ICN Research Group
Internet-Draft
Intended status: Experimental
Expires: January 4, 2018

H. Asaeda
X. Shao
NICT
T. Turletti
Inria

July 3, 2017

Contrace: Traceroute Facility for Content-Centric Network draft-asaeda-icnrg-contrace-03

Abstract

This document describes the traceroute facility for Content-Centric Network (CCN), named "Contrace". Contrace investigates: 1) the routing 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. In addition, it discovers a gateway that supports different protocols such as CCN and NDN.

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 http://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 January 4, 2018.

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 routing 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 routing 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 potentially discovers devices and functions/applications 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 routing 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.

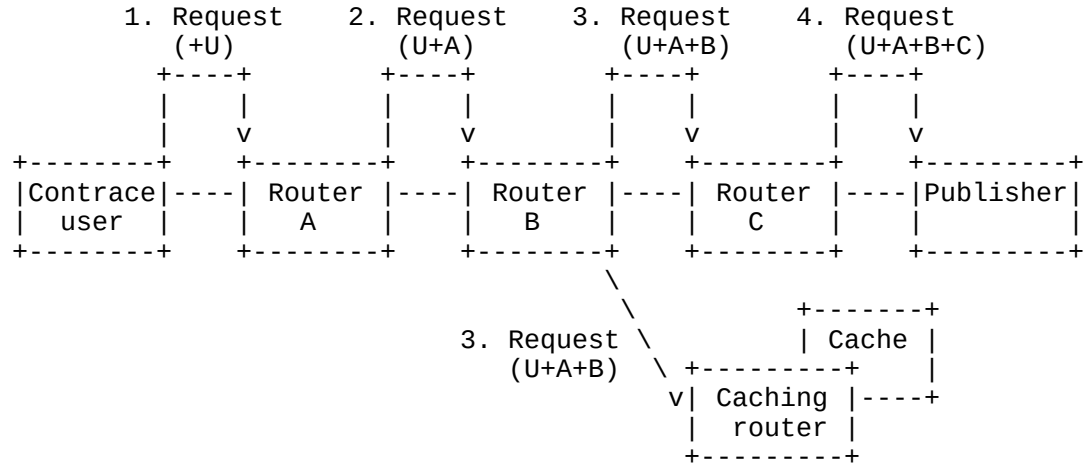

Figure 1: Request messages forwarded by consumer and routers. Contrace user and routers (i.e., Router A,B,C) insert their own Report blocks into the Request message and forward the message toward the content forwarder (i.e., caching router and publisher)

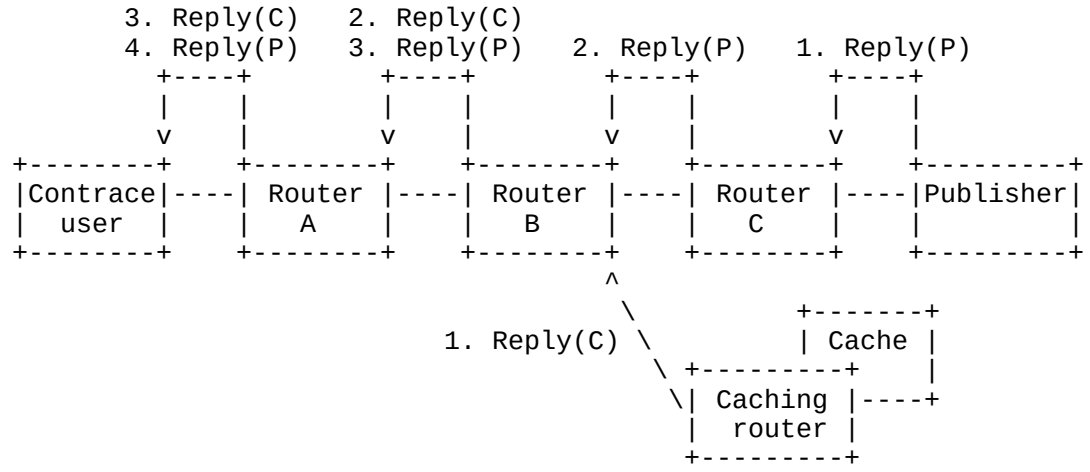

Figure 2: Reply messages forwarded by publisher and routers. Each router forwards the Reply message, and finally the Contrace user receives two Reply messages: one from the publisher and the other from the caching router.

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 routing 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 router facilitating name-based content/device/function name or characteristic retrieval in the path between consumer and publisher.

Scheme name
: It indicates a URI and protocol such as "ccnx:/" and "ndn:/". This document considers the protocol for name-based content/device/function name or characteristic retrieval.

Gateway
: It is a router supporting multiple scheme names in the path between consumer and publisher. The router has multiple FIBs for different protocols and establishes the connections with different neighbor routers for each protocol.

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

Content forwarder
It is either a caching 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 11. The Reply message consists of a fixed header, Request block TLV, Report block TLV(s), and Reply block/sub-block TLV(s) Figure 14.

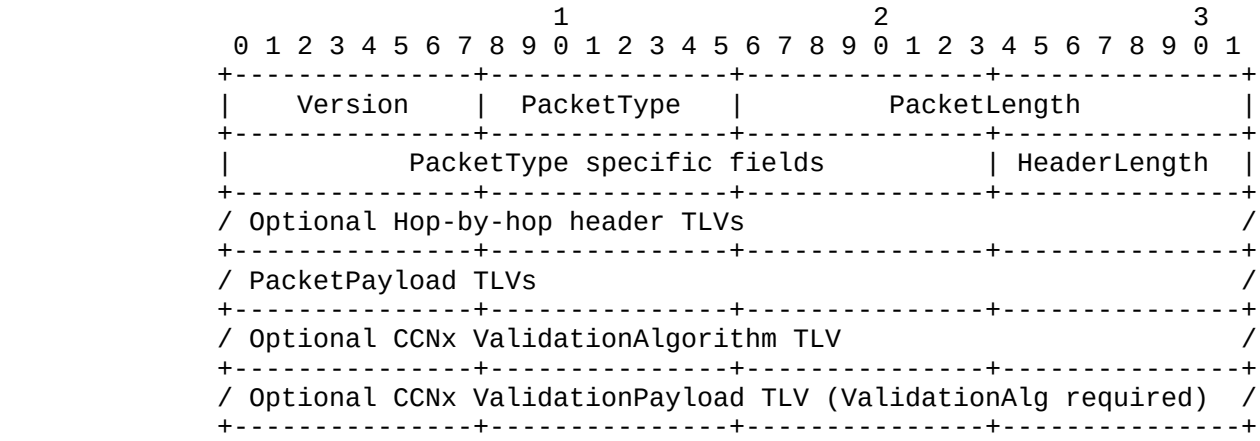

Figure 3: Packet format [1]

The Request and Reply Type values in the fixed header are PT_PING_TRACE_REQ and PT_PING_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_PING_TRACE_REQ to PT_PING_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_PING_TRACE_REQ
5	PT_PING_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_PING
6	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 Contrace Request message format is as shown in Figure 6. It consists of a fixed header, Request block TLV (Figure 7), Report

block TLV(s) (Figure 11), 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_PING_TRACE_REQ.

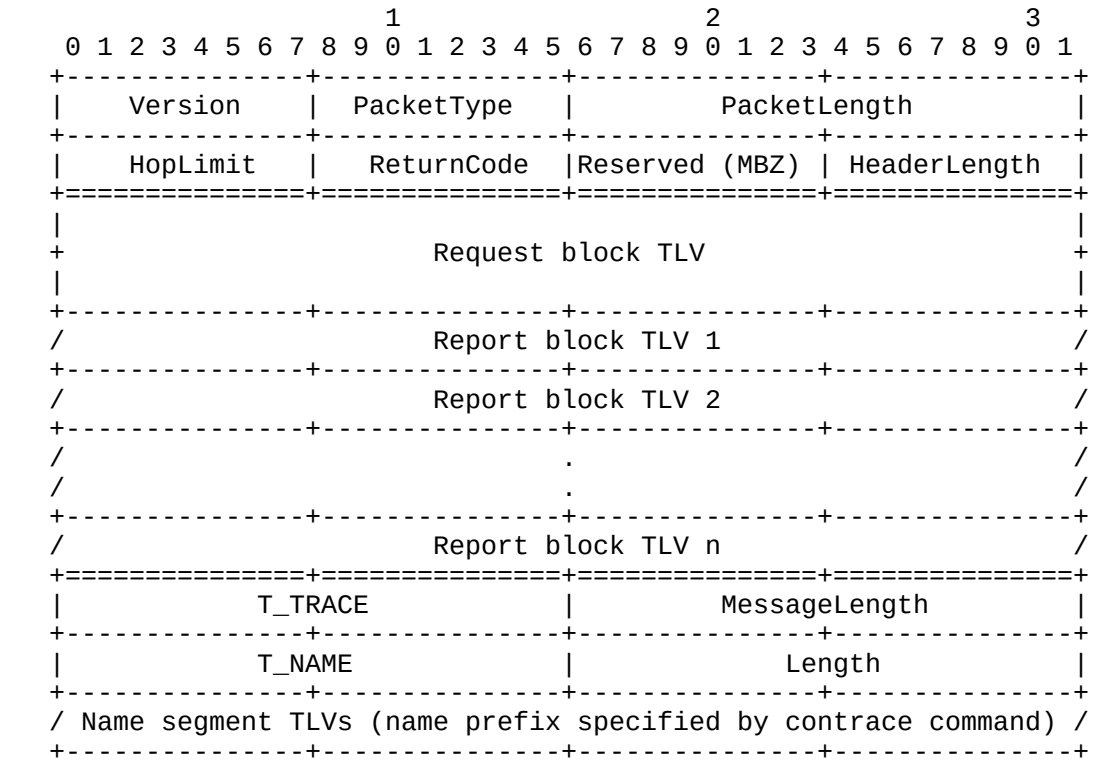

Figure 6: Request message consists of a fixed header, Request block TLV, Report block TLV(s), and Name TLV

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	INVALID_REQUEST	Invalid Contrace Request is received.
0x03	NO_ROUTE	This router has no route for the name prefix and no way to determine a potential route.
0x04	NO_INFO	This router has no cache information for the specified name prefix, device information, or function.
0x05	NO_SPACE	There was not enough room to insert another Report block in the packet.
0x06	NO_GATAWAY	Contrace Request arrived on a non-gateway router.
0x07	INFO_HIDDEN	Information is hidden from this trace because of some policy.
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 MUST insert the Request block TLV (Figure 7) and the Report block TLV (Figure 11) of its own to the Request message before sending it through the Incoming face(s).

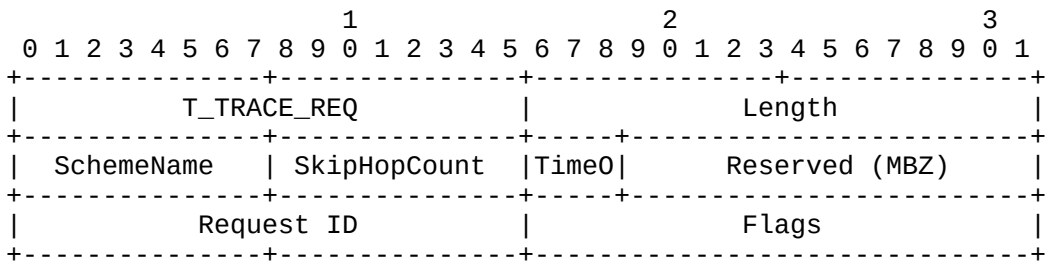

Figure 7: Request block TLV (hop-by-hop header)

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_REPORT
%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 Value field in octets. For the Request block, it MUST be 4 in the current specification.

SchemeName: 8 bits

Currently, the following scheme names are defined.

Code	Scheme name
0	ccnx:/
1	ndn:/
2	cefore:/
%x03-%FF	Not assigned

Figure 9: Scheme Names

SkipHopCount: 8 bits

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

TimeO: 3 bits

Timeout value (seconds). This Timeout value means a [Contrace Reply Timeout] value (seconds) requested by the Contrace user later described in Section 8.1. A Contrace user requests routers along the path to keep the PIT entry for the Request until this timeout value expires. Note that, because of some security concern (Section 10.5), a router along the path may configure the shorter timeout value than this requested timeout value. In that case, the Request may be timed out and the Contrace user may not receive the Reply as expected.

Request ID: 16 bits

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

Flags: 16 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 10. Note that code %x01 and %x02 are exclusive options; that is, only one of them should be turned on at once.

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	Force trace. Request to multiple upstream routers simultaneously (name_prefix, device_name, and function_name)
%x16	Discovery of gateway supporting specified scheme name (name_prefix, device_name, and function_name)
%x32	Function's or application's version number retrieval (function_name)
%x64-%x32768	Not assigned

Figure 10: 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 fixed header of the Request message until one of these routers is ready to send a Reply. In the Report block TLV (Figure 11), the Request Arrival Time and the Node Identifier MUST be inserted.

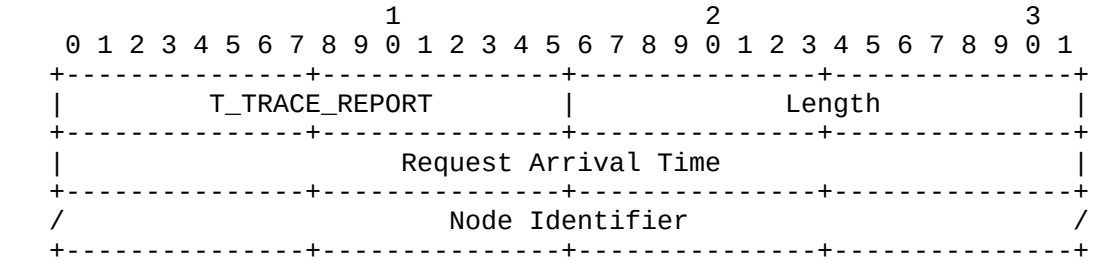

Figure 11: Report block TLV (hop-by-hop header)

Type: 16 bits

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

Length: 16 bits

Length of Value field in octets.

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:

```
request_arrival_time
= ((tv.tv_sec + 32384) << 16) + ((tv.tv_nsec << 7) / 1953125)
```

The constant 32384 is the number of seconds from Jan 1, 1900 to Jan 1, 1970 truncated to 16 bits. ((tv.tv_nsec << 7) / 1953125) is a reduction of ((tv.tv_nsec / 1000000000) << 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 all-zeros if unknown or unnumbered. Since we may not always rely on the IP addressing architecture, it would be necessary to define the identifier uniqueness (e.g., by specifying the protocol family) for this field. However, defining such uniqueness is out of scope of this document. Potentially, this field may be defined as a new TLV, which might be defined in the document for the CCNx TLV format[1].

3.2. Reply Message

When a content forwarder receives a Contrace Request message from the appropriate adjacent neighbor router, it would insert a Reply block TLV and Reply sub-block TLV(s) of its own to the Request message and turn the Request into the Reply by changing the Type field of the fixed header of the Request message from PT_PING_TRACE_REQ to PT_PING_TRACE_REPLY. The Reply message (see Figure 12) would then be forwarded back toward the Contrace user in a hop-by-hop manner.

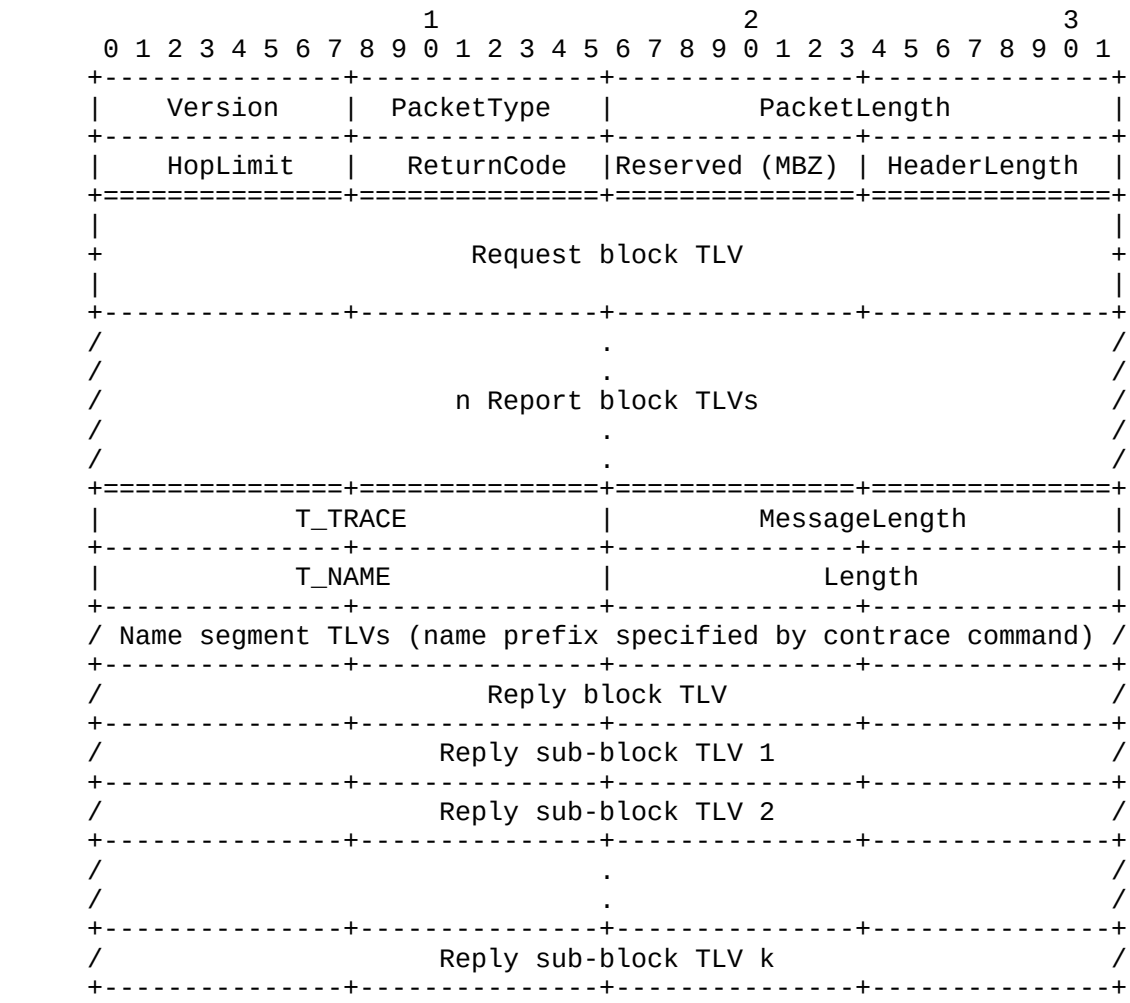

Figure 12: Reply message consists of a fixed header, Request block TLV, Report block TLV(s), Name TLV, and Reply block/sub-block TLV(s)

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]
8	T_TRACE_REPLY
9 - 12	Reserved [1]
%x0FFE	T_PAD [1]
%x0FFF	T_ORG [1]
%x1000-%x1FFF	Reserved [1]

Figure 13: CCNx Message Type Namespace

3.2.1. Reply Block

The Reply block TLV is an envelope for Reply sub-block TLV(s) (explained in Section 3.2.1.1).

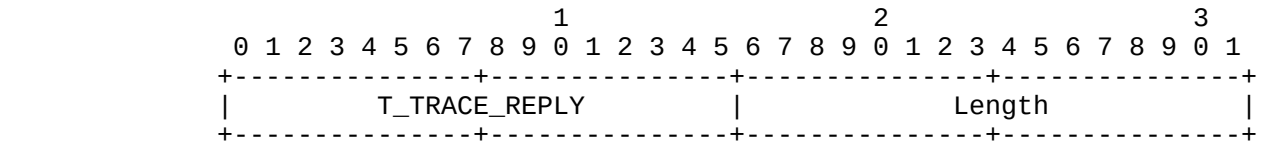

Figure 14: Reply block TLV (packet payload)

Type: 16 bits

Format of the Value field. For the Report block TLV, the type value MUST be T_TRACE_REPLY.

Length: 16 bits

Length of Value field in octets. This length is a total length of Reply sub-block(s).

3.2.1.1. Reply Sub-Block

In addition to the Reply block, a router on the traced path will add one or multiple Reply sub-blocks followed by the Reply block before sending the Reply to its neighbor router.

The Reply sub-block is flexible for various purposes. For instance, operators and developers may want to obtain various characteristics of content such as content's ownership and copyright, or other cache

states and conditions. Various information about device or function (or application) may be also retrieved by the variety of Reply sub-blocks. In this document, Reply sub-block TLVs for T_TRACE_CONTENT and T_TRACE_CONTENT_OWNER (Figure 15) and for T_TRACE_GATEWAY (Figure 16) are defined; other Reply sub-block TLVs will be defined in separate document(s).

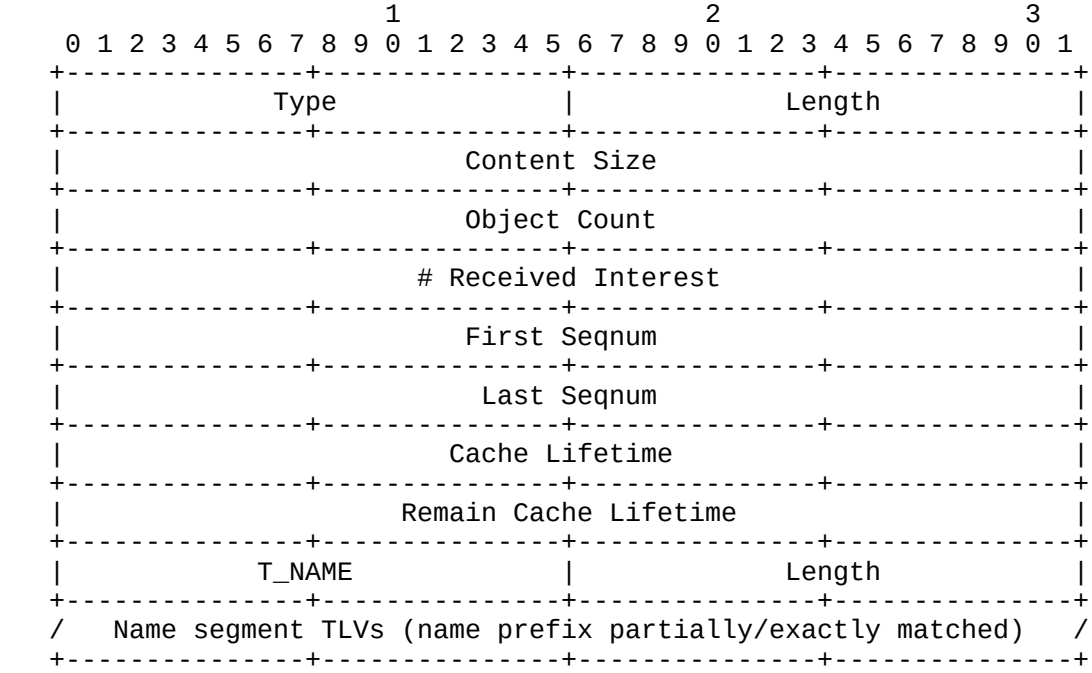

Figure 15: Reply sub-block TLV for T_TRACE_CONTENT and T_TRACE_CONTENT_OWNER (packet payload)

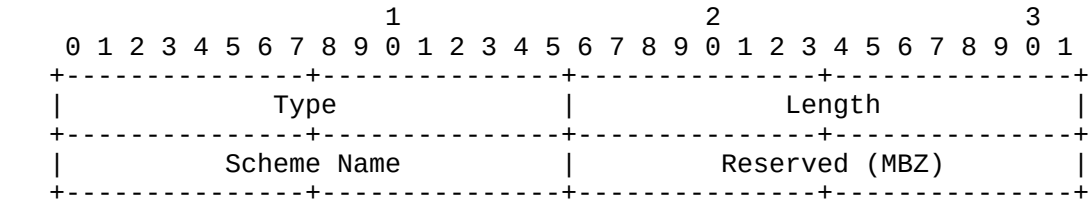

Figure 16: Reply sub-block TLV for T_TRACE_GATEWAY (packet payload)

Code	Type name
0	T_TRACE_CONTENT
1	T_TRACE_CONTENT_OWNER
2	T_TRACE_GATEWAY
3	T_TRACE_DEVICE
4	T_TRACE_FUNCTION
%x0FFF	T_ORG
%x1000-%x1FFF	Reserved (Experimental Use)

Figure 17: Contrace Reply Type Namespace

Type: 16 bits

Format of the Value field. For the Reply sub-block TLV, the type value MUST be one of the type value defined in the Contrace Reply Type Namespace (Figure 17). T_TRACE_CONTENT is specified when the cache information is replied from a caching router. T_TRACE_CONTENT_OWNER is specified when the content information is replied from a publisher. T_TRACE_GATEWAY is used to discover a gateway that has a FIB for the specified scheme name.

Length: 16 bits

Length of Value field in octets.

Scheme Name: 8 bits

The code of the scheme name defined in Figure 9.

Content Size: 32 bits

The total size (MB) of the (cached) content objects. Note that the maximum size expressed by 32 bit field is 65 GB.

Object Count: 32 bits

The number of the (cached) content objects.

Received Interest: 32 bits

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

First Seqnum: 32 bits

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

Last Seqnum: 32 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 insert the necessary values, the "Request ID" (in the Request block) and the "Node Identifier" (in the Report block), in the Request and Report blocks. Contrace user’s program MUST also record the Request ID at the corresponding PIT entry. The Request ID is a unique identifier for the Contrace Request.

After the Contrace user’s program sends the Request message, until the Reply times out, 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.

4.1.1. Gateway Discovery

A Contrace Request can be used for gateway discovery; if a Contrace user invokes a Contrace Request with a scheme name (e.g., ccnx:/ or ndn:/) and the "gateway discovery" flag value (i.e., "%x16" bit as seen in Figure 10), s/he could potentially discover a gateway that

supports different protocols such as CCN and NDN. The Contrace Request for gateway discovery only indicates the routing path information (see Section 4.1.2) and the scheme name whether the router is a gateway or not; it does not provide other information, e.g., cache information.

4.1.2. Routing Path Information

A Contrace user can send a Contrace Request for investigating routing path information for the specified named content. By the Request, the legitimate user can obtain; 1) identifiers (e.g., IP addresses) of intermediate routers, 2) identifier of content forwarder, 3) number of hops between content forwarder and consumer, and 4) RTT between content forwarder and consumer, per name prefix. This Contrace Request is terminated when it reaches the content forwarder. The contrace command enables user to obtain both the routing path information and in-network cache information (see below) in a same time.

4.1.3. In-Network Cache Information

A Contrace user can send a Contrace Request for investigating in-network cache information. By this Request, the legitimate user can obtain; 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, for Content Store (CS) in the content forwarder. This Contrace Request is terminated when it reaches the content forwarder.

4.2. Receiving Contrace Reply

A Contrace user’s program will receive one or multiple Contrace Reply messages from the adjacent neighbor router 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.

After the Contrace user has determined that s/he has traced the whole path or as much as s/he can expect to, s/he might collect statistics by waiting a timeout. Useful statistics provided by Contrace can be seen in Section 9.

5. Router Behavior

5.1. Receiving Contrace Request

5.1.1. Request Packet Verification

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 MUST be silently ignored. The router next examines the value of the "HopLimit" in the fixed header and the value of the "SkipHopCount" in the Request block (Figure 7). If SkipHopCount value is equal or more than the HopLimit value, the Request MUST be silently ignored.

5.1.2. Request Normal Processing

When a router receives a Contrace Request message, it performs the following steps.

1. HopLimit and SkipHopCount are counters that are decremented with each hop. The router terminates the Contrace Request when the HopLimit value becomes zero. Until the SkipHopCount value becomes zero, the router forwards the Contrace Request messages to the upstream router(s) (if it knows) without adding its own Report block and without replying the Request. If the router does not know the upstream router(s), without depending on the SkipHopCount value, it replies the Contrace Reply message with NO_ROUTE return code.

2. The router examines the Flags field of the Request block of received Contrace Request. If the flag value indicates "%x00" or "%x01" bit (as seen in Figure 10) for "cache information discovery", the router examines its FIB and CS. If the router caches the specified content, it inserts own Report block to the message and sends the Reply message with own Reply block and sub-block. If the router does not cache the specified content but knows the neighbor router(s) for the specified name prefix, it inserts own Report block and forwards the Request to the upstream neighbor(s). If the router does not cache the specified content and does not know the upstream neighbor router(s) for the specified name prefix, it replies the Contrace Reply message with NO_ROUTE return code.

3. If the flag value indicates "%x02" bit for "routing path information discovery", the router examines its FIB and CS. If the router caches the specified content, it inserts own Report block to the message and sends the Reply message with own Reply block. The router does not insert any Reply sub-block here. If

the router does not cache the specified content but knows the neighbor router(s) for the specified name prefix, it inserts own Report block and forwards the Request to the upstream neighbor(s). If the router does not cache the specified content and does not know the upstream neighbor router(s) for the specified name prefix, it replies the Contrace Reply message with NO_ROUTE return code.

4. If the flag value indicates "%x04" bit for "publisher discovery", the node receiving the Request message examines whether it owns the requested content as the publisher. If it is the publisher, it sends the Reply message with own Report block and sub-block. If the node is not the publisher but know the upstream neighbor router(s) for the specified name prefix, it adds the own Report block and forwards the Request to the neighbor(s). If the node is not the publisher and does not know the upstream neighbor router(s) for the specified name prefix, it replies the Contrace Reply message with NO_ROUTE return code.

5. When a router receives a Contrace Request in which the "gateway discovery" flag (i.e., "%x16") is set in the Flags field and a scheme name is specified, the router examines whether it has the FIB for the specified scheme name and the connections with the neighbor router(s) for the scheme protocol. If the router is the gateway, it sends the Reply message back toward the Contrace user. If the router does not have the FIB for the specified scheme name or does not connect to any neighbor router for the specified scheme name, the router returns the Reply with NO_GATEWAY return code.

5.2. Forwarding Contrace Request

When a router decides to forward a Request message with its Report block to its upstream router(s), it specifies the Request Arrival Time and Node Identifier in the Report block of the Request message. The router then forwards the Request message upstream toward the publisher or caching router based on the FIB entry.

When the router forwards the Request message, it MUST record the Request ID at the corresponding PIT entry. The router can later decide the PIT entry to correctly forward back the Reply message even if it receives multiple Reply messages within the same timeout period. (See below.)

Contrace supports multipath forwarding. The Request messages can be forwarded to multiple neighbor routers. Some router may have strategy for multipath forwarding; when it sends Interest messages to multiple neighbor routers, it may delay or prioritize to send the

message to the upstream routers. The Contrace Request, as the default, complies with such strategy; a Contrace user could trace the actual forwarding path based on the strategy. On the other hand, there may be the case that a Contrace user wants to discover all potential forwarding paths based on routers' FIBs. If a Contrace user invokes a Contrace Request with the force flag value (i.e., "%x08" bit as seen in Figure 10), the forwarding strategy will be ignored and the router sends Requests to multiple upstream routers simultaneously, and the Contrace user could trace the all potential forwarding paths.

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 configured Contrace Reply Timeout (Section 8.1) passes. 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 a router decides to send a Reply message to its downstream neighbor router or the Contrace user with NO_ERROR return code, it inserts a Report block having the Request Arrival Time and Node Identifier to the hop-by-hop TLV header of the Request message. And then the router inserts the corresponding Reply block and Reply sub-block to the payload. The router does not insert any Reply block/ sub-block if there is an error. The router finally changes the Type field in the fixed header from PT_PING_TRACE_REQ to PT_PING_TRACE_REPLY and forwards the message back as the Reply toward the Contrace user in a hop-by-hop manner.

When a router decides to send the Reply message for the Request for the cache or routing path information discovery, it forms the Reply message including a Reply block and a Reply sub-block with the T_TRACE_CONTENT type value (Figure 15) and various cache information. After the router puts the NO_ERROR return code in the fixed header, it sends the Reply back toward the Contrace user.

When a router decides to send the Reply message for the Request for the publisher discovery, it forms the Reply message including a Reply block and a Reply sub-block with the T_TRACE_CONTENT_OWNER type value (Figure 15) and various cache information. After the router puts the

NO_ERROR return code in the fixed header, it sends the Reply back toward the Contrace user.

When a router decides to send the Reply message for the Request for the gateway discovery, it forms the Reply message including a Reply block and a Reply sub-block with the T_TRACE_GATEWAY type value (Figure 16) and the scheme name (Figure 9). After the router puts the NO_ERROR return code in the fixed header, it sends the Reply back toward the Contrace user.

If a router cannot continue the Request, it MUST put an appropriate ReturnCode in the Request message, change the Type field value in the fixed header from PT_PING_TRACE_REQ to PT_PING_TRACE_REPLY, and forward the Reply message back toward the Contrace user, to terminate the request. See Section 7.

5.4. Forwarding Contrace Reply

When a 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 will note an appropriate ReturnCode in the Request message, change the Type field value in the fixed header from PT_PING_TRACE_REQ to PT_PING_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 having a Reply block and Reply sub-block 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 or Gateway

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

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 Route

If the router cannot determine the routing paths or neighbor routers for the specified name prefix, device name, or function within the specified HopLimit, 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.

7.4. No Information

If the router does not have any information about the specified name prefix, device name, or function within the specified HopLimit, the router MUST note a ReturnCode of NO_INFO in the fixed header of the message, and forwards the message as the Reply back to the Contrace user.

7.5. 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.

7.6. 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).

7.7. 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 7.8.

7.8. 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.

7.9. Administratively Prohibited

If Contrace is administratively prohibited, a router or a publisher rejects the Request message, and the router or the publisher, or 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. Both Contrace users and routers can configure their own Contrace Reply Timeout values. Contrace users, for example, can configure the timeout value by the contrace command. The default [Contrace Reply Timeout] value is 4 (seconds). Routers may want to configure the short timeout values because of some security concern, e.g., Section 10.5. However, the [Contrace Reply Timeout] value SHOULD NOT be larger than 6 (seconds) and SHOULD NOT be lower than 3 (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 name prefix, such as "who can retrieve which name prefix". See Section 10.2.

9. Diagnosis and Analysis

9.1. Number of Hops

A Contrace Request message is forwarded in a hop-by-hop manner and each forwarding router appended its own Report block. We can then verify the number of hops to reach the content forwarder or the publisher.

9.2. Caching Router and Gateway Identification

It is possible to identify the caching routers or a gateway in the path from the Contrace user to the content forwarder, while some routers may hide their identifier (with all-zeros) in the Report blocks (Section 10.1).

9.3. TTL or Hop Limit

By taking the HopLimit from the content forwarder and forwarding TTL threshold over all hops, it is possible to discover the TTL or hop limit required for the content forwarder to reach the Contrace user.

9.4. Time Delay

If the routers have synchronized clocks, it is possible to estimate propagation and queuing delay from the differences between the timestamps at successive hops. However, this delay includes control processing overhead, so is not necessarily indicative of the delay that data traffic would experience.

9.5. Path Stretch

By getting the path stretch "d / P", where "d" is the hop count of the data and "P" is the hop count from the consumer to the publisher, we can measure the improvement in path stretch in various cases, such as different caching and routing algorithms. We can then facilitate investigation of the performance of the protocol.

9.6. Cache Hit Probability

Contrace can show the number of received interests per cache or chunk on a router. By this, Contrace measures the content popularity (i.e., the number of accesses for each content/cache), and you can investigate the routing/caching strategy in networks.

10. Security Considerations

This section addresses some of the security considerations.

10.1. 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" and/or "what kind of information" can be defined for each router. For the former access policy, routers having the specified content can examine the signature enclosed in the Request message and decide whether they should notify the content information in the Reply or not. If the routers decide to not notify the content information, they reply the Contrace Reply with the ReturnCode of ADMIN_PROHIB without appending any Reply (sub-)block TLV. For the latter policy, the permission, whether (1) All (all cache information is disclosed), (2) Partial (cache information with the particular name prefix can (or cannot) be disclosed), or (3) Deny (no cache information is disclosed), is defined at routers.

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.

10.2. 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.

10.3. 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.

10.4. 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.

10.5. Longer or Shorter Contrace Reply Timeout

Routers can configure the Contrace Reply Timeout (Section 8.1), which is the allowable timeout value to keep the PIT entry. If routers configure the longer timeout value, there may be an attractive attack vector against PIT memory. Moreover, especially when the force option (Section 5.2) is specified for the Contrace Request, a number of Reply messages may come back and cause a response storm. (See Section 10.7 for rate limiting to avoid the storm). In order to avoid DoS attacks, routers may configure the shorter timeout value than the user-configured Contrace timeout value. However, if it is too short, the Request may be timed out and the Contrace user does not receive the all Replies and only retrieves the partial path information (i.e., information about part of the tree).

There may be the way to allow for incremental exploration (i.e., to explore the part of the tree the previous operation did not explore), whereas discussing such mechanism is out of scope of this document.

10.6. 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.

10.7. 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.

10.8. Adjacency Verification

Contrace Request and Reply messages MUST be forwarded by adjacent neighbor nodes or routers. Forwarding Contrace messages given from non-adjacent neighbor nodes/routers MUST be prohibited. Such invalid messages SHOULD be silently discarded. Note that defining the secure way to verify the adjacency cannot rely on the way specified in CCNx message format or semantics. An adjacency verification mechanism and the corresponding TLV for adjacency verification using hop-by-hop TLV header will be defined in a separate document.

11. Acknowledgements

The authors would like to thank Spyridon Mastorakis, Ilya Moiseenko, and David Oran for their valuable comments and suggestions on this document.

12. References

12.1. Normative References

[1] Mosko, M., Solis, I., and C. Wood, "CCNx Messages in TLV Format", draft-irtf-icnrg-ccnxmessages-04 (work in progress), March 2017.

[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.

12.2. 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-v2-17 (work in progress), March 2017.

Appendix A. Contrace Command and Options

The contrace command enables the Contrace user to investigate the routing 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] [-g] [-f] [-n] [-o] [-r hop_count] [-s hop_count] [-w wait_time] name_prefix; or,

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

name_prefix
: Name prefix of the content (e.g., ccnx:/news/today) the Contrace user wants to trace. If the Contrace user specifies only a scheme name, e.g., "ccnx:/", s/he must specify "-g" option (i.e., contrace -g ccnx:/). In that case, the Contrace user discovers the router having the FIB of the specified scheme name and the RTT between Contrace user and the router. 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].

g option
: This option enables to discover a gateway that supports specified scheme name and has multiple FIBs. When a Contrace user specifies only a scheme name, e.g., "ccnx:/", this option must be specified and other content name prefix is ignored.

f option
This option enables to ignore the forwarding strategy and send Contrace Requests to multiple upstream routers simultaneously. The Contrace user could then trace the all potential forwarding paths.

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 enables 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 inserts each Report block but 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. Specifying only a scheme name is not allowed with this option.

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", only three routers along the path examine their path and cache information. 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 10, 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

ICNRG
Internet-Draft
Intended status: Informational
Expires: January 4, 2018

J. Seedorf
HFT Stuttgart - Univ. of Applied Sciences
M. Arumaithurai
University of Goettingen
A. Tagami
KDDI Research Inc.
K. Ramakrishnan
University of California
N. Blefari Melazzi
University Tor Vergata

July 3, 2017

Research Directions for Using ICN in Disaster Scenarios
draft-irtf-icnrg-disaster-02

Abstract

Information Centric Networking (ICN) is a new paradigm where the network provides users with named content, instead of communication channels between hosts. This document outlines some research directions for Information Centric Networking with respect to applying ICN approaches for coping with natural or human-generated, large-scale disasters.

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 http://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 January 4, 2018.

Table of Contents

1. Introduction

This document summarizes some research challenges for coping with natural or human-generated, large-scale disasters. In particular, the document discusses potential directions for applying Information Centric Networking (ICN) to address these challenges.

There are existing research approaches (for instance, see further the discussions in the IETF DTN Research Group [dtnrg]) and an IETF specification [RFC5050] for disruptant tolerant networking, which is a key necessity for communicating in the disaster scenarios we are considering in this document (see further Section 3.1). 'Disconnection tolerance' can thus be achieved with these existing DTN approaches. However, while these approaches can provide independence from an existing communication infrastructure (which indeed may not work anymore after a disaster has happened), ICN offers as key concepts suitable naming schemes and multicast communication which together enable many key (publish/subribe-based) use cases for communication after a disaster (e.g. message prioritisation, one-to-many delivery of important messages, or group communication among rescue teams, see further Section 4). One could

add such features to existing DTN protocols and solutions; however, in this document we explore the use of ICN as starting point for building a communication architecture that works well before and after a disaster. We discuss the relationship between the ICN approaches (for enabling communication after a disaster) discussed in this document with existing work from the DTN community in more depth in Section 3.3 .

'Emergency Support and Disaster Recovery' is also listed among the ICN Baseline Scenarios in [RFC7476] as a potential scenario that 'can be used as a base for the evaluation of different information-centric networking (ICN) approaches so that they can be tested and compared against each other while showcasing their own advantages' [RFC7476] . In this regard, this document complements [RFC7476] by investigating the use of ICN approaches for 'Emergency Support and Disaster Recovery' in depth and discussing the relationship to existing work in the DTN community.

Section 2 gives some examples of what can be considered a large-scale disaster and what the effects of such disasters on communication networks are. Section 3 outlines why ICN can be beneficial in such scenarios and provides a high-level overview on corresponding research challenges. Section 4 describes some concrete use cases and requirements for disaster scenarios. In Section 5 , some concrete ICN-based solutions approaches are outlined.

2. Disaster Scenarios

An enormous earthquake hit Northeastern Japan (Tohoku areas) on March 11, 2011, and caused extensive damages including blackouts, fires, tsunamis and a nuclear crisis. The lack of information and means of communication caused the isolation of several Japanese cities. This impacted the safety and well-being of residents, and affected rescue work, evacuation activities, and the supply chain for food and other essential items. Even in the Tokyo area that is 300km away from the Tohoku area, more than 100,000 people became 'returner' refugees, who could not reach their homes because they had no means of public transportation (the Japanese government has estimated that more than 6.5 million people would become returner refugees if such a catastrophic disaster were to hit the Tokyo area).

That earthquake in Japan also showed that the current network is vulnerable against disasters. Mobile phones have become the lifelines for communication including safety confirmation: Besides (emergency) phone calls, services in mobile networks commonly being used after a disaster include network disaster SMS notifications (or SMS 'Cell Broadcast' [cellbroadcast]), available in most cellular networks. The aftermath of a disaster puts a high strain on

available resources due to the need for communication by everyone. Authorities such as the President/Prime-Minister, local authorities, Police, fire brigades, and rescue and medical personnel would like to inform the citizens of possible shelters, food, or even of impending danger. Relatives would like to communicate with each other and be informed about their wellbeing. Affected citizens would like to make enquiries of food distribution centres, shelters or report trapped, missing people to the authorities. Moreover, damage to communication equipment, in addition to the already existing heavy demand for communication highlights the issue of fault-tolerance and energy efficiency.

Additionally, disasters caused by humans such as a terrorist attack may need to be considered, i.e. disasters that are caused deliberately and willfully and have the element of human intent. In such cases, the perpetrators could be actively harming the network by launching a Denial-of-Service attack or by monitoring the network passively to obtain information exchanged, even after the main disaster itself has taken place. Unlike some natural disasters that are predictable using weather forecasting technologies and have a slower onset and occur in known geographical regions and seasons, terrorist attacks may occur suddenly without any advance warning. Nevertheless, there exist many commonalities between natural and human-induced disasters, particularly relating to response and recovery, communication, search and rescue, and coordination of volunteers.

The timely dissemination of information generated and requested by all the affected parties during and the immediate aftermath of a disaster is difficult to provide within the current context of global information aggregators (such as Google, Yahoo, Bing etc.) that need to index the vast amounts of specialized information related to the disaster. Specialized coverage of the situation and timely dissemination are key to successfully managing disaster situations. We believe that network infrastructure capability provided by Information Centric Networks can be suitable, in conjunction with application and middleware assistance.

3. Research Challenges and Benefits of ICN

3.1. High-Level Research Challenges

Given a disaster scenario as described in Section 2 , on a high-level one can derive the following (incomplete) list of corresponding technical challenges:

- Enabling usage of functional parts of the infrastructure, even when these are disconnected from the rest of the network: Assuming

that parts of the network infrastructure (i.e. cables/links, routers, mobile bases stations, ...) are functional after a disaster has taken place, it is desirable to be able to continue using such components for communication as much as possible. This is challenging when these components are disconnected from the backhaul, thus forming fragmented networks. This is especially true for today's mobile networks which are comprised of a centralised architecture, mandating connectivity to central entities (which are located in the core of the mobile network) for communication. But also in fixed networks, access to a name resolution service is often necessary to access some given content.

- o Decentralised authentication and trust: In mobile networks, users are authenticated via central entities. While special services important in a disaster scenario exist and may work without authentication (such as SMS 'Cell Broadcast' [cellbroadcast] or emergency calls), user-to-user (or user-to-authorities) communication is normally not possible without being authenticated via a central entity in the network. In order to communicate in fragmented or disconnected parts of a mobile network, hence the challenge of decentralising user authentication arises. Independently of the network being fixed or mobile, data origin authentication of content retrieved from the network is challenging when being 'offline' (e.g. disconnected from servers of a security infrastructure such as a PKI). As the network suddenly becomes fragmented or partitioned, trust models may shift accordingly to the change in authentication infrastructure being used (e.g., one may switch from a PKI to a web-of-trust model such as PGP).

- o Delivering/obtaining information and traffic prioritization in congested networks: Due to broken cables, failed routers, etc., it is likely that in a disaster scenario the communication network has much less overall capacity for handling traffic. Thus, significant congestion can be expected in parts of the infrastructure. It is therefore a challenge to guarantee message delivery in such a scenario. This is even more important as in the case of a disaster aftermath, it may be crucial to deliver certain information to recipients (e.g. warnings to citizens) with higher priority than other content.

- o Delay/Disruption Tolerant Approach: Fragmented networks makes it difficult to support end-to-end communication. However, communication in general and especially during disaster can tolerate some form of delay. E.g. in order to know if his/her relatives are safe or a 'SOS' call need not be supported in an

end-to-end manner. It is sufficient to improve communication resilience in order to deliver such important messages.

- Energy Efficiency: Long-lasting power outages may lead to batteries of communication devices running out, so designing energy-efficient solutions is very important in order to maintain a usable communication infrastructure.

- Contextuality: Like any communication in general, disaster scenarios are inherently contextual. Aspects of geography, the people affected, the rescue communities involved, the languages being used and many other contextual aspects are highly relevant for an efficient realization of any rescue effort and, with it, the realization of the required communication.

The list above is most likely incomplete; future revisions of this document intend to add additional challenges to the list.

3.2. How ICN can be Beneficial

Several aspects of ICN make related approaches attractive candidates for addressing the challenges described in Section 3.1 . Below is an (incomplete) list of considerations why ICN approaches can be beneficial to address these challenges:

- Routing-by-name: ICN protocols natively route by named data objects and can identify objects by names, effectively moving the process of name resolution from the application layer to the network layer. This functionality is very handy in a fragmented network where reference to location-based, fixed addresses may not work as a consequence of disruptions. For instance, name resolution with ICN does not necessarily rely on the reachability of application-layer servers (e.g. DNS resolvers). In highly decentralised scenarios (e.g. in infrastructureless, opportunistic environments) the ICN routing-by-name paradigm effectively may lead to a ’replication-by-name’ approach, where content is replicated depending on its name.

- Authentication of named data objects: ICN is built around the concept of named data objects. Several proposals exist for integrating the concept of ’self-certifying data’ into a naming scheme (see e.g. [RFC6920]). With such approaches, the origin of data retrieved from the network can be authenticated without relying on a trusted third party or PKI.

- Content-based access control: ICN promotes a data-centric communication model which is better suited to content-based security (e.g. allowing access to content only to a specific user

or class of users); this functionality could facilitate trusted communications among peer users in isolated areas of the network.

- Caching: Caching content along a delivery path is an inherent concept in ICN. Caching helps in handling huge amounts of traffic, and can help to avoid congestion in the network (e.g. congestion in backhaul links can be avoided by delivering content from caches at access nodes).

- Sessionless: ICN does not require full end-to-end connectivity. This feature facilitates a seemless aggregation between a normal network and a fragmented network, which needs DTN-like message forwarding.

- Potential to run traditional IP-based services (IP-over-ICN): While ICN and DTN promote the development of novel applications that fully utilize the new capabiliticbies of the ICN/DTN network, work in [Trossen2015] has shown that an ICN-enabled network can transport IP-based services, either directly at IP or even at HTTP level. With this, IP- and ICN/DTN-based services can coexist, providing the necessary support of legacy applications to affected users, while reaping any benefits from the native support for ICN in future applications.

- Opportunities for traffic engineering and traffic prioritization: ICN provides the possibility to perform traffic engineering based on the name of desired content. This enables priority based replication depending on the scope of a given message [Psaras2014] . In addition, as [Trossen2015] , among others, have pointed out, the realization of ICN services and particularly of IP-based services on top of ICN provide further traffic engineering opportunities. The latter not only relate to the utilization of cached content, as outlined before, but to the ability to flexbily adapt to route changes (important in unreliable infrastructure such as in disaster scenarios), mobility support without anchor points (again, important when parts of the infrastructure are likely to fail) and the inherent support for multicast and multihoming delivery.

3.3. ICN as Starting Point vs. Existing DTN Solutions

There has been quite some work in the DTN (Delay Tolerant Networking) community on disaster communication (for instance, see further the discussions in the IETF DTN Research Group [dtnrg]). However, most DTN work lacks important features such as publish/subscribe (pub/sub) capabilities, caching, multicast delivery, and message prioritisation based on content types, which are needed in the disaster scenarios we consider. One could add such features to existing DTN protocols and

solutions, and indeed individual proposals for adding such features to DTN protocols have been made (e.g. [Greifenberg2008] [Yoneki2007] propose the use of a pub/sub-based multicast distribution infrastructure for DTN-based opportunistic networking environments).

However, arguably ICN---having these intrinsic properties (as also outlined above)---makes a better starting point for building a communication architecture that works well before and after a disaster. For a disaster-enhanced ICN system this would imply the following advantages: a) ICN data mules would have built-in caches and can thus return content for interests straight on, b) requests do not necessarily need to be routed to a source (as with existing DTN protocols), instead any data mule or end-user can in principle respond to an interest, c) built-in multi-cast delivery implies energy-efficient large-scale spreading of important information which is crucial in disaster scenarios, and d) pub/sub extension for popular ICN implementations exist [COPSS2011] which are very suitable for efficient group communication in disasters and provide better reliability, timeliness and scalability as compared to existing pub/sub approaches in DTN [Greifenberg2008] [Yoneki2007] .

Finally, most DTN routing algorithms have been solely designed for particular DTN scenarios. By extending ICN approaches for DTN-like scenarios, one ensures that a solution works in regular (i.e. well-connected) settings just as well (which can be important in reality, where a routing algorithm should work before and after a disaster). It is thus reasonable to start with existing ICN approaches and extend them with the necessary features needed in disaster scenarios.

4. Use Cases and Requirements

This Section describes some use cases for the aforementioned disaster scenario (as outlined in Section 2) and discusses the corresponding technical requirements for enabling these use cases.

- Delivering Messages to Relatives/Friends: After a disaster strikes, citizens want to confirm to each other that they are safe. For instance, shortly after a large disaster (e.g., Earthquake, Tornado), people have moved to different refugee shelters. The mobile network is not fully recovered and is fragmented, but some base stations are functional. This use case imposes the following high-level requirements: a) People must be able to communicate with others in the same network fragment, b) people must be able to communicate with others that are located in different fragmented parts of the overall network. More concretely, the following requirements are needed to enable the use case: a) a mechanism for scalable message forwarding scheme that dynamically adapts to changing conditions in disconnected

networks, b) DTN-like mechanisms for getting information from disconnected island to another disconnected island, c) data origin authentication so that users can confirm that the messages they receive are indeed from their relatives or friends, and d) the support for contextual caching in order to provide the right information to the right set of affected people in the most efficient manner.

- o Spreading Crucial Information to Citizens: State authorities want to be able to convey important information (e.g. warnings, or information on where to go or how to behave) to citizens. These kinds of information shall reach as many citizens as possible. i.e. Crucial content from legal authorities shall potentially reach all users in time. The technical requirements that can be derived from this use case are: a) Data origin authentication, such that citizens can confrim the authenticity of messages sent by authorities, b) mechanisms that guarantee the timeliness and loss-free delivery of such information, which may include techniques for prioritizing certain messages in the network depending on who sent them, and c) DTN-like mechanisms for getting information from disconnected island to another disconnected island.

It can be observed that different key use cases for disaster scenarios imply overlapping and similar technical requirements for fulfilling them. As discussed in Section 3.2 , ICN approaches are envisioned to be very suitable for addressing these requirements with actual technical solutions. In [Robitzsch2015] , a more elaborate set of requirements is provided that addresses, among disaster scenarios, a communication infrastructure for communities facing several geographic, economic and political challenges.

5. Solution Design

This section outlines some ICN-based approaches that aim at fulfilling the previously mentioned use cases and requirements. Overall, the focus is on delivery of messages and not real-time communication. While most probably users would like to conduct real-time voice/video calls after a disaster, in the extreme scenario we consider (with users being scattered over different fragmented networks, see Section 2), somewhat delayed message delivery appears to be inevitable, and full-duplex real-time communication seems infeasible to achieve. Thus, the assumption is that - for a certain amount of time at least (i.e. the initial period until the regular communication infrastructure has been repaired) - users would need to live with message delivery and publish/subscribe services but without real-time communication. Note, however, that a) in principle ICN can

support VoIP calls, and b) message delivery includes voice messages (e.g. whatsapp voice messages).

- ICN 'data mules': To facilitate the exchange of messages between different network fragments, mobile entitites can act as ICN 'data mules' which are equipped with storage space and move around the disaster-stricken area gathering information to be disseminated. As the mules move around, they deliver messages to other individuals or points of attachment to different fragments of the network. These 'data mules' could have a pre-determined path (an ambulance going to and from a hospital), a fixed path (drone/robot assigned specifically to do so) or a completely random path (doctors moving from one camp to another). An example of a many-to-many communication service for fragmented networks based on ICN data mules has been proposed in [Tagami2016].

- Priority-dependent or popularity-dependent name-based replication: By allowing spatial and temporal scoping of named messages, priority based replication depending on the scope of a given message is possible. Clearly, spreading information in disaster cases involves space and time factors that have to be taken into account as messages spread. A concrete approach for such scope-based prioritisation of ICN messages in disasters, called 'NREP', has been proposed [Psaras2014] , where ICN messages have attributes such as user-defined priority, space, and temporal-validity. These attributes are then taken into account when prioritizing messages. In [Psaras2014] , evaluations show how this approach can be applied to the use case 'Delivering Messages to Relatives/Friends' decribed in Section 4. In [Seedorf2016], a scheme is presented that enables to estimate the popularity of ICN interest messages in a completely decentralized manner among data mules in a scenario with random, unpredictable movements of ICN data mules. The approach exploits the use of nonces associated with end user requests, common in most ICN architectures. It enables for a given ICN data mule to estimate the overall popularity (among end-users) of a given ICN interest message. This enables data mules to optimize content dissemination with limited caching capabilities by prioritizing interests based on their popularity.

- Information Resilience through Decentralised Forwarding: In a dynamic or disruptive environment, such as the aftermath of a disaster, both users and content servers may dynamically join and leave the network (due to mobility or network fragmentation). Thus, users might attach to the network and request content when the network is fragmented and the corresponding content origin is not reachable. In order to increase information resilience, content cached both in in-network caches and in end-user devices

should be exploited. A concrete approach for the exploitation of content cached in user devices is presented in [Sourlas2015] . The proposal in [Sourlas2015] includes enhancements to the NDN router design, as well as an alternative Interest forwarding scheme which enables users to retrieve cached content when the network is fragmented and the content origin is not reachable. Evaluations show that this approach is a valid tool for the retrieval of cached content in disruptive cases and can be applied to tackle the challenges presented in Section 3.1 .

- Energy Efficiency: A large-scale disaster causes a large-scale blackout and thus a number of base stations (BSs) will be operated by their batteries. Capacities of such batteries are not large enough to provide cellular communication for several days after the disaster. In order to prolong the batteries' life from one day to several days, different techniques need to be explored: Priority control, cell-zooming, and collaborative upload. Cell zooming switches-off some of the BSs because switching-off is the only way to reduce power consumed at the idle time. In cell zooming, areas covered by such inactive BSs are covered by the active BSs. Collaborative communication is complementary to cell zooming and reduces power proportional to a load of a BS. The load represents cellular frequency resources. In collaborative communication, end-devices delegate sending and receiving messages to and from a base station to a representative end-device of which radio propagation quality is better. The design of an ICN-based publish/subscribe protocol that incorporates collaborative upload is ongoing work. In particular, the integration of collaborative upload techniques into the COPSS (Content Oriented Publish/ Subscribe System)} framework is envisioned [COPSS2011] .

- Data-centric confidentiality and access control: In ICN, the requested content is not anymore associated to a trusted server or an endpoint location, but it can be retrieved from any network cache or a replica server. This call for 'data-centric' security, where security relies on information exclusively contained in the message itself, or, if extra information provided by trusted entities is needed, this should be gathered through offline, asynchronous, and non interactive communication, rather than from an explicit online interactive handshake with trusted servers. The ability to guarantee security without any online entities is particularly important in disaster scenarios with fragmented networks. One concrete cryptographic technique is 'Ciphertext-Policy Attribute Based Encryption' (CP-ABE), allowing a party to encrypt a content specifying a policy, which consists in a Boolean expression over attributes, that must be satisfied by those who want to decrypt such content. Such encryption schemes tie confidentiality and access-control to the transferred data, which

can be transmitted also in an unsecured channel, enabling the source to specify the set of nodes allowed to decrypt.

- o Decentralised authentication of messages: Self-certifying names provide the property that any entity in a distributed system can verify the binding between a corresponding public key and the self-certifying name without relying on a trusted third party. Self-certifying names thus provide a decentralized form of data origin authentication. However, self-certifying names lack a binding with a corresponding real-world identity. Given the decentralised nature of a disaster scenario, a PKI-based approach for binding self-certifying names with real-world identities is not feasible. Instead, a Web-of-Trust can be used to provide this binding. Not only are the cryptograohic signatures used within a Web-of-Trust independent of any central authority; there are also technical means for making the inherent trust relationships of a Web-of-Trust available to network entities in a decentralised, 'offline' fashion, such that information received can be assessed based on these trust relationships. A concrete scheme for such an approach has been published in [Seedorf2014] , where also concrete examples for fulfilling the use case 'Delivering Messages to Relatives/Friends' with this approach are given.

6. Conclusion

This document has outlined some research directions for Information Centric Networking (ICN) with respect to applying ICN approaches for coping with natural or human-generated, large-scale disasters. The document has described high-level research challenges for enabling communication after a disaster has happened as well as a general rationale why ICN approaches could be beneficial to address these challenges. Further, concrete use cases have been described and how these can be addressed with ICN-based approaches has been discussed.

Finally, the document provided an overview of examples of existing ICN-based solutions that address the previously outlined research challenges. These concrete solutions demonstrate that indeed the communication challenges in the aftermath of a disaster can be addressed with techniques that have ICN paradigms at their base, validating our overall reasoning. However, further, more detailed challenges exist and more research is necessary in all areas we discussed: efficient content distribution and routing in fragmented networks, traffic prioritization, security, and energy-efficiency.

In order to deploy ICN-based solutions for disaster-aftermath communication in actual mobile networks, standardized ICN baseline protocols are a must: It is unlikely to expect all user equipment in a large-scale mobile network to be from the same vendor. In this

respect, the work being done in the IRTF ICNRG is very useful as it works towards standards for concrete ICN protocols that enable interopability among solutions from different vendors. These protocols - currently being standardized in the IRTF INCRG - provide a good foundation for deploying ICN-based disaster-aftermath communication and thereby addressing key use cases that arise in such situations (as outlined in this document).

7. References

7.1. Normative References

[RFC5050] Scott, K. and S. Burleigh, "Bundle Protocol Specification", RFC 5050, DOI 10.17487/RFC5050, November 2007, <http://www.rfc-editor.org/info/rfc5050>.

[RFC6920] Farrell, S., Kutscher, D., Dannewitz, C., Ohlman, B., Keranen, A., and P. Hallam-Baker, "Naming Things with Hashes", RFC 6920, DOI 10.17487/RFC6920, April 2013, <http://www.rfc-editor.org/info/rfc6920>.

[RFC7476] Pentikousis, K., Ed., Ohlman, B., Corujo, D., Boggia, G., Tyson, G., Davies, E., Molinaro, A., and S. Eum, "Information-Centric Networking: Baseline Scenarios", RFC 7476, DOI 10.17487/RFC7476, March 2015, <http://www.rfc-editor.org/info/rfc7476>.

7.2. Informative References

[cellbroadcast] Wikipedia, "Cell Broadcast - Wikipedia, https://en.wikipedia.org/wiki/Cell_Broadcast", (online).

[COPSS2011] Chen, J., Arumaithurai, M., Jiao, L., Fu, X., and K. Ramakrishnan, "COPSS: An Efficient Content Oriented Publish/Subscribe System", Seventh ACM/IEEE Symposium on Architectures for Networking and Communications Systems (ANCS), 2011.

[dtnrg] Fall, K. and J. Ott, "Delay-Tolerant Networking Research Group - DTNRG", https://irtf.org/dtnrg.

[Greifenberg2008] Greifenberg, J. and D. Kutscher, "Efficient publish/ subscribe-based multicast for opportunistic networking with self-organized resource utilization", Advanced Information Networking and Applications-Workshops, 2008.

[Psaras2014]
Psaras, I., Saino, L., Arumaithurai, M., Ramakrishnan, K., and G. Pavlou, "Name-Based Replication Priorities in Disaster Cases", 2nd Workshop on Name Oriented Mobility (NOM), 2014.

[Robitzsch2015]
Robitzsch, S., Trossen, D., Theodorou, C., Barker, T., and A. Sathiaseel, "D2.1: Usage Scenarios and Requirements"", H2020 project RIFE, public deliverable, 2015.

[Seedorf2014]
Seedorf, J., Kutscher, D., and F. Schneider, "Decentralised Binding of Self-Certifying Names to Real-World Identities for Assessment of Third-Party Messages in Fragmented Mobile Networks", 2nd Workshop on Name Oriented Mobility (NOM), 2014.

[Seedorf2016]
Seedorf, J., Kutscher, D., and B. Gill, "Decentralised Interest Counter Aggregation for ICN in Disaster Scenarios", Workshop on Information Centric Networking Solutions for Real World Applications (ICNSRA), 2016.

[Sourlas2015]
Sourlas, V., Tassiulas, L., Psaras, I., and G. Pavlou, "Information Resilience through User-Assisted Caching in Disruptive Content-Centric Networks", 14th IFIP NETWORKING, May 2015.

[Tagami2016]
Tagami, A., Yagyu, T., Sugiyama, K., Arumaithurai, M., Nakamura, K., Hasegawa, T., Asami, T., and K. Ramakrishnan, "Name-based Push/Pull Message Dissemination for Disaster Message Board", The 22nd IEEE International Symposium on Local and Metropolitan Area Networks (LANMAN), 2016.

[Trossen2015]
Trossen, D., "IP over ICN - The better IP?", 2015 European Conference onNetworks and Communications (EuCNC), June/July 2015, pp. 413 - 417.

[Yoneki2007]
Yoneki, E., Hui, P., Chan, S., and J. Crowcroft, "A socio-aware overlay for publish/subscribe communication in delay tolerant networks", Proceedings of the 10th ACM Symposium on Modeling, Analysis, and Simulation of Wireless and Mobile Systems, 2007.

Appendix A. Acknowledgment

The authors would like to thank Ioannis Psaras for useful comments. Also, the authors are grateful to Christopher Wood and Daniel Corujo for valuable feedback and suggestions on concrete text for improving the document. Further, the authors would like to thank Joerg Ott and Dirk Trossen for valuable comments and input, in particular regarding existing work from the DTN community which is highly related to the ICN approaches suggested in this document. Also, Akbar Rahman provided useful comments and usggestions, in particular regarding existing disaster warning mechanisms in today’s mobile phone networks.

This document has been supported by the GreenICN project (GreenICN: Architecture and Applications of Green Information Centric Networking), a research project supported jointly by the European Commission under its 7th Framework Program (contract no. 608518) and the National Institute of Information and Communications Technology (NICT) in Japan (contract no. 167). The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the GreenICN project, the European Commission, or NICT. More information is available at the project web site http://www.greenicn.org/.

Authors’ Addresses

Jan Seedorf
HFT Stuttgart - Univ. of Applied Sciences
Schellingstrasse 24
Stuttgart 70174
Germany

Phone: +49 711 8926 2801
Fax: +49 711 8926 2553
Email: jan.seedorf@hft-stuttgart.de

Mayutan Arumaithurai
University of Goettingen
Goldschmidt Str. 7
Goettingen 37077
Germany

Phone: +49 551 39 172046
Fax: +49 551 39 14416
Email: arumaithurai@informatik.uni-goettingen.de

Atsushi Tagami
KDDI Research Inc.
2-1-15 Ohara
Fujimino, Saitama 356-85025
Japan

Phone: +81 49 278 73651
Fax: +81 49 278 7510
Email: tagami@kddi-research.jp

K. K. Ramakrishnan
University of California
Riverside CA
USA

Email: kkramakrishnan@yahoo.com

Nicola Blefari Melazzi
University Tor Vergata
Via del Politecnico, 1
Roma 00133
Italy

Phone: +39 06 7259 7501
Fax: +39 06 7259 7435
Email: blefari@uniroma2.it

ICNRG
Internet-Draft
Intended status: Informational
Expires: December 31, 2017

A. Rahman
D. Trossen
InterDigital Inc.
D. Kutscher
R. Ravindran
Huawei

June 29, 2017

Deployment Considerations for Information-Centric Networking (ICN) draft-rahman-icnrg-deployment-guidelines-02

Abstract

Information-Centric Networking (ICN) is now reaching technological maturity after many years of fundamental research and experimentation. This document provides a number of deployment considerations in the interest of helping the ICN community move forward to the next step of live deployments. The major deployment configurations for ICN are first described including the main overlay and underlay approaches. Then proposed deployment migration paths are outlined to address major practical issues such as network and application migration. Next, selected ICN trial experiences are summarized. Finally, protocol areas that require further standardization are identified to facilitate future interoperable ICN deployments.

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 http://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 December 31, 2017.

Table of Contents

1. Introduction

The ICNRG charter identifies deployment guidelines as an important topic area for the ICN community. Specifically, the charter states that defining concrete migration paths for ICN deployments which avoid forklift upgrades, and defining practical ICN interworking configurations with the existing Internet paradigm, are key topic areas that require further investigation. Also, it is well understood that results and conclusions from any mid to large-scale ICN experiments in the live Internet will also provide useful guidance for deployments.

However, so far outside of some preliminary investigations such as [I-D.paik-icn-deployment-considerations], there has not been much progress on this topic. This draft attempts to fill some of these gaps by defining clear deployment configurations for ICN, and associated migration pathways for these configurations. Also, selected deployment trial experiences of ICN technology are summarized. Finally, recommendations are made for potential future IETF standardization of key protocol functionality that will facilitate interoperable ICN deployments going forward.

2. Terminology

This document assumes readers are, in general, familiar with the terms and concepts that are defined in [RFC7927]. In addition, this document defines the following terminology:

Deployment - In the context of this document, deployment refers to the final stage of the process of setting up an ICN network that is (1) integrated and interoperable with the rest of the Internet, and (2) ready for useful work (e.g. transmission of end user video and text) in a live environment.

Information-Centric Networking (ICN) - A data-centric network architecture where accessing data by name is the essential network primitive. See [ICNterm] for further information.

Network Function Virtualization (NFV): A networking approach where network functions (e.g. firewalls, load balancers) are modularized as software logic that can run on general purpose hardware, and thus are specifically decoupled from the previous generation of proprietary and dedicated hardware. See

[I-D.irtf-nfvrg-gaps-network-virtualization] for further information.

Software-Defined Networking (SDN) - A networking approach where the control and data plane for switches are separated, allowing for realizing capabilities such as traffic isolation and programmable forwarding actions. See [RFC7426] for further information.

3. Deployment Configurations

In this section, we present various deployment options for ICN. These are presented as "configurations" that allow for studying these options further. While this document will outline experiences with various of these configurations (in Section 5), we will not provide an in-depth technical or commercial evaluation for any of them - for this we refer to existing literature in this space such as [Tateson].

3.1. Wholesale Replacement

ICN has often been described as a "clean-slate" approach with the goal to renew or replace the current IP routing fabric of the Internet. As such, existing routing hardware as well as ancillary services are not taken for granted. This clean-slate view is reflected as deployment configurations we label as "wholesale replacement" of large part of the existing Internet infrastructure. For instance, such configuration would see existing IP routers being replaced by ICN-specific forwarding and routing elements, such as NFD (Named Data Networking Forwarding Daemon) [NFD], CCN routers [Jacobson] or PURSUIT forwarding nodes [IEEE_Communications]. All major ICN approaches have explored this option as one of their paths to deployment.

While such replacement could be seen as exclusive for ICN deployments, some ICN approaches [POINT] rely on the deployment of infrastructure upgrades, here SDN switches. Such upgrades, while being possibly utilized for a "clean slate" ICN deployment would not necessary be used exclusively for an ICN deployment. Different proposals have been made for various ICN approaches to enable the operation over an SDN transport [Reed][CONET][C_FLOW].

3.2. ICN-as-an-Overlay

Similar to other significant changes to the Internet routing fabric, particularly the transition from IPv4 to IPv6 or the introduction of IP multicast, this deployment configuration foresees the creation of an ICN overlay. Note that this overlay approach is sometimes, informally, also referred to as a tunneling approach. The overlay

approach could be done directly such as ICN-over-UDP as described in [CCNx_UDP]. Alternatively, the overlay could be done via ICN-in-L2-in-IP as in [IEEE_Communications] which describes a recursive layering process.

Another flavor of overlay would be embedding ICN semantics into existing protocols. A recently announced approach is [Hybrid_ICN] where the ICN names are mapped to IPv6 addresses. Another approach used in the Network of Information (NetInf) is to define a convergence layer to map NetInf semantics to HTTP [I-D.kutscher-icnrg-netinf-proto]. Regardless of the flavor, however, the overlay approach results in islands of ICN deployments over existing IP-based infrastructure.

3.3. ICN-as-an-Underlay

Proposals such as [POINT] and [White] outline the deployment option of using an ICN underlay that would integrate with existing IP-based services by deploying application layer gateways at appropriate locations, while possibly still allowing for native ICN applications to emerge. The main reasons for such configuration option is the backward-compatible introduction of ICN technology, while reaping the benefits of ICN in terms of multicast delivery, mobility support, fast indirection due to location independence, in-network computing and possibly more.

3.3.1. Core Network

In this sub-option, a core network would utilize edge-based protocol mapping onto the native ICN underlay. For instance, [POINT] proposes to map HTTP transactions, or some other IP based transactions such as CoAP, directly onto an ICN-based message exchange. This mapping is realized at the network attachment point, such as realized in access points or customer premise equipment, which in turn provides a standard IP interface to existing user devices. Towards peering networks, such network attachment point turns into a modified border gateway/proxy, preserving the perception of an IP-based core network towards any peering network.

The work in [White] proposes a similar deployment configuration. Here, the target is the use of ICN for content distribution within CDN server farms, i.e., the protocol mapping is realized at the ingress of the server farm where the HTTP-based retrieval request is served, while the response is delivered through a suitable egress node translation.

3.3.2. Edge Network

Native ICN networks may be found at the edge of the network, mostly proposed for Internet of Things (IoT) deployments, which allows the possibility of introducing new network architectures and protocols, and in this context ICN can be a possible candidate considering its suitability for IoT and fixed network scenarios [I-D.zhang-icnrg-icniot]. The integration with the current IP protocol suite takes place at an application gateway/proxy at the edge network boundary, e.g., translating incoming CoAP request/ response transactions [RFC7252] into ICN message exchanges or vice versa. Furthermore, ICN will allow us to evolve the role of gateways/proxies as ICN message security should be preserved through the protocol translation function of a gateway/proxy and thus offer a substantial gain.

The work in [VSER] positions ICN as an edge service gateway driven by an generalized ICN based service orchestration system with its own compute and network virtualization controllers to manage an ICN infrastructure. The platform also offers service discovery capabilities to enable user applications to discover appropriate ICN service gateways. To exemplify a use case scenario, the platform shows the realization of a multi-party audio/video conferencing service over such a edge cloud deployment of ICN routers realized over commodity hardware platforms. This platform has also been extended to offer seamless mobility and mobility as a service [VSER-Mob] features.

3.4. ICN-as-a-Slice

The objective of Network slicing [NGMN]is to multiplex a general pool of compute, storage and bandwidth resources among multiple services with exclusive SLA requirements on transport level QoS and security. These services could include both connectivity services like LTE-as-a-service or OTT services like VoD or other IoT services. Such a framework can also be used to realize ICN slices with its own control and forwarding plane over which one or more end-user services can be delivered.

An ICN slice can itself be overlaid over IP or can be underlaid using generalized programmable data planes like P4/POF. Such a generalized network slicing framework should be able to offer service slices to be realized over both IP and ICN. Network slicing will rely heavily on network softwarization and programmability using SDN/NFV technologies for efficient utilization of available resources without compromising on the slice requirements. Coupled with the view of ICN functions as being "chained service functions" [RFC7665], an ICN deployment within such a slice could be realized within the emerging

control plane that is targeted for adoption in future (e.g., 5th generation mobile) network deployments. Finally, it should be noted that ICN is not creating the network slice, but instead that the slice is created to run a 5G-ICN instance [Ravindran].

At the level of the specific technologies involved, the 5G-ICN slice requires compatibility for instance at the level of the forwarding/ data plane depending on if it is realized as an overlay or using programmable data planes. With SDN emerging for new network deployments, some ICN approaches will need to integrate with SDN as a data plane forwarding function, as briefly discussed in Section 3.1.

4. Deployment Migration Paths

After outlining the various ICN deployment configurations in Section 3, we now focus on the various migration paths that will have importance to the various stakeholders that are usually involved in the deployment of a technology at (ultimately) large scale. We can identify these stakeholders as:

- Application developers and service providers
- ISPs, both as core as well as access network providers, and also ICN network providers
- CDN providers (due to the strong relation of the ICN proposition to content delivery)

Note that our presentation purely focuses on technological aspects of such migration. Economic or regulatory aspects, such as studied in [Tateson], [Techno_Economic] and [Internet_Pricing] are left out of our discussion.

4.1. Application and Service Migration

The internet is full of applications and services, utilizing the innovation capabilities of the many protocols defined over the packet level IP service. HTTP provides one convergence point for these services with many web development frameworks based on the semantics provided by the hypertext transfer protocol. In recent years, even services such as video delivery have been migrating from the traditional RTP-over-UDP delivery to the various HTTP-level streaming solutions, such as DASH [DASH] and others. Nonetheless, many non-HTTP services exist, all of which need consideration when migrating from the IP-based internet to an ICN-based one.

The underlay deployment configuration options presented in Section 3.3.1 and Section 3.3.2 aim at providing some level of

backward compatibility to this existing ecosystem through a proxy based message flow mapping mechanism (e.g., mapping of existing HTTP/TCP/IP message flows to HTTP/TCP/IP/ICN message flows). A related approach of mapping TCP/IP to TCP/ICN message flows is described in [Moiseenko]

Alternatively, ICN as an overlay (Section 3.2), as well as ICN-as-a-Slice (Section 3.4), allow for the introduction of the full capabilities of ICN through new service interfaces as well as operations in the network. With that, these approaches of deployment are likely to aim at introducing new services capitalizing on those ICN capabilities.

4.2. Content Delivery Network Migration

A significant number of services and applications are devoted to content delivery in some form, either as video delivery services, social media platforms, and many others. Content delivery networks (CDNs) are deployed to assist these services through localizing the content requests and therefore reducing latency and possibly increase utilization of available bandwidth as well as reducing the load on origin servers. Similar to the previous sub-section, the underlay deployment configurations presented in Section 3.3.1 and Section 3.3.2 aim at providing a migration path for existing CDNs. This is also highlighted in the BIER WG use case draft [I-D.ietf-bier-use-cases], specifically with potential benefits in terms of utilizing multicast in the delivery of content but also reducing load on origin as well as delegation server. We return to this benefit in the trial experiences in Section 5.

4.3. Edge Network Migration

Edge networks often see the deployment of novel network level technology, e.g., in the space of IoT. Such IoT deployments have for many years relied, and often still do, on proprietary protocols for reasons such as increased efficiency, lack of standardization incentives and others. Utilizing the underlay deployment configuration in Section 3.3.2, application gateways/proxies can integrate such edge deployments into IP-based services, e.g., utilizing CoAP [RFC7252] based machine-to-machine (M2M) platforms such as oneM2M [oneM2M] or others.

Another area of increased edge network innovation is that of mobile (access) networks, particularly in the context of the 5th generation of mobile networks (often called "5G"). With the proliferation of network softwarization (using technologies like service orchestration frameworks leveraging NFV and SDN concepts) access networks and other network segments, the ICN-as-a-Slice deployment configuration in

Section 3.4 provides a suitable migration path for integration non-IP-based edge networks into the overall system through virtue of realizing the relevant (ICN) protocols in an access network slice.

4.4. Core Network Migration

Migrating the core network of the Internet requires not only significant infrastructure renewal but also the fulfillment of the significant performance requirements, particularly in terms of throughput. For those parts of the core network that would see a migration to an SDN-based optical transport, such as proposed by major operators such as AT&T, the ICN-as-a-Slice deployment configuration in Section 3.4 could see the introduction of native ICN solutions within slices provided by the SDN-enabled transport network or as virtual network functions, allowing for isolating the ICN traffic while addressing the specific ICN performance benefits and constraints within such isolated slice. For ICN solutions that natively work on top of SDN, the underlay deployment configuration in Section 3.3.1 provides an additional migration path, preserving the IP-based services and applications at the edge of the network, while realizing the core network routing through an ICN solution (possibly itself realized in a slice of the SDN transport network).

5. Deployment Trial Experiences

In this section, we will outline trial experiences, often conducted within international collaborative project efforts. Our focus here is on the realization of the various deployment configurations in Section 3, and we therefore categorize the trial experiences according to these deployment configurations. While a large body of work exists at the simulation or emulation level, we specifically exclude these studies from our presentation to retain the focus on real life experiences.

5.1. ICN-as-an-Overlay

5.1.1. FP7 PURSUIT Efforts

Although the FP7 PURSUIT [IEEE_Communications] efforts were generally positioned as a wholesale replacement of IP (Section 3.1), the project realized its experimental test bed as an L2 VPN-based overlay between several European, US as well as Asian sites, i.e., following the overlay deployment configuration presented in Section 3.2. Software-based forwarders were utilized for the ICN message exchange, while native ICN applications, e.g., for video transmissions, were showcased. At the height of the project efforts, about 70+ nodes were active in the (overlay) network with presentations given at several conferences as well as to the ICNRG.

5.1.2. FP7 SAIL Trial

The Network of Information (NetInf) is the approach to Information-Centric Networking developed by the European Union (EU) FP7 SAIL project (http://www.sail-project.eu/). NetInf provides both name-based forwarding with CCNx-like semantics and name resolution (for indirection and late-binding). The NetInf architecture supports different deployment options through its convergence layer abstraction. In its first prototypes and trials, NetInf was deployed mostly in an HTTP embedding and in a UDP overlay following the overlay deployment configuration in Section 3.2. Reference [SAIL_NetInf] describes several trials including a stadium environment large crowd scenario and a multi-site testbed, leveraging NetInf's Routing Hint approach for routing scalability.

5.1.3. NDN Testbed

The Named Data Networking (NDN) is one of the research projects funded by the National Science Foundation (NSF) of the USA as part of the Future Internet Architecture Program. The original NDN proposal was positioned as a wholesale replacement of IP (Section 3.1). However, in several trials, NDN generally follows the overlay deployment configuration of Section 3.2 to connect institutions over the public Internet across several continents. The use cases covered in the trials include real-time video-conferencing, geo-locating, and interfacing to consumer applications. Typical trials involve several hundred NDN enabled nodes (https://named-data.net/ndn-testbed/).

5.1.4. ICN2020 Efforts

ICN2020 is an ICN related research project funded by the EU as part of the H2020 research and innovation program (http://www.icn2020.org/). ICN2020 has a specific focus to advance ICN towards real-world deployments through innovative applications and global scale experimentation. Both NDN and CCN approaches are within the scope of the project.

ICN2020 was kicked off in late 2016 and so has not yet published results relating to deployment trials. The plan, however, is to involve ICN testbeds in EU, Japan and the USA and federate them. The GEANT testbed (https://www.geant.org/) is being considered as one means to federate the different ICN testbeds in the overlay deployment configuration of Section 3.2 over the public Internet.

5.2. ICN-as-an-Underlay

5.2.1. H2020 POINT and RIFE Efforts

POINT and RIFE are two more ICN related research projects funded by the EU as part of the H2020 effort. The efforts in the H2020 POINT+RIFE projects follow the underlay deployment configuration in Section 3.3.1, although this is mixed with utilizing an overlay deployment to provide multi-national connectivity. However, underlay SDN-based deployments do exist at various project partner sites, e.g., at Essex University, without any overlaying being realized. Edge-based network attachment points (NAPs) provide the IP/HTTP-level protocol mapping onto ICN protocol exchanges, while the SDN underlay (or the VPN-based L2 underlay) is used as a transport network.

The multicast as well as service endpoint surrogate benefits in HTTP-based scenarios, such as for HTTP-level streaming video delivery, have been demonstrated in the deployed POINT test bed with 80+ nodes being utilized. Demonstrations of this capability have been given to the ICNRG in 2016, while public demonstrations were also provided at events such as Mobile World Congress in 2016 [MWC_Demo]. The trial has also been accepted by the ETSI MEC group as a proof-of-concept with a demonstration at the ETSI MEC World Congress in 2016.

While the afore-mentioned demonstrations all use the overlay international deployment, both H2020 efforts plan ICN underlay trials in the summer and fall of 2017. One such trial will involve commercial end users located in the Primetel network in Cyprus with the use case centered on IPTV and HLS video dissemination. Another trial is planned for fall 2017 in the community network of "guifi.net" in the Barcelona region, where the solution will be deployed in 40 households, providing general Internet connectivity to the residents. Standard IPTV STBs as well as HLS video players will be utilized in accordance with the aim of this deployment configuration, namely to provide application and service migration.

5.2.2. H2020 FLAME Efforts

Starting in January 2017, the H2020 FLAME efforts aims at providing an experimental ground for the aforementioned POINT/RIFE solution in initially two city-scale locations, namely in Bristol and Barcelona. This trial will again follow the underlay deployment configuration in Section 3.3.1 as per POINT/RIFE approach. Currently, experiments are ongoing,conducted by the city/university joint venture Bristol-is-Open (BIO), to ensure the readiness of the city-scale SDN transport network for such experiments. A third trial of the aforementioned ETSI MEC PoC is planned for mid 2017. This trial will showcase operational benefits provided by the ICN underlay for the scenario of

a location-based game. These benefits aim at reduced network utilization through improved video delivery performance (multicast of all captured videos to the service surrogates deployed in the city at six locations) as well as reduced latency through the playout of the video originating from the local NAP instead of a remote server.

Ensuring the technology readiness and the early trialing of the ICN capabilities lays the ground for the goal of the H2020 FLAME efforts to conduct 23 large-scale experiments in the area of Future Media Internet (FMI) throughout 2018 and 2019. Standard media service functions as well as applications will ultimately utilize the ICN underlay in the delivery of their experience. The platform, which includes the ICN capabilities, will utilize concepts of SFC, integrated with NFV and SDN capabilities of the infrastructure. The ultimate goal of these platform efforts is the full integration of ICN into the overall media function platform for the provisioning of advanced (media-centric) internet services.

5.2.3. CableLabs Content Delivery System

The work in [White] proposes an underlay deployment configuration based on Section 3.3.1. The use case is ICN for content distribution within CDN server farms (which can be quite large and complex) to leverage ICN’s superior in-network caching properties. This "island of ICN" based CDN is then used to service standard HTTP/IP-based content retrieval request coming from the general Internet. This approach acknowledges that whole scale replacement (see Section 3.1) of existing HTTP/IP end user applications and related Web infrastructure is a difficult proposition. [White] does not yet provide results but indicated that experiments will be forthcoming.

6. Deployment Issues Requiring Further Standardization

The ICN Research Challenges [RFC7927] describes key ICN principles and technical research topics. As the title suggests, [RFC7927] is research oriented without a specific focus on deployment or standardization issues. This section addresses this open area by identifying key protocol functionality that that may be relevant for further standardization effort in IETF. The focus is specifically on identifying protocols that will facilitate future interoperable ICN deployments correlating to the scenarios identified in the deployment migration paths in Section 4. The identified list of potential protocol functionality is not exhaustive.

6.1. Protocols for Application and Service Migration

End user applications and services need a standardized approach to trigger ICN transactions. For example, in Internet and Web applications today, there are established socket APIs, communication paradigms such as REST, common libraries, and best practices. We see a need to study application requirements in an ICN environment further and, at the same time, develop new APIs and best practices that can take advantage of ICN communication characteristics.

6.2. Protocols for Content Delivery Network Migration

A key issue in CDNs is to quickly find a location of a copy of the object requested by an end user. In ICN, a Named Data Object (NDO) is typically defined by its name. There already exists [RFC6920] that is suitable for static naming of ICN data objects. Other ways of encoding and representing ICN names have been described in [I-D.irtf-icnrg-ccnxmessages] and [I-D.mosko-icnrg-ccnxurischeme]. Naming dynamically generated data requires different approaches (for example, hash digest based names would normally not work), and there is lack of established conventions and standards.

Another CDN issue for ICN is related to multicast distribution of content. Existing CDNs have started using multicast mechanisms for certain cases such as for broadcast streaming TV. However, as discussed in Section 5.2.1, certain ICN approaches provide substantial improvements over IP multicast, such as the implicit support for multicast retrieval of content in all ICN flavours.

Caching is an implicit feature in many ICN architectures that can improve performance and availability in several scenarios. The ICN in-network caching can augment managed CDN and improve its performance. The details of the interplay between ICN caching and managed CDN need further consideration.

6.3. Protocols for Edge and Core Network Migration

ICN provides the potential to redesign current edge and core network computing approaches. Leveraging ICN’s inherent security and its ability to make name data and dynamic computation results available independent of location, can enable a secure, yet light-weight insertion of traffic into the network without relying on redirection of DNS requests. For this, proxies that translate from commonly used protocols in the general Internet to ICN message exchanges in the ICN domain could be used for the migration of application and services within deployments at the network edge but also in core networks. This is similar to existing approaches for IoT scenarios where a proxy translates CoAP request/responses to other message formats.

For example, [RFC8075] specifies proxy mapping between CoAP and HTTP protocols. However, as mentioned previously, ICN will allow us to evolve the role of gateways/proxies as ICN message security should be preserved through the protocol translation function of a thus offer a substantial gain. Another area is integration of ICN into networks that support virtualized infrastructure in the form of NFV/SDN. Further work is required to validate this idea and document best practices.

6.4. Summary of ICN Protocol Gaps and Potential IETF Efforts

Without claiming completeness, Table 1 maps the open the open ICN issues identified in this document to potential protocol efforts that could address some aspects of the gap.

ICN Gap	Potential Protocol Effort
1-Support of REST APIs	HTTP/CoAP support of ICN semantics
2-Naming	Dynamic naming of ICN data objects
3-Multicast distribution	Multicast enhancements for ICN
4-In-network caching	ICN Cache placement and sharing
5-NFV/SDN Support	Integration of ICN with NFV/SDN
6-ICN mapping	Mapping of HTTP and other protocols onto ICN message exchanges (and vice-versa) while preserving ICN message security

Table 1: Mapping of ICN Gaps to Potential Protocol Efforts

7. Conclusion

This document provides high level deployment considerations for the ICN community. Specifically, the major configurations of possible ICN deployments are identified as (1) wholesale replacement of existing Internet infrastructure; (2) ICN-as-an-Overlay; (3) ICN-as-an-Underlay; and (4) ICN-as-a-Slice. Existing ICN trial systems

mainly fall either under the ICN-as-an-Overlay or ICN-as-an-Underlay configuration.

In terms of deployment migration paths, ICN-as-an-Underlay offers a clear migration path for existing CDN, edge and core networks to go to an ICN paradigm. ICN-as-a-Slice is an attractive deployment option for future 5G systems (i.e., for 5G radio and core networks) which will naturally support network slicing, but this still has to be validated through actual trial experiences. Finally, for the crucial issue of existing application and service migration to ICN, various mapping schemes are possible to mitigate impacts. For example, HTTP/TCP/IP flows may be mapped to ICN message flows at a proxy in the ICN-as-an-Underlay configurations leaving the massive number of existing end point applications/services untouched or minimally impacted.

Finally, this document describes a set of technical features in ICN that warrant potential future IETF specification work. This will aid initial and incremental deployments to proceed in an interoperable manner. The fundamental details of the potential protocol specification effort, however, are best left for future study by the appropriate WGs and/or BoFs.

8. IANA Considerations

This document requests no IANA actions.

9. Security Considerations

ICN was purposefully designed from the start to have certain intrinsic security properties. The most well known of which are authentication of delivered content and (optional) encryption of the content. [RFC7945] has an extensive discussion of various aspects of ICN security including many which are relevant to deployments. Specifically, [RFC7945] points out that ICN access control, privacy, security of in-network caches, and protection against various network attacks (e.g. DoS) have not yet been fully developed due to the lack of real deployments. [RFC7945] also points out relevant advances occurring in the ICN research community that hold promise to address each of the identified security gaps. Lastly, [RFC7945] points out that as secure communications in the existing Internet (e.g. HTTPS) becomes the norm, that major gaps in ICN security will inevitably slow down the adoption of ICN.

In addition to the security findings of [RFC7945], this document has highlighted that all anticipated ICN deployment configurations will involve co-existence with existing Internet infrastructure and applications. Thus even the basic authentication and encryption

properties of ICN content will need to account for interworking with non-ICN content to preserve end-to-end security. For example, in the edge network underlay deployment configuration described in Section 3.3.2, the gateway/proxy that translates HTTP or CoAP request/responses into ICN message exchanges will need to support a model to preserve end-to-end security.

10. Acknowledgments

The authors want to thank Alex Afanasyev, Xavier de Foy, Hannu Flinck, Dave Oran, and Prakash Suthar for their very useful reviews and comments to the document.

11. Informative References

[C_FLOW] Suh, J. and et al., "C_FLOW: Content-Oriented Networking over OpenFlow", Open Networking Summit, April, 2012, <http://opennetsummit.org/archives/apr12/site/pdf/snu.pdf>.

[CCNx_UDP] PARC, "CCNx Over UDP", 2015, <https://www.ietf.org/proceedings/interim-2015-icnrg-04/slides/slides-interim-2015-icnrg-4-5.pdf>.

[CONET] Veltri, L. and et al., "CONET Project: Supporting Information-Centric Functionality in Software Defined Networks", Workshop on Software Defined Networks, , 2012, <http://netgroup.uniroma2.it/Stefano_Salsano/papers/salsano-icc12-wshop-sdn.pdf>.

[DASH] DASH, "DASH Industry Forum", 2017, <http://dashif.org/>.

[Hybrid_ICN] Cisco, "Hybrid ICN: Cisco Announces Important Steps toward Adoption of Information-Centric Networking", 2017, <http://blogs.cisco.com/sp/cisco-announces-important-steps-toward-adoption-of-information-centric-networking>.

[I-D.ietf-bier-use-cases] Kumar, N., Asati, R., Chen, M., Xu, X., Dolganow, A., Przygienda, T., arkadiy.gulko@thomsonreuters.com, a., Robinson, D., Arya, V., and C. Bestler, "BIER Use Cases", draft-ietf-bier-use-cases-04 (work in progress), January 2017.

[I-D.irtf-icnrg-ccnxmessages]
marc.mosko@parc.com, m., Solis, I., and C. Wood, "CCNx Messages in TLV Format", draft-irtf-icnrg-ccnxmessages-04 (work in progress), March 2017.

[I-D.irtf-nfvrg-gaps-network-virtualization]
Bernardos, C., Rahman, A., Zuniga, J., Contreras, L., Aranda, P., and P. Lynch, "Network Virtualization Research Challenges", draft-irtf-nfvrg-gaps-network-virtualization-05 (work in progress), March 2017.

[I-D.kutscher-icnrg-netinf-proto]
Kutscher, D., Farrell, S., and E. Davies, "The NetInf Protocol", draft-kutscher-icnrg-netinf-proto-01 (work in progress), February 2013.

[I-D.mosko-icnrg-ccnxurischeme]
marc.mosko@parc.com, m. and c. cwood@parc.com, "The CCNx URI Scheme", draft-mosko-icnrg-ccnxurischeme-01 (work in progress), April 2016.

[I-D.paik-icn-deployment-considerations]
Paik, E., Yun, W., Kwon, T., and h. hgchoi@mmlab.snu.ac.kr, "Deployment Considerations for Information-Centric Networking", draft-paik-icn-deployment-considerations-00 (work in progress), July 2013.

[I-D.zhang-icnrg-icniot]
Zhang, Y., Raychadhuri, D., Grieco, L., Baccelli, E., Burke, J., Ravindran, R., Wang, G., Lindgren, A., Ahlgren, B., and O. Schelen, "Design Considerations for Applying ICN to IoT", draft-zhang-icnrg-icniot-01 (work in progress), June 2017.

[ICNterm] Wissingh, B., "Information-Centric Networking (ICN): Terminology", 2017, <https://datatracker.ietf.org/doc/draft-wissingh-icnrg-terminology/>.

[IEEE_Communications]
Trossen, D. and G. Parisis, "Designing and Realizing an Information-Centric Internet", Information-Centric Networking, IEEE Communications Magazine Special Issue, 2012.

[Internet_Pricing]
Trossen, D. and G. KBiczok, "Not Paying the Truck Driver: Differentiated Pricing for the Future Internet", ReArch Workshop in conjunction with ACM Context, December, 2010.

[Jacobson]
Jacobson, V. and et al., "Networking Named Content", Proceedings of ACM Context, , 2009.

[Moiseenko]
Moiseenko, I. and D. Oran, "TCP/ICN : Carrying TCP over Content Centric and Named Data Networks", 2016, <http://conferences2.sigcomm.org/acm-icn/2016/proceedings/p112-moiseenko.pdf>.

[MWC_Demo]
InterDigital, "InterDigital Demo at Mobile World Congress (MWC)", 2016, <http://www.interdigital.com/download/56d5c71bd616f892ba001861>.

[NFD] NDN, "NFD - Named Data Networking Forwarding Daemon", 2017, <https://named-data.net/doc/NFD/current/>.

[NGMN] NGMN, "NGMN 5g Initiative White Paper", 2015, <https://www.ngmn.org/uploads/media/NGMN_5G_White_Paper_V1_0.pdf>.

[oneM2M] OneM2M, "oneM2M Service Layer Standards for M2M and IoT", 2017, <http://www.onem2m.org/>.

[POINT] Trossen, D. and et al., "POINT: IP Over ICN - The Better IP?", European Conference on Networks and Communications (EuCNC), , 2015.

[Ravindran]
Ravindran, R., Chakraborti, A., Amin, S., Azgin, A., and G. Wang, "5G-ICN : Delivering ICN Services over 5G using Network Slicing", IEEE Communication Magazine, May, 2016, <https://arxiv.org/abs/1610.01182>.

[Reed] Reed, M. and et al., "Stateless Multicast Switching in Software Defined Networks", ICC 2016, Kuala Lumpur, Malaysia, 2016.

[RFC6920] Farrell, S., Kutscher, D., Dannewitz, C., Ohlman, B., Keranen, A., and P. Hallam-Baker, "Naming Things with Hashes", RFC 6920, DOI 10.17487/RFC6920, April 2013, <http://www.rfc-editor.org/info/rfc6920>.

[RFC7252] Shelby, Z., Hartke, K., and C. Bormann, "The Constrained Application Protocol (CoAP)", RFC 7252, DOI 10.17487/RFC7252, June 2014, <http://www.rfc-editor.org/info/rfc7252>.

[RFC7426] Haleplidis, E., Ed., Pentikousis, K., Ed., Denazis, S., Hadi Salim, J., Meyer, D., and O. Koufopavlou, "Software-Defined Networking (SDN): Layers and Architecture Terminology", RFC 7426, DOI 10.17487/RFC7426, January 2015, <http://www.rfc-editor.org/info/rfc7426>.

[RFC7665] Halpern, J., Ed. and C. Pignataro, Ed., "Service Function Chaining (SFC) Architecture", RFC 7665, DOI 10.17487/RFC7665, October 2015, <http://www.rfc-editor.org/info/rfc7665>.

[RFC7927] Kutscher, D., Ed., Eum, S., Pentikousis, K., Psaras, I., Corujo, D., Saucez, D., Schmidt, T., and M. Waehlisch, "Information-Centric Networking (ICN) Research Challenges", RFC 7927, DOI 10.17487/RFC7927, July 2016, <http://www.rfc-editor.org/info/rfc7927>.

[RFC7945] Pentikousis, K., Ed., Ohlman, B., Davies, E., Spirou, S., and G. Boggia, "Information-Centric Networking: Evaluation and Security Considerations", RFC 7945, DOI 10.17487/RFC7945, September 2016, <http://www.rfc-editor.org/info/rfc7945>.

[RFC8075] Castellani, A., Loreto, S., Rahman, A., Fossati, T., and E. Dijk, "Guidelines for Mapping Implementations: HTTP to the Constrained Application Protocol (CoAP)", RFC 8075, DOI 10.17487/RFC8075, February 2017, <http://www.rfc-editor.org/info/rfc8075>.

[SAIL_NetInf] FP7, "SAIL Prototyping and Evaluation", 2013, <http://www.sail-project.eu/wp-content/uploads/2013/05/SAIL_DB4_v1.1_Final_Public.pdf>.

[Tateson] Tateson, J. and et al., "Final Evaluation Report on Deployment Incentives and Business Models", 2010, <http://www.psirp.org/files/Deliverables/FP7-INFSO-ICT-216173-PSIRP-D4.6_FinalReportOnDeplIncBusinessModels.pdf>.

[Techno_Economic] Trossen, D. and A. Kostopolous, "Techno-Economics Aspects of Information-Centric Networking", Journal for Information Policy, Volume 2, 2012.

[VSER] Ravindran, R., Liu, X., Chakraborti, A., Zhang, X., and G. Wang, "Towards software defined ICN based edge-cloud services", CloudNetworking(CloudNet), IEEE Internation Conference on, IEEE Internation Conference on CloudNetworking(CloudNet), 2013.

[VSER-Mob] Azgin, A., Ravindran, R., Chakraborti, A., and G. Wang, "Seamless Mobility as a Service in Information-centric Networks", ACM ICN Sigcomm, IC5G Workshop, 2016.

[White] White, G. and G. Rutz, "Content Delivery with Content Centric Networking, CableLabs White Paper", 2010, <http://www.cablelabs.com/wp-content/uploads/2016/02/Content-Delivery-with-Content-Centric-Networking-Feb-2016.pdf>.

Authors' Addresses

Akbar Rahman
InterDigital Inc.
1000 Sherbrooke Street West, 10th floor
Montreal H3A 3G4
Canada

Email: Akbar.Rahman@InterDigital.com
URI: http://www.InterDigital.com/

Dirk Trossen
InterDigital Inc.
64 Great Eastern Street, 1st Floor
London EC2A 3QR
United Kingdom

Email: Dirk.Trossen@InterDigital.com
URI: http://www.InterDigital.com/

Dirk Kutscher
Huawei German Research Center
Riesstrasse 25
Munich 80992
Germany

Email: ietf@dkutscher.net
URI: http://www.Huawei.com/

Ravi Ravindrna
Huawei Research Center
2330 Central Expressway
Santa Clara 95050
USA

Email: ravi.ravindran@huawei.com
URI: http://www.Huawei.com/

ICN Research Group
Internet-Draft
Intended status: Informational

Expires: January 3, 2018

Prakash Suthar
Milan Stolic
Anil Jangam
Cisco Systems
July 02 2017

Native Deployment of ICN in LTE, 4G Mobile Networks
draft-suthar-icnrg-icn-lte-4g-02

Abstract

LTE, 4G mobile networks use IP transport which is not optimized for data transport. IP unicast routing from server to clients is used for delivery of multimedia content to User Equipment (UE), where each user gets separate stream. From bandwidth and routing perspective this approach is inefficient. Multicast and broadcast technologies have emerged recently for mobile networks, but their deployments are very limited or at an experimental stage due to complex architecture and radio spectrum issues. ICN is a rapidly emerging technology with built in features for efficient multimedia data delivery, however majority of the work is focused on fixed networks. The main focus of this draft is on native deployment of ICN in cellular mobile networks by using ICN into 3GPP protocol stack. ICN has an inherent capability for multicast, anchorless mobility, security and it is optimized for data delivery using local caching at the edge. The native ICN or dual stack (along with IP) deployment will bring all inherent benefits and help in optimizing mobile networks.

http://www.ietf.org/shadow.html

Table of Contents

1 Introduction

LTE mobile technology is built as all IP network. It uses IP routing protocols such as OSPF, ISIS, BGP etc. to establish network routes to route the data traffic to end user's device. Stickiness of IP address to a device is the key to get connected to a mobile network and the same IP address is maintained through the session until the device gets detached or moves to another network.

One of the key protocols used in 4G and LTE networks is GPRS Tunneling protocol (GTP). GTP, DIAMETER and other protocols are built on top of IP. One of the biggest challenges with IP based routing is that it is not optimized for data transport although it is the most efficient communication protocol. By native implementation of Information Centric Networking (ICN) in 3GPP, we can re-architect mobile network and optimize its design for efficient data transport by leveraging the caching feature of ICN. ICN also offers an opportunity to leverage inherent capabilities of multicast, anchorless mobility management, authentication. This draft provides insight into different options for deploying ICN in mobile networks and how they impact mobile providers and end-users.

1.1 Terminology

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 RFC 2119 [RFC2119].

2. LTE, 4G Mobile Network

2.1 Network Overview

With the introduction of LTE, mobile networks moved to all-IP transport for all elements such as eNodeB, MME, SGW/PGW, HSS, PCRF, routing and switching etc. Although LTE network is data-centric, it has support for legacy Circuit Switch features like voice and SMS through transitional CS fallback and flexible IMS deployment [GRAYSON]. For each mobile device attached to the radio (eNodeB) there is a separate overlay tunnel (GPRS Tunneling Protocol, GTP) between eNodeB and Mobile gateways (i.e. SGW, PGW). This tunnel is used to carry user traffic between gateways and mobile devices so the data can only be distributed using unicast mechanism.

It is also important to understand the overhead of a GTP and IPSec protocols because it has impact on the carried user data traffic. All mobile backhaul traffic is encapsulated using GTP tunnel, which has overhead of 8 bytes on top of IP and UDP [NGMN]. Additionally, if IPSec is used for security (which is often required if the Service

provider is using a shared backhaul), it adds additional overhead based upon IPSec tunneling model (tunnel or transport), and encryption and authentication header algorithm used. If we factor Advanced Encryption Standard (AES) encryption with packet size the overhead can vary significantly [IPSEC2].

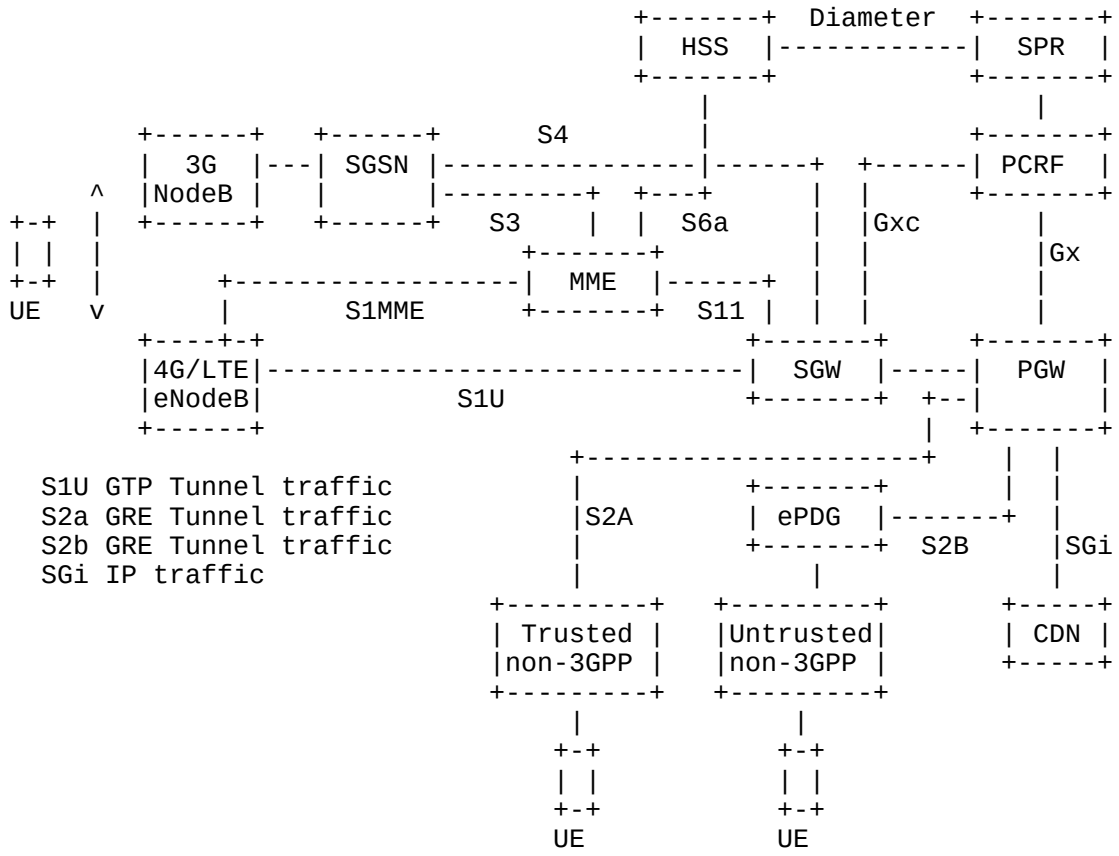

Figure-1: LTE, 4G Mobile Network Overview

When any UE is powered-up, it attaches to a mobile network based on its configuration and subscription. After successful attach procedure, UE registers with the mobile core network and IPv4 and/or IPv6 address is assigned. A default bearer is created for each UE and it is assigned to default Access Point Name (APN).

The data delivered to mobile devices is unicast inside GTP tunnel. If we consider combined impact of GTP, IPSec and unicast traffic, the data delivery is not efficient. IETF has developed various header compression algorithms to reduce the overhead associated with IP

packets. Some of techniques are robust header compression (ROHC) and enhanced compression of the real-time transport protocol (ECRTP) so that impact of overhead created by GTP, IPsec etc. is reduced to some extent [BROWER]. For commercial mobile networks, 3GPP has adopted different mechanisms for header compression to achieve efficiency in data delivery [TS25.323], and can also be used in ICN as well.

2.2 QoS Challenges

During the attach procedure, default bearer is created for each UE and it is assigned to the default Access Point Name (APN). The QoS parameters such as uplink/downlink bandwidth assigned during initial attach are minimal. Additional dedicated bearer(s) with enhanced QoS parameters can be established depending on the specific application requirements.

While all traffic within a certain bearer gets the same treatment, QoS parameters supporting these requirements can be very granular in different bearers. These values vary for the control, management and user traffic, and depending on the application key parameters, such as latency, jitter (important for voice and other real-time applications), packet loss and queuing mechanism (strict priority, low-latency, fair etc.) can be very different.

Implementation of QoS for mobile networks is done at two stages: at content prioritization/marking and transport marking, and congestion management. From the transport perspective, QoS is defined at layer-2 as class of service (CoS) and at layer-3 either as DiffServ code point (DSCP) or type of service (ToS). The mapping of CoS to DSCP takes place at layer-2/3 switching and routing elements. 3GPP has specified QoS Class Identifier (QCI) which represents different types of content and equivalent mapping to DSCP at transport layer [TS23.203] [TS23.401]; however, this again requires manual configuration at different elements and if there is misconfiguration at any place in the path it will not work properly.

In summary QoS configuration for mobile network for user plane (for user traffic) and transport in IP based mobile network is complex and it require synchronization of parameters among different platforms. Any misconfiguration in IP QoS can result in poor subscriber experience. By deploying ICN, we intend to enhance the subscriber experience using its inherent capabilities.

2.3 Data Transport Using IP

The data delivered to mobile devices is unicast inside GTP tunnel from a eNodeB to a PDN gateway (PGW), as described in 3GPP specifications [TS23.401]. While the technology exists to address the

issue of possible multicast delivery, there are many difficulties related to multicast protocol implementation on the RAN side of the network. Transport networks in the backhaul and core have addressed the multicast delivery long time ago and have implemented it in most cases in their multi-purpose integrated transport, but the RAN part of the network is still lagging behind due to complexities related to mobility of the clients, handovers, and the fact that the potential gain to the Service Providers may not justify the investment. With that said, the data delivery in the mobility remains greatly unicast.

To ease the burden on the bandwidth of the SGi interface, caching is introduced in a similar manner as with many Enterprises. In the mobile networks, whenever possible, a cached data is delivered. Caching servers are placed at a centralized location, typically in the Service Provider's Data Center, or in some cases lightly distributed in the Packet Core locations with the PGW nodes close to the Internet and IP services access (SGi interface). This is a very inefficient concept because traffic has to traverse the entire backhaul path for the data to be delivered to the end-user. Other issues, such as out-of-order delivery contribute to this complexity and inefficiency but could be addressed at the IP transport level.

The data delivered to mobile devices is unicast inside a GTP tunnel. If we consider combined impact of GTP, IPSec and unicast traffic, the data delivery is not efficient. By deploying ICN, we intend to either terminate GTP tunnel at the edge by leveraging control and user plane separation or replace it with the native ICN protocols.

2.4 Virtualizing Mobile Networks

The Mobile packet core deployed in a major service provider network is either based on dedicated hardware or large capacity x86 platforms in some cases. With adoption of Mobile Virtual Network Operators (MVNO), public safety network, and enterprise mobility network, we need elastic mobile core architecture. By deploying mobile packet core on a commercially off the shelf (COTS) platform using virtualized infrastructure (NFVI) framework and end-to-end orchestration, we can simplify new deployments and provide optimized total cost of ownership (TCO).

While virtualization is growing and many mobile providers use hybrid architecture consisting of dedicated and virtualized infrastructures, the control and data delivery planes are still the same. There is also work underway to separate control plane and user plane so that the network can scale better. Virtualized mobile networks and network slicing with control and user plane separation provide mechanism to evolve GTP-based architecture to open-flow SDN-based signaling for LTE and proposed 5G. Some of early architecture work for 5G mobile

technologies provide mechanism for control and user plane separation and simplifies mobility call flow by introduction of open-flow based signaling is described in [TS23.501] and [TS23.214].

4. Data Transport Using ICN

For mobile devices, the edge connectivity to the network is between radio and a router or mobile edge computing (MEC) [MECSPEC] element. MEC has the capability of processing client requests and segregating control and user traffic at the edge of radio rather than sending all requests to the mobile gateway.

MEC transforms radio into an intelligent service edge that is capable of delivering highly personalized services directly from the edge of the network, while providing the best possible performance to the client. MEC can be an ideal candidate for ICN forwarder in addition to its usual function of managing mobile termination. In addition to MEC, other transport elements, such as routers, can work as ICN forwarders.

Data transport using ICN is different compared to IP-based transport. It evolves the Internet infrastructure by introducing uniquely named data as a core Internet principle. Communication in ICN takes place between content provider (producer) and end user (consumer) as described in figure 2.

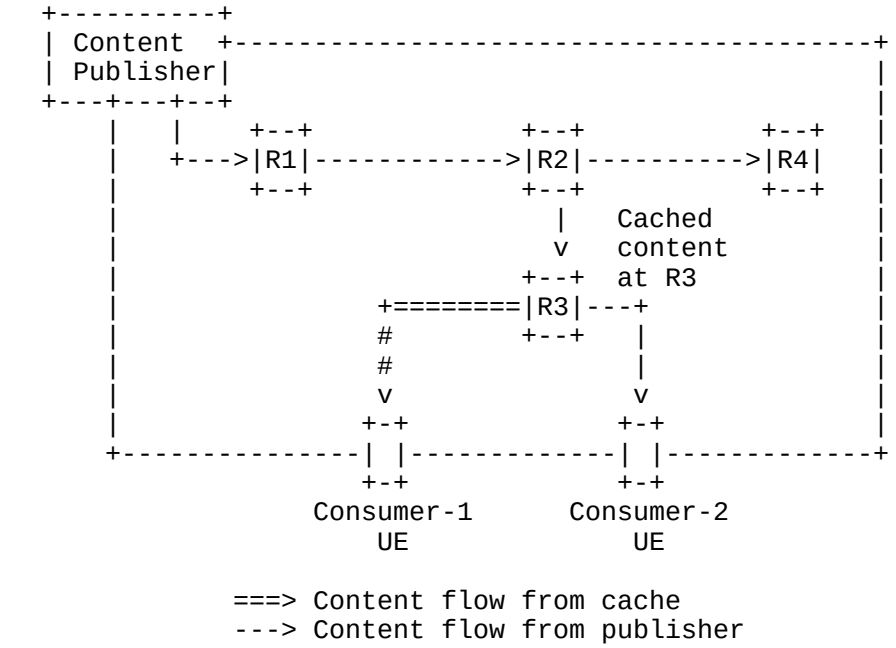

Fig. 2. ICN Architecture

Every node in a physical path between a client and a content provider is called ICN forwarder or router, and it has the ability to route the request intelligently and also cache the content so that it can be delivered locally for subsequent request from any other client. For mobile network, transport between a client and a content provider consists of radio network + mobile backhaul and IP core transport + Mobile Gateways + Internet + content data network (CDN).

In order to understand suitability of ICN for mobile networks, we will discuss the ICN framework describing protocols architecture and different types of messages, and then consider how we can use this in a mobile network for delivering content more efficiently. ICN uses two types of packets called "interest packet" and "data packet" as described in figure 3.

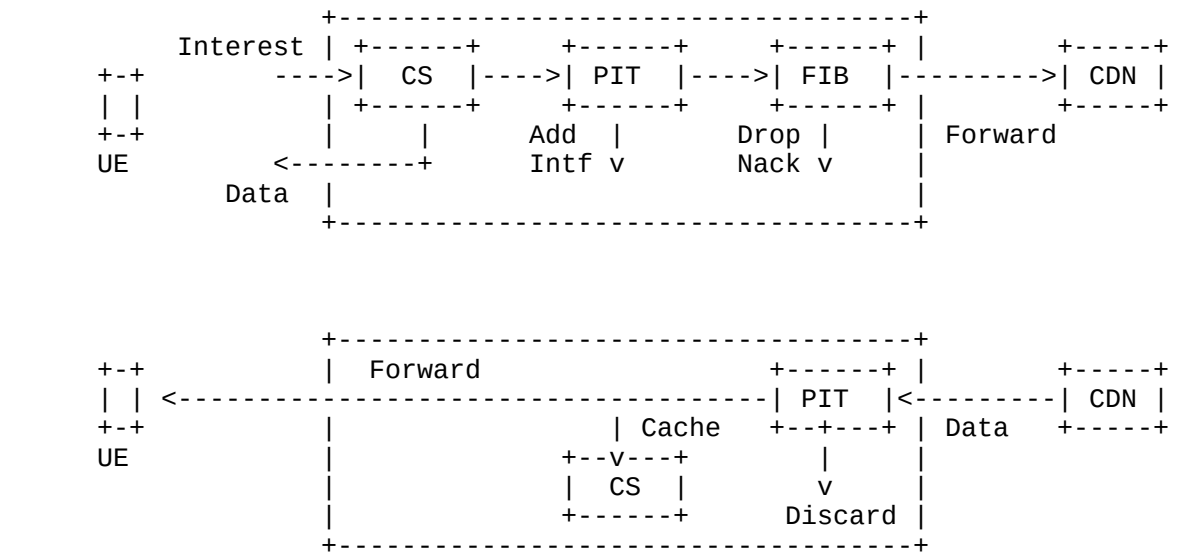

Fig. 3. ICN Interest, Data Packet and Forwarder

For LTE network, when a mobile device wants to get certain content, it will send an Interest message to the closest eNodeB.

Interest packet follows the TLV format [CCNxTLV] and contains mandatory fields such as name of the content and nonce. Name and nonce together uniquely identify an Interest packet. Nonce is also used to detect looping Interest messages. Interest packet also contains optional fields such as selector and guider fields. Selectors provides a specific filtering action during matching and

selection of the name prefixes. Guiders provides specific set of rules on how the Interest packet can be processed at the forwarder.

First ICN router will receive Interest packet and perform lookup if request for such content has come earlier from any other client. If yes, it is served from the local cache, otherwise request is forwarded to the next-hop ICN router. Each ICN router maintains three data structures, namely Pending Interest Table (PIT), Forwarding Information Base (FIB), and Content Store (CS). The Interest packet travels hop-by-hop towards content provider. Once the Interest reaches the content provider it will return a Data packet containing information such as content name, signature, signed key and data.

Data packet travels in reverse direction following the same path taken by the interest packet so routing symmetry is maintained. Details about algorithms used in PIT, FIB, CS and security trust models are described in various resources [CCN], here we explained the concept and its applicability to the LTE network.

5. ICN Deployment in 4G and LTE Networks

5.1 Recommendations in LTE Signaling

In this section we analyze signaling messages which are required for different procedures, such as attach, handover, tracking area update etc. The goal of analysis is to see if there is any benefit to replace IP-based protocols with ICN for LTE signaling in the current architecture. It is important to understand the concept of point of attachment (POA). When UE connects to a network it has at least three POAs:

1. eNodeb managing location or physical POA

2. Authentication and Authorization (MME, HSS) managing identity or authentication POA

3. Mobile Gateways (SGW, PGW) managing logical or session management POA.

In current architecture IP transport is used for all the messages associated with Control Plane for mobility and session management. IP is embedded very deeply into these messages and TLV carrying additional attributes as a layer-3 transport . Physical POA in eNodeB handles both mobility and session management for any UE attached to 4G, LTE network. The number of mobility management messages between different nodes in an LTE network per signaling procedure are given below in figure 4.

Normally two types of UE devices attach to LTE network: SIM based (need 3GPP mobility protocol for authentications) or non-SIM based (which connect to WiFi network), and authentication is required for both of these device types. For non-SIM based devices, AAA is used for authentication. We do not propose to change UE authentication procedures for user data transport using ICN, or any other mobility management messaging. A separate study would be required to analyze impact of ICN on mobility management messages structures and flows. We are merely analyzing the viability of implementing ICN as a transport for Control plane messages.

LTE Signaling Procedures	MME	HSS	SGW	PGW	PCRF
Attach	10	2	3	2	1
Additional default bearer	4	0	3	2	1
Dedicated bearer	2	0	2	2	1
Idle-to-connect	3	0	1	0	0
Connect-to-Idle	3	0	1	0	0
X2 handover	2	0	1	0	0
S1 handover	8	0	3	0	0
Tracking area update	2	0	0	0	0
Total	34	2	14	6	3

Fig. 4. Signaling Messages in LTE Gateways

It is important to note that even if we don’t implement ICN in MME and HSS, they still need to support either dual stack or native ICN UE capabilities. When UE initiates attach request using the identity as ICN, MME must be able to parse that message and create a session. MME forwards UE authentication to HSS so HSS must be able to authenticate ICN capable UE and authorize create session [TS23.401].

Anchorless mobility [ALM] has made some important comments on how mobility management is done in ICN. Author comments about handling mobility without having a dependency on the core network function e.g. MME. However, location update to the core network would still be required to support some of the legal compliance requirements such as lawful intercept and emergency services.

The main advantage of ICN is in caching and reusing the content, which does not apply to the transactional signaling exchange. After analyzing LTE signaling call-flows [TS23.401] and messages inter-dependencies [Fig 4], our recommendation is that it is not beneficial to deploy ICN for control plane and mobility management functions.

5.2 Recommendations in the User Plane

We will consider figure 1 to discuss different mechanisms to deploy ICN in mobile networks. We consider the following options:

1. Dual stack ICN deployment in UE

2. Native ICN Deployments in UE

3. ICN Deployment in eNodeB

4. ICN Deployment in mobile gateways (SGW/PGW)

5.2.1 Dual stack ICN Deployments in UE

The control and user plane communications in LTE, 4G mobile networks are specified in 3GPP documents [TS23.323] [TS23.203] [TS23.401]. It is important to understand that UE can be either consumer (receiving content) or publisher (pushing content for other clients). The protocol stack inside mobile device (UE) is complex as it has to support multiple radio connectivity access to eNodeB(s).

Figure 5 below provides high level description of protocol stack, where IP is defined at two layers: (1) at user plane communication, (2) Transport layer. User plane communication takes place between Packet Data Convergence Protocol (PDCP) and Application layer, whereas transport layer is at GTP protocol stack.

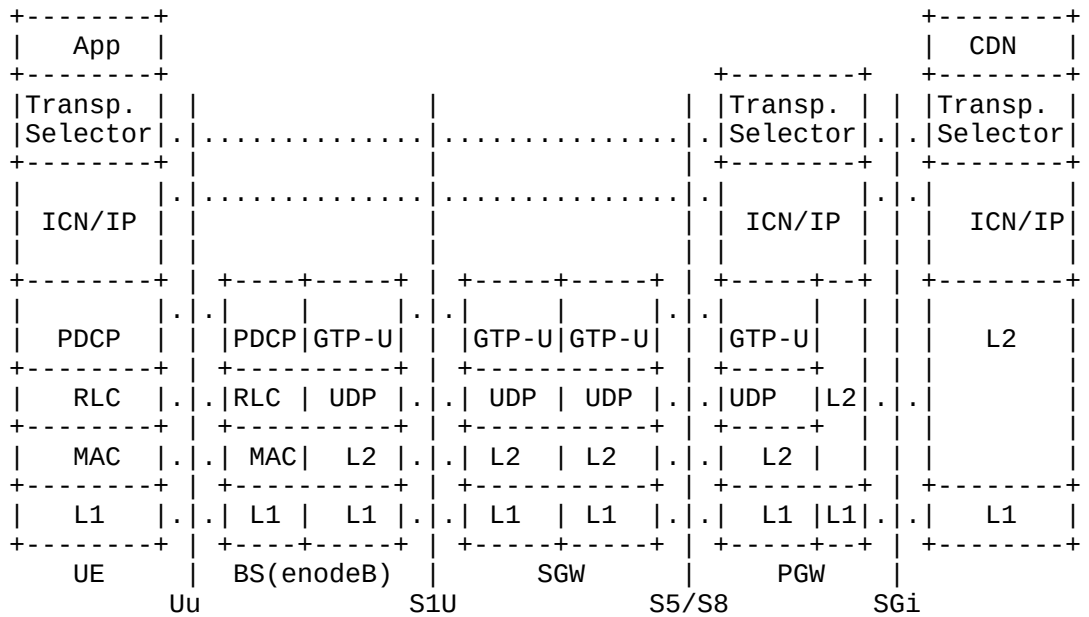

Fig. 5. Dual stack ICN Deployment in UE

The protocol interactions and impact of supporting tunneling of ICN packet into IP or to support ICN natively are described in figure 6 and figure 7 respectively.

The protocols and software stack used inside LTE capable UE support both 3G and LTE software interworking and handover. Latest 3GPP Rel.13 onward specification describe the use of IP and non-IP protocols to establish logical/session connectivity. We intend to leverage the non-IP protocol based mechanism to deploy ICN protocol stack in UE as well as in eNodeB and mobile gateways (SGW, PGW).

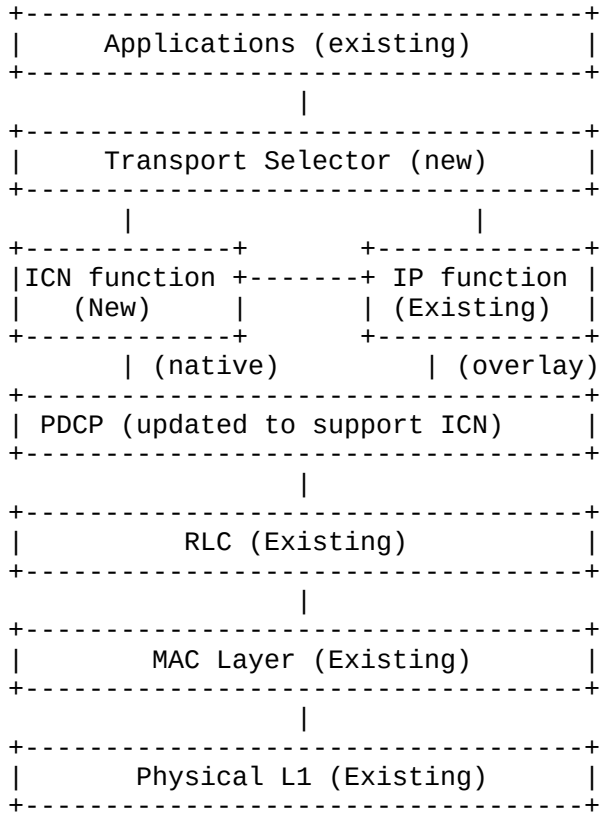

Fig. 6. Dual stack ICN protocol interactions

1. Application layer has the option of selecting either ICN or IP transport layer as well as the radio interface to send and receive data traffic. Our proposal is to provide a common Application Programming Interface (API) to the application

developers such that there is no impact on the application development when they choose either ICN or IP transport for exchanging the traffic with the network. We introduce a transport selector function to handle the interaction of application with the multiple transport options.

2. The transport selector helps determine what type of transport (e.g. ICN or IP) as well as type of radio interface (e.g. LTE or WiFi or both) is used to send and receive the traffic. Application layer can make the decision to select a specific transport based on preference e.g. content location, content type, content publisher, congestion, cost, quality of service etc. There can be an Application Programming Interface (API) to exchange parameters required for transport selection. The southbound interactions of Transport Selector will be either to IP or ICN at network layer.

3. ICN function (forwarder) is introduced in parallel to the existing IP layer. ICN forwarder contains functional capabilities to forward ICN packets, e.g. Interest packet to eNodeB or response "data packet" from eNodeB to the application.

4. For dual stack scenario, when UE is not supporting ICN at transport layer, we use IP underlay to transport ICN packets. ICN function will use IP interface to send Interest and Data packets for fetching or sending data using ICN protocol function. This interface will use ICN overlay over IP using any overlay tunneling mechanism.

5. To support ICN at network layer in UE, PDCP layer has to be aware of ICN capabilities and parameters. PDCP is located in the Radio Protocol Stack in the LTE Air interface, between IP (Network layer) and Radio Link Control Layer (RLC). PDCP performs following functions [TS36.323]:

   a) Data transport by listening to upper layer, formatting and pushing down to Radio Link Layer (RLC)

   b) Header compression and decompression using ROHC (Robust Header Compression)

   c) Security protections such as ciphering, deciphering and integrity protection

   d) Radio layer messages associated with sequencing, packet drop detection and re-transmission etc.

6. No changes are required for lower layer such as RLC, MAC and Physical (L1) because they are not IP aware.

One key point to understand in this scenario is that ICN is deployed as an overlay on top of IP.

5.2.2 Native ICN Deployments in UE

We propose to implement ICN natively in UE by modifying PDCP layer in 3GPP protocols. Figure 7 below provides a high level protocol stack description where ICN is used at two different layers:

1. at user plane communication

2. at transport layer

User plane communication takes place between PDCP and application layer, whereas transport layer is a substitute of GTP protocol. Removal of GTP protocol stack is significant change in mobile architecture because GTP is used not just for routing but for mobility management functions such as billing, mediation, policy enforcement etc.

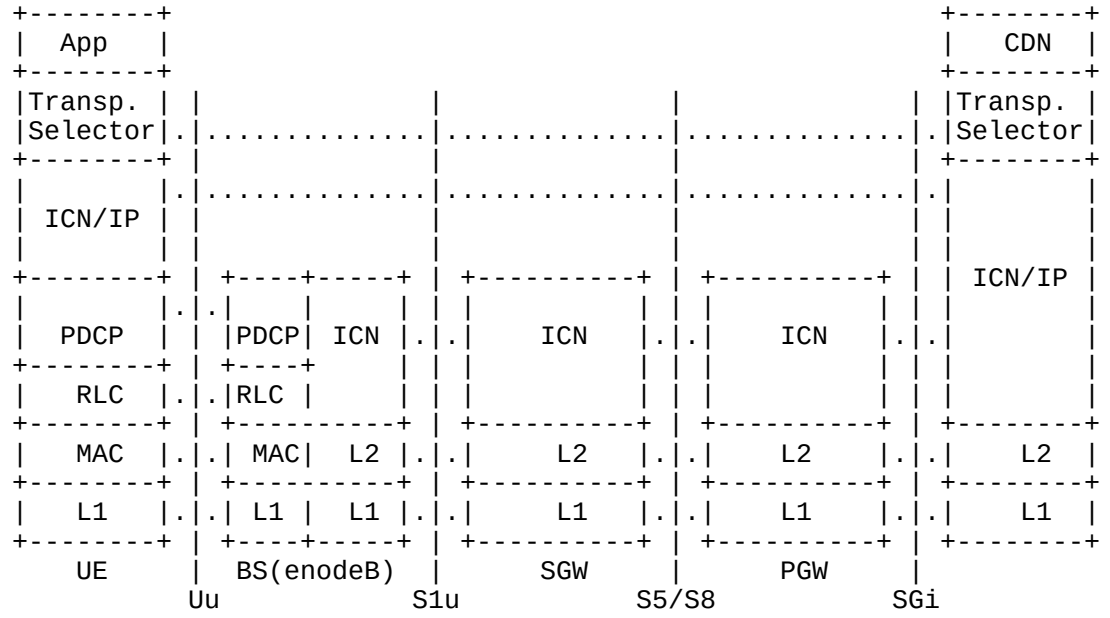

Fig. 7. Native ICN Deployment in UE

If we implement ICN natively in UE, communication between UE and

eNodeB will change and also we will not need to tunnel user plane traffic from eNodeB to mobile packet core (SGW, PGW) using GTP tunnel.

For native ICN deployment, Application is configured to use ICN forwarder so there is no need for Transport Selector. Also to support ICN at network layer in UE, we need to modify existing PDCP layer. PDCP layer has to be aware of ICN capabilities and parameters.

Native implementation will also provide opportunities to develop new use cases leveraging ICN capabilities such as seamless mobility, UE to UE content delivery using radio network without interactions with mobile gateways, etc.

5.2.3 ICN Deployment in eNodeB

eNodeB is physical point of attachment for UE, where radio protocols are converted into IP transport protocol as depicted in figure 6 and figure 7 for dual stack/overlay and native ICN respectively. When UE performs attach procedures, it is assigned an identity either as IP, dual stack (IP and ICN), or ICN. UE can initiate data traffic using any of three different options:

1. Native IP (IPv4 or IPv6)

2. Native ICN

3. Dual stack IP (IPv4/IPv6) or ICN

UE encapsulates user data transport request into PDCP layer as described in section 5.2.1 and send the information on air interface to eNodeB. eNodeB receives the information and using PDCP [TS 36.323], de-encapsulate air-interface messages and convert them to forward to core mobile gateways (SGW, PGW). In order to support ICN natively in eNodeB, it is proposed to provide transport selector capabilities in eNodeB (similar as provided in UE), which provides following functions:

1. It decides the forwarding strategy for user data request coming from UE. The strategy can make decision based on preference indicated by the application such as, congestion, cost, quality of service, etc.

2. eNodeB to provide open Application Programming Interface (API) to external management systems to provide programming capability to eNodeB to program the forwarding strategies.

3. eNodeB shall be upgraded to support three different types of

transport IP, ICN, and dual stack IP+ICN towards mobile gateways, as depicted in figure 8. It is also recommended to deploy IP and/or ICN forwarding capabilities into eNodeB for efficient transfer of data between eNodeB and mobile gateways. There can be four choices for forwarding data request towards mobile gateways:

a) Assuming eNodeB is IP enabled and UE requests for IP transfer, eNodeB forwards data over IP.

b) Assuming eNodeB is ICN enabled and UE request for ICN transfer, eNodeB forwards data over ICN.

c) Assuming eNodeB is IP enabled and UE request ICN, eNodeB overlays ICN on IP and forwards the user plane traffic over IP.

d) Assuming eNodeB is ICN enabled and UE request IP, eNodeB overlays IP on ICN and forwards the user plane traffic over ICN [IPoICN].

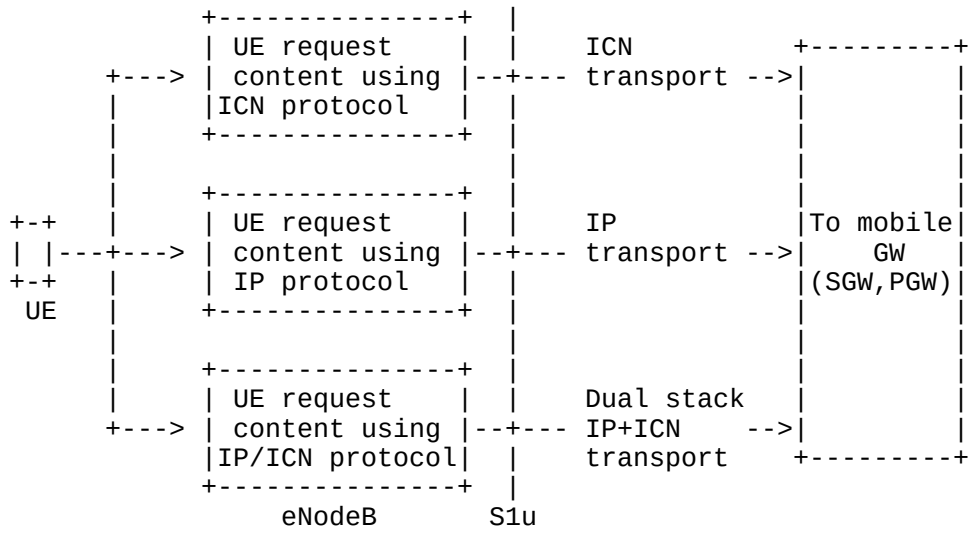

Fig. 8. Native ICN Deployment in eNodeB

5.2.4 ICN Deployment in Packet Core (SGW, PGW) Gateways

Mobile gateways a.k.a. Evolved Packet Core (EPC) include SGW, PGW, which performs session management for UE from the initial attach to disconnection. When UE is powered on, it performs NAS signaling and after successful authentication it attaches to PGW. PGW is an

anchoring point for UE and responsible for service creations, authorization, maintenance etc. Entire functionality is managed using IP address(es) for UE.

In order to implement ICN in EPC, the following functions are needed.

1. Insert ICN function at session management layer as additional functionality with IP stack. Session management layer is used for performing attach procedures and assigning logical identity to user. After successful authentication by HSS, MME sends create session request (CSR) to SGW and SGW to PGW.

2. When MME sends Create Session Request message (step-12 in [TS23.401]) to SGW or PGW, it contains Protocol Configuration Option Information Element (PCO IE) containing UE capabilities. We can use PCO IE to carry ICN related capabilities information from UE to PGW. This information is received from UE during initial attach request in MME. Details of available TLV which can be used for ICN are given in subsequent sections. UE can support either native IP, or ICN+IP, or native ICN. IP is referred to as both IPv4 and IPv6 protocols.

3. For ICN+IP capable UE, PGW assigns the UE both IP address and ICN identity. UE selects the either of the identities during initial attach procedures and registers with network for session management. For ICN capable UE it will provide only ICN attachment. For native IP capable UE there is no change.

4. In order to support ICN capable attach procedures and use ICN for user plane traffic, PGW needs to have full ICN protocol stack functionalities. Typical ICN capabilities include functions such as content store (CS), Pending Interest Table (PIT), Forwarding Information Base (FIB) capabilities etc. If UE requests ICN in PCO IE, then PGW registers UE with ICN names. For ICN forwarding, PGW caches content locally using CS functionality.

5. Protocol configuration options information elements described in [TS24.008] (see Figure 10.5.136 on page 598) and [TS24.008] (see Table 10.5.154 on page 599) provide details for different fields.

   - Octet 3 (configuration protocols defines PDN types) which contains details about IPv4, IPv6, both or ICN.

   - Any combination of Octet 4 to Z can be used to provide additional information related to ICN capability. It is most important that PCO IE parameters are matched between UE and

mobile gateways (SGW, PGW) so that they can be interpreted properly and UE can attach successfully.

6. Deployment of ICN functionalities in SGW and PGW should be matched with UE and eNodeB because they will exchange ICN protocols and parameters.

7. Mobile gateways SGW, PGW will also need ICN forwarding and caching capability.

6. Security Considerations

To ensure only authenticated UEs are connected to the network, LTE mobile network implements various security mechanisms. From perspective of ICN deployment in user plane, it need to take care of following security aspects:

1. UE authentication and authorization

2. Radio or air interface security

3. Denial of service attacks on mobile gateway, services

4. Content positioning either in transport or servers

5. Content cache pollution attacks

6. Secure naming, routing, and forwarding

7. Application security

Security over the LTE air interface is provided through cryptographic technique. When UE is powered-up it performs key exchange between UE’s USIM and HSS/Authentication Center using NAS messages including ciphering and integrity protections between UE and MME. Details of security UE authentication, key exchange, ciphering and integrity protections message are given in 3GPP call flow [TS23.401].

LTE is an all IP network and uses IP transport in its mobile backhaul (e.g. between eNodeB and core network). In case of provider owned backhaul, it may not implement security mechanisms; however, they are necessary in case it uses shared or a leased network. The native IP transport continue to leverage security mechanism such as Internet key exchange (IKE) and the IP security protocol (IPsec). More details of mobile backhaul security are provided in 3GPP network security [TS33.310] and [TS33.320]. When mobile backhaul is upgraded to support dual stack (IP+ICN) or native ICN, it is required to implement security techniques which are deployed in mobile backhaul.

When ICN forwarding is enabled on mobile transport routers, we need to deploy security practices based on RFC7476 and RFC7927.

Some of the key functions supported by LTE mobile gateway (SGW, PGW) are content based billing, deep packet inspection (DPI), lawful intercept (LI). For ICN based user plane traffic, it is required to integrate ICN security for session between UE and gateway; however, in ICN network, since only consumers are in possession of decryption keys can access the content, some of the services provided mobile gateway mentioned above may not work. Further research in this area is needed.

7. Summary

Authors have discussed complexities of LTE network and key dependencies for deploying ICN in user plane data transport. Different deployment options described covers aspects such as inter-operability and multi-technology, which is a reality for any service provider. The ICN deployment options described in section 5, we currently evaluating using LTE gateway software and ICN simulator. One can deploy ICN for data transport in user plane either as an overlay, dual stack (IP + ICN) or natively (by integrating ICN with CDN, eNodeB, SGW, PGW and transport network etc.). It is important to understand that for the actual deployment scenarios, additional research study is required to identify dependencies specific to a mobile network.

As far as control plane signaling is concerned, our observation is that further research is required to understand the benefits of using ICN to complement or replace traditional control plane signaling. As a starting step towards ICN deployment, it is recommended that the focus should be on enhancement of user data plane.

Mobile Edge Computing (MEC) [CHENG] provides capabilities to deploy functionalities such as content data Network (CDN) caching and mobile user plane functions (UPF) [TS23.501]. Recent research for delivering real-time video content using ICN has also been proven to be efficient [NDNRTC] and we can be used towards realizing the benefits of ICN deployment in eNodeB, MEC, mobile gateways (SGW, PGW) and CDN. The key aspect for ICN is in its seamless integration in LTE and 5G networks with tangible benefits so that we can optimize content delivery using simple and scalable architecture. Authors will continue to explore how the ICN forwarding in MEC could be used in efficient delivery of unicast and multicast traffic.

7 References

7.1 Normative References

[GRAYSON] Grayson M, Shatzkamer K, Wainner S.; Cisco Press book "IP Design for Mobile Networks" by. page 108-112.

[IPSEC1] Cisco IPSec overhead calculator tool <https://cway.cisco.com/tools/ipsec-overhead-calc/ipsec-overhead-calc.html>.

[IPSEC2] IPSec Bandwidth Overhead Using AES <http://packetpushers.net/ipsec-bandwidth-overhead-using-aes/>.

[BROWER] Brower, E.; Jeffress, L.; Pezeshki, J.; Jasani, R.; Ertekin, E. "Integrating Header Compression with IPsec", Military Communications Conference, 2006. MILCOM 2006. IEEE, On page(s): 1 - 6.

[TS25.323] 3GPP TS25.323 Rel. 14 (2017-03) Packet Data Convergence Protocol (PDCP) specification.

[TS23.501] 3GPP TS23.501 Rel. 15 (2017-06) System Architecture for the 5G System.

[TS23.203] 3GPP TS23.203 Rel. 14 (2017-03) Policy and charging control and QoS architecture

[TS23.401] 3GPP TS23.401 Rel. 14 (2017-03) E-UTRAN Access procedures architecture

[TS33.310] 3GPP TS33.310 Rel. 14 (2016-12) LTE Network Domain Security (NDS); Authentication Framework (AF)

[TS33.320] 3GPP TS33.320 Rel. 14 (2016-12) Security of Home Node B (HNB) / Home evolved Node B (HeNB)

[TS24.008] 3GPP TS24.008 Rel. 14 (2017-06) Mobile radio interface Layer 3 specification.

[TS23.501] 3GPP TS23.501 Rel. 14 (2017-06) System Architecture for the 5G System

[TS23.214] 3GPP TS23.214 Rel. 14 (2017-06) Architecture enhancements for control and user plane separation of EPC nodes

[TS36.323] 3GPP TS36.323 Rel. 14 (2017-06) Evolved Universal

Terrestrial Radio Access (E-UTRA); Packet Data Convergence Protocol (PDCP) specification

[RFC7476] Information-Centric Networking: Baseline Scenarios

[RFC7927] Information-Centric Networking (ICN) Research Challenges

7.2 Informative References

[MECSPEC] European Telecommunication Standards Institute (ETSI) MEC specification ETSI-GS-MEC-IEG-001 V1.1.1 (2015-11).

[NDNTLV] NDN Interest Packet Format Specification 0.2-2. https://named-data. net/doc/ndn-tlv/interest.html.

[CCNxTLV] CCNx Messages in TLV Format https://datatracker.ietf.org/doc/draft-irtf-icnrg-ccnxmessages/

[NDNPUB] Named Data Networking <http://named-data.net/publications/>.

[CCN] Content Centric Networking <http://www.ccnx.org and http://blogs.parc.com/ccnx/documentation-guide/>.

[NDN] Lixia Z., Lan W. et al. SIGCOMM Named Data Networking

[ALM] J. Aug'e, G. Carofiglio et al. "Anchor-less producer mobility in icn," in Proceedings of the 2Nd ACM Conference on Information-Centric Networking, ACM-ICN '15, pp. 189-190, ACM, 2015.

[VNIIDX] Cisco Visual Networking Index (VNI) dated 16 Feb 2016, <http://www.cisco.com/c/en/us/solutions/service-provider/visual-networking-index-vni/index.html>.

[NDNRTC] Peter Gusev,Zhehao Wang, Jeff Burke, Lixia Zhang et. All, IEICE Trans Communication, RealtimeStreaming Data Delivery over Named Data Networking, Vol E99-B, No.5 May 2016.

[CHENG] Chengchao L., F. Richard Yu, Information-centric network fucntion virtualization over 5G mobile wireless networks, IEEE network (Volume:29, Issue:3), page 68-74, 01 June 2015.

[NGMN] Backhaul Provisioning for LTE-Advanced & Small Cells

              <https://www.ngmn.org/uploads/media/150929_NGMN_P-
              SmallCells_Backhaul_for_LTE-Advanced_and_Small_Cells.pdf>

   [IPoICN]   IP Over ICN - The Better IP?
              <http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=7194109>

Authors' Addresses

   Prakash Suthar
   9501 Technology Blvd.
   Rosemont, Illinois 50618

   EMail: psuthar@cisco.com

   Milan Stolic
   9501 Technology Blvd.
   Rosemont, Illinois 50618

   EMail: mistolic@cisco.com

   Anil Jangam
   3625 Cisco Way
   San Jose, CA 95134
   USA

   Email: anjangam@cisco.com

ICN Research Group
Internet-Draft
Intended status: Informational
Expires: December 28, 2017

Y. Zhang
D. Raychadhuri
WINLAB, Rutgers University
L. Grieco
Politecnico di Bari (DEI)
E. Baccelli
INRIA
J. Burke
UCLA REMAP
R. Ravindran
G. Wang
Huawei Technologies
A. Lindgren
B. Ahlgren
RISE SICS
O. Schelen
Lulea University of Technology

June 26, 2017

Design Considerations for Applying ICN to IoT
draft-zhang-icnrg-icniot-01

Abstract

The Internet of Things (IoT) promises to connect billions of objects to the Internet. After deploying many stand-alone IoT systems in different domains, the current trend is to develop a common, "thin waist" of protocols over a horizontal unified, defragmented IoT architecture. Such an architecture will make objects accessible to applications across organizations and domains. Towards this goal, quite a few proposals have been made to build an application-layer based unified IoT platform on top of today’s host-centric Internet. However, there is a fundamental mismatch between the host-centric nature of todays Internet and mostly information-centric nature of the IoT system. To address this mismatch, an information-centric network (ICN) architecture can provide a common set of protocols and services, called ’ICN-IoT’, which can be used to build IoT platforms. ICN-IoT leverages the salient features of ICN, and thus provides naming, security, mobility support,scalability, and efficient content and service delivery.

This draft summarizes general IoT demands, and covers the challenges and design considerations ICN faces to realize a ICN-IoT framework based on ICN architecture.

Status of This Memo

Copyright Notice

Table of Contents

1. IoT Motivation

During the past decade, many Internet of Things (IoT) systems have been developed and deployed in different domains. The recent trend, however, is to evolve towards a more unified IoT architecture, in which a large number of objects connect to the Internet, available for interactions among themselves, as well as interactions with many different applications across boundaries of administration and domains. General IoT applications involve sensing, processing, and secure content distribution occurring at various timescales and at multiple levels of hierarchy depending on the application requirements. This requires the system to adopt a unified architecture providing pull, push and publish/subscribe mechanisms using application abstractions, common naming, payload, encryption and signature schemes. This requires open APIs to be generic enough to support commonly used interactions between consumers, content producer, and IoT services, as opposed to proprietary APIs that are common in today's systems. Building a unified IoT architecture, however, poses great challenges on the underlying network and systems. To name a few, it needs to support 50-100 Billion networked objects [1], many of which are mobile. The objects will have

extremely heterogeneous means of connecting to the Internet, often with severe resource constraints. Interactions between the applications and objects are often real-time and dynamic, requiring strong security and privacy protections. In addition, many IoT applications are inherently information centric (e.g., data consumers usually need data sensed from the environment without any reference to the sub-set of sensors that will provide the asked information).

Taking a general IoT perspective, we first motivate the discussion of ICN for IoT using well known scenarios. Then we discuss the IoT requirements generally applicable to many well known IoT scenarios. We then discuss how the current application-layer unified IoT architectures fail to meet these requirements. We follow this by key ICN features that makes it a better candidate to realize an unified IoT framework. We then discuss IoT design challenges from an ICN perspective and requirements posed towards its design.

2. Motivating ICN for IoT

ICN offers many features including name-based networking, content object security, caching, computing and storage, mobility, context-aware networking (see Section 3.6) and support for ad hoc networking features, all of which have to be realized in an application-specific means in the context of IP-IoT. These compelling features enable a distributed and intelligent data distribution platform to support heterogeneous IoT services with features like device bootstrapping with minimal configuration, simpler protocols to aid self-organizing among the IoT elements, natural support for compute and caching logic at strategic points in the network. We discuss these features through the following scenarios that are difficult to realize over IP today, and whose characteristics we argue match the features offered by ICN.

- Smart Mobility: Smarter end-user devices and Machine-to-Machine (M2M) connection are undergoing a significant growth. By 2021, there will be more than 10 billion mobile devices and connection, including smartphones, tablets, wearables, vehicles [1]. Involved fields range from medical and healthcare, fitness, clothing, to environmental monitoring [40]. In particular, one of the most affected domain is transportation and the so-called Intelligent Transport Systems (ITS) [42]. It aims at providing multi-modal transportation, embracing public and private municipal, regional, national, trans-national vehicles and fleets. This extremely heterogeneous eco-system of means of transportation is made available to users and citizens through advanced services. These services are able to fulfill usability requirements while pursuing system level objectives, thus including: (i) the reduction of the CO2 footprint, (ii) the real-time delivery of specific goods,

(iii) the reduction of traffic within urban areas, (iv) the provisioning of pleasant journeys to tourists, and (v) the general commitment of satisfactory travel time and experience [117]. In this context, IoT technologies can play a pivotal, in particular, Traffic Management Systems (TMS) aided by IoT technologies are creating novel approaches to traffic modeling [47]. Moreover, such features enable advanced design paradigms (e.g., Mobility as a Service (MaaS) [39]) with huge implications in systems architectures [48]. As a consequence, smart mobility support can be a significant use case of ICN-IoT. The important ICN features that corroborate mobility support are:

* The location independence of content allows one to manage consumer mobility in a simpler way than IP. Different from Mobile IP, that needs 'triangular routing' to locate moving hosts, ICN envisions that the consumers just needs to re-issue content requests after changing the attachment point [43];

* Since content is not bound to a specific location, it can be cached anywhere in the network. This caching mechanism adds redundancy to the system. Therefore, if the producer loses connectivity while it is moving, a content request can be resolved to an intermediate node en-route or routed towards a caching node [43];

* The content request-response communication paradigm decouples publications and subscriptions in time and space. Therefore, entities involved are not aware of each other and do not need to be connected at the same time [44];

* The use of in-network Name Resolution Service design allows to identify content name's current location in the network, thanks to its network function of updating named entity location information [56].

From a technological perspective, open challenges are: (i) interoperability across different IoT technologies; (ii) namespace design able to harmonize ITS standards; (iii) scalable data-sharing model across real-time (and non real-time) traffic sources; (iv) definition of travel-centric services based on ICN-IoT; (v) seamless support to mobility; (vi) content authentication and cryptography.

o Smart Building: Buildings are gaining smart capabilities that allow to enhance comfort, provide safety and security, manage efficiently energy [101]. In particular, smart buildings are no longer simple energy consumer, but can also be prosumers with on-site energy generation systems. These systems can improve

building’s usability towards: (i) Smart heating, ventilation, and air conditioning (HVAC), (ii) Smart lightings, (iii) Plug loads, (iv) Smart windows. The main requirements of those sub-systems are [101]: (i) context awareness; (ii) resource-constrained devices; (iii) interoperability across heterogeneous technologies; (iv) security and privacy protection. The ICN paradigm could ease the fulfillment of those requirements because, usually, smart building services are information centric by design: this means that every time an autonomic management loop is established within the smart building to control some physical variables of interest, the information exchanged between users, sensors, actuators, and controllers do not immediately translate to specific nodes within the building but could be provided by multiple sensors / gateways. The relevance of ICN in Smart Building is recognized in literature with reference to the several frameworks deployed in various environments. For instance, in [61], nodes are distributed in different rooms, floors, and buildings of a campus university and their energy consumption and individual behavior are monitored. Smart home application is investigated in [103], by evaluating data retrieval delay and data packet loss. Moreover, [104] designed and tested lighting control over NDN in a theater. In this context, specific ICN challenges are: (i) design of a scalable namespace for uniquely identifying the information of interest, (ii) data-sharing model across heterogeneous systems, (iii) self-organizing functionalities for improving network connections between end-nodes, utilities and the control center, (iv) authentication procedures to grant data confidentiality and integrity.

- Smart Grid: Smart Grids are increasingly transforming into cyber-physical systems [18] with the goal of maximum automation towards efficiency and minimal human intervention. The system is very complex comprising of power distribution grids, end user applications (e.g. EV charging systems, appliances etc), smart monitoring systems (spanning end user and the power grids), heterogeneous energy producing sources (including prosumers), and load distribution and balancing systems. Current smart grid systems are managed using Supservisory Control and Data Acquisition (SCADA) frameworks that are centralized and highly restrictive unidirectional communication support [19]. Hence the requirement is towards : 1) greater flexibility to distribute the energy from the feeder through real-time reconfiguration of multiple monitoring devices (e.g. phasor measurement units (PMUs)), and management operations which require efficient data delivery infrastructure; 2) large scale data delivery infrastructure, which also include latency sensitive applications, inter-connecting heterogeneous smart grid producing, monitoring and consuming end user devices; 3) Resiliency,which is critical to

the operation and protection of the grid; 4)Security, to protect mission critical grid applications from network intrusions ; 5) understanding machine-to-machine traffic patterns for optimal placement of storage and computing for maximum efficiency. Smart grids can benefit from ICN in the following ways [20] :

* Smart grid will benefit from naming content than hosts, as it is more likely that data generated by one subsystem will be useful for multiple entities. Further, naming content allows to enable many-to-many model of communication, which is very in-efficient in host-centric architectures.

* ICN features of in-network computing, storage and caching will enable better use of network resources and benefit diverse application needs varying from applications that has low bitrate and is latency tolerant (e.g. smart grid and energy pricing) to higher data rate ones with stringent delay/ disruption requirements (e.g. synchrophasor measurements). Also it is typical in smart grid systems to have applications consuming the same data at different rates in which case in-network caching and computing could help.

* Host-centric networking exposes a mission critical infrastructure like smart grid infrastructure to intrusion and DOS attacks, this is directly related to exposing the IP addresses of critical applications and subsystems. Naming content, service or device de-couples it from the location, reducing the exposure to target a specific smart grid subsystem based on a geographical context.

* ICN’s name based networking offers the potential for self-configuration both during bootstrapping and during the regular operation of the grid allowing scalable operation and self-recovery during faults or maintenance tasks in the system.

o Smart Industrial Automation : In a smart and connected industry environment, there is a multitude of equipment with sensors that generate large volumes of data during normal operation. This range from highly time-critical data for real-time control of production processes, to less time-critical data that is collected to central cloud environment for control room monitoring, to pure log data without latency requirements that is mainly kept for a posteriori analysis. Industrial wireless networks are harsh environments with lots of potential interference at the same time as hard reliability and real-time requirements are placed by many applications. This means that available network capacity is not always high, so congestion is likely to be experienced by traffic with less stringent timing requirements. One such example is when

errors occur in the production process, a mobile workforce will need to investigate the problem on-site and will need high resolution data from the faulty machine as well as other process data from other parts of the plant. The mobile workforce will locally perform diagnostics or maintenance and they rely on the information from the production system both for safety and to solve any issues in the plant. They rely on both historical data in order to pinpoint the root cause of the problems, as well as the current data flows in order to assess the present state of the equipment under control. High resolution measurements are generated close to the mobile workforce while the historic data has to be retrieved from the historian servers. Multiple workers involved in the process will access the same data, possibly with a slight time-shift. The network thus need to support a mobile users to get access to data flows in a way suitable for their physical location and task requirements. Introducing ICN functionality into the system can introduce several benefits that will enhance the working experience and productivity for the mobile workforce.

* When using ICN, naming of data can be done in a way that corresponds well to the current names often used in industrial scenarios as the hierarchical names defined by OPC Foundation [10] maps well to the CCN/NDN name space.

* ICN provides the possibility to get newest data without knowing the location of the caches or whether a particular piece of data is available locally or in a central repository. Also gives the possibility to get either local high-resolution data or remote low-resolution data (no need to store all data centrally, which is maybe not even possible due to large data volumes). May require known naming conventions or routing policies that can route interests to the right location.

* Reduces network usage as unnecessary data is not transmitted, and data accessed by multiple workers is only sent once.

* Workforce mobility between different access points in the factory is inherently supported without the need to maintain connection state.

* Removing tedious configurations in clients since that is provided by the infrastructure.

* Allow sharing of large data volumes between users that are in physical proximity without introducing additional traffic on the backbone.

* Caching of data means avoiding database accesses to a distributed redundant database in the central infrastructure with consistency requirements.

3. IoT Architectural Requirements

A unified IoT platform has to support interactions among a large number of mobile devices across the boundaries of organizations and domains. As a result, it naturally poses stringent requirements in every aspect of the system design. Below, we outline a few important requirements that a unified IoT platform has to address.

3.1. Naming

An important step towards realizing a unified IoT architecture is the ability to assign names that are unique to each device, data items generated by these devices, or a group of devices towards a common objective. Naming has the following requirements. Firstly, names need to be persistent against dynamic features that are common in IoT systems, such as lifetime, mobility or migration. Secondly, names that are derived from the keys need to be self-certifying, for both device-centric communication and content-centric communication. For device-centric communication, the binding between device names and the device must be secure. For content-centric communication, the binding between the names and the content has to be secure. Thirdly, names usually serve multiple purposes: routing, security (self-certifying) or human-readability. For IoT applications, the choice of flat versus human readable names needs to be made considering application and network requirements such as privacy and network level scalability, and the name space explosion that may occur because of complex relationship between name hierarchies [120] which might confound application logic. In order to ensure the trustworthiness of the names, a name certificate service (NCS) needs to be considered. Such a service acts as a certificate authority in assigning names, which are themselves public keys or appropriately bound to the name for verification at the consumer’s end. In short, the NCS must provide services analogous to those provided by a Public Key Infrastructure (PKI). In ICN, users may either generate their own public keys and submit them to the NCS for registration, or may contact the NCS to acquire public keys. Consequently, the NCS publishes approved cryptographic suites, object categories and object description formats, as well as allows users to self-certify themselves.

3.2. Security and Privacy

A variety of security and privacy concerns exist in IoT. For example the unified IoT architecture makes physical objects accessible to applications across organizations and domains. Further, it often integrates with critical infrastructure and industrial systems with life safety implications, bringing with it significant security challenges and regulatory requirements [13], as will be discussed in Section 6.3. Security and privacy thus become a serious concern, as does the flexibility and usability of the design approaches. Beyond the overarching trust management challenge, security includes data integrity, authentication, and access control at different layers of the IoT architecture. Privacy includes several aspects: (1) privacy of data producer/consumer that is directly related to each individual vertical domain such as heath, electricity, etc., (2) privacy of data content, and (3) privacy of contextual information such as time and location of data transmission [65].

3.3. Scalability

Cisco predicts there will be around 50 Billion IoT devices such as sensors, RFID tags, and actuators, on the Internet by 2020 [1]. As mentioned above, a unified IoT platform needs to name every entity such as data, device, service etc. Scalability has to be addressed at multiple levels of the IoT architecture including naming, security, name resolution, routing and forwarding level. Mobility adds further challenge in terms of scalability. Particularly with respect to name resolution the system should be able to register/update/resolve a name within a short latency. In addition scalability is also affected because of IoT system specific features such as IoT resource object count, state and rate of information updates generated by the sensing devices.

3.4. Resource Constraints

IoT devices can be broadly classified as type 1, type 2, and type 3 devices, with type 1 the most resource-constrained and type 3 the most resource-rich [45]. In general, there are the following types of resources: power, computing, storage, bandwidth, and user interface.

Power constraints of IoT devices limit how much data these devices can communicate, as it has been shown that communications consume more power than other activities for embedded devices [46]. Flexible techniques to collect the relevant information are required, and uploading every single produced data to a central server is undesirable. Computing constraints limit the type and amount of processing these devices can perform. As a result, more complex

processing needs to be conducted in cloud servers or at opportunistic points, example at the network edge, hence it is important to balance local computation versus communication cost.

Storage constraints of the IoT devices limit the amount of data that can be stored on the devices. This constraint means that unused sensor data may need to be discarded or stored in aggregated compact form time to time. Bandwidth constraints of the IoT devices limit the amount of communication. Such devices will have the same implication on the system architecture as with the power constraints; namely, we cannot afford to collect single sensor data generated by the device and/or use complex signaling protocols. It is also worth mentioning that idle chatter in the background is strongly discouraged to maintain connectivity or other volatile state.

User interface constraints refer to whether the device is itself capable of directly interacting with a user should the need arise (e.g., via a display and keypad or LED indicators) or requires the network connectivity, either global or local, to interact with humans.

The above discussed device constraints also affect application performance with respect to latency.

3.5. Traffic Characteristics

IoT traffic can be broadly classified into local area traffic and wide area traffic. Local area traffic is among nearby devices. For example, neighboring cars may work together to detect potential hazards on the highway, sensors deployed in the same room may collaborate to determine how to adjust the heating level in the room. These local area communications often involve data aggregation and filtering, have real time constraints, and require fast device/data/service discovery and association. At the same time, the IoT platform has to also support wide area communications. For example, in Intelligent Transportation Systems, re-routing operations may require a broad knowledge of the status of the system, traffic load, availability of freights, whether forecasts and so on. Wide area communications require efficient data/service discovery and resolution services.

While traffic characteristics for different IoT systems are expected to be different, certain IoT systems have been analyzed and shown to have comparable uplink and downlink traffic volume in some applications such as [2], which means that we have to optimize the bandwidth/energy consumption in both directions. Further, IoT traffic demonstrates certain periodicity and burstiness [2]. As a

result, when provisioning the system, the shape of the traffic volume has to be properly accounted for.

3.6. Contextual Communication

Many IoT applications rely on dynamic contexts in the IoT system to initiate, maintain and terminate communication among IoT devices. Here, we refer to a context as attributes applicable to a group of devices that share some common features, such as their owners may have a certain social relationship or belong to the same administrative group, or the devices may be present in the same location. There are two types of contexts: long-term quasi static contexts and short-term dynamic contexts. In this draft, we focus on the latter, which are more challenging to support, requiring fast formation, update, lookup and association For example, cars traveling on the highway may form a "cluster" based upon their temporal physical proximity as well as the detection of the same event. These temporary groups are referred to as contexts. IoT applications need to support interactions among the members of a context, as well as interactions across contexts.

Temporal context can be broadly categorized into two classes, long-term contexts such as those that are based upon social contacts as well as stationary physical locations (e.g., sensors in a car/ building), and short-term contexts such as those that are based upon temporary proximity (e.g., all taxicabs within half a mile of the Time Square at noon on Oct 1, 2013). Between these two classes, short-term contexts are more challenging to support, requiring fast formation, update, lookup and association.

3.7. Handling Mobility

There are several degrees of mobility in a unified IoT architecture, ranging from static as in fixed assets to highly dynamic in vehicle-to-vehicle environments.

Mobility in the IoT architecture can mean 1) the data producer mobility (i.e., location change), 2) the data consumer mobility, 3) IoT Network mobility (e.g., a body-area network in motion as a person is walking); and 4) disconnection between the data source and destination pair (e.g., due to unreliable wireless links). The requirement on mobility support is to be able to deliver IoT data below an application’s acceptable delay constraint in all of the above cases, and if necessary to negotiate different connectivity or security constraints specific to each mobile context. More detailed discussions are presented in Section 6.7.

3.8. Storage and Caching

Storage and caching plays a very significant role depending on the type of IoT ecosystem, also a function subjected to privacy and security guidelines. Caching is usually done for increasing data availability in the network and reliability purposes, especially in wireless scenarios in the network access. Storage is more important for IoT, storing data for long term analysis. Data is stored in strategic locations in the network to reduce control and computation overhead. In a unified IoT architecture, depending on application requirements, content caching will be strictly driven by application level policies considering privacy requirements. If for certain kind of IoT data pervasive caching is allowed, intermediate nodes don't need to always forward a content request to its original creator; rather, receiving a cached copy is sufficient for IoT applications. This optimization may greatly reduce the content access latencies.

Furthermore considering hierarchical nature of IoT systems, ICN architectures enable flexible heterogeneous and potentially fault-tolerant approach to storage providing persistence at multiple levels.

Hence in the context of IoT while ICN allows resolution to replicated stored copies, it should also strive for the balance between content security/privacy and regulations considering application requirements.

3.9. Communication Reliability

IoT applications can be broadly categorized into mission critical and non-mission critical. For mission critical applications, reliable communication is one of the most important features as these applications have strong QoS requirements such as low latency and probability of error during information transfer. To summarize, reliable communication desires the following capabilities for the underlying system: (1) seamless mobility support under normal operating conditions, (2) efficient routing in the presence of intermittent disconnection, (3) QoS aware routing, (4) support for redundancy at all levels of a system (device, service, network, storage etc.), and (5) support for rich and diverse communication patterns, both within an IoT domain consisting of multiple IoT nodes and one or more gateway nodes to the Internet and across multiple such domains.

3.10. Self-Organization

The unified IoT architecture should be able to self-organize to meet various application requirements, especially the capability to quickly discover heterogeneous and relevant (local or global) devices/data/services based on the context. This discovery can be achieved through an efficient publish-subscribe service, or through private community grouping/clustering based upon trust and other security requirements. In the former case, the publish-subscribe service must be efficiently implemented, able to support seamless mobility, in- network caching, name-based routing, etc. In the latter case, the IoT architecture needs to discover the private community groups/clusters efficiently.

Another aspect of self-organization is decoupling the sensing Infrastructure from applications. In a unified IoT architecture, various applications run on top of a vast number of IoT devices. Upgrading the firmware of the IoT devices is a difficult work. It is also not practical to reprogram the IoT devices to accommodate every change of the applications. The infrastructure and the application specific logics need to be decoupled. A common interface is required to dynamically configure the interactions between the IoT devices and easily modify the application logics on top of the sensing/actuating infrastructure [30] [31].

3.11. Ad hoc and Infrastructure Mode

Depending upon whether there is communication infrastructure, an IoT system can operate either in ad-hoc or infrastructure mode.

For example, a vehicle may determine to report its location and status information to a server periodically through cellular connection, or, a group of vehicles may form an ad-hoc network that collectively detect road conditions around them. In the cases where infrastructure is unavailable, one of the participating nodes may choose to become the temporary gateway.

The unified IoT architecture needs to design a common protocol that serves both modes. Such a protocol should address the challenges that arise in these two modes: (1) scalability and low latency for the infrastructure mode and (2) efficient neighbor discovery and ad-hoc communication for the ad-hoc mode. Finally we note that hybrid modes are very common in realistic IoT systems.

3.12. IoT Platform Management

An IoT platforms' service, control and data plane will be governed by its own management infrastructure which includes distributed and centralized middleware, discovery, naming, self-configuring, analytic functions, and information dissemination to achieve specific IoT system objectives [25][26][27]. Towards this, new IoT management mechanisms and service metrics need to be developed to measure the success of an IoTdeployment. Considering an IoT systems' defining characteristics such as, its potential large number of IoT devices, objective to save power, mobility, and ad hoc communication, autonomic self-management mechanisms become very critical. Further considering its hierarchical information processing deployment model, the platform needs to orchestrate computational tasks according to the involved sensors and the available computation resources which may change over time. An efficient computation resource discovery and management protocol is required to facilitate this process. The trade-off between information transmission and processing is another challenge.

4. State of the Art

Over the years, many stand-alone IoT systems have been deployed in various domains. These systems usually adopt a vertical silo architecture and support a small set of pre-designated applications. A recent trend, however, is to move away from this approach, towards a unified IoT architecture in which the existing silo IoT systems, as well as new systems that are rapidly deployed. By unified, we mean all the application and network components that use common APIs to interact with each other. This will make their data and services accessible to general Internet applications (as in ETSI- M2M and oneM2M standards). In such a unified architecture, resources can be accessed over Internet and shared across the physical boundaries of the enterprise. However, current approaches to achieve this objective are mostly based upon service overlays over the Internet, whose inherent inefficiencies due to IP protocol [56] hinders the architecture from satisfying the IoT requirements outlined earlier, particularly in terms of scalability, security, mobility, and self-organization, discussed more in Section 4.2.

4.1. Silo IoT Architecture

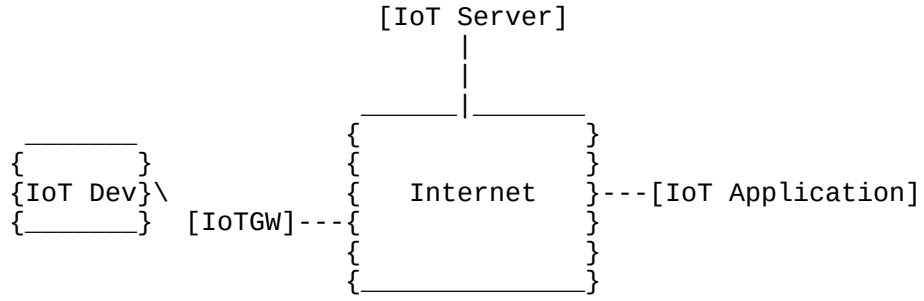

Figure 1:Silo architecture of standalone IoT systems

A typical standalone IoT system is illustrated in Figure 1, which includes devices, a gateway, a server and applications. Many IoT devices have limited power and computing resources, unable to directly run normal IP access network (Ethernet, WIFI, 3G/LTE etc.) protocols. Therefore they use the IoT gateway to the server. Through the IoT server, applications can subscribe to data collected by devices, or interact with devices.

There have been quite a few popular protocols for standalone IoT systems, such as DF-1, MelsecNet, Honeywell SDS, BACnet, etc. However, these protocols are operating at the device-level abstraction, instead of information driven, which may sometimes lead to a fragmented protocol space that requires a higher-level solution for better interoperability.

4.2. Application-Layer Unified IoT Solutions

The current approach to a unified IoT architecture is to make IoT gateways and servers adopt standard APIs. IoT devices connect to the Internet through the standard APIs and IoT applications subscribe and receive data through standard control and data APIs. Building on top of today’s Internet this application-layer unified IoT architecture is the most practical approach towards a unified IoT platform. Towards this, there are ongoing standardization efforts including ETSI[3], oneM2M[4]. Network operators can use frameworks to build common IOT gateways and servers for their customers. In addition, IETF’s CORE working group [5] is developing a set of protocols like CoAP (Constrained Application Protocol) [78], that is a lightweight protocol modeled after HTTP [79] and adapted specifically for the Internet of Things (IoT). CoAP adopts the Representational State Transfer (REST) architecture with Client-Server interactions. It uses UDP as the underlying transport protocol with reliability and multicast support. Both CoAP and HTTP are considered as the suitable

application level protocols for Machine-to-Machine communications, as well as IoT. For example, oneM2M (which is one of leading standards for unified M2M architecture) has both the protocol bindings to HTTP and CoAP for its primitives. Figure 2 shows the architecture adopted in this approach.

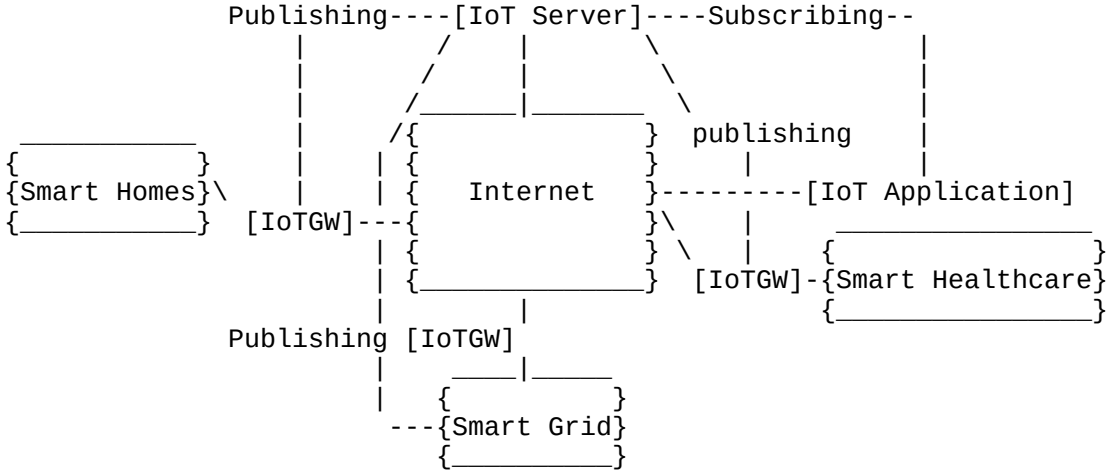

Figure 2: Implementing an open IoT architecture through standardized APIs on the IoT gateways and the server

4.2.1. Weaknesses of the Application-Layer Approach

The above application-layer approach can work with many different protocols, but the system is built upon today's IP network, which has inherent weaknesses towards supporting a unified IoT system. As a result, it cannot satisfy some of the requirements we outlined in Section 3:

- Naming. In current application-layer IoT systems the naming scheme is host centric, i.e., the name of a given resource/service is linked to the device that can provide it. In turn, device names are coupled to IP addresses, which are not persistent in mobile scenarios. On the other side, in IoT systems the same service/ resource could be offered by different devices.

- Security and Trust. In IP, the security and trust model is based on session established between two hosts. Session-based protocols rely on the exchange of several messages before a secure session is established. Use of such protocols in constrained IoT devices

can have serious consequences in terms of energy efficiency because transmission and reception of messages is often more costly than the cryptographic operations. The problem may be amplified with the number of nodes the constrained device has to interact with because of both the computation cost and per session key state required to be managed by the constrained device. Also the trust management schemes are still relatively weak, focusing on securing communication channels rather than managing the data that needs to be secured directly. Though key management in ICN is no less complex than in host based interactions, the benefits is associated with the security credentials in the content instead of the host. Trust is via keys that are bound to names through certificates whose private keys are held by the principals of the system, with IP focusing on the channel model of security while ICN focusing on the object model.

- Mobility. The application-layer approach uses IP addresses as names at the network layer, which hinders the support for device/service mobility or flexible name resolution. Further the Layer 2/3 management, and application-layer addressing and forwarding required to deploy current IoT solutions limit the scalability and management of these systems.

- Resource constraints. The application-layer approach requires every device to send data to an aggregator, gateway or to the IoT server. Resource constraints of the IoT devices, especially in power and bandwidth, could seriously limit the performance of this approach. On the other hand, ICN supports in-network computing/caching/storage, which can alleviate this problem.

- Traffic Characteristics. In this approach, applications are written in a host-centric manner suitable for point-to-point communication. IoT requires multicast support that is challenging the application-layer based IoT systems today, which has only limited deployment in current Internet.

- Contextual Communications. This application-layer based IoT approach may not react to dynamic contextual changes in a timely fashion. The main reason is that context lists are usually kept at the IoT server in this approach, and they cannot help efficiently route requests information at the network layer.

- Storage and Caching. The application-layer approach supports application-centric storage and caching but not what ICN envisions at the network layer, or flexible storage enabled via name-based routing or name-based lookup.

- o Self-Organization. The application-layer approach is topology-based as it is bound to IP semantics, and thus does not sufficiently satisfy the self-organization requirement. In addition to topological self-organization, IoT also requires data- and service-level self-organization [97], which is not supported by this approach.

- o Ad-hoc and infrastructure mode. As mentioned above, the overlay-based approach lacks self-organization, and thus does not provide efficient support for the ad-hoc mode of communication.

4.2.2. Suitability of Delay Tolerant Networking(DTN)

In [21][22], delay-tolerant networking (DTN) has been considered to support future IoT architecture. DTN was created to support information delivery in the presence of network disruptions and disconnections, which has been extended to support heterogeneous networks and name-based routing. The DTN Bundle Protocol is able to achieve some of these same advantages and could be beneficially used in an IoT network to, for example, decouple sender and receiver. The DTN architecture is however centered around named endpoints (endpoint IDs), which usually correspond to a host or a service, and is mainly a way to transport data, while ICN provides a different paradigm centered around named data that addresses additional issues for IoT applications [23] through features such as information naming, information discovery, information request and dissemination. Also, the endpoint IDs could be used to also identify named content, enabling the use of the bundle protocol as a transport mechanism for an information-centric system. Such a use of the bundle protocol as transport would however still require other components from an ICN architecture such as naming conventions, so since the exact transport is not a major focus of the issues in this draft, most of of the discussions are applicable to a generic ICN architecture in general.

5. Advantages of using ICN for IoT

A key concept of ICN is the ability to name data independently from the current location at which it is stored, which simplifies caching and enables decoupling of sender and receiver. Using ICN to design an architecture for IoT data potentially provides many such advantages compared to using traditional host-centric networks and other new architectures. This section highlights general benefits that ICN could provide to IoT networks.

- o Naming of Devices, Data and Services. The heterogeneity of both network equipment deployed and services offered by IoT networks leads to a large variety of data, services and devices. While using a traditional host-centric architecture, only devices or

their network interfaces are named at the network level, leaving to the application layer the task to name data and services. This causes different applications to use different naming schemes, and no consistent mapping from application layer names to network names exist. In many common applications of IoT networks, data and services are the main goal, and ICN provides an intuitive way to name those in a way that can be utilized on the network layer as well. Communication with a specific device is often secondary, but when needed, the same ICN naming mechanisms can be used. The network distributes content and provides a service, instead of only sending data between two named devices. In this context, data content and services can be provided by several devices, or group of devices, hence naming data and services is often more important than naming the devices. This naming mechanism also enables self-configuration of the IoT system.

- Security and privacy. ICN advocates the model of object security to secure data in the network. This concept is based on the idea of securing information objects unlike session-based security mechanisms which secure the communication channel between a pair of nodes. ICN provides data integrity through Name-Data Integrity, i.e., the guarantee that the given data corresponds to the name with which it was addressed. Signature-based schemes can additionally provide data authenticity, meaning establishing the origin, or provenance, of the data, for example, by cryptographically linking a data object to the identity of a publisher. Confidentiality can be handled on a per object basis based on keys established at the application level. All of this means that the actual transmission of data does not have to be secured as the same security mechanisms protect the data after generation until consumed by a client, regardless of whether it is in transit over a communication channel or stored in an intermediate cache. In an ICN network, each individual object within a stream of immutable objects could potentially be retrieved from a cache in a different location. Having a trust relationship with each of these different caches is not realistic. Through Name-Data Integrity, ICN automatically guarantees data integrity to the requester regardless of the location from where it is delivered. The Object Security model also ensures that the content is readily available in a secure state in the device constraints are severe enough that it is not able to perform the required cryptographic operations for Object Security, it may be possible to offload this operation to a trusted gateway to which only a single secure channel needs to be established. ICN can also derive a name from a public key; cryptographic hash of a public key also enables them to be self-certifying, i.e., authenticating the resource object does not require an external authority [25][26].

- Distributed Caching and Processing. While caching mechanisms are already used by other types of overlay networks, IoT networks can potentially benefit even more from caching and in-network processing systems, because of their resource constraints. Wireless bandwidth and power supply can be limited for multiple devices sharing a communication channel, and for small mobile devices powered by batteries. In this case, avoiding unnecessary transmissions with IoT devices to retrieve and distribute IoT data to multiple places is important, hence processing and storing such content in the network can save wireless bandwidth and battery power. Moreover, as for other types of networks, applications for IoT networks requiring shorter delays can benefit from local caches and services to reduce delays between content request and delivery.

- Decoupling between Sender and Receiver. IoT devices may be mobile and face intermittent network connectivity. When specific data is requested, such data can often be delivered by ICN without any consistent direct connectivity between devices. Apart from using structured caching systems as described previously, information can also be spread by forwarding data opportunistically.

6. ICN Design Considerations for IoT

This section outlines some of the ICN specific design considerations and challenges that must be considered when adopting an ICN design for IoT applications and systems, and describes some of the trade offs that will be involved in order to support large scale IoT deployment with diverse application requirements.

Though ICN integrates content/service/host abstraction, name-based routing, compute, caching/storage as part of the network infrastructure, IoT requires special considerations given heterogeneity of devices and interfaces such as for constrained networking [61][119][121], data processing, and content distribution models to meet specific application requirements which we identify as challenges in this section.

6.1. Naming Devices, Data, and Services

The ICN approach of named data and services (i.e., device independent naming) is typically desirable when retrieving IoT data. However, data centric naming may also pose challenges.

- Naming of devices: Naming devices can be useful in an IoT system. For example, actuators require clients to act on a specific node of the deployed network, e.g. to switch it on or off; or it could be necessary to access to a particular device for administrator

purposes. This can be achieved through the specific name that uniquely identify the network entity of interest. Moreover, a persistent name allows a device to change attachment point without loosing its identity. A friendly way to address devices is using contextual hierarchical names, where the same types of names as for data objects can be used. To ensure that the device is always reached, it is important that it is possible to disable caching and request aggregation, if used, for such names.

- Size of data/service name: Content name can have variable length. Since each name has to uniquely identify the content and can also include self-certifying properties (e.g., the hash of the content is bound to the name), its length can reach high values. In particular, according to the specific application, content name size can exceed Data size. This can be the case of IoT sensed values that usually consist in few bytes: data could be as small as a short integer in case of temperature values, or one-byte in case of control messages of an actuator state (on/off). Moreover, a too long name would probably incur in fragmentation at the link layer, and related problems such as, several transmissions, delay and security issues. Viable solutions to handle ICN packets fragmentation and reassembly have been investigated in literature. For instance, the work in [105] proposes to perform the operations hop-by-hop: each hop fragments the packet that has to be forwarded and reassembles the packet received for further processing. This mechanism allows to efficiently handle the recovery of lost or corrupted fragments locally, thus reducing packet delivery failures that require application-level retransmissions.

- Hash-based content name: Hash algorithms are commonly used to name content in order to verify that the content is the one requested. This is only possible in contexts where the requested object is already existing, and where there is a directory service to look up names or learned through a manifest service. This approach is suitable for systems with large data objects where it is important to verify the content.

- Hierarchical names: The use of hierarchical names, such as in the CCN and NDN architectures make it easier to create names a priori and also provides a convenient way to use the same naming scheme for node names. Since the names are not self-certifying, this will require other mechanisms for verification of object integrity. If routing is also done on the hierarchical names, the system will loose some of its location independence and caching will mostly only be done on the path to the publisher.

- Semantic and Metadata based content name: A semantic-based naming approach can allow a successful name retrieving through keywords

(for example, 'noise level at position X'), even if a perfect matching of name is not available [62]. Moreover, enriching contents with metadata allows to better describe them and to establish association between similar ones. However this mechanism require more advanced functionality for matching of such metadata in data objects to the semantics of the name (such as comparing the position information of an object with the position information of the requested name). The need for such potentially computationally heavy tasks in intermediate nodes in the network may be considered understanding the trade-offs in terms of application and network performance.

- Naming of services: Similar to naming of devices or data, services can be referred to with a unique identifier, provided by a specific device or by someone assigned by a central authority as the service provider. It can however also be a service provided by anyone meeting some certain metadata conditions. Example of services include content retrieval, that takes a content name/ description as input and returns the value of that content, and actuation, that takes an actuation command as input and possibly returns a status code afterwards.

- Trust: Names can be used to verify the authenticity and integrity of the data. To provide security functionalities through names, it is possible to use different approaches. On one hand, hierarchical, schematized, Web-of-Trust models allow the public key verification. On the other hand, self-certifying names allow in-network integrity check of the name-key or name-content binding without the need of a Public Key Infrastructure (PKI) or other third party to establish whether the key is trustworthy or not. This can be realized (i) directly: the hash of the content is bound to the name; or (ii) indirectly: first, the hash of the content is signed with the secret key of the publisher, then the public key of the publisher and the signed hash are bound to the name [44]. The hash algorithm can be applied to already existing contents and where there is a directory service or manifests to look up names. In case of contents not yet published, but generated on demand, the hash cannot be known a priori. Thus, different trust mechanisms should be investigated. Moreover, self-certified names approach can hide content semantics, thus making names less human friendly. Since trends show that users prefer to find contents through search engine using keywords, non-human-friendly names could be a barrier unless the content is enriched with keywords. But, this problem does not concern M2M applications. In fact, human-readable names may not be useful in a context of just communicating machines.

- Flexibility: Further challenges arise for hierarchical naming schema: referring to requirements on "constructible names" and "on-demand publishing" [35][36]. TThe former entails that each user is able to construct the name of a desired data item through specific algorithms and that it is possible to retrieve information also using partially specified names. The latter refers the possibility to request a content that has not yet been published in the past, thus triggering its creation.

- Scoping : From an application’s point of view, scopes are used to gather related data. From the network’s perspective, instead, scopes are used to mark where the content is available[65]. For instance, nodes involved in caching coordination can vary according to scope[66]. As a consequence, scoping allows to limit packet request propagation, improving bandwidth and energy resources usage, and control content dissemination thanks to access control rules, different for each scope[64]. However, relying on scoping for security/privacy has been shown to not work all that well for IP, and is unlikely to work well for ICN either. However, scoping may be useful to limit interest propagation, provide a simple means to attain context-sensitive communication, etc. Finally, perimeter- and channel-based access control is often violated in current networks to enable over-the-wire updates and cloud-based services, so scoping is unlikely to replace a need for data-centric security in ICN.

- Confidentiality: As names can reveal information about the nature of the communication or more importantly violate privacy, mechanisms for name confidentiality should be available in the ICN-IoT architecture. To grant confidentiality protection, some approaches have been proposed in order to handle access control in ICN naming scheme such as Attribute-Based Encryption [63] and access control delegation scheme [64]. In the first solution, a Trusted Third Party assigns a set of attributes to each network entity. Then, a publisher (i) encrypts the data with a random key; (II) generates the metadata for the decryption phase; (iii) creates an access policy used to encrypt the random key; (iv) appended the encrypted key to the content name. When the consumer receives the packet, if its attributes satisfy the hidden policy in the name, it can get the random key protected in the name and decrypt the data. The second solution introduces a new trusted network entity (i.e., Access Control Provide). In this case, when a publisher generates a content, it also creates an access control policy and send it to an Access Control Provider. This network entity stores the access control policy, to which it associates a Uniform Resource Identifier (URI). This URI is sent to the publisher and included in the advertisements of the content. Then, when a subscriber tries to access a protected content, it

can authenticate himself and request authorization for the particular policy to the Access Control Provider through the URI.

6.2. Name Resolution

Inter-connecting numerous IoT entities, as well as establishing reachability to them, requires a scalable name resolution system considering several dynamic factors like mobility of end points, service replication, in-network caching, failure or migration [57] [69] [70] [91]. The objective is to achieve scalable name resolution handling static and dynamic ICN entities with low complexity and control overhead. In particular, the main requirements/challenges of a name space (and the corresponding Name Resolution System where necessary) are [50] [52]:

- Scalability: The first challenge faced by ICN-IoT name resolution system is its scalability. Firstly, the approach has to support billions of objects and devices that are connected to the Internet, many of which are crossing administrative domain boundaries. Second of all, in addition to objects/devices, the name resolution system is also responsible for mapping IoT services to their network addresses. Many of these services are based upon contexts, hence dynamically changing, as pointed out in [57]. As a result, the name resolution should be able to scale gracefully to cover a large number of names/services with wide variations (e.g., hierarchical names, flat names, names with limited scope, etc.). Notice that, if hierarchical names are used, scalability can be also supported by leveraging the inherent aggregation capabilities of the hierarchy. Advanced techniques such as hyperbolic routing [86] may offer further scalability and efficiency.

- Deployability and inter-operability: Graceful deployability and interoperability with existing platforms is a must to ensure a naming schema to gain success on the market [7]. As a matter of fact, besides the need to ensure coexistence between IP-centric and ICN-IoT systems, it is required to make different ICN-IoT realms, each one based on a different ICN architecture, to inter-operate.

- Latency: For real-time or delay sensitive M2M application, the name resolution should not affect the overall QoS. With reference to this issue it becomes important to circumvent too centralized resolution schema (whatever the naming style, i.e, hierarchical or flat) by enforcing in-network cooperation among the different entities of the ICN-IoT system, when possible [95]. In addition, fast name lookup are necessary to ensure soft/hard real time services [106][107][108]. This challenge is especially important

for applications with stringent latency requirements, such as health monitoring, emergency handling and smart transportation [109].

- o Locality and network efficiency: During name resolution the named entities closer to the consumer should be easily accessible (subject to the application requirements). This requirement is true in general because, whatever the network, if the edges are able to satisfy the requests of their consumers, the load of the core and content seek time decrease, and the overall system scalability is improved. This facet gains further relevance in those domains where an actuation on the environment has to be executed, based on the feedbacks of the ICN-IoT system, such as in robotics applications, smart grids, and industrial plants [97].

- o Agility: Some data items could disappear while some other ones are created so that the name resolution system should be able to effectively take care of these dynamic conditions. In particular, this challenge applies to very dynamic scenarios (e.g., VANETs) in which data items can be tightly coupled to nodes that can appear and disappear very frequently.

6.3. Security and Privacy

Security and privacy is crucial to all the IoT applications applications including the use cases discussed in Section 2 and subjected to the information context. To exemplify this, in one recent demonstration,it was shown that passive tire pressure sensors in cars could be hacked adversely affecting the automotive system [74], while at the same time the information can be used by a public traffic management system to improve road safety. The ICN paradigm is information-centric as opposed to state-of-the-art host-centric Internet. Besides aspects like naming, content retrieval and caching this also has security implications. ICN advocates the model of trust in content rather than a direct trust in network host mode. This brings in the concept of Object Security which is contrary to session-based security mechanisms such as TLS/DTLS prevalent in the current host-centric Internet. Object Security is based on the idea of securing information objects unlike session-based security mechanisms which secure the communication channel between a pair of nodes for unicast, (or among a set of nodes for multicast/broadcast). This reinforces an inherent characteristic of ICN networks i.e. to decouple senders and receivers. Even session based trust association can be realized in ICN [83], that offers host-independence allowing authentication and authorization to be separated from session encryption, allowing multiple end points to meet specific service objectives. In the context of IoT, the Object Security model has several concrete advantages. Many IoT applications have data and

services are the main goal and specific communication between two devices is secondary. Therefore, it makes more sense to secure IoT objects instead of securing the session between communicating endpoints. Though ICN includes data-centric security features the mechanisms have to be generic enough to satisfy multiplicity of policy requirements for different applications. Furthermore security and privacy concerns have to be dealt in a scenario-specific manner with respect to network function perspective spanning naming, name-resolution, routing, caching, and ICN-APIs. The work by the JOSE WG [80] provides solution approaches to address some of these concerns for object security for constrained devices and should be considered to see what can be applied to an ICN architecture. In general, we feel that security and privacy protection in IoT systems should mainly focus on the following aspects: confidentiality, integrity, authentication and non-repudiation, and availability. Even though, implementing security and privacy methods in IOT systems faces different challenges than in other systems, like IP. Specifically, below we discuss the challenges in the constrained and infrastructure part of the network.

- In the resource-constrained nodes, energy limitation is the biggest challenge. Moreover, it has to deliver its data over a wireless link for a reasonable period of time on a coin cell battery. As a result, traditional security/privacy measures are impractical to be implemented in the constrained part. In this case, one possible solution might be utilizing the physical wireless signals as security measures [75] [55].

- In the infrastructure part, we have several new threats introduced by ICN-IoT [85] particularly in architectures employing name resolution service [119]. Below we list several possible attacks to a name resolution service that is critical to ICN-IoT :

    1. Each IoT device is given an ICN name. The name spoofing attack is a masquerading threat, where a malicious user A claims another user B’s name and attempts to associate it with A’s own network address NA-A, by announcing the mapping (ID-B, NA-A). The consequence of this attack is a denial of service as it can cause traffic directed for B to be directed to A’s network address.

    2. The stale mapping attack is a message manipulation attack involving a malicious name resolution server. In this attack, if a device moves and issues an update, the malicious name resolution server can purposely ignore the update and claim it still has the most recent mapping. Perhaps worse, a name resolution server can selectively choose which (possibly

stale) mapping to give out during queries. The result is a denial of service.

3. The third potential attack, false announcement attack, is an information modification attack that results in illegitimate resource consumption. User A, which is in network NA1, claims its ID-A binds to a different network address, (ID-A, NA2). Thus A can direct its traffic to network NA2, which causes NA2's network resources to be consumed.

4. The collusion attack is an example of an information modification attack in which a malicious user, its network and the location where the mapping is stored collude with each other. The objective behind the malicious collusion is to allow for a fake mapping involving a false network address to pass the verification and become stored in the storage place.

5. An intruder may insert fake/false sensor data into the network. The consequence might be an increase in delay and performance degradation for network services and applications.

- As far as the IoT application server is concerned, data privacy is one of the biggest concerns. IoT data is collected and stored on such servers, which usually run learning algorithms to extract patterns from such data. In this case, it is important to adopt a framework that enables privacy-preserving learning techniques. The framework defines how data is collected, modified (to satisfy the privacy requirement), and transmitted to application developers.

6.4. Caching

In-network caching helps bring data closer to consumers, but its usage differs in constrained and infrastructure part of the IoT network.

Caching in ICN-IoT faces several challenges:

- An important challenge is to determine which nodes on the routing path should cache the data. According to [52], caching the data on a subset of nodes can achieve a better gain than caching on every en-route routers. In particular, the authors propose a "selective caching" scheme to locate those routers with better hit probabilities to cache data. According to [53], selecting a random router to cache data is as good as caching the content everywhere. In [88], the authors suggest that edge caching provides most of the benefits of in-network caching typically discussed in NDN, with simpler deployment. However, it and other

papers consider workloads that are analogous to today's CDNs, not the IoT applications considered here. Further work is likely required to understand the appropriate caching approach for IoT applications.

- Another challenge in ICN-IoT caching is what to cache for IoT applications. In many IoT applications, customers often access a stream of sensor data, and as a result, caching a particular sensor data item for longer time may not be beneficial. In [90], proposed a caching scheme that ensures that older instances of the same sensor stream were first to be evicted from the cache when needed. In [55], the authors suggest to cache IoT services on intermediate routers, and in [57], the authors suggest to cache control information such as pub/sub lists on intermediate nodes. In addition, it is yet unclear what caching means in the context of actuation in an IoT system. For example, it could mean caching the result of a previous actuation request (using other ICN mechanisms to suppress repeated actuation requests within a given time period), or have little meaning at all if actuation uses authenticated requests as in [89].

- Another challenge is that the efficiency of distributed caching may be application dependent. When content popularity is heterogeneous, some content is often requested repeatedly. In that case, the network can benefit from caching. Another case where caching would be beneficial is when devices with low duty cycle are present in the network and when access to the cloud infrastructure is limited. In [90], it is also shown that there are benefits to caching in the network when edge links are lossy, in particular if losses occur close to the content producer, as is common in wireless IoT networks. However, using distributed caching mechanisms in the network is not useful when each object is only requested at most once, as a cache hit can only occur for the second request and later. It may also be less beneficial to have caches distributed throughout ICN nodes in cases when there are overlays of distributed repositories, e.g., a cloud or a Content Distribution Network (CDN), from which all clients can retrieve the data. Using ICN to retrieve data from such services may add some efficiency, but in case of dense occurrence of overlay CDN servers the additional benefit of caching in ICN nodes would be lower. Another example is when the name refers to an object with variable content/state. For example, when the last value for a sensor reading is requested or desired, the returned data should change every time the sensor reading is updated. In that case, ICN caching may increase the risk that cache inconsistencies result in old data being returned.

6.5. Storage

Storage is useful for IoT systems both at longer and small time scales.

Long terms storage can be distributed at vantage points including both the edge and the main IoT service aggregation points such as in the data centers, the difference being in the size of data, processing intelligence and heterogeneity of information that has to be dealt at the two points. The purpose of long terms storage at the edge is to analyze, filter, aggregate and re-publish data for consumption by either by the parent service components or directly by the consumers. The aggregation service points, republish data to be presented as part of the global pub/sub service to interested consuming parties. Long term storage for IoT data also serves the purpose of data backup and replication. Specifically, we face several issues here. Firstly, we need to decide how many replicas we should have for each stream of IoT data, and where we should store these replicas. Given that many IoT applications consume data locally, storage locations should be kept near to data sources as well. Since IoT data are mostly appended to the end of a stream, instead of being updated, managing multiple replicas becomes easier. Secondly, we need to adopt a mechanism that can efficiently route traffic to the nearest data replica. ICN provides several solutions to this problem. For example, global name resolution service (GNRS) can keep track of each replica’s location [56].

Short-term in-network storage (here storage refers to temporary buffer when an outgoing link is not available) helps improve communication reliability, especially when network links are unreliable, such as wireless links. ICN-IoT could adopt a generalized storage-aware routing algorithm to support delay and disruption tolerance in the routing layer. Each router employs in-network storage that facilitates store vs. forward decisions in response to varying link quality and disconnections [111]. These decisions are based on both short-term and long-term path quality metrics. In addition, packets along paths that become disconnected are handled by a disruption tolerant networking (DTN) mode of the protocol with delayed delivery and replication features. In particular, each router maintains two types of topology information: (i) An intra-partition graph is formed by collecting flooded link state advertisements which carry fine-grained, time-sensitive information about the intra-network links; (ii) A DTN graph is maintained via epidemically disseminated link-state advertisements which carry connection probabilities between all nodes in the network. In-network storage faces the following challenges: (1) when to store and how long to store the data, and (2) the next step after the short-term storage. In [90] the authors also shows that it is

beneficial to store data even for shorter periods of time (and even if only a single requester exist) if the network is lossy such that retransmissions and error recovery can be done locally instead of end-to-end.

6.6. Routing and Forwarding

ICN-IoT supports both device-to-device (D2D) communication and device-to-infrastructure (D2I) communication. Some D2D communications are within a single IoT domain, while others might cross IoT domains involving data forwarding within the source IoT domain, in the infrastructure network, and within the destination IoT domain. D2I communications involve data forwarding within the source IoT domain and in the infrastructure network. Data forwarding within an IoT domain can adopt sensor network popular routing protocols such as RPL [81], AODV[82], etc. The main challenge it faces is the resource constraint of the IoT nodes. In order to address this challenge, we could adopt a light-weight, much shorter ICN name for each communicating party within an IoT domain (see Section 6.12 for details). Before we leave the IoT domain, the gateway node will translate the party's short ICN name to its original ICN name. Data forwarding in the ICN infrastructure part can adopt either direct name-based routing or indirect routing using a name resolution service (NRS).

- In direct name-based routing, packets are forwarded by the name of the data [91][61][71] or the name of the destination node [72]. Here, the main challenge is to keep the ICN router state required to route/forward data low. This challenge becomes more serious when a flat naming scheme is used due to the lack of aggregation capabilities.

- In indirect routing, packets are forwarded based upon the locater of the destination node, and the locater is obtained through the name resolution service. In particular, the name-locater binding can be done either before routing (i.e., static binding) or during routing (i.e., dynamic binding). For static binding, the router state is the same as that in traditional routers, and the main challenge is the need to have fast name resolution, especially when the IoT nodes are mobile. For dynamic binding, ICN routers need to main a name-based routing table, hence the challenge of keeping the state information low. At the same time, the need of fast name resolution is also critical.

6.7. Mobility Management

Considering the diversity of IoT applications mobility ranges from tracking sensor data from mobile human beings to large fleets of diverse mobile elements such as drones, vehicles, trucks, trains associated with a transport infrastructure. These mobility could be over heterogeneous access infrastructure ranging from short range 802.15.4 to cellular radios. Further, handling information delivery in ad hoc setting involving vehicles, road side units (RSU) and the corresponding infrastructure based services offers more challenges. ICN architectures has generally been shown to handle consumer and producer mobility [59], and even suitability to V2V scenarios [60]. Networking tools to handle mobility varies with application requirements, which varies from being tolerant to packet losses and latency to those that are mission critical with stringent requirement on both these QoS metrics.

Related to this, the challenge is to quantify the cost associated with mobility management both in the control and forwarding plane, to handle both static binding versus dynamic binding (dynamic binding here refers to enabling seamless mobility) of named resources to its location when either or both consumer and producer is mobile.

During a network transaction, either the data producer or the consumer may move away and thus we need to handle the mobility to avoid information loss. ICN may differentiate mobility of a data consumer from that of a producer:

- When a consumer moves to a new location after sending out the request for Data, the Data may traverse to the previous point of attachment (PoA) but leaving copies of it through its previous path, which can be retrieved by the consumer by retransmitting its request, a technique used by direct routing approach. Indirect routing approach doesn’t differentiate between consumer and producer mobility [91], as it only requires an update to the name resolution system, which can update the routers to rebind the named resource to its new location, while using late-binding to route the packet from the previous PoA to the new one.

- If the data producer itself has moved, the challenge is to control the control overhead while searching for a new data producer (or for the same data producer in its new position) [58]. To this end, flooding techniques could be used rediscover the producer, or the direct routing techniques can be enhanced with late-binding feature to enable seamless mobility [59].

6.8. Contextual Communication

Contextualization through metadata in ICN control or application payload allows IoT applications to adapt to different environments. This enables intelligent networks which are self-configurable and enable intelligent networking among consumers and producers [55]. For example, let us look at the following smart transportation scenario: "James walks on NYC streets and wants to find an empty cab closest to his location." In this example, the context is the relative locations of James and taxi drivers. A context service, as an IoT middleware, processes the contextual information and bridges the gap between raw sensor information and application requirements. Alternatively, naming conventions could be used to allow applications to request content in namespaces related to their local context without requiring a specific service, such as /local/geo/ mgrs/4QFJ/123/678 to retrieve objects published in the 100m grid area 4QFJ 123 678 of the military grid reference system (MGRS). In both cases, trust providers may emerge that can vouch for an application's local knowledge.

However, extracting contextual information on a real-time basis is very challenging:

- We need to have a fast context resolution service through which the involved IoT devices can continuously update its contextual information to the application (e.g., each taxi's location and Jame's information in the above example). Or, in the namespace driven approach, mechanisms for continuous nearest neighbor queries in the namespace need to be developed.

- The difficulty of this challenge grows rapidly when the number of devices involved in a context as well as the number of contexts increases.

6.9. In-network Computing

In-network computing enables ICN routers to host heterogeneous services catering to various network functions and applications needs. Contextual services for IoT networks require in-network computing, in which each sensor node or ICN router implements context reasoning [55]. Another major purpose of in-network computing is to filter and cleanse sensed data in IoT applications, that is critical as the data is noisy as is [73].

Named Function Networking [113] describes an extension of the ICN concept to named functions processed in the network, which could be used to generate data flow processing applications well-suited to, for example, time series data processing in IoT sensing applications.

Related to this, is the need to support efficient function naming. Functions, input parameters, and the output result could be encapsulated in the packet header, the packet body, or mixture of the two (e.g. [31]). If functions are encapsulated in packet headers, the naming scheme affects how a computation task is routed in the network, which IoT devices are involved in the computation task (e.g. [54]), and how a name is decomposed into smaller computation tasks and deployed in the network for a better performance.

Another is challenge is related to support computing-aware routing. Normal routing is for forwarding requests to the nearest source or cache and return the data to the requester, whereas the routing for in-network computation has a different purpose. If the computation task is for aggregating sensed data, the routing strategy is to route the data to achieve a better aggregation performance [51].

In-network computing also includes synchronization challenges. Some computation tasks may need synchronizations between sub-tasks or IoT devices, e.g. a device may not send data as soon as it is available because waiting for data from the neighbours may lead to a better aggregation result; some devices may choose to sleep to save energy while waiting for the results from the neighbours; while aggregating the computation results along the path, the intermediate IoT devices may need to choose the results generated within a certain time window.

6.10. Self-Orgnization

General IoT deployments involves heterogeneous IoT systems consisting of embedded systems, aggregators and service gateways in a IoT domain. To scale IoT deployments to large scale, scope-based self-organization is required. This relates to IoT system middleware functions [118] which include device bootstrapping and discovery, assigning local/global names to device and/or content, security and trust management functions towards device authentication and data privacy. ICN based on-boarding protocols have been studied [96] and has shown to offer significant savings compared to existing approaches. These challenges span both the constrained devices as well as interaction with the aggregators and the service gateways which may have to contact external services like authentication servers to on-board devices. A critical performance optimization metric of these functions while operating at scale is to have low control and data overhead in order to maximize energy efficiency. Further, in the infrastructure part scalable name-based resolution mechanisms, pub/sub services, storage and caching, and in-network computing techniques should be studied to meet the scope-based content dissemination needs of an ICN-IoT system.

6.11. Communications Reliability

ICN offers many ingredients for reliable communication such as multi-home interest anycast over heterogeneous interfaces, caching, and forwarding intelligence for multi-path routing leveraging state-based forwarding in protocols like CCN/NDN. However these features have not been analyzed from the QoS perspective when heterogeneous traffic patterns are mixed in a router, in general QoS for ICN is an open area of research [121]. In-network reliability comes at the cost of a complex network layer; hence the research challenges here is to build redundancy and reliability in the network layer to handle a wide range of disruption scenarios such as congestion, short or long term disconnection, or last mile wireless impairments. Also an ICN network should allow features such as opportunistic store and forward mechanism to be enabled only at certain points in the network, as these mechanisms also entail overheads in the control and forwarding plane overhead which will adversely affect application throughput, Please see the discussion on in-network storage (Section 6.5) for more details .

6.12. Resource Constraints and Heterogeneity

An IoT architecture should take into consideration resource constraints of (often) embedded IoT nodes. Having globally unique IDs is a key feature in ICN, which may consist of tens of bytes. Each device would have a persistent and unique ID no matter when and where it moves. It is also important for ICN-IoT to keep this feature. However, always carrying the long ID in the packet header may not be always feasible over a low-rate layer-2 protocol such as 802.15.4. To solve this issue, ICN can operate using lighter-weight packet header and a much shorter locally unique ID (LUID in short). In this way, we map a device’s long global ID to its short LUID when we reach the local area IoT domain. To cope with collisions that may occur in this mapping process, we let each domain have its own global ID to LUID mapping which is managed by a gateway deployed at the edge of the domain. Different from NAT and other existing domain-based or gateway-based solutions, ICN-IoT does not change the identity the application uses. The applications, either on constrained IoT devices or on the infrastructure nodes, still use the long global IDs to identify each other, while the network performs translation which is transparent to these applications. An IoT node carries its global ID no matter where it moves, even when it is relocated to another local IoT domain and is assigned with a new LUID. This ensures the global reach-ability and mobility handling yet still considers resource constraints of embedded devices.

In addition, the optimizations for other components of the ICN-IoT system (described in earlier subsections) can lead to optimized energy efficiency as well.

7. Differences from T2TRG

T2TRG [9] is a IoT research group under IRTF focusing on research challenges of realizing IoT solutions considering IP as the narrow waist. IP-IoT has been a research topic over a decade and with active industry solutions, hence this group provides an venue to study advanced issues related to IP-IoT security, provisioning, configuration and inter-operability considering various heterogeneous application environments. ICN-IoT is a recent research effort, where the objective to exploit ICN feature of name based routing and security, caching, multicasting, mobility etc in an end-to-end manner to enable IoT services spanning both ad hoc, infrastructure and hybrid scenarios. More detailed comparison of IP-IoT versus ICN-IoT is given in Section 4.

8. Security Considerations

ICN puts security in the forefront of its design which ICN-IoT can leverage to build applications with varying security requirements, which has been discussed quite elaborately in this draft. This is an informational draft and doesn’t create new considerations beyond what has been discussed.

9. Conclusions

This draft offers a comprehensive view of the benefits and design challenges of using ICN to deliver IoT services, not only because of its suitability for constraint networks but also towards ad hoc and infrastructure environments. The draft begins by motivating the need for ICN-IoT by considering popular IoT scenarios and then delves into understanding the IoT requirements from application and networking perspective. We then discuss why current approach of application layer unified IoT solutions based on IP falls short of meeting these requirements, and how ICN architecture is a more suitable towards this. We then elaborate on the design challenges in realizing an ICN-IoT architecture at scale and one that offers reliability, security, energy efficiency, mobility, self-organization among others to accommodate varying IoT service needs.

10. Acknowledgements

We thank all the contributors, reviewers and the valuable comments offered by the chairs to improve this draft.

11. Informative References

[1] Cisco System Inc., CISCO., "Cisco visual networking index: Global mobile data traffic forecast update.", 2016-2021.

[2] Shafig, M., Ji, L., Liu, A., Pang, J., and J. Wang, "A first look at cellular machine-to-machine traffic: large scale measurement and characterization.", Proceedings of the ACM Sigmetrics , 2012.

[3] The European Telecommunications Standards Institute, ETSI., "http://www.etsi.org/.", 1988.

[4] Global Intiative for M2M Standardization, oneM2M., "http://www.onem2m.org/.", 2012.

[5] Constrained RESTful Environments, CoRE., "https://datatracker.ietf.org/wg/core/charter/.", 2013.

[6] Ghodsi, A., Shenker, S., Koponen, T., Singla, A., Raghavan, B., and J. Wilcox, "Information-Centric Networking: Seeing the Forest of the Trees.", Hot Topics in Networking , 2011.

[7] Dong, L., Zhang, Y., and D. Raychaudhuri, "Enhance Content Broadcast Efficiency in Routers with Integrated Caching.", Proceedings of the IEEE Symposium on Computers and Communications (ISCC) , 2011.

[8] NSF FIA project, MobilityFirst., "http://mobilityfirst.winlab.rutgers.edu/", 2010.

[9] Thing-to-Thing Research Group, T2TRG., "https://datatracker.ietf.org/rg/t2trg/about/", 2017.

[10] OPC Foundation, OPC., "https://opcfoundation.org/", 2017.

[11] Kim, B., Lee, S., Lee, Y., Hwang, I., and Y. Rhee, "Mobiiscape: Middleware Support for Scalable Mobility Pattern Monitoring of Moving Objects in a Large-Scale City.", Journal of Systems and Software, Elsevier, 2011.

[12] Dietrich, D., Bruckne, D., Zucker, G., and P. Palensky, "Communication and Computation in Buildings: A Short Introduction and Overview", IEEE Transactions on Industrial Electronics, 2010.

[13] Keith, K., Falco, F., and K. Scarfone, "Guide to Industrial Control Systems (ICS) Security", NIST, Technical Report 800-82 Revision 1, 2013.

[14] Darianian, M. and Martin. Michael, "Smart home mobile RFID-based Internet-of-Things systems and services.", IEEE, ICACTE, 2008.

[15] Zhu, Q., Wang, R., Chen, Q., Chen, Y., and W. Qin, "IOT Gateway: Bridging Wireless Sensor Networks into Internet of Things", IEEE/IFIP, EUC, 2010.

[16] Biswas, T., Chakrabort, A., Ravindran, R., Zhang, X., and G. Wang, "Contextualized information-centric home network", ACM, Sigcomm, 2013.

[17] Huang, R., Zhang, J., Hu, Y., and J. Yang, "Smart Campus: The Developing Trends of Digital Campus", 2012.

[18] Yan, Y., Qian, Y., Hu, Y., and J. Yang, "A Survey on Smart Grid Communication Infrastructures: Motivations, Requirements and Challenges", IEEE Communications Survey and Tutorials, 2013.

[19] Chai, W., Katsaros, K., Strobbe, M., and P. Romano, "Enabling Smart Grid Applications with ICN", ICN Sigcomm, 2015.

[20] Katsaros, K., Chai, W., Wang, N., and G. Pavlou, "Information-centric Networking for Machine-to-Machine Data Delivery: A Case Study in Smart Grid Applications", IEEE Network, 2014.

[21] Mael, A., Maheo, Y., and F. Raimbault, "CoAP over BP for a delay-tolerant internet of things", Future Internet of Things and Cloud (FiCloud), IEEE, 2015.

[22] Patrice, R. and H. Rivano, "Tests Scenario on DTN for IOT III Urbanet collaboration", Dissertation, INRIA, 2015.

[23] Kevin, F., "Comparing Information-Centric and Delay-Tolerant Networking", Local Computer Networks (LCN), 2012 IEEE 37th Conference on. IEEE, 2012..

[24] Miao, Y. and Y. Bu, "Research on the Architecture and Key Technology of Internet of Things (loT) Applied on Smart Grid", IEEE, ICAEE, 2010.

[25] Castro, M. and A. Jara, "An analysis of M2M platforms: challenges and opportunities for the Internet of Things", IMIS, 2012.

[26] Gubbi, J., Buyya, R., and S. Marusic, "Internet of Things (IoT): A vision, architectural elements, and future directions", Future Generation Computer Systems, 2013.

[27] Vandikas, K. and V. Tsiatsis, "Performance Evaluation of an IoT Platform. In Next Generation Mobile Apps, Services and Technologies(NGMAST)", Next Generation Mobile Apps, Services and Technologies (NGMAST), 2014.

[28] Zhang, Y., Yu, R., Nekovee, M., Liu, Y., Xie, S., and S. Gjessing, "Cognitive Machine-to-Machine Communications: Visions and Potentials for the Smart Grid", IEEE, Network, 2012.

[29] Zhou, H., Liu, B., and D. Wang, "Design and Research of Urban Intelligent Transportation System Based on the Internet of Things", Springer Link, 2012.

[30] Alessandrelli, D., Petracca, M., and P. Pagano, "T-Res: enabling reconfigurable in-network processing in IoT-based WSNs.", International Conference on Distributed Computing in Sensor Systems (DCOSS) , 2013.

[31] Kovatsch, M., Mayer, S., and B. Ostermaier, "Moving application logic from the firmware to the Cloud: towards the thin server architecture for the internet of things.", in Proc. 6th Int. Conf. on Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS) , 2012.

[32] Zhang, M., Yu, T., and G. Zhai, "Smart Transport System Based on the Internet of Things", Applied Mechanics and Materials, 2012.

[33] Zhang, A., Yu, R., Nekovee, M., and S. Xie, "The Internet of Things for Ambient Assisted Living", IEEE, ITNG, 2010.

[34] Savola, R., Abie, H., and M. Sihvonen, "Towards metrics-driven adaptive security management in E-health IoT applications.", ACM, BodyNets, 2012.

[35] Jacobson, V., Smetters, D., Plass, M., Stewart, P., Thornton, J., and R. Braynard, "VoCCN: Voice-over Content-Centric Networks", ACM, ReArch, 2009.

[36] Piro, G., Cianci, I., Grieco, L., Boggia, G., and P. Camarda, "Information Centric Services in Smart Cities", ACM, Journal of Systems and Software, 2014.

[37] Gaur, A., Scotney, B., Parr, G., and S. McClean, "Smart City Architecture and its Applications Based on IoT - Smart City Architecture and its Applications Based on IoT", Procedia Computer Science, Volume 52, 2015, Pages 1089-1094.

[38] Herrera-Quintero, L., Banse, K., Vega-Alfonso, J., and A. Venegas-Sanchez, "Smart ITS sensor for the transportation planning using the IoT and Bigdata approaches to produce ITS cloud services", 8th Euro American Conference on Telematics and Information Systems (EATIS), Cartagena, 2016, pp. 1-7.

[39] Melis, A., Pardini, M., Sartori, L., and F. Callegati, "Public Transportation, IoT, Trust and Urban Habits", Internet Science: Third International Conference, INSCI 2016, Florence, Italy, September 12-14, 2016, Proceedings.

[40] Tonneau, A., Mitton, N., and J. Vandaele, "A Survey on (mobile) Wireless Sensor Network Experimentation Testbeds", 2014 IEEE International Conference on Distributed Computing in Sensor Systems, Marina Del Rey, CA, 2014, pp. 263-268.

[41] Zhilin, Y., "Mobile phone location determination and its impact on intelligent transportation systems", IEEE Transactions on Intelligent Transportation Systems, vol. 1, no. 1, pp. 55-64, Mar 2000.

[42] Papadimitratos, P., La Fortelle, A., Evenssen, K., Brignolo, R., and S. Cosenza, "Vehicular communication systems: Enabling technologies, applications, and future outlook on intelligent transportation", IEEE Communications Magazine, vol. 47, no. 11, pp. 84-95, November 2009.

[43] Zhang, Yu., Afanasyev, A., Burke, J., and L. Zhang, "A survey of mobility support in named data networking", Computer Communications Workshops (INFOCOM WKSHPS), 2016 IEEE Conference on. IEEE, 2016.

[44] Xylomenos, G., Ververidis, C., Siris, V., and N. Fotiou et al, "A survey of information-centric networking research", IEEE Communications Surveys and Tutorials, Volume: 16, Issue: 2, Second Quarter 2014 .

[45] Mavromoustakis, C., Mastorakis, G., and J. Batalla, "Internet of Things (IoT) in 5G Mobile Technologies", ISBN,3319309137,Springer.

[46] Firner, S., Medhekar, S., and Y. Zhang, "PIP Tags: Hardware Design and Power Optimization", in Proceedings of HotEmNets, 2008.

[47] Masek, P., Masek, J., Frantik, P., and R. Fujdiak, "A Harmonized Perspective on Transportation Management in Smart Cities: The Novel IoT-Driven Environment for Road Traffic Modeling", Sensors, Volume 16, Issue 11, 2016.

[48] Abreu, D., Velasquez, K., Curado, M., and E. Monteiro, "A resilient Internet of Things architecture for smart cities", Annals of Telecommunications, Volume 72, Issue 1, Pages 19-30, 2017.

[49] Ravindran, R., Biswas, T., Zhang, X., Chakrabort, A., and G. Wang, "Information-centric Networking based Homenet", IEEE/IFIP, 2013.

[50] Dannewitz, C., D' Ambrosio, M., and V. Vercellone, "Hierarchical DHT-based name resolution for information-centric networks", 2013.

[51] Fasoloy, E., Rossiy, M., and M. Zorziy, "In-network Aggregation Techniques for Wireless Sensor Networks: A Survey", IEEE Wireless Communications, 2007.

[52] Chai, W., He, D., and I. Psaras, "Cache "less for more" in information-centric networks", ACM, IFIP, 2012.

[53] Eum, S., Nakauchi, K., Murata, M., Shoji, Yozo., and N. Nishinaga, "Catt: potential based routing with content caching for icn", IEEE Communication Magazine, 2012.

[54] Drira, W. and F. Filali, "Catt: An NDN Query Mechanism for Efficient V2X Data Collection", Eleventh Annual IEEE International Conference on Sensing, Communication, and Networking Workshops (SECON Workshops), 2014.

[55] Eum, S., Shvartzshnaider, Y., Francisco, J., Martini, R., and D. Raychaudhuri, "Enabling internet-of-things services in the mobilityfirst future internet architecture", IEEE, WoWMoM, 2012.

[56] Raychaudhuri, D., Nagaraj, K., and A. Venkatramani, "Mobilityfirst: a robust and trustworthy mobility-centric architecture for the future internet.", ACM SIGMOBILE Mobile Computing and Communications Review 16.3 (2012): 2-13.

[57] Sun, Y., Qiao, X., Cheng, B., and J. Chen, "A low-delay, lightweight publish/subscribe architecture for delay-sensitive IOT services", IEEE, ICWS, 2013.

[58] Azgin, A., Ravindran, R., and GQ. Wang, "Mobility study for Named Data Networking in wireless access networks", IEEE, ICC, 2014.

[59] Azgin, A., Ravindran, R., Chakraborti, A., and GQ. Wang, "Seamless Producer Mobility as a Service in Information Centric Networks", ACM ICN Sigcomm, IC5G Workshop, 2016.

[60] Wang, L., Wakikawa, R., Kuntz, R., and R. Vuyyuru, "Data Naming in Vehicle-to-Vehicle Communications", IEEE, Infocm, Nomen Workshop, 2012.

[61] Baccelli, E., Mehlis, C., Hahm, O., Schmidt, T., and M. Wahlisch, "Information Centric Networking in the IoT:Experiments with NDN in the Wild", ACM, ICN Siggcomm, 2014.

[62] Simona, C. and M. Mongiello, "Pushing the role of information in ICN", Telecommunications (ICT), 2016 23rd International Conference on. IEEE, 2016..

[63] Li, B., Huang, D., Wang, Z., and Y. Zhu, "Attribute-based Access Control for ICN Naming Scheme", IEEE Transactions on Dependable and Secure Computing, vol.PP, no.99, pp.1-1..

[64] Polyzos, G. and N. Fotiou, "Building a reliable Internet of Things using Information-Centric Networking", Journal of Reliable Intelligent Environments, vol.1, no.1, 2015..

[65] Pandurang, K., Xu, W., Trappe, W., and Y. Zhang, "Temporal privacy in wireless sensor networks: Theory and practice", ACM Transactions on Sensor Networks (TOSN) 5, no. 4 (2009): 28..

[66] Trossen, D., Sarela, M., and K. Sollins, "Arguments for an information-centric internetworking architecture.", ACM SIGCOMM Computer Communication Review 40.2 (2010): 26-33.

[67] Zhang, G., Li, Y., and T. Lin, "Caching in information centric networking: A survey.", Computer Networks 57.16 (2013): 3128-3141.

[68] Gronbaek, I., "Architecture for the Internet of Things (IoT): API and interconnect", IEEE, SENSORCOMM, 2008.

[69] Tian, Y., Liu, Y., Yan, Z., Wu, S., and H. Li, "RNS-A Public Resource Name Service Platform for the Internet of Things", IEEE, GreenCom, 2012.

[70] Roussos, G. and P. Chartier, "Scalable id/locator resolution for the iot", IEEE, iThings,CPSCom, 2011.

[71] Amadeo, M. and C. Campolo, "Potential of information-centric wireless sensor and actuator networking", IEEE, ComManTel, 2013.

[72] Nelson, S., Bhanage, G., and D. Raychaudhuri, "GSTAR: generalized storage-aware routing for mobilityfirst in the future mobile internet", ACM, MobiArch, 2011.

[73] Trappe, W., Zhang, Y., and B. Nath, "MIAMI: methods and infrastructure for the assurance of measurement information", ACM, DMSN, 2005.

[74] Rouf, I., Mustafa, H., Taylor, T., Oh, S., Xu, W., Gruteser, M., Trappe, W., and I. Seskar, "Security and privacy vulnerabilities of in-car wireless networks: A tire pressure monitoring system case study", USENIX, 2010.

[75] Liu, R. and W. Trappe, "Securing Wireless Communications at the Physical Layer", Springer, 2010.

[76] Xiao, L., Greenstein, L., Mandayam, N., and W. Trappe, "Using the physical layer for wireless authentication in time-variant channels", IEEE Transactions on Wireless Communications, 2008.

[77] Sun, S., Lannom, L., and B. Boesch, "Handle system overview", IETF, RFC3650, 2003.

[78] Shelby, Z., Hartke, K., and C. Bormann, "The Constrained Application Protocol (CoAP)", RFC 7252, DOI 10.17487/RFC7252, June 2014, <http://www.rfc-editor.org/info/rfc7252>.

[79] Fielding, R., Ed. and J. Reschke, Ed., "Hypertext Transfer Protocol (HTTP/1.1): Message Syntax and Routing", RFC 7230, DOI 10.17487/RFC7230, June 2014, <http://www.rfc-editor.org/info/rfc7230>.

[80] Barnes, R., "Use Cases and Requirements for JSON Object Signing and Encryption (JOSE)", RFC 7165, DOI 10.17487/RFC7165, April 2014, <http://www.rfc-editor.org/info/rfc7165>.

[81] Winter, T., Ed., Thubert, P., Ed., Brandt, A., Hui, J., Kelsey, R., Levis, P., Pister, K., Struik, R., Vasseur, JP., and R. Alexander, "RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks", RFC 6550, DOI 10.17487/RFC6550, March 2012, <http://www.rfc-editor.org/info/rfc6550>.

[82] Perkins, C., Belding-Royer, E., and S. Das, "Ad hoc On-Demand Distance Vector (AODV) Routing", RFC 3561, DOI 10.17487/RFC3561, July 2003, <http://www.rfc-editor.org/info/rfc3561>.

[83] marc.mosko@parc.com, m., Uzun, E., and C. Wood, "CCNx Key Exchange Protocol Version 1.0", draft-wood-icnrg-ccnxkeyexchange-01 (work in progress), October 2016.

[84] Sun, S., "Hypertext Transfer Protocol (HTTP/1.1): Message Syntax and Routing", 2014.

[85] Liu, X., Trappe, W., and Y. Zhang, "Secure Name Resolution for Identifier-to-Locator Mappings in the Global Internet", IEEE, ICCCN, 2013.

[86] Boguna, M., Fragkiskos, P., and K. Dmitri, "Sustaining the internet with hyperbolic mapping", Nature Communications, 2010.

[87] Shang, W., "Securing building management systems using named data networking", IEEE Network 2014.

[88] Fayazbakhsh, S. and et. et al, "Less pain, most of the gain: Incrementally deployable icn", ACM, Siggcomm, 2013.

[89] Burke, J. and et. et al, "Securing instrumented environments over Content-Centric Networking: the case of lighting control", INFOCOM, Computer Communications Workshop, 2013.

[90] Rao, A., Schelen, O., and A. Lindgren, "Performance Implications for IoT over Information Centric Networks", Performance Implications for IoT over Information Centric Networks, ACM CHANTS 2016.

[91] Li, S., Zhang, Y., Dipankar, R., and R. Ravindran, "A comparative study of MobilityFirst and NDN based ICN-IoT architectures", IEEE, QShine, 2014.

[92] Chen, J., Li, S., Yu, H., Zhang, Y., and R. Ravindran, "Exploiting icn for realizing service-oriented communication in iot", IEEE, Communication Magazine, 2016.

[93] Quevedo, J., Corujo, D., and R. Aguiar, "A Case for ICN usage in IoT environments", Global Communications Conference GLOBECOM, IEEE, Dec 2014, Pages 2770-2775.

[94] Lindgren, A., Ben Abdesslem, F., Ahlgren, B., and O. Schelen, "Design Choices for the IoT in Information-Centric Networks", IEEE Annual Consumer Communications and Networking Conference (CCNC) 2016.

[95] Grieco, L., Alaya, M., and K. Drira, "Architecting Information Centric ETSI-M2M systems", IEEE, Pervasive and Computer Communications Workshop (PERCOM), 2014.

[96] Compagno, A., Conti, M., and R. Dorms, "OnboardICNg: a Secure Protocol for On-boarding IoT Devices in ICN", ICN, Sigcomm, 2016.

[97] Grieco, L., Rizzo, A., Colucci, R., Sicari, S., Piro, G., Di Paola, D., and G. Boggia, "IoT-aided robotics applications: technological implications, target domains and open issues", Elsevier Computer Communications, Volume 54, 1 December, 2014.

[98] InterDigital, WhitePaper., "Standardized M2M Software Development Platform", 2011.

[99] Boswarthick, D., "M2M Communications: A Systems Approach", 2012.

[100] Swetina, J., Lu, G., Jacobs, P., Ennesser, F., and J. Song, "Toward a standardized common M2M service layer platform: Introduction to oneM2M", IEEE Wireless Communications, Volume 21, Number 3, June 2014.

[101] Wang, L., Wang, Z., and R. Yang, "Intelligent Multiagent Control System for Energy and Comfort Management in Smart and Sustainable Buildings", IEEE Transactions on Smart Grid, vol. 3, no. 2, pp. 605-617, June 2012..

[102] Lawrence, T., Boudreau, M., and L. Helsen, "Ten questions concerning integrating smart buildings into the smart grid, Building and Environment", Building and Environment, Volume 108, 1 November 2016, Pages 273-283..

[103] Hassan, A. and D. Kim, "Named data networking-based smart home", ICT Express 2.3 (2016): 130-134..

[104] Burke, J., Horn, A., and A. Marianantoni, "Authenticated lighting control using named data networking", UCLA, NDN Technical Report NDN-0011 (2012)..

[105] Afanasyev, A., "Packet fragmentation in ndn: Why ndn uses hop-by-hop fragmentation.", UCLA, NDN Technical Report NDN-0032 (2015)..

[106] Quan, Wei., Xu, C., Guan, J., Zhang, H., and L. Grieco, "Scalable Name Lookup with Adaptive Prefix Bloom Filter for Named Data Networking", IEEE Communications Letters, 2014.

[107] Wang, Yi., Pan, T., Mi, Z., Dai, H., Guo, X., Zhang, T., Liu, B., and Q. Dong, "NameFilter: Achieving fast name lookup with low memory cost via applying two-stage Bloom filters", INFOCOM, 2013.

[108] So, W., Narayanan, A., Oran, D., and Y. Wang, "Toward fast NDN software forwarding lookup engine based on Hash tables", ACM, ANCS, 2012.

[109] Amadeo, M., Campolo, C., Iera, A., and A. Molinaro, "Named data networking for IoT: An architectural perspective", IEEE, EuCNC, 2014.

[110] Amadeo, M., Campolo, C., Iera, A., and A. Molinaro, "Information centric networking in iot scenarios: The case of a smart home", IEEE ICC, June 2015.

[111] Somani, N., Chanda, A., Nelson, S., and D. Raychaudhuri, "Storage- Aware Routing for Robust and Efficient Services in the Future Mobile Internet", Proceedings of ICC FutureNet V, 2012.

[112] Blefari Melazzi, N., Detti, A., Arumaithurai, M., and K. Ramakrishnan, "Internames: A name-to-name principle for the future internet", QShine, August 2014.

[113] Sifalakis, M., Kohler, B., Christopher, C., and C. Tschudin, "An information centric network for computing the distribution of computations", ACM, ICN Sigcomm, 2014.

[114] Lu, R., Lin, X., Zhu, H., and X. Shen, "SPARK: a new VANET-based smart parking scheme for large parking lots", INFOCOM, 2009.

[115] Wang, H. and W. He, "A reservation-based smart parking system", The First International Workshop on Cyber-Physical Networking Systems, 2011.

[116] Qian, L., "Constructing Smart Campus Based on the Cloud Computing and the Internet of Things", Computer Science 2011.

[117] Project, BonVoyage., "European Unions - Horizon 2020, http://bonvoyage2020.eu/", 2016.

[118] Li, S., Zhang, Y., Raychaudhuri, D., Ravindran, R., Zheng, Q., Wang, GQ., and L. Dong, "IoT Middleware over Information-Centric Network", Global Communications Conference (GLOBECOM) ICN Workshop, 2015.

[119] Li, S., Chen, J., Yu, H., Zhang, Y., Raychaudhuri, D., Ravindran, R., Gao, H., Dong, L., Wang, GQ., and H. Liu, "MF-IoT: A MobilityFirst-Based Internet of Things Architecture with Global Reachability and Communication Diversity", IEEE International Conference on Internet-of-Things Design and Implementation (IoTDI), 2016.

[120] Adhatarao, S., Chen, J., Arumaithurai, M., and X. Fu, "Comparison of naming schema in ICN", IEEE LANMAN, June , 2016.

[121] Campolo, C., Corujo, D., Iera, A., and R. Aguiar, "Information-centric Networking for Internet-of-things: Challenges and Opportunities", IEEE Networks, Jan , 2015.

Authors' Addresses

Prof.Yanyong Zhang
WINLAB, Rutgers University
671, U.S 1
North Brunswick, NJ 08902
USA

Email: yyzhang@winlab.rutgers.edu

Prof. Dipankar Raychadhuri
WINLAB, Rutgers University
671, U.S 1
North Brunswick, NJ 08902
USA

Email: ray@winlab.rutgers.edu

Prof. Luigi Alfredo Grieco
Politecnico di Bari (DEI)
Via Orabona 4
Bari 70125
Italy

Email: alfredo.grieco@poliba.it

Prof. Emmanuel Baccelli
INRIA
Room 148, Takustrasse 9
Berlin 14195
France

Email: Emmanuel.Baccelli@inria.fr

Jeff Burke
UCLA REMAP
102 East Melnitz Hall
Los Angeles, CA 90095
USA

Email: jburke@ucla.edu

Ravishankar Ravindran
Huawei Technologies
2330 Central Expressway
Santa Clara, CA 95050
USA

Email: ravi.ravindran@huawei.com

Guoqiang Wang
Huawei Technologies
2330 Central Expressway
Santa Clara, CA 95050
USA

Email: gq.wang@huawei.com

Anders Lindgren
RISE SICS
Box 1263
Kista SE-164 29
SE

Email: anders.lindgren@ri.se

Bengt Ahlgren
RISE SICS
Box 1263
Kista, CA SE-164 29
SE

Email: bengt.ahlgren@ri.se

Olov Schelen
Lulea University of Technology
Lulea SE-971 87
SE

Email: lov.schelen@ltu.se