

Workgroup: ICNRG  
Internet-Draft: draft-irtf-icnrg-icnping-12  
Published: 28 August 2023  
Intended Status: Experimental  
Expires: 29 February 2024  
Authors: S. Mastorakis  
University of Notre Dame  
D. Oran J. Gibson  
Network Systems Research and Design Unaffiliated  
I. Moiseenko R. Droms  
Apple Inc Unaffiliated

## ICN Ping Protocol Specification

### Abstract

This document presents the design of an ICN Ping protocol. It includes the operations 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 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 29 February 2024.

### Copyright Notice

Copyright (c) 2023 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](#)
  - [1.1. Requirements Language](#)
  - [1.2. Terminology](#)
- [2. Background on IP-Based Ping Operation](#)
- [3. Ping Functionality Challenges and Opportunities in ICN](#)
- [4. ICN Ping Echo CCNx Packet Formats](#)
  - [4.1. ICN Ping Echo Request CCNx Packet Format](#)
  - [4.2. Ping Echo Reply CCNx Packet Format](#)
- [5. ICN Ping Echo NDN Packet Formats](#)
  - [5.1. ICN Ping Echo Request NDN Packet Format](#)
  - [5.2. Ping Echo Reply NDN Packet Format](#)
- [6. Forwarder Handling](#)
- [7. Protocol Operation For Locally-Scoped Namespaces](#)
- [8. Security Considerations](#)
- [9. IANA Considerations](#)
- [10. Acknowledgements](#)
- [11. References](#)
  - [11.1. Normative References](#)
  - [11.2. Informative References](#)
- [Appendix A. Ping Client Application \(Consumer\) Operation](#)
- [Authors' Addresses](#)

## 1. Introduction

Ascertaining 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 ascertain reachability of names is required.

This document proposes protocol mechanisms for a ping equivalent in ICN ([CCNx](#) [[RFC8609](#)] and [NDN](#) [[NDNTLV](#)]) networks. A non-normative appendix suggests useful properties for an ICN ping client application, analogous to IP ping, that originates echo requests and processes echo replies.

In order to carry out meaningful experimentation and deployment of ICN protocols, tools to manage and debug the operation of ICN architectures and protocols are needed analogous to ping and traceroute used for TCP/IP. This document describes the design of a management and debugging protocol analogous to the ping protocol of

TCP/IP, which will aid the experimental deployment of ICN protocols. As the community continues its experimentation with ICN architectures and protocols, the design of ICN Ping might change accordingly. ICN Ping is designed as a "first line of defense" tool to troubleshoot ICN architectures and protocols. As such, this document is classified as an experimental RFC. Note that a measurement application is needed to make proper use of ICN Ping in order to compute various statistics, such as the variance, average, maximum and minimum RTT values as well as loss rates.

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.

### **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 BCP 14 [[RFC2119](#)] [[RFC8174](#)] when, and only when, they appear in all capitals, as shown here.

### **1.2. Terminology**

This specification uses the terminology defined in [[RFC8793](#)]. To aid the understanding of readers, we additionally define the following terms:

\*Producer's name: The name prefix that a request must carry in order to reach a producer over an ICN network.

\*Named Data: A synonym for a content object.

\*Round Trip Time (RTT): The time between sending a request for a specific piece of named data and receiving the corresponding piece of named data.

\*Sender: An entity that sends a request for named data or a piece of named data.

\*Name of a sender: An alias of producer's name.

\*Border forwarder: The forwarder that is the border of a network region where a producer's name is directly routable (i.e., the producer's name is present in the FIB of forwarders within this network region).

## **2. Background on IP-Based Ping Operation**

In IP-based ping, an IP address is specified by the user 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.

Each 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, the communication paradigm is based exclusively on named objects. An Interest is forwarded across the network based on the name prefix that it carries. Eventually, a content object is retrieved either from a producer application or some forwarder's Content Store (CS).

IP-based ping was built as an add-on measurement and debugging tool 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 address, and delivers IP packets to one or more of these interfaces. ICN identifies units of data in the network with a variable length name consisting of a hierarchical list of name 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 delivery 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 while resulting in a more robust overall forwarding architecture, has implications for 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 and only one of them forwarded onward. 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 operational state of an ICN forwarder.

\*Test the reachability of a producer or a data repository (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) this producer or repository.

\*Test whether a specific named object is cached in some on-path CS (e.g., a video segment with a name `"/video/_seq=1"`), and, if so, return the administrative name of the corresponding forwarder (e.g., a forwarder with an administrative name `"/ISP/forwarder1"`).

\*Perform some simple network performance measurements, such as RTT and loss rate.

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 valuable to have a mechanism to steer consecutive ping echo requests for the same name towards an individual path. Such a capability was initially published in [[PATHSTEERING](#)] and has been specified for CCNx and NDN in [[I-D.irtf-icnrg-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 those requests being aggregated 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. However, avoiding PIT aggregation could lead to PIT DoS attacks.

#### **4. ICN Ping Echo CCNx Packet Formats**

In this section, we describe the Echo Packet Format according to the CCNx packet format [[RFC8569](#)], where messages exist within outermost containments (packets). Specifically, we specify 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:

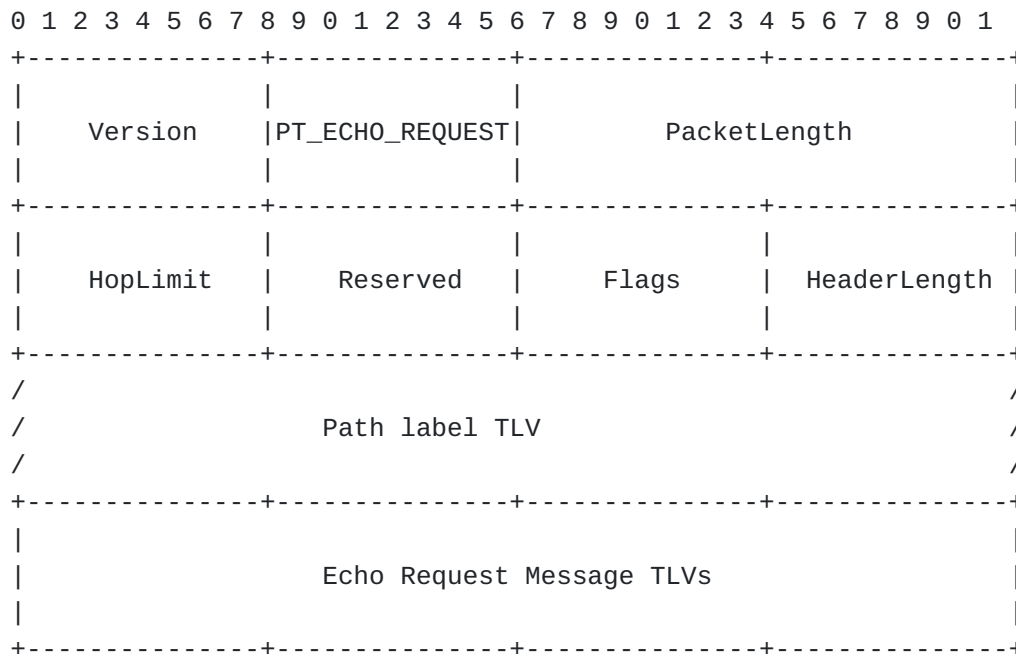


Figure 1: Echo Request CCNx Packet Format

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 *PT\_ECHO\_REQUEST*. See [Section 9](#) for the value assignments.

Compared to the typical format of a CCNx packet header from [\[RFC8569\]](#), in order to enable path steering of Echo Requests, there is an optional fixed header Path label TLV as specified in section 3.1 of [\[I-D.irtf-icnrg-pathsteering\]](#) added to the packet header:

The message format of an echo request is presented below:

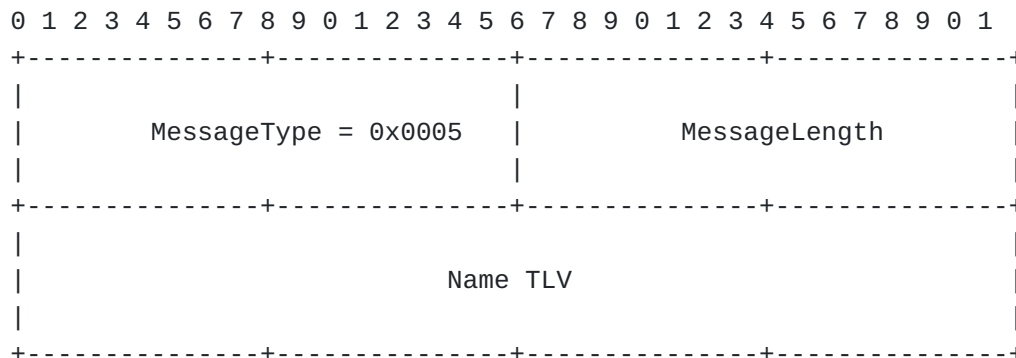


Figure 2: Echo Request Message Format

The echo request message is of type *T\_DISCOVERY*. The Name TLV has the structure described in [\[RFC8609\]](#). The name consists of the prefix that we would like to ping appended with a nonce typed name

segment (T\_NONCE) 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. See Section 9 for the value assigned to this name segment type. 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.

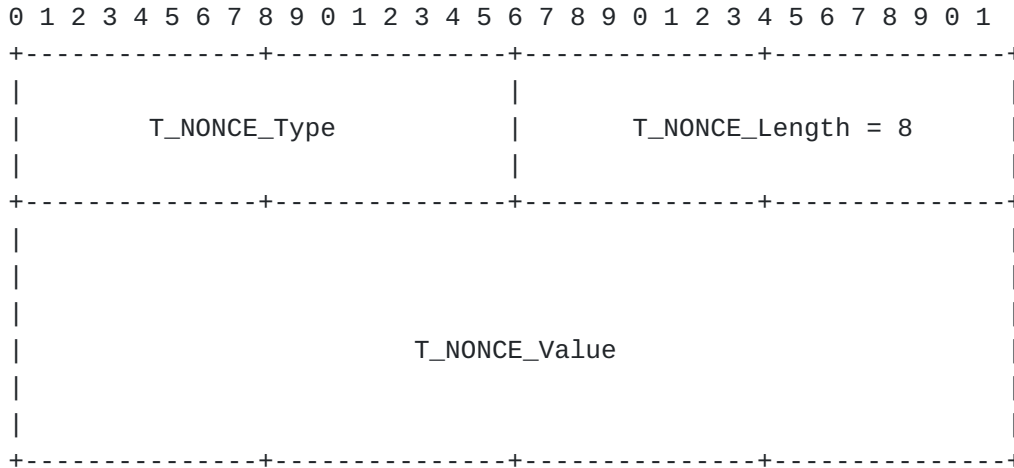


Figure 3: T\_NONCE Name Segment TLV for Echo Request messages

#### 4.2. Ping Echo Reply CCNx Packet Format

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

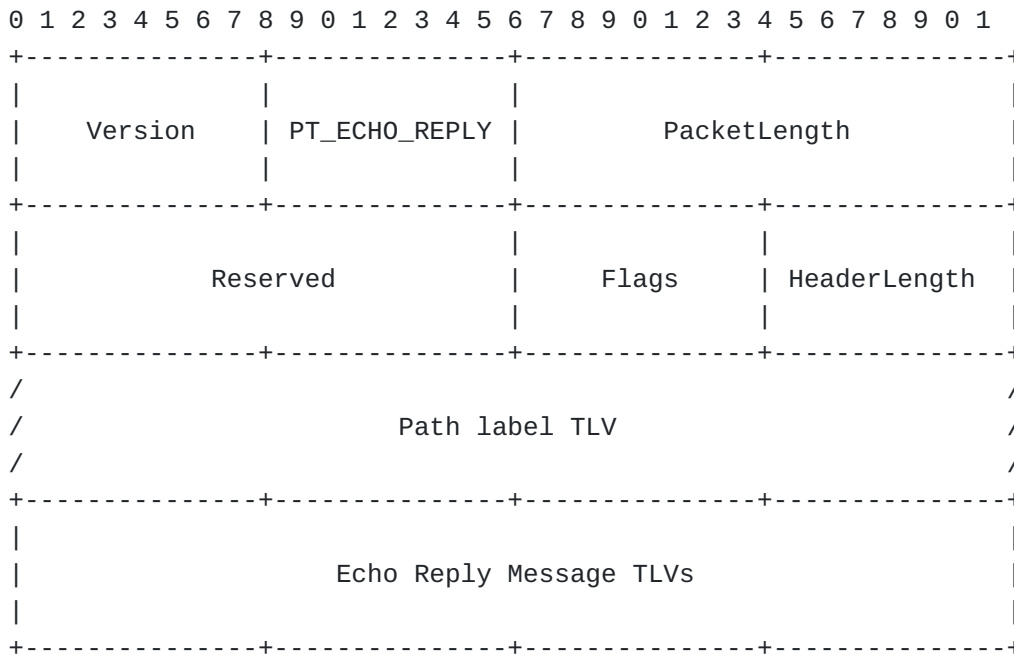


Figure 4: Echo Reply CCNx Packet Format



The header of an echo reply consists of the header fields of a CCNx Content Object and a hop-by-hop Path label TLV. The value of the packet type field is PT\_ECHO\_REPLY. See [Section 9](#) for the value assignments. The Path label header TLV from section 3.1 of [\[I-D.irtf-icnrg-pathsteering\]](#) is as defined for the echo request packet.

A ping echo reply message is of type T\_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.

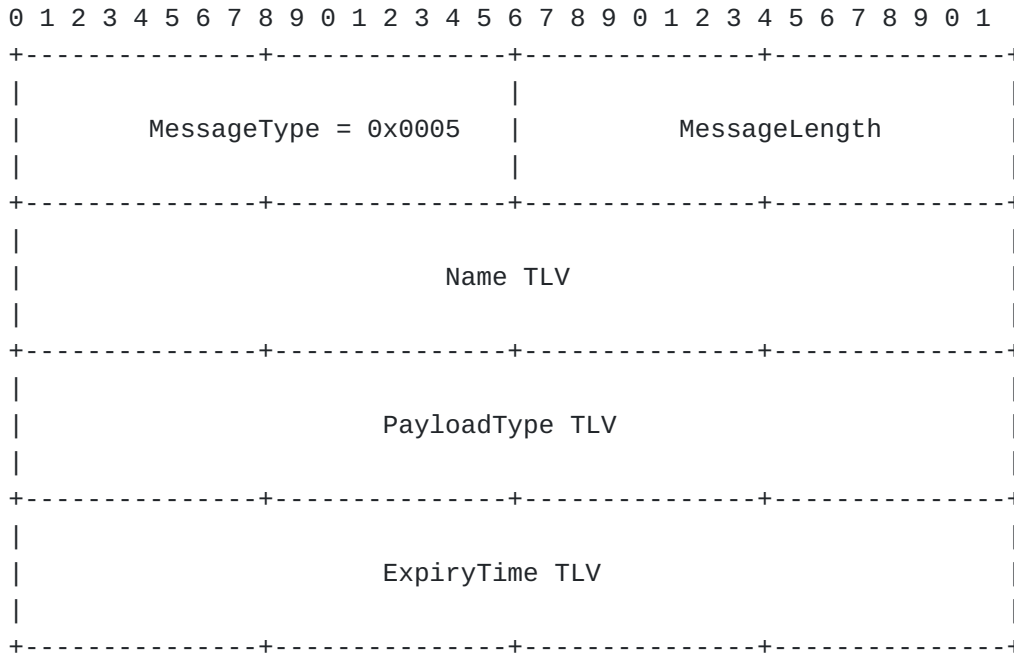


Figure 5: Echo Reply Message Format

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](#)).

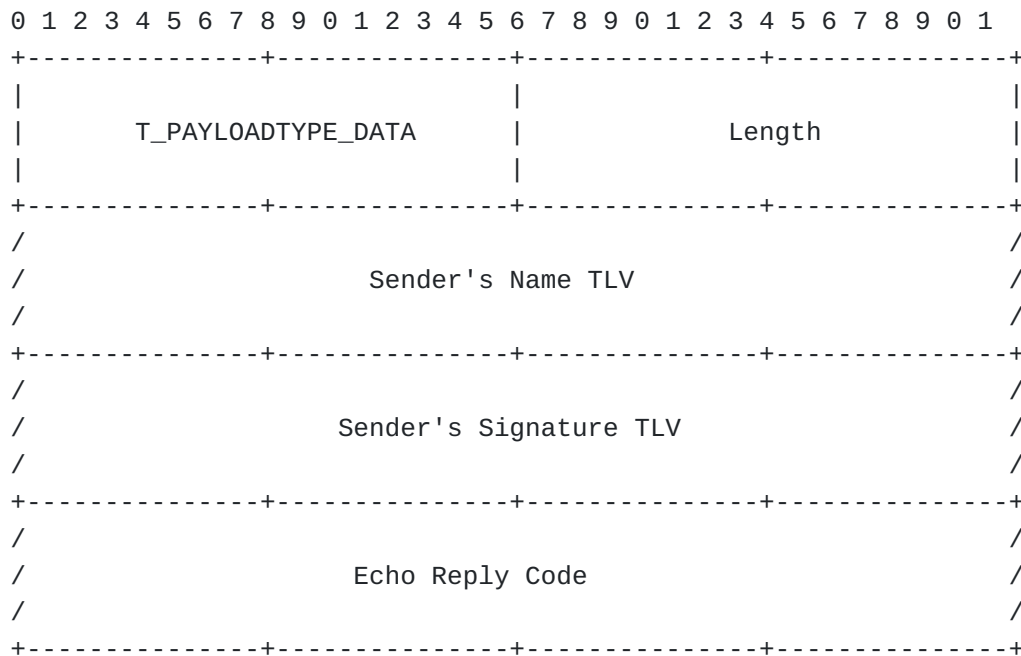


Figure 6: Echo Reply Message Format

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](#) [[RFC9344](#)] after a successful verification of the sender's name.

The types of the Echo Reply Code field are the following:

- \*T\_ECHO\_RETURN\_FORWARDER: Indicates that the target name matched the administrative name of a forwarder.
- \*T\_ECHO\_RETURN\_APPLICATION: Indicates that the target name matched a prefix served by an application.
- \*T\_ECHO\_RETURN\_OBJECT: Indicates that the target name matched the name of an object in a forwarder's CS.

## 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
                ApplicationParameters?

```

Figure 7: Echo Request NDN Packet Format

The name field of an echo request consists of the name 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 (added as a KeywordNameComponent). When the "ApplicationParameters" element is present, a parametersSha256DigestComponent is added as the last name segment.

An echo request MAY carry a Path label TLV in the [NDN Link Adaptation Protocol](#) [NDNLPv2] as specified in [I-D.irtf-icnrg-pathsteering].

Since the NDN packet format does not provide a mechanism to prevent the network from caching specific data packets, we use the MustBeFresh element for echo requests (in combination with a Freshness Period TLV of value 1 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
            Name
            MetaInfo
            Content
            Signature

```

Figure 8: Echo Reply NDN Packet Format

An echo reply MAY contain a Path label TLV in the [NDN Link Adaptation Protocol](#) [NDNLPv2] as specified in [I-D.irtf-icnrg-pathsteering], 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
```

Figure 9: 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 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 2*OCTET
```

Figure 10: Echo Reply Code TLV

## 6. Forwarder Handling

We present the workflow of the forwarder's operation in [Figure 11](#). When a forwarder receives an echo request, it first extracts the message's base name (i.e., the request name with the Nonce name segment excluded as well as the suffix "ping" and the ParametersSha256DigestComponent 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 Path label 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 segment 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 Path label TLV allows the forwarders to steer the echo requests along the same path.

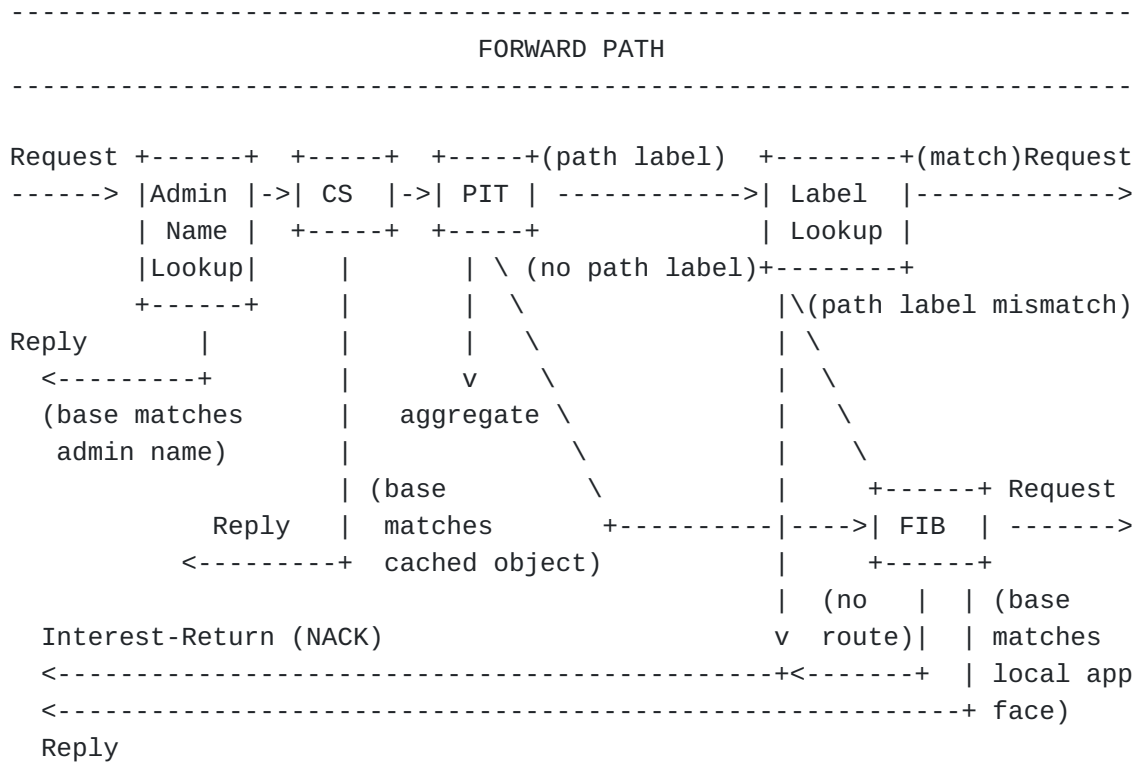


Figure 11: Forwarder Operation

## 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 be mounted by a compromised forwarder in the case of an echo reply with the CCNx packet format if that forwarder includes in the reply the name of a victim forwarder. This could convince a client to direct 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.

Another example of an attack could be the ICN equivalent of port knocking, where an attacker tries to discover certain forwarder implementations for the purpose of exploiting potential vulnerabilities.

## 9. IANA Considerations

IANA will assign 0x05 to "PT\_ECHO\_REQUEST" and 0x06 to "PT\_ECHO\_REPLY" in the CCNx Packet Types registry established by [\[RFC8609\]](#).

IANA will assign 0x0003 to "T\_NONCE" in the Name Segment Type IANA Registry for CCNx established by [\[RFC8609\]](#).

IANA will create an "Echo Reply Code" registry. IANA will assign 0x01 to "T\_ECHO\_RETURN\_FORWARDER", 0x02 to "T\_ECHO\_RETURN\_APPLICATION", and 0x03 to "T\_ECHO\_RETURN\_OBJECT" in the "Echo Reply Code" registry.

## 10. Acknowledgements

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

## 11. References

### 11.1. 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>>.
- [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>>.

**[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>>.

## 11.2. Informative References

**[I-D.irtf-icnrg-pathsteering]** Moiseenko, I. and D. R. Oran, "Path Steering in CCNx and NDN", Work in Progress, Internet-Draft, draft-irtf-icnrg-pathsteering-03, 23 July 2023, <<https://datatracker.ietf.org/doc/html/draft-irtf-icnrg-pathsteering-03>>.

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

**[NDNTLV]** "NDN Packet Format Specification.", 2021, <<https://named-data.net/doc/NDN-packet-spec/current/>>.

**[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.

**[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>>.

**[SNAMP]** Afanasyev, A., "SNAMP: Secure namespace mapping to scale NDN forwarding", 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 Path label TLV, since no TLV value is known. After an echo reply containing a Path label TLV is received, each subsequent echo request can 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 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 Path label 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 Notre Dame  
South Bend, IN  
United States of America

Email: [smastor2@nd.edu](mailto:smastor2@nd.edu)

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

Email: [daveoran@orandom.net](mailto:daveoran@orandom.net)

Jim Gibson  
Unaffiliated  
Belmont, MA  
United States of America

Email: [jcgibson61@gmail.com](mailto:jcgibson61@gmail.com)

Ilya Moiseenko  
Apple Inc  
Cupertino, CA  
United States of America

Email: [iliamo@mailbox.org](mailto:iliamo@mailbox.org)

Ralph Droms  
Unaffiliated  
Hopkinton, MA  
United States of America

Email: [rdroms.ietf@gmail.com](mailto:rdroms.ietf@gmail.com)