      Network Coding for Content-Centric Networking / Named Data Networking:
                   Requirements and Challenges
                draft-matsuzono-nwcrg-nwc-ccn-reqs-01

Abstract

   This document describes the current research outcomes regarding
   Network Coding (NC) for Content-Centric Networking (CCN) / Named Data
   Networking (NDN), and clarifies the requirements and challenges for
   applying NC into CCN/NDN.

Status of This Memo

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

   Internet-Drafts are working documents of the Internet Engineering
   Task Force (IETF). Note that other groups may also distribute
   working documents as Internet-Drafts. The list of current Internet-
   Drafts is at https://datatracker.ietf.org/drafts/current/.

   Internet-Drafts are draft documents valid for a maximum of six months
   and may be updated, replaced, or obsoleted by other documents at any
   time. It is inappropriate to use Internet-Drafts as reference
   material or to cite them other than as "work in progress."

   This Internet-Draft will expire on September 6, 2018.

Table of Contents

1. Introduction

   Information-Centric Networks in general, and Content-Centric
   Networking (CCN) [15] or Named Data Networking (NDN) [16] in
   particular, have emerged as a novel communication paradigm advocating
   to retrieve data through their names. This paradigm pushes content
   awareness into the network layer. It is expected to enable consumers
   to obtain the content they desire in a straightforward and efficient
   manner from the heterogenous networks they may be connected to. The
   CCN/NDN architecture has introduced innovative ideas and has
   stimulated research in a variety of areas, such as in-network
   caching, name-based routing, multi-path transport, content security,
   and so on. One key benefit of requesting content by name is that it
   removes the need to establish a session between the client and a
   specific server, and that content can thereby be retrieved from
   multiple sources.

In parallel, there has been a growing interest from both academia and
industry to better understand fundamental aspects of Network Coding
(NC) toward enhancing key system performance metrics such as data
throughput, robustness and reduction in the required number of
transmissions through connected networks, point-to-multipoint
connections, etc. Typically, NC is a technique mainly used to encode
packets to recover lost source packets at the receiver, and to
effectively get the desired information in a fully distributed
manner. In addition, NC can be used for security enhancements
[2][3][4][5].

NC aggregates multiple packets with parts of the same content
together, and may do this at the source or at other nodes in the
network. As such, network coded packets are not connected to a
specific server, as they may have evolved within the network. Since
NC focuses on what information should be encoded in a network packet,
rather than the specific host where it has been generated, it is in
line with the CCN/NDN core networking layer (described in more detail
later on). NC has already been implemented for information/content
dissemination (e.g. [6][7][8]). NC provides CCN/NDN with the highly
beneficial potential to effectively disseminate information in a
completely independent and decentralized manner. [9] first suggested
to exploit NC techniques to enhance key system performances in ICN,
and others have considered NC in ICN use cases such as content
dissemination [10], seamless mobility [11], joint caching and network
coding [12][13], low-latency video streaming [14], etc.

In this document, we consider how NC can be applied to the CCN/NDN
architecture and describe the requirements and potential challenges
for making CCN/NDN-based communications better using the NC
technology. Please note that providing specific solutions (e.g., NC
optimization methods) to enhance CCN/NDN performance metrics by
exploiting NC is out of scope of this document.

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 [1].

2.1. Definitions

The terminology regarding NC used in this document is described
below. It is aligned with RFCs produced by the FEC Framework
(FECFRAME) IETF Working Groups as well as recent activities in the
Network Coding Research Group [18].

o Random Linear Coding (RLC): Particular case of Linear Coding using
   a set of random coding coefficients.

o Generation, or (IETF) Block: With Block Codes, the set of content
   data that are logically grouped into a Block, before doing
   encoding.

o Generation Size: With Block Codes, the number k of content data
   belonging to a Block.

o Encoding Vector: A set of coding coefficients used to generate a
   certain coded packet through linear coding. The number of nonzero
   coefficients in the Coding Vector defines its density

o Finite Field: Finite fields, used in Linear Codes, have the
   desired property of having all elements (except zero) invertible
   for + and * and all operations over any elements do not result in
   an overflow or underflow. Examples of Finite Fields are prime
   fields $\{0..p^m-1\}$, where p is prime. Most used fields use p=2 and
   are called binary extension fields $\{0..2^m-1\}$, where m often
   equals 1, 4 or 8 for practical reasons.

o Finite Field size: The number of elements in a finite field. For
   example the binary extension field $\{0..2^m-1\}$ has size $q=2^m$.

o Block Coding: Coding technique where the input Flow(s) must be
   first segmented into a sequence of blocks, FEC encoding and
   decoding being performed independently on a per-block basis.

o Sliding Window Coding or Convolutional Coding: General class of
   coding techniques that rely on a sliding encoding window. This is
   an alternative solution to Block Coding.

o Fixed or Elastic Sliding Window Coding: Coding technique that
   generates repair data on-the-fly, from the set of source data
   present in the sliding encoding window at that time, usually by
   using Linear Coding. The sliding window may be either of fixed
   size or of variable size over the time (also known as "elastic
   sliding window").

o Feedback: Feedback information sent by a decoding node to a node
   (or from a consumer to a publisher in case of End-to-End Coding).
   The nature of information contained in a feedback packet varies,
   depending on the use-case. It can provide reception and/or
   decoding statistics, or the list of available source packets
   received or decoded, or the list of lost source packets that
   should be retransmitted, or a number of additional repair packet
   needed to have a full rank linear system.

Concerning CCN/NDN, the following terminology and definitions are
used.

o Consumer: A node requesting content. It initiates communication
   by sending an interest packets.

o Publisher: A node providing content. It originally creates or
   owns the content.

o Forwarding Information Base (FIB): A lookup table in a content
   router containing the name prefix and corresponding destination
   interface to forward the interest packets.

o Pending Interest Table (PIT): A lookup table populated by the
   interest packets containing the name prefix of the requested data,
   and the outgoing interface used to forward the received data
   packets.

o Content Store (CS): A storage space for a router to cache content
   objects. It is also known as in-network cache.

o Content Object: A unit of content data delivered through the CCN/
   NDN network.

o Content Flow: A sequence of content objects associated with the
   unique content name prefix.

2.2. NDN/CCN Background

Armed with the terminology above, we briefly explain the key concepts
of CCN/NDN. Both protocols are similar in principle, and different
on some implementation choices.

In a CCN network, there are two types of packets at the network
level: interest and data. The consumer request a content by sending
an "interest" message, that carries the name of the data. On
difference to note here in CCN and NDN is that in later versions of
CCN, the interest must carry a full name, while in NDN it may carry a
name prefix (and receive in return any data with a name matching this
prefix).

Once a router receives an "interest" message, it performs a series of
look-up: first it checks in the Content Store if it has a copy of the
requested content available. If it does, it returns the data and the
transaction has successfully completed.

If it does not, it performs a look-up of the PIT to see if there is
already an outgoing request for the same data. If there is not, then

it creates an entry in the PIT that lists the name included in the
interest, and the interfaces from which it received the interest.
This is used later to send the data back, since interest packets do
not carry a source field that identifies the requester. If there is
already a PIT entry for this name, then it is updated with the
incoming interface of this new request and the interest is discarded.

After the PIT look-up, the interest undergoes a FIB lookup to select
an outgoing interface. The FIB lists name prefixes and their
corresponding forwarding interfaces, to send the interface towards a
router that possesses a copy of the requested data.

Once a copy of the data is retrieved, it is send back to the
requester(s) using the trail of PIT entries; intermediate node remove
the PIT state every time that an interest is satisfied, and may store
the data in their content store.

Data packets carry some information to validate the data, in
particular that the data is indeed the one that corresponds to the
name. This is required since authentication of the object is crucial
in CCN/NDN. However, this step is optional at intermediate routers,
so as to speed up the processing.

The key aspect of CCN/NDN is that the consumer of the content does
not establish a session with a specific server. Indeed, the node
that returns the content is not aware of the network location of the
requester and the requester is not aware of the network location of
the node that provides the content. This in theory allows the
interests to follow different paths within a network, or even to be
sent over totally different networks.

3. Advantage given by NC and CCN/NDN

Both NC for large scale content dissemination [7] and CCN/NDN can
contribute to effective content/information delivery while working
jointly. They both bring similar benefits such as throughput/
capacity gain and robustness enhancement. The difference between
their approaches is that, the former considers content flow as
algebraic information to combine [17], while the latter focuses on
content/information itself at the networking layer. Because these
approaches are complementary, it is natural to combine them. The
CCN/NDN core abstraction at networking layer through name makes
network stack simple as it enables applications to take maximum
advantage of multiple simultaneous connectivities due to its simpler
relationship with the layer 2 [15].

CCN/NDN itself, however, cannot provide reliable and robust content
dissemination. This requires some specific CCN/NDN transport (i.e.,

strategy layer) [15]. NC can enable the CCN/NDN transport system to
effectively distribute and cache data associated with multi-path data
retrieval. Furthermore, NC may further enhance CCN/NDN security
[23]. In this context, it should be natural that there is much room
for considering NC integration into CCN/NDN transport exploiting in-
network caching and multi-path transmission [9] and seamless mobility
[11] [28].

From the perspective of NC transport mechanism, NC is divided into
two major categories: one is coherent NC, and the other is non-
coherent NC [30]. In coherent NC, source and destination nodes
exactly know network topology and coding operations at intermediate
nodes. When multiple consumers are trying to receive the same
content such as live video streaming, coherent NC could enable the
optimal throughput by making the content flow sent over the
constructed optimal multicast trees [24].

However, it requires fully adjustable and specific name-based routing
mechanism for CCN/NDN, and an intense computational task for central
coordination. In the case of non-coherent NC that often utilizes
RLC, they do not need to know network topology and intermediate
coding operations. Since non-coherent NC works in a completely
independent and decentralized manner, this approach is more feasible
especially in the large scale use cases that are intended with CCN/
NDN. This document thus focuses on non-coherent NC with RLC.

4. Requirements

This section presents the NC requirements for ICN/CCN in terms of
network architecture and protocol. The current document focuses on
NC in a block coding manner.

4.1. Content Naming

Naming content objects is as important for CCN/NDN as naming hosts is
for today's Internet [19]. Before performing network coding for
specified content in CCN/NDN, the overall content should be split
into small content objects to avoid packet fragmentation that could
cause unnecessary packet processing and degrades throughput. The
size of content objects should be within the allowable packet size so
as to avoid packet fragmentation in CCN/NDN network, and then network
coding should be applied into a set of the content objects.

Each coded packet MAY have a unique name as the original content
object has in CCN/NDN, since PIT/FIB/CS operations need a unique name
to identify the coded data. As a way of naming coded packet, the
encoding vector and the identifier of generation can be used as a
part of the content object name [10]. For instance, when the block

size (also called generation size) is k and the encoding vector is
[1,0,0,0], the name would be like /CCN.com/video-A/k/1000. This
naming scheme is simple and can support the delivery of coded packets
with exactly the same operations in the FIB/PIT/CS as for original
source packets. However, such a naming way requires the consumer to
know the naming structure (through a specific name resolution scheme
for instance) in order for nodes to specify the exact name of
generated coded data packet to retrieve it. From this point of view,
it could shift the generation of the encoding vector from the content
producer onto the content requester.

If a naming schema such as above is used, it would be valuable to
reconsider whether Interest should carry full names (as in CCN) or
prefixes (as in NDN) as multiple network coded packets could match a
response to a specific prefix for a given generation, such as
/CCN.com/video-A/k. In the latter case allowing partial name
matching, the content requestor may not be able to obtain degrees of
freedom. Thus, extensions in the TLV header of the Interest would be
used to specify further network coding information so as to limit
coded packets to be received (for instance, by specifying the encoded
vectors the content requestor receives (also called decoding matrix)
as in [9]). However, it may incur a largely increased size of TLV
header. Without such coding information, the forwarding node would
need to maintain some records regarding interest packets sent before,
in order to provide new degrees of freedom.

Coded packet MAY have a name that indicates that it is a coded
packet, and move the coding information into a metadata field in the
payload (i.e., the name includes only data type, original or coded
packet, etc). This however would preclude network coding on packets
without prior decoding them (for instance, in the CS of forwarding
nodes). It would not be beneficial for applications or services that
may not need to understand the packet payload. Due to the
possibility that multiple coded packets may have a same name, as
described above, some mechanism needs for the content requestor to
obtain innovative coded packets. It would also require some
mechanism to insert the multiple innovative packets into the CS. If
the coding information of coded packet are encrypted together with
the payload (for instance, at source coding), the content requestor
or forwarding nodes would incur extra computational overhead for
decryption of the packet to interpret the coding information.

4.2. Transport

The pull-based request-response feature of CCN/NDN is the fundamental
principle of its transport layer; one Interest retrieves at most one
Data packet. It is important to not violate this rule, as it would

open denial of service attacks issues, and thus the following basic
operation should be considered to apply NC to CCN/NDN.

4.2.1. Scope of Network Coding

It should be discussed whether the network can update data packets
that are being received in transit, or if only the data that matches
an interest can be subject to network coding operations. In the
latter case, the network coding is performed on an end-to-end basis
(where one end is the consumer, and the other end is any node that is
able to respond to the Interest). In the former case, NC happens
anywhere in the network that is able to update the data. As CCN/NDN
has mechanisms in place to ensure the integrity of the data during
transfer, NC in the network introduce complexities that would require
special consideration for the integrity mechanisms to still work.

Similarly, caching of network coded packets at intermediate node may
be valuable, but may prevent the node caching the coded content to
validate the content.

4.2.2. Consumer Operation

To attain NC benefits associated with in-network caching, consumers
need to issue interests directing the router (or publisher) to
forward innovative coded packets if available. The reason why this
directive is needed is that delay-sensitive applications such as
live-video streaming may want to sequentially get original packets
rather than coded packets cached in routers due to real-time
constraint. Issuing such an interest is possible by using optional
TLV (Type Length Value) header contained in Interest TLV packet
format which allows network elements to add or modify information on
the fly. Consumer can put an instruction into it, and for instance,
if routers detect that it is better for consumer to get coded packets
rather than original packets, routers can modify it to do so. After
receiving interests having the instruction in optional header, the
router with useful coded packets forward them.

As another solution, consumer issues interests specifying unique
names for each coded packets. In this case, a unified naming scheme
considering both original and coded packets is required. Moreover,
in the case of NC end-to-end approach, publishers need to get
feedback from the corresponding receivers to adjust some coding
parameters. To deal with this, a receiver may have to request a
specific interest name to reach the corresponding publisher and put
required information into the optional header.

4.2.3. Router Operation

   Routers need to appropriately handle PIT entries to accommodate
   interests for coded packets as well as original packets. Moreover,
   in order to decode as necessary, nodes need to know the coding vector
   used for each coded packet (note: since all the data for a specific
   content may not come through the same path/network, intermediate
   nodes may never be able to decode). In a typical case, the coding
   vector used for each coded packet is attached to the header of coded
   data. In regard to this point, the generation size (also called
   block size) for NC should be set to a reasonable value so that the
   total coded packet size including header needed for expressing the
   coding vector information and data message fits into the allowable
   packet size. It may be useful to use compression techniques for
   coding vectors [20][21].

   Router may try to forward useful independent coded packets toward
   downstream nodes in order to respond to received interests for coded
   packets. Routers thus need to determine whether or not they can
   generate useful coded packets for consumers. Assuming that the size
   of the Finite Field in use is not relatively small, re-encoding using
   enough cached packets has a strong probability of making independent
   coded packets [24]. If router does not have enough cached packets to
   newly produce independent coded packets, it relays received interests
   to upstream nodes to receive a new original or independent coded
   packet and pass it to downstream nodes. In another possible case,
   when receiving interests for only original packets, routers may try
   to decode and get all the original packets and store them (if there
   are fully available cache capacity), enabling faster response to the
   interests. Since there is a tradeoff between NC encoding/decoding
   calculation cost and cache capacity, and the usage efficacy of re-
   encoding or decoding at router, router should need to determine how
   to response to receiving interests according to the use case (e.g.,
   delay-sensitive or delay-tolerant application) and the router
   situation such as available cache space and computational capability.

   Some proposed schemes [10]require that the router maintain a tally of
   the interests for a specific name and generation, so as to know how
   many degrees of freedom have been provided already for the NC
   packets. Scalability and practicality of maintaining such scheme at
   intermediate routers should considered.

   To enable fast loss recovery cooperating with in-network caching, a
   transport mechanism of in-network loss detection and recovery
   [28][14] at router as well as consumer-driven mechanism should be
   considered.

4.2.4. **Publisher Operation**

   The procedure for splitting an overall content into small content
   objects is responsible for the original publisher. When applying NC
   for the content, the publisher performs NC over the content objects,
   and naming processing for the coded packets. If the producer takes
   the lead in determining the used encoding vectors and generating the
   coded packets, there are the two possible end-to-end cases; 1)
   content requestors obtain the names of coded packets through a
   certain mechanism, and send the correspond interests toward the
   publisher to get the coded packets already generated at the
   publisher, and 2) the publisher determines the encoding vectors after
   receiving interests specifying them. In the former case, although
   content requestors cannot flexibly specify an encoding vector for
   generating the coded packet to retain, but the latency for getting
   the coded data can be reduced compared to the latter case where
   additional NC operations need after receiving interests. According
   to application requirement for latency, such NC operation strategy
   should be considered.

4.3. **In-network Caching**

   Caching is an essential technique to improve throughput and latency
   in various applications. In-network caching CCN/NDN essentially
   supports at network level is highly beneficial by exploiting NC to
   enable effective multicast transmission [29], multipath data
   retrieval[10] [11], fast loss recovery [14], and so on. However,
   there are several issues to be considered.

   As a general issue, there are limitations of cache capacity, and
   caching policy affects on consumer's performances [22] [25] [26]. It
   is thus highly significant for routers to determine which packets
   should be cached and discarded. Since delay-sensitive applications
   often do not require in-network cache for a long period due to their
   real-time constraints, routers have to know the necessity for caching
   received packets to save the caching volume. This could be possible
   by putting a flag into optional header of data packets at publisher
   side. When receiving data packets with the flag meaning no necessity
   for cache, routers just have to forward them to downstream nodes. On
   the other hand, when receiving original packets or coded packets
   without the flag, router may cache them based on a specified
   replacement policy.

   One key aspect of in-network caching is whether or not intermediate
   nodes can cache NC packets without first decoding them. If in-
   network caches store coded packets, they need to be able to validate
   that the packets are not compromised, so as to avoid cache pollution
   attacks. Without having all the packets in a generation, the cache

cannot decode the packets to check if it is authenticated. Caching
of coded packets would require some mechanism to validate coded
packets. In addition, when coded packets have a same name, it would
also require some mechanism to identify them.

4.4. Seamless Mobility

This subsection presents how NC can achieve seamless mobility [11]
[28] and clarify the requirements. A key feature of CCN/NDN is that
it is sessionless and that multiple interests can be send to
different copies of the content in parallel. CCN/NDN enables a
consumer to retrieve the content from multiple sources that are
distributed and asynchronous.

In this context, network coding provide a mechanism to ensure that
the Interests sent to multiple copies of the content retrieve
innovative packets, even in the case of packet losses on some of the
paths/networks to these copies. NC adds a reliability layer to CCN
in a distributed and asynchronous manner. One key benefit is that
the link between the consumer and the multiple copies acts as a
virtual logical link, upon which rate adaptation mechanism can be
performed.

This naturally applies to mobility event, where the consumer may
connect between multiple access points before a mobility event (make-
before-break handoff). In such mobility event, the consumer is
connected first to the previous access point, then to both the
previous and next access points, then finally only to the next access
points. With CCN, the consumer only sends interests on the available
interfaces. Requesting network coded packets ensures that during the
phase where it is connected to the previous and the next APs at the
same time, it does not receive duplicate data, but does not miss on
any content either. By combining NC with CCN, the consumer receives
additional degrees of freedom with any innovative packet it receives
on either interface.

Further discussion is [TBD].

4.5. Security and Privacy

This subsection describes the requirement for security and privacy
provided by NC in CCN/NDN, such as data integrity especially when
intermediate nodes perform re-encoding, as in the case of hash
restrictions for original data packets, and so on.

Network coding impacts the security mechanisms of CCN/NDN. In
particular, CCN/NDN is designed to prevent modification of the Data
packets. Because Data packets for a specific name can be self-

authenticated, they can be validated on the delivery path, and can
also be cached at untrusted intermediate nodes. Network coding may
bring up issues if intermediate nodes are allowed to modify packets
by performing additional network coding operations. Intermediate
nodes may also be caching network coded packets without having the
ability to perform validation of the content and therefore open
themselves to cache pollution attacks.

In CCN/NDN, content objects can be encrypted to support access
control or privacy. If the coding information of coded packet is
included in the encrypted data payload, extra computational overhead
occurs.

5. Challenges

This section presents several primary challenges and research items
to be considered when applying NC into CCN/NDN.

5.1. Adopting Convolutional Coding

Several block coding approaches have been proposed so far, but there
is still no sufficient discussion and application of convolutional
coding approach (e.g., sliding or elastic window coding) in CCN/NDN.
Convolutional coding is often appropriate to situations where a fully
or partially reliable delivery of continuous data flows is needed,
especially when these data flows feature realtime constraints. As in
[31] on an end-to-end basis, it would be advantageous for continuous
content flow to adopt sliding window coding in CCN/NDN. In this
case, the publisher needs to appropriately set coding parameters and
let content requestor know the information, and content requestor
needs to send interest (i.e., feedback information) about the data
reception status. Since CCN/NDN advocates hop-by-hop communication,
it would be worth discussing and investigating how convolutional
coding can be applied in a hop-by-hop fashion and the benefits. In
particular, assuming that NC could occur at intermediate nodes with
some useful data packets stored in the CS as described in the
previous section, both the encoding window and CS management would be
required, and the feasibility and practicality should be considered.

5.2. Rate and Congestion Control

Adding redundancy using coded packets may cause further network
congestion and adversely affect overall throughput performance. In
particular, in a situation where fair bandwidth sharing is more
desirable, each streaming flow must adapt to the network conditions
to fairly consume the available link bandwidth. It is thus
indispensable that each content flow cooperatively implements
congestion control to adjust the consumed bandwidth to stabilize the

network condition (i.e., to achieve low packet loss rate, delay, and
jitter).

5.3. Security and Privacy

A variety of security and privacy concerns would exist in NC and CCN/
NDN. This subsection focuses on the description of security and
privacy challenges related to NC for CCN/NDN. [TBD]

5.4. Routing Scalability

This subsection focuses on the challenges of routing mechanisms such
as scalability and protocol overhead, and so on.

6. Security Considerations

This document does not impact the security of the Internet. Security
considerations related to NC for CCN/NDN are described in the
previous Section.

7. References

7.1. Normative References

[1]   Bradner, S., "Key words for use in RFCs to Indicate
           Requirement Levels", BCP 14, RFC 2119,
           DOI 10.17487/RFC2119, March 1997,
           <https://www.rfc-editor.org/info/rfc2119>.

7.2. Informative References

[2]   Cai, N. and R. Yeung, "Secure network coding", Proc.
           International Symposium on Information Theory
           (ISIT), IEEE, June 2002.

[3]   Lima, L., Gheorghiu, S., Barros, J., Mdard, M., and A.
           Toledo, "Secure Network Coding for Multi-Resolution
           Wireless Video Streaming", IEEE Journal of Selected Area
           (JSAC), vol. 28, no. 3, April 2002.

[4]   Gkantsidis, C. and P. Rodriguez, "Cooperative Security for
           Network Coding File Distribution", Proc. Infocom, IEEE,
           April 2006.

[5]   Vilea, J., Lima, L., and J. Barros, "Lightweight security
           for network coding", Proc. ICC, IEEE, May 2008.

[6]   Dimarkis, A., Godfrey, P., Wu, Y., Wainwright, M., and K.
           Ramchandran, "Network Coding for Distributed Storage
           Systems", IEEE Trans. Information Theory, vol. 56, no.9,
           September 2010.

[7]   Gkantsidis, C. and P. Rodriguez, "Network coding for large
           scale content distribution", Proc. Infocom, IEEE, March
           2005.

[8]   Seferoglu, H. and A. Markopoulou, "Opportunistic Network
           Coding for Video Streaming over Wireless", Proc. Packet
           Video Workshop (PV), IEEE, November 2007.

[9]   Montpetit, M., Westphal, C., and D. Trossen, "Network
           Coding Meets Information-Centric Networking: An
           Architectural Case for Information Dispersion Through
           Native Network Coding", Proc. Workshop on Emerging Name-
           Oriented Mobile Networking Design (NoM), ACM, June 2012.

[10]   Saltarin, J., Bourtsoulatze, E., Thomos, N., and T. Braun,
           "NetCodCCN: a network coding approach for content-centric
           networks", Proc. Infocom, IEEE, April 2016.

[11]   Ramakrishnan, A., Westphal, C., and J. Saltarin, "Adaptive
           Video Streaming over CCN with Network Coding for Seamless
           Mobility", Proc. International Symposium on Multimedia
           (ISM), IEEE, December 2016.

[12]   Wang, J., Ren, J., Lu, K., Wang, J., Liu, S., and C.
           Westphal, "An Optimal Cache Management Framework for
           Information-Centric Networks with Network Coding", Proc.
           Networking Conference, IFIP/IEEE, June 2014.

[13]   Wang, J., Ren, J., Lu, K., Wang, J., Liu, S., and C.
           Westphal, "A Minimum Cost Cache Management Framework for
           Information-Centric Networks with Network Coding",
           Computer Networks, Elsevier, August 2016.

[14]   Matsuzono, K., Asaeda, H., and T. Turletti, "Low Latency
           Low Loss Streaming using In-Network Coding and Caching",
           Proc. Infocom, IEEE, May 2017.

[15]   Jacobson, V., Smetters, D., Thornton, J., Plass, M.,
           Briggs, N., and R. Braynard, "Networking Named Content",
           Proc. CoNEXT, ACM, December 2009.

[16]   Zhang, L., Afanasyev, A., Burke, J., Jacobson, V., Claffy,
            K., Crowley, P., Papadopoulos, C., Wang, L., and B. Zhang,
            "Named data networking", ACM Comput. Commun. Rev., vol.
            44, no. 3, July 2014.

[17]   Koetter, R. and M. Medard, "An Algebraic Approach to
            Network Coding", IEEE/ACM Trans. on Networking, vol. 11,
            no 5, Oct. 2003.

[18]   Adamson, B., Adjih, C., Bilbao, J., Firoiu, V., Fitzek,
            F., Lochin, E., Masucci, A., Montpetit, M., Pedersen, M.,
            Peralta, G., Roca, V., Saxena, P., and S. Sivakumar,
            "Network Coding Taxonomy", draft-irtf-nwcrg-network-
            coding-taxonomy-05 (work in progress), September 2017.

[19]   Kutscher, et al., D., "Information-Centric Networking
            (ICN) Research Challenges", RFC 7927, July 2016.

[20]   Thomos, N. and P. Frossard, "Toward one Symbol Network
            Coding Vectors", IEEE Communications letters, vol. 16, no.
            11, November 2012.

[21]   Lucani, D., Pedersen, M., Heide, J., and F. Fitzek,
            "Fulcrum Network Codes: A Code for Fluid Allocation of
            Complexity", available at http://arxiv.org/abs/1404.6620,
            April 2014.

[22]   Perino, D. and M. Varvello, "A reality check for content
            centric networking", Proc. SIGCOMM Workshop on
            Information-centric networking (ICN'11), ACM, August 2011.

[23]   Wu, Q., Li, Z., Tyson, G., Uhlig, S., Kaafar, M., and G.
            Xie, "Privacy-Aware Multipath Video Caching for Content-
            Centric Networks", IEEE Journal of Selected Area
            (JSAC) vol. 38, no. 8, June 2016.

[24]   Wu, Y., Chou, P., and K. Jain, "A comparison of network
            coding and tree packing", Proc. ISIT, IEEE, June 2004.

[25]   Podlipnig, S. and L. Osz, "A Survey of Web Cache
            Replacement Strategies", Proc. ACM Computing Surveys vol.
            35, no. 4, December 2003.

[26]   Rossini, G. and D. Rossi, "Evaluating CCN multi-path
            interest forwarding strategies", Elsevier Computer
            Communication, vol.36, no. 7, April 2013.

   [27]   Chai, W., He, D., Psaras, I., and G. Pavlou, "Cache Less
             for More in Information-centric Networks", Journal
             Computer Communications, vol. 37. no. 7, April 2013.

   [28]   Carofiglio, G., Muscariello, L., Papalini, M., Rozhnova,
             N., and X. Zeng, "Leveraging ICN In-network Control for
             Loss Detection and Recovery in Wireless Mobile networks",
             Proc. ICN ACM, September 2016.

   [29]   Ali, M. and U. Niesen, "Coding for Caching: Fundamental
             Limits and Practical Challenges", IEEE Communications
             Magazine vol. 54, no. 8, August 2016.

   [30]   Koetter, R. and F. Kschischang, "An algebraic approach to
             network coding", IEEE Trans. Netw. vol.11, no.5, October
             2008.

   [31]   Tournoux, P., Lochin, E., Lacan, J., Bouabdallah, A., and
             V. Roca, "On-the-Fly Erasure Coding for Real-Time Video
             Applications", IEEE Trans. Multimeda vol.13, no.4, August
             2011.

Authors' Addresses

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

   Email: matsuzono@nict.go.jp

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

   Email: asaeda@nict.go.jp

Cedric Westphal
Huawei
2330 Central Expressway
Santa Clara, California 95050
USA

Email: cedric.westphal@huawei.com