

Internet Engineering Task Force  
Internet Draft  
Intended Status: Informational  
Expires: April 26, 2012

Th. Zahariadis, Ed.  
Synelixix  
Y. Le Louedec  
Orange Labs  
Ch. Timmerer  
UNI-KLU  
S. Spirou  
Intracom  
D. Griffin  
UCL  
October 24, 2011

Catalogue of Advanced Use Cases for  
Content Delivery Network Interconnection

draft-fmn-cdni-advanced-use-cases-00

Abstract

The purpose of this draft is to complement the current CDNi WG use-cases with a catalogue of advanced, longer-term CDN interconnection use cases. The work has been the contribution of six European Commission (EC) co-funded projects, which are part of the EC Future Media networks (FMN) cluster.

Status of this Memo

This Internet-Draft is submitted to IETF in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

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."

The list of current Internet-Drafts can be accessed at  
<http://www.ietf.org/lid-abstracts.html>

The list of Internet-Draft Shadow Directories can be accessed at  
<http://www.ietf.org/shadow.html>

## Copyright and License Notice

Copyright (c) 2011 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 (<http://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. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

This document may contain material from IETF Documents or IETF Contributions published or made publicly available before November 10, 2008. The person(s) controlling the copyright in some of this material may not have granted the IETF Trust the right to allow modifications of such material outside the IETF Standards Process.

Without obtaining an adequate license from the person(s) controlling the copyright in such materials, this document may not be modified outside the IETF Standards Process, and derivative works of it may not be created outside the IETF Standards Process, except to format it for publication as an RFC or to translate it into languages other than English.

## Table of Contents

1	Introduction . . . . .	4
1.1	Terminology . . . . .	4
1.2	Abbreviations . . . . .	4
2	Advanced/Long-term CDNi Use Cases . . . . .	5
2.1	Use Case 1: Caching-CDN interconnection . . . . .	5
2.2	Use Case 2: CDN-CDN interconnections at large scale . . . . .	6
2.3	Use Case 3: Dynamic adaptive streaming over HTTP in multi-CDNs . . . . .	7
2.4	Use Case 4: Dynamic expansion of CDN capacity and geographical reach . . . . .	8
3	Relationship between CDNI and Information-Centric Networking . . . . .	9
4	Acknowledgements . . . . .	11
4.1	List of Contributors . . . . .	12
5	References . . . . .	12
5.1	Normative References . . . . .	12
5.2	Informative reference . . . . .	13
	Authors' Addresses . . . . .	14

## 1 Introduction

The purpose of this draft is to complement the current CDNi Work Group short-term use-cases with a catalogue of advanced, longer-term CDN interconnection use cases. The proposed catalogue of use cases is coming from or inspired by work in the European Commission (EC) Future Media Network (FMN) cluster research projects. Though they are beyond the current short-term objectives of the CDNi WG, the proposed use cases could drive future work in the CDNi WG, especially in case of WG re-chartering (once the present goals will be reached). The use cases are derived from ongoing research work in six EC co-funded projects where concrete design, implementation and evaluation work is being undertaken to validate the approaches.

Moreover, this draft compares CDNs with Information-Centric Networking (ICN) which have similar goals and use cases. We discuss here this relation for the benefit of people and organizations working in these areas.

### 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 RFC2119 [RFC2119].

The present document reuses terminology defined by CDNi WG documents [I-D.ietf-cdni-problem-statement], [I-D.ietf-cdni-use-cases] and [I-D.ietf-cdni-requirements].

### 1.2 Abbreviations

[Ed. Note: List of abbreviations to be updated later]

- o CSP: Content Service Provider
- o dCDN: downstream CDN
- o FMN: Future Media Networks
- o ICN: Information-Centric Networking
- o NSP: Network Service Provider
- o QoE: Quality of Experience
- o SLA: Service Level Agreement
- o uCDN: upstream CDN

- o MPD: Media Presentation Description
- o DASH: Dynamic Adaptive Streaming over HTTP

## 2 Advanced/Long-term CDNi Use Cases

This draft includes four advanced CDNi use cases. The first use case introduces a Caching-CDN interconnection that put emphasis on the in-the network caching mechanism. As CDNi WG mainly targets interconnection between two CDNs, the second use case extends this use case to large-scale CDNs. The third use case introduces support of dynamic adaptive streaming over HTTP (DASH) in a multi-CDNs context, which may be today's most promising video distribution method as it overcomes limitations posed by firewalls and promises efficient distribution utilizing CDNs and network caching resources. Finally, the forth use case proposes the dynamic expansion of CDN capacity and geographical reach.

### 2.1 Use Case 1: Caching-CDN interconnection

Some telecom operators and NSP deploy caching servers, a.k.a. "transparent caching" servers, in their IP networks in order to cache content by CDNs over Internet, with as goal to relax and balance the load on their IP networks.

Today in most situations the two overlay networks - the upstream CDN (uCDN) and the downstream CDN (dCDN) set of transparent caching servers - run independently from each other. There is no specific interconnection interface between the two overlay networks.

There could be some interest to set up interfaces between these overlay networks (of course on condition that their owners get the right business incentives for).

Here are two illustrations of the potential benefits (this is not an exhaustive list):

- Setting up a "logging interface" between the two overlay networks could be beneficial for providing the uCDN (and beyond the Content Service Providers, CSP) with useful logging, monitoring, reporting data.
- Setting up a "CDN metadata interface" between the two overlay networks could be beneficial for allowing the uCDN to request that a given content file be purged from, or invalidated in, any downstream caching server.

The current charter of the CDNi WG states "the WG will not define

support for transparent caching across CDNs". The first priority is indeed to meet the goals and milestones specified in the current charter.

This use case proposal aims at recommending that this "caching-cdn interconnection" use case be considered in case of WG re-chartering (once the present goals will be reached).

The first difference with the present cdn-cdn interconnection model addressed by the CDNi WG is that there is no request routing interface in this "caching-cdn interconnection" use case. Then different sub-cases can be envisioned, depending on which CDNi interfaces are leveraged (with or without logging interface, with or without CDNI metadata interface, etc.). In a first approach, this "caching-cdn interconnection" use case could therefore be considered as a sub-case of the current CDNi WG's cdn-cdn interconnection model.

Note. Once such specific interfaces are set up between the uCDN and the dCDN set of transparent caching servers, the latter ones cannot be any more considered as fully transparent, at least from the viewpoint of the uCDN. This should call for a slight evolution of the terminology.

## 2.2 Use Case 2: CDN-CDN interconnections at large scale

The current focus of the CDNi WG is on the interconnection between two CDNs.

The purpose here would be to investigate situations where (possibly much) more than two CDNs are involved in a CDN federation.

As stated by the Amplification Principle defined in [RFC3439] "there do exist non-linearities which do not occur at small to medium scale, but occur at large scale". As a result, the number of involved CDNs could impact on the requirements and constraints, especially in terms of scalability, and therefore on the conceivable technical options to ensure interconnection.

As an illustration of the amplification principle application in CDNi, the initial considerations, and potential issues, about request routing in CDN interconnect scenarios presented in [I-D.stiemerling-cdni-routing-cons] could be even more critical in large scale CDN federations.

This would possibly lead to the definition of specific sub cases corresponding to cdn-cdn interconnections at large scale. Yet the

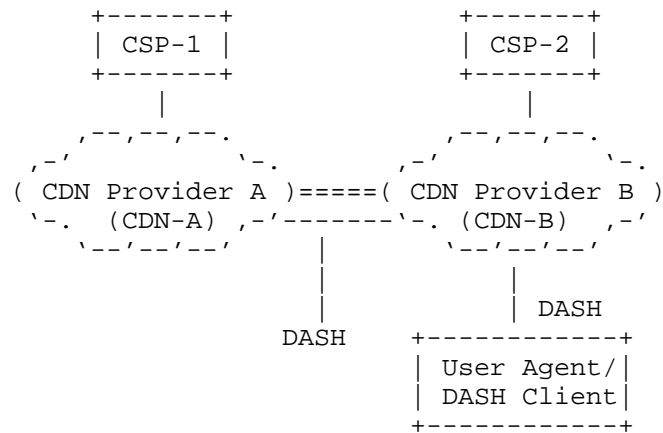
first priority (and prerequisite before investigating such large scale use cases) is to meet the goals and milestones specified in the current charter, i.e. to specify adequate interfaces for interconnecting two CDNs. So this proposal aims at recommending that this "cdn-cdn interconnections at large scale" use case be considered in case of WG re-chartering (once the present goals will be reached).

### 2.3 Use Case 3: Dynamic adaptive streaming over HTTP in multi-CDNs

The ever-growing video traffic on the Internet requires more efficient content distribution mechanisms. Compared to existing RTP/RTSP-based streaming or HTTP progressive download, Dynamic Adaptive Streaming over HTTP (DASH) enables stateless communication between a client and the corresponding server, better content adaptability and support for live media services [Stockhammer2011]. Intrinsic characteristic of DASH is the content fragmentation into various representations comprising segments (or sub-segments) of different encodings of one or several media components. These components/segments are transferred, along with content metadata descriptors referred to as media presentation description (MPD), to the origin servers. Using MPD, clients request the segments utilizing HTTP GET or partial GET requests.

DASH is considered as a promising solution for efficient and high-quality delivery of streaming services over the Internet and is currently standardized as 3GP-DASH, MPEG-DASH, and OIPF HAS. It supports among others, re-use of existing technologies (codecs, containers, etc.), deployment on top of HTTP-CDNs, re-use of HTTP origin and cache/proxy servers, re-use of existing media play-out engines, as well as on-demand, live and time-shifted delivery.

For example, DASH may be exploited within specific CDNs enabling clients to retrieve content hosted by given surrogates, taking into account the scale, the coverage and the reliability of HTTP-based CDN systems. Additionally, DASH can be also utilized as a delivery solution in a CDN Interconnect environment, enabling content acquisition among different providers. In such a case, a content service provider (e.g., CSP-1 in the following figure) will first ingest the prepared content into CDN-A, so that each surrogate can act as a DASH server offering the streaming service to the requesting End-User, acting as DASH client, connected with a different CDN (e.g., CDN-B). Towards this, CDN Interconnection requires interfaces among CDNs to be capable of facilitating their collaboration besides ensuring content streaming efficiently.



=== CDN Interconnection

In other words, in an interconnected CDN environment the definition of a control interface is required, which will enable DASH clients to acquire specific information from a hosting CDN over multiple-CDNs. Towards these, the anticipated interface must be able of providing/delivering:

- The Media Presentation Description that contains metadata to construct appropriate HTTP-URLs to access segments and to provide the streaming service to the user.
- The Number of source surrogates: one or multiple (content segments can be acquired from multiple sources).
- The location of source surrogates: e.g., locality-aware capabilities for choosing the nearest source(s), as a matter of geographical (internal or external) or network-based metrics (e.g., using latency or cost-based metrics).
- Delivery protocols: Definition of the delivery protocols used for the delivery of segments.
- Additional metadata for content delivery (e.g., metadata for content-aware networks).

#### 2.4 Use Case 4: Dynamic expansion of CDN capacity and geographical reach

ISPs may offer a set of specialized network services which may be invoked by their customers, including CDN providers, with appropriate prior agreements and possible payments. The network service of most relevance to this use case is the provision of caching resources located within the ISP. A CDN provider making use of such a service



may invoke new caching resources within a local or remote ISP to dynamically create new CDN surrogate nodes. The newly created resources are provided by the network provider but under the control of the CDN provider.

This is an alternative way of dealing with increased load on a CDN, and a CDN provider who is able to invoke dynamic nodes is therefore able to expand dynamically to accommodate increased traffic demand or extend geographical reach. Such a CDN provider could a) advertise its elasticity to upstream CDNs which could simplify/enhance its resource-routing algorithm decisions, or b) advertise its new capabilities dynamically as it expands/contracts. These announcements could be made part of the CDNi control interface interactions and contributions could be made on the protocol extensions to announce capabilities dynamically and also to propose elasticity metrics.

### 3 Relationship between CDNI and Information-Centric Networking

CDN interconnection has similar goals and use cases to Information-Centric Networking (ICN). We discuss here this relation for the benefit of people and organizations working in these areas. We start with a very brief description of CDNs and ICN in order to have a common understanding. We then discuss similarities and differences in terms of objectives, technical approach, deployment and business models. Finally, we explore the possibility of interaction and coexistence of ICN and CDN in a future Internet. CDNs are real working systems, while ICN is still at the research stage. It is important to note that our analysis is based on the state of ICN research and the assumption that ICN will meet its design goals.

A CDN is a privately owned overlay network that aims to optimize delivery of content from (Content Service Providers) CSPs to End Users. CDNs are independent of each other. Optimization is in terms of performance, availability and cost. CDNs operate by serving content to User Agents from managed caches - called surrogates - at the edge of the network. CDNs may also use reserved network resources. The CDN provider collaborates with NSPs on surrogate placement and network resource reservation. The deployment, extension and (part of) the operation of CDNs are centrally managed. To maintain transparency for End Users, normal content locators (e.g., URLs) are used to access content. However, the CDN treats those locators as simple content identifiers when selecting a surrogate, in effect, decoupling the locator from the content location.

ICN shares many of the goals of CDNs. Optimization of delivery with ICN usually entails caching, QoS-aware routing and traffic differentiation, to various degrees. Unlike CDNs, ICN aims at

operating natively at the NSP level (although operation as an overlay is also possible). An NSP deploys ICN-enabled elements (mainly routers) with minimal planning in terms of content demand. The ICN reach grows with each new ICN-enabled NSP, without any central management. For content access, ICN uses explicit content identifiers that are independent from the content location.

A main objective of both CDNs and ICN is the effective delivery of content from CSPs to End Users. ICN also aims at accommodating End Users as small CSPs, i.e., End Users who want to share content. The decoupling of content identifiers from the content location is an explicit goal in ICN, while in CDN this is a by-product.

The main elements of a CDN are a set of content servers and a request redirection system. The content servers are carefully placed to be close to the User Agents and can be populated prior to any requests based on foreseen demand. An End User requests content with a normal URL, which triggers a part of the redirection system that resides at the seeming location of the content. The redirection system selects a content server based on criteria aiming to optimize some aspect of the delivery task. The selected content server then delivers the requested content to the User Agent.

In ICN, any edge router with storage capabilities can act as a content server, which is populated based on actual demand. Alternatively, or in addition, QoS-aware routing and traffic differentiation techniques are used to establish a path from the content source to the User Agent. Content is requested with a dedicated identifier. This identifier can be used to route the request to the content source or to an intermediate content server, while constructing the path. Alternatively, the identifier can be used as an index in a directory system to retrieve content metadata. The metadata are then used to setup a path from the content source to the User Agent.

A CDN constitutes an administrative domain, whereas in ICN an administrative domain can be as granular as an ICN router. When viewed in terms of administrative domains, CDN interconnection resembles the interconnection of ICN elements/domains. As such, request routing in CDNI and in ICN presents similar technical challenges. Following this thinking, the Content Distribution Metadata and CDNI Metadata of CDNI become related to the content metadata that are used in ICN to establish a path.

A CDN is deployed as an overlay with all its elements independent from NSPs and belonging to the same CDN Provider. The CDN Provider works closely with the CSP in order to ingest and place content. There's also close collaboration between the CDN Provider and the

NSPs for server placement and reservation of resources. Once the CDN is operational, any unforeseen change in CSP requirements, network conditions or End User demand will force the CDN Provider to re-design the deployment. To avoid this, CDNs are designed with over-provisioning and redundancy. On the other hand, ICN is deployed as NSP infrastructure. Each NSP owns and independently manages the ICN elements in its domain. There's no "ICN Provider" to oversee the deployment of ICN. The system grows as each NSP becomes ICN-enabled. In order to use ICN, CSPs need to provide content source servers and content metadata. ICN is designed with flexibility in mind, which permits coping with many unforeseen events.

In a typical business case for CDNs, a CDN is deployed in an NSP domain and the NSP acts both as a CDN Provider and a (sole) CSP. NSPs use this model to offer content services, such as IPTV, to their End Users and so to differentiate their business. In another model for CDNs, a CDN Provider agrees with several NSPs to deploy content servers and reserve resources in their domain. The CDN Provider then sells content delivery services to interested CSPs.

The ICN business model is very similar to that for connectivity services. An NSP deploys ICN in its domain and makes transit or peering agreements with other ICN-enabled NSPs. The NSP then offers content delivery services to CSPs. It is also possible for the NSP to offer content consumption services to its End Users.

In a future Internet where ICN is deployed by some NSPs, it would be difficult to meet end-to-end the ICN performance goals, because of the "holes" between ICN domains. In such a case, CDNs can act as overlay bridges, because they might already have content close to the downstream ICN domain. In essence, from the ICN point of view, CDN Providers would become CSPs. However, the incentive for the CDN Provider is unclear, as such a move could undermine its own content delivery service. Borrowing from the motivation for CDNI, a CDN Provider who wants to extend its reach, instead of interfacing with a neighboring CDN, could interface with an ICN-enabled NSP. This is of course a scenario that only market forces and time can validate.

#### 4 Acknowledgements

This draft has been produced by the Future Media Networks (FMN) cluster of the Networked Media Systems FP7 projects. The work leading to these results has received funding from the European Union's Seventh Framework Programme (FP7/2007-2013) in (alphabetic order): ALICANTE (FP7-ICT-248652; <http://www.ict-alicante.eu/>), COAST (FP7-ICT-248036; <http://www.coast-fp7.eu/>), COMET (FP7-ICT-248784; <http://www.comet-project.org/>), ENVISION (FP7-ICT-248565; <http://www.envision-project.org/>),

NEXTMEDIA (FP7-ICT-249065; <http://www.fi-nextmedia.eu/>) and  
OCEAN (FP7-ICT-248775; <http://www.ict-ocean.eu/>).

#### 4.1 List of Contributors

Andrzej Beben, Warsaw University of Technology, Poland;  
Christian Timmerer, UNI-KLU, Austria;  
David Florez Rodriguez, Telefonica R&D, Spain;  
David Griffin, Yiannis Psaras, Raul Landa, UCL, UK;  
Daniel Negru, Labri, France;  
Evangelos Pallis, Petros Anapliotis, TEIC, Greece;  
George Xilouris, DEMOKRITOS, Greece;  
Isidro Laso, European Commission, Belgium (on a personal basis);  
Klaus Satzke, Alcatel-Lucent Bell Labs, Germany;  
Michalis Georgiadis, PrimeTel, Cyprus;  
Ning Wang, University of Surrey, UK;  
Spiros Spirou, Intracom Telecom, Greece;  
Theodore Zahariadis (editor), Synelixis, Greece;  
Yannick Le Louedec, Orange Labs, France;  
Yiping Chen, Daniel Negru, CNRS-LaBRI, France.

## 5 References

### 5.1 Normative References

[RFC3439] R. Bush, D. Meyer, "Internet Architectural Guidelines,"  
RFC 3439, <http://www.ietf.org/rfc/rfc3439.txt> (updates  
RFC 1958), December 2002.

[I-D.ietf-cdni-problem-statement] Niven-Jenkins, B., Faucheur, F.,

and N. Bitar, "Content Distribution Network Interconnection (CDNI) Problem Statement", draft-ietf-cdni-problem-statement-00 (work progress), September 2011.

[I-D.ietf-cdni-requirements] Leung, K. and Y. Lee, "Content Distribution Network Interconnection (CDNI) Requirements", draft-ietf-cdni-requirements-00 (work in progress), September 2011.

[I-D.ietf-cdni-use-cases] Bertrand, G., Stephan, E., Watson, G., Burbridge, T., Ma, K., "Use Cases for Content Delivery Network Interconnection", draft-ietf-cdni-use-cases-00 (work in progress), September 2011.

## 5.2 Informative reference

[I-D.stiemerling-cdni-routing-cons] Stiemerling, M., "Considerations on Request Routing for CDNI", draft-stiemerling-cdni-routing-cons-00, July 2011

[Stockhammer2011] T. Stockhammer, "Dynamic adaptive streaming over HTTP: standards and design principles", ACM Proceedings of the second annual ACM conference on Multimedia systems, 2011.

[3GP-DASH] ETSI TS 126 247 v10.0.0 (2011-06) Universal Mobile Telecommunications System (UMTS); LTE; Transparent end-to-end Packet-switched Streaming Service (PSS); Progressive Download and Dynamic Adaptive Streaming over HTTP (3GP-DASH) (3GPP TS 26.247 version 10.0.0 Release 10).

Authors' Addresses

Yannick Le Louedec  
Orange Labs, France  
Email: yannick.loulouedec@orange.com

Christian Timmerer  
UNI-KLU, Austria  
Email: christian.timmerer@itec.uni-klu.ac.at

Spiros Spirou  
Intracom, Greece  
Email: spis@intracom.com

David Griffin  
UCL, UK  
Email: dgriffin@ee.ucl.ac.uk

Theodore Zahariadis  
Synelixis, Greece  
Email: zahariad@synelixis.com

Network Working Group  
Internet-Draft  
Intended status: Standards Track  
Expires: April 7, 2013

D. Kutscher  
NEC  
S. Farrell  
E. Davies  
Trinity College Dublin  
October 4, 2012

The NetInf Protocol  
draft-kutscher-icnrg-netinf-protocol-00

## Abstract

This document defines a conceptual protocol and corresponding node requirements for NetInf nodes in a NetInf network. A NetInf network offers an information-centric paradigm that supports the creation, location, exchange and storage of Named Data Objects (NDOs). NetInf nodes can provide different services to other NetInf nodes, e.g., forwarding requests for information objects, delivering corresponding response messages, name resolution services etc. This (abstract) protocol is intended to be run over some "convergence layer" that handles transport issues. Two "wire" formats are defined, one that uses HTTP for message transfer and one layered on UDP.

## 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 April 7, 2013.

## Copyright Notice

Copyright (c) 2012 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

(<http://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 . . . . .	3
2. Principles and Assumptions . . . . .	3
3. Convergence Layer Architecture . . . . .	5
4. The NetInf Protocol - Overview . . . . .	7
5. Protocol Details . . . . .	9
5.1. GET/GET-RESP . . . . .	9
5.2. PUBLISH/PUBLISH-RESP . . . . .	11
5.3. SEARCH/SEARCH-RESP . . . . .	13
6. Convergence Layer Specifications . . . . .	14
6.1. HTTP CL . . . . .	14
6.2. UDP CL . . . . .	17
7. Security Considerations . . . . .	18
8. Acknowledgements . . . . .	19
9. References . . . . .	19
9.1. Normative References . . . . .	19
9.2. Informative References . . . . .	20
Authors' Addresses . . . . .	20



## 1. Introduction

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]

Syntax definitions in this memo are specified according to ABNF [RFC5234].

There is an open-source implementation available that implements (most of) this. See <http://sourceforge.net/projects/netinf/> for code and <http://village.n4c.eu/getputform.html> for access to a test server.

## 2. Principles and Assumptions

A NetInf network provides an information-centric networking (ICN) environment in which units of data content can be identified and accessed using a URI-based naming scheme. NetInf nodes in a NetInf network support the creation, location, exchange and storage of these units of content. In order to support interoperable implementation of the NetInf design, [ref.netinf-db2] the following assumptions are made here:

- o all nodes can take on all NetInf roles (but do not have to);
- o as necessary, nodes may access a Name Resolution System (NRS) and/or a (possibly name based) message routing infrastructure for NetInf messages; and
- o the NetInf protocol can be used directly to access content.

The NetInf protocol operates on Named Data Objects (see [ref.netinf-db2]) referred to as NDOs. An NDO is an ordered collection of octets associated with a name. The NetInf protocol is designed to cache, locate and transmit complete NDOs.

The NetInf protocol is specified so that NDOs can in principle be retrieved from nodes anywhere in the network to which messages can be routed. This routing is intended to be driven by the names of the NDOs, with the option to use an NRS, but this specification does not discuss how routing, nor calling to an NRS, is carried out. Routing will also depend on the underlying Convergence Layer protocol (see Section 3) in use at that node.

Nodes offering NetInf services may return locators in some cases. These locators designate network locations where an NDO might

potentially be available for retrieval, but locators may not be usable outside of some (possibly hard-to-characterise) domain, or for more than a limited period of time, due to mobility of nodes or time limited access through "pinholes" in middleboxes such as firewalls and Network Address Translators (NATs). Accordingly, a design goal is to enable preferential use of names, with locators mostly as hints to improve efficiency. For this reason one can argue that locators ought not be made available to applications using the NetInf protocol in a form that would allow them to try to use the locator outside the NetInf protocol. NDOs may have multiple locators in order to indicate specific interfaces or to reflect attachment to multiple addressing domains. Locators also typically map to a specific instance (copy) of an NDO residing at a given host.

Locators are an example of NDO associated data that may be stored in association with the data content of the NDO. Other types of such data include "metadata" relating to the data content of the NDO, information describing the history of the copy of the NDO in the node where it is stored and search terms that are applicable to the NDO content. The term "affiliated data" will be used to describe the overall set of data, other than the actual octets of the content, stored or transmitted in association with an NDO. This affiliated data could be described as metadata of the NDO but we will reserve that term for a subset of the affiliated data that is usually constructed by the publisher of the NDO describing the content data of the NDO that is sent out in tandem with the data content. The NetInf protocol allows this affiliated data to be transmitted, in whole or in part, in association with NetInf messages.

NDO names will often be based on hash-function output values, and since the preferred hash-function will change over time and may change depending on location, this implies that NDOs can also have more than one name. There may also be cases where truncated hash values are desired (e.g., in cases where packets must be kept below some small size and fitting an entire request into one packet is required), and in such cases collisions will occur, or can easily be generated by bad actors. There are also cases where it is desirable to use a name to refer to some "dynamic" NDO, whose octets change (e.g., perhaps the current weather report) and there are cryptographic methods for doing this. This all means that there is no strict 1:1 mapping between names and NDOs, however, we do expect that for most objects, for most ICN deployments, there will in practice be one NDO that is named by each name. That is, each name usually does refer to just one object, and this protocol is designed to work best for that case.

The following NetInf services are assumed to be implemented on nodes through the NetInf protocol:

- o caching of NDOs, both locally originated and acquired through network operations with the NetInf protocol;
- o requesting the fetching of an NDO using its name, possibly with the addition of a locator, from elsewhere in the network;
- o responding to NetInf protocol NDO fetch operations using a name referring to one of its locally known NDOs, which may have been locally generated or acquired from another NetInf node and cached here, by returning either or both of the data named in the operation or affiliated data including locator(s) referring to a node where that NDO is (assumed to be) available;
- o initiating a search for NDOs matching specified search criteria;
- o responding to search requests received by determining if any locally known NDOs meet the search criteria according to locally determined algorithms;
- o NDO publication via sending out the name and, optionally, either or both of the content data and some affiliated data, such as locators, to other nodes;
- o according to locally determined policy, the ability to accept or reject NDO publication requests that are delivered to the node, and to cache either or both of the objects and/or information about those that are accepted;
- o according to locally determined policy, after carrying out local processing, the ability to forward NetInf messages to other nodes or discard them;
- o managing the data affiliated with the NDO as well as the content data; and
- o local cache management, driven by local policy and (optionally) whatever cache directives are carried in NetInf messages.

### 3. Convergence Layer Architecture

The idea of the Convergence Layer (CL) is to provide a means to transport NetInf messages between pairs of nodes that offer NetInf services. Any protocol that allows NetInf messages to be passed without loss of information can be used as a NetInf Convergence Layer (NetInf-CL) protocol.

This document does not cover the bit-level specification of any CL

protocol. The individual CL protocols will provide their own specification regarding their bit-level format.

Different CLs can be used in the various regions forming a global NetInf network. Where a message has to pass through several intermediate NetInf-capable nodes from source to destination, the NetInf protocol layer at each node is responsible for selecting the appropriate link and CL to forward messages.

Each CL has to offer the following minimal set of capabilities:

- o unidirectional point-to-point transport of NetInf messages from source to destination,
- o preservation of message boundaries,
- o reliable transmission of message octets, and
- o in-order delivery of message octets to the destination node.

If an underlying protocol used by a particular CL cannot offer these capabilities natively, then the CL is responsible for synthesising these capabilities by appropriate means, e.g., use of retransmission or insertion of sequence numbers. However, this does not prevent a CL that uses a more capable underlying protocol from implementing additional capabilities, e.g., bidirectional connections that allow a single connection to send NDOs in both directions.

The CL itself does not specify the properties of the messages, how they are interpreted, and the way nodes should interact with them, as that is what is specified in the present document.

The CL architecture is inspired by, and similar to, the concept used in Delay-Tolerant Networking. [RFC4838][RFC5050].

However, in contrast to DTN-CLs, the NetInf-CL concept does not include the handling of fragments of an NDO "above" the CL. This is the main difference between the CL concept as used in DTNs and ICNs. Put another way, a DTN-CL may result in a bundle being fragmented, and those fragments are only re-assembled at the final bundle destination. In the case of an NetInf-CL, if an NDO is fragmented or chunked within the CL, then those fragments or chunks are reassembled at the next ICN node and that fragmentation or chunking is not visible to the ICN protocol. One can also consider that the DTN Bundle Protocol (BP)[RFC5050], which runs over a DTN-CL, can itself, with an appropriate extension such as the "BPQ" extension, [I-D.farrell-dtnrg-bpq] be an NetInf-CL. That is, a concrete instance of this protocol could use the BP with the BPQ extension as

an NetInf-CL.

#### 4. The NetInf Protocol - Overview

This protocol assumes that NDOs are named using URIs, and in particular via the "ni" URI scheme [I-D.farrell-decade-ni] which **MUST** be supported. There are a set of extensions to the "ni" URI scheme [I-D.hallambaker-decade-ni-params] that **MAY** be supported by nodes. However, other URI forms **MAY** also be used in the NetInf protocol, in particular as locators, and nodes **SHOULD** support at least fetching of "http" URLs.

Nodes are assumed to be capable of discriminating between names and locators, based on the URI scheme or otherwise.

The most common operations for a NetInf node will be fetching (using a GET message) an NDO or responding to such queries. The response to the GET message will, if possible, contain the octets making up the specified NDO and **MAY** contain

- o one or more URIs (typically locators) that could subsequently be used to retrieve the octets of the NDO either via this NetInf protocol or by alternative, locator-specific, means, and/or
- o other affiliated data such as metadata relevant to the NDO.

There are some circumstances in which it **MAY** be appropriate for the response to the GET message to contain only one or more locators and, optionally, other affiliated data. Examples of this situation occur if the responding node is aware that the object content can be returned more effectively using an alternative protocol or from an alternative source because of bandwidth limitations on the links connecting the responding node.

In addition to GET, there is the analagous PUBLISH operation where one node sends URIs and/or NDO octets to another. There is also a SEARCH operation, where one node submits a search query and receives a set of URIs and optional meta-data in response.

GET, PUBLISH and SEARCH messages **MAY** be forwarded by any node that receives them if there is good reason and local policy indicates that this would not result in excessive usage of network resources.

If a request message is forwarded, then a response message **MUST NOT** be sent for that request while the overall "transaction" is still in progress. That is, a node that forwards a request does not answer that request itself until it gets an answer from elsewhere.

Response messages MUST be forwarded by routers to the node from which the corresponding request message was received. The routing mechanisms that are used to ensure responses are correctly forwarded in this way are not specified here.

Since this specification does not determine how message routing, nor use of an NRS is done, we do not otherwise specify how or when messages are to be forwarded.

Nodes that want to make a locally stored NDO available with a specific name can use the PUBLISH message to announce that data to the network. This message MAY "push" the octets of the NDO into other nodes' caches. (If those nodes are willing to take them.) The reasoning behind this is that in many circumstances pushing just a name or a locator will not be helpful because the node with the NDO may be located behind a middlebox that will not allow access to the data from "outside." Pushing the complete NDO to a node that is accessible from the originating node but is also accessible from outside the middlebox "interior," can allow global access, e.g., by caching the NDO on a server in the DMZ ("DeMilitarized Zone") of an enterprise network or in a server provided by a home user's ISP.(Internet Service Provider). The publisher MAY also push affiliated data for the NDO, including additional locators and content metadata that can be stored in a node's NDO cache. The caching node MAY choose to store just the affiliated data without the content data depending on local policy.

As in the case of routing messages generally, this specification does not determine the node(s) to which an NDO can be "pushed."

Finally, NetInf nodes can send a SEARCH message to other NetInf nodes. In response, a NetInf node can perform a local search (i.e., of its local cache) As a response, any of the NetInf nodes that receives the SEARCH message returns a set of "ni" URIs of objects matching the search query. It may also return other types of URI such as "http" URIs. Searching of a node's local cache is the main goal for the SEARCH operation, but if a set of nodes were to forward SEARCH messages, then a global search (e.g., a Google-like service) service could be offered.

NDOs together with any affiliated data are represented using MIME objects. [RFC2045]. Placing as much of the affiliated data linked to the NDO in a multipart MIME object along with the octets of the actual object allows for significant specification and code re-use. For example, we do not need to invent a new typing scheme nor any associated registration rules nor registries.

As an example we might have a MIME object of that is multipart/mixed

and contains image/jpeg and application/json body parts, with the named image in the former and loosely structured associated data in the latter. The "ni" scheme parameters draft discusses such examples. This means that the details of the verification of name-data integrity supported by the ni name scheme also depend on the MIME type(s) used.

MIME also simplifies the specification of schemes that make use of digital signatures, reusing techniques from existing systems including Secure MIME (S/MIME) [RFC5751] and the Cryptographic Message Syntax (CMS) [RFC5652].

Note that (as specified in [I-D.farrell-decade-ni]) two "ni" URIs refer to the same object when the digest algorithm and values are the same, and other fields within the URI (e.g., the authority) are not relevant. Two ni names are identical when they refer to the same object. This means that a comparison function for ni names MUST only compare the digest algorithms and values.

## 5. Protocol Details

We define the GET, PUBLISH and SEARCH messages in line with the above. GET and PUBLISH MUST be supported. SEARCH SHOULD be supported. Each message has an associated response.

This means that GET and PUBLISH MUST be implemented and SEARCH SHOULD be implemented. In terms of services, GET and PUBLISH SHOULD be operational but SEARCH MAY be turned off.

### 5.1. GET/GET-RESP

The GET message is used to request an NDO from the NetInf network. A node responding to the GET message would send a GET-RESP that is linked to the GET request using the msg-id from the GET message as the msg-id for corresponding GET-RESP messages if it has an instance of the requested NDO.

The "ni" form or URI MUST be supported. Other forms of URI MAY be supported.

The msg-id SHOULD be chosen so as to be highly unlikely to collide with any other msg-id and MUST NOT contain information that might be personally identifying, e.g., an IP address or username. A sufficiently long random string SHOULD be used for this.

The ext field is to handle future extensibility (e.g., for message authenticators) and allows for the inclusion of a sequence of type,

length value tuples. No extensions for GET messages are defined at this point in time.

```
get-req = GET msg-id URI [ ext ]  
get-resp = status msg-id [ 1*URI ] [ ext ] [ object ]  
  
ext = json-coded-string
```

Figure 1: GET/GET-RESP Message Format

Any node that receives a GET message and does not have an instance of the NDO referenced in the message **MUST** either

- o forward the message to another node, or
- o generate a GET response message with an appropriate status code and the msg-id from the GET message as the response msg-id.

If the message is forwarded, the node **SHOULD** maintain state that will allow it to generate the GET response message if a matching response message is not received for forwarding within a reasonable period of time after the GET message was forwarded.

If the node has an instance of the NDO, the response **MAY** contain zero or more URIs that **MUST** be either locators for the specified object or else alternative names for that object. If the receiving node has a copy of the relevant object in its cache it **SHOULD** include the object in the response. Possible reasons for not including the object would include situations where the GET message was received via a low-bandwidth interface but where the node "knows" that returning a locator will allow the requestor faster access to the object octets. Alternatively, the node may only be maintaining the affiliated data for the NDO and not the content data if it has not yet received the content data or has discarded it due to cache size limitations.

The object **MUST** be encoded as a MIME object. If there is affiliated data linked to the object this **MUST** also be encoded using MIME and integrated with the object in a multipart/mixed MIME object.

If the receiving node does not have a cached copy of the object it **MAY** choose to forward the message depending on local policy. Such forwarding could be based on name-based routing, on an NRS lookup or other mechanisms (e.g. a node might have a default route).

If an get-resp is received with an object that is not MIME encoded or of an unknown MIME type then that **MUST** be treated as an application/



octet-stream for the purposes of name-data integrity verification.

get-resp messages MAY include extensions as with all others.

## 5.2. PUBLISH/PUBLISH-RESP

The PUBLISH message allows a node to push the name, and optionally, alternative names, locators, a copy of the object octets and/or object meta-data. Ignoring extensions, only a status code is expected in return.

A msg-id MUST be included as in a GET message.

A URI containing a name MUST be included. The "ni" URI scheme SHOULD be used for this name.

The message MAY also contain additional URIs that represent either alternative names or locators where the identical object can be found and metadata relating to the published content. As mentioned in Section 4 it is the responsibility of the receiving node to discriminate between those URIs used as names and those used as locators.

The object octets MAY be included. This is intended to handle the case where the publishing node is not able to receive GET messages for objects. An implementation SHOULD test (or "know") its local network context sufficiently well to decide if the object octets ought to be included or not. Methods for checking this are out of scope of this specification.

A node receiving a PUBLISH message chooses what information from the message, if any, to cache according to local policy and availability of resources. It is RECOMMENDED that a node that receives a PUBLISH message containing the object octets verify that the digest in the name under which the content is published matches with the digest of the received data.

One way to "fill a cache" if the object octets are not included in the PUBLISH would be for the recipient of the PUBLISH to simply request the object octets using GET and cache those. (There is no point in sending a PUBLISH without the octets and without any locator.) This behaviour is, of course, an implementation issue.

In some cases it may make sense for a (contactable) node to only publish the name and metadata about the object. The idea here is that the metadata could help with routing or name resolution or search. Since we are representing both NDO octets and affiliated data such as the metadata as MIME objects, we need to tell the

receiver of the PUBLISH message whether or not that message contains the full object. We do this via the "full-ndo-flag" which, if present, indicates that the PUBLISH message contains enough data so the receiver of the PUBLISH message has sufficient data to provide a complete answer a subsequent GET message for that name, i.e., data content and affiliated data.

If a node receives a PUBLISH message for an NDO which already exists in its cache, the received information SHOULD be used to complete or update the node's cached information for the NDO:

- o If the object octets are included and the node currently does not have the octets cached, the data content MAY be added to the cache. Again it is RECOMMENDED that the received data has the correct digest as specified in the NDO name, and
- o Items in the affiliated data MAY be merged into cached affiliated data, including adding additional locators to the list of known locators for the NDO and merging any content metadata with previously received metadata. If there is a conflict, the choice of metadata to be stored is a matter of policy.

It is RECOMMENDED that a timestamp be recorded whenever the cached information for an NDO is updated and that this timestamp be stored in the affiliated data and the most recent timestamp returned with any subsequent GET or SEARCH request that references the NDO.

Extensions ("ext") MAY be included as in a GET request. One such HTTP CL-specific extension ("meta") is defined in Section 6.1 below.

```
pub-req = PUBLISH msg-id 1*URI [ ext ] [ [ full-ndo-flag ] object ]
pub-resp = status msg-id [ ext ]
```

Figure 2: PUBLISH/PUBLISH-RESP Message Format

The response to a PUBLISH message is a status code and the msg-id from the PUBLISH message and optional extensions.

A node receiving a PUBLISH message MAY choose to forward the message to other nodes whether or not it chooses to cache any information. If this node does not cache the information but does forward the PUBLISH message, it should postpone sending a response message until a reasonable period of time has elapsed during which no other responses to the PUBLISH message are received for forwarding. However, the node MAY send an extra response message, even if it forwards the PUBLISH message, if the sender of the PUBLISH message

would have expected the receiving node to cache the object (e.g., because of a contractual relationship) but it was unable to do so for some reason.

### 5.3. SEARCH/SEARCH-RESP

The SEARCH message allows the requestor to send a set of query tokens containing search keywords. The response is either a status code or a multipart MIME object containing a set of metadata body parts, each of which MUST include a name for an NDO that is considered to match the query keywords.

```
search-req = SEARCH msg-id [ 1*token ] [ ext ]  
search-resp = status msg-id [ results ] [ ext ]
```

Figure 3: SEARCH/SEARCH-RESP Message Format

In the case where the response contains results, these MUST take the form of an application/json MIME object containing an array of results. Each result MUST have a "name" field with a URI as the value of that field. Any other fields in array elements SHOULD contain metadata that is intended to allow the requestor to select which, if any, of the names offered to retrieve.

The URIs included in a search-resp SHOULD be names, but MAY be locators, to be distinguished by the requestor as in the case of GET responses.

The intent of the SEARCH message is to allow nodes to search one another's caches, but without requiring us to fix the details (ontology) for NDO content metadata. While this main intended use-case does not involve forwarding of SEARCH messages that is not precluded.

As with PUBLISH messages, if a SEARCH message is forwarded, the forwarding node postpones sending an empty SEARCH response until a reasonable time is elapsed to see if alternative node responds to the SEARCH.

If a SEARCH at a node identifies an NDO that is included in the results of a search, the tokens that were used for the search MAY be recorded in the affiliated data cached with the NDO. Each set of search tokens for which a "match" is obtained should be recorded separately resulting in an array of set of tokens. If the search mechanisms used provides a reliability measure, this MAY also be recorded and the measure may be used to limit the size of the search

tokens array by discarding (or never inserting) sets of tokens with low reliability scores.

SEARCH messages MAY include extensions as for other messages.

## 6. Convergence Layer Specifications

This section specifies two convergence layers that represent instantiations of the NetInf protocol. The first, based on HTTP, is intended for using NetInf in existing web infrastructures, whereas the second, based on UDP, provides an efficient datagram-based hop-by-hop message transport that can be used to query for GET requests sent to an NRS node or for multicasting such requests in a local network.

### 6.1. HTTP CL

The basic idea with the HTTP CL is to use forms for NetInf protocol requests and Multipart MIME HTTP response bodies for NetInf protocol responses. This has been done to allow web browsers to be able to easily interact with NetInf and because there are many tools available that make implementation relatively easy. Note though that the NetInf HTTP CL is also intended for use between NetInf infrastructure nodes.

The HTTP CL assumes that the client knows the address of the HTTP server to which it will send requests. Clients MAY use the authority part of an ni URI, if one is present to select the HTTP responder. NetInf HTTP responders MUST accept requests sent to the following paths:

/netinfproto/get for NetInf GET requests

/netinfproto/publish for NetInf PUB requests

/netinfproto/search for NetInf SEARCH requests

So for example a client would send an HTTP request containing a NetInf GET to `http://example.com/netinfproto/get`

NetInf HTTP responders SHOULD also make ni URIs available at the relevant well-known URL [RFC5785] for the ni URI.  
[I-D.farrell-decade-ni]

NetInf protocol requests use forms. The mapping of the fields from the abstract protocol is as shown in Figure 4. [[NOTE: this is a bit inconsistent now, and just reflects SF's code.]]

Abstract Protocol Field	Form field	Comments (field type in form)
URI	urival, URI loc1,loc2	usually an ni URI (text) or locator
msg-id	msgid	a message identifier (text)
ext	ext	extension(s) (JSON encoded string)
full-ndo-flag	fullPut	true if object supplied (checkbox)
object	octets	object octets (file specification)
n/a	rform	response format required, can be "html" or "json" (radio)
token	tokens	one text field with all search keywords (text)

Figure 4: Form fields used in NetInf requests

## Notes for Figure 4:

For GET messages: "URI" and "msgid" are mandatory.  
"loc1" and "loc2" are optional.  
"ext" may be used in future but no values currently defined.

For PUBLISH messages: "URI" and "msgid" are mandatory.  
"loc1", "loc2", "ext", "rform" and "fullPut" are optional.  
If "rform" is absent, the "json" value is assumed.  
If "fullPut" is absent, a "false" value is assumed.  
If "fullPut" is present and set to "true", "octets" must be present.  
If present, "octets" contains a file specification and the object octets.  
If present, "ext" may contain a "meta" item. The value of "ext" MUST be a JSON object string and the value of the "meta" item MUST be a (subsidiary) object, e.g., the "ext"

```
string might be
{ "meta": { "mi1": 5, "mi2": { ...},
"mi3": "abcd". "mi4": [...] }}}
```

For SEARCH messages: "msgid" and "tokens" are mandatory.  
"rform" is optional.  
If "rform" is absent, the "json" value is assumed.  
"ext" may be used in future but no values currently defined.

HTTP responses for each command can differ.

For GET, the a successful HTTP response (HTTP response code 2xx) will contain either an application/json (if no object is returned) or else a multipart/mixed with two (and exactly two:-) body parts, the first being an application/json and the second containing the object octets, with whatever MIME type is appropriate.

The application/json component will consist of a JSON object that SHOULD contain the following named fields:

NetInf	A string describing the version of the NetInf protocol in use (e.g., "V0.1a").
ni	The "canonicalized" form of the NDO as a URI in the ni scheme: "canonicalized" means that the URI has empty netloc and query string fields. For example: "ni:///sha-256-64;gf2yhPY9Mu0" or "nih:/sha256-32;8lfdb284;d".
msgid	The value of the msgid field in the GET message that resulted in this response.
ts	The timestamp of the last update of the cached information in the cache from which the NDO is being sent.
status	A code, taken from the HTTP 2xx response codes indicating what has been returned (200 if both affiliated data and content has been returned and 203 if only affiliated data is returned).
ct	The MIME content type of the NDO content data, if known. Empty string if not yet known.

- loclist    Array of locator names (strings) from where the NDO might potentially be retrieved.
- metadata    A JSON object containing any named items copied in from "meta" object(s) supplied by any PUBLISH messages received at the node that sent the response plus an entry named "publish" which contains a string indicating the class of node and software that generated the cache entry.
- searches    A JSON array of objects each containing a set of strings representing search tokens and information about the search mechanism that resulted in a match with the NDO during a previous search.

For PUBLISH, the HTTP response will contain an application/json or text/html response, depending on the value of the rform form field. (If rform is missing json is the default.) The application/json structure is as for a GET response. The text/html document will provide a report of the successful publication of the NDO and whatever other relevant information from the affiliated information seems appropriate for inspection by a human user.

For SEARCH, the HTTP response will contain an application/json or text/html response, depending on the value of the rform form field. (If rform is missing json is the default.) The application/json structure is similar to the previous structures, but has a "results" object that contains an array of object details.

## 6.2. UDP CL

The UDP CL implements the NetInf protocol with a UDP datagram services, i.e., all NetInf messages are mapped to individual UDP messages. The purpose is to provide a light-weight datagram-based CL that can be used to implement NetInf transport protocols on top and that can provide efficient communication for querying NRSSs, and request broadcasting/multicasting. The UDP CL provides no hop-by-hop flow control, retransmission and fragmentation/re-assembly.

The UDP CL has two sending modes: 1) send to specified destination IP address and 2) send to the well-known IPv4 multicast address 225.4.5.6. For both unicast and multicast the UDP port number is 2345. All request and response messages are JSON objects, i.e., unordered sets of name/value pairs.

For UDP CL messages, the following JSON names for name/value pairs are defined (not all objects have to be present in all messages):

```
version  # the NetInf UDP CL protocol version -- currently
          # "NetInfUDP/1.0"

msgType  # the message type (e.g., GET)

uri      # the NI URI

msgId    # the message ID (must be unique per CL hop and
          # request/response pair)

locators # an array of locators

instance # an UDP CL speaker identifier (must be unique per IP host,
          # e.g., process ID and per process ID
```

Figure 5: UDP CL JSON request structure

This version of the specification defines the GET request and the corresponding GET response only.

GET request A GET request provides the following objects:

```
version:  "NetInfUDP/1.0"

msgType:  "GET"

uri:      name of the requested NDO

msgId:    message ID (see above)
```

GET reponse A GET response provides the following objects:

```
version:  "NetInfUDP/1.0"

msgType:  "GET-RESP"

uri:      name of the requested NDO

msgId:    message ID (see above)

locators: a list of locator strings
```

## 7. Security Considerations

For privacy preserving reasons requestors SHOULD attempt to limit the personally identifying information (PII) included with search requests. Including fine-grained search keywords can expose



requestor PII. For this reason, we RECOMMEND that requestors include more coarse grained keywords and that responders include sufficient meta-data to allow the requestor to refine their search based on the meta-data in the response.

Similarly, search responders SHOULD consider whether or not they respond to all or some search requests as exposing one's cached content can also be a form of PII if the cached content is generated at the behest of the responder.

Name-data integrity validation details are TBD for some common MIME types.

Users need to be aware that the affiliated data is NOT protected by the name-data integrity as this applies only to the data content octets.

[[More TBD no doubt.]]

## 8. Acknowledgements

This work has been supported by the EU FP7 project SAIL.

Claudio Imbrenda and Christian Dannewitz contributed to early versions of this document whilst working at NEC and the University of Paderborn respectively.

## 9. References

### 9.1. Normative References

[I-D.farrell-decade-ni]

Farrell, S., Kutscher, D., Dannewitz, C., Ohlman, B., Keranen, A., and P. Hallam-Baker, "Naming Things with Hashes", draft-farrell-decade-ni-10 (work in progress), August 2012.

[I-D.hallambaker-decade-ni-params]

Hallam-Baker, P., Stradling, R., Farrell, S., Kutscher, D., and B. Ohlman, "The Named Information (ni) URI Scheme: Optional Features", draft-hallambaker-decade-ni-params-03 (work in progress), June 2012.

[RFC2045] Freed, N. and N. Borenstein, "Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies", RFC 2045, November 1996.

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC5234] Crocker, D. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", STD 68, RFC 5234, January 2008.
- [RFC5652] Housley, R., "Cryptographic Message Syntax (CMS)", STD 70, RFC 5652, September 2009.
- [RFC5785] Nottingham, M. and E. Hammer-Lahav, "Defining Well-Known Uniform Resource Identifiers (URIs)", RFC 5785, April 2010.

## 9.2. Informative References

- [I-D.farrell-dtnrg-bpq]  
Farrell, S., Lynch, A., Kutscher, D., and A. Lindgren,  
"Bundle Protocol Query Extension Block",  
draft-farrell-dtnrg-bpq-01 (work in progress), March 2012.
- [RFC4838] Cerf, V., Burleigh, S., Hooke, A., Torgerson, L., Durst, R., Scott, K., Fall, K., and H. Weiss, "Delay-Tolerant Networking Architecture", RFC 4838, April 2007.
- [RFC5050] Scott, K. and S. Burleigh, "Bundle Protocol Specification", RFC 5050, November 2007.
- [RFC5751] Ramsdell, B. and S. Turner, "Secure/Multipurpose Internet Mail Extensions (S/MIME) Version 3.2 Message Specification", RFC 5751, January 2010.
- [ref.netinf-db2]  
SAIL, "NetInf Content Delivery and Operations", SAIL Project Deliverable D-3.2 , May 2012.

## Authors' Addresses

Dirk Kutscher  
NEC  
Kurfuersten-Anlage 36  
Heidelberg,  
Germany

Phone:  
Email: kutscher@neclab.eu

Stephen Farrell  
Trinity College Dublin  
Dublin, 2  
Ireland

Phone: +353-1-896-2354  
Email: [stephen.farrell@cs.tcd.ie](mailto:stephen.farrell@cs.tcd.ie)

Elwyn Davies  
Trinity College Dublin  
Dublin, 2  
Ireland

Phone: +44 1353 624 579  
Fax:  
Email: [davieseb@scss.tcd.ie](mailto:davieseb@scss.tcd.ie)  
URI:



ICN Research Group  
Internet-Draft  
Intended status: Informational  
Expires: April 23, 2013

L. Li  
X. Xu  
J. Wang  
Z. Hao  
ZTE Corporation  
October 20, 2012

Information-Centric Network in an ISP  
draft-li-icnrg-icn-isp-01

Abstract

Information-Centric Network (ICN) may be deployed over different underlying networks, e.g. ad hoc networks, DTN and ISP's networks. This document discusses deploying ICN in an ISP's existing networks and ICN design for ISPs.

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 April 23, 2013.

Copyright Notice

Copyright (c) 2012 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 (<http://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 . . . . .	3
2. Terminology . . . . .	3
3. Deployment Considerations in an ISP . . . . .	3
4. Routing and Caching Control . . . . .	4
5. ICN with a Centralized Controller . . . . .	5
6. Security Considerations . . . . .	8
7. References . . . . .	8
7.1. Normative References . . . . .	8
7.2. Informative References . . . . .	8
Authors' Addresses . . . . .	8

## 1. Introduction

Information-Centric Network (ICN) may be deployed over different underlying networks, e.g. ad hoc networks, DTN and ISP's networks. This document discusses deploying ICN in an ISP's existing networks and ICN design for ISPs.

## 2. 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].

## 3. Deployment Considerations in an ISP

Information-centric networks can be deployed on top of layer-3 or layer-2 networks. It should be preferable for ISPs to deploy ICN as an overlay network on top of layer-3 networks, for the following considerations: firstly, in the case of incremental deployment, packets between newly deployed content routers have to go through ordinary routers which do not understand ICN protocols; secondly, content routers should be preferably deployed in areas with requirements of reducing cost or improving Quality of Service (QoS), and there is no necessity of deployment in areas where QoS requirements can be fulfilled, and link cost is lower.

Content routers may be deployed at the edges of networks close to content consumers, for the following considerations: firstly, early cache hit at network edges means better QoS and more link cost savings; secondly, deploying caches at network edges can mitigate the impact of unstable wireless link in the case of mobile access users; thirdly, it is easier to handle the requests since traffic is light at network edges, and cache hits at network edges reduce the load at content routers in core network which forwarding high volume traffic.

Content routers with huge cache spaces may be deployed in core networks to achieve high cache hit rates. Research on cache, e.g. [web\_caching] and [cooperative\_caching], shows that both cache size and serving user number affect cache hit rate. Though early cache hit is better, cache hit rate at network edge is limited. An edge content router's cache hit rate is limited by its cache size and serving user number. Firstly, in order to reach a high cache hit rate, huge cache space is needed. But it's costly to deploy huge cache spaces in large number of edge content routers. Secondly, fewer users are served by an edge content router. As a result, a large proportion of content requests are for one-time access

contents, and hit rate is limited at network edges.

It is not necessary to deploy a deep hierarchy of content routers in an ISP. On one hand, it is easier to deploy fewer content routers in current network. On the other hand, it is preferable that the cache space of a content router is much bigger than the one in a lower tier, which means the number of tiers is small. Because of the Zipf-like distribution of content requests, the cache size must grow exponentially when the tier grows. Otherwise, cache hit rate of each non-bottom tier is very low.

#### 4. Routing and Caching Control

There are two ways to collect topology data and generate routing table, namely, self-generation and centralized generation. In the self-generation way, content routers run routing protocols to exchange topology data inside an AS or among ASes. Then each content router runs a routing algorithm locally to generate a routing table independently. Alternatively, inside an AS, content routing tables can be generated in a centralized way. In this way, one or more controllers collect topology data, and generate routing tables for all the content routers. Then the controller(s) sends route entries to content routers.

There are also two ways to control caching. A content router can decide to cache a content or not on its own by running a cache replacement algorithm like LRU or LFU. However, an ISP may also want to use centralized controller(s) to enforce some cache policies.

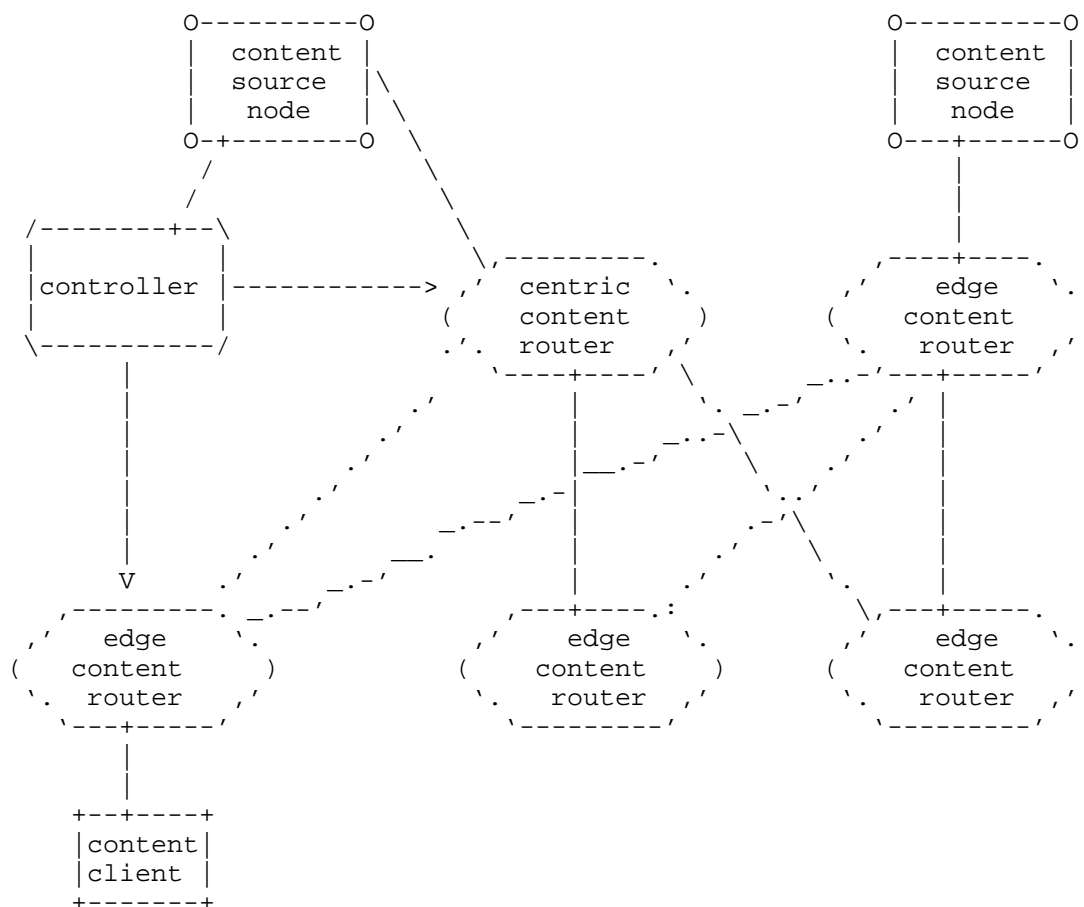
An ISP may utilize centralized controller(s) to enforce routing and cache policy under following considerations. First, to meet QoS requirement, an ISP may decide routing path and cache resource assignment based on factors like content type, content download frequency and distance to content source. Second, to reduce link cost, an ISP may assign more cache resource for the contents passing through costly links by controlling routing path and/or cache priority. Third, to balance link load and cache load, an ISP may optimize routes based on load status. Fourth, an ISP may provide better services to paid users or content providers by controlling routing path and/or cache priority.

To control routing and caching, an ICN controller may need to collect not only topology data and traffic data, but also content data like content type and content download frequency.



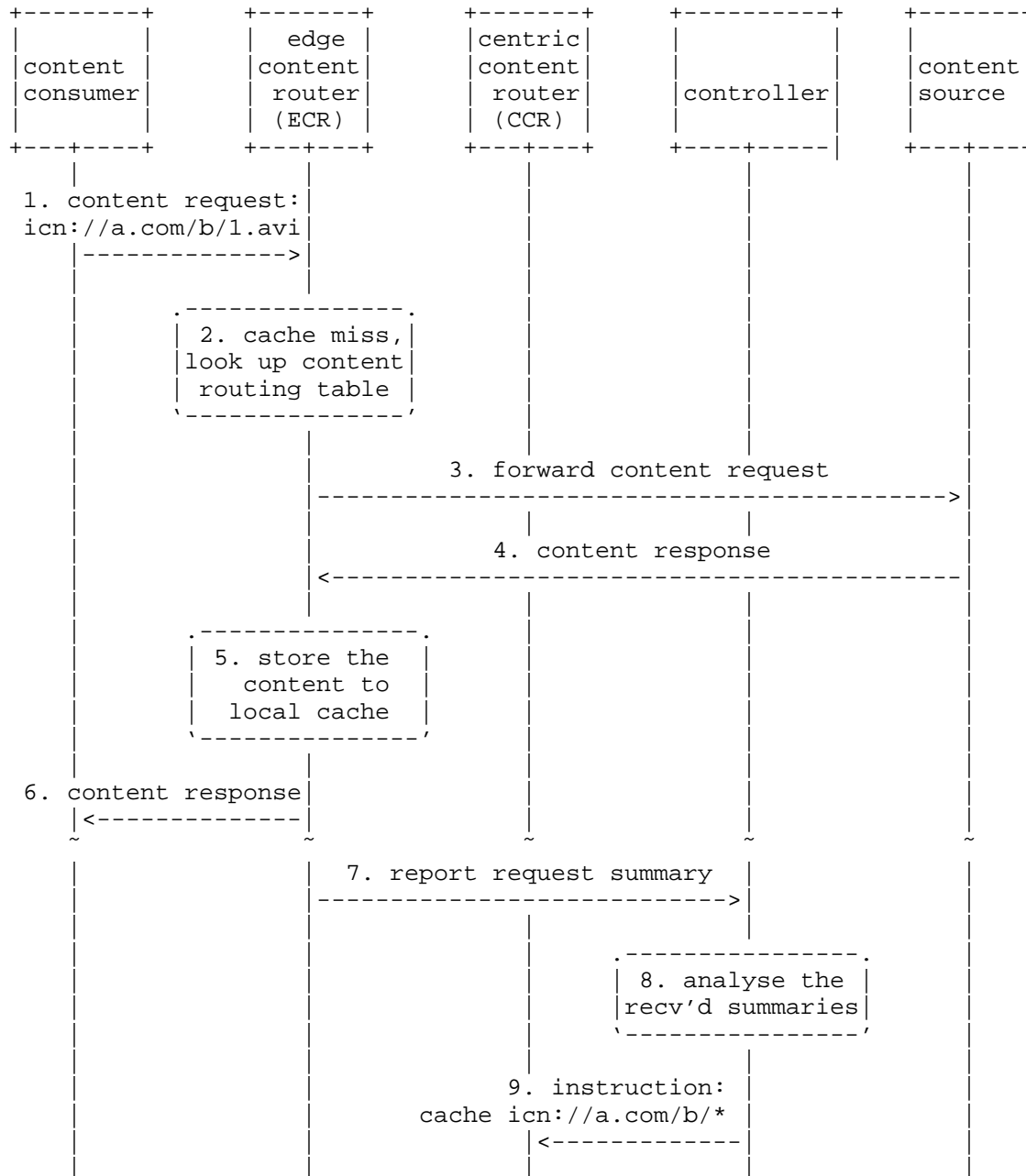
## 5. ICN with a Centralized Controller

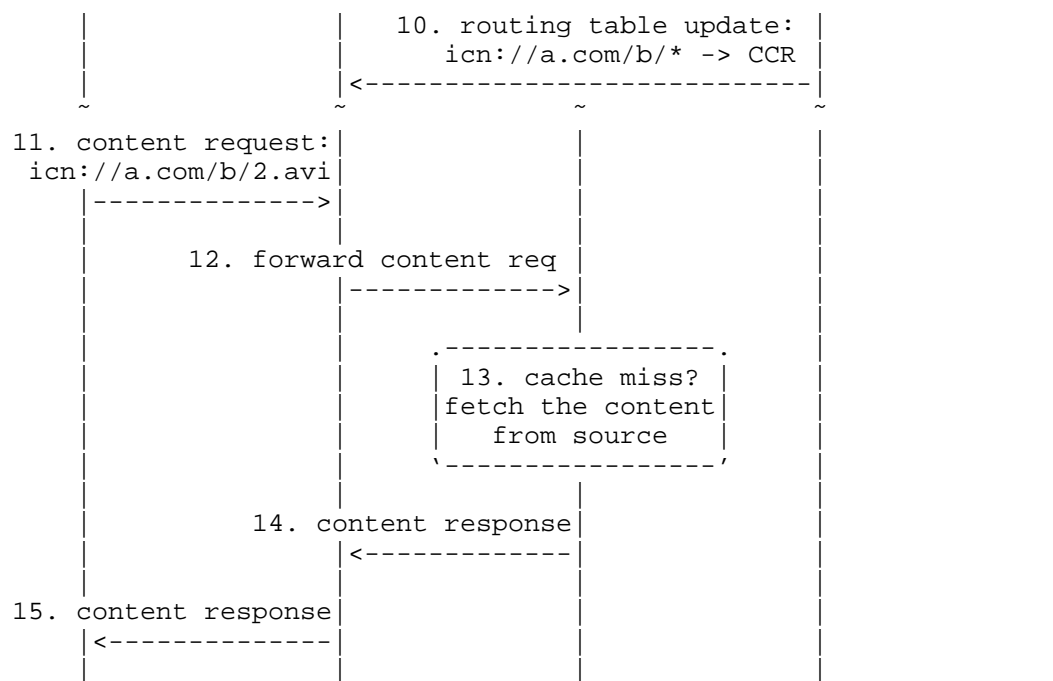
The figure below shows an example of ICN in an ISP. In this example, there are two tiers of content routers, a tier of edge content routers with small cache spaces and a tier of centric content routers with huge cache spaces. To store massive contents, the centric content routers use cache clusters. The ICN network is an overlay network deployed over IP network. An ICN controller is responsible for generating routing tables and sending route entries to content routers.



The figure below depicts the roll of a centralized controller in the ICN. Content routing tables of the routers and caching policy of the centric content router (CCR) are all generated by the controller according to its analysis of collected information, and ISP policies can be enforced. When a content router starts up, it discovers the

controller in the domain, registers at the controller, and obtains its initial content routing table which is updated by the controller afterward.





As shown by steps 1 to 6, upon receiving the content consumer's first request to a video content in `icn://a.com/b/`, the edge content router looks up the routing table, and forwards the request to the content source. Upon receiving the response, it decides independently to cache the content for a later use, according to a local cache replacement algorithm.

As shown by steps 7 to 10, the controller collects request statistic and generate routing tables and CCR caching policy in a centralized way. Each content router generates a summary of requests it recently received by some sampling techniques, and sends the summary to the controller periodically. The controller generates content routing table according to analysis of the summaries and the ISP's policies, and sends the routing table updates to the routers. The controller may decide that the centric content router stores entire or parts of a content source site with higher request frequency. The centric content router may prefetch the contents from the source site. The edge content routers update their routing tables accordingly. A routing table item in an aggregated form (in this example, `icn://a.com/b/*`) will direct the requests to the centric content router.

As shown by steps 11 to 15, upon receiving the content consumer's second request to a video content in `icn://a.com/b/`, the edge content

router forwards the request to the centric content routers.

## 6. Security Considerations

TBD

## 7. References

### 7.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

### 7.2. Informative References

[cooperative\_caching]

Wolman, A., Voelker, G., Sharma, N., Cardwell, N., Karlin, A., and H. Levy, "On the Scale and Performance of Cooperative Web Proxy Caching", ACM Symposium on Operating Systems Principles, 1999.

[web\_caching]

Breslau, L., Cao, P., Fan, L., Phillips, G., and S. Shenker, "Web Caching and Zipf-like Distributions: Evidence and Implications", INFOCOM, 1999.

## Authors' Addresses

Lichun Li  
ZTE Corporation  
Zijinghua Road 68  
Yuhuatai District, Nanjing 210012  
P. R. China

Email: li.lichun1@zte.com.cn

Xin Xu  
ZTE Corporation  
Zijinghua Road 68  
Yuhuatai District, Nanjing 210012  
P. R. China

Email: xu.xin18@zte.com.cn

Jun Wang  
ZTE Corporation  
Zijinghua Road 68  
Yuhuatai District, Nanjing 210012  
P. R. China

Email: wang.jun17@zte.com.cn

Zhenwu Hao  
ZTE Corporation  
Zijinghua Road 68  
Yuhuatai District, Nanjing 210012  
P. R. China

Email: hao.zhenwu@zte.com.cn



ICNRG  
Internet-Draft  
Intended Status: Informational  
Expires: May 10, 2013

K. Pentikousis  
Huawei  
B. Ohlman  
Ericsson  
November 6, 2012

ICN Baseline Scenarios  
draft-pentikousis-icn-scenarios-00

## Abstract

This document presents scenarios for information-centric networking (ICN) which can be used to establish a common understanding about potential experimental setups where different approaches can be tested and compared against each other. The scenarios are primarily based on published literature, that is, they have all been considered in one or more performance evaluation studies, which are already available to the community. The scenarios selected for inclusion in this first draft aim to exercise a variety of aspects that an ICN solution can address. They include a) general aspects, such as, network efficiency, mobility support, multicast and caching performance, real-time communication efficacy, disruption and delay tolerance; and b) ICN-specific aspects, such as, information security and trust, persistence, availability, provenance, and location independence.

## Status of this Memo

This Internet-Draft is submitted to IETF in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

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."

The list of current Internet-Drafts can be accessed at  
<http://www.ietf.org/lid-abstracts.html>

The list of Internet-Draft Shadow Directories can be accessed at

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

#### Copyright and License Notice

Copyright (c) 2012 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 (<http://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
2	ICN Baseline Scenarios . . . . .	3
2.1	Social Networking . . . . .	3
2.2	Real-time A/V Communications . . . . .	4
2.3	Mobile Networking . . . . .	5
2.4	Infrastructure Sharing . . . . .	6
2.5	Content Dissemination . . . . .	7
2.6	Energy Efficiency . . . . .	8
2.7	Delay and Disruption Tolerance . . . . .	8
3	Security Considerations . . . . .	8
4	IANA Considerations . . . . .	8
5	Acknowledgments . . . . .	8
6	Informative References . . . . .	9
	Authors' Addresses . . . . .	10

#### 1 Introduction

Information-centric networking (ICN) marks a fundamental shift in communications and networking. In contrast with the omnipresent, and very successful we may add, host-centric paradigm, based on perpetual connectivity and the end-to-end principle, ICN changes the focal point of the network architecture from the "end host" to "information" (or content, or data). In this paradigm, connectivity can be intermittent in general; end-host and in-network storage can be capitalized upon transparently as bits in the network and on some storage device have exactly the same value; mobility, multicasting and multiaccess are supported by default; and energy efficiency is a



design consideration from the beginning.

Although interest in ICN is growing rapidly, ongoing work on different architectures, such as, for example, NetInf [NetInf], CCN and NDN [CCN], the publish-subscribe Internet (PSI) architecture [PSI], and the data-oriented architecture [DONA] is far from being completed. The increasing interest and the plethora of ICN approaches make this a very active research area but, on the downside, it makes it more difficult to compare different proposals on an equal ground.

It is not uncommon that different researchers select different performance evaluation scenarios in order to highlight the advantages of their approach. This is reasonable and should be expected to some degree. As Ahlgren et al. note [SoA], describing these architectures is akin to shooting a moving target. We find that comparing these different approaches is often even more tricky. Nevertheless, certain scenarios seem to emerge where said ICN architectures could showcase their superiority over current systems, in general, and against each other, in particular.

This document collects several scenarios from the published ICN literature and aims to use them as foundation for the baseline scenarios to be considered by the IRTF Information-Centric Networking Research Group (ICNRG) in its future work. The list of scenarios can obviously change, as input from the research group is received.

## 2 ICN Baseline Scenarios

This section presents a number of scenarios grouped into several categories. Note that certain evaluation scenarios span across these categories, so the boundaries between them should not be considered rigid and inflexible. The goal is that each scenario should be described at a sufficient level of detail so that it can serve as the base for comparative evaluations of different approaches. This will need to include reference configurations, specifications of traffic mixes and traffic loads. These specifications/configurations should preferably come as sets that describe extremes as well as "typical" usage scenarios.

### 2.1 Social Networking

Social networking applications proliferated over the past decade based on overlay content dissemination systems that require large infrastructure investments to rollout and maintain. Content dissemination is at the heart of the ICN paradigm and, therefore, we

would expect that they are a "natural fit" for showcasing the superiority of ICN over traditional client-server TCP/IP-based systems.

Mathieu et al. [ICN-SN], for instance, illustrate how an ISP can capitalize on CCN to deploy a short-message service akin to Twitter at a fraction of the complexity of today's systems. Their key observation is that such a service can be seen as a combination of multicast delivery and caching. That is, a single user addresses a large number of recipients, some of which receive the new message immediately as they are online at that instant, while others receive the message whenever they connect to the networks.

Earlier work by Arianfar et al. [CCR] considers a similar pull-based content retrieval scenario using a different architecture, pointing to significant performance advantages. Although the authors consider a different network topology and do not explicitly say that their evaluation scenario is addressing social networking, the similarities are easy to spot: "followers" obtain content put "on the network" by a single user relying solely on network primitives. That is, in both evaluations there is no need for a classic client-server architecture (let alone a cloud-based infrastructure) to intermediate between content providers and consumers.

This scenario aims to exercise each ICN architecture in terms of network efficiency, multicast support, and caching performance.

## 2.2 Real-time A/V Communications

Real-time audio and video (A/V) communications include an array of services ranging from one-to-one voice calls to multi-party multi-media conferences with video and whiteboard support to augmented reality. Real-time communications have been studied (and deployed widely) in the context of packet- and circuit-switched networks for decades. The stringent quality of service requirements that this type of communication imposes on network infrastructure is well-known. However, the ICN community has, so far, only scratched the surface of this area with respect to illustrating the benefits of adopting an information-centric approach as opposed to a host-centric one.

Notably, Jacobson et al. [VoCCN] presented an early evaluation where the performance of a VoIP call over an information-centric approach was compared with that of an off-the-shelf VoIP implementation using RTP/UTP. The results indicated that despite the extra cost of adding security support in the former case, performance was virtually identical in the two cases evaluated in a testbed. However, the

experimental setup was quite rudimentary and the evaluation considered a single voice call only. This scenario does illustrate that VoIP is feasible with at least one ICN approach, but it would need to be further enhanced to include more comprehensive metrics as well as standardized call arrival patterns, for example, following well-established methodologies from the quality of service/experience (QoS/QoE) evaluation toolbox.

Given the wide-spread deployment of real-time A/V communications, an ICN approach should show not only feasibility but highlight that complexity is significantly reduced when compared to a classic IP-based A/V application. For example, with respect to multimedia conferencing, Zhu et al. [ACT] describe the design of a distributed audio conference tool based on NDN. The design includes ICN-based conference discovery, speakers discovery and voice data distribution. The reported evaluation results point to gains in scalability and security. Moreover, Chen et al. [G-COPSS] explore the feasibility of implementing a Massively Multiplayer Online Role Playing Game (MMORPG) based on CCNx and show that stringent temporal requirements can be met while scalability is significantly improved when compared to an IP client-server system.

In short, scenarios in this category should illustrate not only feasibility but increased scalability, reliability, and capacity to meet stringent QoS/QoE requirements when compared to established host-centric solutions.

### 2.3 Mobile Networking

IP mobility management relies on mobility anchors to provide ubiquitous connectivity to end-hosts as well as moving networks. This is a natural choice for a host-centric paradigm that requires end-to-end connectivity and continuous network presence [SCES]. An implicit assumption in host-centric mobility management frameworks is that the mobile node aims at connecting to a particular peer, not at retrieving information [EEMN]. However, with ICN new ideas about mobility management should come to the forefront, which capitalize on the different nature of the paradigm.

For example, Dannewitz et al. [N-Scen], consider a scenario where a multiaccess end-host can retrieve email securely using a combination of cellular and wireless local area network connectivity. This scenario borrows elements from previous work, e.g. [DTI], and develops them further with respect to multiaccess. Unfortunately, Dannewitz et al. [N-Scen] do not present any results demonstrating that an ICN approach is indeed better. That said, the scenario is interesting as it considers content specific to a single user (i.e.

her mailbox) and does point to a decrease in complexity. It is also compatible with recent work in the Distributed Mobility Management (DMM) Working Group within the IETF. Finally, Xylomenos et al. [PSIMob] as well as [EEMN] argue that an information-centric architecture can avoid the complexity of having to manage tunnels to maintain end-to-end connectivity as is the case with mobile anchor-based protocols such as Mobile IP (and its variants).

Overall, mobile networking scenarios have not been developed in detail, let alone evaluated in a wide scale. We expect that in the coming period more papers will address this topic, each perhaps proposing its own evaluation scenario. The scenarios in mobile networking will be naturally coupled with those discussed in the previous sections as more users access social networking and A/V applications through mobile devices.

Mobile networking scenarios should aim to exercise service continuity for those applications that require it, decrease complexity and control signaling for the network infrastructure, as well as increase wireless capacity utilization by taking advantage of the broadcast nature of the medium.

## 2.4 Infrastructure Sharing

A key idea in ICN is that the network should secure information objects per se, not the communications channel that they are delivered over. This means that hosts attached to an information-centric network can share resources in an unprecedented scale, especially when compared to what is possible in an IP network. All devices with network access and storage capacity can contribute their resources increasing the value of an information-centric network (perhaps) much faster than Metcalfe's law.

For example, Jacobson et al. [CBIS] argue that in ICN the "where and how" to obtain information are new degrees of freedom. They illustrate this with a scenario involving a photo sharing application which takes advantage of whichever access network connectivity is available at the moment (WLAN, Bluetooth, and even SMS) without requiring a centralized infrastructure to synchronize between numerous devices. It is important to highlight that since the focus of the communication changes, keep-alives in this scenario are simply unnecessary, as devices participating in the testbed network contribute resources in order to maintain user content consistency, not link state information as is the case in the host-centric paradigm. This means that the notion of "infrastructure" may be completely different in the future.

Carofiglio et al., for instance, present early work on an analytical framework that attempts to capture the storage/bandwidth tradeoff and can be used as a basis for a network planning tool [SHARE]. In addition, Chai et al. [CL4M] explore the benefits of ubiquitous caching throughout an information-centric network and argue that "caching less can actually achieve more." These two papers indicate that there is a lot of work to be done in the area of how to use optimally all resources that end hosts bring into the network.

Scenarios in this category, therefore, would cover the communication/computation/storage tradeoffs that an ICN network deployment must consider, including network planning, perhaps capitalizing on user-provided resources, as well as operational and economical aspects to illustrate the superiority of ICN over other approaches, including federations of IP-based Content Distribution Networks (CDNs).

## 2.5 Content Dissemination

Content dissemination has attracted more attention than other aspects of ICN, perhaps due to a misunderstanding of what the first "C" in CCN stands for. Decentralized content dissemination with on-the-fly aggregation of information sources was envisaged in [N-Scen] where information objects can be dynamically assembled based on hierarchically structured subcomponents. For example, a video stream could be associated with different audio streams and subtitle sets, which all can be obtained from different sources. Semantics and content negotiation, on behalf of the user was also considered, e.g. for the case of popular tunes. Effectively this scenario has the information consumer issuing independent requests for content based on information identifiers, and stitching the pieces together irrespective of "where" or "how" they were obtained.

Content dissemination scenarios have a large overlap with the scenarios described above [DONA, PSI, PSI-Mob, NetInf, CCN, CBIS, CCR], just to name a few. In addition, Chai et al. present a hop-by-hop hierarchical content resolution approach [CURLING], which employs receiver-driven multicast over multiple domains, advocating another content dissemination approach.

Scenarios in this category abound in the literature, including stored and streaming A/V distribution, file distribution, mirroring and bulk transfers, SVN-type of services, as well as traffic aggregation. We expect that in particular for content dissemination both extreme as well as typical scenarios can be specified drawing data from current CDN deployments.

## 2.6 Energy Efficiency

As mentioned earlier, energy efficiency can be tackled by ICN in ways that it cannot in a host-centric paradigm. For example, the work by Guan et al. [EECCN] indicates that CCN may be much more energy-efficient than traditional CDNs for delivering popular content given the current networking equipment energy consumption levels.

Evaluating energy efficiency does not require the definition of new scenarios, but does require the establishment of clear guidelines so that different ICN approaches can be compared not only in terms of scalability, for example, but also in terms to power consumption.

## 2.7 Delay and Disruption Tolerance

Delay Tolerant Networking (DTN) [DTN] was originally designed for special use cases, such as interstellar networking, use of data mules, and so on. With the advent of sensor networks and peer-to-peer (P2P) networking between mobile nodes, DTN is becoming a more commonplace type of networking. ICN does not build on the familiar communication abstraction of end-to-end connectivity between a set of nodes. This makes it possible to include DTN support in ICN natively. Thus it is of interest to evaluate to which extent different ICN technologies can support DTN scenarios.

Important aspects to be evaluated with respect to delay and disruption tolerance include, but are not limited to, name resolution, routing and forwarding in disconnected parts of the network; support for unidirectional links; number of round trips needed to complete a data transfer, and so on.

## 3 Security Considerations

TBD

## 4 IANA Considerations

This document presents no IANA considerations.

## 5 Acknowledgments

TBD

## 6 Informative References

- [NetInf] Ahlgren, B. et al., "Design considerations for a network of information", Proc. CoNEXT Re-Arch Workshop. ACM, 2008.
- [CCN] Jacobson, V. et al., "Networking Named Content", Proc. CoNEXT. ACM, 2009.
- [PSI] Trossen, D. and Parisis, G., "Designing and realizing an information-centric internet", IEEE Commun. Mag., vol. 50, no. 7, July 2012.
- [DONA] Koponen, T. et al., "A Data-Oriented (and Beyond) Network Architecture", Proc. SIGCOMM. ACM, 2007.
- [SoA] Ahlgren, B. et al., "A survey of information-centric networking", IEEE Commun. Mag., vol. 50, no. 7, July 2012.
- [ICN-SN] Mathieu, B. et al., "Information-centric networking: a natural design for social network applications", IEEE Commun. Mag., vol. 50, no. 7, July 2012.
- [CCR] Arianfar, S. et al., "On content-centric router design and implications", Proc. CoNEXT Re-Arch Workshop. ACM, 2010.
- [VoCCN] Jacobson, V. et al., "VoCCN: Voice-over Content-Centric Networks", Proc. CoNEXT Re-Arch Workshop. ACM, 2009.
- [ACT] Zhu, Z. et al., "ACT: Audio Conference Tool Over Named Data Networking", Proc. SIGCOMM ICN Workshop. ACM, 2011.
- [G-COPSS] Chen, J. et al., "G-COPSS: A Content Centric Communication Infrastructure for Gaming Applications", Proc. ICDCS. IEEE, 2012.
- [SCES] Allman, M. et al., "Enabling an Energy-Efficient Future Internet through Selectively Connected End Systems", Proc. HotNets-VI. ACM, 2007.
- [EEMN] Pentikousis, K., "In Search of Energy-Efficient Mobile Networking", IEEE Commun. Mag., vol. 48, no. 1, Jan. 2010.
- [N-Scen] Dannewitz, C. et al., "Scenarios and research issues for a Network of Information", Proc. MobiMedia. ICST, 2012.
- [DTI] Ott, J. and Kutscher, D., "Drive-thru Internet: IEEE 802.11b for 'automobile' users", Proc. INFOCOM. IEEE, 2004.

- [PSIMob] Xylomenos, G. et al., "Caching and Mobility Support in a Publish-Subscribe Internet Architecture", IEEE Commun. Mag., vol. 50, no. 7, July 2012.
- [CBIS] Jacobson, V. et al., "Custodian-Based Information Sharing", IEEE Commun. Mag., vol. 50, no. 7, July 2012.
- [SHARE] Carofiglio, G. et al., "Bandwidth and storage sharing performance in information centric networking", Proc. SIGCOMM ICN Workshop. ACM, 2011.
- [CL4M] Chai, W. K. et al., "Cache 'Less for More' in Information-centric Networks", Proc. Networking. IFIP, 2012.
- [CURLING] Chai, W. K. et al., "CURLING: Content-Ubiquitous Resolution and Delivery Infrastructure for Next-Generation Services", IEEE Commun. Mag., vol. 49, no. 3, Mar. 2011.
- [EECCN] Guan, K. et al., "On the Energy Efficiency of Content Delivery Architectures ", Proc. ICC Workshops. IEEE, 2011.
- [DTN] Fall, K., "A delay-tolerant network architecture for challenged internets", Proc. SIGCOMM. ACM, 2003.

#### Authors' Addresses

Kostas Pentikousis  
Huawei Technologies  
Carnotstrasse 4  
10587 Berlin  
Germany

Email: k.pentikousis@huawei.com

Borje Ohlman  
Ericsson Research  
S-16480 Stockholm  
Sweden

Email: Borje.Ohlman@ericsson.com



Network Working Group  
Internet-Draft  
Intended status: Informational  
Expires: April 18, 2013

J. Ren  
City University of Hong Kong  
W. Liu  
Huawei Technologies  
J. Wang  
F. Tang  
City University of Hong Kong  
October 15, 2012

On the Content Retrieval In Information-Centric Network  
draft-ren-icn-content-retrieval-00

Abstract

Information-Centric Network(ICN), as an emerging network architecture, focus on delivering content more efficiently. This brief paper discusses some issues about content retrieval in the Information-Centric Network. It first lists a set of basic requirements on the basis of previous literatures. Then, it tries to identify the fundamental functionalities required to satisfy the foregoing requirements. Last, it summarizes the implementation status of such functionalities in various Information-Centric Network architectures.

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 April 18, 2013.

Copyright Notice

Copyright (c) 2012 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  
(<http://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. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

1. Introduction . . . . .	3
2. Terminology . . . . .	3
3. Requirements of content retrieval . . . . .	4
4. Fundamental functionalities for content retrieval . . . . .	4
5. Summary . . . . .	5
5.1. Content name resolution . . . . .	5
5.2. k-anycast and multicast . . . . .	6
5.3. Content replication . . . . .	7
5.4. In-network cache discovery . . . . .	8
6. IANA Considerations . . . . .	9
7. Security Considerations . . . . .	9
8. References . . . . .	9
8.1. Normative References . . . . .	9
8.2. Informative References . . . . .	10
Authors' Addresses . . . . .	10

## 1. Introduction

Information-Centric Network (also known as content-oriented network [CON], data-oriented network [DON], or name-based network [NBN]) is a new communication paradigm which puts the content at the center of the network architecture. It is motivated by the mismatch between the increasing demand for highly scalable and efficient distribution of content and inefficiently content delivery of the traditional host-centric communication paradigm.

Content distribution, including file sharing and media streaming, has contributed to the ever-increasing Internet traffic nowadays. Content consumers, in general, are more concerned about what content they want than where that content resides. Unfortunately, current Internet is based on a host-centric communication paradigm where a consumer has to specify where the content is and the network only can deliver the content from the specified source to the consumer. Moreover, it is usually difficult to exploit many superior technologies, such as multicast, k-anycast, etc.

To overcome the aforementioned issues, several Information-Centric Network designs have been proposed. In these designs, each content is given a unique name which is not associated with its location. Consumers can use the content name to request the content which they intend to get. More importantly, all the routings are based on the content name, which enables the request to be routed to any potential content sources. Moreover, many technologies, including content replication, caching, k-anycast etc., can now be used to achieve faster content retrieval.

Although the Information-Centric Network is considered as a promising way to solve the aforementioned issues, ICN is still a work in progress and there is a long way to go for standardizing it. This paper tries to list a set of basic requirements and fundamental functionalities for content retrieval in ICN. We hope this work will benefit the design of ICN.

## 2. 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 [RFC2119].

Content original provider: an end entity that publishes and provides the content.

Content consumer: an end entity that wants to retrieve a content.

Content holder: any entity that has a complete copy of the content (including provider, router, cache, and other devices which hold the content).

Content name: every content is given a unique name. Content consumer requests the content by specifying its name rather than its location.

### 3. Requirements of content retrieval

To provide efficient content delivery, several requirements should be satisfied:

1. Persistency and unique content name. Each content should be given a unique name. This name should be valid as long as the corresponding content is available. In other words, the content name should not be changed no matter where it is.
2. Availability. Availability means the content should be reachable at all times with low-latency. Replication can be used to guarantee availability, and the network is responsible for finding the nearby copies which have low-latency.
3. Failure recovery. End-to-end communication is often disrupted due to the endpoint mobility and link/router failure. These disruptions should not disturb the content delivery.
4. Authenticity. Any communication entities, including router, end-device, application, etc., can verify whether the content comes from the authenticated source.
5. Bandwidth efficiency. Since the volume of content is huge, appropriate technologies should be used to minimize bandwidth consumption.

### 4. Fundamental functionalities for content retrieval

The requirements listed above are the most important ones. More requirements will be added further. To satisfy these requirements, several fundamental functionalities should be provided:

1. Content name resolution. Name resolution service translates content name into network location. In ICN, multiple content holders can provide the same content. The requests should be directed to the intended content holder according to some policies, such as lowest delivery latency, minimum bandwidth utilization, etc.

2. K-anycast. K-anycast is a communication model which allows K servers to cooperate with each other to accomplish content delivery. With k-anycast, the network can deliver the content much faster and provide quick failure recovery.

3. Multicast. Multicast can be used to deliver content to all the content consumers interested in the content. It can effectively save the network bandwidth.

4. Content replication. Content can be stored in end-hosts according to some off-line replication policies. Well-designed replication policy can reduce the content retrieval time.

5. In-network cache discovery. In-network caching is considered as one of the most significant properties of ICN. To fully utilize the caches, however, the network needs a mechanism to discover the caches.

## 5. Summary

This section summarizes the implementation status of the aforementioned fundamental functionalities in different ICN projects. In this draft, EU FP7 projects, US NSF projects and some academic projects are compared.

### 5.1. Content name resolution

SAIL/4 WARD	A Multilevel Distributed Hash Table (MDHT) is used to establish a global resolution system.
PSIRP/PUIRUIT	Following a pub-sub communication model, a rendezvous system is employed to notify the appropriate content holder to send content to the requested consumers.
CONVERGENCE	Dedicate Name-System-Nodes (similar to DNS) are deployed.
TRIAD	An Internet Relay Protocol (INRP) is designed to perform name-to-address conversion in TRIAD by using the routing information maintained by relay nodes.
COMET	A content resolution function is used to resolve content names to content properties.
COAST	Search engine is used to partially provide content name resolution service.
Postcards from the Edge	A File Name Resolution System is employed to resolve file names to potential cached locations.
MobilityFirst	A Global name resolution service (GNRS) is developed for Globally Unique Flat Identifier (GUID) to Network Address (NA) mapping. The GNRS is implemented based on DHT between routers.
NDN	Packets are routed based on the content name instead of resolving the content name to an underlying address.

## 5.2. k-anycast and multicast

SAIL/4 WARD	Different chunks are retrieved from different locations via several concurrent HTTP connections.
PSIRP/PUIRUIT	Forwarding Identifier (FID) is used for source routing. Multiple FIDs can be integrated to form a multicast tree.
CONVERGENCE	Traditional point-to-point sessions between two upper layer entities can be supported by coupling the upper layer entities with named-service-assess-points. This functionality can be extended to support multicast.
TRIAD	The EXPRESS single-source model of multicast can be supported.
COMET	NOT MENTIONED.
COAST	Multicast can be achieved through an Information Overlay.
Postcards from the Edge	Multicast trees are set up by using a Rendezvous Point (RP).
MobilityFirst	Multicast GUID are mapped to multiple consumers by GNRS.
NDN	Since PIT(Pending Interest Table) includes the set of interfaces over which Interests have arrived, multicast functionality can naturally be supported.

### 5.3. Content replication

SAIL/4 WARD	The content, called Information Objects (IOs), can be stored in the NetInf architecture to speed up content retrieval.
PSIRP/PUIRUIT	The most popular objects are replicated to different k storage devices by using a greedy algorithm.
CONVERGENCE	Multiple replica nodes can be provisioned as Content Delivery Network.
TRIAD	Content are replicated at multiple sites and DNS lookups will be redirected to the nearby site.
COMET	Replication can be supported. However, no concrete scheme is proposed.
COAST	The content may be stored/cached at the Information Overlay or at the lower hierarchy layer.
Postcards from the Edge	Replication can be supported. However, no concrete scheme is proposed.
MobilityFirst	The global GUID-NA mapping information are replicated in k different ASes.
NDN	Replication can be supported. However, no concrete scheme is proposed.

#### 5.4. In-network cache discovery



SAIL/4 WARD	The cache information is advertised in local area.
PSIRP/PUIRUIT	The cache which is on the content request path can be used.
CONVERGENCE	The cache which is on the content request path can be used.
TRIAD	The cache which is on the content request path can be used.
COMET	The cache which is on the content request path can be used.
COAST	The cache discovery can be achieved by an Information Overlay.
Postcards from the Edge	A Caching Service Protocol is developed, which exchanges the cache information between different Cache-and-Forward Nodes (CNFs).
MobilityFirst	Delay-Tolerant Networking (DTN) liked Caching and forwarding design is proposed.
NDN	The cache which is on the content request path can be used. Cache can also be discovered through flooding content request.

## 6. IANA Considerations

This document makes no request of IANA.

## 7. Security Considerations

Security issues are not discussed in this memo.

## 8. References

### 8.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

## 8.2. Informative References

- [COAST] COAST Home Page, "<http://www.coast-fp7.eu/>".
- [COMET] COMET Home Page,  
"<http://www.comet-project.org/overview.html>".
- [CON] C. Jaeyoung, et al., "A Survey on content-oriented networking for efficient content delivery", 2011.
- [CONVERGENCE]  
CONVERGENCE Home Page, "<http://www.ict-convergence.eu/>".
- [DON] T. Koponen, et al., "A data-oriented (and beyond) network architecture", 2007.
- [MobilityFirst]  
MobilityFirst Home Page,  
"<http://mobilityfirst.winlab.rutgers.edu/>".
- [NBN] V. Jacobson, et al., "Networking named content", 2009.
- [NDN] NDN Home Page, "<http://www.named-data.net/index.html>".
- [PSIRP] PSIRP Home Page, "<http://www.psirp.org/>".
- [PURSUIT] PURSUIT Home Page,  
"<http://www.fp7-pursuit.eu/PursuitWeb/>".
- [Postcards-From-The-Edge]  
Postcards from the Edge Home Page,  
"<http://www.nets-find.net/Funded/Postcards.php>".
- [SAIL] SAIL Home Page, "<http://www.sail-project.eu/>".
- [TRIAD] TRIAD Home Page, "<http://gregorio.stanford.edu/triad/>".
- [WARD] 4WARD Home Page, "<http://www.4ward-project.eu/index.php>".

Authors' Addresses

Jing Ren  
City University of Hong Kong  
Tat Chee Avenue  
Hong Kong  
P.R. China

Email: [jingren@cityu.edu.hk](mailto:jingren@cityu.edu.hk)

Will Liu  
Huawei Technologies  
Bantian, Longgang District  
Shenzhen 518129  
P.R. China

Email: [liushucheng@huawei.com](mailto:liushucheng@huawei.com)

JianPing Wang  
City University of Hong Kong  
Tat Chee Avenue  
Hong Kong  
P.R. China

Email: [jianwang@cityu.edu.hk](mailto:jianwang@cityu.edu.hk)

Fei Tang  
City University of Hong Kong  
Tat Chee Avenue  
Hong Kong  
P.R. China

Email: [feitang@cityu.edu.hk](mailto:feitang@cityu.edu.hk)

