An Alternative Delta Time encoding for CCNx using Interval Time from RFC5497
draft-gundogan-icnrg-ccnx-timetlv-00

Abstract

CCNx utilizes Delta Time for a number of functions. When using CCNx in environments with constrained nodes and/or bandwidth constrained networks, it is valuable to have a compressed representation of delta time. In order to do so, either accuracy or dynamic range has to be sacrificed. Since the current uses of delta time do not require both simultaneously, one can consider a logarithmic encoding such as that specified in RFC5497. This document updates _CCNx messages in TLV Format_ (RFC8609) to specify this alternative encoding.

Status of This Memo

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

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

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

This Internet-Draft will expire on May 7, 2020.

Copyright Notice

Copyright (c) 2019 IETF Trust and the persons identified as the document authors. All rights reserved.
1. Introduction

CCNx utilizes Time values for a number of functions. Some of these are expressed as absolute time, others as delta time. When using CCNx in environments with constrained nodes and/or bandwidth constrained networks, it is valuable to have a compact representation of time values. For example [I-D.irtf-icnrg-icnlowpan] specifies a compression scheme useful over IEEE 804.15.4 networks. However, any compact time representation has to sacrifice either accuracy or dynamic range or both. For some time uses this is relatively straightforward to achieve, for other uses, it is not. This document discusses the various cases, and proposes a compact encoding that can be considered for some uses of absolute time.
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].

3. Usage of Time values CCNx

3.1. Relative Time usage in CCNx

CCNx, as currently specified in [RFC8569], utilizes delta time for only the lifetime of an Interest message (sections 2.1, 2.2, 2.4.2, 10.3). It is a hop-by-hop header value, and is currently encoded via the T_INTLIFE TLV as a 64-bit integer ([RFC8609] section 3.4.1). While formally an optional TLV, in all but some corner cases every Interest Message is expected to carry the Interest Lifetime TLV, and hence having compact encoding is particularly valuable for keeping Interest messages short.

Since the current uses of delta time do not require both accuracy and dynamic range simultaneously, one can consider a logarithmic encoding such as that specified in [RFC5497] and outlined in Section 4. This document updates CCNx messages in TLV Format ([RFC8609]) to permit this alternative encoding for selected Time values. See Section 6 for the specific actions needed to register this alternative compact representation of Interest Lifetime.

3.2. Absolute Time usage in CCNx

CCNx, as currently specified in [RFC8569], utilizes absolute time for various important functions. Each of these absolute time usages poses a different challenge for a compact representation. These are discussed in the following subsections.

3.2.1. Signature Time

_Signature Time_ is the time the signature of a content object was generated (sections 8.2-8.4 [RFC8569]). This is a content message TLV stating the time the signature on the content object was generated, in milliseconds since the UTC epoch (i.e. An NTP timestamp). It is currently encoded via the T_SIGTIME TLV as a 64-bit unsigned integer (see section 3.6.4.1.4.5 [RFC8609]).

Given that a signature time could be essentially at any time in the past, and is included in the hash securing the content object, it seems there is no practical way to define an alternative compact encoding that preserves its semantics and security properties; hence we don’t consider it further as a candidate.
3.2.2. Expiry Time

_Expiry Time_ indicates the expiry time of a content object (section 4 [RFC8569]). This is a content message TLV (covered by the hash and signature of the content object) stating the expiration time of the content object in milliseconds since the UTC epoch (i.e. An NTP timestamp). It is currently encoded via the T_EXPIRY TLV as a 64-bit unsigned integer (see section 3.6.2.2.2 [RFC8609]).

Expiry time could be in the past, or in the future, potentially by a large delta, and is included in the hash securing the content object. Therefore, it seems there is no practical way to define an alternative compact encoding that preserves its semantics and security properties; hence we don’t consider it further as a candidate.

3.2.3. Recommended Cache Time

_Recommended Cache Time_ (RCT) for a content object (see section 4 [RFC8569]) is a hop-by-hop header stating the expiration time for a cached content object in milliseconds since the UTC epoch (i.e. An NTP timestamp). It is currently encoded via the T_CACHETIME TLV as a 64-bit unsigned integer (see section 3.4.2 [RFC8609]).

Recommended cache time could be far in the future, but cannot be in the past and is likely to be a reasonably short offset from the current time. Therefore, there are a couple of alternatives for defining a compact representation. These are:

- Allow the compact representation to be interpreted as a delta time rather than an absolute time, since the semantics associated with an absolute time value do not seem to be critical to the utility of this value.

- Define an alternative absolute time base against which the RCT delta can be measured. One potential candidate would be a _message generation time_ included as a message TLV (not a hop-by-hop TLV). While doing this would not save any space if used for just RCT (it would actually make the message bigger) if its cost could be amortized across multiple other time TLVs in the same message, it might be a win.

This possibility is for further analysis and discussion as the document progresses.
4. A Compact Time Representation with Logarithmic Range

This document defines a compact time representation with logarithmic range support that is inspired by [RFC5497]. Compact time-codes are one or two octets wide and represent time-values that range from milliseconds to days. Figure 1 depicts the logarithmic nature of this time representation.

```
+--------------------------------------------------------------+
<table>
<thead>
<tr>
<th>milliseconds</th>
<th>days</th>
</tr>
</thead>
<tbody>
<tr>
<td>exponent (b)</td>
<td>mantissa (a)</td>
</tr>
</tbody>
</table>
| +-------------------+-------------------+
```

Figure 1: A logarithmic range representation allows for higher precision in the smaller time ranges and still supports large time deltas.

Time-codes encode an exponent and a mantissa as illustrated in Figure 2. Appropriate sizes for the exponent (b) and mantissa (a) are yet to be discussed.

```
<-- one or two octets wide -->
+++++++++++++++++++++++++++++
| exponent (b) | mantissa (a) |
+++++++++++++++++++++++++++++
```

Figure 2: A time-code with exponent and mantissa to encode a logarithmic range time representation.

5. Alternatives for a compact time representation in CCNx messages

A straightforward way to accommodate the TimeTLV approach as an alternative encoding is to simply allow a 1 or 2-byte length as an alternative to the 8-byte length while retaining the existing TLV Registry entries. While this has backward compatibility problems, the authors recommend this approach for the following reasons:

- Both CCNx RFCs are experimental and not Standards Track, hence expectations for forward and backward compatibility are not as stringent. "Flag day" upgrades of deployed CCNx networks, while inconvenient, are still feasible.

- The major use case for these compressed encodings are smaller-scale IoT and/or sensor networks where the population of consumers, producers, and forwarders is reasonably small.

- Since the current TLVs have hop-by-hop semantics, they are not covered by any signed hash and hence may be freely re-encoded by
any forwarder. That means a forwarder supporting the new encoding can translate freely between the two encodings.

- The alternative of assigning new TLV registry values does not substantially mitigate the interoperability problems anyway.

6. IANA Considerations

Please change the registry for the T_INTLIFE to permit a length of 1 or 2 in addition to a length of 8, as follows:

```
+---------------+---------------+---------------+---------------+
|             T_INTLIFE         |               1               |
| INTERVAL_TIME |               +---------------+
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
+-----------------------------+
```

```
+---------------+---------------+---------------+---------------+
|             T_INTLIFE         |               2               |
| INTERVAL_TIME |               +---------------+
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
+-----------------------------+
```

Figure 3: Alternate Delta Time encoding based on RFC4574

7. Security Considerations

This document makes no semantic changes to RFC8965, nor to any of the security properties of the message encodings of RFC8609, and hence has the same security considerations as those two existing documents.

8. References

8.1. Normative References


8.2. Informative References


Authors’ Addresses

Cenk Gundogan
HAW Hamburg
Berliner Tor 7
Hamburg D-20099
Germany
Phone: +4940428758067
Email: cenk.gundogan@haw-hamburg.de
URI: http://inet.haw-hamburg.de/members/cenk-gundogan

Thomas C. Schmidt
HAW Hamburg
Berliner Tor 7
Hamburg D-20099
Germany
Email: t.schmidt@haw-hamburg.de
URI: http://inet.haw-hamburg.de/members/schmidt
Dave Oran
Network Systems Research and Design
4 Shady Hill Square
Cambridge, MA  02138
USA

Email: daveoran@orandom.net

Matthias Waehlisch
link-lab & FU Berlin
Hoenower Str. 35
Berlin  D-10318
Germany

Email: mw@link-lab.net
URI:  http://www.inf.fu-berlin.de/~waehl