
Stream: Internet Research Task Force (IRTF)
RFC: [9507](#)
Category: Experimental
Published: March 2024
ISSN: 2070-1721
Authors: S. Mastorakis D. Oran I. Moiseenko
University of Notre Dame Network Systems Research and Design Apple Inc.
J. Gibson R. Droms
Unaffiliated Unaffiliated

RFC 9507

Information-Centric Networking (ICN) Traceroute Protocol Specification

Abstract

This document presents the design of an Information-Centric Networking (ICN) Traceroute protocol. This includes the operation of both the client and the forwarder.

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

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/rfc9507>.

Copyright Notice

Copyright (c) 2024 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
1.1. Requirements Language	3
2. Background on IP-Based Traceroute Operation	3
3. Traceroute Functionality Challenges and Opportunities in ICN	4
4. ICN Traceroute CCNx Packet Formats	5
4.1. ICN Traceroute Request CCNx Packet Format	6
4.2. ICN Traceroute Reply CCNx Packet Format	7
5. ICN Traceroute NDN Packet Formats	10
5.1. ICN Traceroute Request NDN Packet Format	10
5.2. ICN Traceroute Reply NDN Packet Format	10
6. Forwarder Operation	11
7. Protocol Operation for Locally Scoped Namespaces	12
8. Security Considerations	13
9. IANA Considerations	14
10. References	14
10.1. Normative References	14
10.2. Informative References	14
Appendix A. Traceroute Client Application (Consumer) Operation	15
Authors' Addresses	16

1. Introduction

In TCP/IP, routing and forwarding are based on IP addresses. To ascertain the route to an IP address and to measure the transit delays, the traceroute utility is commonly used. In Information-Centric Networking (ICN), routing and forwarding are based on name prefixes. To this end, the ability to ascertain the characteristics of at least one of the available routes to a

name prefix is a fundamental requirement for instrumentation and network management. These characteristics include, among others, route properties such as which forwarders were transited and the delay incurred through forwarding.

In order to carry out meaningful experimentation and deployment of ICN protocols, new tools analogous to ping and traceroute used for TCP/IP are needed to manage and debug the operation of ICN architectures and protocols. This document describes the design of a management and debugging protocol analogous to the traceroute protocol of TCP/IP; this new management and debugging protocol will aid the experimental deployment of ICN protocols. As the community continues its experimentation with ICN architectures and protocols, the design of ICN Traceroute might change accordingly. ICN Traceroute is designed as a tool to troubleshoot ICN architectures and protocols. As such, this document is classified as an Experimental RFC.

This specification uses the terminology defined in [\[RFC8793\]](#).

This RFC represents the consensus of the Information-Centric Networking Research Group (ICNRG) of the Internet Research Task Force (IRTF).

1.1. 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.

2. Background on IP-Based Traceroute Operation

In IP-based networks, traceroute is based on the expiration of the Time To Live (TTL) IP header field. Specifically, a traceroute client sends consecutive packets (depending on the implementation and the user-specified behavior, such packets can be either UDP datagrams, ICMP Echo Request packets, or TCP SYN packets) with a TTL value increased by 1, essentially performing an expanding ring search. In this way, the first IP packet sent will expire at the first router along the path, the second IP packet at the second router along the path, etc., until the router (or host) with the specified destination IP address is reached. Each router along the path towards the destination responds by sending back an ICMP Time Exceeded packet, unless explicitly prevented from doing so by a security policy.

The IP-based traceroute utility operates on IP addresses and in particular depends on the IP packets having source IP addresses that are used as the destination address for replies. Given that ICN forwards based on names rather than destination IP addresses, that the names do not refer to unique endpoints (multi-destination), and that the packets do not contain source addresses, a substantially different approach is needed.

3. Traceroute Functionality Challenges and Opportunities in ICN

In the Named Data Networking (NDN) and Content-Centric Networking (CCNx) protocols, the communication paradigm is based exclusively on named objects. An Interest message is forwarded across the network based on its name. Eventually, it retrieves a Content Object from either a producer application or some forwarder's Content Store (CS).

An ICN network differs from an IP network in at least four important ways (four of which are as follows):

- IP identifies interfaces to an IP network with a fixed-length address and delivers IP packets to one or more interfaces. ICN identifies units of data in the network with a variable-length name consisting of a hierarchical list of segments.
- An IP-based network depends on the IP packets having source IP addresses that are used as the destination address for replies. On the other hand, ICN Interests do not have source addresses, and they are forwarded based on names, which do not refer to a unique endpoint. Data packets follow the reverse path of the Interests based on hop-by-hop state created during Interest forwarding.
- An IP network supports multi-path, single-destination, stateless packet forwarding and delivery via unicast; a limited form of multi-destination selected delivery with anycast; and group-based multi-destination delivery via multicast. In contrast, ICN supports multi-path and multi-destination stateful Interest forwarding and multi-destination data delivery to units of named data. This single forwarding semantic subsumes the functions of unicast, anycast, and multicast. As a result, consecutive (or retransmitted) ICN Interest messages may be forwarded through an ICN network along different paths and may be forwarded to different data sources (e.g., end-node applications, in-network storage) holding a copy of the requested unit of data. The ability to discover multiple available (or potentially all) paths towards a name prefix is a desirable capability for an ICN Traceroute protocol, since it can be beneficial for congestion control purposes. Knowing the number of available paths for a name can also be useful in cases where Interest forwarding based on application semantics/preferences is desirable.
- In the case of multiple Interests with the same name arriving at a forwarder, a number of Interests may be aggregated in a common Pending Interest Table (PIT) entry. Depending on the lifetime of a PIT entry, the round-trip time of an Interest-Data exchange might vary significantly (e.g., it might be shorter than the full round-trip time to reach the original content producer). To this end, the round-trip time experienced by consumers might also vary even under constant network load.

These differences introduce new challenges, new opportunities, and new requirements regarding the design of ICN Traceroute. Following this communication model, a traceroute client should be able to express traceroute requests directed to a name prefix and receive responses.

Our goals are as follows:

- Trace one or more paths towards an ICN forwarder (for troubleshooting purposes).
- Trace one or more paths through which a named data object can be reached in the sense that Interest packets can be forwarded towards the application hosting the object.
- Test whether a specific named object is cached in some on-path CS, and, if so, trace the path towards it and return the identity of the corresponding forwarder.
- Perform transit delay network measurements.

To this end, a traceroute target name can represent:

- An administrative name that has been assigned to a forwarder. Assigning a name to a forwarder implies the presence of a management application running locally that handles Operations, Administration, and Maintenance (OAM) operations.
- A name that includes an application's namespace as a prefix.
- A named object that might reside in some in-network storage.

In order to provide stable and reliable diagnostics, it is desirable that the packet encoding of a traceroute request enable the forwarders to distinguish this request from a normal Interest while also diverging as little as possible from the forwarding behavior for an Interest packet. In the same way, the encoding of a traceroute reply should minimize any processing differences from those employed for a data packet by the forwarders.

The term "traceroute session" is used for an iterative process during which an endpoint client application generates a number of traceroute requests to successively traverse more distant hops in the path until it receives a final traceroute reply from a forwarder. It is desirable that ICN Traceroute be able to discover a number of paths towards the expressed prefix within the same session or subsequent sessions. To discover all the hops in a path, we need a mechanism (Interest Steering) to steer requests along different paths. Such a capability was initially published in [[PATHSTEERING](#)] and has been specified for CCNx and NDN in [[RFC9531](#)].

In the case of traceroute requests for the same prefix from different sources, it is also important to have a mechanism to avoid aggregating those requests in the PIT. To this end, we need some encoding in the traceroute requests to make each request for a common prefix unique, hence avoiding PIT aggregation and further enabling the exact matching of a response with a particular traceroute packet.

The packet types and formats are presented in [Section 4](#). Procedures for determining and indicating that a destination has been reached are included in [Section 6](#).

4. ICN Traceroute CCNx Packet Formats

In this section, we present the CCNx packet formats [[RFC8609](#)] of ICN Traceroute where messages exist within outermost containments (packets). Specifically, we propose two types of traceroute packets: a traceroute request and a traceroute reply.

4.1. ICN Traceroute Request CCNx Packet Format

The format of the traceroute request packet is presented below:

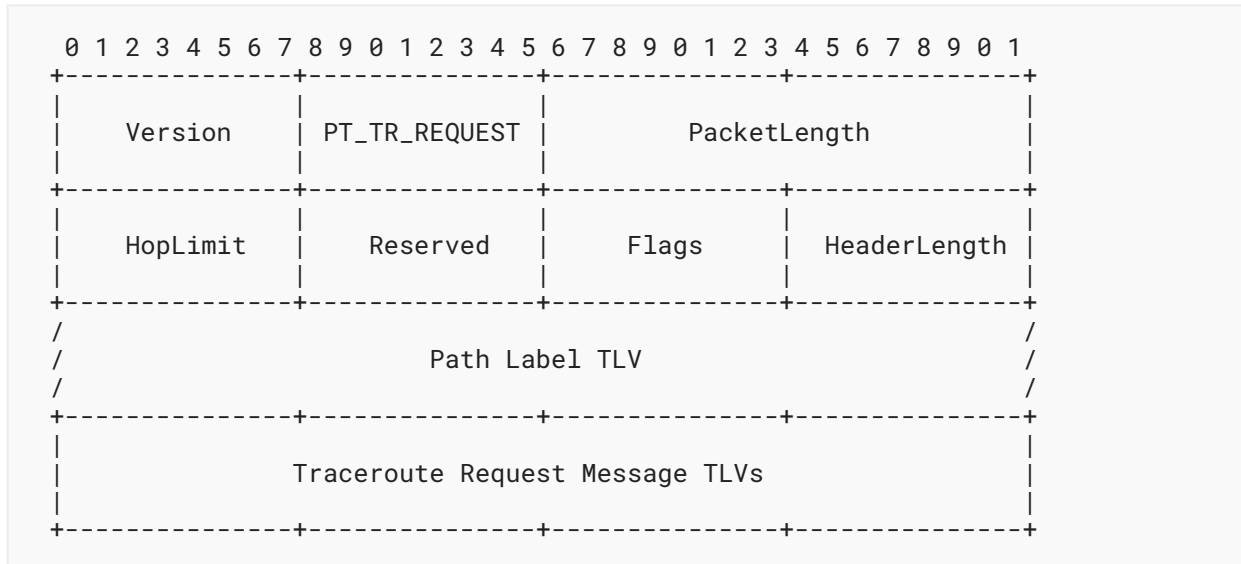


Figure 1: Traceroute Request CCNx Packet Format

The existing packet header fields have functionality similar to that of the header fields of a CCNx Interest packet. The value of the packet type field is PT_TR_REQUEST. See [Section 9](#) for the value assignment.

In contrast to the typical format of a CCNx packet header [[RFC8609](#)], there is a new optional fixed header added to the packet header:

- A Path Steering hop-by-hop header TLV, which is constructed hop by hop in the traceroute reply and included in the traceroute request to steer consecutive requests expressed by a client towards a common forwarding path or different forwarding paths. The Path Label TLV is specified in [[RFC9531](#)].

The message of a traceroute request is presented below:

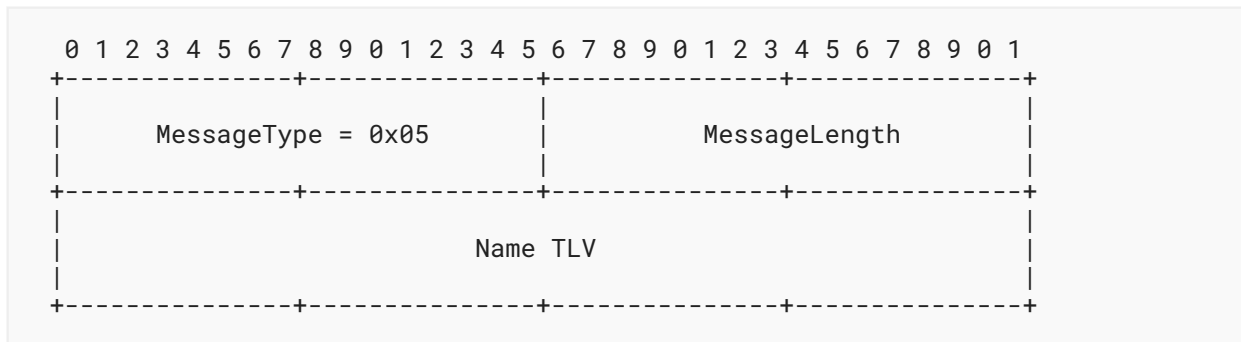


Figure 2: Traceroute Request Message Format

The traceroute request message is of type T_DISCOVERY. The Name TLV has the structure described in [RFC8609]. The name consists of the target (destination) prefix appended with a nonce typed name as its last segment. The nonce can be encoded as a base64-encoded string with the URL-safe alphabet as defined in Section 5 of [RFC4648], with padding omitted. The format of this TLV is a 64-bit nonce. See [RFC9508] for the value assignment. The purpose of the nonce is to avoid Interest aggregation and allow client matching of replies with requests. As described below, the nonce is ignored for CS checking.

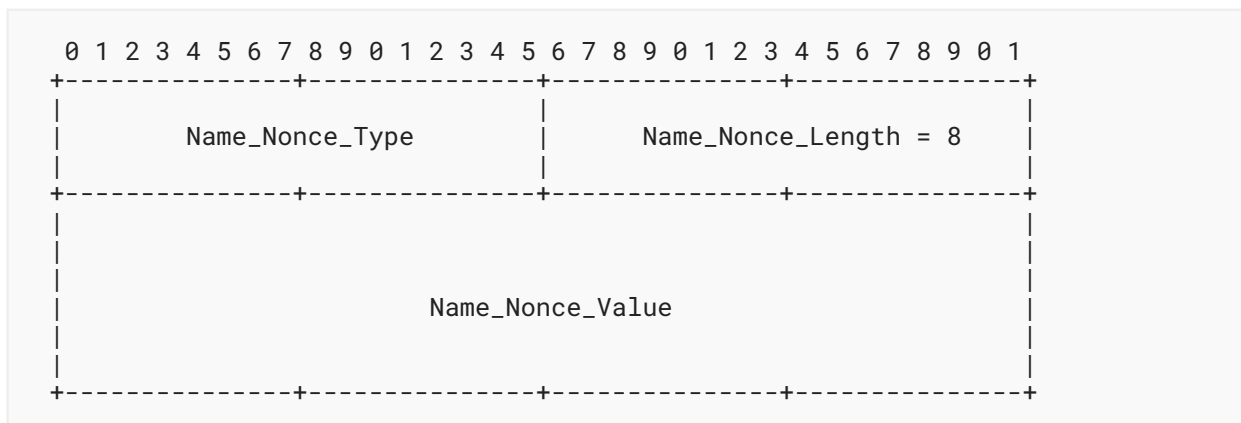


Figure 3: Name Nonce Typed Segment TLV

4.2. ICN Traceroute Reply CCNx Packet Format

The format of a traceroute reply packet is presented below:

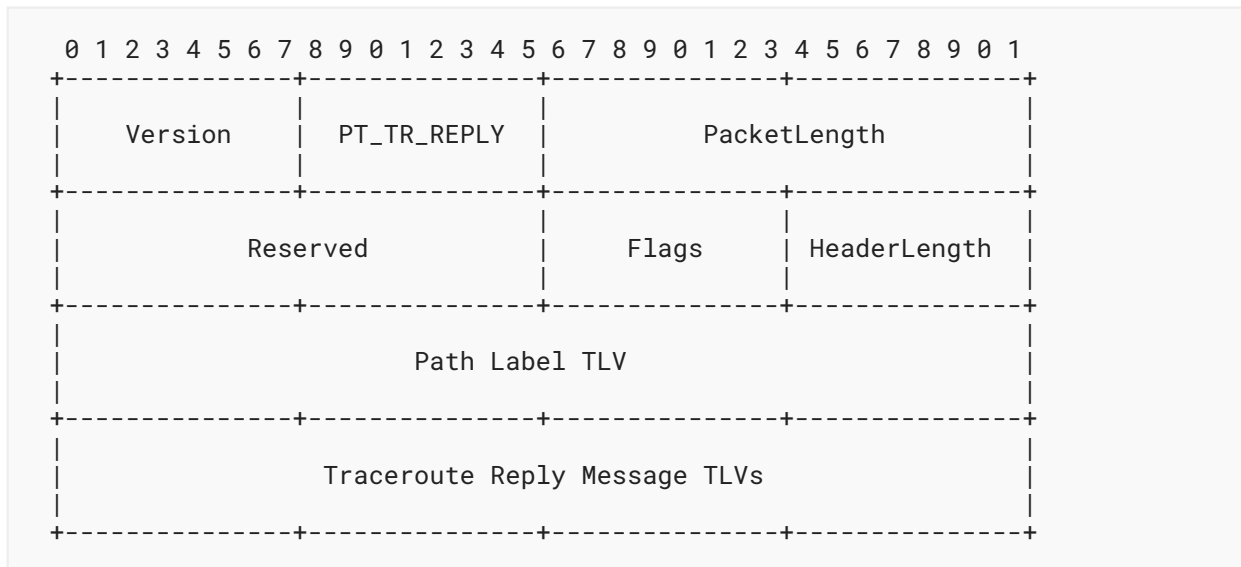


Figure 4: Traceroute Reply CCNx Packet Format

The header of a traceroute reply consists of the header fields of a CCNx Content Object and a hop-by-hop Path Steering TLV. The value of the packet type field is PT_TR_REPLY. See [Section 9](#) for the value assignment.

A traceroute reply message is of type T_OBJECT and contains a Name TLV (name of the corresponding traceroute request), a PayloadType TLV, and an ExpiryTime TLV with a value of 0 to indicate that replies must not be returned from network caches.

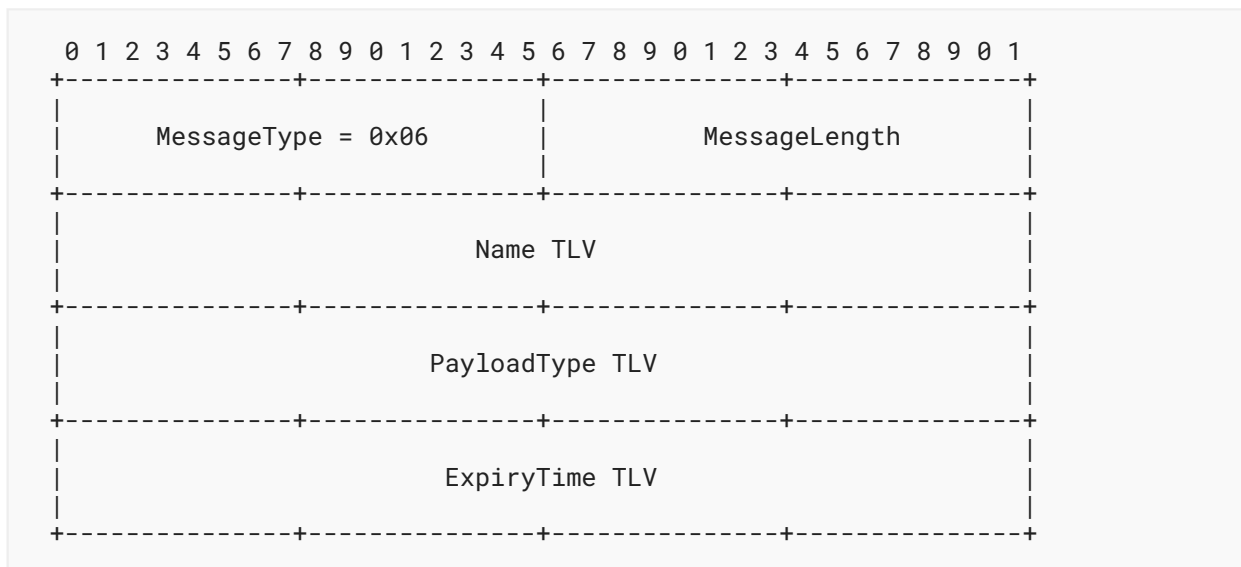


Figure 5: Traceroute Reply Message Format

The PayloadType TLV is presented below. It is of type T_PAYLOADTYPE_DATA, and the data schema consists of three TLVs:

- 1) the name of the sender of this reply (with the same structure as a CCNx Name TLV),
- 2) the sender's signature of their own name (with the same structure as a CCNx ValidationPayload TLV), and
- 3) a TLV with return codes to indicate whether the request was satisfied due to the existence of a local application, a CS hit, a match with a forwarder's name, or the HopLimit value of the corresponding request reaching 0.

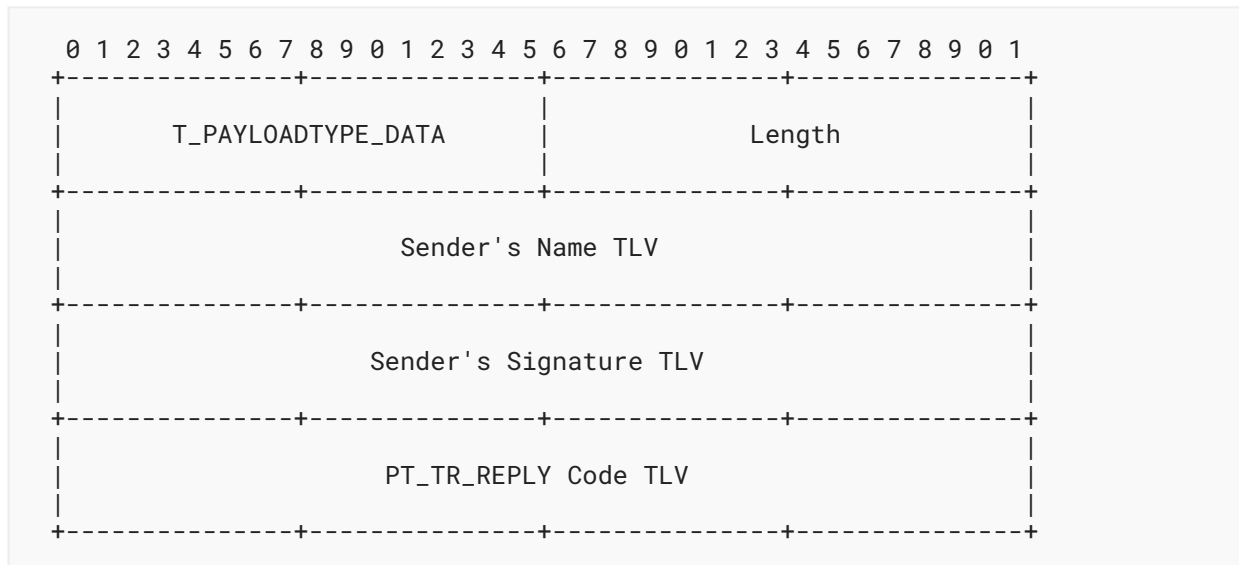


Figure 6: Traceroute Reply PayloadType TLV Format

The goal of including the name of the sender in the reply is to enable the user to reach this entity directly to ask for further management/administrative information using generic Interest-Data exchanges or by employing a more comprehensive management tool, such as [CCNinfo \[RFC9344\]](#), after a successful verification of the sender's name.

The structure of the PT_TR_REPLY Code TLV is presented below (16-bit value). The four assigned values are as follows:

- 1: Indicates that the target name matched the administrative name of a forwarder (as served by its internal management application).
- 2: Indicates that the target name matched a prefix served by an application (other than the internal management application of a forwarder).
- 3: Indicates that the target name matched the name of an object in a forwarder's CS.
- 4: Indicates that the HopLimit reached 0.

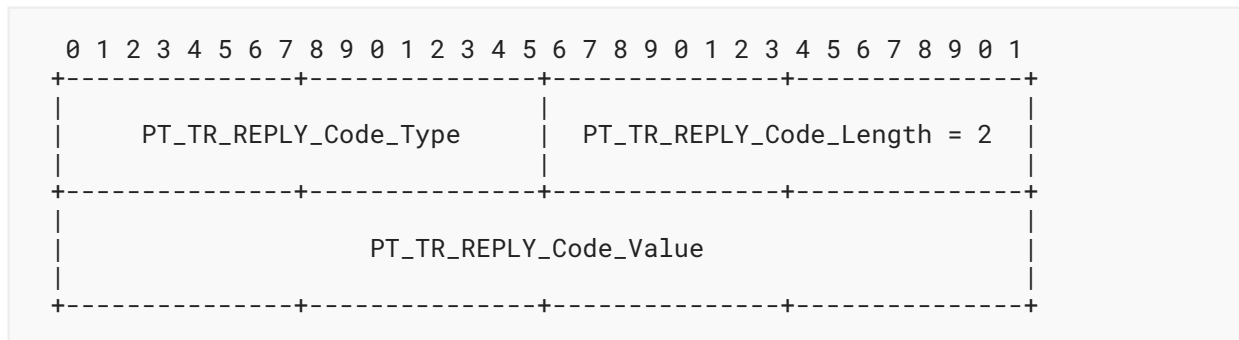


Figure 7: PT_TR_REPLY Code TLV

5. ICN Traceroute NDN Packet Formats

In this section, we present the ICN Traceroute Request and Reply packet formats according to the NDN packet format specification [NDNTLV].

5.1. ICN Traceroute Request NDN Packet Format

A traceroute request is encoded as an NDN Interest packet. Its format is as follows:

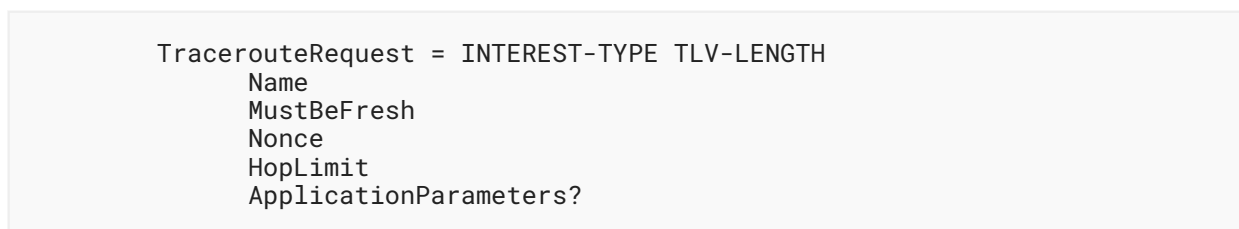


Figure 8: Traceroute Request NDN Packet Format

The name of a request consists of the target name, a nonce value (it can be the value of the Nonce field), and the suffix "traceroute" to denote that this Interest is a traceroute request (added as a KeywordNameComponent [NDNTLV]). When the "ApplicationParameters" element is present, a ParametersSha256DigestComponent (Section 6) is added as the last name segment.

A traceroute request **MAY** carry a Path Label TLV in the [NDN Link Adaptation Protocol \[NDNLPv2\]](#) as specified in [RFC9531].

Since the NDN packet format does not provide a mechanism to prevent the network from caching specific data packets, we instead use the MustBeFresh TLV for requests (in combination with a FreshnessPeriod TLV with a value of 1 for replies) to avoid fetching cached traceroute replies with a freshness period that has expired [REALTIME].

5.2. ICN Traceroute Reply NDN Packet Format

A traceroute reply is encoded as an NDN Data packet. Its format is as follows:

```
TracerouteReply = DATA-TLV TLV-LENGTH
                  Name
                  MetaInfo
                  Content
                  Signature
```

Figure 9: Traceroute Reply NDN Packet Format

A traceroute reply **MAY** carry a Path Label TLV in the [NDN Link Adaptation Protocol \[NDNLPv2\]](#) as specified in [\[RFC9531\]](#), since it might be modified in a hop-by-hop fashion by the forwarders along the reverse path.

The name of a traceroute reply is the name of the corresponding traceroute request while the format of the MetaInfo field is as follows:

```
MetaInfo = META-INFO-TYPE TLV-LENGTH
           ContentType
           FreshnessPeriod
```

Figure 10: MetaInfo TLV

The value of the ContentType TLV is 0. The value of the FreshnessPeriod TLV is 1, so that the replies are treated as stale data (almost instantly) as they are received by a forwarder.

The content of a traceroute reply consists of the following two TLVs: Sender's Name (an NDN Name TLV) and Traceroute Reply Code. There is no need to have a separate TLV for the sender's signature in the content of the reply, since every NDN Data packet carries the signature of the data producer.

The Traceroute Reply Code TLV format is as follows (with the values specified in [Section 4.2](#)):

```
PT_TR_REPLYCode = TRREPLYCODE-TLV-TYPE TLV-LENGTH 2*OCTET
```

Figure 11: Traceroute Reply Code TLV

6. Forwarder Operation

When a forwarder receives a traceroute request, the HopLimit value is checked and decremented, and the target name (i.e., the name of the traceroute request without the last Nonce name segment as well as the suffix "traceroute" and the ParametersSha256DigestComponent in the case of a request with the NDN packet format) is extracted.

If the HopLimit has not expired (i.e., is greater than 0), the forwarder will forward the request upstream based on CS lookup, PIT creation, Longest Name Prefix Match (LNPM) lookup, and (if present) the path steering value. If no valid next hop is found, an InterestReturn indicating "No Route" in the case of CCNx or a network NACK in the case of NDN is sent downstream.

If HopLimit equals 0, the forwarder generates a traceroute reply. This reply includes the forwarder's administrative name and signature, and a Path Label TLV. This TLV initially has a null value, since the traceroute reply originator does not forward the request and thus does not make a path choice. The reply will also include the corresponding PT_TR_REPLY Code TLV.

A traceroute reply will be the final reply of a traceroute session if any of the following conditions are met:

- If a forwarder has been given one or more administrative names, the target name matches one of them.
- The target name exactly matches the name of a Content Object residing in the forwarder's CS (unless the traceroute client application has chosen not to receive replies due to CS hits as specified in [Appendix A](#)).
- The target name matches (in an LNPM manner) a FIB entry with an outgoing face referring to a local application.

The PT_TR_REPLY Code TLV value of the reply is set to indicate the specific condition that was met. If none of those conditions were met, the PT_TR_REPLY Code is set to 4 to indicate that the HopLimit reached 0.

A received traceroute reply will be matched to an existing PIT entry as usual. On the reverse path, the Path Steering TLV of a reply will be updated by each forwarder to encode its choice of next hop(s). When included in subsequent requests, this Path Steering TLV allows the forwarders to steer the requests along the same path.

7. Protocol Operation for Locally Scoped Namespaces

In this section, we elaborate on two alternative design approaches in cases where the traceroute target prefix corresponds to a locally scoped namespace not directly routable from the client's local network.

The first approach leverages the NDN Link Object [[SNAMP](#)]. Specifically, the traceroute client attaches to the expressed request a Link Object that contains a number of routable name prefixes, based on which the request can be forwarded across the Internet until it reaches a network region where the request name itself is routable. A Link Object is created and signed by a data producer allowed to publish data under a locally scoped namespace. The way that a client retrieves a Link Object depends on various network design factors and is out of scope for this document.

At the time of this writing, and based on the current deployment of the Link Object by the NDN team [[NDNLPv2](#)], a forwarder at the border of the region where an Interest name becomes routable has to remove the Link Object from the incoming Interests. The Interest state maintained along the entire forwarding path is based on the Interest name regardless of whether it was forwarded based on this name or a prefix in the Link Object.

The second approach is based on prepending a routable prefix to the locally scoped name. The resulting prefix will be the name of the traceroute requests expressed by the client. In this way, a request will be forwarded based on the routable part of its name. When it reaches the network region where the original locally scoped name is routable, the border forwarder rewrites the request name and deletes its routable part. A forwarder will perform this rewriting operation on a request if the following two conditions are met:

- 1) the routable part of the request name matches a routable name of the network region adjacent to the forwarder (assuming that a forwarder is aware of those names), and
- 2) the remaining part of the request name is routable across the network region of this forwarder.

The state along the path depends on whether the request is traversing the portion of the network where the locally scoped name is routable. In this case, the forwarding can be based entirely on the locally scoped name. However, where a portion of the path lies outside the region where the locally scoped name is routable, the border router has to rewrite the name of a reply and prepend the routable prefix of the corresponding request to ensure that the generated replies will reach the client.

8. Security Considerations

A reflection attack could occur in the case of a traceroute reply with the CCNx packet format if a compromised forwarder includes in the reply the name of a victim forwarder. This could redirect the future administrative traffic towards the victim. To foil such reflection attacks, the forwarder that generates a traceroute reply **MUST** sign the name included in the payload. In this way, the client is able to verify that the included name is legitimate and refers to the forwarder that generated the reply. Alternatively, the forwarder could include in the reply payload their routable prefix(es) encoded as a signed NDN Link Object [[SNAMP](#)].

This approach does not protect against on-path attacks where a compromised forwarder that receives a traceroute reply replaces the forwarder's name and the signature in the message with its own name and signature to make the client believe that the reply was generated by the compromised forwarder. To foil such attack scenarios, a forwarder can sign the reply message itself. In such cases, the forwarder does not have to sign its own name in the reply message, since the message signature protects the message as a whole and will be invalidated in the case of an on-path attack. Additionally, a forwarder could swap out the name of a traceroute request with a name of its choosing. In this case, however, the response with the spoofed name will not be received by a client, since the change of name would invalidate the state in the PIT on the path back to the client.

Signing each traceroute reply message can be expensive and can potentially lead to computation attacks against forwarders. To mitigate such attack scenarios, the processing of traceroute requests and the generation of the replies **SHOULD** be handled by a separate management application running locally on each forwarder. The serving of traceroute replies is thereby separated from load on the forwarder itself. The approaches used by ICN applications to manage load may also apply to the forwarder's management application.

Interest flooding attack amplification is possible in the case of the second approach for dealing with locally scoped namespaces as described in [Section 7](#). A border forwarder will have to maintain extra state to prepend the correct routable prefix to the name of an outgoing reply, since the forwarder might be attached to multiple network regions (reachable under different prefixes) or a network region attached to this forwarder might be reachable under multiple routable prefixes.

We also note that traceroute requests have the same privacy characteristics as regular Interests.

9. IANA Considerations

IANA has assigned 0x07 to "PT_TR_REQUEST" and 0x08 to "PT_TR_REPLY" in the "CCNx Packet Types" registry established by [\[RFC8609\]](#).

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>>.
- [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>>.
- [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>>.
- [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>>.

10.2. Informative References

- [NDNLPv2] NDN team, "NDNLPv2: Named Data Networking Link Adaptation Protocol v2", February 2023, <<https://redmine.named-data.net/projects/nfd/wiki/NDNLPv2>>.

[NDNTLV] NDN project team, "NDN Packet Format Specification", February 2024, <<https://named-data.net/doc/NDN-packet-spec/current/>>.

[PATHSTEERING] Moiseenko, I. and D. Oran, "Path switching in content centric and named data networks", ICN '17: Proceedings of the 4th ACM Conference on Information-Centric Networking, pp. 66-76, DOI 10.1145/3125719.3125721, September 2017, <<https://dl.acm.org/doi/10.1145/3125719.3125721>>.

[REALTIME] Mastorakis, S., Gusev, P., Afanasyev, A., and L. Zhang, "Real-Time Data Retrieval in Named Data Networking", 2018 1st IEEE International Conference on Hot Information-Centric Networking (HotICN), Shenzhen, China, pp. 61-66, DOI 10.1109/HOTICN.2018.8605992, August 2018, <<https://ieeexplore.ieee.org/document/8605992>>.

[RFC4648] Josefsson, S., "The Base16, Base32, and Base64 Data Encodings", RFC 4648, DOI 10.17487/RFC4648, October 2006, <<https://www.rfc-editor.org/info/rfc4648>>.

[RFC9344] Asaeda, H., Ooka, A., and X. Shao, "CCNinfo: Discovering Content and Network Information in Content-Centric Networks", RFC 9344, DOI 10.17487/RFC9344, February 2023, <<https://www.rfc-editor.org/info/rfc9344>>.

[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, March 2024, <<https://www.rfc-editor.org/info/rfc9508>>.

[RFC9531] Moiseenko, I. and D. Oran, "Path Steering in Content-Centric Networking (CCNx) and Named Data Networking (NDN)", RFC 9531, DOI 10.17487/RFC9531, March 2024, <<https://www.rfc-editor.org/info/rfc9531>>.

[SNAMP] Afanasyev, A., Yi, C., Wang, L., Zhang, B., and L. Zhang, "SNAMP: Secure namespace mapping to scale NDN forwarding", 2015 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), Hong Kong, China, pp. 281-286, DOI 10.1109/INFOCOMW.2015.7179398, April 2015, <<https://ieeexplore.ieee.org/abstract/document/7179398>>.

Appendix A. Traceroute Client Application (Consumer) Operation

This section is an informative appendix regarding the proposed traceroute client operation.

The client application is responsible for generating traceroute requests for prefixes provided by users.

The overall process can be iterative: the first traceroute request of each session will have a HopLimit of 1 to reach the first hop forwarder, the second request will have a HopLimit of 2 to reach the second hop forwarder, and so on.

When generating a series of requests for a specific name, the first request will typically not include a Path Label TLV, since no TLV value is known. After a traceroute reply containing a Path Label TLV is received, each subsequent request might include the received path steering value in the Path Label header TLV to drive the requests towards a common path as part of checking network performance. To discover more paths, a client can omit the Path Label TLV in future requests. Moreover, for each new traceroute request, the client has to generate a new nonce and record the time that the request was expressed. The client also sets the lifetime of the traceroute request, which carries the same semantics as the Interest Lifetime [RFC8609] in an Interest.

Moreover, the client application might not wish to receive replies due to CS hits. In CCNx, a mechanism to achieve that would be to use a Content Object Hash Restriction TLV with a value of 0 in the payload of a traceroute request message. In NDN, the exclude filter selector can be used.

When it receives a traceroute reply, the client would typically match the reply to a sent request and compute the round-trip time of the request. It should parse the Path Label value and decode the reply's payload to parse the sender's name and signature. The client should verify that both the received message and the forwarder's name have been signed by the key of the forwarder, whose name is included in the payload of the reply (by fetching this forwarder's public key and verifying the contained signature). In the case that the client receives a PT_TR_REPLY Code TLV with a valid value, it can stop sending requests with increasing HopLimit values and potentially start a new traceroute session.

In the case that a traceroute reply is not received for a request within a certain time interval (lifetime of the request), the client should time out and send a new request with a new nonce value up to a maximum number of requests to be sent specified by the user.

Authors' Addresses

Spyridon Mastorakis

University of Notre Dame
South Bend, IN
United States of America
Email: smastor2@nd.edu

Dave Oran

Network Systems Research and Design
Cambridge, MA
United States of America
Email: daveoran@orandom.net

Ilya Moiseenko

Apple Inc.
Cupertino, CA
United States of America
Email: liamo@mailbox.org

Jim Gibson

Unaffiliated

Belmont, MA

United States of America

Email: jcgibson61@gmail.com**Ralph Droms**

Unaffiliated

Hopkinton, MA

United States of America

Email: rdroms.ietf@gmail.com