

ICNRG  
Internet-Draft  
Updates: 8569, 8609 (if approved)  
Intended status: Experimental  
Expires: 1 September 2026

H. Asaeda  
H.H. Hlaing  
NICT  
M.E. Mosko  
28 February 2026

CCNx Content Versioning  
draft-irtf-icnrg-ccnxversioning-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 1 September 2026.

## Copyright Notice

Copyright (c) 2026 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document.

## Table of Contents

1. Introduction . . . . .	2
2. Terminology . . . . .	3
2.1. Definitions . . . . .	3
3. Protocol Description . . . . .	3
4. TLV Types . . . . .	4
4.1. Version . . . . .	4
4.2. CurrentVersion . . . . .	5
4.3. Examples . . . . .	6
5. Version Query . . . . .	6
6. Version Response . . . . .	7
7. IANA Considerations . . . . .	7
7.1. CCNx Name Segment Type Registry . . . . .	7
7.2. CCNx Message Type Registry . . . . .	8
8. Security Considerations . . . . .	8
9. Acknowledgements . . . . .	8
10. References . . . . .	8
10.1. Normative References . . . . .	8
10.2. Informative References . . . . .	8
Authors' Addresses . . . . .	9

## 1. Introduction

This document specifies a content versioning scheme for CCNx. Content versions are explicitly named by defining a new CCNx Name Segment type to carry the version information. Each version number is an unsigned integer that increments by 1 for successive versions. Given a name with a serial number in a version TLV in a CCNx Name Segment, the next (or any prior) version number can be ascertained accordingly.

If a consumer does not already know the full name (including the version number) of an existing version of the content, this document describes a method for discovering the current (latest) version number of content known to the producer. It introduces a new TLV type CurrentVersion that is used in a ContentObject Payload to wrap a CCNx LINK TLV containing the full name of the latest known version.

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.

Versioned content is retrieved by including the version as a CCNx Name Segment TLV in the Interest name. By convention, it makes sense to make it a trailing Name Segment (possibly followed by a chunk number if [I-D.irtf-icnrg-ccnxchunking] is employed). Producers or forwarders (e.g., caching routers) respond with the corresponding Data packet using the standard CCNx Content Object to Interest matching rules from 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 an 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.

The addition of content versioning does not affect the operation of CCNx forwarders, only consumers and producers that desire to support versioned content. The specifications outlined in this document are 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
TBA by IANA	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

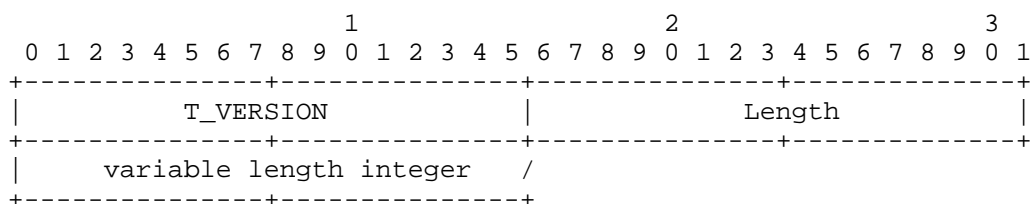


Figure 1: Version TLV

An Interest packet may include a chunk number [I-D.irtf-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 Query-Response exchange.

Type	Abbrev	Name	Description
TBA by IANA	T_CURVERSION	Current Version Name	The Current Version Name wraps a CCNx LINK to the most recent versioned content.

Table 2: CurrentVersion

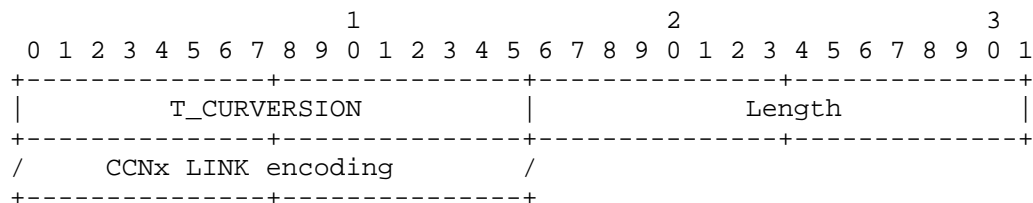


Figure 2: CurrentVersion TLV

#### 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
```

#### 5. Version Query

Consumers can ascertain the highest known version number of versioned content by issuing a Version Query. This is an Interest packet which 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 3). 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.

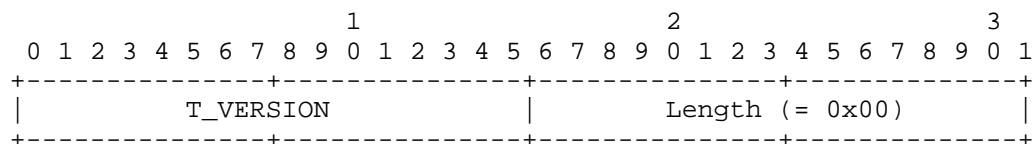


Figure 3: Version Query containing Version TLV with no version number

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

## 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 4).

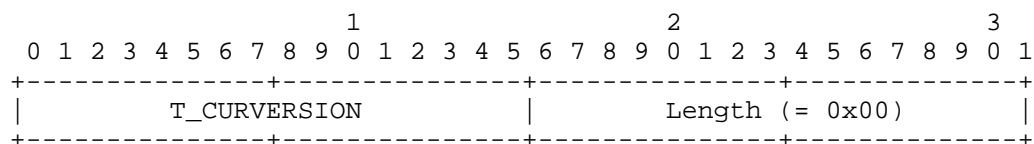


Figure 4: Version Response containing CurrentVersion TLV with no version number

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 signed 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

This document defines one CCNx Name Segment type, T\_VERSION, whose suggested value is %x0004.

## 7.2. CCNx Message Type Registry

This document defines one CCNx Message type, T\_CURVERSION, whose suggested value is %x0007.

## 8. Security Considerations

TBD

## 9. Acknowledgements

The authors would like to thank David Oran for his valuable comments and suggestions on this document.

## 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://doi.org/10.1587/transcom.2018EII0001>>.

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

[I-D.irtf-icnrg-ccnxchunking]

Mosko, M. and H. Asaeda, "CCNx Content Object Chunking", Work in Progress, Internet-Draft, draft-irtf-icnrg-ccnxchunking-04, 6 February 2026, <<https://datatracker.ietf.org/doc/html/draft-irtf-icnrg-ccnxchunking-04>>.

[I-D.irtf-icnrg-reflexive-forwarding]

Oran, D. R., KUTSCHER, D., Asaeda, H., and K. Calvert, "Reflexive Forwarding for CCNx and NDN Protocols", Work in Progress, Internet-Draft, draft-irtf-icnrg-reflexive-forwarding-02, 23 September 2025, <<https://datatracker.ietf.org/doc/html/draft-irtf-icnrg-reflexive-forwarding-02>>.

[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](mailto:asaeda@nict.go.jp)

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

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