ICN Research Group
Internet-Draft
Intended status: Experimental
Expires: May 4, 2017

H. Asaeda
X. Shao
NICT
T. Turletti
Inria

October 31, 2016

# Contrace: Traceroute Facility for Content-Centric Network
## draft-asaeda-icnrg-contrace-00

Abstract

This document describes the traceroute facility for Content-Centric Network (CCN), named "Contrace". Contrace investigates: 1) the forwarding path information per name prefix, device name, and function/application, 2) the Round-Trip Time (RTT) between content forwarder and consumer, and 3) the states of in-network cache per name prefix.

Status of This Memo



Copyright Notice

Table of Contents

## 1. Introduction

In Content-Centric Network (CCN) or Named-Data Network (NDN), publishers provide content through the network, and receivers retrieve content by name. In this network architecture, routers forward content requests by means of their Forwarding Information Bases (FIBs), which are populated by name-based routing protocols. CCN/NDN also enables receivers to retrieve content from an in-network cache.

In CCN/NDN, while consumers do not generally need to know which content forwarder is transmitting the content to them, operators and developers may want to identify the content forwarder and observe the forwarding path information per name prefix for troubleshooting or investigating the network conditions.

Traceroute [5] is a useful tool for analyzing the routing conditions in IP networks as it provides intermediate router addresses along the path between source and destination and the Round-Trip Time (RTT) for the path. However, this IP-based network tool cannot trace the name prefix paths used in CCN/NDN. Moreover, given a source-rooted forwarding path per name prefix, specifying a forwarding source (i.e., router or publisher) for any content is difficult, because we do not always know which branch of the source tree the consumer is on. Additionally, it is not feasible to flood the entire source-rooted tree to find the path from a source to a consumer. Furthermore, such IP-based network tool does not allow the states of the in-network cache to be discovered.

This document describes the specification of "Contrace", an active network measurement tool for investigating the path and caching condition in CCN. Contrace is designed based on the work originally published in [4].

Contrace consists of the Contrace user command and the Contrace forwarding function implementation on a content forwarder (e.g., router). The Contrace user (e.g., consumer) invokes the contrace command (described in Appendix A) with the name prefix of the content, the device name, or the function (or application) name. The Contrace command initiates the Contrace "Request" message (described in Section 3.1). The Request message, for example, obtains forwarding path and cache information. When an appropriate adjacent neighbor router receives the Request message, it retrieves cache information. If the router is not the content forwarder for the request, it inserts its "Report" block (described in Section 3.1.2)

into the Request message and forwards the Request message to its upstream neighbor router(s) decided by its FIB.  These two message types, Contrace Request and Reply messages, are encoded in the CCNx TLV format [1].

In this way, the Contrace Request message is forwarded by routers toward the content publisher, and the Contrace Report record is inserted by each intermediate router.  When the Request message reaches the content forwarder (i.e., a router or the publisher who has the specified cache or content), the content forwarder forms the Contrace "Reply" message (described in Section 3.2) and sends it to the downstream neighbor router.  The Reply message is forwarded back toward the Contrace user in a hop-by-hop manner.  This request-reply message flow, walking up the tree from a consumer toward a publisher, is inspired by the design of the IP multicast traceroute facility [6].

Contrace supports multipath forwarding.  The Request messages can be forwarded to multiple neighbor routers.  When the Request messages forwarded to multiple routers, the different Reply messages will be forwarded from different routers or publisher.  To support this case, PIT entries initiated by Contrace remain until the defined timeout value is expired.

```
        1. Request    2. Request    3. Request    4. Request
           (+U)          (U+A)         (U+A+B)       (U+A+B+C)
          +----+        +----+        +----+        +----+
          |    |        |    |        |    |        |    |
          |    v        |    v        |    v        |    v
 +--------+    +--------+    +--------+    +--------+    +---------+
 |Contrace|----| Router |----| Router |----| Router |----|Publisher|
 |  user  |    |   A    |    |   B    |    |   C    |    |         |
 +--------+    +--------+    +--------+    +--------+    +---------+
                                      \
                                       \          +-------+
                             3. Request \         | Cache |
                                (U+A+B)  \ +---------+    |
                                          v| Caching |----+
                                           |  router |
                                           +---------+
```

Figure 1: Request messages forwarded by consumer and routers. Contrace user and routers (i.e., Router A,B,C) insert their own Report blocks into the Request message and forward the message toward the content forwarder (i.e., caching router and publisher)

```
           3. Reply(C)   2. Reply(C)
           4. Reply(P)   3. Reply(P)   2. Reply(P)   1. Reply(P)
             +----+        +----+        +----+        +----+
             |    |        |    |        |    |        |    |
             v    |        v    |        v    |        v    |
    +--------+    +--------+    +--------+    +--------+    +----------+
    |Contrace|----| Router |----| Router |----| Router |----|Publisher|
    |  user  |    |   A    |    |   B    |    |   C    |    |         |
    +--------+    +--------+    +--------+    +--------+    +----------+
                                         ^
                                          \          +-------+
                               1. Reply(C) \         | Cache |
                                            \ +---------+    |
                                             \| Caching |----+
                                              |  router |
                                              +---------+
```

Figure 2: Reply messages forwarded by publisher and routers. Each router forwards the Reply message, and finally the Contrace user receives two Reply messages: one from the publisher and the other from the caching router.

Contrace facilitates the tracing of a routing path and provides: 1) the RTT between content forwarder (i.e., caching router or publisher) and consumer, 2) the states of in-network cache per name prefix, and 3) the forwarding path information per name prefix.

In addition, Contrace identifies the states of the cache, such as the following metrics for Content Store (CS) in the content forwarder: 1) size of the cached content, 2) number of the cached chunks of the content, 3) number of the accesses (i.e., received Interests) per cache or chunk, and 4) lifetime and expiration time per cache or chunk. The number of received Interests per cache or chunk on the routers indicates the popularity of the content.

Furthermore, Contrace implements policy-based information provisioning that enables administrators to "hide" secure or private information, but does not disrupt the forwarding of messages. This policy-based information provisioning reduces the deployment barrier faced by operators in installing and running Contrace on their routers.

## 2. Terminology

In this document, the key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" are to be interpreted as described in RFC 2119 [2],

and indicate requirement levels for compliant Contrace implementations.

### 2.1. Definitions

Since Contrace requests flow in the opposite direction to the data flow, we refer to "upstream" and "downstream" with respect to data, unless explicitly specified.

Router
: It is a CCN-capable router in the path between consumer and publisher.

Node
: It is a router, publisher, or consumer.

Content forwarder
: It is either a router or a publisher that holds the cache (or content) and forwards it to consumers.

Contrace user
: It is a node that invokes the contrace command and initiates the Contrace Request.

Incoming face
: The face on which data is expected to arrive from the specified name prefix.

Outgoing face
: The face to which data from the publisher or router is expected to transmit for the specified name prefix. It is also the face on which the Contrace Request messages are received.

## 3. Contrace Message Formats

Contrace uses two message types: Request and Reply. Both messages are encoded in the CCNx TLV format ([1], Figure 3). The Request message consists of a fixed header, Request block TLV Figure 7, and Report block TLV(s) Figure 10. The Reply message consists of a fixed header, Request block TLV, Report block TLV(s), and Reply block TLV(s) Figure 13.

```
                      1                   2                   3
  0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
 +---------------+---------------+---------------+---------------+
 |    Version    |  PacketType   |         PacketLength          |
 +---------------+---------------+---------------+---------------+
 |           PacketType specific fields          | HeaderLength  |
 +---------------+---------------+---------------+---------------+
 / Optional Hop-by-hop header TLVs                               /
 +---------------+---------------+---------------+---------------+
 / PacketPayload TLVs                                            /
 +---------------+---------------+---------------+---------------+
 / Optional CCNx ValidationAlgorithm TLV                         /
 +---------------+---------------+---------------+---------------+
 / Optional CCNx ValidationPayload TLV (ValidationAlg required)  /
 +---------------+---------------+---------------+---------------+
```

Figure 3: Packet format [1]

The Request and Reply Type values in the fixed header are PT_TRACE_REQ and PT_TRACE_REPLY, respectively (Figure 4). These messages are forwarded in a hop-by-hop manner. When the Request message reaches the content forwarder, the content forwarder turns the Request message into a Reply message by changing the Type field value in the fixed header from PT_TRACE_REQ to PT_TRACE_REPLY and forwards back to the node that has initiated the Request message.

| Code | Type name |
|---|---|
| 1 | PT_INTEREST [1] |
| 2 | PT_CONTENT [1] |
| 3 | PT_RETURN [1] |
| 4 | PT_TRACE_REQ |
| 5 | PT_TRACE_REPLY |

Figure 4: Packet Type Namespace

Each Contrace message MUST begin with a fixed header with either a Request or Reply type value to specify whether it is a Request message or Reply message. Following a fixed header, there can be a sequence of optional hop-by-hop header TLV(s) for a Request message. In the case of a Request message, it is followed by a sequence of Report blocks, each from a router on the path toward the publisher or caching router.

At the beginning of PacketPayload TLVs, one top-level TLV type, T_TRACE (Figure 5), exists at the outermost level of a CCNx protocol message. This TLV indicates that the Name segment TLV(s) and Reply block TLV(s) would follow in the Request or Reply message.

| Code | Type name |
| --- | --- |
| 1 | T_INTEREST [1] |
| 2 | T_OBJECT [1] |
| 3 | T_VALIDATION_ALG [1] |
| 4 | T_VALIDATION_PAYLOAD [1] |
| 5 | T_TRACE |

Figure 5: Top-Level Type Namespace

### 3.1. Request Message

When a Contrace user initiates a trace request (e.g., by contrace command described in Appendix A), a Contrace Request message is created and forwarded to its upstream router through the Incoming face(s) determined by its FIB.

The packet format of the Contrace Request message is as shown in Figure 6. It consists of a fixed header, Request block TLV (Figure 7), Report block TLV(s) (Figure 10), and Name TLV. The Type value of Top-Level type namespace is T_TRACE (Figure 5). The Type value for the Report message is PT_TRACE_REQ.

```
                        1                   2                   3
     0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
    +---------------+---------------+---------------+---------------+
    |    Version    | PT_TRACE_REQ  |          PacketLength         |
    +---------------+---------------+---------------+---------------+
    |    HopLimit   |   ReturnCode  |Reserved (MBZ) | HeaderLength  |
    +===============+===============+===============+===============+
    |                                                               |
    +                       Request block TLV                       +
    |                                                               |
    +---------------+---------------+---------------+---------------+
    /                       Report block TLV 1                      /
    +---------------+---------------+---------------+---------------+
    /                       Report block TLV 2                      /
    +---------------+---------------+---------------+---------------+
    /                               .                               /
    /                               .                               /
    +---------------+---------------+---------------+---------------+
    /                       Report block TLV n                      /
    +===============+===============+===============+===============+
    |            T_TRACE            |         MessageLength         |
    +---------------+---------------+---------------+---------------+
    |            T_NAME             |             Length            |
    +---------------+---------------+---------------+---------------+
    / Name segment TLVs (name prefix specified by contrace command) /
    +---------------+---------------+---------------+---------------+
```

Figure 6: Packet format of the Request message

HopLimit: 8 bits

HopLimit is a counter that is decremented with each hop. It limits the distance a Request may travel on the network.

ReturnCode: 8 bits

ReturnCode is used for the Reply message. This value is replaced by the content forwarder when the Request message is returned as the Reply message (see Section 3.2). Until then, this field MUST be transmitted as zeros and ignored on receipt.

| Value | Name | Description |
|---|---|---|
| 0x00 | NO_ERROR | No error |
| 0x01 | WRONG_IF | Contrace Request arrived on an interface to which this router would not forward for the specified name/function toward the publisher. |
| 0x02 | NO_ROUTE | This router has no route for the named prefix and no way to determine a potential route. |
| 0x03 | NO_INFO | This router has no cache information for the specified name prefix, device information, or function. |
| 0x04 | NO_SPACE | There was not enough room to insert another Report block in the packet. |
| 0x05 | INFO_HIDDEN | Information is hidden from this trace because of some policy. |
| 0x06 | REACHED_GW | Contrace Request arrived on an IP gateway (e.g., a NAT or firewall) that hides the information between this router and the Contrace user. |
| 0x0E | ADMIN_PROHIB | Contrace Request is administratively prohibited. |
| 0x0F | UNKNOWN_REQUEST | This router does not support/recognize the Request message. |
| 0x80 | FATAL_ERROR | A fatal error is one where the router may know the upstream router but cannot forward the message to it. |

Reserved (MBZ): 8 bits

The reserved fields in the Value field MUST be transmitted as zeros and ignored on receipt.

### 3.1.1. Request Block

When a Contrace user transmits the Request message, it would insert the Request block TLV (Figure 7) and the Report block TLV (Figure 10) of its own to the Request message before sending it through the Incoming face(s).

```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+---------------+---------------+---------------+---------------+
|          T_TRACE_REQ          |             Length            |
+---------------+---------------+---------------+---------------+
|   Request ID  | SkipHopCount  |     Flags     |Reserved (MBZ) |
+---------------+---------------+---------------+---------------+
```

Figure 7: Request block TLV (hop-by-hop header)

| Code | Type name |
|---|---|
| 1 | T_INTLIFE [1] |
| 2 | T_CACHETIME [1] |
| 3 | T_MSGHASH [1] |
| 4 - 7 | Reserved [1] |
| 8 | T_TRACE_REQ |
| 9 | T_TRACE_RPT |
| %x0FFE | T_PAD [1] |
| %x0FFF | T_ORG [1] |
| %x1000-%x1FFF | Reserved [1] |

Figure 8: Hop-by-Hop Type Namespace

Type: 16 bits

Format of the Value field. For the single Request block TLV, the type value MUST be T_TRACE_REQ. For all the available types for hop-by-hop type namespace, please see Figure 8.

Length: 16 bits

Length of Type, Length, and Value fields in octets. For the Request block, it MUST be 8 in the current specification.

Request ID: 8 bits

This field is used as a unique identifier for this Contrace Request so that duplicate or delayed Reply messages can be detected.

SkipHopCount: 8 bits

Number of skipped routers. This value MUST be lower than the value of HopLimit at the fixed header.

Flags: 8 bits

The trace conditions specified as the contrace command options (described in Appendix A) are transferred in the Flags field. The trace conditions depend on the specified name (i.e., name_prefix, device_name, or function_name) as shown in Figure 9.

| Code | Type name |
|---|---|
| %x01 | Cache retrieval allowing partial match (name_prefix) |
| %x02 | No cache information required (name_prefix) |
| %x04 | Publisher reachability (name_prefix and device_name) |
| %x08 | Function's or application's version number retrieval (function_name) |
| %x16 | Not assigned |
| %x32 | Not assigned |
| %x64 | Not assigned |
| %x128 | Not assigned |

Figure 9: Codes and types specified in Flags field

### 3.1.2. Report Block

A Contrace user and each upstream router along the path would insert its own Report block TLV without changing the Type field of the Request message until one of these routers is ready to send a Reply. In the Report block TLV (Figure 10), the Request Arrival Time and the Node Identifier are inserted.

```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+---------------+---------------+---------------+---------------+
|          T_TRACE_RPT          |             Length            |
+---------------+---------------+---------------+---------------+
|                                                               |
+                     Request Arrival Time                      +
|                                                               |
+---------------+---------------+---------------+---------------+
/                        Node Identifier                        /
+---------------+---------------+---------------+---------------+
```

Figure 10: Report block TLV (hop-by-hop header)

Type: 16 bits

Format of the Value field. For the Request block TLV(s), the type value(s) MUST be T_TRACE_RPT.

Length: 16 bits

Length of Type, Length, and Value fields in octets. The length of the value field is Length field minus 4.

Request Arrival Time: 32 bits

The Request Arrival Time is a 32-bit NTP timestamp specifying the arrival time of the Contrace Request packet at this router. The 32-bit form of an NTP timestamp consists of the middle 32 bits of the full 64-bit form; that is, the low 16 bits of the integer part and the high 16 bits of the fractional part.

The following formula converts from a UNIX timeval to a 32-bit NTP timestamp:

```
query_arrival_time
= (tv.tv_sec + 32384) << 16 + ((tv.tv_usec << 10) / 15625)
```

The constant 32384 is the number of seconds from Jan 1, 1900 to Jan 1, 1970 truncated to 16 bits. ((tv.tv_usec << 10) / 15625) is a reduction of ((tv.tv_usec / 100000000) << 16).

Note that Contrace does not require all the routers on the path to have synchronized clocks in order to measure one-way latency.

Node Identifier: variable length

This field specifies the Contrace user or the router identifier (e.g., IPv4 address) of the Incoming face on which packets from the publisher are expected to arrive, or 0 if unknown or unnumbered. Note that although it would be necessary to define the identifier uniqueness (e.g., by specifying the protocol family) for this field, defining such uniqueness is [TBD] as we may not always rely on the IP addressing architecture.

### 3.2. Reply Message

When a content forwarder receives a Contrace Request message from the appropriate adjacent neighbor router, it would insert the Reply block TLV(s) of its own to the Request message and turn the Request into the Reply by changing the Type field of the Request message from PT_TRACE_REQ to PT_TRACE_RPT. The Reply message (see Figure 11) would then be forwarded back toward the Contrace user in a hop-by-hop manner.

```
                        1                   2                   3
     0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
    +---------------+---------------+---------------+---------------+
    |    Version    |PT_TRACE_REPLY |         PacketLength          |
    +---------------+---------------+---------------+---------------+
    |    HopLimit   |   ReturnCode  |Reserved (MBZ) | HeaderLength  |
    +===============+===============+===============+===============+
    |                                                               |
    +                       Request block TLV                       +
    |                                                               |
    +---------------+---------------+---------------+---------------+
    /                               .                               /
    /                               .                               /
    /                      n Report block TLVs                      /
    /                               .                               /
    /                               .                               /
    +===============+===============+===============+===============+
    |            T_TRACE            |         MessageLength         |
    +---------------+---------------+---------------+---------------+
    |            T_NAME             |             Length            |
    +---------------+---------------+---------------+---------------+
    / Name segment TLVs (name prefix specified by contrace command) /
    +---------------+---------------+---------------+---------------+
    /                       Reply block TLV 1                       /
    +---------------+---------------+---------------+---------------+
    /                       Reply block TLV 2                       /
    +---------------+---------------+---------------+---------------+
    /                               .                               /
    /                               .                               /
    +---------------+---------------+---------------+---------------+
    /                       Reply block TLV k                       /
    +---------------+---------------+---------------+---------------+
```

Figure 11: Reply message consists of a fixed header, Request block TLV, Report block TLV(s), Name TLV, and Reply block TLV(s)

| Code | Type name |
| --- | --- |
| 0 | T_NAME [1] |
| 1 | T_PAYLOAD [1] |
| 2 | T_KEYIDRESTR [1] |
| 3 | T_OBJHASHRESTR [1] |
| 5 | T_PAYLDTYPE [1] |
| 6 | T_EXPIRY [1] |
| 7 | T_TRACE_REPLY_CONTENT |
| 8 | T_TRACE_REPLY_DEVICE |
| 9 | T_TRACE_REPLY_FUNCTION |
| 10 - 12 | Reserved [1] |
| %x0FFE | T_PAD [1] |
| %x0FFF | T_ORG [1] |
| %x1000-%x1FFF | Reserved [1] |

Figure 12: CCNx Message Type Namespace

### 3.2.1. Reply Block

Three kinds of Reply block TLVs exist, Reply content block TLV, Reply device block TLV, and Reply function block TLV; however, in this document, only the Reply content block TLV as shown in Figure 13 is defined, and other Reply block TLVs, Reply device block TLV and Reply function block TLV are [TBD].

```
                       1                   2                   3
   0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
  +---------------+---------------+---------------+---------------+
  |     T_TRACE_REPLY_CONTENT     |             Length            |
  +---------------+---------------+---------------+---------------+
  |                          Content Size                         |
  +---------------+---------------+---------------+---------------+
  |                          Object Count                         |
  +---------------+---------------+---------------+---------------+
  |                       # Received Interest                     |
  +---------------+---------------+---------------+---------------+
  |                          First Seqnum                         |
  +---------------+---------------+---------------+---------------+
  |                           Last Seqnum                         |
  +---------------+---------------+---------------+---------------+
  |                                                               |
  +                         Cache Lifetime                        +
  |                                                               |
  +---------------+---------------+---------------+---------------+
  |                                                               |
  +                      Remain Cache Lifetime                    +
  |                                                               |
  +---------------+---------------+---------------+---------------+
  |            T_NAME             |             Length            |
  +---------------+---------------+---------------+---------------+
  /       Name segment TLVs (name prefix partially matched)       /
  +---------------+---------------+---------------+---------------+
```

Figure 13: Reply (content) block TLV (packet payload)

Type: 16 bits

Format of the Value field. For the Report block TLV, the type value MUST be T_TRACE_REPLY_CONTENT when the content or cache information is replied for the trace.

Length: 16 bits

Length of Type, Length, and Value fields in octets.

Content Size: 16 bits

The total size (MB) of the (cached) content objects. Note that the maximum size expressed by 16 bit field is 65 GB.

Object Count: 16 bits

The number of the (cached) content objects.

# Received Interest: 16 bits

The number of the received Interest messages to retrieve the content.

First Seqnum: 16 bits

The first sequential number of the (cached) content objects.

Last Seqnum: 16 bits

The last sequential number of the (cached) content objects. Above First Seqnum and this Last Seqnum do not guarantee the consecutiveness of the cached content objects.

Cache Lifetime: 32 bits

The elapsed time after the oldest content object in the cache is stored. The Cache Lifetime is a 32-bit NTP timestamp, and the formula converts from a UNIX timeval to a 32-bit NTP timestamp is same as that of Section 3.1.2.

Remain Cache Lifetime: 32 bits

The lifetime of a content object, which is removed first among the cached content objects. The Remain Cache Lifetime is a 32-bit NTP timestamp.

## 4. Contrace User Behavior

### 4.1. Sending Contrace Request

A Contrace user initiates a Contrace Request by sending the Request message to the adjacent neighbor router(s) of interest. As a typical example, a Contrace user invokes the contrace command (detailed in Appendix A) that forms a Request message and sends it to the user's adjacent neighbor router(s).

When the Contrace user's program initiates a Request message, it MUST fill all values in the Request and Report blocks such as the "Request ID" (in the Request block) and the "Node Identifier" (in the Report block). Contrace user's program MUST also record the Request ID at the corresponding PIT entry. The Request ID is a unique identifier for this Contrace Request.

The Contrace user's program then sends the Request message. Until the Contrace user receives the Reply or the Reply times out, to verify the Reply, the Contrace user's program MUST keep the following

information; Request ID and Flags specified in the Request block, Node Identifier and Request Arrival Time specified in the Report block, and HopLimit specified in the fixed header.

### 4.2. Receiving Contrace Reply

A Contrace user's program will receive a Contrace Reply from the adjacent neighbor router(s) that has previously received and forwarded the Request message(s). When the program receives the Reply, it MUST compare the kept Request ID and the Request ID noted in the Reply. If they do not match, the Reply message SHOULD be silently discarded.

If the number of the Report blocks in the received Reply is more than the initial HopLimit value (which was inserted in the original Request) + 1, the Reply SHOULD be silently ignored.

## 5. Router Behavior

### 5.1. Receiving Contrace Request

Upon receiving a Contrace Request message, a router MUST examine whether the message comes from a valid adjacent neighbor node. If it is invalid, the Request SHOULD be silently ignored.

When a router receives a Request without an error, the router retrieves the cache information from its CS. If the router is the caching router that caches the requested content, it sends the Reply message. See Section 5.3. Otherwise, the router forwards the Request message to its upstream router(s). See Section 5.2.

If a router cannot continue the Request, it MUST note an appropriate ReturnCode in the Request message, change the Type field value in the fixed header from PT_TRACE_REQ to PT_TRACE_REPLY, and forward the message as the Reply back to the Contrace user. See Section 7 for details.

### 5.2. Forwarding Contrace Request

When a router decides to forward the Request messages, the router prepares a Report block TLV to be inserted to the hop-by-hop TLV header of the Request message with the Request Arrival Time and Node Identifier and forwards the Request upstream toward the publisher or caching router.

When the router forwards the Request message, it MUST record the "Request ID" indicated in the Request block at the corresponding PIT entry. The router can then distinguish the neighbor node to forward

back the Reply message even if duplicate or delayed Reply messages are detected.

Contrace supports multipath forwarding. The Request messages can be forwarded to multiple neighbor routers. When the Request messages forwarded to multiple routers, the different Reply messages will be forwarded from different routers or publisher. To support this case, PIT entries initiated by Contrace remain until the defined Contrace Reply Timeout value (Section 8.1) expires. In other words, unlike the ordinary Interest-Data communications in CCN, the router SHOULD NOT remove the PIT entry created by the Contrace Request before the timeout value expires, even if the router receives the Contrace Reply.

Contrace Requests SHOULD NOT result in PIT aggregation in routers during the Request message transmission.

### 5.3. Sending Contrace Reply

When Contrace on a router decides to send the Reply message, it inserts the Report block the Request Arrival Time and Node Identifier to the the hop-by-hop TLV header and the Reply block(s) to the CCNx message TLV. The router then turns the Type field of the Request message from PT_TRACE_REQ to PT_TRACE_RPT and forwards the message back as the Reply toward the Contrace user in a hop-by-hop manner.

### 5.4. Forwarding Contrace Reply

When the router receives a Contrace Reply whose Request ID matches the one in the original Contrace Request block TLV from a valid adjacent neighbor node, it MUST relay the Contrace Reply back to the Contrace user. If the router does not receive the corresponding Reply within the [Contrace Reply Timeout] period, then it removes the corresponding PIT entry and terminates the trace.

Contrace Replies MUST NOT be cached in routers upon the Reply message transmission.

## 6. Publisher Behavior

Upon receiving a Contrace Request message, a publisher MUST examine whether the message comes from a valid adjacent neighbor node. If it is invalid, the Request SHOULD be silently ignored.

If a publisher cannot accept the Request, it MUST note an appropriate ReturnCode in the Request message, change the Type field value in the fixed header from PT_TRACE_REQ to PT_TRACE_REPLY, and forward the

message as the Reply back to the Contrace user. See Section 7 for details.

If a publisher accepts the Request forwarded by a valid adjacent neighbor node, it retrieves the local content information. The Reply message is transmitted back to the neighbor node that had forwarded the Request message.

## 7. Contrace Termination

When performing an expanding hop-by-hop trace, it is necessary to determine when to stop expanding. There are several cases an intermediate router might return a Reply before a Request reaches the caching router or the publisher.

### 7.1. Arriving at publisher

A Contrace Request can be determined to have arrived at the publisher.

### 7.2. Arriving at router having cache

A Contrace Request can be determined to have arrived at the router having the specified content cache within the specified HopLimit.

### 7.3. No space

If appending the Report block would make the Contrace Request packet longer than the MTU of the Incoming face, or longer than 1280 bytes (especially in the situation supporting IPv6 as the payload [3]), the router MUST note a ReturnCode of NO_SPACE in the fixed header of the message, and forwards the message as the Reply back to the Contrace user.

### 7.4. No route

If the router cannot determine the forwarding paths or neighbor routers for the specified named prefix, device name, or function, the router MUST note a ReturnCode of NO_ROUTE in the fixed header of the message, and forwards the message as the Reply back to the Contrace user.

### 7.5. Fatal error

A Contrace Request has encountered a fatal error if the last ReturnCode in the trace has the 0x80 bit set (see Section 3.1).

### 7.6. Contrace Reply Timeout

If a Contrace user or a router encounters the Request or Reply message whose expires its own [Contrace Reply Timeout] value (Section 8.1, which is used to time out a Contrace Reply such as the case of Section 7.7.

### 7.7. Non-Supported Node

Cases will arise in which a router or a publisher along the path does not support Contrace. In such cases, a Contrace user and routers that forward the Contrace Request will time out the Contrace request.

### 7.8. Administratively Prohibited

If Contrace is administratively prohibited, a router or a publisher rejects the Request message and its downstream router will reply the Contrace Reply with the ReturnCode of ADMIN_PROHIB.

## 8. Configurations

### 8.1. Contrace Reply Timeout

The [Contrace Reply Timeout] value is used to time out a Contrace Reply such as the case of Section 7.7. The default [Contrace Reply Timeout] value is 8 (seconds), and this timeout value can be manually changed by Contrace users using the contrace command and routers. However, the [Contrace Reply Timeout] value SHOULD NOT be larger than 10 (seconds) and SHOULD NOT be lower than 4 (seconds).

### 8.2. HopLimit in Fixed Header

If a Contrace user does not specify the HopLimit value in a fixed header for a Request message as the HopLimit, the HopLimit is set to 32. Note that a Contrace user specifies 0 as the HopLimit, it is an invalid Request and discarded.

### 8.3. Access Control

A router MAY configure the valid or invalid networks to enable an access control. The access control can be defined per named prefix, such as "who can retrieve which named prefix". See Section 9.2.

## 9. Security Considerations

This section addresses some of the security considerations.

## 9.1. Policy-Based Information Provisioning for Request

Although Contrace gives excellent troubleshooting cues, some network administrators or operators may not want to disclose everything about their network to the public, or may wish to securely transmit private information to specific members of their networks. Contrace provides policy-based information provisioning allowing network administrators to specify their response policy for each router.

The access policy regarding "who is allowed to retrieve what kind of information" can be defined for each router. The permission, whether (1) all cache information is disclosed, (2) partially disclosed (e.g., except the request specifying the default name prefix (e.g., ccnx:/)), or (3) not disclosed at all, is defined at a router and a publisher. It is RECOMMENDED that the Contrace Request with the default name prefix is only allowed to the approved Contrace users with access control. See Section 9.2 for more detail.

On the other hand, we entail that each router does not disrupt forwarding Contrace Request and Reply messages. When a Request message is received, the router SHOULD insert Report block. Here, according to the policy configuration, the Node Identifier field in the Report block MAY be null (i.e., all-zeros), but the Request Arrival Time field SHOULD NOT be null. At last, the router SHOULD forward the Request message to the upstream router toward the content forwarder if no fatal error occurs.

## 9.2. Filtering of Contrace Users Located in Invalid Networks

A router MAY support an access control mechanism to filter out Requests from invalid Contrace users. For it, invalid networks (or domains) could, for example, be configured via a list of allowed/ disallowed networks (as seen in Section 8.3). If a Request is received from the disallowed network (according to the Node Identifier in the Request block), the Request SHOULD NOT be processed and the Reply with the ReturnCode of INFO_HIDDEN may be used to note that. The router MAY, however, perform rate limited logging of such events.

## 9.3. Topology Discovery

Contrace can be used to discover actively-used topologies. If a network topology is a secret, Contrace Requests may be restricted at the border of the domain, using the ADMIN_PROHIB return code.

### 9.4. Characteristics of Content

Contrace can be used to discover what publishers are sending to what kinds of contents. If this information is a secret, Contrace Requests may be restricted at the border of the domain, using the ADMIN_PROHIB return code.

### 9.5. Limiting Request Rates

A router may limit Contrace Requests by ignoring some of the consecutive messages. The router MAY randomly ignore the received messages to minimize the processing overhead, i.e., to keep fairness in processing requests, or prevent traffic amplification. No error is returned. The rate limit is left to the router's implementation.

### 9.6. Limiting Reply Rates

Contrace supporting multipath forwarding may result in one Request returning multiple Reply messages. In order to prevent abuse, the routers in the traced path MAY need to rate-limit the Replies. No error is returned. The rate limit function is left to the router's implementation.

### 9.7. Adjacency Verification

Contrace Request and Reply messages MUST be forwarded by adjacent neighbor nodes or routers. Defining the secure way to verify the adjacency is [TBD]. Note that forwarding Contrace messages given from non-adjacent neighbor nodes/routers MUST be prohibited. Such invalid messages SHOULD be silently discarded.

## 10. References

### 10.1. Normative References

[1] Mosko, M., Solis, I., and C. Wood, "CCNx Messages in TLV Format", draft-irtf-icnrg-ccnxmessages-03 (work in progress), June 2016.

[2] Bradner, S., "Key words for use in RFCs to indicate requirement levels", RFC 2119, March 1997.

[3] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", RFC 2460, December 1998.

### 10.2. Informative References

[4] Asaeda, H., Matsuzono, K., and T. Turletti, "Contrace: A Tool for Measuring and Tracing Content-Centric Networks", IEEE Communications Magazine, Vol.53, No.3, pp.182-188, March 2015.

[5] Malkin, G., "Traceroute Using an IP Option", RFC 1393, January 1993.

[6] Asaeda, H., Mayer, K., and W. Lee, "Mtrace Version 2: Traceroute Facility for IP Multicast", draft-ietf-mboned-mtrace-v2-16 (work in progress), October 2016.

## Appendix A. Contrace Command and Options

The contrace command enables the Contrace user to investigate the forwarding path based on the name prefix of the content (e.g., ccnx:/news/today), device name, and function (or application) name. The name prefix, device name, and function name (or application name) are mandatory but exclusive options; that is, only one of them should be used with the contrace command at once.

The usage of contrace command is as follows:

```
Usage:contrace [-p] name_prefix [-n] [-o] [-r hop_count] [-s
      hop_count] [-w wait_time]; or,

Usage:contrace device_name | function_name (or application_name) [-r
      hop_count] [-s hop_count] [-w wait_time]
```

name_prefix
   Name prefix of the content (e.g., ccnx:/news/today) the Contrace user wants to trace. If the Contrace user does not know the name prefix of the content, the default name prefix, e.g., "ccnx:/" can be specified. However, according to the security consideration and the policy configuration in Section 9, Reply with the default name prefix MAY be limited by routers (only for the permitted users decided by some means). If the default name prefix is specified and the content forwarder allows it, the Contrace user obtains all cache information from the content forwarder. The -p option allows a partial match for the name prefix; otherwise, an exact match is required.

device_name
   Device name (e.g., ccnx:/%device/server-A, ccnx:/%device/sensor-123) the Contrace user wants to trace. Here, we assume the contrace command with the "%device" prefix indicates the trace

request for specified device/server/node, but defining the syntax of device name specification is [TBD].

function_name (or application_name)
Function name (e.g., ccnx:/%function/firewall, ccnx:/%function/transcoding/mpeg2-h.264) or application name (e.g., ccnx:/%application/mplayer) the Contrace user wants to trace. Here, we assume the contrace command with the "%function" or "%application" prefix indicates the trace request for specified function or application, but defining the syntax of function or application name specification is [TBD].

n option
This option can be specified if a Contrace user only needs the routing path information to the specified content/cache and RTT between Contrace user and content forwarder (i.e., cache information is not given).

o option
This option can be specified if a Contrace user needs to trace the path to the content publisher. If this option is specified, each router along the path to the publisher only forwards the Request message; it does not insert each Report block and does not send Reply even if it caches the specified content. The publisher (who has the complete set of content and is not a caching router) replies the Reply message.

r option
Number of traced routers. If the Contrace user specifies this option, only the specified number of hops from the Contrace user trace the Request; each router inserts its own Report block and forwards the Request message to the upstream router(s), and the last router stops the trace and sends the Reply message back to the Contrace user. This value is set in the "HopLimit" field located in the fixed header of the Request. For example, when the Contrace user invokes the Contrace command with this option such as "-r 3", the two upstream routers along the path append their Report blocks in the Request message, and the next (and last) router sends back the Reply message. If there is a caching router within the hop count along the path, the caching router sends back the Reply message and terminates the trace request. If the last router does not have the corresponding cache, it replies the Reply message with NO_INFO return code (described in Section 3.1) with no Reply block TLV inserted. The Request messages are terminated at publishers; therefore, although the maximum value for this option a Contrace user can specify is 255, the Request messages should be in general reached at the publisher within significantly lower than 255 hops.

s option

Number of skipped routers. If the Contrace user specifies this option, the number of hops from the Contrace user simply forward the Contrace Request messages without adding its own Report block and without replying the Request, and the next upstream router starts the trace. This value is set in the "SkipHopCount" field located in the Request block TLV. For example, when the Contrace user invokes the Contrace command with this option such as "-s 3", the three upstream routers along the path only forwards the Request message, but does not append their Report blocks in the hop-by-hop headers and does not send the Reply messages even though they have the corresponding cache. The Request messages are terminated at publishers; therefore, although the maximum value for this option a Contrace user can specify is 255, if the Request messages reaches the publisher, the publisher silently discards the Request message and the request will be timed out.

w option

This option defines the Contrace timeout value (in seconds) that the Contrace user will wait for the Reply. After the timeout, the Contrace user terminates the Request and silently discards the Reply message even if s/he receives the Reply. Note that routers along the path can configure the Contrace Reply timeout Section 8.1, which is the allowable timeout value to keep the PIT entry. In order to avoid DoS attacks Section 9, routers MAY configure the shorter timeout value than the user-configured Contrace timeout value. If it is shorter, the Request may be timed out and the Contrace user may not receive the Reply as expected.

Authors' Addresses

Hitoshi Asaeda
National Institute of Information and Communications Technology
4-2-1 Nukui-Kitamachi
Koganei, Tokyo 184-8795
Japan

Email: asaeda@nict.go.jp

Xun Shao
National Institute of Information and Communications Technology
4-2-1 Nukui-Kitamachi
Koganei, Tokyo 184-8795
Japan

Email: x-shao@nict.go.jp

Thierry Turletti
Inria
2004 Route des Lucioles
Sophia Antipolis  06902
France

Email: thierry.turletti@inria.fr