Network Coding Research Group
Internet-Draft
Intended status: Informational
Expires: March 23, 2020

K. Matsuzono
H. Asaeda
NICT
C. Westphal
Huawei

September 20, 2019

# Network Coding for Content-Centric Networking / Named Data Networking: Requirements and Challenges

**draft-irtf-nwcrg-nwc-ccn-reqs-02**

## 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 March 23, 2020.

## Copyright Notice

Copyright (c) 2019 IETF Trust and the persons identified as the document authors. All rights reserved.

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

Information-Centric Networks (ICN) in general, and Content-Centric Networking (CCN) [15] or Named Data Networking (NDN) [17] 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].

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 [33]. 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 [26]. However, it requires a fully adjustable and specific routing mechanism, and an intense computational task for central coordination. In the case of non-coherent NC that often utilizes RLC, it is not required to know either network topology nor intermediate coding operations [27]. Since non-coherent NC works in a completely independent and decentralized manner, this approach is more feasible especially in the large scale use cases.

NC mixes multiple packets together with parts of the same content, 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 been mixed 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. NC has already been implemented for information/content dissemination (e.g. [6] [7] [8]). Montpetit, et al., first suggested to exploit NC techniques to enhance key system performances in ICN [9]. NC provides CCN/NDN with the highly beneficial potential to effectively disseminate information in a completely independent and decentralized manner.

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 IRTF Coding for Efficient Network Communications Research Group (NWCRG)[20].

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

- Generation, or (IETF) Block: With Block Codes, the set of Source Symbols of the input Flow(s) that are logically grouped into a Block, before doing encoding.

- Generation Size, Code Dimension, or (IETF) Block Size: With Block Codes, the number of Source Symbols, k, belonging to a Block.

- Coding Vector: A set of Coding Coefficients used to generate a certain Repair Symbol through Linear Coding. The number of nonzero coefficients in the Coding Vector defines its density

- 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.

- 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.

- 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.

- 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.

- 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").

- 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.

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

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

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

- 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.

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

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

- 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 CCN [16], the interest must carry a full name, while in NDN [18] 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 sent back to the requester(s) using the trail of PIT entries; intermediate nodes 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. Advantages provided 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 [19], 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, does not provide reliable and robust content dissemination by default. 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 [9]. Furthermore, NC can contribute to improving both caching performance and cache privacy that CCN/NDN newly poses at the networking layer [25]. Others also have considered NC in CCN/NDN use cases such as content dissemination with in-network caching [10] [12] [13], seamless mobility [11] [31], low-latency video streaming [14], etc. In this context, it should be natural that there is much room for considering NC integration into CCN/NDN.

## 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 [21]. 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 degrade 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/CS operations typically need a unique name to identify the coded data. As a way of naming coded packet, the coding 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 coding 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 PIT/CS as for original source packets. Since such a naming way enables consumer to specify coded packets to receive, it could shift the generation of the coding vector from the content producer onto the content requester (described in Section 4.2.2).

If a naming schema such as above is used, it would be valuable to reconsider whether Interests 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, and thus it may be useful to use compression techniques for coding vectors [22] [23]. Without such coding information, the forwarding node would need to maintain some records regarding interest packets sent before (described in Section 4.2.3).

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). 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, some mechanism is needed for the content requestor to obtain innovative coded packets. As described in Section 4.3, a mechanism to manage the multiple innovative packets in the CS would be required as well. In addition, extra computational overhead would occur when the payload is being encrypted (described in Section 4.5).

## 4.2. Transport

The pull-based request-response feature of CCN/NDN is a fundamental principle of its transport layer; one Interest retrieves at most one Data packet. It is believed that 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. In any case, such security considerations must be addressed if this rule were to be violated.

#### 4.2.1. Scope of Network Coding

It should be discussed whether the network can recode 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 obtain NC benefits associated with in-network caching, consumer needs 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. As described in Section 4.1, because coded packet can have a name explicitly different from original source packets, issuing such an interest is possible.

When issuing interests specifying unique names with k and coding vectors for each coded packets, consumer appropriately receives innovative coded packets if they are available at some nodes and can be forwarded to the consumer. However, consumer needs 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. In the case of NC end-to-end approach, if consumer want to adjust some coding parameters at publisher, some specific scheme would be required.

#### 4.2.3. Router Operation

Routers need to forward linearly independent coded packets toward downstream nodes if incoming interests for coded packets does not specify some coding parameters such as the coding vector to be used. 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 independent packets has a strong probability of making

independent coded packets [26]. However, without enough cached packets, router needs to determine whether or not to an independent coded packet can be forwarded to the interface at which the interest arrived. To deal with this issue, some proposed schemes [10] require that the router maintains a tally of the interests for a specific name, generation and the corresponding interface, 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 be considered. In addition, some transport mechanism of in-network loss detection and recovery [31] [14] at router as well as consumer-driven mechanism could be indispensable in order to enable fast loss recovery and enhance NC gains. After determining that independent coded packet cannot be provided, according to the FIB, the router relays received interests to upstream nodes to receive a new original or independent coded packet. In this context, to effectively and quickly retrieve independent coded data, appropriately setting the FIB and efficient interest forwarding strategies should be also considered.

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 re-encoding or decoding leads to extra computational overhead, routers 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.

#### 4.2.4. Publisher Operation

The procedure for splitting an overall content into small content objects (described in Section 4.1) is the responsibility of 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 coding 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 coding vectors after receiving interests specifying them. In the former case, although content requestors cannot flexibly specify an coding 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. The common benefit in such end-to-end cases is that if the publisher adds signature on the coded packets, data verification

can be possible throughout. According to application requirement for latency, such NC operation strategy should be considered.

#### 4.2.5. Backward Compatibility

Network Coding operations should be applied in addition to the regular network behavior. As such, nodes should be able to not support network coding (either in forwarding the packets, but also in the caching mechanism). Network Coding operations should function alongside regular network operations. A network coding framework should be compatible with a regular framework, so as to allow backward compatibility and smooth migration from one framework to the other.

### 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 [32], 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 [24] [28] [29]. 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. They may be caching the coded packets without having the ability to perform validation of the content (described in Section 4.5). Therefore, 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] [31] and clarify the requirements. A key feature of CCN/NDN is that it is sessionless and that multiple interests can be sent to different copies of the content in parallel. CCN/NDN enables a consumer to retrieve the content from multiple copies that are distributed and asynchronous. The key benefit is that the link between the consumer and the multiple copies acts as a virtual logical link, upon which rate adaptation mechanism (say, for video streaming) can be performed.

In this context, NC adds a reliability layer network to CCN in a distributed and asynchronous manner, because NC provides a mechanism to ensure that the Interests sent to multiple copies of the content in parallel retrieve innovative packets, even in the case of packet losses on some of the paths/networks to these copies. This naturally applies to mobility events, 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. By combining NC with CCN, requesting 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. The consumer receives additional degrees of freedom with any innovative packet it receives on either interface. From this point of view, an effective interest forwarding strategy for obtaining innovative packets should be considered for consumer to achieve seamless mobility.

### 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. In addition, 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.

In CCN/NDN, content objects can be encrypted to support access control or privacy. 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. With consideration for low computation overhead, some mechanism supporting both NC and access control/privacy should be considered.

## 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 [34] 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). From this point of view, a router supported approach would be effective, but an effective deployment scenario is needed.

As described in Section 4.4, NC can contribute to seamless mobility by obtaining innovative packets without receiving duplicated packets through a virtual logical link to multiple copies of the content. To achieve seamless mobility while improving overall throughput or latency, an effective rate adaptation mechanism upon the virtual logical link is also challenging.

### 5.3. Security and Privacy

CCN/NDN introduces new security and privacy issues at the networking layer different from IP network, such as cache poisoning and pollution attack, DoS attack using interest packets, and so on.

NC could be utilized to mitigate some security or privacy issues CCN/NDN introduces. For instance, assuming that consumers can utilize multipath data retrieval and caching in CCN/NDN with NC, cache privacy and anonymity set for consumers can be improved as well as caching performance due to the diversity of caching content along different paths.

On the other hand, considering NC operations over CCN/NDN, the issues related to in-network caching add additional complexity. In order to avoid cache poisoning attack which tries to fill routers cache with polluted content, router needs to check whether or not the content is validated. However, in the case of performing NC and generating a new coded data at routers, a validation mechanism to accurately verify coded data as quickly as possible should be considered while maintaining in-network cache benefits (lower latency and network resource saving). If router can cache some valid coded data, it needs to put a great deal of thought into the effectiveness with respect to cache pollution attack, since coded data newly generated may be unpopular. Moreover, Denial of Service (DoS) attacks may target either the routers or the publishers performing NC to pose unnecessary coded data, impose higher NC computation load, and increase the number of PIT entries, which requires some careful considerations to avoid them.

NC also offers a new surface of attack; for instance, if the coding vector is exposed at the network layer, it would have to be protected (and validated) so as to avoid modifications by an attacker (and allow for verification) on the path of the packet.

In this context, from the perspective of both feasibility and practicability, a more effective approach with consideration for

security and privacy would be needed in order to accelerate the deployment of CCN/NDN with NC.

### 5.4. Routing Scalability

In CCN/NDN, a name-based routing protocol without a resolution process streamlines the routing process and reduces the overall latency. As in IP routing, the growth in the routing table size has become a concern. This may require a hierarchical naming scheme so as to improve the routing scalability by enabling aggregation of routing information. Moreover, it is a challenge that content requestors efficiently obtain linearly independent coded packets using multipath retrieval in a fully distributed manner, in order to fully leverage NC over CCN/NDN to improve throughput or reduce latency. This would require some efficient routing mechanism to appropriately set the FIB and also requires some efficient interest forwarding strategy. Such routing coordination may create routing scalability issues. From another NC perspective, as described Section 4.2.2, when issuing interests specifying unique names for each coded packet, consumers need in advance to know how to specify the names of the coded data through some specific name resolution scheme, and routers may need to appropriately set the FIBs. In this context, it would be challenging to achieve effective and scalable routing for interests requesting coded data as well as to simplify the routing process.

## 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] Mosko, M. and et al., "Content-Centric Networking (CCNx) Messages in TLV Format", RFC 8609, July 2019, <https://tools.ietf.org/html/rfc8609>.

[17] 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.

[18] NDN Packet Format, "NDN Packet Format Specification 0.3 documentation", Sept. 2019, <https://named-data.net/doc/NDN-packet-spec/current/>.

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

[20] Adamson, B. and et al., "Taxonomy of Coding Techniques for Efficient Network Communications", RFC 8406, June 2018, <https://tools.ietf.org/html/rfc8406>.

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

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

[23] 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.

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

[25] 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.

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

[27] Ho, T., Medard, M., Koetter, R., Karger, R., Effros, D., Shi, M., and B. Leong, "A Random Linear Network Coding Approach to Multicast", IEEE Trans. Information Theory, vol. 52, no.10, October 2006.

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

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

[30] 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.

[31] 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.

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

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

[34] 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