ICNRG
Internet-Draft
Intended status: Experimental
Expires: August 16, 2020

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

February 13, 2020

ICN Ping Protocol Specification
draft-mastorakis-icnrg-icnping-06

Abstract

This document presents the design of an ICN Ping protocol. It 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 August 16, 2020.

Table of Contents

1. Introduction 2
1.1. Requirements Language 2
2. Background on IP-Based Ping Operation 3
3. Ping Functionality Challenges and Opportunities in ICN . . . 3
4. ICN Ping Echo CCNx Packet Formats 5
4.1. ICN Ping Echo Request CCNx Packet Format 5
4.2. Ping Echo Reply CCNx Packet Format 8
5. ICN Ping Echo NDN Packet Formats 11
5.1. ICN Ping Echo Request NDN Packet Format 11
5.2. Ping Echo Reply NDN Packet Format 12
6. Forwarder Handling 13
7. Protocol Operation For Locally-Scoped Namespaces 14
8. Security Considerations 15
9. Acknowledgements 15
10. References 15
10.1. Normative References 15
10.2. Informative References 15
Appendix A. Ping Client Application (Consumer) Operation 16
Authors' Addresses 17

1. Introduction

Determining data plane reachability to a destination and taking coarse performance measurements of round trip time are fundamental facilities for network administration and troubleshooting. In IP, where routing and forwarding are based on IP addresses, ICMP echo and ICMP echo response are the protocol mechanisms used for this purpose, generally exercised through the familiar ping utility. In ICN, where routing and forwarding are based on name prefixes, the ability to determine reachability of names is required.

This document proposes protocol mechanisms for a ping equivalent in ICN networks. A non-normative appendix suggests useful properties for an ICN ping client application, analogous to IP ping, that originates echo requests and process echo 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 Ping Operation

In IP-based ping, an IP address is specified, either directly, or via translation of a domain name through DNS. The ping client application sends a number of ICMP Echo Request packets with the specified IP address as the IP destination address and an IP address from the client's host as the IP source address.

An ICMP Echo Request is forwarded across the network based on its destination IP address. If it eventually reaches the destination, the destination responds by sending back an ICMP Echo Reply packet to the IP source address from the ICMP Echo Request.

If an ICMP Echo Request does not reach the destination or the Echo reply is lost, the ping client times out. Any ICMP error messages, such as "no route to destination", generated by the ICMP Echo Request message are returned to the client and reported.

3. Ping Functionality Challenges and Opportunities in ICN

In ICN protocols (e.g., NDN and CCNx), 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).

IP-based ping was built as an add-on on top of an already existing network architecture. In ICN, we have the opportunity to incorporate diagnostic mechanisms directly in the network layer protocol, and hopefully provide more powerful diagnostic capability than can be realized through the layered ICMP Echo approach.

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. This can lead to a significant variance in round-trip times, which might not be desirable in the case of a network troubleshooting mechanism like ping.

- 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.

These differences introduce new challenges, new opportunities and new requirements in the design of an ICN ping protocol. Following this communication model, a ping client should be able to express ping echo requests with some name prefix and receive responses.

Our goals are the following:

- Test the reachability and the operation of an ICN forwarder.

- Test the reachability of an application (in the sense of whether Interests for a prefix that it serves can be forwarded to it) and discover the forwarder with local connectivity to (an instance of) the application.

- Test whether a specific named object is cached in some on-path CS, and, if so, return the administrative name of the corresponding forwarder.

- Perform some simple network performance measurements.

To this end, a ping name can represent:

- An administrative name that has been assigned to a forwarder.

- 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 ping echo request enable the forwarders to distinguish a ping from a normal Interest, while also allowing for forwarding behavior to be as similar as possible to that of an Interest packet. In the same way, the encoding of a ping echo reply should allow for forwarder processing as close as possible to that used for data packets.

The ping protocol should also enable relatively robust round-trip time measurements. To this end, it is important to have a mechanism to steer consecutive ping echo requests for the same name towards an individual path [PATHSTEERING].

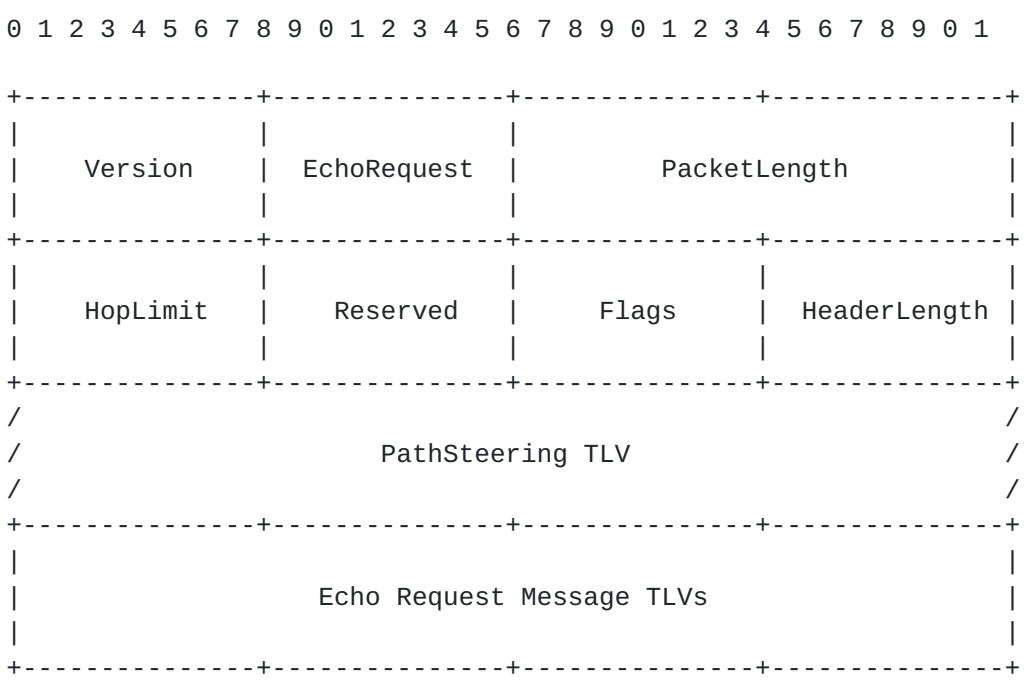
It is also important, in the case of ping echo requests for the same name from different sources, to have a mechanism to avoid aggregating those requests in the PIT. To this end, we need some encoding in the ping echo requests to make each request for a common name unique, hence avoiding PIT aggregation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

4. ICN Ping Echo CCNx Packet Formats

In this section, we describe the Echo Packet Format according to the CCNx packet format [CCNMessages], where messages exist within outermost containments (packets). Specifically, we propose two types of ping packets, an echo request and an echo reply packet type.

4.1. ICN Ping Echo Request CCNx Packet Format

The format of the ping echo request packet is presented below:

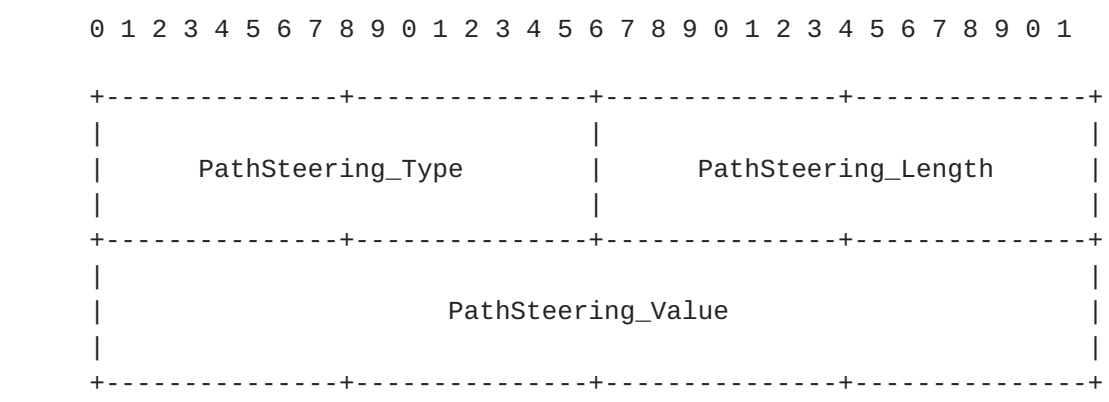


Echo Request CCNx Packet Format

The existing packet header fields have the same definition as the header fields of a CCNx Interest packet. The value of the packet type field is Echo Request. 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].

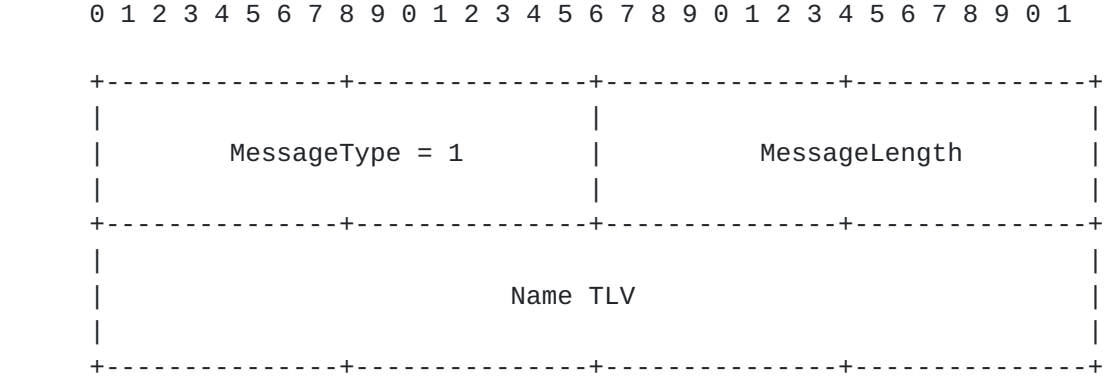
Compared to the typical format of a CCNx packet header [CCNMessages], there is a new optional fixed header TLV added to the packet header:

- A PathSteering hop-by-hop header TLV, which is constructed hop-by-hop in the echo reply and included in the echo request to steer consecutive echo requests expressed by a ping client towards a common forwarding path. An example of such a scheme is presented in [PATHSTEERING].



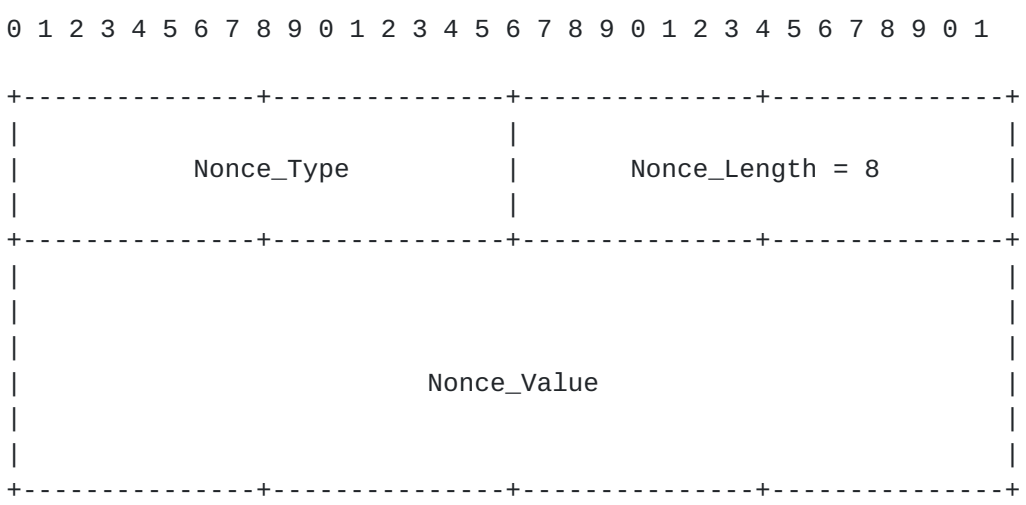
PathSteering TLV

The message of an echo request is presented below:



Echo Request Message Format

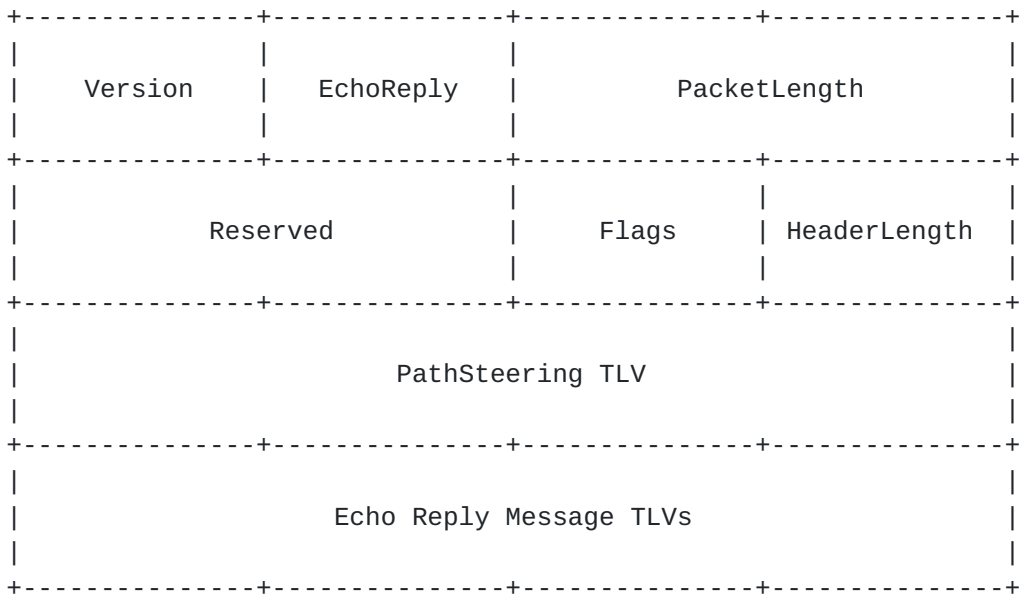
The echo request message is of type Interest in order to leverage the Interest forwarding behavior provided by the network. The Name TLV has the structure described in [CCNMessages]. The name consists of the prefix that we would like to ping appended with a nonce typed name component as its last component. The exact numeric value of this field type is to be assigned in the Name Component Type IANA Registry for CCNx (see section 4.5 of [CCNMessages]. The value of this TLV is a 64-bit nonce. 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.



Nonce Typed Name Component TLV

4.2. Ping Echo Reply CCNx Packet Format

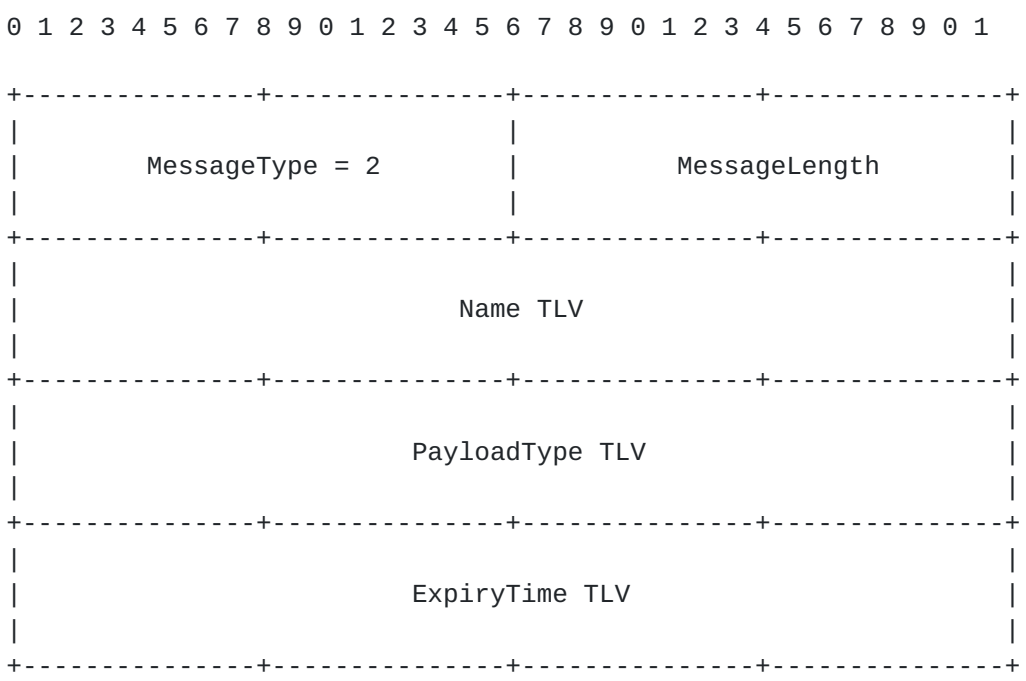
The format of a ping echo reply packet is presented below:



Echo Reply CCNx Packet Format

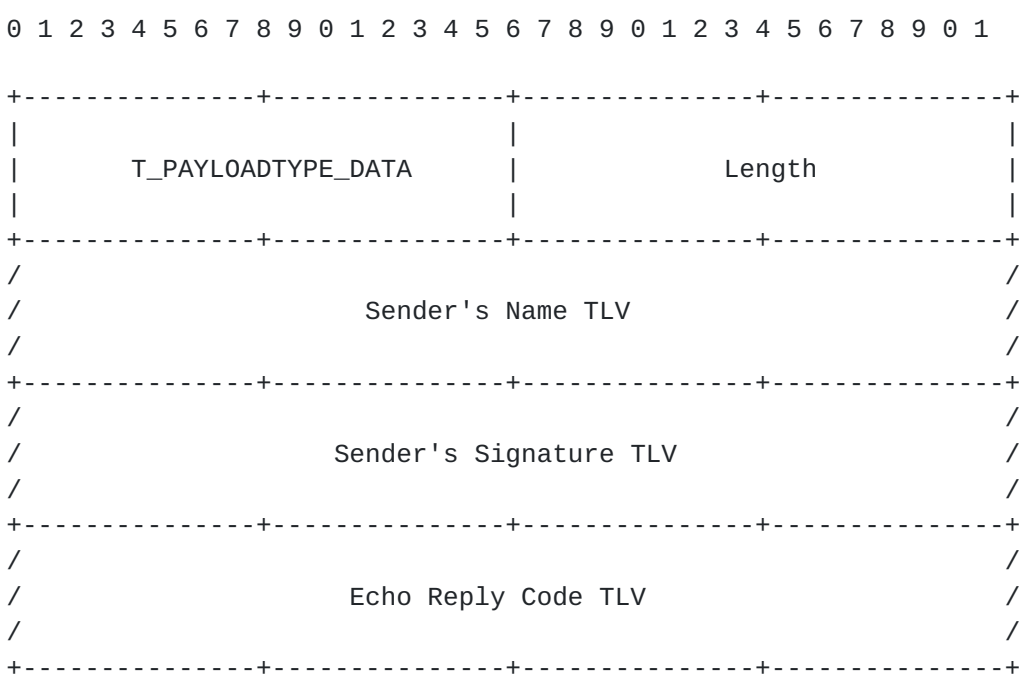
The header of an echo reply consists of the header fields of a CCNx Content Object and a hop-by-hop PathSteering TLV. The value of the packet type field is Echo Reply. The exact numeric value of this field type is to be assigned in the Packet Type IANA Registry for CCNx (see section 4.1 of [CCNMessages]). The PathSteering header TLV is as defined for the echo request packet.

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



Echo Reply Message Format

The PayloadType TLV is presented below. It is of type T_PAYLOADTYPE_DATA, and the data schema consists of 3 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 a return code to indicate what led to the generation of this reply (i.e., existence of a local application, a CS hit or a match with a forwarder's administrative name as specified in Section 6).

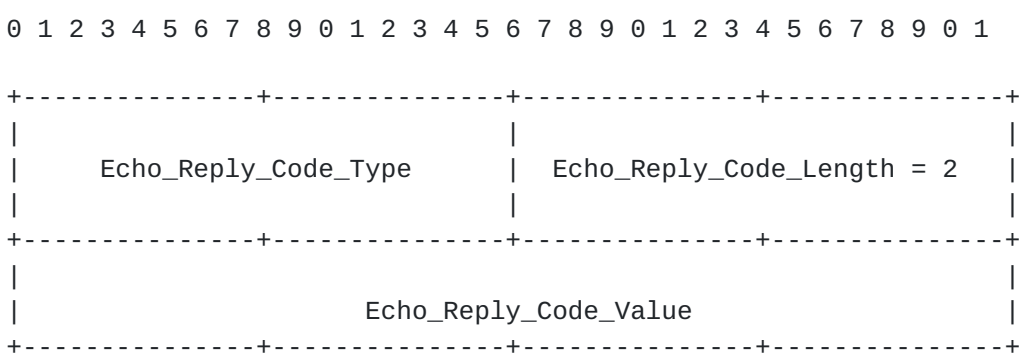


Echo Reply Message Format

The goal of including the name of the sender in the echo 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 Echo Reply Code TLV is presented below (16-bit value). The defined values are the following:

- 1: Indicates that the target name matched the administrative name of a forwarder.
- 2: Indicates that the target name matched a prefix served by an application.
- 3: Indicates that the target name matched the name of an object in a forwarder's CS.



Echo Reply Code TLV

5. ICN Ping Echo NDN Packet Formats

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

5.1. ICN Ping Echo Request NDN Packet Format

An echo request is encoded as an NDN Interest packet. Its format is the following:

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

Echo Request NDN Packet Format

The name of an echo request consists of the prefix to be pinged, a nonce value (it can be the value of the Nonce field) and the suffix "ping" to denote that this Interest is a ping 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 not provide a mechanism to prevent the network from caching specific data packets, we use the MustBeFresh selector for echo requests (in combination with a

Freshness Period TLV of value 0 for echo replies) to avoid fetching cached echo replies with an expired freshness period [REALTIME].

5.2. Ping Echo Reply NDN Packet Format

An echo reply is encoded as an NDN Data packet. Its format is the following:

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

Echo Reply NDN Packet Format

Compared to the format of a regular NDN Data packet, an echo 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 an echo reply is the name of the corresponding echo 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 an echo reply consists of the following 2 TLVs: Sender's name (with a structure similar as an NDN Name TLV) and Echo 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 Echo Reply Code TLV format is the following (with the values specified in Section 4.2):

```
EchoReplyCode ::= ECHOREPLYCODE-TLV-TYPE TLV-LENGTH BYTE{2}
```

Echo Reply Code TLV

6. Forwarder Handling

When a forwarder receives an echo request, it first extracts the message's base name (i.e., the request name with the Nonce name component excluded and the suffix "ping" in the case of an echo request with the NDN packet format).

In some cases, the forwarder originates an echo reply, sending the reply downstream through the face on which the echo request was received. This echo reply includes the forwarder's own name and signature and the appropriate echo reply code based on the condition that triggered the reply generation. It also includes a pathSteering TLV, initially containing a null value (since the echo reply originator did not forward the request and, thus, does not make a path choice).

The forwarder generates and returns an echo reply in the following cases:

- Assuming that a forwarder has been given one or more administrative names, the echo request base name exactly matches any of the forwarder's administrative name(s).

- The echo request's base name exactly matches the name of a content-object residing in the forwarder's CS (unless the ping client application has chosen not to receive replies due to CS hits as specified in Appendix A).

- The echo request base name matches (in a Longest Prefix Match manner) a FIB entry with an outgoing face referring to a local application.

If none of the conditions to reply to the echo request are met, the forwarder will attempt to forward the echo request upstream based on the path steering value (if present), the results of the FIB LPM lookup and PIT creation (based on the name including the nonce typed name component and the suffix "ping" in the case of an echo request with the NDN packet format). If no valid next-hop is found, an InterestReturn is sent downstream indicating "no route" (as with a failed attempt to forward an ordinary Interest).

A received echo reply will be matched to an existing PIT entry as usual. On the reverse path, the path steering TLV of an echo reply will be updated by each forwarder to encode its next-hop choice. When included in subsequent echo requests, this pathSteering TLV allows the forwarders to steer the echo 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 pinged 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 ping 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 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 usage of the LINK Object by the NDN team, a forwarder at the border of the region where an Interest name becomes routable must remove the LINK Object from 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 its name or a routable 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 echo 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. Within the network region that the locally-scoped prefix is routable, the state is based only on it. To ensure that the generated replies reach the ping client, the border forwarder has also to rewrite the name of a reply and prepend the routable prefix of the corresponding echo request.

8. Security Considerations

A reflection attack could in the case of an echo 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 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].

Interest flooding attack amplification is possible in the case of the second approach to deal with locally-scoped namespaces described in Section 7. To eliminate such amplification, a border forwarder will have to maintain extra state in order 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.

9. Acknowledgements

The authors would like to thank Mark Stapp for the fruitful discussion on the objectives of the ICN ping protocol.

10. References

10.1. 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>.

10.2. 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, in Proceedings of the IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)", 2015.

Appendix A. Ping Client Application (Consumer) Operation

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

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

When generating a series of echo requests for a specific name, the first echo request will typically not include a PathSteering TLV, since no TLV value is known. After an echo reply containing a PathSteering TLV is received, each subsequent echo request can include the received path steering value in the PathSteering 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 steering TLV in future requests. Moreover, for each new ping echo 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 an echo request, which will have identical semantics to the lifetime of an Interest.

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

When it receives an echo 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). The client can also decode the Echo Reply Code TLV to understand the condition that triggered the generation of the reply.

In the case that an echo 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 some maximum number of requests to be sent specified by the user.

Authors' Addresses

Spyridon Mastorakis
University of Nebraska, Omaha
Omaha, NE
US

Email: smastorakis@unomaha.edu

Jim Gibson
Cisco Systems
Cambridge, MA
US

Email: gibson@cisco.com

Ilya Moiseenko
Apple Inc
Cupertino, CA
US

Email: iliamo@mailbox.org

Ralph Droms
Google Inc.
Cambridge, MA
US

Email: rdroms.ietf@gmail.com

Dave Oran
Network Systems Research and Design
Cambridge, MA
US

Email: daveoran@orandom.net