
Stream: Internet Research Task Force (IRTF)

RFC: [8793](#)

Category: Informational

Published: June 2020

ISSN: 2070-1721

Authors:

B. Wissingh

C. Wood

A. Afanasyev

L. Zhang

TNO

University of California Irvine

Florida International University

UCLA

D. Oran

C. Tschudin

Network Systems Research & Design

University of Basel

RFC 8793

Information-Centric Networking (ICN): Content-Centric Networking (CCNx) and Named Data Networking (NDN) Terminology

Abstract

Information-Centric Networking (ICN) is a novel paradigm where network communications are accomplished by requesting named content instead of sending packets to destination addresses. Named Data Networking (NDN) and Content-Centric Networking (CCNx) are two prominent ICN architectures. This document provides an overview of the terminology and definitions that have been used in describing concepts in these two implementations of ICN. While there are other ICN architectures, they are not part of the NDN and CCNx concepts and as such are out of scope for this document. This document is a product of the Information-Centric Networking Research Group (ICNRG).

Status of This Memo

This document is not an Internet Standards Track specification; it is published for informational purposes.

This document is a product of the Internet Research Task Force (IRTF). The IRTF publishes the results of Internet-related research and development activities. These results might not be suitable for deployment. This RFC represents the consensus of the Information-Centric Networking Research Group of the Internet Research Task Force (IRTF). Documents approved for publication by the IRSG are not a candidate for any level of Internet Standard; see Section 2 of RFC 7841.

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at <https://www.rfc-editor.org/info/rfc8793>.

Copyright Notice

Copyright (c) 2020 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document.

Table of Contents

1. Introduction
2. A Sketch of the Big Picture of ICN
3. Terms by Category
 - 3.1. Generic Terms
 - 3.2. Terms Related to ICN Nodes
 - 3.3. Terms Related to the Forwarding Plane
 - 3.4. Terms Related to Packet Types
 - 3.5. Terms Related to Name Types
 - 3.6. Terms Related to Name Usage
 - 3.7. Terms Related to Data-Centric Security
4. Semantics and Usage
 - 4.1. Data Transfer
 - 4.2. Data Transport
 - 4.3. Lookup Service
 - 4.4. Database Access
 - 4.5. Remote Procedure Call
 - 4.6. Publish/Subscribe
5. IANA Considerations
6. Security Considerations
7. References
 - 7.1. Normative References

[7.2. Informative References](#)

[Acknowledgments](#)

[Authors' Addresses](#)

1. Introduction

Information-centric networking (ICN) is an architecture to evolve the Internet infrastructure from the existing host-centric design to a data-centric architecture, where accessing data by name becomes the essential network primitive. The goal is to let applications refer to data independently of their location or means of transportation, which enables native multicast delivery, ubiquitous in-network caching, and replication of data objects.

As the work on this topic continues to evolve, many new terms are emerging. The goal of this document is to collect the key terms with a corresponding definition as they are used in the CCNx and NDN projects. Among the important documents for these projects are [\[RFC8569\]](#), [\[RFC8609\]](#), and [\[NDNTLV\]](#). Other ICN projects such as [\[NETINF\]](#), [\[PSIRP\]](#), or [\[MOBILITY-FIRST\]](#) are not covered and may be the subject of other documents.

In this document, to help provide context for the individual defined terms, we first sketch the bigger picture of an ICN network by introducing the basic concepts and identifying the major components of the architecture in [Section 2](#); after which, in [Section 3](#), ICN-related terms are listed by different categories. Readers should be aware that in this organization, some terms may be used in other definitions before they themselves are defined.

While this terminology document describes both confidentiality and integrity-related terms, it should be noted that ICN architectures like NDN and CCNx generally do not provide data confidentiality, which is treated in these architectures as an application-layer concern.

This document represents the consensus of the Information-Centric Networking Research Group (ICNRG). It has been reviewed extensively by the Research Group (RG) members active in the specific areas of work covered by the document. It is not an IETF product and is not intended for standardization in the IETF.

2. A Sketch of the Big Picture of ICN

In networking terms, an ICN is a delivery infrastructure for named data. For other complementing views, see [Section 4](#).

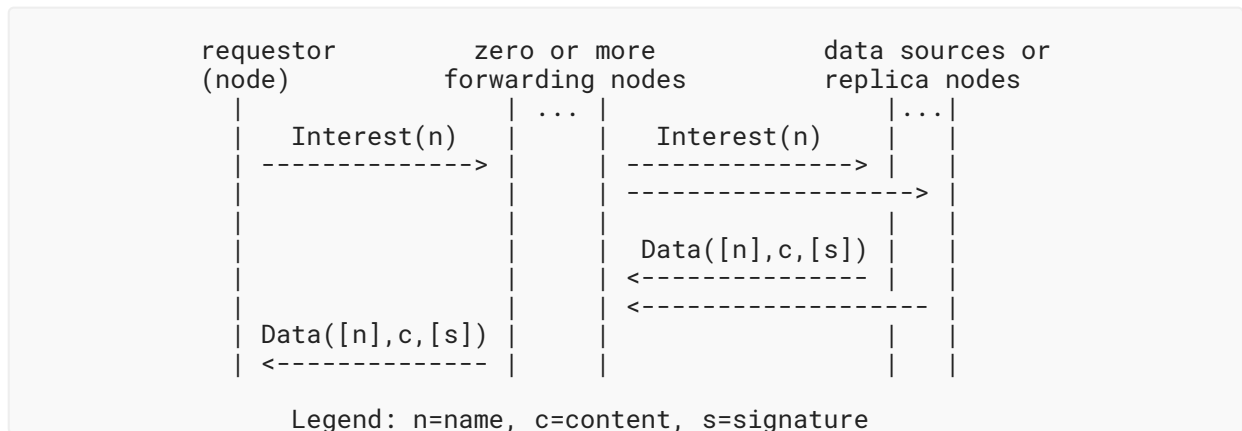


Figure 1: Request-Reply Protocol of ICN Networking.

The following list describes the basic ICN concepts needed to discuss the implementation of this service abstraction.

Request-Reply Protocol (Interest and Data Packet):

An ICN's lookup service is implemented by defining two types of network packet formats: [Interest packets](#) that request content by name and [Data packets](#) that carry the requested content. The returned Data packet must match the request's parameters (e.g., having a partially or fully matching name). If the request is ambiguous and several Data packets would satisfy the request, the ICN network returns only one matching Data packet (thus achieving flow balance between Interest and Data packets over individual Layer 2 interfaces).

Packet and Content Names:

Without a strong cryptographic binding between the name of a [Data packet](#) and its content, Data packet names would be useless for fetching specific content. In ICN, verification of a Data packet's name-to-content binding is achieved through cryptographic means either by (1) a cryptographic signature that explicitly binds an application-chosen name to a Data packet's content or by (2) relying on an implicit name (cryptographic hash of the Data packet with or without application-chosen name) that the data consumer obtained through other means.

Data Authenticity and Encryption:

Any data consumer or network element can (in principle) validate the authenticity of a [Data packet](#) by verifying its cryptographic name-to-content binding. Note that data authenticity is distinct from data trustworthiness, though the two concepts are related. A packet is authentic if it has a valid name-to-content binding, but it may still be unwise to "trust" the content for any particular purpose.

Trust:

Data authenticity is distinct from data trustworthiness, though the two concepts are related. A packet is authentic if it has a valid name-to-content binding. A packet is trustworthy, i.e., it comes from a reputable or trusted origin, if this binding is valid in the context of a trust model. The trust model provides assurance that the name used for a given piece of content is appropriate for the value of the content. [Section 6](#) discusses this further.

Segmenting and Versioning:

An ICN network will be engineered for some packet size limit. As application-level data objects will often be considerably larger, objects must be segmented into multiple [Data packets](#). The names for these Data packets can, for example, be constructed by choosing one application-level object name to which a different suffix is added for each segment. The same method can be used to handle different versions of an application-level object by including a version number in the name of the overall object.

Packet and Frame:

NDN and CCNx introduce Protocol Data Units (PDUs), which typically are larger than the maximum transmission unit of the underlying networking technology. We refer to PDUs as packets and the (potentially fragmented) packet parts that traverse MTU-bound Layer 2 interfaces as frames. Handling Layer 2 technologies that lead to fragmentation of ICN packets is done inside the ICN network and is not visible at the service interface.

ICN Node:

A node within an ICN network can fulfill the role of a data producer, a data consumer, and/or a forwarder for [Interest](#) and [Data packets](#). When a forwarder has connectivity to neighbor nodes, it performs Interest and Data packet forwarding in real time. It can also behave as a store and forward node carrying an Interest or Data packet for some time before forwarding it to the next node. An ICN node may also run routing protocols to assist its Interest forwarding decisions.

Forwarding Plane:

The canonical way of implementing packet forwarding in an ICN network relies on three data structures that capture a node's state: a Forwarding Interest Base (FIB), a Pending Interest Table (PIT), and a [Content Store](#) (CS). It also utilizes Interest forwarding strategies, which take input from both FIB and measurements to make Interest forwarding decisions. When a node receives an [Interest packet](#), it checks its CS and PIT to find a matching entry; if no match is found, the node records the Interest in its PIT and forwards the Interest to the next hop(s) towards the requested content, based on the information in its FIB.

3. Terms by Category

3.1. Generic Terms

Information-Centric Networking (ICN):

A networking architecture that retrieves [Data packets](#) in response to [Interest packets](#). Content-Centric Networking (CCNx 1.x) and Named Data Networking (NDN) are two realizations (designs) of an ICN architecture.

Data Packet Immutability:

After a [Data packet](#) is created, the cryptographic signature binding the name to the content ensures that if either the content or the name changes, that change will be detected and the packet discarded. If the content carried in a Data packet is intended to be mutable, versioning of the name should be used so that each version uniquely identifies an immutable instance of the content. This allows disambiguation of various versions of content such that coordination among the various instances in a distributed system can be achieved.

3.2. Terms Related to ICN Nodes

ICN Interface:

A generalization of the network interface that can represent a physical network interface (ethernet, Wi-Fi, bluetooth adapter, etc.), an overlay inter-node channel (IP/UDP tunnel, etc.), or an intra-node inter-process communication (IPC) channel to an application (unix socket, shared memory, intents, etc.).

Common aliases include: face.

ICN Consumer:

An ICN entity that requests [Data packets](#) by generating and sending out [Interest packets](#) towards local (using intra-node interfaces) or remote (using inter-node interfaces) ICN Forwarders.

Common aliases include: consumer, information consumer, data consumer, consumer of the content.

ICN Producer:

An ICN entity that creates [Data packets](#) and makes them available for retrieval.

Common aliases include: producer, publisher, information publisher, data publisher, data producer.

ICN Forwarder:

An ICN entity that implements stateful forwarding.

Common aliases include: ICN router.

ICN Data Node:

An ICN entity that temporarily stores and potentially carries an Interest or [Data packet](#) before forwarding it to next ICN entity. Note that such ICN data nodes do not have all the properties of data nodes as employed in the Delay Tolerant Networking (DTN) [[RFC4838](#)] specifications.

3.3. Terms Related to the Forwarding Plane

Stateful Forwarding:

A forwarding process that records incoming [Interest packets](#) in the PIT and uses the recorded information to forward the retrieved [Data packets](#) back to the consumer(s). The recorded information can also be used to measure data-plane performance, e.g., to adjust interest forwarding-strategy decisions.

Common aliases include: ICN Data plane, ICN Forwarding.

Forwarding Strategy:

A module of the ICN stateful forwarding (ICN data) plane that implements a decision on where/how to forward the incoming [Interest packet](#). The forwarding strategy can take input from the Forwarding Information Base (FIB), measured data-plane performance parameters, and/or use other mechanisms to make the decision.

Common aliases include: Interest forwarding strategy.

Upstream (forwarding):

Forwarding packets in the direction of Interests (i.e., Interests are forwarded upstream): consumer, router, router, ..., producer.

Downstream (forwarding):

Forwarding packets in the opposite direction of Interest forwarding (i.e., Data and [Interest Nacks](#) are forwarded downstream): producer, router, ..., consumer(s).

Interest Forwarding:

A process of forwarding [Interest packets](#) using the [Names](#) carried in the Interests. In case of stateful forwarding, this also involves creating an entry in the PIT. The forwarding decision is made by the [Forwarding Strategy](#).

Interest Aggregation:

A process of combining multiple [Interest packets](#) with the same [Name](#) and additional restrictions for the same Data into a single PIT entry.

Common aliases include: Interest collapsing.

Data Forwarding:

A process of forwarding the incoming [Data packet](#) to the interface(s) recorded in the corresponding PIT entry (entries) and removing the corresponding PIT entry (entries).

Satisfying an Interest:

An overall process of returning content that satisfies the constraints imposed by the Interest, most notably a match in the provided [Name](#).

Interest Match in FIB (longest prefix match):

A process of finding a FIB entry with the longest [Name](#) (in terms of [Name components](#)) that is a prefix of the specified Name. See [Section 3.5](#) for terms related to name prefixes.

Interest Match in PIT (exact match):

A process of finding a PIT entry that stores the same [Name](#) as specified in the Interest (including Interest restrictions, if any).

Data Match in PIT (all match):

A process of finding (a set of) PIT entries that can be satisfied with the specified [Data packet](#).

Interest Match in CS (any match):

A process of finding an entry in a router's [Content Store](#) that can satisfy the specified Interest.

Pending Interest Table (PIT):

A database that records received and not-yet-satisfied Interests with the interfaces from where they were received. The PIT can also store interfaces to where Interests were forwarded, and information to assess data-plane performance. Interests for the same Data are aggregated into a single PIT entry.

Forwarding Information Base (FIB):

A database that contains a set of prefixes, each prefix associated with one or more faces that can be used to retrieve [Data packets](#) with [Names](#) under the corresponding prefix. The list of faces for each prefix can be ranked, and each face may be associated with additional information to facilitate forwarding-strategy decisions.

Content Store (CS):

A database in an ICN router that provides caching.

In-Network Storage:

An optional process of storing a [Data packet](#) within the network (opportunistic caches, dedicated on/off path caches, and managed in-network storage systems), so it can satisfy an incoming Interest for this Data packet. The in-network storages can optionally advertise the stored Data packets in the routing plane.

Opportunistic Caching:

A process of temporarily storing a forwarded [Data packet](#) in the router's memory (RAM or disk), so it can be used to satisfy future Interests for the same Data, if any.

Common aliases include: on-path in-network caching.

Managed Caching:

The process of achieving the temporary, permanent, or scheduled storage of a selected (set of) [Data packet\(s\)](#).

Common aliases include: off-path in-network storage.

Managed In-Network Storage:

An entity acting as an ICN publisher that implements managed caching.

Common aliases include: repository, repo.

ICN Routing Plane:

An ICN protocol or a set of ICN protocols to exchange information about [Name](#) space reachability.

ICN Routing Information Base (RIB):

A database that contains a set of prefix-face mappings that are produced by running one or multiple routing protocols. The RIB is used to populate the FIB.

3.4. Terms Related to Packet Types

Interest Packet:

A network-level packet that expresses the request for a [Data packet](#) using either an exact name or a name prefix. An Interest packet may optionally carry a set of additional restrictions (e.g., Interest selectors). An Interest may be associated with additional information to facilitate forwarding and can include Interest lifetime, hop limit, forwarding

hints, labels, etc. In different ICN designs, the set of additional associated information may vary.

Common aliases include: Interest, Interest message, information request.

Interest Nack:

A packet that contains the [Interest packet](#) and optional annotation, which is sent by the ICN router to the interface(s) the Interest was received from. An Interest Nack is used to inform downstream ICN nodes about the inability to forward the included Interest packet. The annotation can describe the reason.

Common aliases include: network NACK, Interest return.

Data Packet:

A network-level packet that carries payload, uniquely identified by a name, that is directly secured through cryptographic signature mechanisms.

Common aliases include: data, data object, content object, content object packet, data message, named data object, named data.

Link:

A type of [Data packet](#) whose body contains the [Name](#) of another Data packet. This inner Name is often a [Full Name](#), i.e., it specifies the Packet ID of the corresponding Data packet, but this is not a requirement.

Common aliases include: pointer.

Manifest:

A type of [Data packet](#) that contains [Full Name Links](#) to one or more Data Packets. Manifests group collections of related Data packets under a single Name. Manifests allow both large Data objects to be conveniently split into individual Content Objects under one name, and to represent sets of related Content Objects as a form of "directory". Manifests have the additional benefit of amortizing the signature verification cost for each Data packet referenced by the inner [Links](#). Manifests typically contain additional metadata, e.g., the size (in bytes) of each linked Data packet and the cryptographic hash digest of all Data contained in the linked Data packets.

3.5. Terms Related to Name Types

Name:

A [Data packet](#) identifier. An ICN name is hierarchical (a sequence of name components) and usually is semantically meaningful, making it expressive, flexible, and application-specific

(akin to an HTTP URL). A Name may encode information about application context, semantics, locations (topological, geographical, hyperbolic, etc.), a service name, etc.

Common aliases include: data name, interest name, content name.

Name component:

A sequence of bytes and optionally a numeric type representing a single label in the hierarchical structured name.

Common aliases include: name segment (as in CCNx).

Packet ID:

A unique cryptographic identifier for a [Data packet](#). Typically, this is a cryptographic hash digest of a Data packet (such as SHA256 [RFC6234]), including its name, payload, meta information, and signature.

Common aliases include: implicit digest.

Selector:

A mechanism (condition) to select an individual [Data packet](#) from a collection of Data packets that match a given Interest that requests data using a prefix or exact [Name](#).

Common aliases include: interest selector, restrictor, interest restrictor.

Nonce:

A field of an [Interest packet](#) that transiently names an Interest instance (instance of Interest for a given name). Note: the specifications defining nonces in NDN do not necessarily satisfy all the properties of nonces as discussed in [RFC4949].

Exact Name:

A [Name](#) that is encoded inside a [Data packet](#) and that typically uniquely identifies this Data packet.

Full Name:

An exact [Name](#) with the Packet ID of the corresponding [Data packet](#).

Prefix Name:

A [Name](#) that includes a partial sequence of Name components (starting from the first one) of a Name encoded inside a [Data packet](#).

Common aliases include: prefix.

3.6. Terms Related to Name Usage

Naming conventions:

A convention, agreement, or specification for the [Data packet](#) naming. a Naming convention structures a namespace.

Common aliases include: Naming scheme, ICN naming scheme, namespace convention.

Hierarchically structured naming:

The naming scheme that assigns and interprets a [Name](#) as a sequence of labels (Name components) with hierarchical structure without an assumption of a single administrative root. A structure provides useful context information for the Name.

Common aliases include: hierarchical naming, structured naming.

Flat naming:

The naming scheme that assigns and interprets a [Name](#) as a single label (Name component) without any internal structure. This can be considered a special (or degenerate) case of structured names.

Segmentation:

A process of splitting large application content into a set of uniquely named [Data packets](#). When using hierarchically structured names, each created Data packet has a common prefix and an additional component representing the segment (chunk) number.

Common aliases include: chunking.

Versioning:

A process of assigning a unique [Name](#) to the revision of the content carried in the [Data packet](#). When using a hierarchically structured Name, the version of the Data packet can be carried in a dedicated Name component (e.g., prefix identifies data, unique version component identifies the revision of the data).

Fragmentation:

A process of splitting PDUs into Frames so that they can be transmitted over a Layer 2 interface with a smaller MTU size.

3.7. Terms Related to Data-Centric Security

Data-Centric Security:

A security property associated with the [Data packet](#), including data (Data-Centric) integrity, authenticity, and optionally confidentiality. These security properties stay with the Data packet regardless of where it is stored and how it is retrieved.

Common aliases include: directly securing Data packet.

Data Integrity

A cryptographic mechanism to ensure the consistency of the [Data packet](#) bits. The Data integrity property validates that the Data packet content has not been corrupted during transmission, e.g., over lossy or otherwise unreliable paths, or been subject to deliberate modification.

Data Authenticity

A cryptographic mechanism to ensure trustworthiness of a [Data packet](#) based on a selected (e.g., by a consumer/producer) trust model. Typically, data authenticity is assured through the use of asymmetric cryptographic signatures (e.g., RSA, ECDSA) but can also be realized using symmetric signatures (e.g., Hashed Message Authentication Code (HMAC)) within trusted domains.

Data Confidentiality

A cryptographic mechanism to ensure secrecy of a [Data packet](#). Data confidentiality includes separate mechanisms: [Content confidentiality](#) and [Name confidentiality](#).

Content Confidentiality

A cryptographic mechanism to prevent an unauthorized party to get access to the plain-text payload of a [Data packet](#). Can be realized through encryption (symmetric, asymmetric, hybrid) and proper distribution of the decryption keys to authorized parties.

Name Confidentiality

A cryptographic mechanism to prevent an observer of Interest-Data exchanges (e.g., intermediate router) from gaining detailed meta information about the [Data packet](#). This mechanism can be realized using encryption (same as content confidentiality) or obfuscation mechanisms.

4. Semantics and Usage

The terminology described above is the manifestation of intended semantics of NDN and CCNx operations (What do we expect the network to do?). In this section, we summarize the most commonly proposed use cases and interpretations.

4.1. Data Transfer

The networking view of NDN and CCNx is that the request/reply protocol implements a basic, unreliable data transfer service for single, named packets.

4.2. Data Transport

Data transfer can be turned into a data transport service for application-level objects by additional logic. This transport logic must understand and construct the series of names needed to reassemble the segmented object. Various flavors of transport can be envisaged (reliable, streaming, mailbox, etc.).

4.3. Lookup Service

In a more distributed systems view of the basic request/reply protocol, NDN and CCNx provide a distributed lookup service: given a key value (=name), the service will return the corresponding value.

4.4. Database Access

A lookup service can be turned into a database access protocol by using the namespace structure to specify names as access keys into a database. Therefore, a name prefix stands for a collection or table of a database, while the rest of the name specifies the query expression to be executed.

4.5. Remote Procedure Call

The names as defined in this document for Interests and Data can refer to Remote Procedure call functions, their input arguments, and their results. For a comprehensive view of how to construct RPC or other remote invocation systems, see the Association for Computing Machinery (ACM) ICN paper on [\[RICE\]](#). These capabilities can be further extended into a full distributed computing infrastructure such as that proposed in the ACM ICN paper [\[CFN\]](#).

4.6. Publish/Subscribe

The names as defined in this document for Interests and Data can refer to data collections to be subscribed and individual data objects to be published in a Publish-Subscribe application architecture. For a fully worked example of how to construct such an ICN-based system, see the ACM ICN paper [\[LESSONS-LEARNED\]](#).

5. IANA Considerations

This document has no IANA actions.

6. Security Considerations

While the terms defined in this specification do not in and of themselves present new security considerations, the architectures that utilize the terms most certainly do. Readers should look at those specifications (e.g., [RFC8569] and [NDN]) where various security considerations are addressed in detail.

Some of the terms in this document use the words "trust", "trustworthy", or "trust model". We intend that these have their colloquial meanings; however, much work on trust, and specifically on trust schemas for ICN architectures, has been published in the last few years. For example, it is useful to look at [SCHEMATIZING-TRUST] for more information on the approaches taken to formalize notions of trust for current NDN and CCNx systems.

7. References

7.1. Normative References

- [CFN] Krol, M., Mastorakis, S., Kutscher, D., and D. Oran, "Compute First Networking: Distributed Computing meets ICN", ACM ICN , DOI 10.1145/3357150.3357395, September 2019, <<https://dl.acm.org/citation.cfm?id=3357395>>.
- [LESSONS-LEARNED] Nichols, K., "Lessons Learned Building a Secure Network Measurement Framework using Basic NDN", ACM ICN , DOI 10.1145/3357150.3357397, September 2019, <<https://dl.acm.org/citation.cfm?id=3357397>>.
- [MOBILITY-FIRST] Raychaudhuri, D., Nagaraja, K., and A. Venkataramani, "MobilityFirst: a robust and trustworthy mobility-centric architecture for the future internet", ACM SIGMOBILE , Volume 16, Issue 3 , DOI 10.1145/2412096.2412098, July 2012, <<https://dl.acm.org/citation.cfm?id=2412098>>.
- [NDNTLV] Named Data Networking, "NDN Packet Format Specification", <<https://named-data.net/doc/ndn-tlv/>>.
- [NETINF] Dannewitz, C., Kutscher, D., Ohlman, B., Farrell, S., Ahlgren, B., and K. Holger, "Network of Information (NetInf) - An information-centric networking architecture", Computer Communications , Volume 36, Issue 7 , DOI 10.1016/j.comcom.2013.01.009, April 2013, <<https://dl.acm.org/citation.cfm?id=2459643>>.
- [PSIRP] Trossen, D., Tuononen, J., Xylomenos, G., Sarela, M., Zahemszky, A., Nikander, P., and T. Rinta-aho, "From Design for Tussle to Tussle Networking: PSIRP Vision and Use Cases", May 2008, <http://www.psirp.org/files/Deliverables/PSIRP-TR08-0001_Vision.pdf>.

- [RICE] Krol, M., Habak, K., Kutscher, D., Oran, D., and I. Psaras, "RICE: remote method invocation in ICN", ACM ICN , DOI 10.1145/3267955.3267956, September 2018, <<https://dx.doi.org/10.1145/3267955.3267956>>.
- [SCHEMATIZING-TRUST] Yu, Y., Afanasyev, A., Clark, D., Claffy, K. C., Jacobson, V., and L. Zhang, "Schematizing Trust in Named Data Networking", ACM ICN , DOI 0.1145/2810156.2810170, September 2015, <<https://dx.doi.org/10.1145/2810156.2810170>>.

7.2. Informative References

- [NDN] Named Data Networking, "Named Data Networking: Executive Summary", September 2010, <<https://named-data.net/project/execsummary/>>.
- [RFC4838] Cerf, V., Burleigh, S., Hooke, A., Torgerson, L., Durst, R., Scott, K., Fall, K., and H. Weiss, "Delay-Tolerant Networking Architecture", RFC 4838, DOI 10.17487/RFC4838, April 2007, <<https://www.rfc-editor.org/info/rfc4838>>.
- [RFC4949] Shirey, R., "Internet Security Glossary, Version 2", FYI 36, RFC 4949, DOI 10.17487/RFC4949, August 2007, <<https://www.rfc-editor.org/info/rfc4949>>.
- [RFC6234] Eastlake 3rd, D. and T. Hansen, "US Secure Hash Algorithms (SHA and SHA-based HMAC and HKDF)", RFC 6234, DOI 10.17487/RFC6234, May 2011, <<https://www.rfc-editor.org/info/rfc6234>>.
- [RFC8569] Mosko, M., Solis, I., and C. Wood, "Content-Centric Networking (CCNx) Semantics", RFC 8569, DOI 10.17487/RFC8569, July 2019, <<https://www.rfc-editor.org/info/rfc8569>>.
- [RFC8609] Mosko, M., Solis, I., and C. Wood, "Content-Centric Networking (CCNx) Messages in TLV Format", RFC 8609, DOI 10.17487/RFC8609, July 2019, <<https://www.rfc-editor.org/info/rfc8609>>.

Acknowledgments

Marc Mosko provided much guidance and helpful precision in getting these terms carefully formed and the definitions precise. Marie-Jose Montpetit did a thorough IRSG review, which helped a lot to improve the text. Further comments during the IRSG Poll from Stephen Farrell, Ari Keraenen, Spencer Dawkins, Carsten Bormann, and Brian Trammell further improved the document. Additional helpful comments were received as part of the IESG conflict review from Mirja Kuehlewind and Benjamin Kaduk.

Authors' Addresses

Bastiaan Wissingh

TNO

Email: bastiaan.wissingh@tno.nl

Christopher A. Wood

University of California Irvine

Email: caw@heapingbits.net**Alex Afanasyev**

Florida International University

Email: aa@cs.fiu.edu**Lixia Zhang**

UCLA

Email: lixia@cs.ucla.edu**David Oran**

Network Systems Research & Design

Email: daveoran@orandom.net**Christian Tschudin**

University of Basel

Email: christian.tschudin@unibas.ch