Workgroup: ICNRG
Internet-Draft:
draft-irtf-icnrg-icntraceroute-00
Published: March 2020
Intended Status: Experimental
Expires: 20 September 2020
Authors: S. Mastorakis
University of Nebraska, Omaha
J. Gibson
Cisco Systems
I. Moiseenko
Apple Inc
R. Droms
Google Inc.
D. Oran
Network Systems Research and Design

ICN Traceroute Protocol Specification

Abstract

This document presents the design of an ICN Traceroute protocol. This includes the operations both on the client and the forwarder side.

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 2 September 2020.

Copyright Notice

Copyright (c) 2020 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. Code Components extracted from this document must include Simplified BSD License text as described in

Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction

1.1. Requirements Language

2. Background on IP-Based Traceroute Operation

3. Traceroute Functionality Challenges and Opportunities in ICN

4. ICN Traceroute CCNx Packet Format

4.1. ICN Traceroute Request CCNx Packet Format

4.2. Traceroute Reply CCNx Packet Format

5. ICN Traceroute NDN Packet Format

5.1. ICN Traceroute Request NDN Packet Format

5.2. Traceroute Reply NDN Packet Format

6. Forwarder Handling

7. Protocol Operation For Locally-Scoped Namespaces

8. Security Considerations

9. References

9.1. Normative References

9.2. Informative References

Appendix A. Traceroute Client Application (Consumer) Operation

Authors' Addresses

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 ICN, routing and forwarding are based on name prefixes. To this end, the problem of ascertaining the characteristics (i.e., transit forwarders and delays) of at least one of the available routes to a name prefix is a fundamendal requirement for instumentation and network management.

This document describes protocol mechanisms for a traceroute equivalent in ICN networks based on CCN or NDN. This document contains a non-normative appendix section suggesting useful properties for an ICN traceroute client application that originates traceroute requests and processes traceroute replies.

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 [RFC2119].

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 or TCP SYN packets) with a TTL value increased by 1, essentially, performing a 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 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 NDN and CCN protocols, the communication paradigm is based exclusively on named objects. An Interest is forwarded across the network based on its name. Eventually, it retrieves a content object either from a producer application or some forwarder's Content Store (CS).

An ICN network differs from an IP network in at least 4 important ways:

* IP identifies interfaces to an IP network with a fixed-length number, 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 components.

* 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 end-point. 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 that 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 an Interest-Data exchange might significantly vary (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 in 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 the following:

* Trace one or more paths towards an ICN forwarder (for troubleshooting purposes).

* Trace one or more paths along which an application can be reached in the sense that Interest packets can be forwarded toward it.

- *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, which handles Operations, Administration and Management (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 preserving forwarding behavior as similar as possible to that for an Interest packet. In the same way, the encoding of a traceroute reply should allow for processing as similar as possible to that of 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. The mechanism described in [PATHSTEERING] can be used for this purpose.

It is also important, in the case of traceroute requests for the same prefix from different sources, 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, and hence avoid PIT aggregation and further enabling the exact matching of a response with a particular traceroute packet.

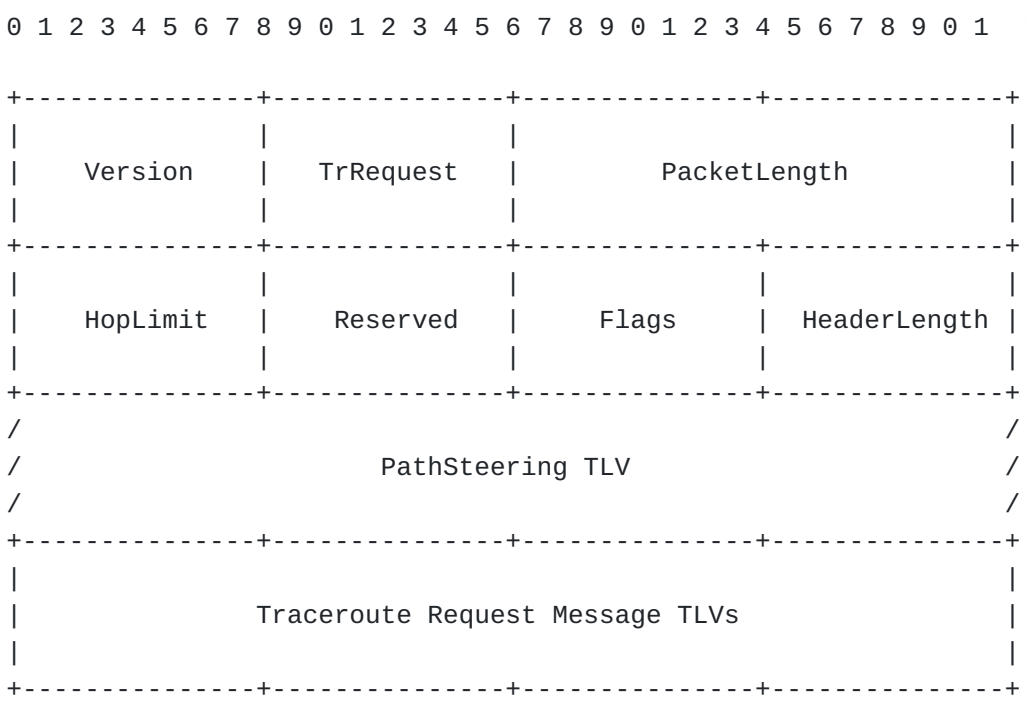
The packet types and format are presented in Section 4. The procedures, e.g. the procedures for determining and indicating that a destination has been reached, are specified in Section 6.

4. ICN Traceroute CCNx Packet Format

In this section, we present the CCNx packet format [CCNMessages] 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 packet type.

4.1. ICN Traceroute Request CCNx Packet Format

The format of the traceroute request packet is presented below:

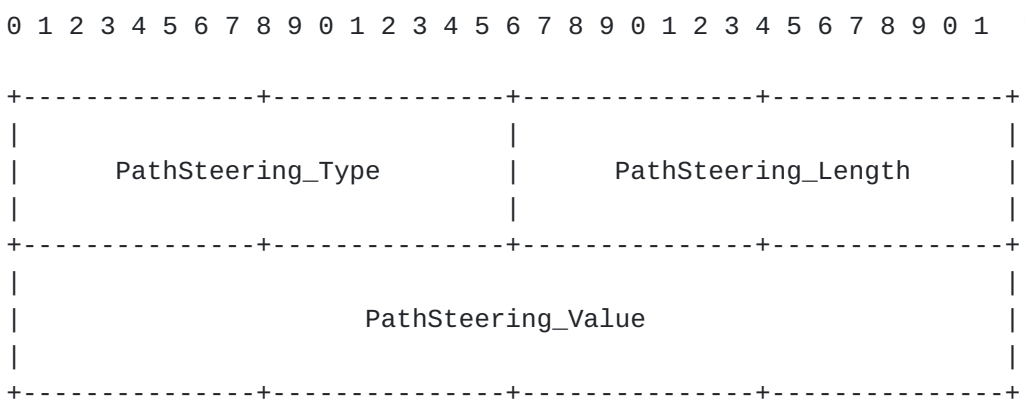


Traceroute Request CCNx Packet Format

The existing packet header fields have similar functionality to the header fields of a CCNx Interest packet. The value of the packet type field is TrRequest. The exact numeric value of this field type is to be assigned in the Packet Type IANA Registry for CCNx (see section 4.1 of [CCNMessages].

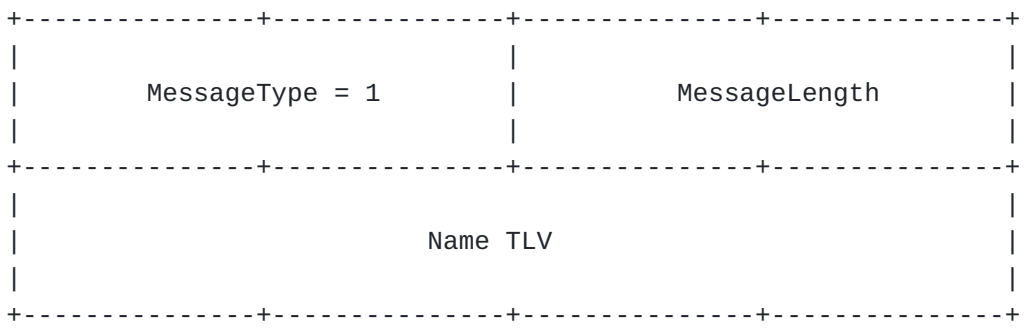
Compared to the typical format of a CCNx packet header [CCNMessages], there is a new optional fixed header TLV added to the packet header:

- 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 or different forwarding paths. An example of such a scheme is presented in [PATHSTEERING].



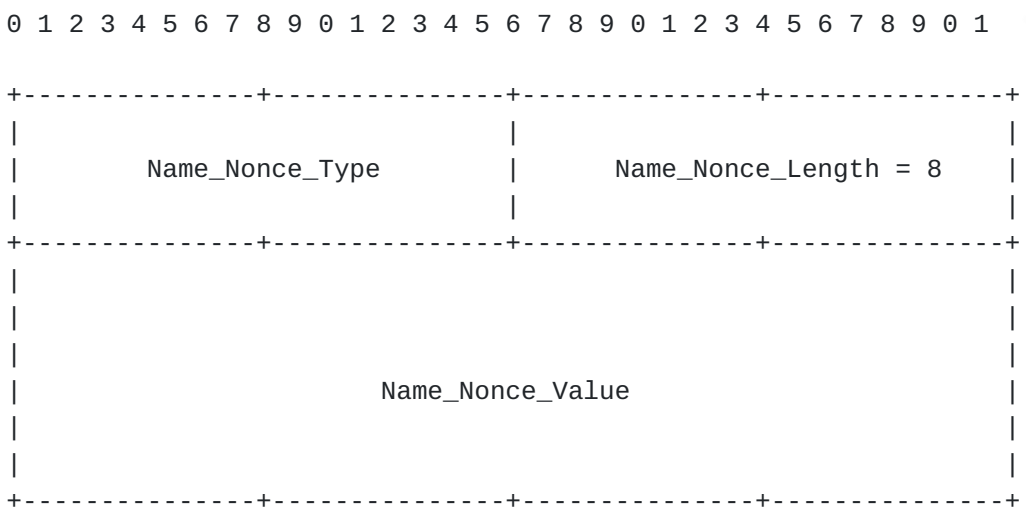
Path Steering TLV

The message of a traceroute request is presented below:



Traceroute Request Message Format

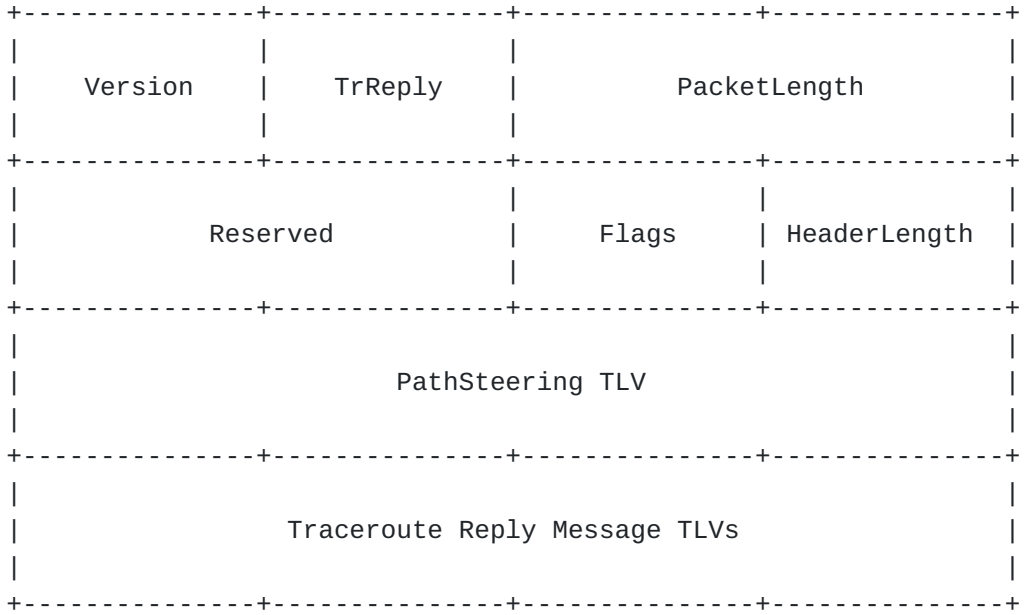
The traceroute request message is of type Interest in order to leverage the Interest forwarding behavior provided by the network. The Name TLV has the structure described in [CCNMessages]. The name consists of the target (destination) prefix appended with a nonce typed name component as its last component (to avoid Interest aggregation and allow exact matching of requests with responses). The value of this TLV is a 64-bit nonce.



Name Nonce Typed Component TLV

4.2. Traceroute Reply CCNx Packet Format

The format of a traceroute reply packet is presented below:

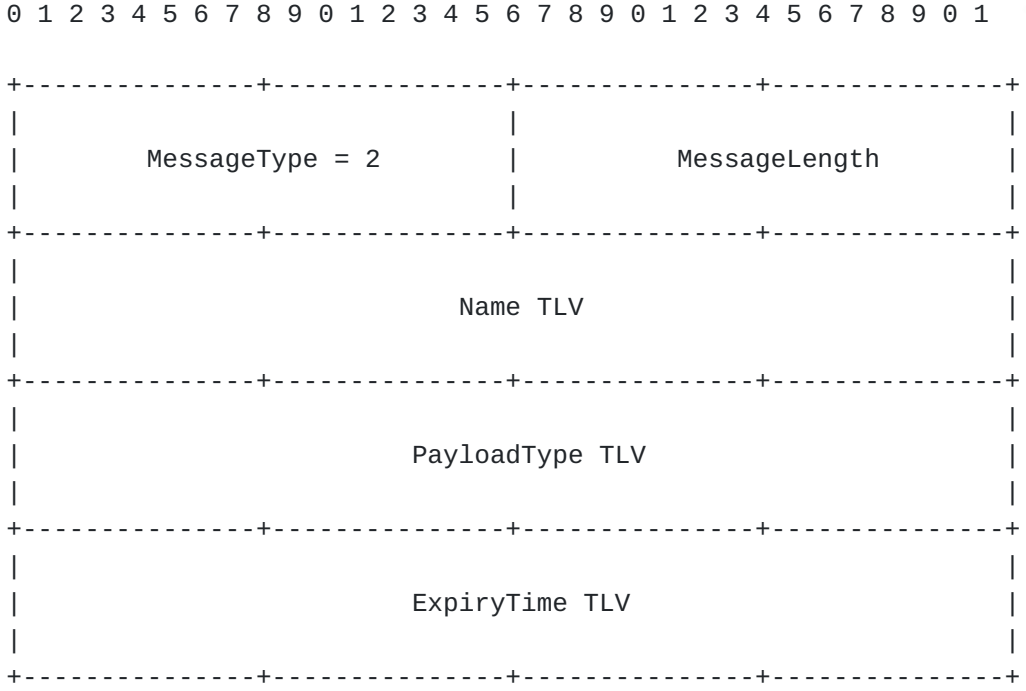


Traceroute Reply CCNx Packet Format

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

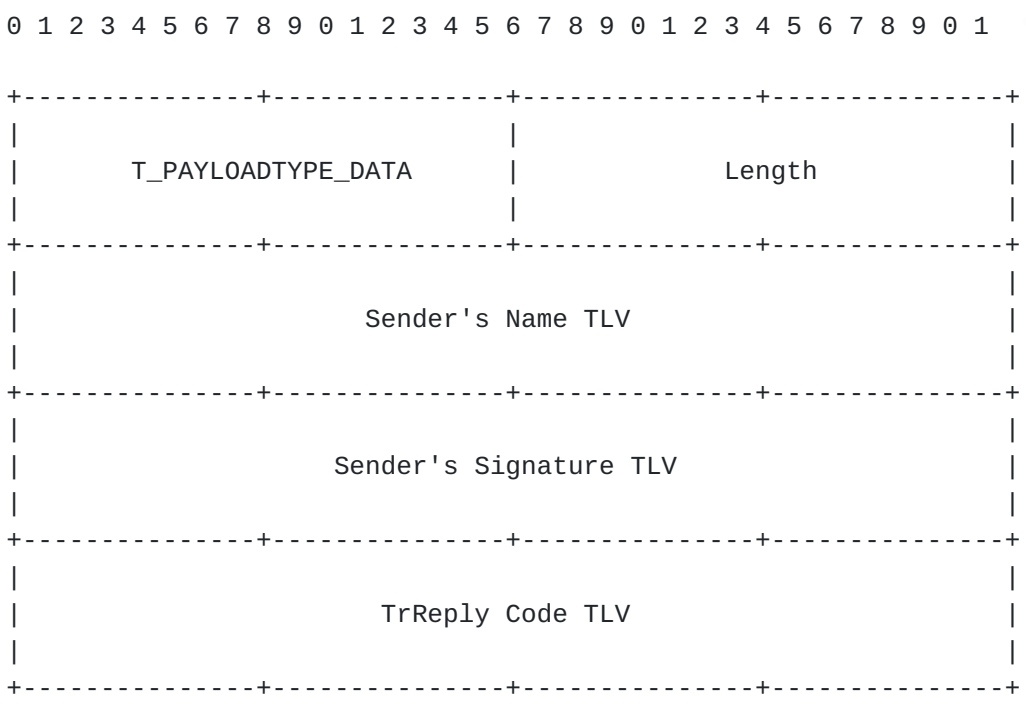
the packet type field is TrReply. The exact numeric value of this field type is to be assigned in the Packet Type IANA Registry for CCNx (see section 4.1 of [CCNMessages].

A traceroute reply message is of type Content Object, 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.



Traceroute Reply Message Format

The PayloadType TLV is presented below. It is of type T_PAYLOADTYPE_DATA, and the data schema consists of 2 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), 3) a TLV with return codes to indicate whether the request was satisfied due to the existence of a local application, a CS hit or a match with a forwarder's name, or the HopLimit value of the corresponding request reached 0.



Traceroute Reply Message 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 [CCNInfo] after a successful verification of the sender's name.

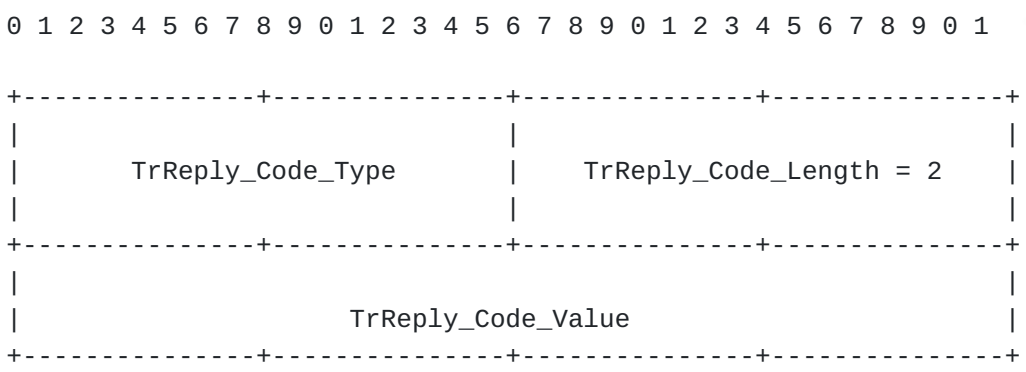
The structure of the TrReply Code TLV is presented below (16-bit value). The potential values are the following:

*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 the Hop limit reached the 0 value.



TrReply Code TLV

5. ICN Traceroute NDN Packet Format

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

5.1. ICN Traceroute Request NDN Packet Format

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

```
TracerouteRequest ::= INTEREST-TYPE TLV-LENGTH
                      Name
                      MustBeFresh
                      Nonce
                      HopLimit
                      Parameters?
```

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.

The "Paremeters" field of the Request contains the following PathSteering TLV:

```
PathSteering TLV ::= PATHSTEERING-TLV-TYPE TLV-LENGTH BYTE{8}
```

PathSteering TLV

Since the NDN packet format does provide a mechanism to prevent the network from caching specific data packets, we will use the MustBeFresh selector for requests (in combination with a Freshness Period TLV of value 0 for replies) to avoid fetching cached

traceroute replies with a freshness period that has expired [REALTIME].

5.2. Traceroute Reply NDN Packet Format

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

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

Traceroute Reply NDN Packet Format

Compared to the format of a regular NDN Data packet, a traceroute reply contains a PathSteering TLV field, which is not included in the security envelope, 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 the following:

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

MetaInfo TLV

The value of the ContentType TLV is 0. The same applies to the value of the FreshnessPeriod TLV, so that the replies are treated as stale data as soon as they are received by a forwarder.

The content of a traceroute reply consists of the following 2 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 the following (with the values specified in Section 4.2):

```
TrReplyCode ::= TRREPLYCODE-TLV-TYPE TLV-LENGTH BYTE{2}
```

Traceroute Reply Code TLV

6. Forwarder Handling

When a forwarder receives a traceroute request, the hop limit value is checked and decremented and the target name (i.e, the name of the traceroute request without the last nonce name component and the suffix "traceroute" in the case of a request with the NDN packet format) is extracted.

If the HopLimit has not expired (its value is greater than 0), the forwarder will forward the request upstream based on CS lookup, PIT creation, LPM lookup and the path steering value, if present. 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 the HopLimit value is equal to zero, the forwarder generates a traceroute reply. This reply includes the forwarder's administrative name and signature, and a PathSteering 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 TrReply 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 a Longest Prefix Match manner) a FIB entry with an outgoing face referring to a local application.

The TrReply Code TLV value of the reply is set to indicate the specific condition that was met. If none of those conditions was met, the TrReply Code is set to 4 to indicate that the hop limit value 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 2 alternative design approaches in cases that 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 the scope of the current draft.

Based on the current deployment of the LINK Object by the NDN team, 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. There are two conditions for a forwarder to perform this rewriting operation on a request: 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 maintained along the path, where the locally-scoped name is not routable, is based on the routable prefix along with the locally-scoped prefix, while within the network region that the locally-scoped prefix is routable is based only on it. To ensure that the generated replies will reach the client, the border forwarder has also to rewrite the name of a reply and prepend the routable prefix of the corresponding request.

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 reply message, since the message signature protects the message as a whole and will be invalidated in the case of an on-path attack.

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 can be handled by a separate management application running locally on each forwarder. Serving traceroute replies therefore 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 to deal with locally-scoped namespaces 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. References

9.1. Normative References

[CCNMessages] Mosko, M., Solis, I., and C. Wood, "CCNx Messages in TLV Format.", 2018, <https://tools.ietf.org/html/draft-irtf-icnrg-ccnxmessages-08>.

[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>.

9.2. Informative References

[CCNInfo] Asaeda, H. and X. Shao, "CCNinfo: Discovering Content and Network Information in Content-Centric Networks.", 2018, <https://tools.ietf.org/html/draft-asaeda-icnrg-ccninfo-01/>.

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

[PATHSTEERING] Moiseenko, I. and D. Oran, "Path switching in content centric and named data networks in Proceedings of the 4th ACM Conference on Information-Centric Networking", 2017.

[REALTIME] Mastorakis, S., Gusev, P., Afanasyev, A., and L. Zhang, "Real-Time Data Retrieval in Named Data Networking in Proceedings of the 1st IEEE International Conference on Hot Topics in Information-Centric Networking", 2017.

[SNAMP] Afanasyev, A. and , "SNAMP: Secure namespace mapping to scale NDN forwarding, IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)", 2015.

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 value 1 to reach the first hop forwarder, the second of value 2 to reach the second hop forwarder and so on and so forth.

When generating a series of requests for a specific name, the first one will typically not include a PathSteering TLV, since no TLV value is known. After a traceroute reply containing a PathSteering TLV is received, each subsequent request might include the received path steering value in the PathSteering header TLV to drive the requests towards a common path as part of checking the network performance. To discover more paths, a client can omit the PathSteering 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. It will also set the lifetime of a request, which will have semantics similar to the lifetime of 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 PathSteering value and decode the reply's payload to parse the 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 an TrReply 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 Nebraska, Omaha
Omaha, NE
United States of America

Email: smastorakis@unomaha.edu

Jim Gibson
Cisco Systems
Cambridge, MA
United States of America

Email: gibson@cisco.com

Ilya Moiseenko
Apple Inc
Cupertino, CA
United States of America

Email: iliamo@mailbox.org

Ralph Droms
Google Inc.
Cambridge, MA
United States of America

Email: rdroms.ietf@gmail.com

Dave Oran
Network Systems Research and Design
Cambridge, MA
United States of America

Email: daveoran@orandom.net