Stream: Internet Research Task Force (IRTF)
RFC: 9531
Category: Experimental
Published: February 2024
ISSN: 2070-1721
Authors: I. Moiseenko, *Apple, Inc.*
D. Oran, *Network Systems Research and Design*

RFC 9531
Path Steering in Content-Centric Networking (CCNx) and Named Data Networking (NDN)

Abstract

Path steering is a mechanism to discover paths to the producers of Information-Centric Networking (ICN) Content Objects and steer subsequent Interest messages along a previously discovered path. It has various uses, including the operation of state-of-the-art multi-path congestion control algorithms and for network measurement and management. This specification derives directly from the design published in "Path Switching in Content Centric and Named Data Networks" (4th ACM Conference on Information-Centric Networking) and, therefore, does not recapitulate the design motivations, implementation details, or evaluation of the scheme. However, some technical details are different, and where there are differences, the design documented here is to be considered definitive.

This document is a product of the IRTF Information-Centric Networking Research Group (ICNRG). It is not an IETF product and is not an Internet Standard.

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/rfc9531.

Copyright Notice

Table of Contents

1. Introduction

Path steering is a mechanism to discover paths to the producers of ICN Content Objects and steer subsequent Interest messages along a previously discovered path. It has various uses, including the operation of state-of-the-art multi-path congestion control algorithms and for network measurement and management. This specification derives directly from the design published in

[Moiseenko2017] and, therefore, does not recapitulate the design motivations, implementation details, or evaluation of the scheme. That publication should be considered a normative reference as it is not likely a reader will be able to understand all elements of this design without first having read the reference. However, some technical details are different, and where there are differences, the design documented here is to be considered definitive.

Path discovery and subsequent path steering in ICN networks is facilitated by the symmetry of forward and reverse paths in the Content-Centric Networking (CCNx) and Named Data Networking (NDN) architectures. Path discovery is achieved by a consumer endpoint transmitting an ordinary Interest message and receiving a Content (Data) message containing an end-to-end path label constructed on the reverse path by the forwarding plane. Path steering is achieved by a consumer endpoint including a path label in the Interest message, which is forwarded to each nexthop through the corresponding egress interfaces in conjunction with Longest Name Prefix Match (LNPM) lookup in the Forwarding Information Base (FIB).

This document is a product of the IRTF Information-Centric Networking Research Group (ICNRG). It was supported by the ICNRG participants during its development and through Research Group Last Call. It has received detailed review by experts in both the CCNx and NDN communities.

1.1. Path Steering as an Experimental Extension to ICN Protocol Architectures

There are a number of important use cases to justify extending ICN architectures such as CCNx [RFC8569] or NDN [NDN] to provide these capabilities. These are summarized as follows:

- Support the discovery, monitoring, and troubleshooting of multi-path network connectivity, based on names and name prefixes. Analogous functions have been shown to be a crucial operational capability in multicast and multi-path topologies for IP. The canonical tools are the well-known *traceroute* and *ping*. For point-to-multipoint MPLS, the more recent MPLS traceroute [RFC8029] protocol is used. Equivalent diagnostic functions have been defined for CCNx through the ICN Ping [RFC9508] and ICN Traceroute [RFC9507] specifications; both of which are capable of exploiting path steering, if available.
- Perform accurate online measurement of network performance, which generally requires multiple consecutive packets to follow the same path under control of an application.
- Improve the performance and flexibility of multi-path congestion control algorithms. Congestion control schemes, such as [Mahdian2016] and [Song2018], depend on the ability of a consumer to explicitly steer packets onto individual paths in a multi-path and/or multi-destination topology.
- Allow a consumer endpoint to mitigate content poisoning attacks by directing its Interests onto the network paths that bypass poisoned caches.

The path discovery machinery described here may (and likely will) discover paths with varying properties. [RFC9217] discusses a number of open questions in path-aware networking, among which is how to assess and exploit paths having different properties. Experimenting with ICN path steering may be helpful in further elucidating these questions and perhaps shedding light on which path properties are most useful for the use cases cited above.

One nuance compared to other path-aware networking approaches is that ICN path steering piggybacks path discovery on the base ICN data exchange rather than having a separate path advertisement or discovery mechanism. That means when the recorded path comes back in an ICN Data message response, the properties of the path are known only implicitly to the consumer as opposed to being explicitly labeled. That makes the question of what properties a consumer uses to choose a path one of observation or measurement rather than advance selection based on an explicit, advertised property (e.g., SCION [SCION]).

The utility and overall technical quality of this path steering capability can be assessed by how well it enables the above use cases and what performance and robustness effects it has on the underlying ICN protocols and their use in various applications. A few of the open questions that should be addressed through experimentation with path steering include:

- How much more accurate and useful are measurements of RTT, packet loss, etc. through ping and traceroute when utilizing path steering?
- How much is the performance and robustness of multi-path forwarding enhanced by the use of this explicit path steering capability?

1.2. Requirements Language

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.

1.3. Terminology

This document uses the general ICN terms that are defined in [RFC8793]. In addition, we define the following terms specific to path steering:

Path Discovery: The process of sending an Interest message requesting discovery of a path and, if successful, receiving a Data message containing a path label for the path the corresponding Interest traversed.

Path Steering: The process of sending an Interest message containing the path label of a previously discovered path so that the forwarders use that path when forwarding that particular Interest message.

Path Label: An optional field in the packet indicating a particular path from a consumer to either a producer or a forwarder cache that can respond with the requested item. In an Interest message, the path label gets built up hop by hop as the Interest traverses a path. In a Data message, the path label carries the full path information back to the consumer for use in one or more subsequent Interest messages.

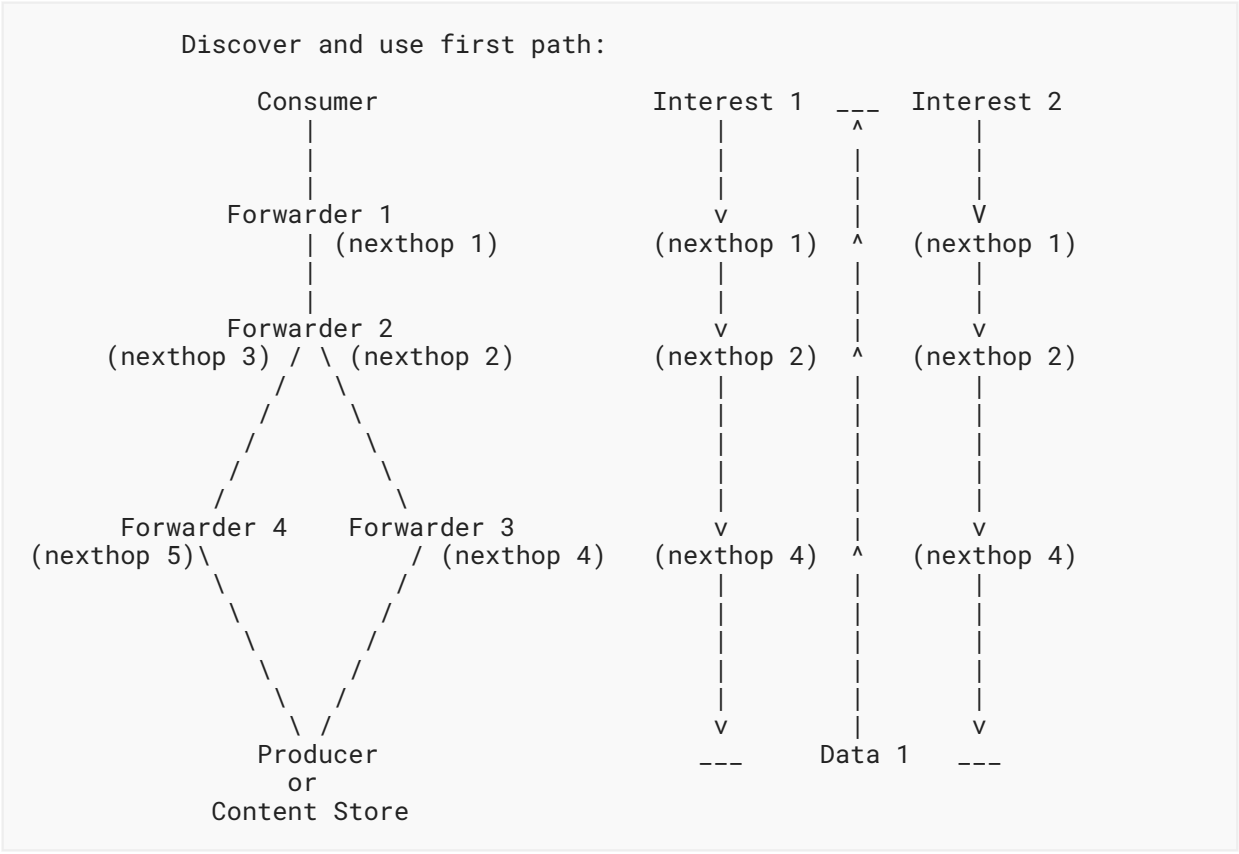
Nexthop Label: One entry in a path label representing the next hop for the corresponding forwarder to use when a path-steered Interest message arrives at that forwarder. A sequence of Nexthop Labels constitutes a full path label.

2. Essential Elements of ICN Path Discovery and Path Steering

We elucidate the design using CCNx semantics [RFC8569] and extend its CCNx Message Formats [RFC8609] defined in Section 3.2. While the terminology is slightly different, this design can also be applied to NDN by extending its bespoke packet encodings [NDNTLV] (see Section 3.3).

2.1. Path Discovery

End-to-end Path Discovery for CCNx is achieved by creating a *path label* and placing it as a hop-by-hop TLV in a CCNx Content (Data) message. The path label is constructed hop by hop as the message traverses the reverse path of transit CCNx forwarders, as shown in the first example in Figure 1. The path label is updated by adding the Nexthop Label of the interface at which the Content (Data) message has arrived to the existing path label. Eventually, when the Content (Data) message arrives at the consumer, the path label identifies the complete path the Content (Data) message took to reach the consumer. As shown in the second example in Figure 1, when multiple paths are available, subsequent Interests may be able to discover additional paths by omitting a path steering TLV and obtaining a new path label on the returning Interest.


Discover and use first path:
Consumer
Forwarder 1
(nexthop 1)
Forwarder 2
(nexthop 3)
(nexthop 2)
Forwarder 4
Forwarder 3
(nexthop 5)
(nexthop 4)
Producer
or
Content Store
Interest 1
Interest 2
(nexthop 1)
(nexthop 1)
(nexthop 2)
(nexthop 2)
(nexthop 4)
(nexthop 4)
Data 1

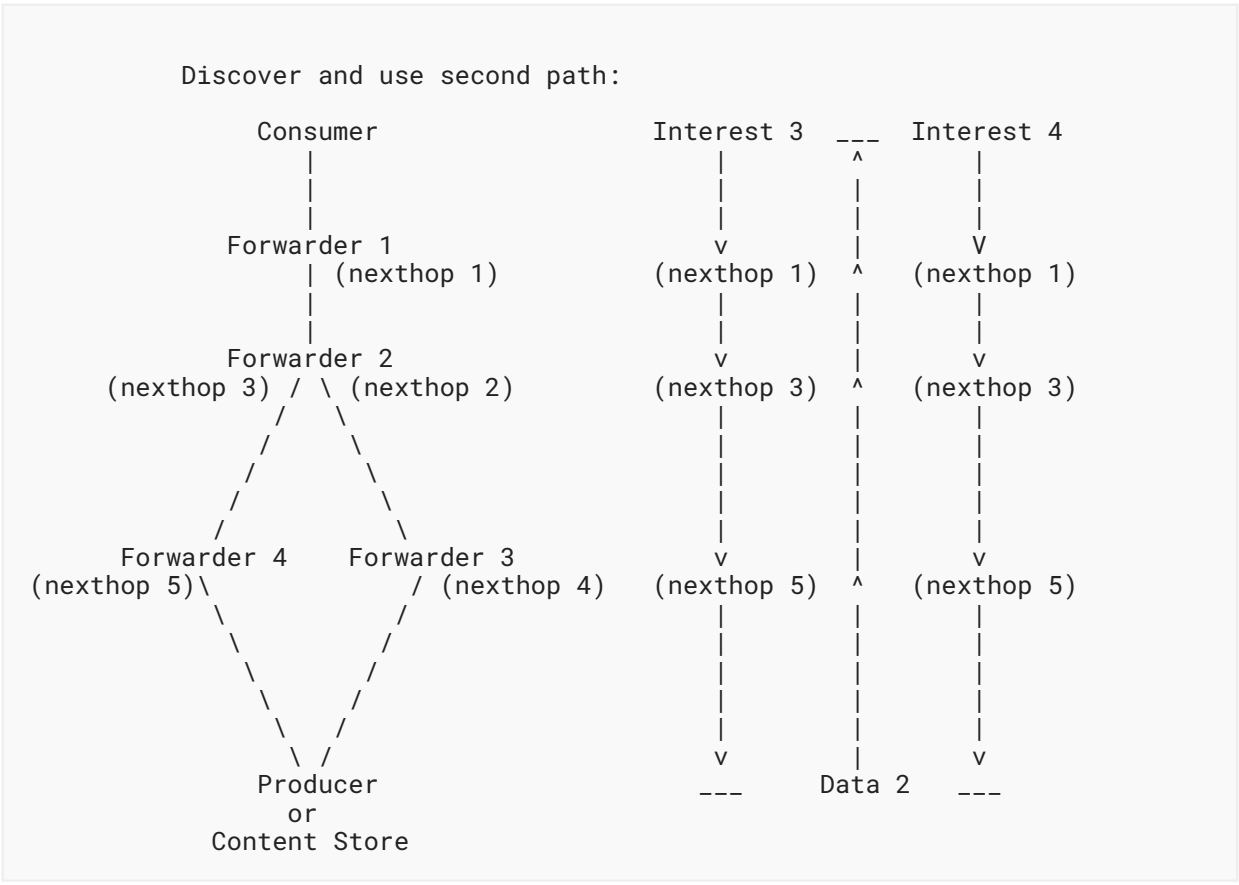


Figure 1: Basic Example of Path Discovery and Steering

2.2. Path Steering

Due to the symmetry of forward and reverse paths in CCNx, a consumer application can reuse a discovered path label to fetch the same or a similar (e.g., next chunk, next Application Data Unit, or next pointer in a Manifest [FLIC]) Content (Data) message over the discovered network path. This *path steering* is achieved by processing the Interest message's path label at each transit ICN forwarder and forwarding the Interest through the specified nexthop among those identified as feasible by LNPM FIB lookup (Figure 2).

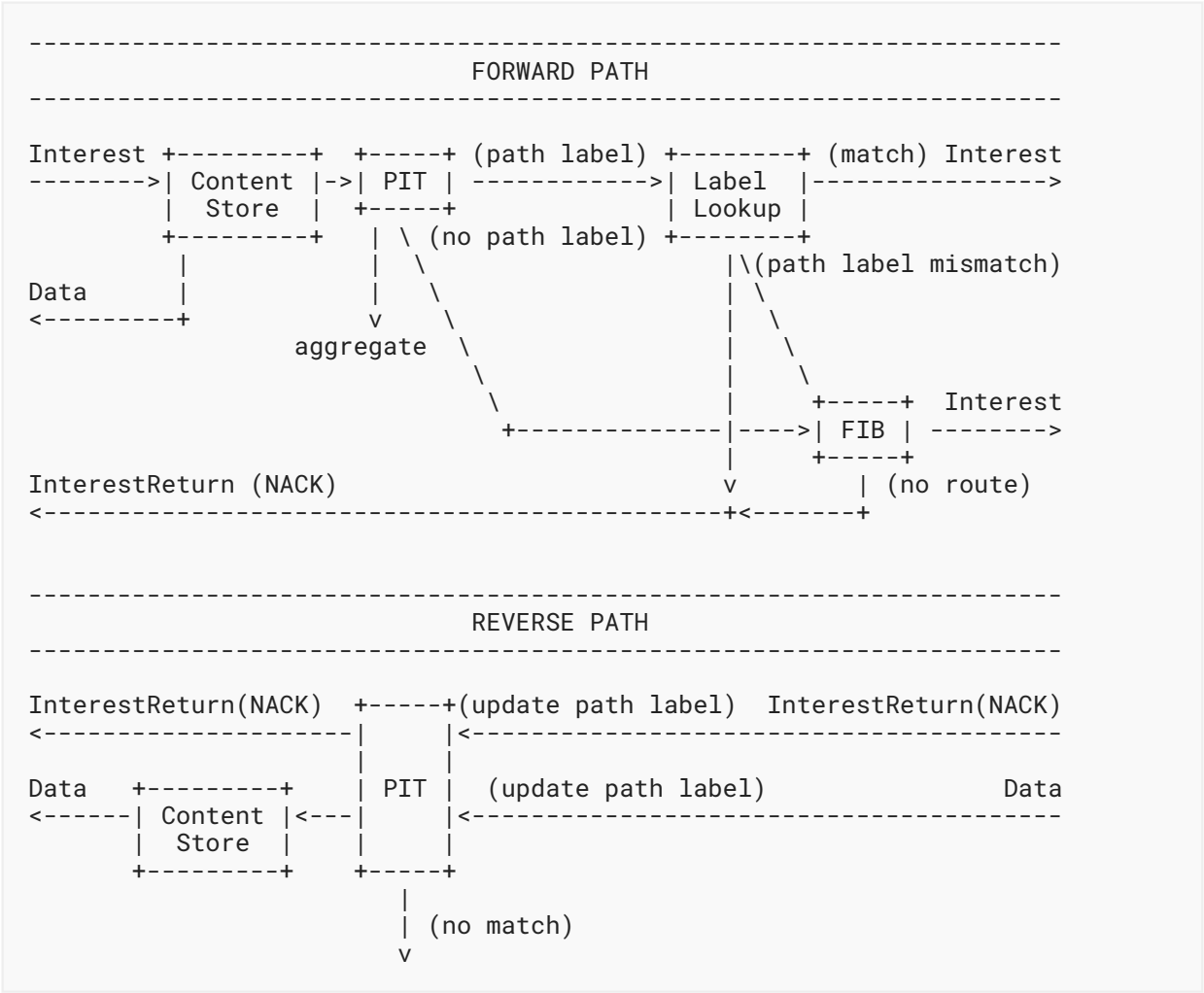


Figure 2: Path Steering CCNx/NDN Data Plane

2.3. Handling Path Steering Errors

Over time, the state of interfaces and the FIB on forwarders may change such that, at any particular forwarder, a given nexthop is no longer valid for a given prefix. In this case, the path label will point to a now-invalid nexthop. This is detected by failure to find a match between the decoded nexthop ID and the nexthops of the FIB entry after LNPM FIB lookup.

On detecting an invalid path label, the forwarder **SHOULD** respond to the Interest with an InterestReturn. Therefore, we define a new *invalid path label* response code for the InterestReturn message and include the current path label as a hop-by-hop header. Each transit forwarder processing the InterestReturn message updates the path label in the same manner as Content (Data) messages so that the consumer receiving the InterestReturn (NACK) can easily identify which path label is no longer valid.

A consumer may alternatively request that a forwarder detecting the inconsistency forward the Interest by means of normal LNPM FIB lookup rather than return an error. The consumer endpoint, if it cares, can keep enough information about outstanding Interests to determine if the path label sent with the Interest fails to match the path label in the corresponding returned Content (Data) and use that information to replace stale path labels. It does so by setting the FALLBACK_MODE flag of the path label TLV in its Interest message.

2.4. Interactions with Interest Aggregation

If two or more Interests matching the same Pending Interest Table (PIT) entry arrive at a forwarder, under current behavior, they will be aggregated whether or not they carry identical path label TLVs. This may or may not be appropriate. For example, multiple Interests with different modes (e.g., one with DISCOVERY_MODE and one without) will get aggregated; therefore, the behavior of the forwarder might be dependent on the arrival order of those Interests. In particular:

- If the DISCOVERY_MODE Interest arrives first, it will be forwarded and potentially discover a new path, while the other Interest will be aggregated. If that Interest carried no path label, its behavior is essentially unchanged, but if it carried a path label without specifying DISCOVERY_MODE, the consumer's intent for the Interest to traverse the specified path will be ignored, and it is indeterminate if the chosen path will actually be used.
- If the two Interests arrive in the reverse order, the DISCOVERY MODE Interest will be aggregated, and the consumer issuing it will not achieve its desire to discover a new path.

Multiple Interests intended to discover paths (i.e., by carrying the DISCOVERY_MODE flag defined in Section 3.1) might also be aggregated by a forwarder. This limits the ability to discover multiple paths in parallel and, instead, must be discovered incrementally in subsequent exchanges. In other words, aggregated Interests will all discover only one single path carried by one single Data packet. This has implications for management applications, like traceroute [RFC9507], which would likely perform much better if they discover paths in parallel. Hence, when employing path steering, it is **RECOMMENDED** that such applications craft their Interests with unique name suffixes in order to avoid being aggregated.

> While path steering still operates correctly if DISCOVERY MODE Interests are aggregated, after further experimentation, it may be appropriate to advise that a forwarder:
>
> - **SHOULD NOT** aggregate Interests carrying different path labels and
> - **SHOULD** apply a rate limit to DISCOVERY_MODE Interests in order to limit redundant traffic.

2.5. How to Represent the Path Label

[Moiseenko2017] presents various options for how to represent a path label, with different trade-offs in flexibility, performance, and space efficiency. For this specification, we choose the *polynomial encoding*, which achieves reasonable space efficiency at the cost of establishing a hard limit on the length of paths that can be represented.

The polynomial encoding utilizes a fixed-size bit array. Each transit ICN forwarder is allocated a fixed-size portion of the bit array. This design allocates 12 bits (i.e., 4095 as a *generator polynomial*) to each intermediate ICN forwarder. This matches the scalability of today's commercial routers that support up to 4096 physical and logical interfaces and usually do not have more than a few hundred active ones.

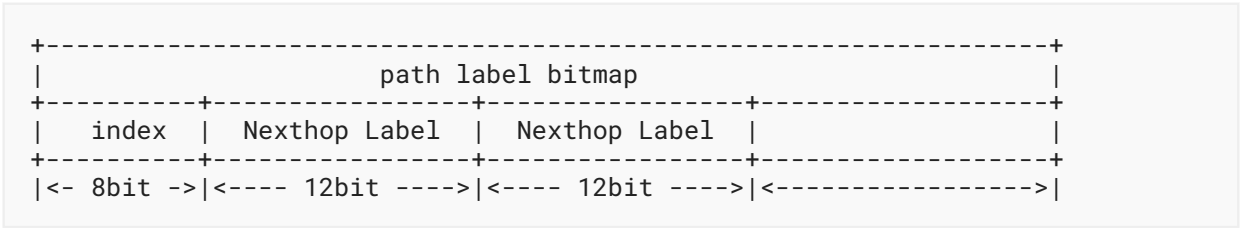


Figure 3: Fixed-Size Path Label

A forwarder that receives a Content (Data) message encodes the Nexthop Label in the next available slot and increments the label index. Conversely, a forwarder that receives an Interest message reads the current Nexthop Label and decrements the label index. Therefore, the extra computation required at each hop to forward either an Interest or Content Object message with a path label is minimized and constitutes a fairly trivial additional overhead compared to FIB lookup and other required operations.

This approach results in individual path label TLV instances being of fixed pre-computed size. While this places a hard upper bound on the maximum number of network hops that can be represented, this is not a significant practical problem in NDN and CCNx, since the size can be preset during Content (Data) message encoding based on the exact number of network hops traversed by the Interest message. Even long paths of 24 hops will fit in a path label bitmap of 36 bytes if the Nexthop Label is encoded in 12 bits.

3. Mapping to CCNx and NDN Packet Encodings

3.1. Path Label TLV

A path label TLV is the tuple: {[Flags], [Path Label Hop Count], [Nexthop Label], [path label bitmap]}.

Flag	Value (hex)
DISCOVERY_MODE	0x00
FALLBACK_MODE	0x01
STRICT_MODE	0x02
Unassigned	0x03-0xFF

Table 1: Path Label Flags

The Path Label Hop Count (PLHC) **MUST** be incremented by NDN and CCNx forwarders if the Interest packet carries a path label and the DISCOVERY_MODE flag is set. A producer node or a forwarder with a cached Data packet **MUST** use the PLHC in calculation of a path label bitmap size that is suitable for encoding the entire path to the consumer. The PLHC **MUST** be set to zero in newly created Data or InterestReturn (NACK) packets. A consumer node **MUST** reuse the PLHC together with the path label bitmap (PLB) in order to correctly forward the Interest(s) along the corresponding network path.

If an NDN or CCNx forwarder supports path labeling, the Nexthop Label **MUST** be used to determine the correct egress interface for an Interest packet carrying either the FALLBACK_MODE or the STRICT_MODE flag. If any particular NDN or CCNx forwarder is configured to decrypt path labels of Interest packets (see Security Considerations), then the forwarder **MUST**:

1. decrypt the path label with its own symmetric key,
2. update the Nexthop Label with outermost label in the path label,
3. decrement the PLHC, and
4. remove the outermost label from the path label.

If any particular NDN or CCNx forwarder is NOT configured to decrypt path labels of Interest packets, then path label decryption **SHOULD NOT** be performed.

The Nexthop Label **MUST** be ignored by NDN and CCNx forwarders if it is present in Data or InterestReturn (NACK) packets. If any particular NDN or CCNx forwarder is configured to encrypt path labels of Data and InterestReturn (NACK) packets (see Security Considerations), then the forwarder **MUST** encrypt the existing path label with its own symmetric key, append the Nexthop Label of the ingress interface to the path label, and increment the PLHC. If any particular NDN or CCNx forwarder is NOT configured to encrypt path labels of Interest packets, then path label encryption **SHOULD NOT** be performed.

NDN and CCNx forwarders **MUST** fall back to Longest Name Prefix Match (LNPM) FIB lookup if an Interest packet carries an invalid Nexthop Label and the FALLBACK_MODE flag is set.

CCNx forwarders **MUST** respond with an InterestReturn packet specifying a T_RETURN_INVALID_PATH_LABEL code if the Interest packet carries an invalid path label and the STRICT_MODE flag is set. This is a new InterestReturn code defined herein (see Section 4 for the value allocation).

CCNx forwarders **MUST** respond with an InterestReturn packet specifying the existing T_RETURN_MALFORMED_INTEREST code if the Interest packet carries a path label TLV with both the FALLBACK_MODE and STRICT_MODE flags set.

3.2. Path Label Encoding for CCNx

Path label is an optional hop-by-hop header TLV that can be present in CCNx Interest, InterestReturn, and Content Object packets.

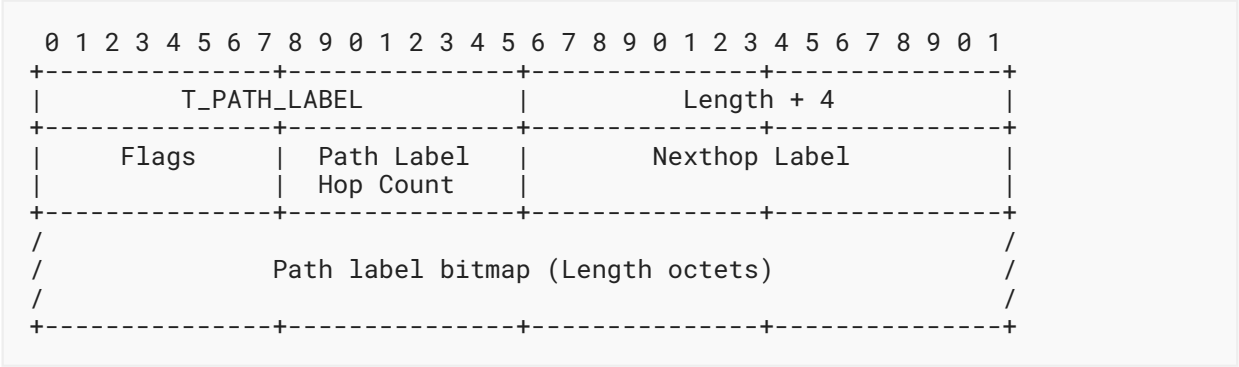


Figure 4: Path Label Hop-by-Hop Header TLV for CCNx

3.3. Path Label Encoding for NDN

Path label is an optional TLV for NDN Interest and Data packets. It is carried in the NDN Link Adaptation Protocol [NDNLPv2], which is used to wrap NDN packets for carriage over various link layer protocols. NDNLPv2 was chosen over the NDN packet itself since it can carry hop-by-hop information that potentially mutates at each hop and, therefore, cannot be included in the secured hash computation or the signature of NDN packets. Further, it can be used instead of the existing `NextHopFaceId` TLV since it not only can specify the single outgoing face for a consumer but manages the selection and forwarding over an entire path. The path label TLV in NDNLPv2 is defined below:

```
PathLabel         = PATH-LABEL-TYPE TLV-LENGTH
                    PathLabelFlags
                    PathLabelBitmap

PathLabelFlags    = PATH-LABEL-FLAGS-TYPE
                    TLV-LENGTH ; == 1
                    OCTET

NexthopLabel      = PATH-LABEL-NEXTHOP-LABEL-TYPE
                    TLV-LENGTH ; == 2
                    2 OCTET

PathLabelHopCount = PATH-LABEL-HOP-COUNT-TYPE
                    TLV-LENGTH ; == 1
                    OCTET

PathLabelBitmap   = PATH-LABEL-BITMAP-TYPE
                    TLV-LENGTH ; == 64
                    64 OCTET
```

Figure 5: Path Label TLV for NDN

Flag	(Suggested) Value (hex)
T_PATH_LABEL	0x0A
T_PATH_LABEL_FLAGS	0x0B
T_PATH_LABEL_BITMAP	0x0D
T_PATH_LABEL_NEXTHOP_LABEL	0x0E
T_PATH_LABEL_HOP_COUNT	0x0F

Table 2: TLV-TYPE Number Assignments for NDN

4. IANA Considerations

IANA has made the following assignments:

1. The value 0x000A has been assigned to T_PATH_LABEL in the "CCNx Hop-by-Hop Types" registry, established by [RFC8609].
2. The value 0x0A has been assigned to T_RETURN_INVALID_PATH_LABEL in the "CCNx Interest Return Code Types" registry, established by [RFC8609].

5. Security Considerations

A path is invalidated by renumbering one or more Nexthop Labels. A malicious consumer can attempt to mount an attack by transmitting Interests with path labels that differ only in a single now-invalid Nexthop Label in order to *brute-force* a valid Nexthop Label. If such an attack succeeds, a malicious consumer would be capable of steering Interests over a network path that may not match the paths computed by the routing algorithm or learned adaptively by the forwarders.

When a label lookup fails, by default, an *invalid path label* InterestReturn (NACK) message is returned to the consumer. This contains a path label identical to the one included in the corresponding Interest message. Therefore, a malicious consumer can analyze the message's Hop Count field to infer which specific Nexthop Label had failed and direct an attack to influence path steering at that hop. This threat can be mitigated by the following countermeasures:

- A Nexthop Label that is larger in size is harder to crack. If Nexthop Labels are not allocated in a predictable fashion by the routers, brute-forcing a 32-bit Nexthop Label requires on average $O(2^{31})$ Interests. However, this specification uses Nexthop Labels with much less entropy (12 bits), so depending on computational hardness is not workable.
- An ICN forwarder can periodically update Nexthop Labels to limit the maximum lifetime of paths. It is **RECOMMENDED** that forwarders update path labels at least every few minutes.
- A void Hop Count field in an *invalid path label* InterestReturn (NACK) message would not give out the information on which a specific Nexthop Label had failed. An attacker might

need to brute-force all Nexthop Labels in all combinations. However, some useful diagnostic capability is lost by obscuring the hop count. For example, the locus of routing churn is harder to pin down through analysis of path-steered pings or traceroutes. A forwarder **MAY** choose to invalidate the hop count in addition to changing Nexthop Labels periodically as described above.

Because ICN forwarders maintain per-face state and forwarding state for Interest messages, state inflation attacks are a general concern. The addition of path steering capabilities in Interest and Data messages does not, however, constitute a meaningful increase in susceptibility to such attacks. This is because:

- The labels that identify each forwarding face is state O(number of faces) and constitutes a small increase to the existing state needed to represent a face.
- Interest message data is placed in the PIT. The path steering header does, in fact, inflate the size of the Interest message and, hence, the PIT state but not by an amount that is a concern. The forwarder needs to protect against state inflation attacks on the PIT in general, and an attacker can mount one just as or more easily by issuing Interests with long names and/or by including Interest payload data.

ICN protocols can be susceptible to a variety of cache poisoning attacks, where a colluding consumer and producer arrange for bogus content (with either invalid or inappropriate signatures) to populate forwarder caches. These are generally confined to on-path attacks. It is also theoretically possible to launch a similar attack without a cooperating producer such that the caches of on-path routers become poisoned with the content from off-path routers (i.e., physical connectivity but no route in a FIB for a given prefix). We estimate that, without any prior knowledge of the network topology, the complexity of this type of attack is in the ballpark of Breadth-First-Search and Depth-First-Search algorithms with the additional burden of transmitting 2^{31} Interests in order to crack a Nexthop Label on each hop. A relatively short periodic update of Nexthop Labels, together with heuristics implemented in the ICN forwarder to foil *label scans*, may successfully mitigate this type of attack.

5.1. Cryptographic Protection of a Path Label

If the countermeasures listed above do not provide sufficient protection against malicious mis-steering of Interests, the path label can be made opaque to the consumer endpoint via hop-by-hop symmetric cryptography applied to the path labels (Figure 6). This method is viable due to the symmetry of forward and reverse paths in CCNx and NDN architectures combined with ICN path steering requiring only reads and writes of the topmost Nexthop Label (i.e., active Nexthop Label) in the path label. This way, a path-steering-capable ICN forwarder receiving a Content (Data) message encrypts the current path label with its own non-shared symmetric key prior to adding a new Nexthop Label to the path label. The Content (Data) message is forwarded downstream with an unencrypted topmost (i.e., active) Nexthop Label and the remaining encrypted content of the path label. As a result, a consumer endpoint receives a Content (Data) message with a unique path label exposing only the topmost Nexthop Label as cleartext. A path

steering forwarder receiving an Interest message performs label lookup using the topmost Nexthop Label, decrypts the path label with its own non-shared symmetric key, and forwards the message upstream.

Cryptographic protection of a path label does not require any key negotiation among ICN forwarders and is no more expensive than Media Access Control Security (MACsec) or IPsec. It is also quite possible that strict hop-by-hop path label encryption is not necessary and path label encryption only on the border routers of the trusted administrative or routing domains may suffice.

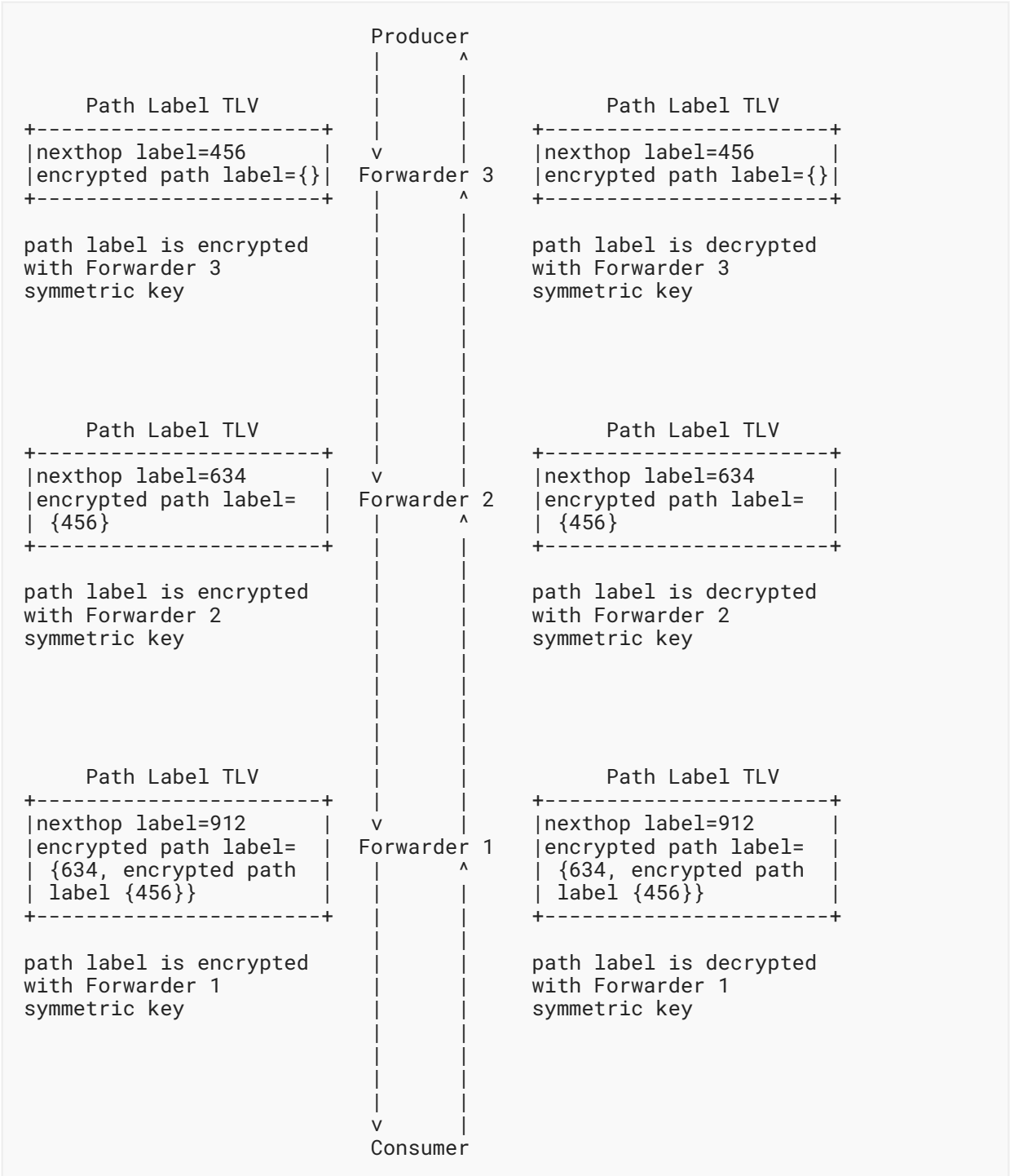


Figure 6: Path Label Protection with Hop-by-Hop Symmetric Cryptography

6. References

6.1. Normative References

[Moiseenko2017] Moiseenko, I. and D. Oran, "Path Switching in Content Centric and Named Data Networks", Proceedings of the 4th ACM Conference on Information-Centric Networking, Pages 66-76, DOI 10.1145/3125719.3125721, DOI 10.1145/3125719.3125721, September 2017, <https://conferences.sigcomm.org/acm-icn/2017/proceedings/icn17-2.pdf>.

[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>.

6.2. Informative References

[FLIC] Tschudin, C., Wood, C. A., Mosko, M., and D. Oran, Ed., "File-Like ICN Collections (FLIC)", Work in Progress, Internet-Draft, draft-irtf-icnrg-flic-05, 22 October 2023, <https://datatracker.ietf.org/doc/html/draft-irtf-icnrg-flic-05>.

[Mahdian2016] Mahdian, M., Arianfar, S., Gibson, J., and D. Oran, "MIRCC: Multipath-aware ICN Rate-based Congestion Control", Proceedings of the 3rd ACM Conference on Information-Centric Networking, Pages 1-10, DOI 10.1145/2984356.2984365, September 2016, <http://conferences2.sigcomm.org/acm-icn/2016/proceedings/p1-mahdian.pdf>.

[NDN] NDN, "Named Data Networking: Executive Summary", <https://named-data.net/project/execsummary/>.

[NDNLPv2] NFD, "NDNLPv2", <https://redmine.named-data.net/projects/nfd/wiki/NDNLPv2>.

[NDNTLV] NDN, "NDN Packet Format Specification v0.3", <https://named-data.net/doc/NDN-packet-spec/current/>.

[RFC8029] Kompella, K., Swallow, G., Pignataro, C., Ed., Kumar, N., Aldrin, S., and M. Chen, "Detecting Multiprotocol Label Switched (MPLS) Data-Plane Failures", RFC 8029, DOI 10.17487/RFC8029, March 2017, <https://www.rfc-editor.org/info/rfc8029>.

[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>.

[RFC9217] Trammell, B., "Current Open Questions in Path-Aware Networking", RFC 9217, DOI 10.17487/RFC9217, March 2022, <https://www.rfc-editor.org/info/rfc9217>.

[RFC9507] Mastorakis, S., Oran, D., Moiseenko, I., Gibson, J., and R. Droms, "Information-Centric Networking (ICN) Traceroute Protocol Specification", RFC 9507, DOI 10.17487/RFC9507, February 2024, <https://www.rfc-editor.org/info/rfc9507>.

[RFC9508] Mastorakis, S., Oran, D., Gibson, J., Moiseenko, I., and R. Droms, "Information-Centric Networking (ICN) Ping Protocol Specification", RFC 9508, DOI 10.17487/RFC9508, February 2024, <https://www.rfc-editor.org/info/rfc9508>.

[SCION] de Kater, C., Rustignoli, N., and A. Perrig, "SCION Overview", Work in Progress, Internet-Draft, draft-dekater-panrg-scion-overview-05, 5 November 2023, <https://datatracker.ietf.org/doc/html/draft-dekater-panrg-scion-overview-05>.

[Song2018] Song, J., Lee, M., and T. Kwon, "SMIC: Subflow-level Multi-path Interest Control for Information Centric Networking", Proceedings of the 5th ACM Conference on Information-Centric Networking, Pages 77-87, DOI 10.1145/3267955.3267971, September 2018, <https://conferences.sigcomm.org/acm-icn/2018/proceedings/icn18-final62.pdf>.

Authors' Addresses

Ilya Moiseenko
Apple, Inc.
Cupertino, CA
United States of America
Email: iliamo@mailbox.org

Dave Oran
Network Systems Research and Design
4 Shady Hill Square
Cambridge, MA 02138
United States of America
Email: daveoran@orandom.net