Internet Engineering Task Force (IETF)
Request for Comments: 9034
Category: Standards Track
ISSN: 2070-1721

L. Thomas
C-DAC
S. Anamalamudi
SRM University-AP
S.V.R. Anand
M. Hegde
Indian Institute of Science
C. Perkins
Lupin Lodge

June 2021

Packet Delivery Deadline Time in the Routing Header for IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs)

Abstract

This document specifies a new type for the 6LoWPAN routing header containing the deadline time for data packets, designed for use over constrained networks. The deadline time enables forwarding and scheduling decisions for time-critical machine-to-machine (M2M) applications running on Internet-enabled devices that operate within time-synchronized networks. This document also specifies a representation for the deadline time values in such networks.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

Low-Power and Lossy Networks (LLNs) are likely to be deployed for real-time industrial applications requiring end-to-end delay guarantees [RFC8578]. A Deterministic Network ("DetNet") typically requires some data packets to reach their receivers within strict time bounds. Intermediate nodes use the deadline information to make appropriate packet forwarding and scheduling decisions to meet the time bounds.

This document specifies a new type for the Elective 6LoWPAN Routing Header (6LoRHE), Deadline-6LoRHE, so that the deadline time (i.e., the time of latest acceptable delivery) of data packets can be included within the 6LoRHE. [RFC8138] specifies the 6LoWPAN Routing Header (6LoRH), compression schemes for RPL (Routing Protocol for Low-Power and Lossy Networks) source routing [RFC6554], header compression of RPL packet information [RFC6553], and IP-in-IP encapsulation. This document also specifies the handling of the deadline time when packets traverse time-synchronized networks operating in different time zones or distinct reference clocks. Time-synchronization techniques are outside the scope of this document. There are a number of standards available for this purpose, including IEEE 1588 [IEEE.1588.2008], IEEE 802.1AS [IEEE.802.1AS.2011], IEEE 802.15.4-2015 Time-Slotted Channel Hopping (TSCH) [IEEE.802.15.4], and more.

The Deadline-6LoRHE can be used in any time-synchronized 6LoWPAN network. A 6TiSCH (IPv6 over the TSCH mode of IEEE 802.15.4) network is used to describe the implementation of the Deadline-6LoRHE, but this does not preclude its use in scenarios other than 6TiSCH. For instance, there is a growing interest in using 6LoWPAN over a Bluetooth Low Energy (BLE) mesh network [6LO-BLEMESH] in industrial IoT (Internet of Things) [IEEE-BLE-MESH]. BLE mesh time synchronization is being explored by the Bluetooth community. There are also cases under consideration in Wi-SUN [PHY-SPEC] [Wi-SUN].

2. Terminology

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

This document uses the terminology defined in [RFC6550] and [RFC9030].

3. 6LoRHE Generic Format

Note: this section is not normative and is included for convenience. The generic header format of the 6LoRHE is specified in [RFC8138]. Figure 1 illustrates the 6LoRHE generic format.

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

Figure 1: 6LoRHE Format

Length: Length of the 6LoRHE expressed in bytes, excluding the first 2 bytes. This enables a node to skip a 6LoRHE if the Type is not recognized or supported.

Type (variable length): Type of the 6LoRHE (see Section 7).

4. Deadline-6LoRHE

The Deadline-6LoRHE (see Figure 3) is a 6LoRHE [RFC8138] that provides the Deadline Time (DT) for an IPv6 datagram in a compressed form. Along with the DT, the header can include the Origination Time Delta (OTD) packet, which contains the time when the packet was enqueued for transmission (expressed as a value to be subtracted from DT); this enables a close estimate of the total delay incurred by a packet. The OTD field is initialized by the sender based on the current time at the outgoing network interface through which the packet is forwarded. Since the OTD is a delta, the length of the OTD field (i.e., OTL) will require fewer bits than the length of the DT field (i.e., DTL).

The DT field contains the value of the deadline time for the packet -- in other words, the time by which the application expects the packet to be delivered to the receiver.

```
   packet_deadline_time = packet_origination_time + max_delay
```

In order to support delay-sensitive, deterministic applications, all nodes within the network should process the Deadline-6LoRHE. The DT and OTD packets are represented in time units determined by a scaling parameter in the Routing Header. The Network ASN (Absolute Slot Number) can be used as a time unit in a time-slotted synchronized network (for instance, a 6TiSCH network, where global time is maintained in the units of slot lengths of a certain resolution).

The delay experienced by packets in the network is a useful metric for network diagnostics and performance monitoring. Whenever a packet crosses into a network using a different reference clock, the DT field is updated to represent the same deadline time, but expressed using the reference clock of the interface into the new network. Then the origination time is the same as the current time when the packet is transmitted into the new network, minus the delay already experienced by the packet, say 'current_dly'. In this way, within the newly entered network, the packet will appear to have originated 'current_dly' time units earlier with respect to the reference clock of the new network.

```
   new_network_origin_time = time_now_in_new_network - current_dly
```

The following example illustrates these calculations when a packet travels between three networks, each in a different time zone (TZ). 'x' can be 1, 2, or 3. Suppose that the deadline time as measured in TZ1 is 1050, and the origination time is 50. Suppose that the difference between TZ2 and TZ1 is 900, and the difference between TZ2 and TZ3 is 3600. In the figure, OT is the origination time as measured in the current time zone, and is equal to DT - OTD, that is, DT - 1000. Figure 2 uses the following abbreviations:

TxA: Time of arrival of packet in the network 'x'

TxD: Departure time of packet from the network 'x'

dlyx: Delay experienced by the packet in the previous network(s)

TZx: The time zone of network 'x'

```
            TZ1                      TZ2                    TZ3
  T1A=50|                 |                             |
        |----  dly1=50    |                             |
        |     \           |                             |
        |      \          |                             |
        |       \ T1D=100 |T2A=1000                     |
        |        -------->|-----           dly2=450     |
        |                 |     \                       |
        |                 |      \                      |
        |                 |       \          T2D=1400   | T3A=5000
        |                 |         ------------------->|---------->
        v                 v                             v

   dly0 = 0          dly1 = T1D-OT1      dly2 = T2D-OT2
                          = 100-50            = 1400 - 950
                          = 50                = 450

   OT1 = T1A-dly0     OT2 = T2A-dly1     OT3 = T3A-dly2
```

```
          = 50                 = 1000-50          = 5000 - 450
                               = 950              = 4550
```

Figure 2: Deadline Time Update Example

There are multiple ways that a packet can be delayed, including queuing delay, Media Access Control (MAC) layer contention delay, serialization delay, and propagation delay. Sometimes there are processing delays as well. For the purpose of determining whether or not the deadline has already passed, these various delays are not distinguished.

5. Deadline-6LoRHE Format

```
        0                   1                   2                   3
        0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |1|0|1| Length  |  6LoRH Type   |D| TU|  DTL  | OTL | BinaryPt  |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |      DT (variable length)     | OTD(variable length)(optional)|
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 3: Deadline-6LoRHE Format

Length (5 bits): Length represents the total length of the Deadline-6LoRHE Type measured in octets.

6LoRH Type: 7 (See Section 7.)

D flag (1 bit): The 'D' flag, set by the sender, qualifies the action to be taken when a 6LoWPAN Router (6LR) detects that the deadline time has elapsed.

If 'D' bit is 1, then the 6LR MUST drop the packet if the deadline time is elapsed.

If 'D' bit is 0, the packet MAY be forwarded on an exception basis, if the forwarding node is NOT in a situation of constrained resource, and if there are reasons to suspect that downstream nodes might find it useful (delay measurements, interpolations, etc.).

TU (2 bits): Indicates the time units for DT and OTD fields. The encodings for the DT and OTD fields use the same time units and precision.

```
00  Time represented in seconds and fractional seconds
01  Reserved
10  Network ASN
11  Reserved
```

DTL (4 bits): Length of the DT field as an unsigned 4-bit integer, encoding the length of the field in hex digits, minus one.

OTL (3 bits): Length of the OTD field as an unsigned 3-bit integer, encoding the length of the field in hex digits. If OTL == 0, the

OTD field is not present. The value of OTL MUST NOT exceed the value of DTL plus one.

For example, DTL = 0b0000 means the DT field in the 6LoRHE is 1 hex digit (4 bits) long. OTL = 0b111 means the OTD field is 7 hex digits (28 bits) long.

BinaryPt (6 bits): If zero, the number of bits of the integer part the DT is equal to the number of bits of the fractional part of the DT. If nonzero, the BinaryPt is a (2's complement) signed integer determining the position of the binary point within the value for the DT. This allows BinaryPt to be within the range [-32,31].

* If BinaryPt value is positive, then the number of bits for the integer part of the DT is increased by the value of BinaryPt, and the number of bits for the fractional part of the DT is correspondingly reduced. This increases the range of DT.

* If BinaryPt value is negative, then the number of bits for the integer part of the DT is decreased by the value of BinaryPt, and the number of bits for the fractional part of the DT is correspondingly increased. This increases the precision of the fractional seconds part of DT.

DT Value (4..64 bits): An unsigned integer of DTL+1 hex digits giving the DT value.

OTD Value (4..28 bits): If present, an unsigned integer of OTL hex digits giving the origination time as a negative offset from the DT value.

Whenever a sender initiates the IP datagram, it includes the Deadline-6LoRHE along with other 6LoRH information. For information about the time-synchronization requirements between sender and receiver, see Section 8.

For the chosen time unit, a compressed time representation is available as follows. First, the application on the originating node determines how many time bits are needed to represent the difference between the time at which the packet is launched and the deadline time, including the representation of fractional time units. That number of bits (say, N_bits) determines DTL as follows:

DTL = ((N_bits - 1) / 4)

The number of bits determined by DTL allows the counting of any number of fractional time units in the range of interest determined by DT and the OT. Denote this number of fractional time units to be Epoch_Range(DTL) (i.e., Epoch_Range is a function of DTL):

Epoch_Range(DTL) = 2^(4*(DTL+1))

Each point of time between OT and DT is represented by a time unit and a fractional time unit; in this section, this combined representation is called a rational time unit (RTU). 1 RTU measures the smallest fractional time that can be represented between two points of time in the epoch (i.e., within the range of interest).

DT - OT cannot exceed 2^(4*(DTL+1)) == 16^(DTL+1). A low value of DTL leads to a small Epoch_Range; if DTL = 0, there will only be 16 RTUs within the Epoch_Range (i.e., Epoch_Range(DTL) = 16^1) for any TU. The values that can be represented in the current epoch are in the range [0, (Epoch_Range(DTL) - 1)].

Assuming wraparound does not occur, OT is represented by the value (OT mod Epoch_Range), and DT is represented by the value (DT mod Epoch_Range). All arithmetic is to be performed modulo (Epoch_Range(DTL)), yielding only positive values for DT - OT.

In order to allow fine-grained control over the setting of the deadline time, the fields for DT and OTD use fractional seconds. This is done by specifying a binary point, which allocates some of the bits for fractional times. Thus, all such fractions are restricted to be negative powers of 2. Each point of time between OT and DT is then represented by a time unit (either seconds or ASNs) and a fractional time unit.

Let OT_abs, DT_abs, and CT_abs denote the true (absolute) values (on the synchronized timelines) for OT, DT, and current time. Let N be the number of bits to be used to represent the integer parts of OT_abs, DT_abs, and CT_abs:

      N = {4*(DTL+1)/2} + BinaryPt

The originating node has to pick a segment size (2^N) so that DT_abs - OT_abs < 2^N, and so that intermediate network nodes can detect whether or not CT_abs > DT_abs.

Given a value for N, the value for DT is represented in the deadline-time format by DT = (DT_abs mod 2^N). DT is typically represented as a positive value (even though negative modular values make sense). Also, let OT = OT_abs mod 2^N and CT = CT_abs mod 2^N, where both OT and CT are also considered as non-negative values.

When the packet is launched by the originating node, CT_abs == OT_abs and CT == OT. Given a particular value for N, then in order for downstream nodes to detect whether or not the deadline has expired (i.e., whether DT_abs > CT_abs), the following is required:

      Assumption 1: DT_abs - OT_abs < 2^N.

Otherwise the ambiguity inherent in the modulus arithmetic yielding OT and DT will cause failure: one cannot measure time differences greater than 2^N using numbers in a time segment of length less than 2^N.

Under Assumption 1, downstream nodes must effectively check whether or not their current time is later than the DT -- but the value of the DT has to be inferred from the value of DT in the 6LoRHE, which is a number less than 2^N. This inference cannot be expected to reliably succeed unless Assumption 1 is valid, which means that the originating node has to be careful to pick proper values for DTL and for BinaryPt.

Since OT is not necessarily provided in the 6loRHE, there may be a danger of ambiguity. Surely, when DT = CT, the deadline time is expiring and the packet should be dropped. However, what if an intermediate node measures that CT = DT+1? Was the packet launched a short time before arrival at the intermediate node, or has the current time wrapped around so that CT_abs - OT_abs > 2^N?

In order to solve this problem, a safety margin has to be provided, in addition to requiring that DT_abs - OT_abs < 2^N. The value of this safety margin is proportional to 2^N and is determined by a new parameter, called the "SAFETY_FACTOR". Then, for safety the originating node MUST further ensure that (DT_abs - OT_abs) < 2^N*(1-SAFETY_FACTOR).

Each intermediate node that receives the packet with the Deadline-6LoRHE must determine whether ((CT - DT) mod 2^N) > SAFETY_FACTOR*2^N. If this test condition is not satisfied, the deadline time has expired. See Appendix A for more explanation about the test condition. All nodes that receive a packet with a Deadline-6LoRHE included MUST use the same value for the SAFETY_FACTOR. The SAFETY_FACTOR is to be chosen so that a packet with the Deadline-6LoRHE included will be tested against the current time at least once during every subinterval of length SAFETY_FACTOR*2^N. In this way, it can be guaranteed that the packet will be tested often enough to make sure it can be dropped whenever CT_abs > DT_abs. The value of SAFETY_FACTOR is specified in this document to be 20%.

> Example: Consider a 6TiSCH network with time-slot length of 10 ms. Let the time units be ASNs (TU == (binary)0b10). Let the current ASN when the packet is originated be 54400, and the maximum allowable delay (max_delay) for the packet delivery be 1 second from the packet origination, then:

```
      deadline_time = packet_origination_time + max_delay

         = 0xD480 + 0x64 (Network ASNs)

         = 0xD4E4 (Network ASNs)

      Then, the Deadline-6LoRHE encoding with nonzero OTL is:

         DTL = 3, OTL = 2, TU = 0b10, BinaryPt = 8, DT = 0xD4E4,
         OTD = 0x64
```

6. Deadline-6LoRHE in Three Network Scenarios

In this section, the Deadline-6LoRHE operation is described for three network scenarios. Figure 4 depicts a constrained time-synchronized LLN that has two subnets, N1 and N2, connected through 6LoWPAN Border Routers (6LBRs) [RFC8929] with different reference clock times, T1 and T2.

```
                          +-------------------+
                          | Time-Synchronized |
                          |      Network      |
                          +---------+---------+
                                    |
                                    |
                                    |
                     +--------------+--------------+
                     |                             |
                  +-----+                       +-----+
                  |     | Backbone              |     | Backbone
             o    |     | router                |     | router
                  +-----+                       +-----+
             o                  o               o
                 o   o   o               o  o   o  o   o  o
            o      LLN    o                o  LLN   o  o
               o   o    o      o             o o o     o  o
         6LoWPAN Network (subnet N1)   6LoWPAN Network (subnet N2)
```

Figure 4: Intra-Network Time Zone Scenario

6.1. Scenario 1: Endpoints in the Same DODAG (N1)

In Scenario 1, shown in Figure 5, the Sender 'S' has an IP datagram to be routed to a Receiver 'R' within the same Destination-Oriented Directed Acyclic Graph (DODAG). For the route segment from the sender to the 6LBR, the sender includes a Deadline-6LoRHE by encoding the deadline time contained in the packet. Subsequently, each 6LR will perform hop-by-hop routing to forward the packet towards the 6LBR. Once the 6LBR receives the IP datagram, it sends the packet downstream towards 'R'.

In the case of a network running in RPL non-storing mode, the 6LBR generates an IPv6-in-IPv6 encapsulated packet when sending the packet downwards to the receiver [RFC9008]. The 6LBR copies the Deadline-6LoRHE from the sender-originated IP header to the outer IP header. The Deadline-6LoRHE contained in the inner IP header is removed.

```
                              +-------+
                   ^          | 6LBR  |       |
                   |          |       |       |
                   |          +-------+       |
           Upward  |         /      /| \      | Downward
           routing |      (F)      /  |  \     | routing
                   |     /  \    (C) |  (D)   |
                   |    /    \    |  | / |\   |
                   |  (A)    (B)  : (E)  : R  |
                   |  /|\     | \   / \       |
                   | S : :    : :  :  :       v
```

Figure 5: Endpoints within the Same DODAG (Subnet N1)

At the tunnel endpoint of the encapsulation, the Deadline-6LoRHE is copied back from the outer header to inner header, and the inner IP packet is delivered to 'R'.

6.2. Scenario 2: Endpoints in Networks with Dissimilar L2 Technologies

In Scenario 2, shown in Figure 6, the Sender 'S' (belonging to DODAG 1) has an IP datagram to be routed to a Receiver 'R' over a time-synchronized IPv6 network. For the route segment from 'S' to 6LBR, 'S' includes a Deadline-6LoRHE. Subsequently, each 6LR will perform hop-by-hop routing to forward the packet towards the 6LBR. Once the deadline time information reaches the 6LBR, the packet will be encoded according to the mechanism prescribed in the other time-synchronized network depicted as "Time-Synchronized Network" in Figure 6. The specific data encapsulation mechanisms followed in the new network are beyond the scope of this document.

```
                              +----------------+
                              | Time-          |
                              | Synchronized   |------R
                              | Network        |
                              +----------------+
                                      |
                                      |
                            ----------+-----------
                     ^                |
                     |            +---+---+
                     |            | 6LBR  |
            Upward   |            |       |
            routing  |            +------++
                     |        (F)/      /| \
                     |       /  \      / |  \
                     |      /    \   (C) |  (D)
                     |    (A)    (B)  |  | / |\
                     |    /|\     |\  : (E)  : :
                     |   S : :    : :   / \
                                       :   :
```

Figure 6: Packet Transmission in Dissimilar L2 Technologies or Internet

For instance, the IP datagram could be routed to another time-synchronized, deterministic network using the mechanism specified in In-situ Operations, Administration, and Maintenance (IOAM) [IOAM-DATA], and then the deadline time would be updated according to the measurement of the current time in the new network.

6.3. Scenario 3: Packet Transmission across Different DODAGs (N1 to N2)

Consider the scenario depicted in Figure 7, in which the Sender 'S' (belonging to DODAG 1) has an IP datagram to be sent to Receiver 'R' belonging to another DODAG (DODAG 2). The operation of this scenario can be decomposed into a combination of Scenarios 1 and 2. For the route segment from 'S' to 6LBR1, 'S' includes the Deadline-6LoRHE. Subsequently, each 6LR will perform hop-by-hop operations to forward the packet towards 6LBR1. Once the IP datagram reaches 6LBR1 of DODAG1, 6LBR1 applies the same rule as described in Scenario 2 while routing the packet to 6LBR2 over a (likely) time-synchronized wired backhaul. The wired side of 6LBR2 can be mapped to the receiver of Scenario 2. Once the packet reaches 6LBR2, it updates the Deadline-6LoRHE by adding or subtracting the difference of time of DODAG2 and sends the packet downstream towards 'R'.

```
                    Time-Synchronized Network
                  -+---------------------------+-
                   |                           |
      DODAG1   +---+---+                   +---+---+   DODAG2
               | 6LBR1 |                   | 6LBR2 |
               |       |                   |       |
               +-------+                   +-------+
           (F)/      /| \              (F)/      /| \
          /  \      /  |  \           /  \      /  |  \
         /    \   (C)  |  (D)        /    \   (C)  |  (D)
       (A)    (B)  |  | / |\       (A)    (B)  |  |   |\
       /|\     |\  : (E)  : :      /|\     |\  : (E)  : :
      S : :    : :   / \          : : :    : :   / \
                    :   :                       :   R
   Network N1, time zone T1      Network N2, time zone T2
```

Figure 7: Packet Transmission in Different DODAGs (N1 to N2)

Consider an example of a 6TiSCH network in which S in DODAG1 generates the packet at ASN 20000 to R in DODAG2. Let the maximum allowable delay be 1 second. The time-slot length in DODAG1 and DODAG2 is assumed to be 10 ms. Once the deadline time is encoded in Deadline-6LoRHE, the packet is forwarded to 6LBR1 of DODAG1. Suppose the packet reaches 6LBR1 of DODAG1 at ASN 20030.

current_time = ASN at 6LBR * slot_length_value

remaining_time = deadline_time - current_time

= ((packet_origination_time + max_delay) - current time)

= (20000 + 100) - 20030

= 30 (in Network ASNs)

= 30 * 10^3 milliseconds

Once the deadline time information reaches 6LBR2, the packet will be encoded according to the mechanism prescribed in the other time-synchronized network.

7. IANA Considerations

This document defines a new Elective 6LoWPAN Routing Header Type, and IANA has assigned the value 7 from the 6LoWPAN Dispatch Page 1 number space for this purpose.

Value	Description	Reference
7	Deadline-6LoRHE	RFC 9034

Table 1: Entry in the "Elective

6LoWPAN Routing Header Type"
Registry

8. Synchronization Aspects

The document supports time representation of the deadline and origination times carried in the packets traversing networks of different time zones having different time-synchronization mechanisms. For instance, in a 6TiSCH network where the time is maintained as ASN time slots, the time synchronization is achieved through beaconing among the nodes as described in [RFC7554]. There could be 6lo networks that employ NTP where the nodes are synchronized with an external reference clock from an NTP server. The specification of the time-synchronization method that needs to be followed by a network is beyond the scope of the document.

The number of hex digits chosen to represent DT, and the portion of that field allocated to represent the integer number of seconds, determines the meaning of t_0, i.e., the meaning of DT == 0 in the chosen representation. If DTL == 0, then there are only 4 bits that can be used to count the time units, so that DT == 0 can never be more than 16 time units (or fractional time units) in the past. This then requires that the time synchronization between sender and receiver has to be tighter than 16 units. If the binary point were moved so that all the bits were used for fractional time units (e.g., fractional seconds or fractional ASNs), the time-synchronization requirement would be correspondingly tighter.

A 4-bit field for DT allows up to 16 hex digits, which is 64 bits. That is enough to represent the NTP 64-bit timestamp format [RFC5905], which is more than enough for the purposes of establishing deadline times. Unless the binary point is moved, this is enough to represent time since year 1900.

For example, suppose that DTL = 0b0000 and the DT bits are split evenly; then we can count up to 3.75 seconds by quarter-seconds.

If DTL = 3 and the DT bits are again split evenly, then we can count up to 256 seconds (in steps of 1/256 of a second).

In all cases, t_0 is defined as specified in Section 5.

t_0 = [current_time - (current_time mod (2^(4*(DTL+1))))]

regardless of the choice of TU.

For TU = 0b00, the time units are seconds. With DTL == 15, and BinaryPt == 0, the epoch is (by default) January 1, 1900, at 00:00 UTC. The resolution is then 2^-32 seconds, which is the maximum possible. This time format wraps around every 2^32 seconds, which is roughly 136 years.

For TU = 0b10, the time units are ASNs. The start time is relative, and updated by a mechanism that is out of scope for this document. With 10 ms slots, DTL = 15, and BinaryPt == 0, it would take over a year for the ASN to wrap around. Typically, the number of hex digits allocated for TU = 0b10 would be less than 15.

9. Security Considerations

The security considerations of [RFC4944] (Section 13), [RFC6282] (Section 6), and [RFC6553] (Section 5) apply. Using a compressed format as opposed to the full inline format is logically equivalent and does not create an opening for a new threat when compared to [RFC6550], [RFC6553], and [RFC6554].

The protocol elements specified in this document are designed to work in controlled operational environments (e.g., industrial process control and automation). In order to avoid misuse of the deadline information that could potentially result in a Denial of Service (DoS) attack, proper functioning of this deadline time mechanism requires the provisioning and management of network resources for supporting traffic flows with deadlines, performance monitoring, and admission control policy enforcement. The network provisioning can be done either centrally or in a distributed fashion. For example, tracks in a 6TiSCH network could be established by a centralized Path Computation Element (PCE), as described in the 6TiSCH architecture [RFC9030].

The security considerations of DetNet architecture [RFC8655] (Section 5) mostly apply to this document as well, as follows. To secure the request and control of resources allocated for tracks, authentication and authorization can be used for each device and network controller devices. In the case of distributed control protocols, security is expected to be provided by the security properties of the protocols in use.

The identification of deadline-bearing flows on a per-flow basis may provide attackers with additional information about the data flows compared to networks that do not include per-flow identification. The security implications of disclosing that additional information deserve consideration when implementing this deadline specification.

Because of the requirement of precise time synchronization, the accuracy, availability, and integrity of time synchronization is of critical importance. Extensive discussion of this topic can be found in [RFC7384].

10. References

10.1. Normative References

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

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

[RFC5905] Mills, D., Martin, J., Ed., Burbank, J., and W. Kasch, "Network Time Protocol Version 4: Protocol and Algorithms Specification", RFC 5905, DOI 10.17487/RFC5905, June 2010, <https://www.rfc-editor.org/info/rfc5905>.

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

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

[RFC6553] Hui, J. and JP. Vasseur, "The Routing Protocol for Low-Power and Lossy Networks (RPL) Option for Carrying RPL Information in Data-Plane Datagrams", RFC 6553, DOI 10.17487/RFC6553, March 2012, <https://www.rfc-editor.org/info/rfc6553>.

[RFC6554] Hui, J., Vasseur, JP., Culler, D., and V. Manral, "An IPv6 Routing Header for Source Routes with the Routing Protocol for Low-Power and Lossy Networks (RPL)", RFC 6554, DOI 10.17487/RFC6554, March 2012, <https://www.rfc-editor.org/info/rfc6554>.

[RFC7384] Mizrahi, T., "Security Requirements of Time Protocols in Packet Switched Networks", RFC 7384, DOI 10.17487/RFC7384, October 2014, <https://www.rfc-editor.org/info/rfc7384>.

[RFC7554] Watteyne, T., Ed., Palattella, M., and L. Grieco, "Using IEEE 802.15.4e Time-Slotted Channel Hopping (TSCH) in the Internet of Things (IoT): Problem Statement", RFC 7554, DOI 10.17487/RFC7554, May 2015, <https://www.rfc-editor.org/info/rfc7554>.

[RFC8138] Thubert, P., Ed., Bormann, C., Toutain, L., and R. Cragie, "IPv6 over Low-Power Wireless Personal Area Network (6LoWPAN) Routing Header", RFC 8138, DOI 10.17487/RFC8138, April 2017, <https://www.rfc-editor.org/info/rfc8138>.

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

[RFC8655] Finn, N., Thubert, P., Varga, B., and J. Farkas, "Deterministic Networking Architecture", RFC 8655, DOI 10.17487/RFC8655, October 2019, <https://www.rfc-editor.org/info/rfc8655>.

[RFC9030] Thubert, P., Ed., "An Architecture for IPv6 over the Time-Slotted Channel Hopping Mode of IEEE 802.15.4 (6TiSCH)", RFC 9030, DOI 10.17487/RFC9030, May 2021,

<https://www.rfc-editor.org/info/rfc9030>.

10.2. Informative References

[6LO-BLEMESH] Gomez, C., Darroudi, S. M., Savolainen, T., and M. Spoerk, "IPv6 Mesh over BLUETOOTH(R) Low Energy using IPSP", Work in Progress, Internet-Draft, draft-ietf-6lo-blemesh-10, 22 April 2021, <https://tools.ietf.org/html/draft-ietf-6lo-blemesh-10>.

[IEEE-BLE-MESH] Leonardi, L., Patti, G., and L. Lo Bello, "Multi-Hop Real-Time Communications Over Bluetooth Low Energy Industrial Wireless Mesh Networks", IEEE Access, Vol 6, pp. 26505-26519, DOI 10.1109/ACCESS.2018.2834479, May 2018, <https://doi.org/10.1109/ACCESS.2018.2834479>.

[IEEE.1588.2008] IEEE, "IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems", DOI 10.1109/IEEESTD.2008.4579760, July 2008, <https://doi.org/10.1109/IEEESTD.2008.4579760>.

[IEEE.802.15.4] IEEE, "IEEE Standard for Low-Rate Wireless Networks", IEEE Standard 802.15.4-2015, DOI 10.1109/IEEESTD.2016.7460875, April 2016, <https://ieeexplore.ieee.org/document/7460875>.

[IEEE.802.1AS.2011] IEEE, "IEEE Standard for Local and Metropolitan Area Networks - Timing and Synchronization for Time-Sensitive Applications in Bridged Local Area Networks", IEEE Std 802.1AS-2011, DOI 10.1109/IEEESTD.2011.5741898, March 2011, <https://doi.org/10.1109/IEEESTD.2011.5741898>.

[IOAM-DATA] Brockners, F., Ed., Bhandari, S., Ed., and T. Mizrahi, Ed., "Data Fields for In-situ OAM", Work in Progress, Internet-Draft, draft-ietf-ippm-ioam-data-12, 21 February 2021, <https://tools.ietf.org/html/draft-ietf-ippm-ioam-data-12>.

[PHY-SPEC] Wi-SUN Alliance, "Wi-SUN PHY Specification V1.0", March 2016, <http://wi-sun.org>.

[RFC8578] Grossman, E., Ed., "Deterministic Networking Use Cases", RFC 8578, DOI 10.17487/RFC8578, May 2019, <https://www.rfc-editor.org/info/rfc8578>.

[RFC8929] Thubert, P., Ed., Perkins, C.E., and E. Levy-Abegnoli, "IPv6 Backbone Router", RFC 8929, DOI 10.17487/RFC8929, November 2020, <https://www.rfc-editor.org/info/rfc8929>.

[RFC9008] Robles, M.I., Richardson, M., and P. Thubert, "Using RPI

Option Type, Routing Header for Source Routes, and IPv6-in-IPv6 Encapsulation in the RPL Data Plane", RFC 9008, DOI 10.17487/RFC9008, April 2021, <https://www.rfc-editor.org/info/rfc9008>.

[Wi-SUN] Harada, H., Mizutani, K., Fujiwara, J., Mochizuki, K., Obata, K., and R. Okumura, "IEEE 802.15.4g Based Wi-SUN Communication Systems", IEICE Transactions on Communications, Volume E100.B, Issue 7, pp. 1032-1043, DOI 10.1587/transcom.2016SCI0002, January 2017, <https://doi.org/10.1587/transcom.2016SCI0002>.

Appendix A. Modular Arithmetic Considerations

Graphically, one might visualize the timeline as follows:

```
              OT_abs        CT_abs        DT_abs
         -------|-------------|-------------|------------------>
```

Figure 8: Absolute Timeline Representation

In Figure 8, the value of CT_abs is envisioned as traveling to the right as time progresses, getting farther away from OT_abs and getting closer to DT_abs. The timeline is considered to be subdivided into time subintervals [i,j] starting and ending at absolute times equal to k*(2^N), for integer values of k. Let I_k = k*(2^N) and I_(k+1) = (k+1)*2^N. Intervals starting at I_k and I_(k+1) may occur at various placements in the above timeline. Even though OT_abs is _always_ less than DT_abs, it could be that DT < OT because of the way that DT and OT are represented within the range [0, 2^N) and similarly for CT_abs and CT compared to OT and DT.

Representing the above situation in time segments of length 2^N (and values OT, CT, DT) results in several cases where the deadline time has not elapsed:

1) OT < CT < DT (e.g., I_k < OT_abs < CT_abs < DT_abs < I_(k+1))

2) DT < OT < CT (e.g., I_k < OT_abs < CT_abs < I_(k+1) < DT_abs)

3) CT < DT < OT (e.g., I_k < OT_abs < I_(k+1) < CT_abs < DT_abs)

In the following cases, the deadline time has elapsed and the packet should be dropped.

4) DT < CT < OT

5) OT < DT < CT

6) CT < OT < DT

Again in Figure 8, consider CT_abs as time moving away from OT_abs and towards DT_abs. For times CT_abs before the expiration of the deadline time, we also have CT_abs - OT_abs == CT - OT mod 2^N and similarly for DT_abs - CT_abs.

As time proceeds, DT_abs - CT_abs gets smaller. When the deadline time expires, DT_abs - CT_abs begins to grow negative. A proper selection for SAFETY_FACTOR allows it to go _slightly negative_ but for an intermediate point to _detect_ that it has gone negative. Note that in modular arithmetic, "slightly negative" means _exactly_ the same as "almost as large as the modulus (i.e., 2^N)". Now consider the test condition ((CT - DT) mod 2^N) > SAFETY_FACTOR*2^N.

(DT_abs - OT_abs) < 2^N*(1-SAFETY_FACTOR) satisfies the test condition when CT_abs == OT_abs (i.e., when the packet is launched). In modular arithmetic, 2^N*(1-SAFETY_FACTOR) == 2^N - 2^N*SAFETY_FACTOR == -2^N*(SAFETY_FACTOR). Then DT_abs - OT_abs < -2^N*(1-SAFETY_FACTOR). Inverting the inequality, OT_abs - DT_abs > 2^N*(1-SAFETY_FACTOR), and thus at launch CT_abs - DT_abs > 2^N*(1-SAFETY_FACTOR).

As CT_abs grows larger, CT_abs - DT_abs gets LARGER in (non-negative) modular arithmetic until the time at which CT_ABS == DT_ABS, and suddenly CT_ABS - DT_abs becomes zero. Also suddenly, the test condition is no longer fulfilled.

As CT_abs grows still larger, CT_abs > DT_abs, and we need to detect this condition as soon as possible. Requiring the SAFETY_FACTOR enables this detection until CT_abs exceeds DT_abs by an amount equal to SAFETY_FACTOR*2^N.

A note about "inverting the inequality". Observe that a < b implies that -a > -b on the real number line. Also, (a - b) == -(b - a). These facts hold also for modular arithmetic.

During the times prior to the expiration of the deadline, for Safe = 2^N*SAFETY_FACTOR we have:

      (DT_abs - 2^N) < OT_abs < CT_abs < DT_abs < DT_abs+Safe

Naturally, DT_abs - 2^N == DT_abs mod 2^N == DT.

Again considering Figure 8, it is easy to see that {CT_abs - (DT_abs - 2^N)} gets larger and larger until the time at which CT_abs = DT_abs, which is the first time at which CT - DT == 0 mod 2^N. As CT_abs increases past the deadline time, 0 < CT_abs - DT_abs < Safe. In this range, any intermediate node can detect that the deadline has expired. As CT_abs increases past DT_abs+Safe, it is no longer possible for an intermediate node to determine with certainty whether or not the deadline time has expired. These statements also apply when reduced to modular arithmetic in the modulus 2^N.

In particular, the test condition no longer allows detection of deadline expiration when the current time becomes later than (DT_abs+Safe). In order to maintain correctness even for packets that are forwarded after expiration (i.e., the 'D' flag), N has to be chosen to be so large that the test condition will not fail -- i.e., that in all scenarios of interest, the packet will be dropped before the current time becomes equal to DT_abs+2^N*SAFETY_FACTOR.

Acknowledgments

The authors thank Pascal Thubert for suggesting the idea and encouraging the work. Thanks to Shwetha Bhandari's suggestions, which were instrumental in extending the timing information to heterogeneous networks. The authors acknowledge the 6TiSCH WG members for their inputs on the mailing list. Special thanks to Jerry Daniel, Dan Frost (Routing Directorate), Charlie Kaufman (Security Directorate), Seema Kumar, Tal Mizrahi, Avinash Mohan, Shalu Rajendran, Anita Varghese, and Dale Worley (General Area Review Team (Gen-ART) review) for their support and valuable feedback.

Authors' Addresses

Lijo Thomas
Centre for Development of Advanced Computing
Vellayambalam
Trivandrum 695033
India

Email: lijo@cdac.in

Satish Anamalamudi
SRM University-AP
Amaravati Campus
Amaravati, Andhra Pradesh 522 502
India

Email: satishnaidu80@gmail.com

S.V.R. Anand
Indian Institute of Science
Bangalore 560012
India

Email: anandsvr@iisc.ac.in

Malati Hegde
Indian Institute of Science
Bangalore 560012
India

Email: malati@iisc.ac.in

Charles E. Perkins
Lupin Lodge
20600 Aldercroft Heights Rd.
Los Gatos, CA 95033
United States of America

Email: charliep@computer.org