Internet Research Task Force (IRTF)
Request for Comments: 9510
Updates: 8609
Category: Experimental
ISSN: 2070-1721

C. Gndoan
Huawei
TC. Schmidt
HAW Hamburg
D. Oran
Network Systems Research and Design
M. Whlisch
TU Dresden

February 2024

Alternative Delta Time Encoding for Content-Centric Networking (CCNx) Using Compact Floating-Point Arithmetic

Abstract

Content-Centric Networking (CCNx) utilizes delta time for a number of functions. When using CCNx in environments with constrained nodes 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. This document updates RFC 8609 ("CCNx messages in TLV Format") to specify this alternative encoding.

This document is a product of the IRTF Information-Centric Networking Research Group (ICNRG).

Status of This Memo

This document is not an Internet Standards Track specification; it is published for examination, experimental implementation, and evaluation.

This document defines an Experimental Protocol for the Internet community. This document is a product of the Internet Research Task Force (IRTF). The IRTF publishes the results of Internet-related research and development activities. These results might not be suitable for deployment. This RFC represents the consensus of the Information-Centric Networking Research Group of the Internet Research Task Force (IRTF). Documents approved for publication by the IRSG are not candidates for any level of Internet Standard; see Section 2 of RFC 7841.

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at https://www.rfc-editor.org/info/rfc9510.

Table of Contents

1. Introduction
2. Terminology
3. Usage of Time Values in CCNx
  3.1. Relative Time in CCNx
  3.2. Absolute Time in CCNx
4. A Compact Time Representation with Logarithmic Range
5. Protocol Integration of the Compact Time Representation
  5.1. Interest Lifetime
  5.2. Recommended Cache Time
6. IANA Considerations
7. Security Considerations
8. References
  8.1. Normative References
  8.2. Informative References
Appendix A. Test Vectors
Appendix B. Efficient Time Value Approximation
Acknowledgments
Authors' Addresses

1. Introduction

CCNx is well suited for Internet of Things (IoT) applications [RFC7927]. Low-Power Wireless Personal Area Network (LoWPAN) adaptation layers (e.g., [RFC9139]) confirm the advantages of a space-efficient packet encoding for low-power and lossy networks. CCNx utilizes absolute and delta time values for a number of functions. When using CCNx in resource-constrained environments, it is valuable to have a compact representation of time values. However, any compact time representation has to sacrifice accuracy or dynamic range. For some time uses, this is relatively straightforward to achieve; for other uses, it is not. As a result of experiments in bandwidth-constrained IEEE 802.15.4 deployments [ICNLOWPAN], this document discusses the various cases of time values, proposes a compact encoding for delta times, and updates [RFC8609] to utilize this encoding format in CCNx messages.

This document has received fruitful reviews by the members of the research group (see the Acknowledgments section). It is the consensus of ICNRG that this document should be published in the IRTF Stream of the RFC series. This document does not constitute an IETF standard.

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 of [RFC8569] and [RFC8609] for CCNx entities.

The following terms are used in the document and defined as follows:

byte: synonym for octet

time value: a time offset measured in seconds

time code: an 8-bit encoded time value as defined in [RFC5497]

3. Usage of Time Values in CCNx

3.1. Relative Time in CCNx

CCNx, as currently specified in [RFC8569], utilizes delta time for

only the lifetime of an Interest message (see Sections 2.1, 2.2, 2.4.2, and 10.3 of [RFC8569]). It is a hop-by-hop header value, and is currently encoded via the T_INTLIFE TLV as a 64-bit integer (Section 3.4.1 of [RFC8609]). While formally an optional TLV, every Interest message is expected to carry the Interest Lifetime TLV in all but some corner cases; hence, having compact encoding is particularly valuable to keep 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 [IEEE.754.2019] and as outlined in Section 4. This document updates CCNx messages in TLV format [RFC8609] to permit this alternative encoding for selected time values.

3.2. Absolute Time 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 and Expiry Time

Signature Time is the time the signature of a Content Object was generated (see Sections 8.2-8.4 of [RFC8569]). Expiry Time indicates the time after which a Content Object is considered expired (Section 4 of [RFC8569]). Both values are content message TLVs and represent absolute timestamps in milliseconds since the POSIX epoch. They are currently encoded via the T_SIGTIME and T_EXPIRY TLVs as 64-bit unsigned integers (see Sections 3.6.4.1.4.5 and 3.6.2.2.2 of [RFC8609], respectively).

Both time values could be in the past or in the future, potentially by a large delta. They are also included in the security envelope of the message. Therefore, it seems there is no practical way to define an alternative compact encoding that preserves its semantics and security properties; therefore, this approach is not considered further.

3.2.2. Recommended Cache Time

Recommended Cache Time (RCT) for a Content Object (Section 4 of [RFC8569]) is a hop-by-hop header stating the expiration time for a cached Content Object in milliseconds since the POSIX epoch. It is currently encoded via the T_CACHETIME TLV as a 64-bit unsigned integer (see Section 3.4.2 of [RFC8609]).

A Recommended Cache Time could be far in the future, but it cannot be in the past and is likely to be a reasonably short offset from the current time. Therefore, this document allows the Recommended Cache Time to be interpreted as a relative time value rather than an absolute time, because the semantics associated with an absolute time value do not seem to be critical to the utility of this value. This document therefore updates the Recommended Cache Time with the following rule set:

* Use absolute time as per [RFC8609]

* Use relative time, if the compact time representation is used (see Sections 4 and 5)

If relative time is used, the time offset recorded in a message will typically not account for residence times on lower layers (e.g., for processing, queuing) and link delays for every hop. Thus, the Recommended Cache Time cannot be as accurate as when absolute time is

used. This document targets low-power networks, where delay bounds are rather loose or do not exist. An accumulated error due to transmission delays in the range of milliseconds and seconds for the Recommended Cache Time is still tolerable in these networks and does not impact the protocol performance.

Networks with tight latency bounds use dedicated hardware, optimized software routines, and traffic engineering to reduce latency variations. Time offsets can then be corrected on every hop to yield exact cache times.

4. A Compact Time Representation with Logarithmic Range

This document uses the compact time representation described in "Information-Centric Networking (ICN) Adaptation to Low-Power Wireless Personal Area Networks (LoWPANs)" (see Section 7 of [RFC9139]) that was inspired by [RFC5497] and [IEEE.754.2019]. Its logarithmic encoding supports a representation ranging from milliseconds to years. Figure 1 depicts the logarithmic nature of this time representation.

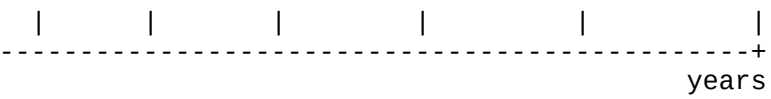


Figure 1: A logarithmic range representation allows for higher precision in the smaller time ranges and still supports large time deltas.

Time codes encode exponent and mantissa values in a single byte. In contrast to the representation in [IEEE.754.2019], time codes only encode non-negative numbers and hence do not include a separate bit to indicate integer signedness. Figure 2 shows the configuration of a time code: an exponent width of 5 bits, and a mantissa width of 3 bits.

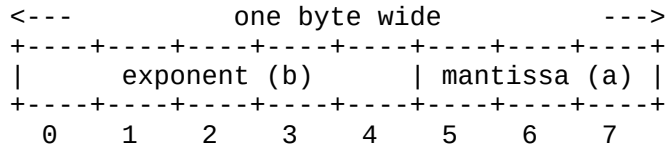


Figure 2: A time code with exponent and mantissa to encode a logarithmic range time representation.

The base unit for time values is seconds. A time value is calculated using the following formula (adopted from [RFC5497] and [RFC9139]), where (a) represents the mantissa, (b) the exponent, and (C) a constant factor set to C := 1/32.

Subnormal (b == 0): (0 + a/8) * 2 * C

Normalized (b > 0): (1 + a/8) * 2^b * C

The subnormal form provides a gradual underflow between zero and the smallest normalized number. Eight time values exist in the subnormal range [0s,~0.0546875s] with a step size of ~0.0078125s between each time value. This configuration also encodes the following convenient numbers in seconds: [1, 2, 4, 8, 16, 32, 64, ...]. Appendix A includes test vectors to illustrate the logarithmic range.

An example algorithm to encode a time value into the corresponding exponent and mantissa is given as pseudocode in Figure 3. Not all time values can be represented by a time code. For these instances, a time code is produced that represents a time value closest to and smaller than the initial time value input.

```
 input: float v    // time value
output:   int a, b // mantissa, exponent of time code

(a, b) encode (v):

    if (v == 0)
        return (0, 0)

    if (v < 2 * C)                              // subnormal
        a = floor (v * 4 / C)                   // round down
        return (a, 0)
    else                                        // normalized
        if (v > (1 + 7/8) * 2^31 * C)           // check bounds
            return (7, 31)                      // return maximum
        else
            b = floor (log2(v / C))             // round down
            a = floor ((v / (2^b * C) - 1) * 8) // round down
            return (a, b)
```

Figure 3: Algorithm in pseudocode.

For example, no specific time code for 0.063 exists. However, this algorithm maps to the closest valid time code that is smaller than 0.063, i.e., exponent 1 and mantissa 0 (the same as for time value 0.0625).

5. Protocol Integration of the Compact Time Representation

A straightforward way to accommodate the compact time approach is to use a 1-byte length field to indicate this alternative encoding while retaining the existing TLV registry entries. This approach has backward compatibility problems, but it is still considered for the following reasons:

* Both CCNx RFCs ([RFC8569] and [RFC8609]) 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.

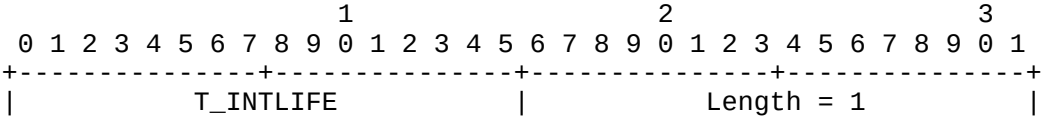
* The major use case for these compressed encodings are smaller-scale IoT and/or sensor networks where the population of consumers, producers, and forwarders is reasonably small.

* 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.

5.1. Interest Lifetime

The Interest Lifetime definition in [RFC8609] allows for a variable-length lifetime representation, where a length of 1 encodes the linear range [0,255] in milliseconds. This document changes the definition to always encode 1-byte Interest Lifetime values in the compact time value representation (see Figure 4). For any other length, Interest Lifetimes are encoded as described in Section 3.4.1 of [RFC8609].



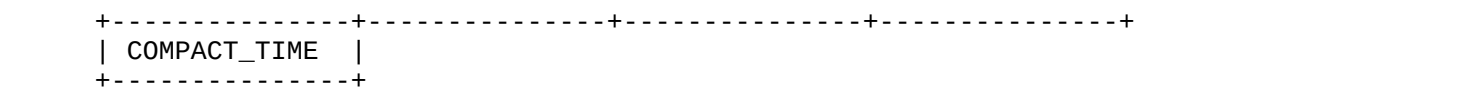


Figure 4: Changes to the definition of the Interest Lifetime TLV.

5.2. Recommended Cache Time

The Recommended Cache Time definition in [RFC8609] specifies an absolute time representation that is of a length fixed to 8 bytes. This document changes the definition to always encode 1-byte Recommended Cache Time values in the compact relative time value representation (see Figure 5). For any other length, Recommended Cache Times are encoded as described in Section 3.4.2 of [RFC8609].

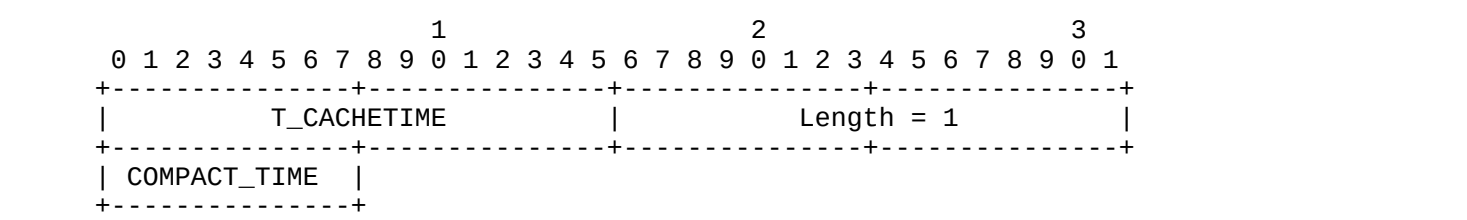


Figure 5: Changes to the definition of the Recommended Cache Time TLV.

The packet processing is adapted to calculate an absolute time from the relative time code based on the absolute reception time. On transmission, a new relative time code is calculated based on the current system time.

6. IANA Considerations

This document has no IANA actions.

7. Security Considerations

This document makes no semantic changes to [RFC8569], nor to any of the security properties of the message encodings described in [RFC8609]; hence, it has the same security considerations as those two documents. Careful consideration is needed for networks that deploy forwarders with support (updated forwarder) and without support (legacy forwarder) for this compact time representation:

Interest Lifetime: A legacy forwarder interprets a time code as an Interest Lifetime between 0 and 255 milliseconds. This may lead to a degradation of network performance, since Pending Interest Table (PIT) entries timeout quicker than expected. An updated forwarder interprets short lifetimes set by a legacy forwarder as time codes, thus configuring timeouts of up to 4 years. This leads to an inefficient occupation of state space.

Recommended Cache Time: A legacy forwarder considers a Recommended Cache Time with length 1 as a structural or syntactical error and it SHOULD discard the packet (Section 10.3.9 of [RFC8569]). Otherwise, a legacy forwarder interprets time codes as absolute time values far in the past, which impacts cache utilization.

8. References

8.1. Normative References

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

[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>.

[RFC8569] Mosko, M., Solis, I., and C. Wood, "Content-Centric Networking (CCNx) Semantics", RFC 8569, DOI 10.17487/RFC8569, July 2019, <https://www.rfc-editor.org/info/rfc8569>.

[RFC8609] Mosko, M., Solis, I., and C. Wood, "Content-Centric Networking (CCNx) Messages in TLV Format", RFC 8609, DOI 10.17487/RFC8609, July 2019, <https://www.rfc-editor.org/info/rfc8609>.

8.2. Informative References

[ICNLOWPAN] Gndoan, C., Kietzmann, P., Schmidt, T., and M. Whlisch, "Designing a LoWPAN convergence layer for the Information Centric Internet of Things", Computer Communications, Vol. 164, No. 1, p. 114-123, Elsevier, December 2020, <https://doi.org/10.1016/j.comcom.2020.10.002>.

[IEEE.754.2019] IEEE, "Standard for Floating-Point Arithmetic", IEEE Std 754-2019, DOI 10.1109/IEEESTD.2019.8766229, June 2019, <https://standards.ieee.org/content/ieee-standards/en/standard/754-2019.html>.

[RFC5497] Clausen, T. and C. Dearlove, "Representing Multi-Value Time in Mobile Ad Hoc Networks (MANETs)", RFC 5497, DOI 10.17487/RFC5497, March 2009, <https://www.rfc-editor.org/info/rfc5497>.

[RFC7927] Kutscher, D., Ed., Eum, S., Pentikousis, K., Psaras, I., Corujo, D., Saucez, D., Schmidt, T., and M. Waehlisch, "Information-Centric Networking (ICN) Research Challenges", RFC 7927, DOI 10.17487/RFC7927, July 2016, <https://www.rfc-editor.org/info/rfc7927>.

[RFC9139] Gndoan, C., Schmidt, T., Whlisch, M., Scherb, C., Marxer, C., and C. Tschudin, "Information-Centric Networking (ICN) Adaptation to Low-Power Wireless Personal Area Networks (LoWPANs)", RFC 9139, DOI 10.17487/RFC9139, November 2021, <https://www.rfc-editor.org/info/rfc9139>.

Appendix A. Test Vectors

The test vectors in Table 1 show sample time codes and their corresponding time values according to the algorithm outlined in Section 4.

Time Code	Time Value (seconds)
0x00	0.0000000
0x01	0.0078125
0x04	0.0312500
0x08	0.0625000
0x15	0.2031250

0x28	1.0000000
0x30	2.0000000
0xF8	67108864.0000000
0xFF	125829120.0000000

Table 1: Test Vectors

Appendix B. Efficient Time Value Approximation

A forwarder frequently converts compact time into milliseconds to compare Interest Lifetimes and the duration of cache entries. On many architectures, multiplication and division perform slower than addition, subtraction, and bit shifts. The following equations approximate the formulas in Section 4, and scale from seconds into the milliseconds range by applying a factor of 2^10 instead of 10^3. This results in an error of 2.4%.

```
b == 0:   2^10 * a * 2^-3 * 2^1 * 2^-5
          = a << 3

b > 0:    (2^10 + a * 2^-3 * 2^10) * 2^b * 2^-5
          = (1 << 5 + a << 2) << b
```

Acknowledgments

We would like to thank the active members of ICNRG for their constructive thoughts. In particular, we thank Marc Mosko and Ken Calvert for their valuable feedback on the encoding scheme and protocol integration. Special thanks also go to Carsten Bormann for his constructive in-depth comments during the IRSG review.

Authors' Addresses

Cenk Gndoan
Huawei Technologies Duesseldorf GmbH
Riesstrasse 25
D-80992 Munich
Germany
Email: cenk.gundogan@huawei.com

Thomas C. Schmidt
HAW Hamburg
Berliner Tor 7
D-20099 Hamburg
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
United States of America
Email: daveoran@orandom.net

Matthias Whlisch
TUD Dresden University of Technology
Helmholtzstr. 10
D-01069 Dresden

Germany
Email: m.waehlisch@tu-dresden.de