ICNRG
Internet-Draft
Updates: 8569, 8609 (if approved)
Intended status: Experimental
Expires: 4 September 2025

H. Asaeda
H.H. Hlaing
NICT
M.E. Mosko

3 March 2025

CCNx Content Versioning
draft-asaeda-icnrg-ccnxcversioning-00

Abstract

This document defines a method for content versioning in CCNx, enabling the differentiation of content sharing the same name using version numbers. This document updates RFC8569 [RFC8569] and RFC8609 [RFC8609].

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 https://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 4 September 2025.

Table of Contents

1. Introduction

Content is constantly updated, yet content names often remain unchanged. This document defines a method for content versioning in CCNx by introducing a new CCNx Name Segment type with TLV encoding to specify content version numbers. Each version number is an unsigned integer that increments by 1 for successive versions. Given a name with a serial number, the next version number can be computed accordingly. In addition, this document describes a method for discovering the current (latest) version number of content. It introduces a new TLV type CurrentVersion that is used in a ContentObject Payload to wrap a CCNx LINK TLV.

Notably, content versioning is not mandatory in CCNx, nor are consumers required to specify a version number to retrieve unversioned content.

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 BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

2.1. Definitions

This document follows the definitions provided in [RFC8793] for general ICN-related terms and in [RFC8569] and [RFC8609] for CCNx-specific terms. Moreover, the following terms are defined:

Versioned content Content assigned a version number in its name, allowing differentiation by version.

Unversioned content Content that is not assigned a version number and remains undistinguished by version.

Version Query An Interest message sent from a consumer to a producer to retrieve the latest version number of specific content. The packet type for a Version Query is PT_INTEREST. It does not request the content object itself.

Version Response A Data message sent from a producer to a consumer containing a LINK to latest version of the content. The packet type of a Version Response is PT_CONTENT. It does not include the content object itself.

3. Protocol Description

This document introduces one new CCNx Name Segment TLV: the Version TLV. As the Version TLV is optional, consumers can send Interest packets without including it (see below). Nevertheless, if consumers wish to retrieve a specific version of content (e.g., the latest version), they MUST specify the version number in the Version TLV and send an Interest packet with this information.

Version numbers establish a sequential order of names and are represented as unsigned integers. They are encoded using network byte order with the minimum necessary bytes. The value of "0" is represented as a single byte (%x00). Version numbers are incremented for consecutive versions of preceding name segments. There is no predefined upper limit for version numbers; the maximum is determined solely by the available number of bytes allocated in the name component.

To retrieve versioned content, consumers specify the content’s version number in the Version TLV of a CCNx Name Segment within an Interest packet. Producers or forwarders (e.g., caching routers) respond with the corresponding Data packet using the standard CCNx Content Object to Interest matching rules Section 9 of [RFC8569].

If a consumer sends an Interest including a Version TLV for unversioned content, the content name is mismatched; therefore, the Interest packet is discarded. Similarly, if a consumer sends an Interest without a Version TLV for versioned content, the content name is mismatched, and the Interest packet is discarded. If a consumer requests versioned content by specifying an unknown version number in Version TLV, the Interest packet is discarded.

If consumers want to ensure that they get the latest version's content, they need to specify the latest version number of the content. To query the latest version number of versioned content, this document defines a special Interest-Data exchange called Version Query-Response (described in Section 5 and Section 6). When consumers send interests for versioned content with Version TLV but no version number specified (i.e., the value specified in the Length field of the Version TLV is 0), producer returns a content object with a Payload of a CurrentVersion TLV that wraps a CCNx LINK Section 6 of [RFC8569] to the latest versioned name.

This document assumes that caching routers manage and process version numbers for versioned content in addition to handling content names. The specifications outlined in this document will be implemented in Cefore [Asaeda2019][Cefore].

4. TLV Types

This section specifies the optional TLV types used to communicate content versions.

4.1. Version

CCNx versioning introduces a new CCNx Name Segment type: Version.

Type	Abbrev	Name	Description
%x0004	T_VERSION	Version Number	The Version Number, as an unsigned integer in network byte order without leading zeros. The value of zero is represented as the single byte %x00.

Table 1: Version

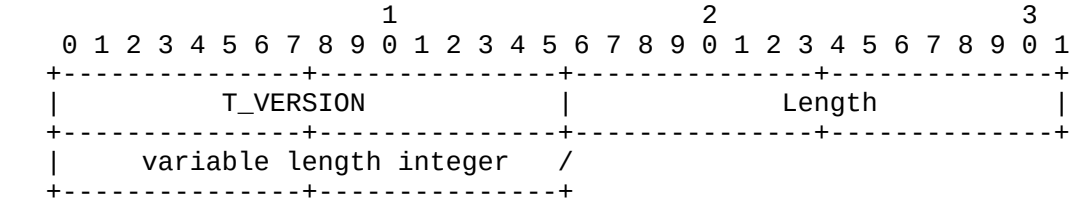

Figure 1: Version TLV

Code	Type name
%x0000	Reserved [RFC8609]
%x0001	T_NAMESEGMENT [RFC8609]
%x0002	T_IPID [RFC8609]
%x0003	T_NONCE [RFC9508]
%x0004	T_VERSION
%x0005	T_CHUNK [I-D.mosko-icnrg-ccnxchunking]
%x0006	T_REFLEXIVE_NAME [I-D.irtf-icnrg-reflexive-forwarding]
%x0007-%x000F	Unassigned
%x0010-%x0013	Reserved [RFC8609]
%x0014-0x0FFE	Unassigned
%x0FFF	T_ORG [RFC8609]
%x1000-0x1FFF	T_APP:00 - T_APP:4096 [RFC8609]
%x2000-0xFFFF	Unassigned

Figure 2: CCNx Name Segment Type Namespace

An Interest packet may include a chunk number [I-D.mosko-icnrg-ccnxchunking], as well as content name and version number. If the version number is included in the CCNx Name TLV, the Version TLV specifying the version number MUST immediately follow the last T_NAMESEGMENT specifying the content name. If both the version number and chunk number are included to identify the content, the Version TLV MUST be followed by the ChunkNumber TLV.

4.2. CurrentVersion

CCNx versioning introduces a new CCNx registry "CCNx Versioning." This registry identifies TLVs that are part of the CCNx Version Request-Response exchange.

Type	Abbrev	Name	Description
%x0007	T_CURVERSION	Current Version Name	The Current Version Name wraps a CCNx LINK to the most recent versioned content.

Table 2: CurrentVersion

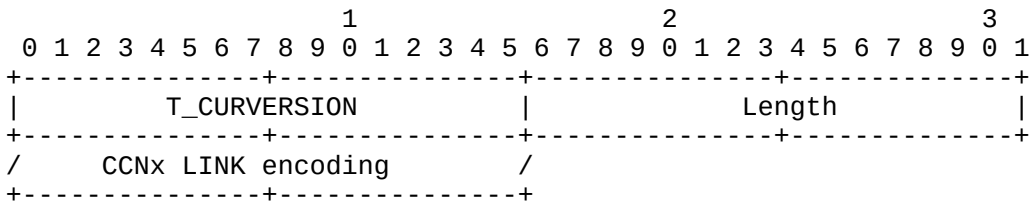

Figure 3: CurrentVersion TLV

Code	Type name
%x0000	T_NAME [RFC8609]
%x0001	T_PAYLOAD [RFC8609]
%x0002	T_KEYIDRESTR [RFC8609]
%x0003	T_OBJHASHRESTR [RFC8609]
%x0004	Unassigned
%x0005	T_PAYLDTYPE [RFC8609]
%x0006	T_EXPIRY [RFC8609]
%x0007	T_CURVERSION
%x0008	T_ENDCHUNK [I-D.mosko-icnrg-ccnxchunking]
%x0009-%x000C	Reserved [RFC8609]
%x000D	T_DISC_REQ [RFC9344]
%x000E	T_DISC_REPLY [RFC9344]
%x0FFE	T_PAD [RFC8609]
%x0FFF	T_ORG [RFC8609]
%x1000-%x1FFF	Reserved [RFC8609]

Figure 4: CCNx Message Type Namespace

4.3. Examples

Below are examples of version names using the labeled content identity URI scheme in human-readable form (ccnx:). In the ccnx: URI form, it is denoted as "Ver". In the following example, the content producer publishes a JPG with version number is 1.

ccnx:/Name=ietf.org/Name=abc.jpg/Ver=1

The following examples indicate content identified using its name, version number, and chunk number.

```
ccnx:/Name=ietf.org/Name=abc.jpg/Ver=1/Chunk=0  --
ccnx:/Name=ietf.org/Name=abc.jpg/Ver=1/Chunk=1  EndChunkNumber=3
ccnx:/Name=ietf.org/Name=abc.jpg/Ver=1/Chunk=2  --
ccnx:/Name=ietf.org/Name=abc.jpg/Ver=1/Chunk=3  EndChunkNumber=3
```

[HA] What if the version number is changed during chunk transmission? Since it may be intentionally requested, we cannot prohibit this as an error. Furthermore, we cannot recognize it is an error. Do we need to say something about this situation?

```
ccnx:/Name=ietf.org/Name=abc.jpg/Ver=1/Chunk=0  --
ccnx:/Name=ietf.org/Name=abc.jpg/Ver=0/Chunk=1  EndChunkNumber=2
ccnx:/Name=ietf.org/Name=abc.jpg/Ver=2/Chunk=2  EndChunkNumber=2
```

5. Version Query

When consumers need to determine the latest version number of content, they can send a special Interest packet called a *Version Query*. Version Query includes CCNx Name Segment TLV(s) specifying the content name and a Version TLV with no version number (i.e., the Length field in the Version TLV is set to 0) (see Figure 5). After producers or forwarders aware of the latest version number receive the Version Query, they transmit a *Version Response* as described in Section 6. After obtaining the latest version number, consumers can send an Interest packet specifying both the content name and version number to retrieve the desired content object. Version Queries are ignored and silently discarded for unversioned content.

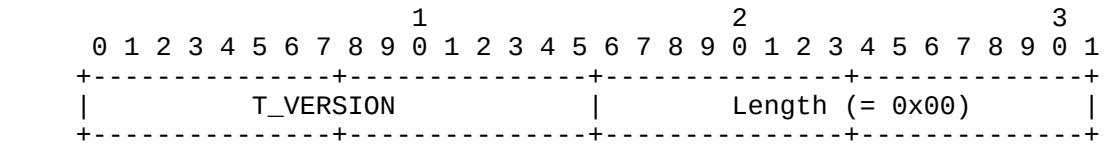

Figure 5: Version Query containing Version TLV with no version number

A Version Query is a standard CCNx Interest. A Version Query normally travels to the authoritative publisher via CCNx routing. The publisher responds to the Version Query with a LINK to the current vesion. The respone may be cached (see Section 6).

It is possible that a CCNx name includes multiple Version fields, for example if an annotation is published as an extension of a previous name. For example ccnx:/Name=foo/Name=x/Ver=1/Name=biblio/Ver=2/Chunk=3. Here, we have a first object ccnx:/Name=foo/Name=x/Ver=1 and a second bibliographic object. In this example, the object "x" is of version 1 and the bibliography information is of version 2.

A Version Query may contain one or more empty version numbers and the responding publisher will fill in the authoritative information for the terminal version and the corresponding fields for earlier version numbers. The Version Query will be routed to the longest-matching publisher, who corresponds to the terminal version number. That publisher only has the current version number of it’s name. Any prior fields only indicate the version on to which this extension name rides.

6. Version Response

Producers send a Version Response as reply to a Version Query. The Version Response contains a Data packet with a CurrentVersion TLV wrapping a CCNx LINK in the Payload. The Link identifies the current version name, and SHOULD include at least one of the ContentObjectHashRestriction or KeyIdRestriction.

If a Version Query is sent for unversioned content, the Version Response includes an empty Current Version TLV (see Figure 6).

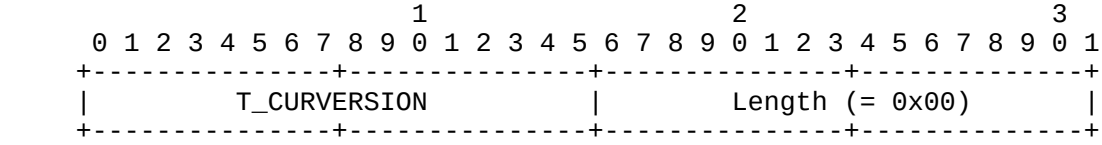

Figure 6: Version Response containing CurrentVersion TLV with no version number

The Version Response MAY include a Recommended Cache Time (RCT) and MUST include an appropriate ExpiryTime. If the response includes an RCT, then intermediate forwarders may cache the content and respond with it within the ExpiryTime. The ExpiryTime should be based on the publisher’s concept of content update interval.

A Version Response SHOULD be singed by a key trusted for the Verion’s namespace.

7. IANA Considerations

As per [RFC8126], this section makes an assignment in one existing registry in the "Content-Centric Networking (CCNx)" registry group. The registration procedure is "RFC Required", which requires only that this document be published as an RFC.

7.1. CCNx Name Segment Type Registry

As shown in Figure 2, this document defines one CCNx Name Segment type, T_VERSION, whose suggested value is %x0004.

7.2. CCNx Message Type Registry

As shown in Figure 4, this document defines one CCNx Message type, T_CURVERSION, whose suggested value is %x0007.

8. Security Considerations

Malicious nodes could generate excessive or invalid Version Queries to overload forwarders and/or producers on the network, provoking unnecessary Version Responses. To mitigate this risk, forwarders or producers MAY randomly ignore Version Queries. This approach reduces processing overhead, ensures fair handling of Version Queries, and minimizes traffic amplification risks. The rate-limiting strategy is left to the implementation of forwarders and producers.

Version query flooding is a reason that a publisher may want to use a Recommended Cache Time on a Version Response. This allows intermediate forwarders to offload Version Query responses, so long as the RCT can match the expected update interval of the content.

Signing a Version Response is a potential computation DDoS on the publisher. This is why a publisher should include a reasonable ExpiryTime on its response so it can respond from cache when possible.

9. Acknowledgements

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>.

[RFC8126] Cotton, M., Leiba, B., and T. Narten, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 8126, DOI 10.17487/RFC8126, June 2017, <https://www.rfc-editor.org/info/rfc8126>.

[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>.

10.2. Informative References

[Asaeda2019] Asaeda, H., Ooka, A., Matsuzono, K., and R. Li, "Cefore: Software Platform Enabling Content-Centric Networking and Beyond", IEICE Transactions on Communications, Vol.E102-B, No.9, DOI 10.1587/transcom.2018EII0001, September 2019, <https://search.ieice.org/bin/summary.php?id=e102-b_9_1792>.

[Cefore] "Cefore", <https://github.com/cefore/>.

[I-D.irtf-icnrg-reflexive-forwarding] Oran, D. R., KUTSCHER, D., and H. Asaeda, "Reflexive Forwarding for CCNx and NDN Protocols", Work in Progress, Internet-Draft, draft-irtf-icnrg-reflexive-forwarding-00, 18 October 2024, <https://datatracker.ietf.org/doc/html/draft-irtf-icnrg-reflexive-forwarding-00>.

[I-D.mosko-icnrg-ccnxchunking] Mosko, M. and H. Asaeda, "CCNx Content Object Chunking", Work in Progress, Internet-Draft, draft-mosko-icnrg-ccnxchunking-03, 20 October 2024, <https://datatracker.ietf.org/doc/html/draft-mosko-icnrg-ccnxchunking-03>.

[RFC8793] Wissingh, B., Wood, C., Afanasyev, A., Zhang, L., Oran, D., and C. Tschudin, "Information-Centric Networking (ICN): Content-Centric Networking (CCNx) and Named Data

Networking (NDN) Terminology", RFC 8793,
DOI 10.17487/RFC8793, June 2020,
<https://www.rfc-editor.org/info/rfc8793>.

Authors' Addresses

Hitoshi Asaeda
National Institute of Information and Communications Technology
Koganei, Tokyo
184-8795
Japan
Email: asaeda@nict.go.jp

Htet Htet Hlaing
National Institute of Information and Communications Technology
Koganei, Tokyo
184-8795
Japan
Email: htethtethlaing@nict.go.jp

Marc Mosko
Kensington, California 94707
United States of America
Email: marc@mosko.org