupt and significantly degrade the 6lowpan overall performance. Attacks against unsecure routing aim mainly to contaminate WPAN networks with false routing information resulting in routing inconsistencies. A malicious node can also snoop packets and then launch replay attacks on the 6lowpan nodes. These attacks can cause harm especially when the attacker is a high-power device, such as laptop. It can also easily drain 6lowpan devices batteries by sending broadcast messages, redirecting routes etc.

A possible solution to address security issues in the 6lowpan networks might consist of implementing application level security, e.g., SSL, on top of link layer security. In such case, link layer security protects from intrusion and the application level security protects from another user peeking at the data and against impersonation.

5. Security Challenges

We summarize the security challenges in 6lowpan networks as it follows (for more information about this section and the following ones, please check the references):

- Minimizing resource consumption and maximizing security performance.

- 6lowpan deployment enables link attacks ranging from passive eavesdropping to active interfering.

- In-network processing involves intermediate nodes in end-to-end information transfer.

- 6lowpan communication characteristics render traditional wired based security schemes unsuitable.

6. Security Requirements

Security requirements for 6lowpan can be listed as it follows:

- Data Confidentiality: make information inaccessible to unauthorized users. For example, a 6lowpan node should not leak some of its collected data to neighboring networks.

- Data Authentication: since an adversary can easily inject messages, the receiver needs to ensure that data are originated from a trusted sources.

- Data Integrity: ensures that the received data is not altered in transit by an adversary.

- Data freshness: this could mean data freshness as well as key freshness. Informally, data freshness implies that each data is recent, and it ensures that no adversary replayed old messages.

- Availability: ensures the survivability of network services to (only) authorized parties when needed, despite a DoS attack(s).

- Robustness: ensures operation continuity despite abnormalities, such as attacks, failed nodes, etc.

- Resiliency: is the network ability to provide and maintain an acceptable level of security in case some nodes are compromised.

- Resistance: is the network ability to prevent the adversary from gaining full control of the network by node replication attack in case some nodes are compromised.

- Energy efficiency: a security scheme must be energy efficient so as to maximize network lifetime.

- Assurance: is the ability to disseminate different information at different assurance levels.

7. Security Threats

Most of the attacks and threats against user and data security in 6lowpan are plausible and MAY be very destructible in effect, because of its wireless radio access and connectivity to the Internet. The security analysis of 6lowpan starts with the appreciation of various threats posed at respective ISO OSI layers. In this section, we classify and discuss the threats in layer-wise order. The suggested threat model assumes that the attacker is fully capable at all times except during the deployment phase.

6lowpan is highly susceptible to physical attacks. i.e., threats due to physical node destruction relocation and masking. By physical attacks, one or multiple 6lowpan nodes can be knocked out permanently, so the losses are irreversible. Physical attack can extract cryptographic secrets from the associated circuitry, modify programming in the nodes, and may allow the malicious node to take control over them. These compromises can result into code modification inside the node and to change the mission-oriented role of full fledged networks, let alone sensors.

In 6lowpan environment, several types of DoS attacks can be triggered in different layers. At the physical Layer, the DoS attacks can be launched by tampering and jamming electromagnetic (EM) signals by swarming the limited resources of 6lowpan devices with the high resource devices very easily.

Attacks on MAC layer involves collision, exhaustion and unfairness. Being always power hungry, 6lowpan devices try to sleep as often as possible in order to conserve it. Such constraints open the door for an attacker to let the device execute a large number of tasks in order to deplete its battery. This is called "sleep deprivation torture" [Stajano]. To achieve such goal, an attacker can for example, target different destination devices with unnecessary packets, possibly in other WPANs, regardless of whether the destination WPAN and/or device actually exist or not. Such attack can also lead to depleting the PAN coordinator battery power, i.e., since the downlink packets have to be explicitly requested from the PAN coordinator, this will keep it busy (as well as an eventual destination).

An attack against network availability can consist of flooding the network by simply transmitting a large number of large(st) packets size. In such case, the attacker may degrade the network performance and drastically reduce the throughput.

In WPAN specification, the replayed message is prevented by the replay protection mechanism, i.e., sequential freshness. In a

replay-protection attack, the malicious node sends many frames containing large counters to a particular receiver, which in turn raises the replay counter up. Then, when a normal device sends a frame with a lower frame counter, it will be rejected by the receiver and thus, leading to DoS attack.

As the ACK frame integrity is not protected, it also opens the door for a malicious node to prevent a legitimate device from receiving a particular frame. This is possible by forging an ACK using the un-encrypted sequence number from the data frame and sending it to the source while creating enough interference, in order to prevent the legitimate receiver from receiving the frame. In such scenario, the source device is led to believe that the frame has been received.

A corrupted device can also attack the key distribution process since the WPAN coordinator announces the IDs of devices who are about the change the link key in plain-text in the beacon frame. Therefore, the attacker can send request packet with the ID of the legitimate node. The goal from such request is to push the coordinator to trigger a key exchange process while the legitimate recipient may not be ready.

Attacks against network layer fall into one of the following categories:

- Spoofed, altered, or replayed routing information: in this attack, the malicious node uses spoofing, altering and/or replaying to target routing information exchanged between nodes in an attempt to create routing loops, attract/repel network traffic, extend/shorten source routes, generate false error messages, etc.

- Selective forwarding: in this attack, the malicious device may refuse to forward certain messages (e.g., by dropping them). In this case, neighboring devices may conclude that the malicious device has failed and thus, try to seek another route. A more subtle form of this attack is when the malicious device selectively forward packets in which case, neighboring nodes won’t be able to reach the conclusion that another route is needed which in turn, would encourage them to re-send the data packets.

- Sinkhole attack: in a sinkhole attack, the malicious device tries to get all traffic from one particular area which can potentially result in a DoS attack. In order to launch a sinkhole attack (aka blackhole attack), the attacker can listen to requests for routes then replies to the requesting nodes that it contains the high quality or shortest path to the base station. Once the malicious device is able to insert itself between the communicating nodes, he/she is able to do anything with the packets passing through it. In

fact, this attack can affect even the nodes that are spatially located farther from the malicious node.

- Sybil attack: in a Sybil attack, a single node presents multiple identities to other nodes in the WPAN. Sybil attacks pose a significant threat to geographic routing protocols and MAY be performed against the distributed storage, routing mechanism, data aggregation, voting, fair resource-allocation and misbehavior detection, etc. Note that it is not easy to detect a Sybil attack in progress (measuring the usage of radio resources MAY lead to detect it, though with very little probability).

- Wormhole attack: in a Wormhole attack, the attacker records packets (or bits) at one location in the network and tunnels them to another one. Such attacks can be devastating to the working of the 6lowpan since it does not require compromising a node in the WPAN; instead, it could be performed at the initial phase when 6lowpan nodes start to discover the neighboring information. Wormhole attacks can target for example, routing function or application.

- Neighbor Discovery attacks: a modified version of the IPv6 Neighbor Discovery protocol (described in [RFC4861]) has been specifically designed for WPAN. However, the modified version (described in [I-D.ietf-6lowpan-nd] inherits threats which applies in the WPAN deployment. This includes unsecured router advertisement, neighbor discovery DoS attacks. Threats against neighbor discovery protocol are described in [RFC3756].

At the transport layer, attacks could be performed by half open and half closed TCP segments. A malicious device can repeatedly forge messages carrying sequence numbers or control flags which will ultimately cause the endpoints to request retransmission of missed frames.

8. Assumptions

[RFC4919] describes two security concerns as follows;

In Section 4.6 Security: Although IEEE 802.15.4 provides AES link layer security, a complete end-to-end security is needed.

In Section 5 Goals: Security threats at different layers must be clearly understood and documented. Bootstrapping of devices into a secure network could also be considered given the location, limited display, high density and ad hoc deployment of devices.

This document will meet the above considerations.

In addition, existing IP security technologies will be simplified to be implemented on the 6lowpan small devices. 6lowpan security architecture will shed off lots of fat from IP security technologies whenever available.

IEEE 802.15.4 AES (Advanced Encryption Standard) will be used for 6lowpan security architecture in conjunction with IP security whenever available.

9. 6Lowpan Security Analysis

In this section, both IEEE 802.15.4 MAC security and IP security are tackled to search for a new 6lowpan trust models and available solution spaces if feasible. The principal object of this analysis is to improve 6lowpan security level as we use IP layer security mechanism as possible regardless of 802.15.4 vulnerable MAC security. 802.15.4 MAC enhancement and amendment are not scope of this document but IEEE 802 standard stuff.

9.1. IEEE 802.15.4 Security analysis

As mentioned earlier, IEEE 802.15.4 MAC layer provides security services that are controlled by the MAC PIB (PAN Information Base). For security purpose, the MAC sublayer maintains an access control list (ACL) in its MAC PIB. By specifying a security suite in the ACL for a communication peer, a device can indicate what security level should be used (i.e., none, access control, data encryption, frame integrity, etc.) for communications with that peer.

A critical function of IEEE 802.15.4 MAC is frame security. Frame security is actually a set of optional services that may be provided by the MAC to the upper layers (applications). The standard strikes a balance between the need for these services in many applications, and the desire to minimize the burden of their implementation on those applications that do not need them. As described in [802.15.4-ACM], if an application does not set any security parameters, then security is not enabled by default. IEEE 802.15.4 defines four packet types: beacon packets, data packets, acknowledgements packets and control packets for the media access control layer. It does not support security for acknowledgement packets. But on the other hand, other packet types can optionally support integrity and confidentiality protection for the packet’s data field.

Due to the variety of applications targeted by IEEE 802.15.4, the processes of authentication and key exchange are not defined in the standard. Devices without the key cannot decrypt the encrypted messages.

In addition, unsecured mode is suitable for some applications in which implementation cost is important, and security is either not required or obtained in other ways. An example of this is that all 6lowpan devices are assigned a default key by the administrator they can exchange data encrypted with that key. This may work in some situations, but this solution is not quite scalable. In this case, 802.15.4 node is very vulnerable.

The security service enables the MAC to select the devices with which

it is willing to communicate. The device may decide to communicate with some devices, and not others. To minimize complexity, the access control service is performed on an individual device basis, rather than on groups or collections of devices.

Unlike file transfer or voice communication applications common with other protocols, IEEE 802.15.4 applications often transmit messages that do not convey secret information.

9.2. IP Security analysis

IPsec can guarantee integrity and optionally confidentiality of IP (v4 or v6) packets exchanged between two peers.

Basically, IPsec works well on non-low-power devices which are not subject to severe constraints on host software size, processing and transmission capacities. IPsec supports AH for authenticating the IP header and ESP for authenticating and encrypting the payload. The main issues of using IPsec are two-fold: (1) processing power and (2) key management. Since these tiny 6lowpan devices do not process huge number of data or communicate with many different nodes, it is not well understood if complete implementation of SADB, policy-database and dynamic key-management protocol are appropriate for these small battery powered devices.

Given existing constraints in 6lowpan environments, IPsec may not be suitable to use in such environments, especially that 6lowpan node may not be able to operate all IPsec algorithms on its own capability either FFD or RFD.

Bandwidth is a very scarce resource in 6lowpan environments. The fact that IPsec additionally requires another header (AH or ESP) in every packet makes its use problematic in 6lowpan environments.

IPsec requires two communicating peers to share a secret key that is typically established dynamically with the Internet Key Exchange (IKEv2) protocol. Thus, it has an additional packet overhead incurred by IKEv2 packets exchange.

As neighbor discovery protocol will be applied to 6lowpan, Secure Neighbor Discovery (SeND) protocol [RFC3971] should be considered to provide security in conjunction with 6lowpan NDP. SeND works well over existing IP networks. However, the crypto-generated address (CGA) (described in [RFC3972]) used in SeND is based on RSA based and thus, requires larger packet-size and processing time than in the case where Elliptic Curve Cryptography (ECC) keying algorithm is used. Therefore, it could be reasonable to use the SeND protocol if it is extended to support ECC for 6lowpan networks application.

10. Key Management in 6Lowpan

In order to provide security in 6lowpans, a robust encryption mechanism MUST be in place. Only the non-tamperable keys can provide an encryption infrastructure that is thorough enough to provide a wide range of security services such as but not limited to authentication, authorization, non-repudiation and prevention from replay attacks. Key management issues are discussed in the following section.

10.1. Existing Key Management Methods

The characteristics of 6lowpan communicating devices and resulting WPANs, such as limited resources at the node and network level, lack of physical protection, unattended operation, and a close interaction with the physical environment, all make it infeasible to implement some of the most popular key exchange techniques in their literal forms for 6lowpans. In this section, we visit three widely known schemes such as trusted-server scheme, pre-distribution scheme and public key cryptography schemes in order to reach a pragmatic key management mechanism for 6lowpans.

The trusted-server scheme relies solely on the server for key agreement between nodes, e.g., Kerberos. If the server is compromised, the trust amongst nodes is severed. Such scheme is not suitable for 6lowpan networks because there is usually no guarantee of seamless communication with a trusted server at anytime.

The key agreement scheme is key pre-distribution, where key information is distributed among all 6lowpan nodes prior to deployment. If the network deployers were to know which nodes were more likely to stay in the same neighborhood before deployment, keys MAY be decided a priori. However, because of the randomness of the deployment, knowing the set of neighbors deterministically might not be feasible. Furthermore, the presence of intruder nodes right from the network deployment and initiation time cannot be rejected outright as implausible. Some schemes like network shared keying, pair-wise keying, and group keying, have been defined as variants of key distribution. On-site key management mechanisms, while warranting the same level of security as key pre-distribution schemes have an obvious edge to cope up with network dynamics.

This class of key management scheme depends on asymmetric cryptography, such as public key certificates that are irreversible singularly. This irreversibility comes at a price-often staked by the limited computation and energy resources of 6lowpan nodes. However, these are the hardest cryptanalyze. Some of the most popular examples include, but are not limited to Diffie-Hellman key

agreement, RSA or ECC [RFC2631]. Recent works on ECC implementation for low power devices has proven its feasibility for sensor networks. ECC provides security with smaller key size that is comparable to security provided by RSA or AES with much higher key size.

Network topologies for 6LowPan (i.e., star and mesh) and presence of FFD and RFD makes cluster based dynamic key management schemes seem the most appropriate. These schemes use Master Keys; Network Keys and Links keys which could be pre-installed for first round and can be distributed by key transport mechanism during later rounds. This scheme provides ease in key distribution and key revocation [ZigBee].

10.2. Issues With Key Management in 6Lowpan

- In a 6lowpan, a malicious node MAY sit amongst other nodes at the deployment phase-a problem of secure key assignment at bootstrap time.

- A node is compromised during the operating time of 6lowpan-A key revocation system MUST be employed.

- In a sleep-mode enabled 6lowpan, keys to sleeping nodes MUST be deprived and reinstated after such nodes resume active state.

- In case the keys are compromised, a mechanism to diagnose security violation MUST be invoked.

- It SHOULD allow and support dynamic addition of a new node.

11. Security Consideration in Bootstrapping a 6lowpan Node

This section aims to discuss how does a node configures itself securely with a IPv6 router in the network. It involves assignment of IPv6 prefix and the default IPv6 router in the 6lowpan. Further details will be collaborated with 6lowpan commissioning/bootstrapping works in near future according to the 6lowpan working group rechartering.

12. Possible Scenarios Using Different Levels of Security

This section may suggest example scenarios with example solutions in different cases (IPsec, SSL, other type of solutions) although this document does not recommend or specify any security solutions. Further details will be collaborated with 6lowpan architecture works in near future according to the 6lowpan working group re-chartering.

13. Security Considerations

This document addresses only security issues around IPv6 over Low Power WPAN.

14. IANA Considerations

There is no IANA considerations.

15. Acknowledgements

Thanks to Myungjong Lee at CUNY, USA, Rabia Iqbal, Mustafa Hasan and Ali Hammad Akbar all at Ajou University for their valuable comments to improve the document. Special thanks to Jung-Hyun Oh for his valuable help on this document.

16. No I-D References

All references shown in this section MUST be added into the Informative References before publishing it officially.

[ieee802.15.4] IEEE Std., 802.15.4-2003, ISBN 0-7381-3677-5, May 2003.

[802.15.4-ACM] Sastry, N. and Wagner, D., Security Considerations for IEEE 802.15.4 Networks, ACM WiSE'04, October 2004.

[Madden] Madden, S. R., Franklin, M. J., Hellerstein, J. M., and Hong, W., "TAG: a Tiny AGgregation service for ad-hoc sensor networks". In Proceedings of the 5th Annual Symposium on Operating Systems Design and Implementation, 2002.

[Stajano] Stajano, F., and Anderson, R., "The Resurrecting Duckling: Security Issues for Ubiquitous Computing". In IEEE Computer Journal, Volume 42, Issue 5, 2002.

[WSN] Shi, E., and Perrig, A., "Designing Secure Sensor Networks", In IEEE Wireless Communications, December 2004.

[MAC802154] Misic V. B., Fung J., and Misic, J., "MAC Layer Security of 802.15.4-Compliant Networks". In MASS 2005 Workshop, IEEE WSN Conference.

[SEC802154] Xiao, Y., Sethi, S., Chen, H. H., and Sun B., "Security Services and Enhancements in the IEEE 802.15.4 Wireless Sensor Networks". In IEEE GlobeCom 2005.

[SECWSN] Chen, X., Makki, K., Yen, K., and Pissinou, N., "Sensor Network Security: A Survey". In IEEE Communications Surveys & Tutorials, Volume 11, No. 2, 2nd Quarter 2009.

[ZigBee] Specification Version 1.0, December 2004.

17. References

17.1. Normative References

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

[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.

17.2. Informative References

[I-D.ietf-6lowpan-nd] Shelby, Z., Chakrabarti, S., and E. Nordmark, "Neighbor Discovery Optimization for Low-power and Lossy Networks", draft-ietf-6lowpan-nd-15 (work in progress), December 2010.

[RFC2631] Rescorla, E., "Diffie-Hellman Key Agreement Method", RFC 2631, June 1999.

[RFC3756] Nikander, P., Kempf, J., and E. Nordmark, "IPv6 Neighbor Discovery (ND) Trust Models and Threats", RFC 3756, May 2004.

[RFC3971] Arkko, J., Kempf, J., Zill, B., and P. Nikander, "SEcure Neighbor Discovery (SEND)", RFC 3971, March 2005.

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

[RFC4301] Kent, S. and K. Seo, "Security Architecture for the Internet Protocol", RFC 4301, December 2005.

[RFC4861] Narten, T., Nordmark, E., Simpson, W., and H. Soliman, "Neighbor Discovery for IP version 6 (IPv6)", RFC 4861, September 2007.

[RFC4919] Kushalnagar, N., Montenegro, G., and C. Schumacher, "IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs): Overview, Assumptions, Problem Statement, and Goals", RFC 4919, August 2007.

[RFC5996] Kaufman, C., Hoffman, P., Nir, Y., and P. Eronen, "Internet Key Exchange Protocol Version 2 (IKEv2)", RFC 5996, September 2010.

Authors' Addresses

Soohong Daniel Park
System Solution Laboratory, Samsung Electronics
416 Maetan-3dong, Yeongtong-gu
Suwon-si, Gyeonggi-do 442-742
KOREA

Phone: +82 31 200 4635
Email: soohong.park@samsung.com

Ki-Hyung Kim
Ajou University
San 5 Wonchun-dong, Yeongtong-gu
Suwon-si, Gyeonggi-do 442-749
KOREA

Phone: +82 31 219 2433
Email: kkim86@ajou.ac.kr

Wassim Michel Haddad
Ericsson
300 Holger Way
San Jose, CA 95134
US

Phone: +1 646 256 2030
Email: Wassim.Haddad@ericsson.com

Samita Chakrabarti
Ericsson
300 Holger Way
San Joe, CA
USA

Email: samita.chakrabarti@ericsson.com

Julien Laganier
Juniper
Sunnyvale, CA
USA

Email: Julien.ietf@laposte.net

6LowPAN Network Working Group
Internet Draft
Expires: September 30, 2011

Hyun K. Kahng
Dae-In, Choi
Korea University
Suyeon, Kim
Mobilab

March 12, 2011

Global connectivity in 6LoWPAN
draft-kahng-6lowpan-global-connectivity-01.txt

Abstract

This document specifies the translation mechanism mapping IPv6 address with 128 bits to the Adaptation Identifier (AID) with 16 bits. When a device in IEEE 802.15.4 domain needs to communicate with other nodes in IPv6 domain, it should acquire source AID and destination AID corresponding to source IPv6 address and destination IPv6 address, respectively from an IPv6 translation-capable gateway. The node will send packets using these AIDs to the gateway, and then the gateway will translate them to normal IPv6 addresses using the mapping table already associated.

Table of Contents

1. Introduction

IETF 6LoWPAN Working Group is an IPv6 based low-power wireless area network and has been working for IPv6 packets to be sent to and received from over IEEE 802.15.4 based networks for applications which require wireless internet connectivity at lower data rates for devices.

However it is well known that the management of addresses for devices that communicate across the two dissimilar domains of IPv6 and IEEE802.15.4 is complicated. IEEE802.15.4 standard packet size is 127 bytes, among which IEEE 64 bit extended addresses may be used. After an association, 16 bits are used as a unique ID in a PAN L2, Still only 102 bytes are available for payload at MAC layer. Now considering the devices need to communicate with other nodes via IPv6 domain, 256 bits of the source and destination addresses seem to be cumbersome in a limited MAC payload fields.

It is obvious that the IP connectivity between 6LowPANs and the global IPv6 networks is necessary. For this connectivity, [I-D.6lowpan-interoperability] proposes the mapping of 16 bits short address and the interoperability between 6LowPAN devices and the external IPV6 networks. However this document does not specify multiple network prefix address but does single network prefix address to use 16 bit short addresses. In the case of the network configuration for the connectivity between 6LowPANs and the external IPv6 networks, multiple network prefix address must be considered for several connections between them. So this document describes the AID assignment mechanism in the gateway not only to support multiple network prefix address but also to map unique IPv6 address to AID with short length. The encoding of AID utilizes 2 bytes; 1 byte is used for identifying each node in IEEE 802.15.4. The other 1 byte is used for identifying the external IPV6 node connected with the node of IEEE 802.15.4. How the value of AID is determined in the gateway is out of scope.

As a result, this document defines an encoding format, LOWPAN_GCHC, for effective compression of Global IPv6 address based on shared state with AID. In addition, this document also introduces additional frame format over the header compression format defined in [RFC 4944]

This document is based on [Interoperability of 6LoWPAN] for the adaptation layer of fragmentation and reassembly, the stateless address auto-configuration based on EUI-64[EUI64].

2. Terminology

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

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

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

AID : Adaptation Identifier

GCHC : Global Connectivity Header Compression

3. Global Connectivity in 6LoWPAN

This section defines the gateway architecture for the Global Connectivity between the global IPv6 nodes and 6LowPANs. Figure 1 shows the example of the IPv6 connection between the gateway and the external IPv6 nodes. The gateway performs new AID assignment operation to be mapped with IPV6 address by the request of IEEE 802.16 node and executes translation operation which maps IPv6 address encoded with 16 bytes to AID with 2 bytes to compress packet header and vice versa.

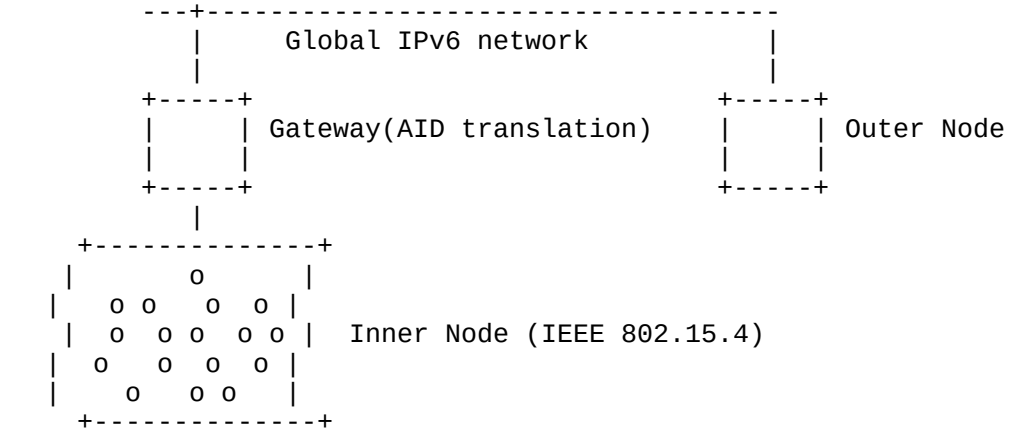


Figure 1: Gateway and node

3.1. AID for outbound traffic

Outbound traffic means the traffic from the inner node to the outer nodes.

1. Each node in 6LowPAN checks if it has an AID identifying its own global IPv6 address.

2. Each node can use the AID if it has. Unless, it should request the new AID to the gateway.

3. Each node in 6LowPAN checks if it has an AID identifying its corresponding global IPv6 address.

4. Each node can use the AID if it has. Unless, it should request to new AID to the gateway.

5. Each node should request to the gateway to delete both AIDs if they didn’t use during a certain time.

3.2. AID for inbound traffic

Inbound traffic means the traffic from the outer node to the inner nodes.

1. The Gateway checks if it has an AID identifying Source IPv6 address of the received inbound traffic.

2. The Gateway can use the AID if it has. Unless, it should assign the new AID for Source IPv6 address in the inbound traffic.

3. The Gateway checks if it has an AID identifying Destination IPv6 address of the received inbound traffic.

4. The Gateway can use the AID if it has. Unless, it should assign the new AID for Destination IPv6 address in the inbound traffic.

5. Each node should request to the gateway to delete both AIDs if they didn’t use during a certain time.

3.3. AID Assignment in the Gateway

An AID must be assigned by the Gateway according to the following assignment method. The length of an AID being assigned is 2 bytes.

1. If the Gateway receives the AID request frame for the specific IPv6 address, it looks up its own address mapping table for the specific IPv6 address.

2. If the Gateway finds the AID matched with the specific IPv6 address, it will return the AID. Otherwise, it will generate and return the new AID not to be duplicated.

3.4. AID deletion in the Gateway

An AID must be deleted by the Gateway according to the following deletion method.

1. If the Gateway receives the AID delete frame for the specific AID, it looks up its own address mapping table for the AID.

2. If the Gateway finds the AID, it will delete the AID. Otherwise, it neglects the frame.

4. Mapping Table Global Connectivity in 6LoWPAN

Gateway has the address mapping table as shown in Figure 2. The length of Global IPv6 address field is 128 bits. The length of AID is 16 bits.

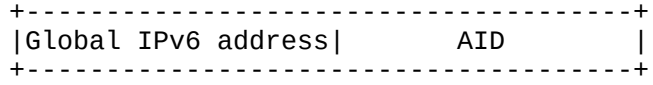


Figure 2: Address Mapping Table

5. Frame Format for AID Assignment

The format shown in Figure 3 and Figure 4 are the payload in the IEEE 802.15.4 MAC protocol data unit (PDU). The Frame format was used in communications to assign AID between internal nodes and gateway. Each AID will represent its corresponding global IP address. The Frame format is defined in Figure 3.

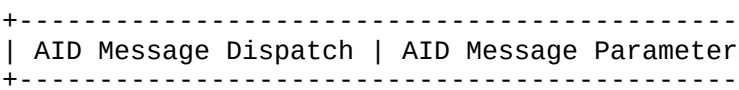


Figure 3: A LoWPAN encapsulated AID Assignment Frame

The details of AID Message Parameters are defined below in Section 6.

The encapsulation format defined in Figure 4 defines LOWPAN_GCHC encoding format for compressing the IPv6 header. To enable effective compression LOWPAN_GCHC relies on AID information pertaining to AID Assignment Frame.

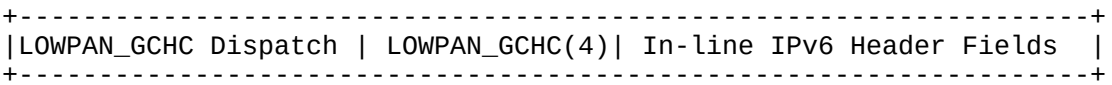


Figure 4: LOWPAN_GCHC Frame

We defined three kinds of frames related to AID assignment as shown Figure 5: AID REQUEST frame, AID REPLY frame and AID DELETE frame. The document allocates the following 3 dispatch type field values for these frame types and 1 dispatch type field value for LOWPAN_GCHC. Figure 5 shows new dispatch value bit pattern as updating Figure 2 in [RFC 4944].

Pattern	Header Type
00 xxxxxx	NALP
01 000001	IPv6
01 000010	LOWPAN_HC1
01 000011	Reserved
...	Reserved
01 001111	Reserved
01 010000	LOWPAN_BC0
01 010001	AID REQUEST
01 010010	AID REPLY
01 010011	AID DELETE
01 010100	LoWPAN_GCHC
...	Reserved
01 111110	Reserved
01 111111	ESC
10 xxxxxx	MESH
11 000xxx	FRAG1
11 001000	Reserved
...	Reserved
11 011111	Reserved
11 100xxx	FRAGN
11 101000	Reserved
...	Reserved
11 111111	Reserved

Figure 5: Dispatch Value Bit Pattern

AID REQUEST : Specifies that the frame is an AID request frame.

AID RESPONSE : Specifies that the frame is an AID response frame.

AID DELETE : Specifies that frame is an AID delete frame.

LOWPAN_GCHC : Specifies that following header is a LOWPAN_GCHC compressed IPV6 header.

5.1. AID REQUEST frame

In case that the node in 6LowPAN needs a new IPV6 connection with the external node in IPV6 network, the node must send AID REQUEST frame to the gateway. The source address and destination address must set to its source address and the destination address of the external node respectively. To get new source AID and destination AID, both AID must be set to zero. The format of AID REQUEST frame is in Figure 6.

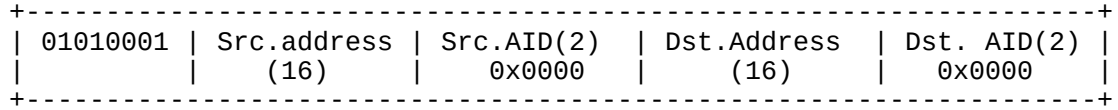


Figure 6: AID REQUEST frame format

5.2. AID REPLY frame

As the response of AID REPLY frame, the gateway must reply to the node in 6LowPAN by sending AID REPLY frame. In case of AID REPLY frame, the frame must be sent to the node in 6LowPAN from the gateway. The gateway must assign unique values as the values of source AID and destination AID. How to calculate and decide the values of the source AID and the destination AID is out of scope except that 1 byte is used for identifying each node in IEEE 802.15.4 and the other 1 byte is used for identifying the external IPV6 node connected with the node of IEEE 802.15.4. AID REPLY frame format is in the Figure 7.

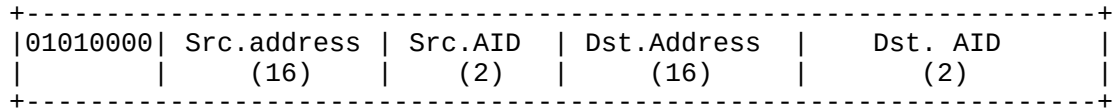


Figure 7: AID REPLY frame format

5.3. AID DELETE frame

The node in 6LowPAN must send AID DELETE frame to the gateway. After the node in 6LowPAN disconnected to the Global IPV6 node, it must send AID DELETE frame to the gateway to inform not to use the values any longer. The format of AID DELETE frame is in the Figure 8.

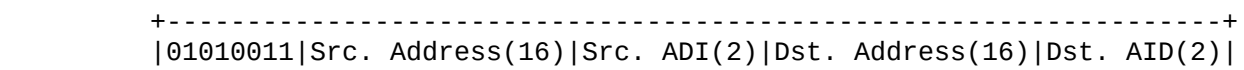


Figure 8: AID DELETE frame format

5.4. LOWPAN_GCHC frame

After AID assignment, LOWPAN_GCHC frame indicates that the following header is encoded using AID values. The LOWPAN_GCHC encoding utilizes 4 octets, 2 of which is source AID and 2 of which is Destination AID. Any information from the uncompressed IPv6 header fields is carried in-line following the LOWPAN_GCHC encoding. The format of LOWPAN_GCHC is in the Figure 9.

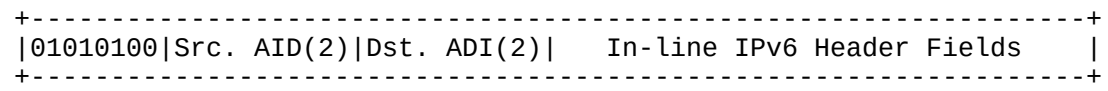


Figure 9: LOWPAN_GCHC frame format

6. Formal Syntax

The following syntax specification uses the augmented Backus-Naur Form (BNF) as described in RFC-2234 [RFC2234].

7. Security Considerations

TBD

8. IANA Considerations

TBD

9. References

9.1. Normative References

[RFC2460] Deering, S. and R. Hinden, "Internet Protocol, Version6 (IPv6) Specification", RFC 2460, December 1998.

[ieee802.15.4] IEEE Computer Society, "IEEE Std. 802.15.4-2003", October 2003.

[RFC4919] N. Kushalnagar, G. Montenegro, C. Schumacher, "IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs): Overview, Assumptions, Problem Statement, and Goals", RFC4919, August 2007.

[RFC4944] G. Montenegro, N. Kushalnagar, J. Hui, D. Culler, "Transmission of IPv6 Packets over IEEE 802.15.4 Networks", RFC4944, September 2007.

[RFC4291] Hinden, R. and S. Deering, "IP Version 6 Addressing Architecture", RFC 4291, February 2006.

9.2. Informative References

[EUI64] "GUIDELINES FOR 64-BIT GLOBAL IDENTIFIER (EUI-64) REGISTRATION AUTHORITY", IEEE, http://standards.ieee.org/regauth/oui/tutorials/EUI64.html.

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

[RFC3561] Perkins, C., Belding-Royer, E., and S. Das, "Ad hoc On-Demand Distance Vector (AODV) Routing", RFC 3561, July 2003.

[RFC3626] Clausen, T. and P. Jacquet, "Optimized Link State Routing Protocol (OLSR)", RFC 3626, October 2003.

[RFC3684] Ogier, R., Templin, F., and M. Lewis, "Topology Dissemination Based on Reverse-Path Forwarding (TBRPF)", RFC 3684, February 2004.

[I-D.6lowpan-interoperability] Ki-Hyung Kim, Seung Wha Yoo, Hee Jung Kim, Soohong Daniel Park, Jae Ho Lee, "Interoperability of 6LoWPAN", draft-daniel-6lowpan-interoperability-01, July 2005

[I-D. 6lowpan-backbone-router] Pascal Thubert, "6LoWPAN Backbone Router", draft-thubert-6lowpan-backbone-router-02, June 2010

Authors' Addresses

Hyun K. Kahng
Korea University
Electronic Information Engineering
Seoul, Korea
Email: kahng@korea.ac.kr

Dae-In Choi
Korea University
Electronic Information Engineering
Seoul, Korea
Email: nbear@korea.ac.kr

Suyeon, Kim
Mobilab
Daegu, Korea
Email: sykim@mobilab.co.kr

Acknowledgement

Funding for the RFC Editor function is currently provided by the Telecommunication Technology Association (TTA)

INTERNET-DRAFT
Intended Status: Experimental
Expires: August 21, 2011

L. Maqueda
KTH
G. Maguire
KTH

March 7, 2011

Guidelines for the Operation of a 6LoWPAN-ND Proxy Gateway
draft-maqueda-6lowpan-pgw-00

Abstract

The IETF 6LoWPAN working group has defined a number of optimizations to adapt traditional IPv6 Neighbor Discovery for Low-power and Lossy Networks (LLNs). As these two ND protocols are incompatible, and Neighbor Discovery has link-local scope, a side effect of these optimizations is that communication between Full Function Devices (FFDs) and 6LoWPAN nodes (6LNs) becomes impossible within the same link, unless the proper proxy mechanisms are applied. This document specifies guidelines for such proxy mechanisms to enable transparent communication between FFDs and 6LNs within the same network link.

Table of Contents

1. Introduction

RFC 4994 [RFC4944] defines an adaptation layer (6LoWPAN) that enables the transmission of IPv6 packets over IEEE 802.15.4 media. However, traditional IPv6 Neighbor Discovery [RFC4861] has proved to be unsuitable for IEEE 802.15.4 links due to its physical and link-layer properties, and to the nature of the target devices 6LoWPAN is intended for. For these reasons, the IETF 6LoWPAN working group suggested a number of improvements/changes to [RFC4861] in draft-ietf-6lowpan-nd [I-D.ietf-6lowpan-nd] in order to adapt traditional IPv6 Neighbor Discovery for 6LoWPAN networks. Although these modifications allow for a more efficient use of each 6LoWPAN node's resources, they cause Neighbor Discovery for IPv6 to be incompatible with this modified version. This incompatibility represents an important constraint for the integration of 6LoWPAN into existing IPv6 networks since it precludes the coexistence of FFDs and 6LoWPAN nodes (6LNs) within the same network link.

This document provides guidelines to overcome this problem, by specifying the proxy operations required to enable transparent communication between FFDs and 6LNs within the same link. It is important to note that the operation described here neither requires modifications to the ND protocol nor human intervention, while permitting each type of device to achieve the maximum benefits of its particular Neighbor Discovery protocol.

1.1. 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 RFC 2119 [RFC2119].

This document requires readers to be familiar with terms and concepts described in [RFC5942], [RFC4861], [RFC4862], [RFC4919], [RFC4944], and [I-D.ietf-6lowpan-nd]. For clarity reasons, a short description of these terms and their corresponding abbreviations is given below. Note that some of these terms override their definitions in the above mentioned documents.

6LP-GW	6LoWPAN Proxy-Gateway. The logic in charge of performing the operations described in this document, grouped as a functional unit. Note that this logic can be implemented in different ways (for example as a separate device or as part of any existing device with forwarding capabilities), as long as at least the two required interfaces (IEEE 802.15.4 and IEEE 802.3) are available.
LLN	Low-power and Lossy Network [RFC5867].
IPv6-ND	IPv6 Neighbor Discovery protocol [RFC4861].
6LoWPAN-ND	6LoWPAN Neighbor Discovery protocol [I-D.ietf-6lowpan-nd].
ND	Neighbor Discovery protocol, either 6LoWPAN-ND or IPv6-ND [RFC4861], [I-D.ietf-6lowpan-nd].
ARO	Address Registration option [I-D.ietf-6lowpan-nd].
ABRO	Authoritative Border Router option [I-D.ietf-6lowpan-nd].
6CO	6LoWPAN Context option [I-D.ietf-6lowpan-nd].
PIO	Prefix Information option [RFC4861].
SLLAO	Source Link-Layer Address option [RFC4861].
TLLAO	Target Link-Layer Address option [RFC4861].
6LR	An intermediate router in the LoWPAN that communicates with other 6LoWPAN routers in the same LoWPAN. 6LoWPAN routers are present only in route-over topologies [I-D.ietf-6lowpan-nd].

6LBR	A border router located at the junction of separate 6LoWPAN networks or between a 6LoWPAN network and another IP network. There may be one or more 6LBRs at the 6LoWPAN network boundary. A 6LBR is the responsible authority for IPv6 Prefix propagation for the 6LoWPAN network it is serving. An isolated LoWPAN also contains a 6LBR in the network, which provides the prefix(es) for the isolated network [I-D.ietf-6lowpan-nd].
6R	6LoWPAN Router, either a 6LR or 6LBR.
6LH	6LoWPAN Host, in contrast with a 6R.
6LN	A 6LoWPAN Node is any host or router participating in a LoWPAN. This term is used when referring to situations in which either a 6LH or 6R can play the role described [I-D.ietf-6lowpan-nd].
RR	Regular IPv6 router, in contrast with a 6R
Router	Either a RR or 6R
Link	A network segment within which all nodes share the same prefix and communication at the IP layer using link-local addresses is possible, regardless of the nature of such addresses or the number of hops required for such communication.

1.2. Assumptions

[I-D.ietf-6lowpan-nd] states that the IPv6-ND protocol optimizations it introduces are compatible with both mesh-under and route-over topologies. The guidelines described here do not affect this compatibility, therefore no assumptions regarding topology will be made unless specifically specified. However, section 3 describes some features that may be of special interest for implementations in the case of route-over topologies.

The link-layer scenario considered here consists only of IEEE 802.15.4 and IEEE 802.3 links. However, the techniques described here may also be applicable to other types of media as long as IPv6-ND is used in one segment while 6LoWPAN-ND is used in the other, but further considerations of such media are out of the scope of this document.

For the same reasons, the mechanisms described here may be used for interconnecting more than two link-layer media, but this is also out of the scope of the present document. Thus for the remainder of this

document, only a two port device with one interface being IEEE 802.3 and the other interface being IEEE 802.15.4 will be explicitly considered. As noted earlier, the 6LP-GW logic could be integrated in a router.

The term "MAC address" will be used indiscriminately in the present document, referring to both 64-bit (IEEE 802.15.4) and 48-bit (IEEE 802.3) MAC addresses since there is an IEEE defined direct mapping from 48-bit MAC addresses to 64-bit addresses [EUI-64].

Due to the nature of the 6LP-GW, the availability of a forwarding mechanism between the two interfaces is assumed. However, there is no assumption regarding this forwarding mechanism, which could be implemented at either layer 2 or 3. Note that, the choice of this forwarding mechanism, need not impose a specific network topology as long as it is applied properly.

6LoWPAN supports different compression mechanisms which may be used or not. This document does not make any assumption regarding compression. However, if compression is used, the 6LP-GW will be responsible for compressing and decompressing packets as required.

Some of the new features 6LoWPAN-ND introduces are defined in [I-D.ietf-6lowpan-nd] as optional. The treatment of such optional features is also considered optional in this document, but the implementation of such features would be required if they are implemented and used in the 6LoWPAN network.

This memo assumes the presence of (at least) one IPv6 router (RR) having (at least) one IPv6 address. The 6LP-GW is assumed to keep track of the IPv6 address(es) of the RR(s) in order to properly receive packets directed to the RR(s) or generate packets on behalf of the RR(s). However, how to carry out this RR-address tracking is beyond the scope of this document (although some advice is provided in section 2.6.2).

1.3. Application scenario

The scenario proposed here assumes that the 6LP-GW is placed at some point between a RR and a 6LoWPAN network. The 6LP-GW extends the RR’s functionality by adding an IEEE 802.15.4 interface in addition to the RR’s existing interfaces (typically IEEE 803.3 and IEEE 802.11). This added interface, together with the forwarding and proxy mechanisms logically turns the RR into a 6LBR, enabling 6LoWPAN devices to share the same network segment as any other FFD, without requiring any further special treatment. Therefore the 6LP-GW MUST perform all the operations necessary to enable ordinary IPv6 hosts to communicate with 6LoWPAN hosts and vice versa. Figure 1 illustrates

a possible application scenario.

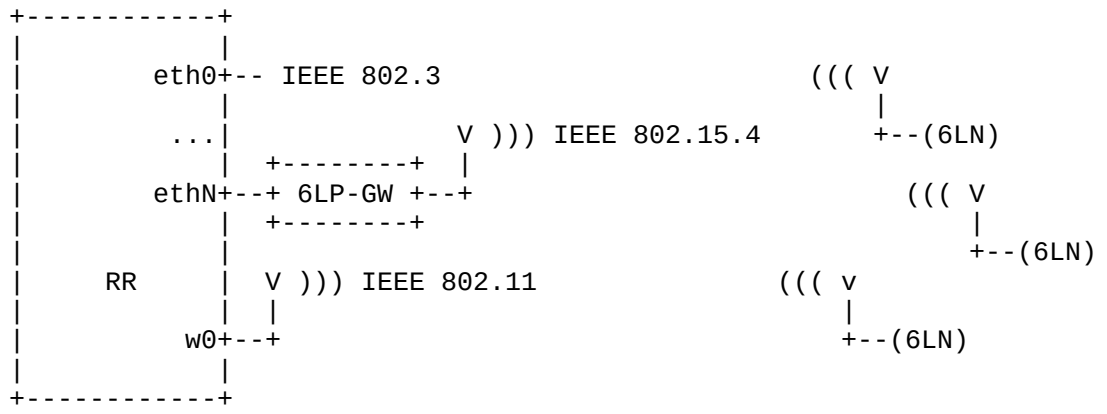


Figure 1: 6LP-GW Application Scenario

2. 6LP-GW Operations

The operations described in this section are considered the minimum required for correct behavior of the 6LP-GW (and thus, the whole network segment). However, section 3 explains some optional features that may be of interest for particular applications.

2.1. 6LP-GW Operational Overview

To enable the operations described above ND packets reaching the 6LP-GW MUST be processed, modified, and, in some cases, hijacked.

The 6LP-GW together with the RR will be seen from the 6LoWPAN side as a 6LBR. In contrast, from the IPv6 side, it is impossible to distinguish between FFDs and 6LoWPAN nodes.

Regarding forwarding, an appropriate MAC translation mechanism MUST be applied when required, since IEEE 802.15.4 MAC addresses are 64 bits long while IEEE 802.3 MAC addresses are 48 bits long. Note that the 6LP-GW is invisible for any of the link segments it internetworks, meaning that it does not need to have either an IPv6 address or a MAC address. Of course the device MAY have an IPv6 address and MAC address associated with an interface, this could be used for management of the device (including loading new software into it), but this functionality lies outside the scope of this document. Additionally, decompression and compression are performed by the 6LP-GW on incoming and outgoing packets when appropriate.

Note also that the proxy mechanisms described below consider only the

case of packets traversing from one segment to the other. The 6LP-GW SHOULD NOT forward unicast packets directed to the same segment they came from, unless the optional routing feature described in section 3.2.3 is implemented.

In addition, some ND options (mainly SLLAO and TLLAO) will require extra processing in order to translate from 48-bit MAC addresses into 64-bit MAC addresses and vice-versa, depending on which segment the packets containing these options originate from. This translation MUST occur whenever a ND option contains a link-layer address.

In all cases, validity checks of the incoming ND messages SHOULD be performed as specified in the corresponding ND document or draft. The specific way to process a ND message will depend on which segment it originates from.

2.2. 6LP-GW Initialization

In addition to the data structures required for the chosen forwarding mechanism, the 6LP-GW MUST maintain a Neighbor Cache (NC) just as if it were a 6LR (or 6LBR). The maintenance procedures for this cache extend those described in [I-D.ietf-6lowpan-nd]. This means that the 6LP-GW MUST create/refresh entries when receiving Neighbor Solicitation messages (NS) and it MUST also remove Neighbor Cache Entries (NCEs) when the registration lifetime expires. Receiving an ARO with zero lifetime will cause the 6LP-GW to immediately delete the corresponding NCE.

In addition, every NCE MUST contain an ARO-pending flag and a Duplicate Address Detection (DAD) timer, whose meanings will be explained later in this section. In case context-based header compression is used [I-D.ietf-6lowpan-hc], the 6LP-GW SHOULD also perform context information maintenance and dissemination just as if it were a 6LBR. At bootstrapping, the 6LP-GW initializes all the data structures needed to create and maintain both NC and Context information. Note that an implementation MAY merge together the chosen forwarding mechanism and NC maintenance.

2.3. Processing Neighbor Solicitation Messages

Neighbor Solicitation messages that reach the 6LP-GW may have originated for different purposes. The appropriate way to process them depends on this purpose and it will differ depending on their origin and their structure. We will consider those originating in the IEEE 802.15.4 segment in section 2.3.1 and those originating in the IEEE 802.3 segment in section 2.3.2.

2.3.1. NS Messages Originating in IEEE 802.15.4 Segment

The processing of the three different types of NS messages that can arrive at the 6LP-GW from the IEEE 802.15.4 segment is defined in the paragraphs below.

2.3.1.1. Multicast NS

A multicast NS can only be sent for the purpose of address resolution. While 6LoWPAN hosts (6LHs) do not perform it, 6Rs MAY do address resolution and therefore this type of NS SHOULD be forwarded unchanged to the IEEE 802.3 interface (with the appropriate MAC translation).

2.3.1.2. Unicast NS not Containing an ARO Option

As defined in [RFC4861], unicast NSs without ARO are sent as probes to test for reachability. Considering that 6LHs do not maintain NCEs for other hosts, but only for 6Rs, it is unlikely that any 6LH sends a unicast NS to any node other than a 6R. However, nothing in [I-D.ietf-6lowpan-nd] precludes 6Rs from sending this type of message to any kind of node. Therefore, a unicast NS message not containing an ARO option MUST be forwarded to the IEEE 802.3 interface unchanged (apart from the appropriate MAC translation) so that it can be processed and responded to by the recipient as defined in [RFC4861] (if the originator of the NS was a 6LH, the recipient will be the RR, if it was a 6R, the recipient could be any node).

2.3.1.3. Unicast NS Containing an ARO Option

Unicast NS messages containing an ARO option are sent as part of the registration procedure. As these messages are only sent to 6Rs, and the 6LP-GW together with the RR is seen as a 6LBR, it is likely that the 6LP-GW will receive such messages having as their destination IPv6 address the RR’s IPv6 address.

Therefore, the 6LP-GW MUST perform the normal operations of a 6R when receiving this type of messages directed to the RR, i.e., the NS message MUST be processed as specified in section 6.5 of [I-D.ietf-6lowpan-nd] in terms of validity checking and NC maintenance, but with some differences as will be explained below. If the IPv6 destination address of the NS message including an ARO does not match the RR’s IPv6 address, then the packet MUST be discarded. If, for some reason, the RR’s IPv6 address is unknown when the NS arrives, then the packet MUST also be discarded.

At this point it is important to note two issues that can impact design decisions:

- Unicast NS messages sent for address registration also have the purpose of performing reachability detection (generally referred to as network unreachability detection-NUD) to determine the reachability of the router [I-D.ietf-6lowpan-nd].

- As the 48-bit EUI space is a subset of the 64-bit EUI space, the 64-to-48 bit MAC address mapping can lead to duplicate addresses. Therefore, the 6LP-GW MUST perform DAD on behalf of 6LoWPAN nodes when registering addresses.

For the above reasons, the 6LP-GW MUST perform not only the registration procedure, but also perform both DAD (in the IPv6-ND way on the IEEE 802.3 interface) and NUD; with both DAD and NUD performed on behalf of the 6LoWPAN node that is trying to register its address.

In order to perform DAD, the 6LP-GW MUST send a NS, formatted as explained in [RFC4862]. The target address of this NS will be the address being registered and the destination address will be the Solicited-node multicast address of this address. For NUD, the NS message originated in the IEEE 802.15.4 segment MUST be forwarded to the IEEE 802.3 segment, so a subsequent Neighbor Advertisement message (NA) response confirms the reachability of the router.

However, DAD is an expensive process as waiting for messages that are not going to receive responses can only be terminated by a timeout [VATMAG98] and it can not be performed in parallel with NUD due to the risk of duplicate addresses. As waiting for both to complete sequentially may delay the autoconfiguration process too much, it is RECOMMENDED to perform DAD only upon registration and then, NUD upon re-registration. Figure 3 describes the registration procedure and section 2.3.1.4 details the DAD process.

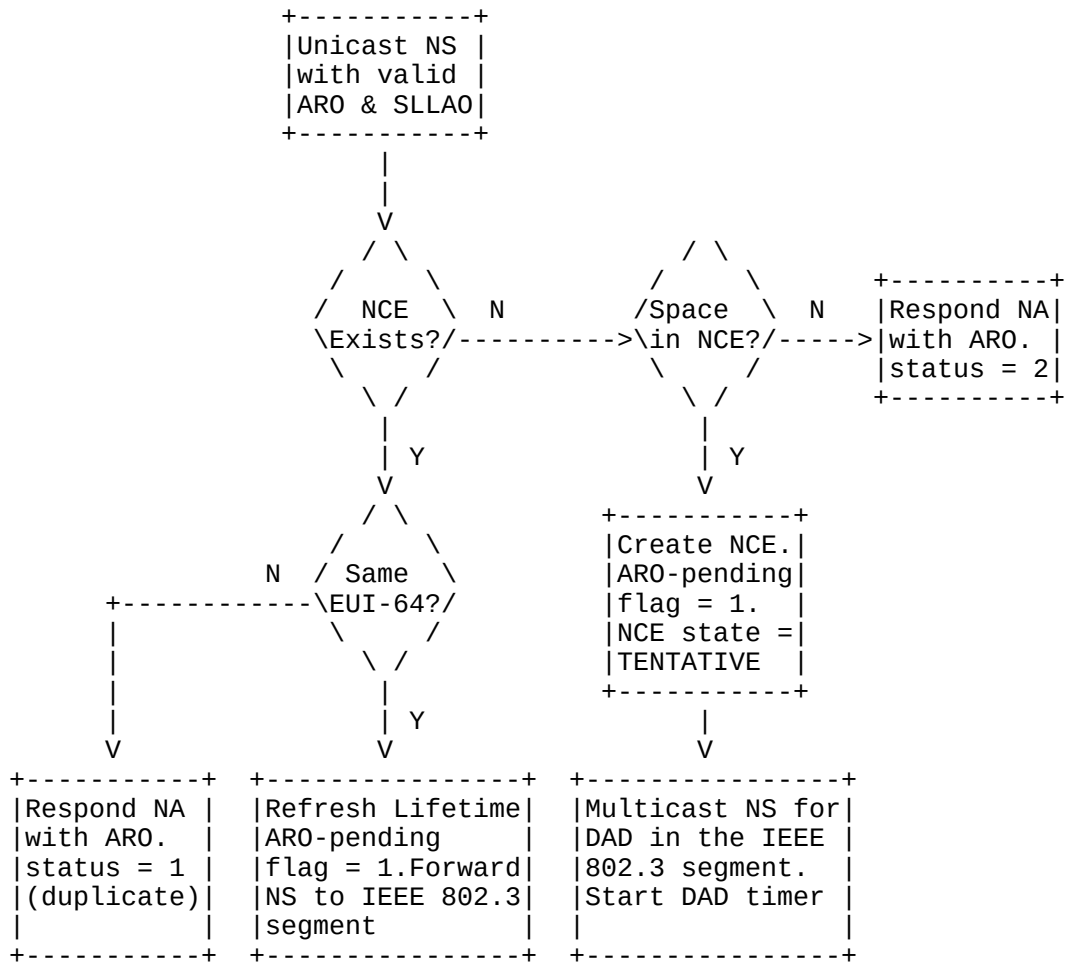


Figure 2: NS with ARO Processing

Optionally, it is possible to speed-up the registration procedure by performing DAD upon reception of the first Router Solicitation (RS) message (RS-triggered DAD), instead of delaying it until receiving a NS. This feature adds some complexity as will be explained in section 3.2.1.

Considering all of the above, upon receipt of a NS message containing valid ARO and a SLLAO options, the 6LP-GW MUST behave as follows (see figure 3).

The 6LP-GW searches its NC for a NCE with same IPv6 address as the IPv6 source address of the incoming NS message; if no matching NCE is

found, then the 6LP-GW creates a new NCE for the node being registered. If there is no space left in the NC, then the registration fails and the 6LP-GW generates a NA including an ARO with status = 2, as specified in section 6.5.2 of [I-D.ietf-6lowpan-nd]. If there is space available in the NC, then a new entry is created with a state value of TENTATIVE and the ARO-pending flag is set to 1. In this final case a NA is not generated in response, but rather the 6LP-GW performs DAD on the IEEE 802.3 segment on behalf of the node that is trying to register, similar to section 5.4 of [RFC4862]. This DAD process is detailed in the next section (2.3.1.4).

If there is a matching NCE, then if the NCE MAC address encoded in EUI-64 differs from the MAC address present in the EUI-64 field of the ARO, the address is a duplicate and the 6LP-GW must generate and send a NA message including an ARO with status = 1 (duplicate), as specified in section 6.5.2 of [I-D.ietf-6lowpan-nd}. If the EUI-64 is the same as present in the ARO and the NCE state is REGISTERED, then this is the case of a re-registration and, therefore, the ARO-pending flag must be set to 1, the registration lifetime must be refreshed, and the received NS message MUST be forwarded to the IEEE 802.3 segment in order to perform NUD (note that the NA message produced in response will be intercepted later - see section 2.4.2). A special case could happen if a 6LN whose registration is in progress (i.e., it has sent a NS with an ARO but the DAD procedure has not completed, hence no NA has been sent in response yet) sends a second NS with ARO. This situation happens if there is a matching NCE, with the same EUI-64, but the NCE state is TENTATIVE. In this case the 6LP-GW SHOULD simply discard the NS with ARO.

Note that in some cases of the above procedure, the 6LP-GW responds with NAs on behalf of the RR. Therefore, for every packet being generated in the 6LP-GW in this situation, the Router flag MUST be 1 and the IPv6 source address MUST be the IPv6 address of the RR attached to the 6LP-GW. This address already should be known due to the previous RS and RA exchange.

It is also important to note that TENTATIVE entries MUST be timed out TENTATIVE_NCE_LIFETIME seconds after their creation in order to leave space in the NC for other hosts, as specified in [I-D.ietf-6lowpan-nd].

2.3.1.4. Performing DAD on Behalf of 6LoWPAN Nodes

DAD SHOULD be performed as specified in [RFC4862] and this specification assumes the existence of the variables RetransTimer (defined in [RFC4861]) and DupAddrDetectTransmits (defined in [RFC4862]). Figure 3 illustrates the DAD operation described in this section.

The 6LP-GW MUST maintain a DAD timer for each NCE in the NC. A DAD timer will be started when the corresponding NS is sent. If no NA is received before the expiration of RetransTimer, then the 6LP-GW will either send another NS or end the DAD process, depending on the value of DupAddrDetectTransmits.

If the DAD process completes successfully (i.e., no duplicate instance of this address is detected), then the 6LP-GW changes the state of the corresponding NCE to REGISTERED, and sets the ARO-pending flag to 0. In addition, the information contained in the NCE will be used to generate and send a NA message with an ARO option with status = 0 (success).

If DAD failed, then a NA with ARO option and status = 1 (duplicate) MUST be generated and sent to the node (in the IEEE 802.15.4 segment) that originated the registration request. This message MUST be sent as specified in section 6.5.2 of [I-D.ietf-6lowpan-nd]. After sending this message, the NCE can be deleted. Note that implementations MAY mark the NCE for deletion and then delete it in next stage in order to avoid race conditions instead of deleting it immediately.

Note also that, when using the (optional) RS-triggered DAD, neither the response will be sent or the NCE will be deleted when DAD completes unless a NS with an ARO option arrives at the 6LP-GW. Section 3.2.1 will explain this behavior in detail.

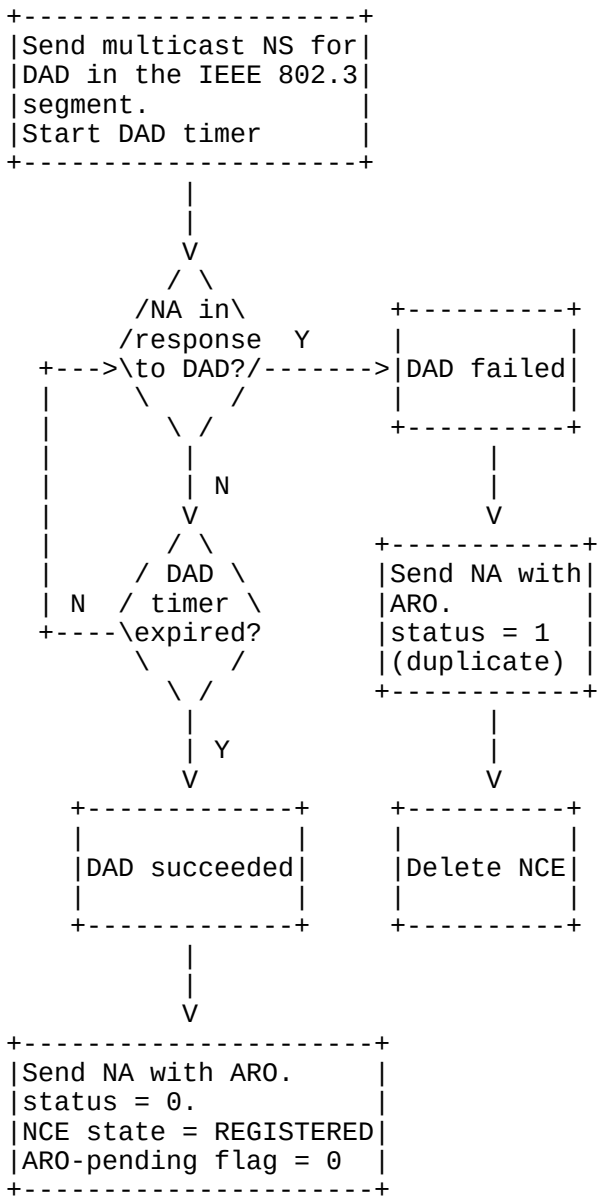


Figure 3: DAD performed on behalf of 6LoWPAN nodes. The diagram illustrates the process assuming DupAddrDetectTransmits = 1.

2.3.2. NS Messages Originating in IEEE 802.3 Segment

As stated in [RFC4861] and [RFC4862], two different types of NS may arrive to the IEEE 802.3 interface: unicast and multicast NSs. These messages can be sent with three different purposes: Address Resolution, NUD, or DAD.

For Address Resolution or DAD, the NS messages are sent to the Solicited-node multicast address, while for NUD, the NS messages are unicast.

6LoWPAN hosts (6LHs) should respond to the unicast NS messages, but as they do not join the Solicited-node multicast group, hence do not listen to the Solicited-node multicast address and, therefore, they will not respond to these multicast NS messages. In contrast, 6LoWPAN Routers (6Rs) do join the Solicited-node multicast group (hence they listen to the Solicited-node multicast address) and thus they should do respond to both, unicast and multicast NS messages.

Note here that the 6LP-GW is aware of every node that is currently reachable in the IEEE 802.15.4 segment due to the registration process.

2.3.2.1. Unicast NS

Unicast NS messages are sent for NUD. These messages MAY be forwarded unchanged (except for the appropriate MAC translation) to the IEEE 802.15.4 segment in order that the target nodes could respond to the NS with a NA. However, as 6LoWPAN nodes are registered with the 6LP-GW, the information contained in its NC is a priori sufficient to generate the response on behalf of the target nodes. Thus, the 6LP-GW SHOULD generate this response so that the wireless nodes save energy as they neither need to receive nor send the NS and NA messages, respectively. It is important to note that 6LoWPAN nodes only register non-link-local addresses with routers; thus, if the incoming NS’s destination address is a link-local address, the 6LP-GW SHOULD generate a link-local (EUI-64-based) IPv6 address for every different EUI-64 address stored in its NC in order to compare it against the destination address of the incoming NS prior to send the corresponding response.

2.3.2.2. Multicast NS

Multicast NSs are sent to perform Address Resolution or DAD. These messages invoke responses from 6Rs, but not from 6LHs (since as noted above 6LHs do not join the Solicited-node multicast group). The 6LP-GW may be aware of which entries in its NC correspond to 6Rs due to previously intercepted NA (Router flag) or RA messages and thus, the 6LP-GW MAY forward these multicast NS messages only to these 6Rs (and behave as specified below for hosts). However, this approach is unreliable since it requires that all 6R nodes have previously sent at least one NA or RA message.

As result a similar approach as used for unicast NS messages is RECOMMENDED, i.e., for every multicast NS message arriving on the IEEE 802.3 interface, the 6LP-GW SHOULD generate an appropriate NA in response (if required), according to the contents of its NC, as follows.

If the target address present in the NS is not a link-local address, then the 6LP-GW MUST search for it in the NC. If the target address is a link-local address, then a link-local (EUI-64 based) address MUST be generated for each NCE present in the NC (according to the link layer address present in the NCEs, according to [RFC4944] and compared to the target address of the incoming NS (since link-local addresses are not registered with 6Rs).

If there is a matching NCE whose state is REGISTERED (regardless of the nature of the target address), then the 6LP-GW MUST generate a NA message in response and send it through the IEEE 802.3 interface. Such an NA message is generated based on the contents of the corresponding NCE, mainly as specified in section 7.2.4 of [RFC4861]:

- If the source address of the NS is the unspecified address, then the destination IPv6 address MUST be the all-nodes multicast address. Otherwise, the destination address MUST be the source address of the NS.

- The target address of the NA MUST be copied from the NS message.

- The Router flag SHOULD be set to the NCE’s isRouter flag value unless the optional feature described in section 3.2.4 is implemented (in that case the isRouter flag will always be zero).

- If the source of the NS is the unspecified address, the node MUST set the Solicited flag to zero. Otherwise the Solicited flag MUST be set to 1.

- o The Override flag SHOULD be set to 1, as RECOMMENDED for this case in section 7.2.4 of [RFC4861].

- o The NA MUST include a TLLAO containing the link-layer address of the node in the NCE, since the 6LP-GW is sending this message "on behalf" of a node receiving a multicast NS message.

If there is a matching NCE whose state is TENTATIVE, then the 6LP-GW MUST NOT respond with a NA [RFC4862] section 5.4.3. If the source IPV6 address of the NS is not the unspecified address, then the NS has been sent for address resolution and SHOULD be discarded. If the source address of the NS is the unspecified address, then the sender is trying to configure a duplicate address. If no NS for DAD has been sent yet on behalf of the corresponding 6LN, the 6LP-GW SHOULD generate a NA including an ARO option with status value = 1, send it to the corresponding 6LN whose address is in the NCE, and delete the NCE.

2.4. Processing Neighbor Advertisement Messages

NA messages are typically generated in response to NS messages. This section specifies the required processing of the different types of NA messages that may arrive at the 6LP-GW.

2.4.1. NA Messages Originating in IEEE 802.15.4 Segment

NA messages may arrive on the IEEE 802.15.4 interface for different purposes. Regardless of the nature of the incoming NA, the 6LP-GW SHOULD update the isRouter flag in the NCE matching the source address of this NA message and forward the NA message to the IEEE 802.3 interface.

Note that if the recommendation proposed in section 2.3.2.2 is followed, then 6LoWPAN nodes will be unlikely to send NA messages in response to NS messages coming from the IEEE 802.3 segment (as the 6LP-GW will not have forwarded the NS to them, hence they do not need to respond).

2.4.2. NA Messages Originating in IEEE 802.3 Segment

NA messages originating in the IEEE 802.3 segment can arrive at the 6LP-GW for different reasons, depending on whether they are unicast or multicast.

2.4.2.1. Unicast NA

A unicast NA message could only be originated as result of a unicast NS message sent from the IEEE 802.15.4 segment. The original NS message could be either a probe sent for reachability confirmation or part of the registration process.

In order to distinguish between these two cases, if the source IPv6 address of the NA is the IPv6 address of the RR, then the 6LP-GW searches its NC for a NCE matching the destination address of the NA. If a matching entry having the ARO-pending flag set to 1 is found, then the incoming NA is the final part of the registration process of a node. In this case, the ARO-pending flag MUST be set to 0 and the NA forwarded (with the appropriate MAC translation) to the IEEE 802.15.4 interface. If the NCE state is other than REGISTERED, it MUST be set to REGISTERED. In addition, an ARO option containing a status value of 0 MUST be appended to the NA before forwarding it to the IEEE 802.15.4 interface.

If the IPv6 source address of the NA is other than the IPv6 address of the RR, or if the NCE matching the destination IPv6 address of the NA has its ARO-pending flag set to 0, then the NA message SHOULD be forwarded unchanged (except for the appropriate MAC translation) since it is responding to a NS sent for NUD.

If no NCE matching the search criteria is found, the message SHOULD be discarded.

2.4.2.2. Multicast NA

Multicast NA messages that may arrive to the 6LP-GW on the IEEE 802.3 interface originated either for quick information propagation (if the link-layer address of the sender changed) or as a response to a NS sent for DAD (meaning that there is a duplicate).

If the target address in the NA message corresponds to a NCE whose state is TENTATIVE, then DAD failed for that NCE. In this case, the 6LP-GW will generate a new NA on behalf of the RR it is attached to, with an ARO option containing a status value of 1. This packet MUST be sent on the IEEE 802.15.4 interface according to [I-D.ietf-6lowpan-nd] section 6.5.2.

If there is no TENTATIVE NCE whose IPv6 address matches the target, then this packet was sent for quick information propagation and it SHOULD be discarded due to its minor importance for 6LoWPAN nodes. However, implementations MAY forward this type of message unchanged (except for the appropriate MAC translation) so that 6LNs can update their NCs quickly.

2.5. Processing RS Messages

In IPv6-ND, RS messages are sent during bootstrapping and they are mainly multicast; hence the IPv6 source address MAY be the unspecified address and the SLLAO SHOULD be included when the IPv6 source address is not the unspecified address as specified in [RFC4861]. In contrast, [I-D.ietf-6lowpan-nd] specifies that routers are not required to send periodic RA messages, therefore, hosts will send RS messages in order to maintain their prefixes and registration lifetimes; these RS will be unicast (unless the link-layer address of the router is not known) and MUST include the SLLAO option.

Note that the whole network bootstrapping process can be optimized by extending the processing of RS messages as proposed in section 3.2.1.

2.5.1. RS Originating in IEEE 802.15.4 Segment

RS messages originating in the IEEE 802.15.4 segment may be unicast or multicast, but they MUST include a SLLAO option so that the router can always respond with a unicast RA. If for any reason a RS message arriving at the 6LP-GW does not include a SLLAO option, that message MUST be discarded. As, RRs on the IEEE 802.3 segment can process these messages without requiring any special treatment from the 6LP-GW, RS messages from the IEEE 802.15.4 segment MUST forwarded unchanged (except for the appropriate MAC translation).

2.5.2. RS Originating in IEEE 802.3 Segment

RS messages coming from the IEEE 802.3 segment will be silently discarded if the source address is the unspecified address. If no SLLAO option is included, then the 6LP-PG will modify the RS by appending a SLLAO option with the link-layer address of the originator of the RS prior to forward it to the IEEE 802.15.4 segment. Note that the ICMPv6 checksum will need to be recalculated.

Section 3.2.4 describes an alternative treatment of these messages that may be of interest for certain applications.

2.6. Processing RA Messages

If optional 6LoWPAN features such as the use of the ABRO or 6CO options are enabled, then the treatment of RA messages is as detailed in section 3.1, otherwise, the processing of RAs is performed as follows.

2.6.1. RA Messages Originating in IEEE 802.15.4 Segment

RA messages originating in the IEEE 802.15.4 segment MAY be forwarded untouched (except for the appropriate MAC translation).

2.6.2. RA Messages Originating in IEEE 802.3 Segment

As seen in previous sections, there are several situations where the 6LP-GW needs to send messages on behalf of the RR. As RAs originating in the IEEE 802.3 segment are necessarily sent by the RR, the 6LP-GW SHOULD save both, the source MAC address and the source IPv6 address of such RA messages for subsequent use. Note here that the RR may have more than one IPv6 address and/or MAC address and therefore, when saving these RR's addresses the 6LP-GW SHOULD perform some out-of-scope management on such addresses in order to ensure both, that the addresses do not reach an obsolete state, and that they are not unnecessarily overwritten when RAs are sent from a different source address.

Apart from performing the proper management of source MAC and IPv6 addresses, RAs originating in the IEEE 802.3 segment SHOULD be forwarded to the IEEE 802.15.4 interface, with the following considerations:

- [RFC4861] states that a RR MAY unicast solicited RAs when the corresponding RS's source address is not the unspecified address, but the usual case is to multicast these RAs to the all-nodes multicast address. While no harm can result from forwarding these multicast messages (i.e., they will reach their destination and accomplish their purpose), one of the main goals of [I-D.ietf-6lowpan-nd] is to reduce multicast traffic. Therefore implementations MAY choose to add a flag to the NCEs (awaiting-RA) in order to mark NCEs soliciting RAs (upon receiving RSs from those 6LNs, and prior to forward these RSs, as specified in section 2.5.1) so that the 6LP-GW can change the destination address of RAs to the address of the 6LN in the marked NCE. Since RRs MUST delay the sending of RAs and they MAY send one RA to respond to several RSs [RFC4861], if more than one 6LN is marked as "awaiting-RA", then the 6LP-GW MUST send a unicast RA per marked NCE, clearing the NCEs' "awaiting-RA" flag when dispatching the corresponding RA.

- 6Rs only need to send periodic RAs if they choose to distribute prefix and/or context information across a route-over topology (see section 3.1.1). In contrast, RRs will send unsolicited periodic RAs as specified in [RFC4861]. If optional context or prefix distribution is performed, these RA messages SHOULD be forwarded (applying the corresponding processing, as mentioned

above and/or in section 3.1) to the IEEE 802.15.4 segment. If no optional context or prefix distribution is performed, then an implementation MAY filter out these periodic RA messages (for example, using the "awaiting-RA" flag in NCEs, so that the RA is discarded if no NCE with this flag set is present in the NC).

- o According to [I-D.ietf-6lowpan-nd], the SLLAO option MUST be included in the RA message. However, this option MAY be omitted in RAs sent by RRs, as stated in [RFC4861], therefore the 6LP-GW MUST include this option in RA messages originating in the IEEE 802.3 segment that do not contain a SLLAO option, prior to forward them to the IEEE 802.15.4 segment.

- o All prefixes other than the link-local (FE80::) prefix are assumed to be off-link in 6LoWPAN-ND and 6LoWPAN hosts will ignore any PIO option whose 'L' (on-link) flag is set ([I-D.ietf-6lowpan-nd] sections 5.4 and 5.7 respectively). However, RRs usually advertise global prefixes with the 'L' flag set in the PIO option within the local network. Considering that the PIO option in RA messages is the only way hosts have to acquire a global address (in the absence of other mechanisms such as DHCPv6 for address configuration) and that the same prefix which is considered to be on-link in the IEEE 802.3 segment is assumed to be off-link in the IEEE 802.15.4 segment, the 6LP-GW MUST always clear this 'L' flag in the PIO option for every RA originating in the IEEE 802.3 segment and directed to the IEEE 802.15.4 segment.

Note that inclusion of a new SLLAO option or the modification of the PIO option necessitates recomputation of ICMPv6 checksum.

2.7. Processing a Redirect

According to [I-D.ietf-6lowpan-nd], redirects are not used by 6LoWPAN-ND in route-over topologies, but they MAY be used in mesh-under topologies. Therefore, these messages SHOULD be forwarded unchanged, regardless of the incoming interface. Some implementations MAY provide mechanisms for manual configuration allowing the user to disable the forwarding of Redirect messages in route-over topologies.

3. Optional Features

3.1. Processing of Optional 6LoWPAN-ND Features

This section describes the operations required to process the features defined as "optional" in [I-D.ietf-6lowpan-nd].

3.1.1. Multihop Prefix and Context Distribution

Despite the 6CO and ABRO options being described as optional features, their processing by the 6LP-GW is RECOMMENDED. While 6CO provides for context-based compression ([I-D.ietf-6lowpan-hc]), ABRO is mandatory for this context dissemination across route-over topologies.

Both options SHOULD be included by the 6LP-GW (one ABRO and one or more 6COs) in RA messages addressed to the IEEE 802.15.4 segment. In order to do so, the 6LP-GW MUST behave similarly to the description in [I-D.ietf-6lowpan-nd] sections 7 and 8, with the following considerations:

- The 6LP-GW is in charge of performing all the tasks related to context information maintenance, as if it were a 6LBR, as per [I-D.ietf-6lowpan-nd] Section 7.2.

- The 6LBR Address in the ABRO option is the address of the RR that the 6LP-GW is attached to.

- The 6LP-GW MUST maintain the ABRO version number in stable storage and be aware not only of its own 6CO options, but also of the changes regarding the PIO in the RAs coming from the RR.

3.1.2. Multihop DAD

6LoWPAN-ND only requires DAD if non-EUI-64 based addresses are used. Multihop DAD in 6LOWPAN-ND is required when non-EUI-64 based addresses are used in a route over topology and more than one 6R is present in the network. Since in our case both 64 and 48-bit MAC addresses are mixed in the same network, the 6LP-GW MUST always perform DAD on behalf of 6LNs (even if 6LNs only use EUI-64 based addresses). For this reason, 6LP-GW implementations are RECOMMENDED to provide mechanisms for supporting multihop DAD in route-over topologies.

Multihop DAD is performed via two new messages described in [I-D.ietf-6lowpan-nd], Duplicate Address Request (DAR) and Duplicate Address Confirmation (DAC). 6LRs send DARs to the 6LBR when they receive a NS with an ARO option from hosts. The 6LBR responds with a DAC after searching for the address contained in the DAR in its NC. The DAC message contains a status field whose value indicates whether the address is a duplicate or not. Finally, the 6LR responds with a NA with an ARO option (whose status value depends on the status value of the received DAC) to the host which sent the original NS with an ARO.

In our case, the 6LP-GW SHOULD act as a 6LBR, processing the DAR as specified in [I-D.ietf-6lowpan-nd] section 8.2.1 and responding with a DAC as specified in [I-D.ietf-6lowpan-nd] section 8.2.4. However, the 6LP-GW will not only compare the IPv6 address contained in the DAR message against entries in its NC, but will also have to perform a traditional DAD on the IEEE 802.3 interface to ensure the uniqueness of the address in the complete network.

Therefore, the 6LP-GW SHOULD maintain the information in the DAR message until DAD has finished. The RECOMMENDED way to implement this is by creating a TENTATIVE NCE (just as done upon the arrival of a NS with an ARO). Note that in this case, the ARO-pending flag in the NCE MUST NOT be set to 1, since neither a NS with an ARO has been received nor a NA with ARO needs to be sent in response. Instead, another flag, for example DAR-pending, SHOULD be added to each NCE in order to process that entry properly (i.e., sending a DAC) once DAD has completed.

Note that all of the above is compatible with the operations regarding regular address registration (section 2.3.1.3) and even with the RS-triggered DAD Optimization described in section 3.2.1. The reason is that Multihop DAD is only performed when a host tries to register with a 6LR, while the regular registration process occurs whenever a host tries to register directly with the 6LBR.

3.2. Optional 6LP-GW Operation and Optimizations

This section suggests some optional optimizations for the operations proposed in this document that may be of interest for certain implementations.

3.2.1. RS-triggered DAD Optimization

As mentioned in previous sections, it is possible to speed up the bootstrapping procedure by triggering DAD on behalf of 6LoWPAN nodes upon reception of the first RS message, instead of waiting until the reception of a NS with an ARO option (i.e., the message actually sent for DAD). Considering the long delay required for DAD, the implementation of this feature is RECOMMENDED.

Implementing this feature requires a few modifications to the operation described in sections 2.3.1.3, 2.3.1.4, and 2.5.1, regarding the processing of NS messages with an ARO option, the way DAD is performed on behalf of 6LoWPAN nodes, and the processing of RS messages originated in the IEEE 802.15.4 segment respectively. These modifications are described below.

In addition, a new NCE state, DUPLICATE, is required in order to mark

NCEs whose IPv6 address is not unique but that can not be deleted yet (i.e., they are still required for specific reasons).

3.2.1.1. Changes to RS processing

If this optimization is implemented, then the 6LP-GW will try to create a NCE upon arrival of the first RS message (originating in the IEEE 802.15.4 segment) and then it will send a multicast NS to perform DAD, as specified in [RFC4862]. This newly created NCE must have its ARO-pending flag set to 0 and its state set to TENTATIVE. If there is no space left in the NC or if the NCE already exists, then the 6LP-GW should omit this step (i.e., neither try to create the NCE nor send the NS for DAD). Note that this extra processing MUST be performed in addition to forwarding the incoming RS to the IEEE 802.3 segment (i.e., first, forwarding the RS message, then creating the NCE and performing DAD, if possible).

Note that no delay is needed before sending the NS for DAD after reception of the RS message, since nodes will wait a random amount of time between 0 and MAX_RTR_SOLICITATION_DELAY before sending their first RS [RFC4861].

3.2.1.2. Changes to Unicast NS with ARO processing

When receiving a NS with an ARO option, it is highly probable that the NCE already exists (with the same EUI-64 as the one present in the ARO EUI-64 field). In that case, the 6LP-GW SHOULD check the NCE state; if the NCE state is REGISTERED, this is the case of a re-registration and thus, proceed as in section 2.3.1.3 for the same case. In case the state is TENTATIVE, that means that either DAD has not completed yet, or it has completed successfully (as otherwise the state would be DUPLICATE instead of TENTATIVE). If DAD has not completed yet, then the 6LP-GW will simply set the ARO-pending flag to 1. If DAD has completed successfully, the 6LP-GW will refresh the NCE Lifetime, set the ARO-pending flag to 1 and forward the NS message to the IEEE 802.15.4 segment. In case the NCE state is DUPLICATE, a NA with an ARO option having status = 1 MUST be sent to the 6LN that originated the registration. After that, the NCE SHOULD be deleted (or marked for deletion).

If the NCE does not exist when a NS with an ARO option arrives, or if it exists but with a different EUI-64 than the present in the ARO option, the 6LP-GW MUST behave as described in section 2.3.1.3 (figure 2) for those cases.

3.2.1.3. Changes to DAD on behalf of 6LoWPAN nodes

DAD is performed as described in section 2.3.1.4, with the only differences taking place right after its completion. If the DAD process completes successfully, then the 6LP-GW MUST check the ARO-pending flag. If set, it MUST be cleared and a NA with an ARO having its status field set to 0 MUST be sent to the 6LN that originated the registration. If the ARO-pending flag is not set, the 6LP-GW MUST perform no change to the NCE until a NS with ARO arrives.

In case DAD fails, if the ARO-pending flag is not set, the 6LP-GW MUST change the NCE state to DUPLICATE. In contrast, if the ARO-pending flag was set, the 6LP-GW MUST send a NA with ARO having its status field set to 1 (duplicate). After that the NCE SHOULD be deleted (or marked for deletion).

3.2.2. ICMPv6 option filtering

Some of the options contained mainly in RA messages could be considered irrelevant for 6LoWPAN networks. Therefore, it may be of interest for implementations to filter out these options, reducing the packet size and therefore reducing power consumption by the IEEE 802.15.4 nodes for their delivery and processing. Examples of options that can be filtered are the Recursive DNS Server Option (defined in [RFC6106]) or the Flags Expansion Option (defined in [RFC5175]). Note that SLLAO, MTU, and PIO options MUST NOT be filtered out. The RECOMMENDED way to perform this filtering would be to simply remove all the options in RA messages originating in the IEEE 802.3 segment that are to be forwarded to the IEEE 802.15.4 except for the following options: SLLAO, MTU, and PIO. However, the list of options that are filtered may differ between implementations; it may even be desirable to allow this list to be manually configured.

It is also possible to filter out irrelevant options of messages originating in the IEEE 802.15.4 segment and directed to the IEEE 802.3 interface, such as the ARO, 6CO, and ABRO. However, this filtering is of minor interest since the receiving and processing power saving would occur in a device that is likely to be plugged into the power mains.

3.2.3. 6LoWPAN-side Routing

An interesting feature for route-over topologies is to provide some sort of routing protocol such as RPL ([I-D.ietf-roll-rpl]) on the IEEE 802.15.4 side of the 6LP-GW. Although this feature is not necessary for the correctness of the protocol (the 6LP-GW could simply rely on 6LRs), it will reduce the traffic, hence enhancing the performance of the network.

Implementations MAY advantage of the routing protocol by letting the 6LP-GW "pretend" that it is the 6LBR or even provide an IPv6 address to the IEEE 802.15.4 interface. However, specific details regarding implementation of this feature are beyond the scope of this document.

It is important to note that this feature is also useful to prevent loops in the network and, unless the feature is used, there SHOULD be some out-of-scope mechanism provided for this purpose, such as for example the use of the Spanning Tree Protocol (STP) [BRIDGES] for layer-2 forwarding.

3.2.4. 6LRs seen as hosts by FFDs

6LRs will never be the first hop for FFDs since the RR will be always in between FFDs and 6LRs. This document tries to equate 6LoWPAN nodes with their homologue FFD nodes. However, some implementations may find the fact that 6LRs are seen as routers by FFDs to be useless. Therefore it is possible for the 6LP-GW to mask the router nature of 6LRs so that they are seen by FFDs as simple hosts. To do so, the 6LP-GW MAY discard RS messages coming from the IEEE 802.3 segment and RA messages originating in the IEEE 802.15.4 segment instead of forwarding them. In addition, the isRouter flag of NA messages coming from or being sent on behalf of 6LNs SHOULD be set to 0 prior to sending them to the IEEE 802.3 segment.

4. Security Considerations

The security considerations of Neighbor Discovery for IPv6 [RFC4861] and Neighbor Discovery Optimization for Low-power and Lossy Networks [I-D.ietf-6lowpan-nd] apply.

When performing DAD on behalf of 6LNs, race conditions involving creating-deleting-NCEs cycles may occur. This problem can be solved by adding a new NCE state (as in section 3.2.1) in order to mark NCEs for deletion instead of deleting them immediately.

5. IANA Considerations

No actions are required from IANA as result of the publication of this document.

6. Contributors

Nicolas Mechin contributed significantly to the production of this document by providing guidance, useful insights, and thorough reviews.

7. Acknowledgments

Thanks to Zach Shelby and Pascal Thubert for their reviews and instrumental advice.

8. References

8.1. Normative References

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

[RFC5942] Singh, H., Beebee, W., and E. Nordmark, "IPv6 Subnet Model: The Relationship between Links and Subnet Prefixes", RFC 5942, July 2010.

[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.

[RFC4919] Kushalnagar, N., Montenegro, G. and C. Schumacher, "IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs): Overview, Assumptions, Problem Statement, and Goals", RFC 4919, August 2007.

[RFC4944] Montenegro, G., Kushalnagar, N., Hui, J., and D. Culler, "Transmission of IPv6 Packets over IEEE 802.15.4 Networks", RFC 4944, September 2007.

[I-D.ietf-6lowpan-nd] Shelby, Z., Chakrabarti, S., and E. Nordmark, "Neighbor Discovery Optimization for Low-power and Lossy Networks", draft-ietf-6lowpan-nd-15, December 2010.

8.2. Informative References

[EUI-64] IEEE Standards Association, "Guidelines for 64-bit Global Identifier (EUI-64) Registration Authority"

[I-D.ietf-6lowpan-hc] Hui, J. and P. Thubert, "Compression Format for IPv6 Datagrams in 6LoWPAN Networks", draft-ietf-6lowpan-hc-13 (work in progress), September 2010.

[I-D.ietf-roll-rpl] Winter, T., Thubert, P., Brandt, A., Clausen, T., Hui, J., Kelsey, R., Levis, P., Pister, K., Struik, R., and J. Vasseur, "RPL: IPv6 Routing Protocol for Low power and Lossy Networks", draft-ietf-roll-rpl-17 (work in progress), December 2010.

[RFC5175] Haberman, B. and R. Hinden, "IPv6 Router Advertisement Flags Option", RFC 5175, March 2008.

[RFC5867] Martocci, J., De Mil, P. and N. Riou, "Building Automation Routing Requirements in Low-Power and Lossy Networks", RFC 5867, June 2010.

[RFC6106] Jeong, J., Park, S., Beloeil, L. and S. Madanapalli, "IPv6 Router Advertisement Options for DNS Configuration", RFC 6106, November 2010.

[VATMAG98] Vatn, J and G. Maguire, "The effect of using co-located care-of addresses on macro handover latency", Fourteenth Nordic Tele-traffic Seminar (NTS 14), August 1998.

[BRIDGES] IEEE Computer Society, "IEEE Standard for Local and metropolitan area networks: Media Access Control (MAC) Bridges", IEEE Std 802.1D (Revision of IEEE Std 802.1D-1998), 2004.

Contributors' Addresses

Nicolas Mechin
Sen.se
10, rue Saint Sebastien
75011 Paris
France

Phone : +33 681 041 721
Email : nicolas@sen.se

Authors' Addresses

Luis Maqueda Ara
Royal Institute of Technology (KTH)
Sen.se
Emmylundsvaegen 5
SE-171 72 Solna
Sweden

Email: lc.maqueda@gmail.com

Gerald Q. Maguire Jr.
School of Information and Communication Technology (ICT)
Royal Institute of Technology (KTH)
Electrum 229
SE-164 40 Kista
Sweden

Email: maguire@kth.se

Individual Submission
Internet-Draft
Intended status: Standards Track
Expires: September 11, 2011

B. Patil, Ed.
T. Savolainen
J. Nieminen
M. Isomaki
Nokia
Z. Shelby
Sensinode

March 10, 2011

Transmission of IPv6 Packets over Bluetooth Low Energy
draft-patil-6lowpan-v6over-btle-01

Abstract

Bluetooth low energy is a low power air interface technology that is defined by the bluetooth SIG. The standard bluetooth radio has been widely implemented and available in mobile phones, notebook computers, audio headsets and many other devices. The low power version of bluetooth is a new specification and enables the use of this air interface with devices such as sensors, smart meters, applicances, etc. There is an added value in the ability to communicate with sensors over IPv6. This document describes how IPv6 is transported over bluetooth low energy using 6LoWPAN techniques.

Status of this Memo

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

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

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

This Internet-Draft will expire on September 11, 2011.

Copyright Notice

Table of Contents

1. Introduction

Bluetooth Low Energy (BT-LE) is a radio technology targeted for devices that operate with coin cell batteries, which means that low power consumption is essential. BT-LE can also be integrated into existing Bluetooth (BT) devices so that devices such as mobile phones and PCs can operate with existing BT accessories as well as BT-LE accessories. An example of a use case for BT-LE accessory is a heart rate monitor that sends data via the mobile phone to a server on the Internet. BT-LE is designed for transferring small amount of data (in most cases less than 10bytes) less frequently (e.g. every 500ms) at modest data rates (e.g. 300kbps). BT-LE enables low cost sensors to send their data over the Internet via a gateway such as a mobile phone. BT-LE is especially attractive technology for Internet of Things applications, such as health monitors, environmental sensing and proximity applications.

Considering the expected explosion in the number of sensors, IPv6 is an ideal protocol due to the large address space it provides. This document describes how IPv6 is used on Bluetooth Low Energy links in a power efficient manner along with efficient application protocols that enable the integration of BT-LE devices into services.

[RFC4944] specifies the transmission of IPv6 over IEEE 802.15.4. The bluetooth low energy link in many respects has similar characteristics to that of IEEE 802.15.4. Many of the mechanisms defined in [RFC4944] can be applied to the transmission of IPv6 on bluetooth low energy links. This document specifies the details of IPv6 transmission over blue-tooth low energy links.

2. Requirements Language

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

2.1. Terminology

Bluetooth Low Energy

Bluetooth low energy is a low power air interface technology specified by the Bluetooth Special Interest Group (SIG).

Bluetooth Network Encapsulation Protocol (BNEP)

Define BNEP.

Gateway

Network element connecting the BT-LE sensors to the Internet. Can be e.g a home gateway or a mobile device.

ND-Proxy

A gateway that operates as a proxy for IPv6 neighbor discovery

CoAP/HTTP Proxy

A gateway that operates as a CoAP/HTTP proxy for the BT-LE sensors. Link local addresses are used between the sensors and the CoAP/HTTP proxy

6to4 prefix

An IPv6 prefix constructed by combining well-known IPv6 prefix with public IPv4 address

6to4/6RD router

A router that has only IPv4 uplink connectivity and thus uses 6to4/6RD prefix in the BT-LE network

3. Bluetooth Low Energy protocol stack

Bluetooth Low Energy is a low power wireless technology developed by the BT-SIG. The lower layer of the BT-LE stack consists of the RF and the Link layer which are implemented in the BT-LE controller. The upper layer consists of the Logical Link Control and Adaptation Protocol (L2CAP), Generic Attribute protocol (GATT) and Generic Attribute profile (GAP) as shown in Figure 1. GATT and BT-LE profiles together enable the creation of applications in a standardized way without using IP. L2CAP provides multiplexing capability by multiplexing the data channels from the above layers. L2CAP also provides fragmentation and reassembly for larger data packets. Link Layer (LL) is responsible for managing the channels and Physical Layer (PHY) transmits and receives the actual packets.

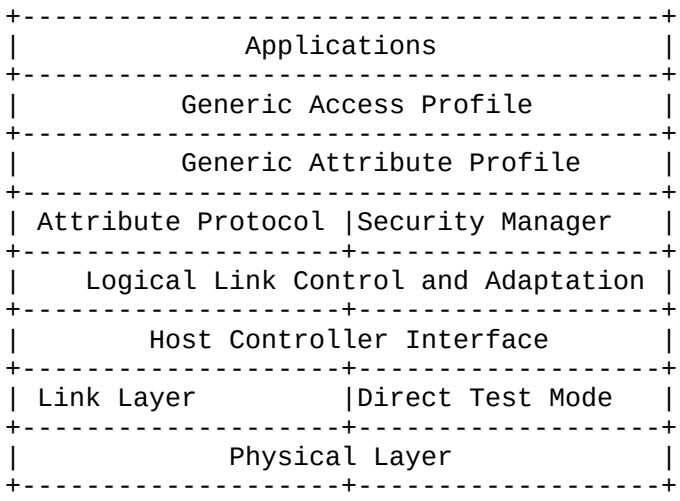


Figure 1: BT-LE Protocol Stack

3.1. Support for IPv6 over BT-LE

The Bluetooth Network Encapsulation Protocol (BNEP) has been developed for encapsulating any network protocol for Bluetooth L2CAP. BNEP assumes that L2CAP supports connection oriented channel. Either a connection oriented channel needs to be added to the current BT-LE specification, over which BNEP, parts of 6LoWPAN, IPv6 and application protocols can be run or a new fixed channel ID may be reserved for BNEP traffic. Figure 2 illustrates IPv6 over BT-LE stack.

Constrained Application Protocol (CoAP) is an application protocol specifically designed for resource constrained environments. CoAP could be run on top of IPv6 supporting requests from the server and requests of cached replies from a CoAP/HTTP proxy in the BT-LE gateway.

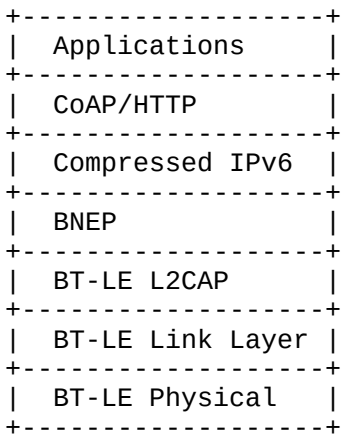


Figure 2: IPv6 over BT-LE Stack

4. Requirements

BT-LE technology sets strict requirements for low power consumption and thus limits the allowed protocol overhead. 6LoWPAN standard [RFC4944] provides useful generic functionality like header compression, link-local IPv6 addresses, Neighbor Discovery and stateless IP-address autoconfiguration for reducing the overhead in 802.15.4 networks. This functionality can be partly applied to BT-LE.

5. Addressing Model

The link model of BLE needs to be considered and what kind of addressing is possible.

6. MTU Issues

Generally the sensors generate data that fits into one Link Layer packet (23 bytes) that is transferred to the collector periodically. IP data packets may be much larger and hence MTU size should be the size of the IP data packet. Larger L2CAP packets can be transferred with the SAR feature of the Link Layer. If an implementation cannot support the larger MTU size (due to cost) then SAR needs to be supported at upper layers.

One option to support SAR would be to implement SAR functionality in

the BNEP layer. Existing SAR functionality defined in [RFC4944] could also be used, taking into account BT-LE specific features such as different MTU in the L2CAP layer.

7. LowPan Adaptation for BLE and frame format

Transmission of IPv6 Packets over IEEE 802.15.4 Networks [RFC4944] defines an adaptation layer between IP and 802.15.4 radio networks. In these networks link layer does not support SAR functionality and thus IP packets must fit into the payload that is available in the 127 octect long physical frame after variable size frame overhead has been added. In BT-LE networks this kind of adaptation is not needed if SAR is supported in the Link Layer. is a

8. IPv6 Address configuration

SLAAC and other means to configure an address on a BLE device. Neighbor Discovery Optimization for Low-power and Lossy Networks [I-D.ietf-6lowpan-hc]. Might also add something about hard-coding well-known gateway or server addresses.

9. IPv6 LLA in BLE

Link local address format in BLE

10. Unicast and Multicast address mapping

Do we have to use multicast addresses in ultra low power network? I dont know whether the same format specified for 802.15.4 can be reused. Will need expert guidance here.

11. Header compression

Compression Format for IPv6 Datagrams in Low Power and Lossy Networks (6LoWPAN) [I-D.ietf-6lowpan-hc].

In [RFC4944] different types of frame formats and related headers have been defined to support fragmentation and mesh addressing. In BT-LE context LoWPAN_HC1 compressed IPv6 header would be used by default. Support for fragmentation and mesh headers can be added if required. In BT-LE link with header compression IPv6 header (originally 40 Bytes) can be compressed to only 2 Bytes with link-local addresses and 26 Bytes with Global addresses. UDP header

(originally 8 Bytes) can be compressed to 4 Bytes. IMO this section should be the same as with 6lowpan.

12. IANA Considerations

This document does not have any IANA requests at this time. This may change with further development of the specification.

13. Security Considerations

The transmission of IPv6 over bluetooth low energy links has similar requirements and concerns for security as zigbee. Security at the IP layer needs to be reviewed as part of the development of the IPv6 over bluetooth low energy specification.

14. Additional contributors

Kanji Kerai and Jari Mutikainen from Nokia have contributed significantly to this document.

15. Normative References

[I-D.ietf-6lowpan-hc]
Hui, J. and P. Thubert, "Compression Format for IPv6 Datagrams in Low Power and Lossy Networks (6LoWPAN)", draft-ietf-6lowpan-hc-15 (work in progress), February 2011.

[I-D.ietf-6lowpan-nd]
Shelby, Z., Chakrabarti, S., and E. Nordmark, "Neighbor Discovery Optimization for Low-power and Lossy Networks", draft-ietf-6lowpan-nd-15 (work in progress), December 2010.

[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.

[RFC4994] Zeng, S., Volz, B., Kinnear, K., and J. Brzozowski, "DHCPv6 Relay Agent Echo Request Option", RFC 4994, September 2007.

Appendix A. Bluetooth Low energy basics

This section will provide background material on the basics of bluetooth low energy.

Authors' Addresses

Basavaraj Patil (editor)
Nokia
6021 Connection drive
Irving, TX 75039
USA

Email: basavaraj.patil@nokia.com

Teemu Savolainen
Nokia
Hermiankatu 12 D
FI-33720 Tampere
Finland

Email: teemu.savolainen@nokia.com

Johanna Nieminen
Nokia
Helsinki
Finland

Email: johanna.1.nieminen@nokia.com

Markus Isomaki
Nokia
Espoo
Finland

Email: markus.isomaki@nokia.com

Zach Shelby
Sensinode

6lowpan Working Group
Internet-Draft
Expires: April 29, 2011

Y. Qiu
J. Zhou
F. Bao
Institute for Infocomm Research

October 26, 2010

Lightweight Key Establishment and Management Protocol in Dymanmic Sensor Networks (KEMP)
draft-qiu-6lowpan-secure-router-01

Abstract

When a sensor node roams within a very large and distributed wireless sensor network, which consists of numerous sensor nodes, its routing path and neighborhood keep changing. In order to provide a high level of security in this environment, the moving sensor node needs to be authenticated to new neighboring nodes as well as to establish a key for secure communication. The document proposes an efficient and scalable protocol to establish and update the secure key in a dynamic wireless sensor network environment. The protocol guarantees that two sensor nodes share at least one key with probability 1 (100%) with less memory and energy cost, while not causing considerable communication overhead.

Status of this Memo

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

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

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

This Internet-Draft will expire on April 29, 2011.

Copyright Notice

Table of Contents

1. Introduction

The demand of wireless sensor networks (WSNs) is growing exponentially. It has turned out that the sensor networks can be widely applied in the areas of healthcare, environment monitoring, and the military. One of the surveys on WSNs points out that, in the near future, wireless sensor networks will be an integral part of our lives, more so than the present-day personal computer [1].

A sensor node has low capability in terms of power, computation, storage and communication. A wireless sensor network is composed of a large number of wireless sensor nodes and multi-hop communication is desired in WSNs. As a result, security in wireless sensor networks has six challenges to overcome: (a) the wireless nature of communication, (b) resource limitations of sensor nodes, (c) very large and dense WSNs, (d) lack of fixed infrastructure, (e) unknown network topology prior to deployment, (f) high risk of physical attacks on unattended sensors [2][3].

The capabilities in term of Scalability, Mobility/Dynamicity Network, Latency, etc. are also listed in the RFC documents, i.e. Routing Requirements for Urban Low-Power and Lossy Networks (RFC 5548)[6], Routing Requirements for Urban Low-Power and Lossy Networks (RFC 5673)[7], Home Automation Routing Requirements in Low-Power and Lossy Networks (RFC 5826)[8], and Building Automation Routing Requirements in Low-Power and Lossy Networks (RFC 5867)[9].

RFC 5548 required local network dynamics SHOULD NOT impact the entire network to be reorganized or re-reconfigured; a viable routing security approach SHOULD be sufficiently lightweight that it may be implemented across all nodes in a U-LLN; the U-LLN MUST deny any node that has not been authenticated to the U-LLN and authorized to participate to the routing decision process.

RFC 5673 addressed the handover speed; a compromised field device does not destroy the security of the whole network; because nodes are usually expected to be capable of routing, the end-node security requirements are usually a superset of the router requirements.

RFC 5826 needed a node MUST authenticate itself to a trusted node that is already associated with the LLN before the former can take part in self-configuration or self-organization. A node that has already authenticated and associated with the LLN MUST deny, to the maximum extent possible, the allocation of resources to any unauthenticated peer. The routing protocol(s) MUST deny service to any node that has not clearly established trust with the HC-LLN.

RFC 5867 listed the possible security keys below: a) a key obtained

from a trust center already operable on the LLN; b) a pre-shared static key as defined by the general contractor or its designee; or c) a well-known default static key.

With the aforementioned limitations of the existing solutions in mind, we now propose a secure protocol in dynamic WSN, addressing all of the following issues:

- A moving sensor node needs to change its attached routers (or cluster heads) frequently.
- A router (or cluster head) needs to ensure a joining node is not a malicious sensor.
- A moving node needs to establish a secure tunnel with the new router (or cluster head).
- The energy consumption for establishing the secure tunnel must be minimal.

One of the important novel features of the proposed protocol is that the router or cluster head is employed as sub-base-stations to execute key establishment. This way, the total dependency on the base station for key establishment can be avoided. Also, this approach reduces the hops between two communicating ends and hence results in reduction of the communication cost.

2. Network Assumptions

In this document, we consider a scenario in which a sensor node roams within a very large and distributed WSN, consisting of a large number of sensor nodes. It is a typical scenario that is widely adopted in hospital environments as the patients or doctors equipped with sensors roam across each department in the hospital. A patient who carries the sensor nodes can move freely within the range of a hospital. When a wireless sensor node is moving, its routing path and neighborhood keep changing. The moving node needs to be authenticated to the new neighbors and to establish a key for secure communication.

This scenario reflects the problems described in Section 1: (a) composition by a large number of sensor nodes; (b) communication based on wireless multi-hop mechanism; (c) no fixed infrastructure; (d) the possible location change of sensor node (patient). Therefore, the challenges of this network assumption are how to establish a secure channel with these routers.

3. Shared-Key Discovery

In the WSN environment, as data transmission consumes much more energy than computation, the probabilistic solution is widely accepted in order to reduce the storage and communication overhead during key establishment.

So far in the literature, numerous random key pre-distribution schemes have been proposed. For example, in Chan et al.'s scheme[4], each sensor node stores a random set of Np dedicated pair-wise keys to achieve the probability p that two nodes share a key. At the key setup phase, each node ID is matched with Np other randomly selected node IDs with probability p. A distinct pair-wise key is generated for each ID pair, and is stored in both nodes' key-chain along with the ID of the other party. During the shared-key discovery phase, each node broadcasts its ID so that neighboring nodes can tell if they share a common pair-wise key. Note that Chan et al.'s scheme reduces the storage overhead by sacrificing key connectivity, but it still provides perfect key resilience.

In this protocol, it is assumed that a sensor node (carried by a patient) can move within a special range (e.g. hospital). As each sensor's memory is severely constrained, each sensor may only store a small set of keys randomly selected from a key pool at the deployment. Two nodes may use any existing key discovery protocol (e.g., the solution proposed in [4]) to find a common key from their own sets. If the common key is not found, the key establishment scheme will be initiated. The reason why binding a general pre-shared key discovery phase to the protocol is to reduce the energy cost as much as possible.

4. Dynamic Authentication and Key Establishment Protocol

4.1. Basic Protocol

Due to the limited storage of sensor nodes, the pre-shared key-pair is not always available between the roaming node and its new neighbors in the circumstance of a dynamic node roaming within large WSNs (e.g., in hospitals and nuclear power plants). Therefore it requires an efficient and scalable protocol to establish and update the keys among nodes for secure communications.

Figure 1 shows the basic architecture and message flow of our protocol for authentication and key establishment in dynamic WSNs. When a dynamic sensor node moves to a new area and wants to attach to a router or a cluster head in this area, it first sends a request message to the base station (refer to Figure 1).

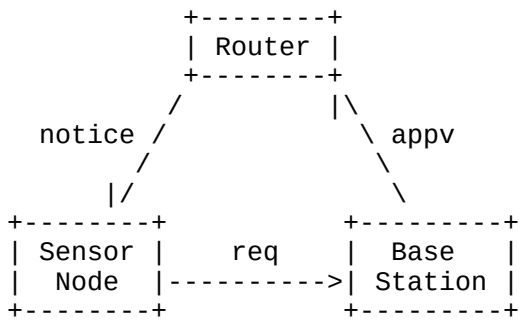


Figure 1. The basic architecture and message flow of KEMP protocol

req={Src=SN, Dst= BS, RT||R0||MAC(K_BN, SN||RT||R0)} (1)

where Src and Dst denote the source and destination address of a message respectively. SN, BS and RT are identifiers for sensor node, base station and router, respectively. R0 denotes a random number generated by the sensor node. MAC indicates the message authentication code algorithm with a key and K_BN is the shared secret key between the base station and the sensor node.

After receiving the req message, the base station will check its revocation list whether the sensor node has been revoked. If the sensor node is acceptable, then the base station verifies the MAC message. If the result is positive, the base station will generate a session key K_NR for the roaming sensor node and the router (or cluster head).

K_NR = H(K_BN, SN||R0||R1) (2)

where H is a keyed one-way hash function, and R1 is the random number selected by the base station. The base station then sends an approval message appv with the session key to the router:

appv = {Src=BS, Dst=RT, E(K_BR, SN||R0||R1||K_NR)} (3)

where E is an encryption algorithm, and K_BR is the shared secret key between the base station and the router.

After receiving the appv message, the router decrypts the payload and extracts the session key KNR, and then sends a notice to the sensor node.

notice = {Src=RT, Dst=SN, R0||R1|| MAC(K_NR, RT||SN|| R0||R1)} (4)

Upon getting the notice message, the sensor node extracts the random numbers R0 and R1. After checking if the received random number R0 is equal to the original R0, the sensor node recalculates the session key K_NR = H(K_BN, SN||R0||R1) and then verifies the MAC value. If the result is positive, the sensor node will use the session key for the communication with this router afterwards. In practice, the router could be any sensor node that the dynamic sensor node wants to connect to.

4.2. Key Management

In order to manage the keys, every sensor node maintains a table, called "Key Cache". Table 1 shows the structure of the Key Cache.

Table 1. The structure of Key Cache

Key Cache in Sensor Node N		
Correspondence Node ID	Key	Key Lifetime
BS	K_BN	T_BN
Node_i	K_Ni	T_Ni
... ...	...	... ...
Node_j	K_Nj	T_Nj
PreSharedKey_x	K_x	T_x
... ...	...	... ...
PreSharedKey_y	K_y	T_y

When a sensor node, say node N, wants to connect to other sensor

node, say node R, it executes the following procedure:

(1) Checks first if there is an existing key pair between them.

(2) Otherwise, processes the subroutine of shared-key discovery to find a common key between node N and node R based on those "PreSharedKeys" in their key caches.

(3) If there is still no common key between them, the sensor node allocates an entry in the key cache, and assigns Node ID as nodeR, Key Stuff as the random number R0 and Key Lifetime as 0, as shown in Table 2.

Table 2. The initial key entry.

Correspondence Node ID	Key	Key Lifetime
Node_R	R0	0

(4) Then the sensor node initiates the procedure of key establishment described in the above section. After receiving the notice message, and recalculating the session key KNR, the sensor node updates the entry's key stuff and key lifetime accordingly.

(5) When the key lifetime is expired, the dynamic sensor node should re-initiate the procedure of key establishment described in the above section.

(6) When the sensor node leaves the range of the connected router, the sensor node deletes the related entry from its cache table in order to save the storage. In case there is no space for adding a new entry, it may first delete the oldest key which has expired or will expire soon.

The base station also maintains a key table (Table 3) that includes the secret keys shared with all of the sensor nodes in the network.

Table 3. The structure of Key Table in basestation

Key Table in Base Station		
Node ID	Key	Key Lifetime
Node_i	K_Bi	T_Bi
... ...	...	... ...
Node_j	K_Bj	T_Bj

If a node is compromised and revoked, its field of key lifetime would be marked as negative.

4.3. Distribution Mode

In WSNs, the more hops between two communicating ends exist, the poorer the traffic performance becomes and the more energy consumption is required. To overcome these problems, we introduce the distribution mode.

The major idea of distribution mode is to deploy the cluster heads as the sub-base-stations because a cluster head is more powerful than normal sensor nodes. The distribution mode includes the following steps:

(1) Each cluster head manages to establish the shared key with its neighboring cluster heads after deployment. There are several ways to do this. One could embed those keys in advance if the topology is known at deployment, or use the basic protocol described in the above sections, via the base station. (As this is a one-time operation, the overheads may be acceptable.)

(2) Each sensor node keeps two base station identifiers (IDs): one is a real base station ID; the other is a sub-base-station (the cluster head) ID. Initially, the ID of sub-base-station is a real base station.

(3) After deployment, the first round for a mobile node to establish the shared key with the nearest cluster head uses the basic protocol, too.

(4) When the mobile node moves, use the basic protocol to establish the shared key with the new cluster head, via the sub-base-station (old cluster head) rather than the real base station.

(5) After successfully establishing the keys, the sensor node updates the ID of sub-base-station with the current cluster head.

(6) For security reasons, each sensor node must reset its sub-base-station ID to the real base station at a specified interval (say a few hours or days, depending on the various applications) and re-establish keys with its near cluster heads via the real base station. If the base station does not receive any request from a sensor node, it considers the sensor node has been

compromised.

The distribution mode could provide an efficient and low energy-cost solution for the shared-key establishment. The basic protocol can provide the stronger protection since it can immediately block and revoke compromised nodes.

5. Security Consideration

In this proposed protocol, the session key K_NR between the sensor node and the router is generated by the base station and sensor node respectively, and the session key is directly sent to the router from the base station by an encrypted packet. Hence, the session key K_NR is never disclosed during transmission. The session key K_NR is only known by the related peers, i.e., the sensor node, the base station and the router.

Referring to equation (2), the session key K_NR is generated by a keyed hash function with the shared key K_BN between sensor node and base station as well as two random numbers, R0 and R1, which are generated by the sensor node and base station respectively. As both R0 and R1 are used only one time, there are not the same session keys K_NR. This property is useful to against the replication attacks.

Since the session key K_NR is generated by a keyed hash function with the secret key K_BN between the sensor node and the base station, the different sensor nodes will have different session keys. This feature is useful to protect sensor node privacy.

Even though an eavesdropper at the edge of the sensor node can monitor and capture the random numbers R0 and R1 as well as the identity of the sensor node, it is still not able to regenerate the session key K_NR due to lack of the secret key K_BN. Without a proper session key, the routers will not forward the packets to next nodes. This attribute could prevent camouflage and traffic attacks.

Due to the fact that no trusted connection is established between sensor node and new router before the connection between them, the proposed protocol employs a random number R1 issued by the base station. The sensor node needs to recalculate the K_NR first based on the R1 together with K_BN and R0. Then using the calculated session key K_NR to verify the received session key K_NR and the random number R1. If the result is positive, then the sensor node will trust that the router is authorized by the base station.

Besides the function of informing the sensor node that the new session key K_NR is ready to use in the router, the notice message also plays an important role to check if the sensor node!_s address is reachable. Without this reachability check, the sensor node may claim that it is at any location rather than its real location. It could launch redirecting attacks.

The path between the base station and the router is secure because the packet between them is encrypted with a pre-shared key K_BR.

The messages from the sensor node to the base station and from the router to the sensor node are authenticated by a keyed hash function. Before accepting the inward message and making further processing, the receivers must verify the authentication. Since the cost of a hash algorithm is very small, the base station and sensor node could avoid the attacks of denial of service.

In order to achieve high efficiency and low energy cost, the protocol deploys a distribution mode which uses the cluster headers as the sub-base-stations. Due to the capability of cluster header, it is not able to recognize any compromised sensor nodes in time; the protocol requires each sensor node to reset its sub-base-station ID to the real base station regularly, and to re-establish keys with its near cluster heads via the real base station. This step is also useful to avoid a sensor node binding a compromised cluster head for long time.

According to the above analysis, this proposed protocol, which is simple and easy to implement, can provide relatively strong protection for sensor node networks.

6. IANA Consideration

This version does not need new values to be assigned by IANA.

7. Conclusions

In this document, we have proposed an efficient and scalable protocol to establish and update the authentication key between any pair of sensor nodes in a dynamic wireless sensor network. Our protocol has the following features:

- It is suitable for both static and dynamic WSNs. Any pair of nodes can establish a key for secure communication.
- A roaming node only deals with its closest router for security. There is no need to change the rest of routing path to the base station.
- The base station can manage a revocation list for lost or compromised roaming nodes.
- The system is scalable and resilient against node compromise.
- The protocol is efficient due to the small number and size of signalling messages.
- The size of each signalling message is smaller than the IEEE 802.15.4 frame size so that it can to avoid packet fragmentation and the overhead for reassembly.
- The distribution mode can considerably reduce the latency.
- Any pair of nodes can establish a key. The protocol guarantees that two sensor nodes share at least one key with probability 1 (100%).

Thanks to above features, the protocol can satisfy the requirements for IPv6 over Low power WPAN Routing [5] and could be the security solution deployed in Routing Requirements for Urban Low-Power and Lossy Networks (RFC 5548)[6], Routing Requirements for Urban Low-Power and Lossy Networks (RFC 5673)[7], Home Automation Routing Requirements in Low-Power and Lossy Networks (RFC 5826)[8], and Building Automation Routing Requirements in Low-Power and Lossy Networks (RFC 5867)[9].

After comparing with some of the popular and latest protocols used in WSNs, our protocol could save about 30% in communication energy, and has the higher probability (100%) of sharing a key between two sensor nodes with less memory cost than those pre-distribution schemes, without incurring in a considerable amount of communication.

8. Normative References

[1] Akyildiz, I., Sankarasubramaniam, Y., and E. Cayirci, "Wireless sensor networks: a survey", Comput. Netw 38, 393-422, 2002.

[2] Camtepe, S. and B. Yener,, "Key Distribution Mechanisms for Wireless Sensor Networks: a Survey", Technical Report TR-05-07; Department of Computer Science, Rensselaer Polytechnic Institute: Troy, NY, USA , Mar. 2005.

[3] Kushalnagar, N., Montenegro, G., and C. Schumacher, "IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs): Overview, Assumptions, Problem Statement, and Goals", RFC 4919, August 2007.

[4] Chan, H., Perrig, A., and D. Song, "Random key predistribution schemes for sensor networks", IEEE Symposium on Research in Security and Privacy Oakland, California, USA, May 2003.

[5] Kim, E., Kaspar, D., Gomez, C., and C. Bormann, "Problem Statement and Requirements for 6LoWPAN Routing", Work in Progress, Aug. 2010.

[6] Dohler, M., Watteyne, T., Winter, T., and D. Barthel, "Routing Requirements for Urban Low-Power and Lossy Networks", RFC 5548, May 2009.

[7] Pister, K., Thubert, P., Dwars, S., and T. Phinney, "Industrial Routing Requirements in Low-Power and Lossy Networks", RFC 5673, October 2009.

[8] Brandt, A., Buron, J., and G. Porcu, "Home Automation Routing Requirements in Low-Power and Lossy Networks", RFC 5826, April 2010.

[9] Martocci, J., De Mil, P., Riou, N., and W. Vermeylen, "Building Automation Routing Requirements in Low-Power and Lossy Networks", RFC 5867, June 2010.

Authors' Addresses

Ying Qiu
Institute for Infocomm Research, Singapore
1 Fusionopolis Way
#21-01 Connexis (South Tower)
Singapore 138632

Phone: +65-6408 2053
Email: qiuying@i2r.a-star.edu.sg

Jianying Zhou
Institute for Infocomm Research, Singapore
1 Fusionopolis Way
#21-01 Connexis (South Tower)
Singapore 138632

Phone: +65-6408 2075
Email: jyzhou@i2r.a-star.edu.sg

Feng Bao
Institute for Infocomm Research, Singapore
1 Fusionopolis Way
#21-01 Connexis (South Tower)
Singapore 138632

Phone: +65-6408 2073
Email: baofeng@i2r.a-star.edu.sg

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: September 2, 2011

B. Sarikaya
F. Xia
Huawei USA

March 1, 2011

Lightweight Secure Neighbor Discovery for Low-power and Lossy Networks
draft-sarikaya-6lowpan-cgand-00

Abstract

This document defines lightweight secure neighbor discovery for low-power and lossy networks. The nodes generate a Cryptographically Generated Address using an Elliptic Curve Cryptography public key, register the Cryptographically Generated Address with a default router and periodically refresh the registration. Modifications to 6lowpan Neighbor Discovery protocol are described for secure neighbor discovery for low-power and lossy networks. Cryptographically generated address and digital signature are calculated using elliptic curve cryptography public key of the node.

Status of this Memo

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

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

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

This Internet-Draft will expire on September 2, 2011.

Copyright Notice

Table of Contents

1. Introduction

Neighbor discovery for IPv6 [RFC4861] and stateless address autoconfiguration [RFC4862] together referred to as neighbor discovery protocols (NDP) are defined for regular hosts operating in wired/wireless links. These protocols are not suitable and require optimizations for resource constrained, low power hosts operating in lossy wireless links. Neighbor discovery optimizations for 6lowpan networks include simple optimizations such as host address registration feature using the address registration option which is sent in unicast Neighbor Solicitation (NS) and Neighbor Advertisement (NA) messages [I-D.ietf-6lowpan-nd].

Neighbor discovery protocols (NDP) are not secure especially when physical security on the link is not assured and vulnerable to attacks. Secure neighbor discovery protocol (SEND) is defined to secure NDP [RFC3971]. Cryptographically generated addresses (CGA) are used in SEND [RFC3972]. SEND mandates the use of RSA signature algorithm which is computationally heavy and not suitable to use for low-power and resource constrained nodes [I-D.cheneau-csi-send-sig-agility]. The use of RSA public key and signature leads to long message sizes not suitable to use in low-bit rate, short range, asymmetric and non-transitive links such as IEEE 802.15.4.

In this document we extend 6lowpan neighbor discovery protocol with cryptographically generated addresses. The nodes generate CGAs and register them with the default router. CGA generation is based on elliptic curve cryptography (ECC)and signature is calculated using elliptic curve digital signature algorithm (ECDSA) known to be lightweight and lead to much smaller packet sizes. The resulting protocol is called Lightweight Secure Neighbor Discovery Protocol (LSEND).

2. Terminology

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

The terminology in this document is based on the definitions in [RFC3971], [RFC3972] in addition to the ones specified in [I-D.ietf-6lowpan-nd].

3. Problem Statement

In this section we state requirements on secure neighbor discovery protocol for low-power and lossy networks.

The protocol MUST be based on Neighbor Discovery Optimization for Low-power and Lossy Networks protocol defined in [I-D.ietf-6lowpan-nd] due to the host-initiated interactions to allow for sleeping hosts, elimination of multicast-based address resolution for hosts, etc.

New options to be added to neighbor solicitation messages MUST lead to minimal packet sizes. Such packet sizes facilitate low-power transmission by resource constrained nodes on lossy links.

CGA generation, signature and key hash calculation MUST avoid the use of SHA-1 which is known to have security flaws. In this document, we use SHA-2 instead of SHA-1 and thus avoid SHA-1's flaws.

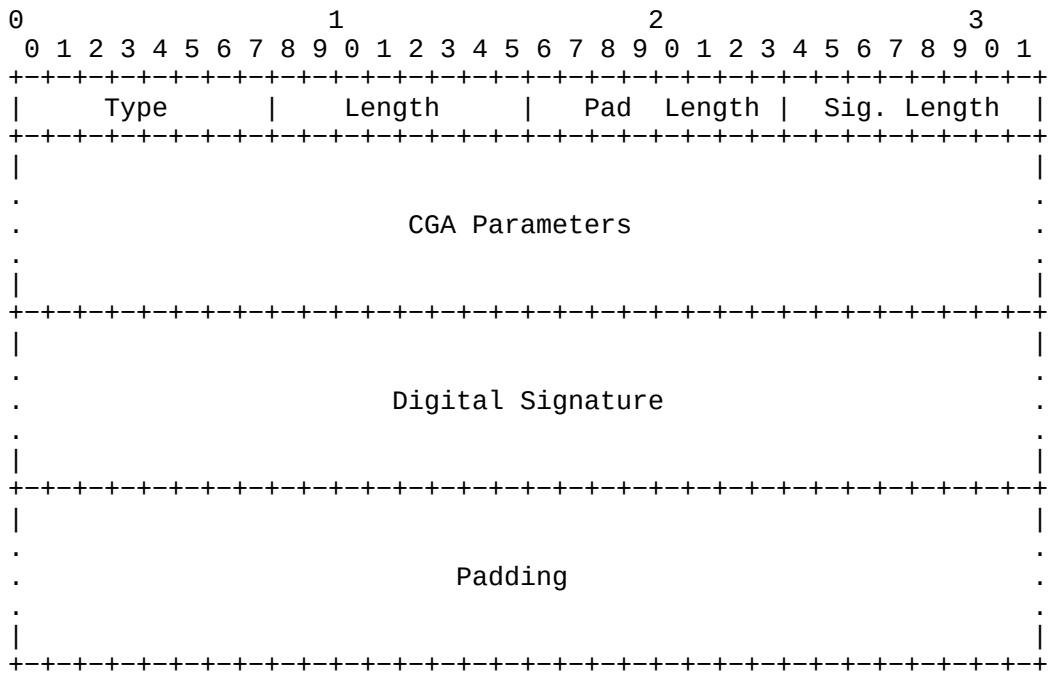
Public key and signature sizes MUST be minimized and signature calculation MUST be lightweight. In this document we adopt ECC and ECDSA with P-256 curve in order to meet this requirement.

4. New Options

4.1. CGA Parameters and Digital Signature Option

This option contains both CGA parameters and the digital signature.

A summary of the CGA Parameters and Digital Signature Option format is shown below.



Type

TBA1 for CGA Parameters and Digital Signature

Length

The length of the option (including the Type, Length, Pad Length, Signature Length, CGA Parameters, Digital Signature and Padding fields) in units of 8 octets.

Pad Length

The length of the Padding field.

Sig Length

The length of the Digital Signature field.

CGA Parameters

The CGA Parameters field is variable-length containing the CGA Parameters data structure described in Section 4 of [RFC3972].

Digital Signature

The Digital Signature field is a variable length field containing a Elliptic Curve Digital Signature Algorithm (ECDSA) signature (with SHA-256 and P-256 curve of [FIPS-186-3]). Digital signature is constructed as explained in Section 4.3.

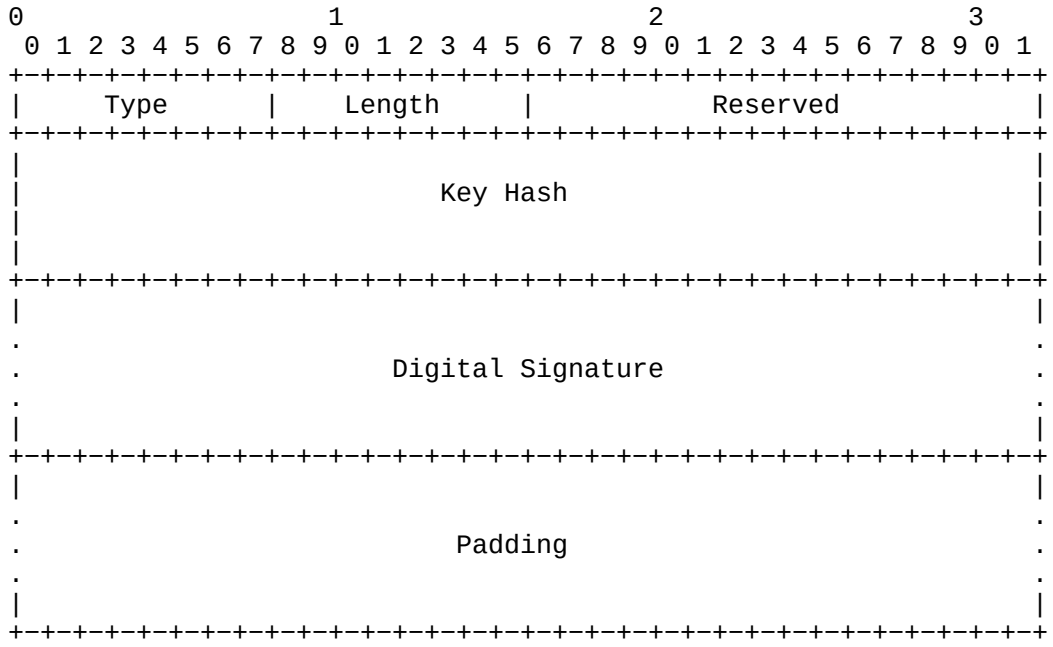
Padding

The Padding field contains a variable-length field making the CGA Parameters and Digital Signature Option length a multiple of 8.

4.2. Digital Signature Option

This option contains the digital signature.

A summary of the Digital Signature Option format is shown below. Note that this option has the same format as RSA Signature Option defined in [RFC3971]. The differences are that Digital Signature field carries Ellictic Curve Cryptography signature not RSA signature and in calculating Key Hash field SHA-2 is used not SHA-1.



Type

TBA2 for Digital Signature

Length

The length of the option (including the Type, Length, Reserved, Key Hash, Digital Signature and Padding fields) in units of 8 octets.

Key Hash

The Key Hash field is a 128-bit field containing the most significant (leftmost) 128 bits of a SHA-2 hash of the public key used for constructing the signature. This is the same as in [RFC3971] except for SHA-1 which has been proved to be flawed in the light of recent attacks [NIST-ST].

Digital Signature

Same as in Section 4.1.

Padding

The Padding field contains a variable-length field containing as many bytes long as remain after the end of the signature.

4.3. Calculation of Digital Signature and CGA Using ECC

Due to the use of Elliptic Curve Cryptography, the following modifications are needed to [RFC3971] and [RFC3972].

Digital signature is constructed by using the sender's private key over the same sequence of octets specified in Section 5.2 of [RFC3971] up to all neighbor discovery protocol options preceding the Digital Signature option containing Elliptic Curve Cryptography digital signature. The signature value is computed using the ECDSA signature algorithm as defined in [SEC1] and hash function SHA-256.

Public Key is the most important parameter in CGA Parameters defined in Section 4.1. Public Key MUST be DER-encoded ASN.1 structure of the type SubjectPublicKeyInfo formatted as ECC Public Key. The AlgorithmIdentifier, contained in ASN.1 structure of type SubjectPublicKeyInfo, MUST be the (unrestricted) id- ecPublicKey algorithm identifier, which is OID 1.2.840.10045.2.1, and the subjectPublicKey MUST be formatted as an ECC Public Key, specified in Section 2.2 of [RFC5480].

Note that the ECC key lengths are determined by the namedCurves parameter stored in ECParameters field of the AlgorithmIdentifier. The named curve to use is secp256r1 corresponding to P-256 which is OID 1.2.840.10045.3.1.7.

5. Protocol Interactions

Lightweight Secure Neighbor Discovery for Low-power and Lossy Networks (LSEND for LLN) modifies Neighbor Discovery Optimization for Low-power and Lossy Networks [I-D.ietf-6lowpan-nd] as explained in this section. Protocol interactions are shown in Figure 1.

6LoWPAN Border Routers (6LBR) send router advertisements (RA). 6LoWPAN Nodes (6LN) or nodes in short receive these RAs and generate their own cryptographically generated addresses using elliptic curve cryptography as explained in Section 4.3. The node sends a neighbor solicitation (NS) message with address registration option (ARO) to 6LBR. Such a NS is called an address registration NS.

A LSEND for LLN node MUST send an address registration NS message after adding CGA Parameters and Digital Signature Option defined in Section 4.1. Source address MUST be set to its crypotographically generated address. A LSEND for LLN node MUST set the Owner Interface Identifier field (EUI-64) in ARO to the rightmost 64 bits of its crypotographically generated address. Subnet Prefix field of CGA Parameters MUST be set to the leftmost 64 bits of its crypotographically generated address. Public Key field of CGA Parameters MUST be set to the node's ECC Public Key.

6LBR receives the address registration NS. 6LBR verifies the source address as described in Section 5.1.2. of [RFC3971] using the claimed source address and CGA Parameters field in the message. After successfully verifying the address 6LBR next does a cryptographic check of the signature included in Digital Signature field in the message. If all checks succeed then 6LBR performs a duplicate address detection procedure first on the address. If that also succeeds 6LBR registers CGA in the neighbor cache. 6LBR also caches the node's public key.

6LBR sends an address registration neighbor advertisement (NA) as a reply to confirm the node's registration. Status is set to 0 to indicate success. This completes initial address registration. The address registration needs to be refreshed after the neighbor cache entry times out.

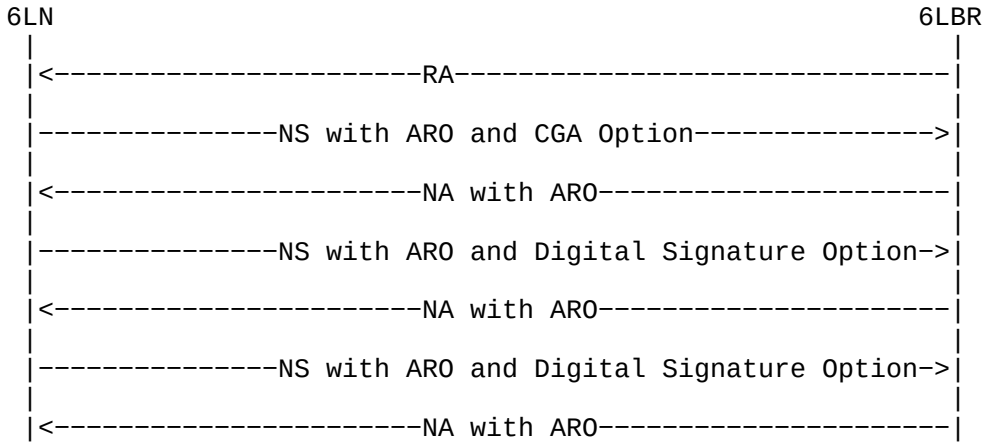


Figure 1: Lightweight SEND for LLA Protocol

In order to refresh the neighbor cache entry, a LSEND for LLN node MUST send an address registration NS message after adding Digital Signature Option defined in Section 4.2. Key hash field is a hash of the node's public key and MUST be set as described in Section 4.2. Digital Signature field MUST be set as described in Section 4.2.

6LBR receives the address registration refresh NS. 6LBR uses the key hash field in Digital Signature Option to find the node's public key from the neighbor cache. 6LBR verifies the digital signature in the NS. In case of successful verification, 6LBR sends back an address registration neighbor advertisement (NA) to the node and sets the status to 0 indicating successful refreshment of the CGA of the node. Similar refresh NS and NA exchanges happen afterwards as shown in Figure 1

5.1. Packet Sizes

Original address registration NS message contains 40 byte header and ARO is 16 octets. DER-encoded ECC Public Key for P-256 curve is 88 octets long. Digital Signature field when using ECDSA for P-256 curve is 71 octets long without padding [I-D.cheneau-csi-ecc-sig-agility].

CGA Parameters and Digital Signature Option's CGA Parameters include 16 octet modifier, 8 octet prefix obtained from the router advertisement message sent from 6LBR, 1 octet collision count and 88 octet Public Key. Digital Signature is 71 octets. The option is 184 octets with Padding 0 octets. The total message size of an original LSEND address registration NS message is 240 octets and such a

message can be encapsulated into three 802.15.4 frames.

An address registration refresh NS message contains an ARO which is 16 octets and digital signature option containing 16 octet key hash and 71 octet signature and 5 octet Padding. The message is 152 octets long with the header. Such a message could be encapsulated in two 802.15.4 frames.

6. Security Considerations

Same considerations regarding the threats to the Local Link Not Covered as in [RFC3971] apply.

The threats discussed in Section 9.2 of [RFC3971] are countered by the protocol described in this document as well.

As to the attacks to the protocol itself, denial of service attacks that involve producing a very high number of packets are deemed unlikely because of the assumptions on the node capabilities in low-power and lossy networks.

7. IANA considerations

This document defines two new options to be used in neighbor discovery protocol messages and new type values for CGA Parameters and Digital Signature Option (TBA1) and Digital Signature Option (TBA2) need to be assigned by IANA.

8. Acknowledgements

TBD.

9. References

9.1. Normative References

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

[RFC3971] Arkko, J., Kempf, J., Zill, B., and P. Nikander, "SEcure Neighbor Discovery (SEND)", RFC 3971, March 2005.

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

[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.

[RFC5480] Turner, S., Brown, D., Yiu, K., Housley, R., and T. Polk, "Elliptic Curve Cryptography Subject Public Key Information", RFC 5480, March 2009.

[I-D.ietf-6lowpan-nd] Shelby, Z., Chakrabarti, S., and E. Nordmark, "Neighbor Discovery Optimization for Low-power and Lossy Networks", draft-ietf-6lowpan-nd-15 (work in progress), December 2010.

9.2. Informative references

[SEC1] "Standards for Efficient Crtptography Group. SEC 1: Elliptic Curve Cryptography", September 2000.

[FIPS-186-3] "National Institute of Standards and Technology, "Digital Signature Standard"", June 2009.

[NIST-ST] "National Institute of Standards and Technology, "NIST Comments on Cryptanalytic Attackts on SHA-1"", January 2009, <http://csrc.nist.gov/groups/ST/hash/statement.html>.

[I-D.cheneau-csi-ecc-sig-agility] Cheneau, T., Laurent, M., Shen, S., and M. Vanderveen, "ECC public key and signature support in Cryptographically Generated Addresses (CGA) and in the Secure Neighbor Discovery (SEND)", draft-cheneau-csi-ecc-sig-agility-02 (work in progress), June 2010.

[I-D.cheneau-csi-send-sig-agility] Cheneau, T., Laurent, M., Shen, S., and M. Vanderveen, "Signature Algorithm Agility in the Secure Neighbor Discovery (SEND) Protocol", draft-cheneau-csi-send-sig-agility-02 (work in progress), June 2010.

Authors' Addresses

Behcet Sarikaya
Huawei USA
1700 Alma Dr. Suite 500
Plano, TX 75075

Phone: +1 972-509-5599
Email: sarikaya@ieee.org

Frank Xia
Huawei USA
1700 Alma Dr. Suite 500
Plano, TX 75075

Phone: +1 972-509-5599
Email: xiayangsong@huawei.com

6LowPAN Network Working Group
Internet draft
Expires: July 31, 2011

Gohel Bakul Chandulal
Dhananjay Singh

Future Internet Team,
National Institute for
mathematical science,
Daejeon, South Korea
February 23, 2011

Global connectivity for 6lowpan
draft-singh-6lowpan-global-connectivity-00.txt

Abstract

This document describes the short AID (adaptation identifier) in place of full IPv6 address, related AID-IPv6 address translation mechanism and frame format of it for effective IPv6 header compression when a IEEE 802.15.4 node communicate with a IPv6 domain. AID generated by IN-node (a node inside the lowpan) for corresponding IPv6 address of OUT-node (a node outside the lowpan), and AID-IPv6 translation table maintained at gateway and IN-node. Conversely packet carries an AID value in place of OUT-node IPv6 address in adaptation header, and translated back to IPv6 at gateway though AID-IPv6 translation table. Also in this document, effective frame format design specified for adaptation layer for global as well as local communication

Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].
Readers are expected to be familiar with all the terms and concepts that are discussed in "IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs): Overview, Assumptions, Problem Statement, and Goals" [RFC4919], and "Transmission of IPv6 Packets over IEEE 802.15.4 Networks" [RFC4944].

AID: Adaptation Identifier

IN-node: a IEEE 802.15.4 node within the PAN (personal area network)

OUT-node: Any node outside the PAN, connected with IN-node through IPv6 Domine

IN-bound traffic: Flow of packet from outside PAN (OUT-node) to inside PAN (IN-node)

OUT-bound traffic: Flow of packet from inside PAN (IN-Node) to outside the PAN (OUT-node)

AITT: AID-IPv6 Translation Table

Table of content

1. Problem statements

1.1 global connectivity of 6lowpan & header compression

6lowpan developed with aim to provide internet connectivity to lowpan (IEEE 802.15.4 network), so IN-node communicates with OUT-node in IPv6 domine. Maximum physical layer packet size of IEEE 802.15.4 is 127 byte, and it left only 102 byte for layers above the MAC layer. Link layer security further consumes 21 byte. IPv6 header is 40 octets in length, and leaves only 41 octets for upper layer. So HC1 header compression was proposed to reduce the IPv6 header size. Further MTU size of IPv6 packet is over 1280 bytes. So it requires fragmentation and reassembling of IPv6 packet. For these reasons an adaptation layer was proposed to accommodate IPv6 packet over IEEE 802.15.4 network.

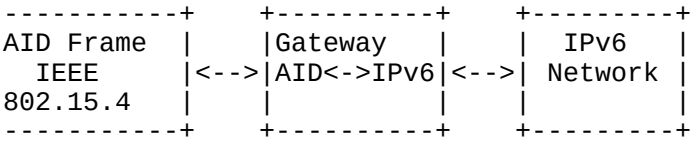


Figure 1. Global connectivity

[RFC 4944] define the IPv6 header compression to reduce the size of IPv6 header, and able to compress 40 byte header minimum up to 2 byte. One byte for header compression filed and one byte for hop limit (inline). Field that cannot be compressed is placed inline next to compressed header within the adaptation frame. When a node communicates across IPv6 internet, it requires full IP address of OUT-node, so full IP address has to put it into the inline according to HC1 compression scheme. Due to the state less auto configuration properties, some way we can save 8 byte or 14 byte for a IN-node addresses, depending on EUI-64 addressed or 16 bit short address used for IN-node. So actual compression for IPv6 addresses of IN-node and OUT-node are up to 20-26 byte / 32 byte with HC1 compression scheme, But it is not efficient compression for global communication. To tackle this problem, In [I-D. Global connectivity in 6LoWPAN] author proposed a short length AID assignment at gateway to map unique IPv6 address to achieve good IPv6 addresses compression for global communication. Conversely, packet within PAN carries short length AID in place of full fledges IPv6 address and convert to full IPv6 address at gateway to route over internet. For IN-bound IPv6 packet, procedure is just reverse. In [I-D. Global connectivity in 6LoWPAN], Author proposed two AID, one for Source and one for Destination and mechanism to generate AID for unique IPv6 address. But, Author does not provide any information regarding frame format with AID, presence of link-addresses in adaptation layer, AID field size as well as mobility scenario and AID mechanism.

1.2 hop limit & HC1 compression

In RFC 4944, hop limit (1 byte) field from IPV6 header which is always carried inline. When Mesh header present, it also carries a hopleft field (4 bits). So it gives rise confliction to algorithm that which field to be considered. One possibility is that both fields require simultaneously when hop limit set different for routing within PAN and outside PAN for outbound packet, but it require additional field that inform such situation, but currently this information field is not present in the adaptation header. Conversely, hope limit field in adaptation layer required revision.

1.3 HC1 compression header and mesh header

For transmission of message within the PAN, mesh header defined in [rfc 4944]. In this scenario first four bits of the HC1 never required, as the both origin and destination link layer address present in mesh header. It's implicit information. Moreover, with use of AID frame, there are no requirements of first four bits of HC1 header at all.
In HC1 compression, for prefix and II ID for OUT-node, only inline option is possible. Therefore, no compression for IPv6 address of OUT-node. For II ID of IN-node, there is an option 'IC: Interface identifier elided'. How can we derive II ID from link-layer? So always we have to put it inline. One possibility is from mesh header, but use of it for global communication causes extra load on header and so no gain. So HC1 header compression header and mesh frame format required revision.

2 Adaptation Identifier (AID)

In [I-D. Global connectivity in 6LoWPAN] author proposed a two AID value for source and destination node IPv6 address. But Use of AID for an IN-node is inappropriate which cause extra load on adaptation header, extra management and lead to certain difficulties in handling it. AID only require for OUT-node, and translation between AID and IPv6 address take place at gateway. Following session explain AID requirements, size of AID field and AID-IPv6 address translation table (AITT).

2.1 Presence of IN-node link layer address and AID

When the frame contains only AID value and does not contain IN-node link layer a ddress lead to certain issues.

Following issues suggest the requirement of link-layer address in adaptation he ader.

(1) In case of any desyncronization between the node and gateway regarding AID v alue, particularly happen in case of a PAN with multiple gateways and IN-node mo bility scenario in which If AID value does not exist at gateway , it cannot repl y back without source link layer address.

(2) Identify the packet whether it is come from an associate node or not.

So each frame SHOULD contain originator IN-node link layer address regardless of AID value.

2.2 drawback of use of AID value for IN-node

(1) Due to the stateless auto configurability characteristics of IPv6 address, w e can configure IPv6 address from link-layer ID or Interface Identifier of a nod e and prefix ID of gateway. So use of AID for IN-node is illogical in presence (section 2.1) of IN-node link layer address in adaptation header.

(2) 16 bit short ID for a node in PAN was chosen to support 2^16 nodes in PAN. I f we use AID for IN-nodes, minimum length of AID field should be 16 bit. Still i t is larger and does not provide effective compression

(3) In PAN with multiple gateways and mobile IN-node, gateway may change frequen tly for IN-node. If we generate AID value of IN-node, it contains many AID value s. Therefore, each time node has to confirm gateway first and then select the co rresponding AID value. As the different gateway contains different AID value, in crease the chance of packet carries wrong source IPv6 address. Further, addition al management require handling the AID value at gateway and IN-node.

Above mentioned reasons (section 2.1 & 2.2) suggests that AID value for IN-no de IPv6 address SHOULD not use and link-layer ID of IN-node SHOULD be present in packet.

2.3. AID value Generation

In [I-D. Global connectivity in 6LoWPAN] author mentioned that new AID value for IPv6 address is generated by gateway. It works fine in static network and network with single gateways. But, in case of PAN with multiple gateways and mobile IN-node deal with multiple gateways, it leads certain problems. If new AID value is generated by gateway, different gateways generate different AID values for same OUT-node IPv6 address, so AID value updated with each gateways. Due to mobility, It is possible that packet reach at another gateway, but that contain different IPv6 address for corresponding AID value and wrong IPv6 destination address is embedded into the packet. But it is not possible when AID-value is only generated by IN-node because IN-node provides same AID value for corresponding IPv6 address to all the gateways. Thus, New AID value SHOULD be generated by IN-node only, thus different gateways and IN-node have same AID-values for corresponding IPv6 address.

2.4 AID field & AITT

Now it is clear that AID value SHOULD use for IPv6 address of OUT-node only. But the question is what will be the size of AID field and AITT format. Lets look at different possible scenario.

(1) PAN with single or few destinations

In many practical situations, data collected though sensors and send it to one central storage system, so all nodes within the PAN communicate only one or few node outside the PAN. In this scenario, AID table format shown in figure 2, is sufficient and efficient. As there are only few destinations, shorter AID field required.

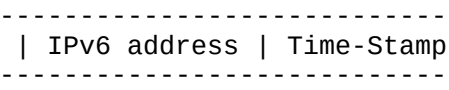


Figure 2 AITT without Link-Layer ID

(2) PAN, with multiple destinations

In this scenario, above mentioned table format can work, but due to larger no. of destinations, require larger AID field size. But we can reduce the no. of AID values requirement hence size of AID field by taking the AID value in combination with link-layer ID of IN-node (fig 3). In another term, maximum number of connections to OUT-node, from an IN-node is always less than or equal to connection from all IN-node. This scheme is particularly yielding when different IN-nodes or group of IN-nodes communicate with corresponding different OUT-nodes. It is also efficient for first scenario.

(3)Combination of link-layer ID with AID value in AITT increases the uniqueness of AID value in AITT (fig 3), and it is particularly helpful in PAN with multiple gateways and IN-node mobility scenario as well as it makes the AID management easier.

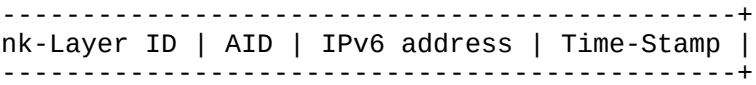


Figure 3. AITT without Link-Layer ID

3. AID messages & AID values mechanism

3.1 AID messages

3.1.1 AID update message

When, IN-node get AID request message (contain IPv6 address of OUT-node) from gateway, IN-node search for existing AID value for corresponding IPv6 address. If it does not present, IN-node generate a new AID value. IN-node sends Updated information to gateway through AID update message. AID update message contains AID value, IPv6 address, time-stamp and hope limit information. Similarly, when gateway is received IPv6 request message from IN-node, gateway reply back IPv6 address corresponding to AID value through AID update message.

3.1.2 AID request message (Gateway to IN-node)

When AID value does not exist for IPv6 address of IN-bound packet at gateway, it sends the AID request message to IN-node for AID value corresponding to IPv6 address. This message contains IPv6 address and in response, IN-node returns the corresponding AID update message. AID request message contains IPv6 address.

3.1.3 IPv6 request message

When AID value does not exist at gateway or IN-node on receiving AID frame, rece iving node sends IPv6 request message to sender to request IPv6 address correspo nding to AID value. In response, sender node return AID update message. IPv6 req uest message contains AID value.

3.2 Mechanism of AID value

3.2.1 For Out bound traffic

1. When IN-node wants to send packet to OUT-node, first it checks the existence of AID value for OUT-node IPv6 address in AITT.

2a. if AID value Present at IN-node for corresponding IPv6 address, it send the AID packet to gateway. But, if gateway does not have AID value, it sends IPv6 re quest message for corresponding AID value to IN-node, and IN-Node reply back AID Update message

2b. if AID value does not present at IN-node, it generates the new AID value for OUT-node IPv6 address and send AID update message to gateway.

3.2.2 For In bound traffic

1. When Gateway received the packet from OUT-node, it checks the existence of AI D value for OUT-node IPv6 address in AITT.

2a. if AID value present at gateway, it send the packet in AID frame to IN-node. But, if IN-node does not have AID value, it send request message to gateway for corresponding IPv6 address. Gateway reply backs the AID update message.

2b. if AID value does not present at gateway, it requests a AID value for given IPv6 address to IN-node, and IN-node reply back AID update message.

3.3 Time stamping & deletion of AID in AID-IPv6 translation table.

Whenever IN-Node generate AID, it also time-stamp the AID value simultaneously a nd send it with AID update message. Whenever transaction (during packet transmis sion) or updation take place in AITT, time-stamp field set back to initial value in correspond AID value. If AID value does not utilized for some threshold peri od, corresponding row is deleted.

4 Frame Format

4.1. 6lowpan TCP/IP Stake

In figure 4, TCP/IP stake shown for AID based 6lowpan. Physical and MAC layer ar e similar to IEEE 802.15.4 standards.

Adaptation layer lies above the MAC layer and use AID frame structure for OUT-no de (global communication) and Local frame structure for IN-node. Routing is take place at adaptation layer and mesh under & mesh over routing is an administrato r choice. in both case, packet has to reach at adaptation layer. Transport laye r mainly use compressed header format. Security layer is optional. Application l ayer keep at top above, and only required application are kept according to need .

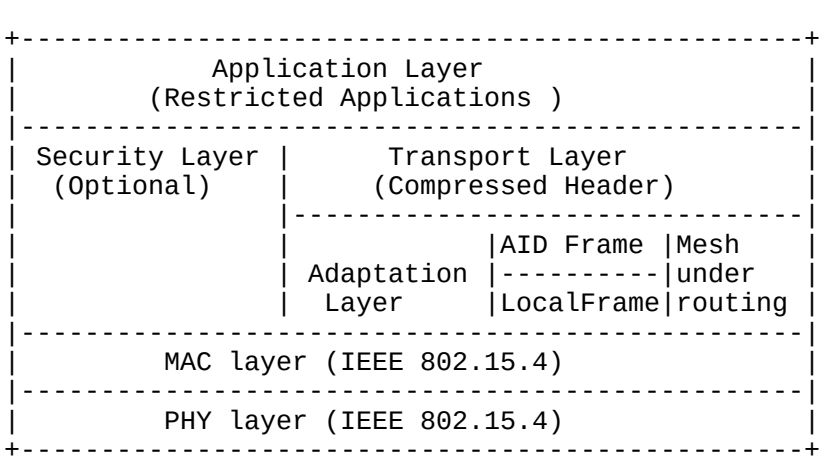


Figure 4. AID based 6lowpan TCP/IP stake

4.2 AID-IPv6 address Translation Table (AITT)
AITT translate the IPv6 address to corresponding AID value and vice versa (fig 5 & 6). It is present in IN-node as well as gateway, but AID frame to IPv6 packet and vice-versa translation take place at the gateway using AID-IPv6 table.

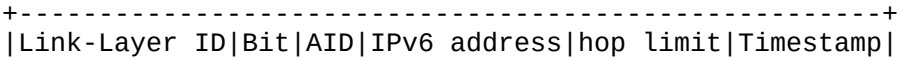


Figure 5. AITT for Gateway

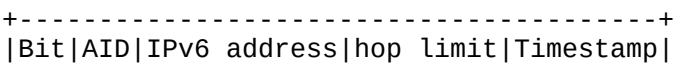


Figure 6. AITT for IN-Node

Link Layer ID of IN-nodes: 16 bits short ID or 64 bit Interface Identifier of IN-node

Bit: Length of AID field in bits (1,2,4,8 bit(s))

AID: AID value

IPv6 address: IPv6 address of corresponding OUT-Node

Hop limit: Hope limit for out bound traffic

Timestamp: Time of last use of AID

4.3. Adaptation Layer Header

Adaptation layer header contains Dispatch field, followed by AID or Local mesh frame and fragmentation header which is optional (fig 7). Dispatch value gives Idea about which type of frame following next (fig 8). Fragmentation header is optional, only present when payload is large and required fragmentation. It is according to [rfc 4944]

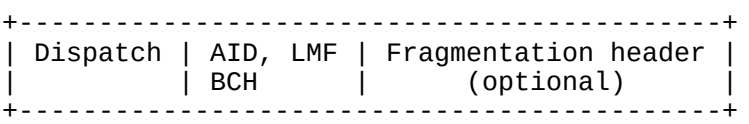


Figure 7. Adaptation layer header

4.3.1 Dispatch field

Dispatch field specify the frame type or field carried in to the adaptation header that follow after the dispatch.

Dispatch	Header Type
00 000000	NALP - Not a lowpan frame
01 000001	IPv6 -IPv6 uncompressed frame
01 000010	AID_1 -AID frame_1_bit_field_size
01 000011	AID_2 -AID frame_2_bit_field_size
01 000101	AID_3 -AID frame_4_bit_field_size
01 000110	AID_4 -AID frame_8_bit_field_size
*********	Reserved
10 100001	BCF - Broadcast Frame
10 100011	LMF - Local Mesh Frame
********	Reserved
01 111111	ESC - Additional dispatch byte follows

Figure 8. Dispatch Type

4.3.2 AID frame

Whenever communication takes place between the IN-node and OUT-node, AID frame i s used. Frame contains the AID value for corresponding IPv6 address.

01 000010		
01 000011	AID frame	Fragmentation header
01 000101	header	(optional)
01 000110		

Figure 9 (a). Dispatches for AID frame

Bound (1)	I (1)	G (1)	NH (4)	Fr (1)	hopeleft (4)	LL ID (16 or 64)	AID (1,2,3,8)

Figure 9 (b). AID frame Header

```
  Bound: 0-  Outbound packet from PAN (Forward to Gateway)
         1-  Inbound packet to PAN (Forward to IN-node at Link Layer ID address)

  Fr: 0- No fragmentation header follows
      1- fragmentation header follows
   I:  0-  16 bit short ID in II ID field
       1-  64 bit interface identifier in II ID field

   G:  0-  Any gateways
       1-  Gateway specified (next to the AID field)
   NH: First Bit
        0-  No Traffic class & flow lable
        1-  Traffic class & flow label field in Inline
       Second Bit
        0- no more header compression
        1- HC2 header compression bits [rfc draft]
       Third & Fourth Bits
        00- Additional header follow
        01- UDP
        10- ICMP
        11- TCP
  Hopeleft: (4 bits) Hope left within the PAN
  LL ID: 16 bit short ID or 64 bit Link Layer ID

  AID: AID value
```

4.3.3 If gateway specified (G set 1)

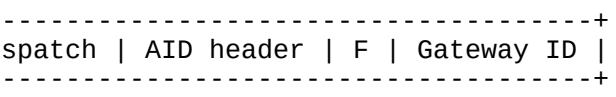


Figure 10. Gateway specified AID frame header

```
   F: 0- 16 bit address of Gateway
      1- 64 bit address of Gateway
   Gateway ID: 16 bits or 64 bits Address of Gateway
```

4.3.4 Local Mesh Frame

Whenever communication occurs between the IN-nodes, Local mesh Frame should use. As in this scenario, AID is not required.

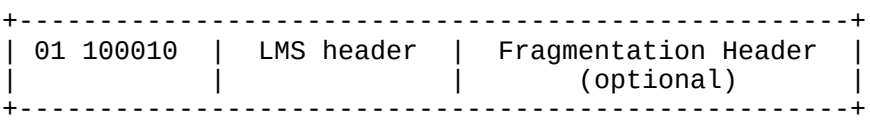


Figure 11. (a) Local mesh frame

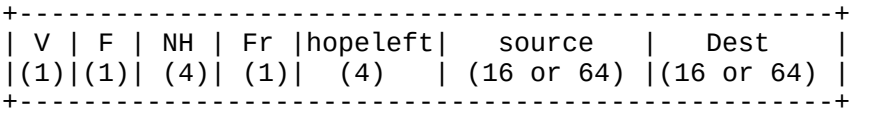


Figure 11. (b) LMS header

V: 0- 16 bit originator ID in source field
1- 64 bit EUI ID in source field

F: 0- 16 bit originator ID in Destination field
1- 64 bit EUI ID in Destination field

NH: Same as in section 4.3.2

Hopleft: Hop count (within the mesh)

Source: 16 bits short or 64 bits EUI address of originator IN-node

Dest: 16 bits short or 64 bits address of finaldestination IN-node

4.3.5 Local Broadcast frame

Whenever mesh routing required flooding mechanism, for that broadcast header is defined in figure xx. It contains dispatch type followed by sequence number of m essage.

Figure 12. Local Broadcast Header

Sequence No: This 8-bit field SHALL be incremented by the Originator whenever it sends a new mesh broadcast

5. Header compression efficiency.

During global communication, as per HC1 header compression [RFC 4944], maximum c ompression is 22 byte out of 40 byte. Further, 1 byte for dispatch and 5 byte fo r fragmentation header if presents. While in case of AID based global communicat ion, maximum compression is 3 byte and 5 bit out of 40 byte. Further, 1 byte for dispatch and 5 byte for fragmentation header if presents.

6. Formal Syntax

The following syntax specification uses the augmented Backus-Naur Form (BNF) as described in RFC-2234[RFC2234].

7. Security Considerations

TBD

8. IANA Considerations

TBD

9. References

9.1 Normative references

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

[ieee802.15.4] IEEE Computer Society, "IEEE Std. 802.15.4-2003", October 2003

[RFC4919] N. Kushalnagar, G. Montenegro, C. Schumacher, "IPv6 over Low-Power Wir eless Personal Area Networks (6LoWPANs): Overview, Assumptions, Problem Statemen t, and Goals", RFC4919, August 2007.

[RFC4944] G. Montenegro, N. Kushalnagar, J. Hui, D. Culler, "Transmission of IPv 6 Packets over IEEE 802.15.4 Networks", RFC4944, September 2007.

9.2 Informative Reference

[I-D. Global connectivity in 6LoWPAN] Hyun K. Kahng, Dae-In, Choi, Suyeon, Kim " Global connectivity in 6LoWPAN" draft-kahng-6lowpan-global-connectivity-00.txt, October, 2010

[RFC4862] Thomson, S., Narten, T., and T. Jinmei, "IPv6 Stateless Address Auto c onfiguration", RFC4862, September 2007

Authors' Address

Gohel Bakul Chandulal
Future Internet Team
National Institute for Mathematical Scinece
Daejeon, South Korea
E-mail: gohel@nims.re.kr

Dhananjay Singh
Future Internet Team
National Institute for Mathematical Scinece
Daejeon, South Korea
E-mail: singh@nims.re.kr

Acknowledgement
The work was supported by NAP of Korea Research Council of Fundamental Science a nd Technology.