ICNRG
Internet-Draft
Intended status: Experimental
Expires: April 22, 2020

I. Moiseenko
UCLA
D. Oran
Network Systems Research and Design

October 20, 2019

Path Steering in CCNx and NDN
draft-oran-icnrg-pathsteering-00

Abstract

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 multipath 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 - ICN'17) and therefore does not recapitulate the design motivations, implementation details, or evaluation of the scheme. Some technical details are different however, and where there are differences, the design documented here is to be considered definitive.

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 April 22, 2020.

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 multipath 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. Some technical details are different however, and where there are differences, the design documented here is to be considered definitive.

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 tree trace [RFC8029] protocol is used. Equivalent diagnostic functions have been defined for CCNx through the ICN Ping [I-D.mastorakis-icnrg-icnping] and ICN Traceroute [I-D.mastorakis-icnrg-icntraceroute] 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 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.

- A consumer endpoint can mitigate content poisoning attacks by directing its Interests onto the network paths that bypass poisoned caches.

Path discovery and subsequent path steering in ICN networks is facilitated by the symmetry of forward and reverse paths in the CCNx and 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) FIB lookup.

1.1. 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].

2. Essential elements of ICN path discovery and path steering

We elucidate the design using CCNx semantics [RFC8569] and extend its Packet Encoding [RFC8609] as defined in Section 3.2. While the terminology is slightly different, this design can be applied also 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 (Figure 1). The path label is updated by adding to the existing path label the nexthop label of the interface at which the Content (Data) message has arrived. 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.

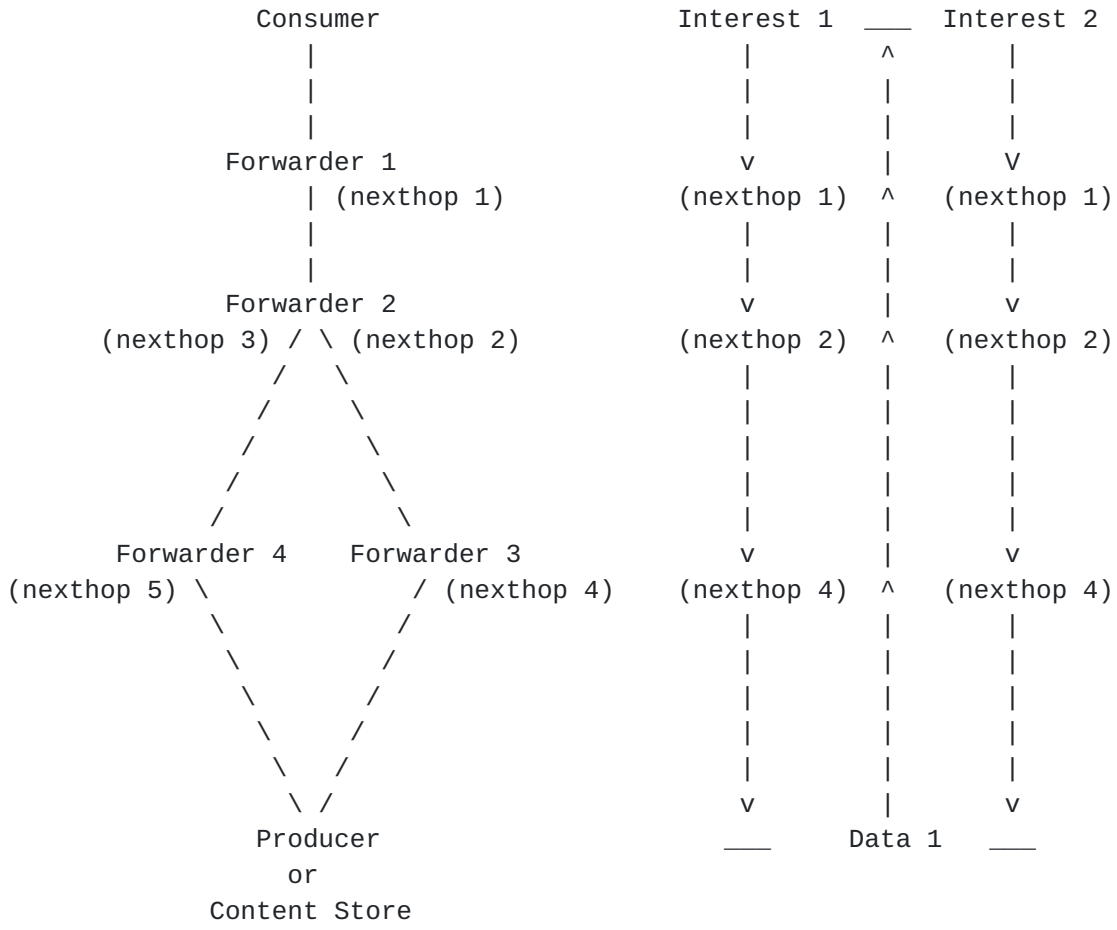


Figure 1: Basic example of path discovery and steering

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 similar (e.g. next chunk, or next Application Data Unit, or next pointer in a Manifest [I-D.icnrg-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).

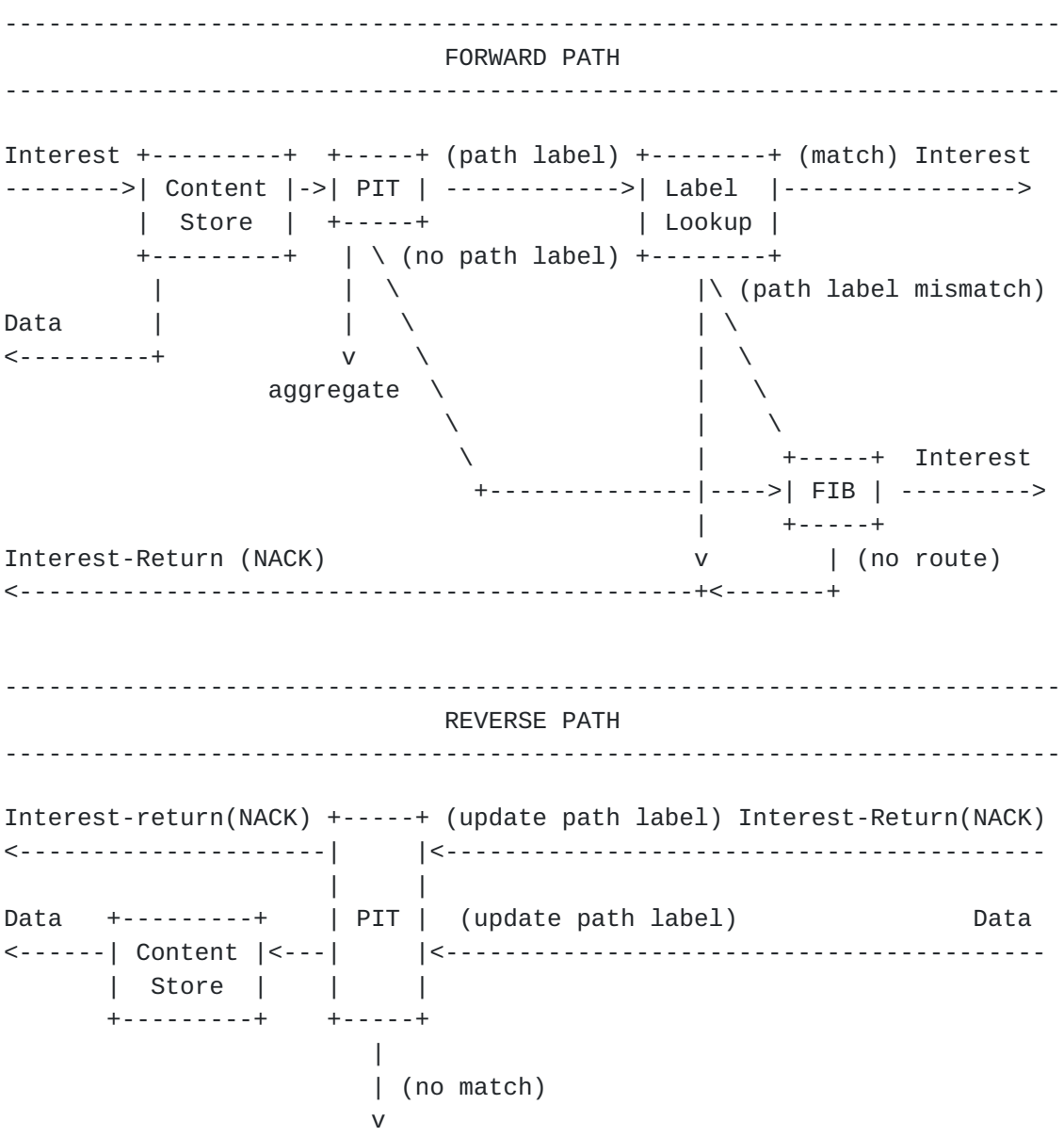


Figure 2: Path Steering CCN / NDN data plane

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 Interest-Return. We therefore define a new

Invalid path label response code for the Interest Return message and include the current path label as a hop-by-hop header. Each transit forwarder processing the Interest-Return message updates the path label in the same manner as Content (Data) messages, so that the consumer receiving the Interest-Return (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 returning 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. How to represent the Path Label

[Moiseenko2017] presents various options for how to represent a path label, with different tradeoffs 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 sized portion of the bit array. This design allocates 12 bits (i.e. 4095 as a _generator polynomial_) to each intermediate ICN forwarder. This should match 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.

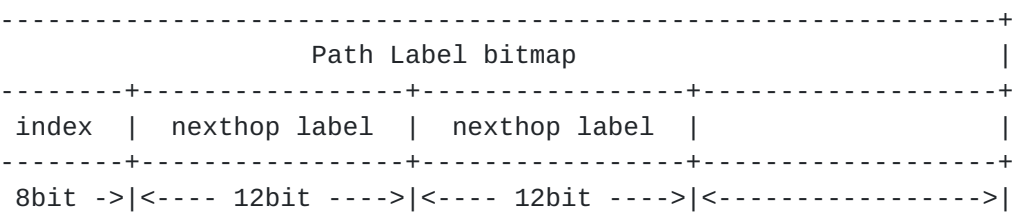


Figure 3: Fixed size path label

A forwarder that receives a Content (Data) message encodes the nexthop label in the next available slot and increments label index. Conversely, a forwarder that receives an Interest message reads the current nexthop label and decrements 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 a practical problem in NDN and CCNx, since the size can be pre-set 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 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

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 DISCOVERY mode flag is set. A producer node or a forwarder with cached data packet MUST use PLHC in calculation of a path label bitmap size suitable for encoding the entire path to the consumer. The Path Label Hop Count (PLHC) MUST be set to zero in newly created Data or Interest-Return (NACK) packets. A consumer node MUST reuse Path Label Hop Count (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 STRICT MODE flag. If any particular NDN or CCNx forwarder is configured to decrypt path labels of Interest packets (Section "Security considerations" (Section 5)), 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 Path Label Hop Count (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 present in Data or Interest-Return (NACK) packets. If any particular NDN or CCNx forwarder is configured to encrypt path labels of Data and Interest-Return (NACK) packets (Section Security Considerations (Section 5)), then the forwarder MUST encrypt existing path label with its own symmetric key, append the nexthop label of the ingress interface to the path label, and increment Path Label Hop Count (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 fallback 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 Interest Return packet specifying the T_RETURN_INVALID_PATH_LABEL code if Interest packet carries an invalid path label and the STRICT MODE flag is set.

CCNx forwarders MUST respond with an Interest Return packet specifying the T_RETURN_MALFORMED_INTEREST code if the Interest packet carries a path label TLV with both 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.

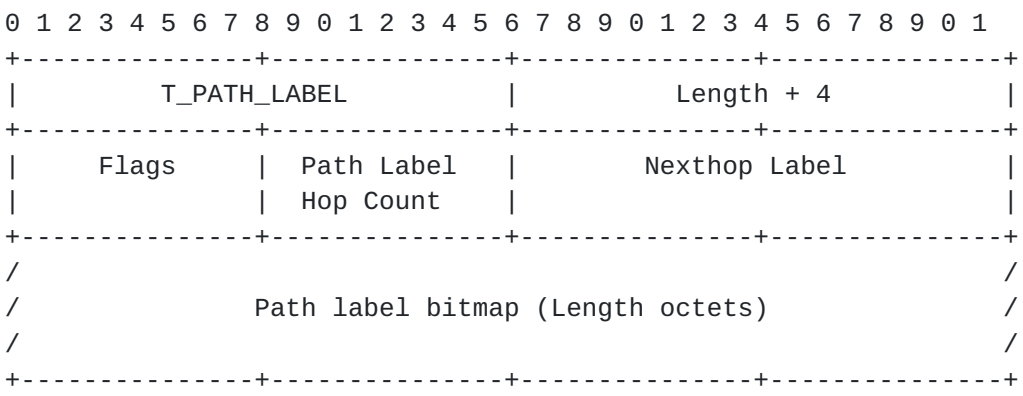


Figure 4: Path label Hop-by-Hop header TLV for CCNx

3.3. Path label encoding for NDN

Path Label is an optional TLV in NDN Interest, Data and NACK packets. The Path Label TLV SHOULD NOT take part in Interest, Data or NACK signature calculation as it is potentially modified at every network hop.

```
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	Value (hex)
PATH-LABEL-TYPE	0x09
PATH-LABEL-FLAGS-TYPE	0x0B
PATH-LABEL-BITMAP-TYPE	0x0D
PATH-LABEL-NEXTHOP-LABEL-TYPE	0x0E
PATH-LABEL-HOP-COUNT-TYPE	0x0F

Table 2: TLV-TYPE number assignments

4. IANA Considerations

IANA is requested to make the following assignments:

1. Please assign the value 0x0004 for T_PATH_LABEL in the *CCNx Hop-by-Hop Types* registry.

2. Please assign the TLV types in Table 2 in the *CCNx Hop-by-Hop type registry*.

3. Please assign the value 0xA for the T_RETURN_INVALID_PATH_LABEL in the *CCNx Interest Return Code Types" registry*.

4. Please create the *CCNx Path Label Flags* registry and assign the values listed in Table 1.

5. Security Considerations

A path is invalidated by renumbering nexthop label(s). A malicious consumer can attempt to mount an attack by transmitting Interests with path labels which 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_ Interest-Return (NACK) message is returned to the consumer. This contains a path label identical to the one included in the corresponding Interest message. A malicious consumer can therefore 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 of larger 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_ Interest-Return (NACK) message would not give out the information on which 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 above.

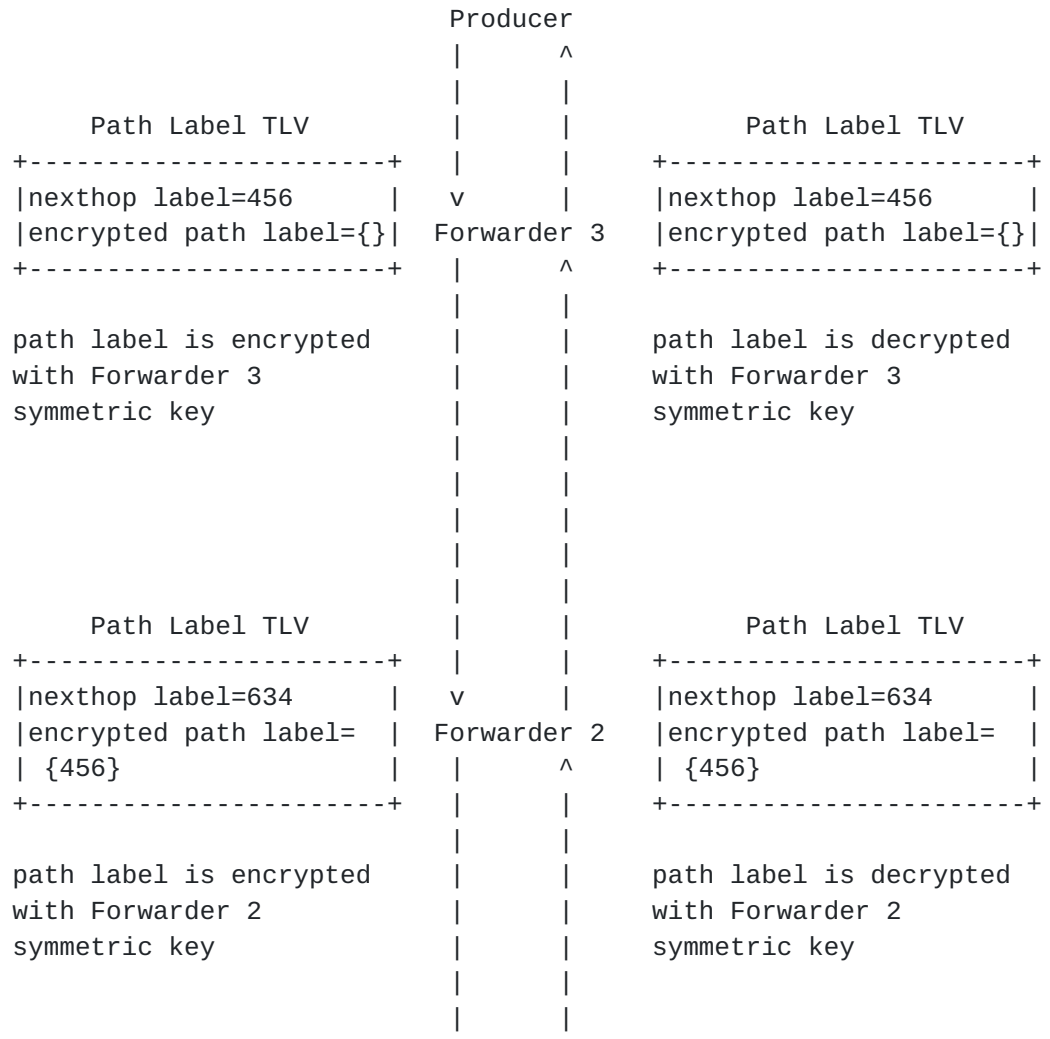
ICN protocols can be susceptible to a variety of cache poisoning attacks, where a colluding consumer and producer arrange for bogus content (with either invalid or inappropriate signatures) to populate forwarder caches. These are generally confined to on-path attacks. It is also theoretically possible 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. Relatively short periodic update of nexthop labels and anti- _label scan_ heuristics implemented in the ICN forwarder 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/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 Data (Content) 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 Data (Content) message is forwarded downstream with unencrypted topmost (i.e active) nexthop label and encrypted remaining content of the path label. As a result, a consumer endpoint receives a Data (Content) 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 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.



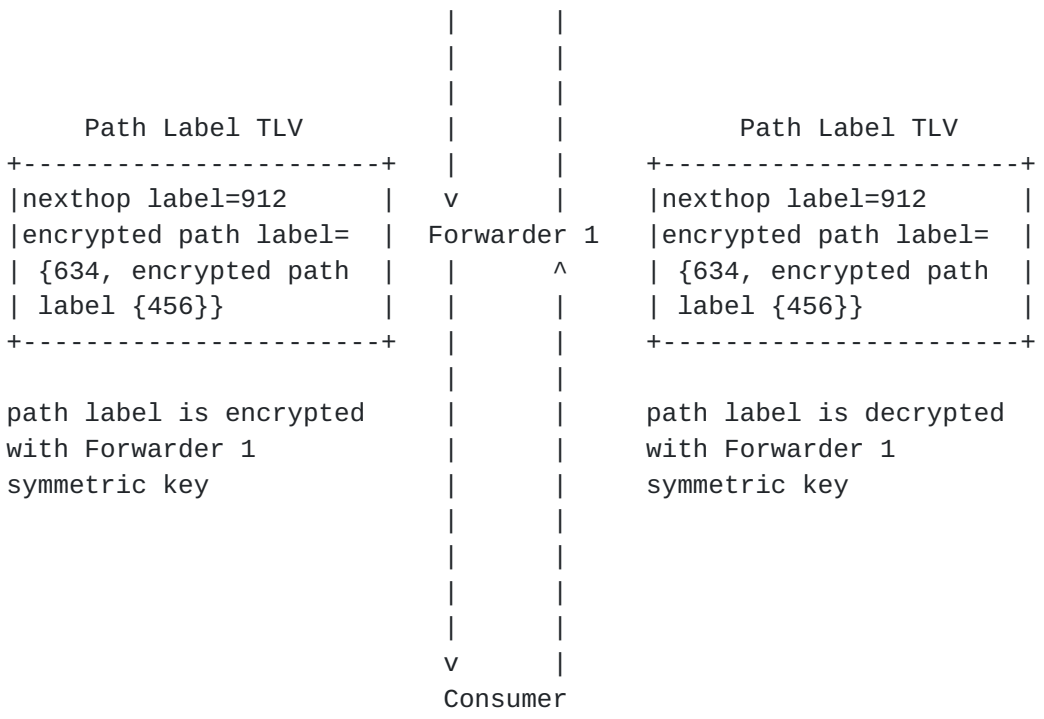


Figure 6: Path label protection with hop-by-hop symmetric cryptography

6. References

6.1. Normative References

[Moiseenko2017]
Moiseenko, I. and D. Oran, "Path Switching in Content Centric and Named Data Networks, in 4th ACM Conference on Information-Centric Networking (ICN 2017)", 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>.

[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

[I-D.icnrg-flic]
Tschudin, C. and C. Wood, "File-Like ICN Collection (FLIC)", draft-icnrg-flic-00 (work in progress), June 2017.

[I-D.mastorakis-icnrg-icnping]
Mastorakis, S., Gibson, J., Moiseenko, I., Droms, R., and D. Oran, "ICN Ping Protocol Specification", draft-mastorakis-icnrg-icnping-05 (work in progress), August 2019.

[I-D.mastorakis-icnrg-icntraceroute]
Mastorakis, S., Gibson, J., Moiseenko, I., Droms, R., and D. Oran, "ICN Traceroute Protocol Specification", draft-mastorakis-icnrg-icntraceroute-05 (work in progress), August 2019.

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

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

[NDNTLV] "NDN Packet Format Specification.", 2016, <http://named-data.net/doc/ndn-tlv/>.

[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>.

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

Authors' Addresses

Ilya Moiseenko
UCLA

Email: iliamo@ucla.edu

Dave Oran
Network Systems Research and Design
4 Shady Hill Square
Cambridge, MA 02138
USA

Email: daveoran@orandom.net