Stream: Internet Research Task Force (IRTF)
RFC: 9273
Category: Informational
Published: August 2022
ISSN: 2070-1721

Authors:
K. Matsuzono *NICT*
H. Asaeda *NICT*
C. Westphal *Huawei*

RFC 9273
Network Coding for Content-Centric Networking / Named Data Networking: Considerations and Challenges

Abstract

This document describes the current research outcomes in Network Coding (NC) for Content-Centric Networking (CCNx) / Named Data Networking (NDN) and clarifies the technical considerations and potential challenges for applying NC in CCNx/NDN. This document is the product of the Coding for Efficient Network Communications Research Group (NWCRG) and the Information-Centric Networking Research Group (ICNRG).

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

Information-Centric Networking (ICN), in general, and Content-Centric Networking (CCNx) [1] or Named Data Networking (NDN) [2], in particular, have emerged as a novel communication paradigm that advocates for the retrieval of data based on 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 CCNx/NDN architecture has introduced innovative ideas and has stimulated research in a variety of areas, such as in-network caching, name-based routing, multipath transport, and content security. One key benefit of requesting content by name is that it eliminates the requirement to establish a session between the client and a specific server, and the content can thereby be retrieved from multiple sources.

In parallel, there has been a growing interest in both academia and industry for better understanding the 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, and redundant transmission on broadcast or point-to-multipoint connections. NC is a technique that is typically used for encoding packets to recover from lost source packets at the receiver and for effectively obtaining the desired information in a fully distributed manner. In addition, in terms of security aspects, NC can be managed using a practical security scheme that deals with pollution attacks [3] and can be utilized for preventing eavesdroppers from obtaining meaningful information [4] or protecting a wireless video stream while retaining the NC benefits [5] [6].

From the perspective of the NC transport mechanism, NC can be divided into two major categories: coherent NC and noncoherent NC [7] [8]. In coherent NC, the source and destination nodes know the exact network topology and the coding operations available at intermediate nodes. When multiple consumers are attempting to receive the same content, such as live video streaming, coherent NC could enable optimal throughput by sending the content flow over the constructed optimal multicast trees [9]. However, it requires a fully adjustable and specific routing mechanism and a large computational capacity for central coordination. In the case of noncoherent NC, which often uses Random Linear Coding (RLC), it is not necessary to know the network topology nor the intermediate coding operations [10]. As noncoherent NC works in a completely independent and decentralized manner, this approach is more feasible in terms of eliminating such a central coordination.

NC combines multiple packets together with parts of the same content and may do this at the source and/or at other nodes in the network. Network coded packets are not associated with a specific server, as they may have been combined within the network. As NC is focused on what information should be encoded in a network packet instead of the specific host at which it has been generated, it is in line with the architecture of the CCNx/NDN core networking layer. NC allows for recovery of missing packets by encoding the information across several packets. ICN is synergistic with NC, as it allows for caching of data packets throughout the network. In a typical network that uses NC, the sender must transmit enough packets to retrieve the original data. ICN offers an opportunity to retrieve network-coded packets directly from caches in the network, making the combination of ICN and NC particularly effective. In fact, NC has already been implemented for information/content dissemination [11] [12] [13]. Montpetit et al. first suggested that NC techniques be exploited to enhance key aspects of system performance in ICN [14]. Although CCNx/NDN excels to exploit the benefits of NC (as described in Section 5), some technical considerations are needed to combine NC and CCNx/NDN owing to the unique features of CCNx/NDN (as described in Section 6).

In this document, we consider how NC can be applied to the CCNx/NDN architecture and describe the technical considerations and potential challenges for making CCNx/NDN-based communications better using the NC technology. It should be noted that the presentation of specific solutions (e.g., NC optimization methods) for enhancing CCNx/NDN performance metrics by exploiting NC is outside the scope of this document.

This document represents the collaborative work and consensus of the Coding for Efficient Network Communications Research Group (NWCRG) and the Information-Centric Networking Research Group (ICNRG). This document was read and reviewed by all the active research group members. It is not an IETF product and is not a standard.

2. Terminology

2.1. Definitions Related to NC

This section provides the terms related to NC used in this document, which are defined in RFC 8406 [15] and produced by the NWCRG.

Source Packet:
A packet originating from the source that contributes to one or more source symbols. The source symbol is a unit of data originating from the source that is used as input to encoding operations. For instance, a Real-time Transport Protocol (RTP) packet as a whole can constitute a source symbol. In other situations (e.g., to address variable size packets), a single RTP packet may contribute to various source symbols.

Repair Packet:
A packet containing one or more coded symbols (also called repair symbol). The coded symbol or repair symbol is a unit of data that is the result of a coding operation, applied either to source symbols or (in case of recoding) source and/or coded symbols. When there is a single repair symbol per repair packet, a repair symbol corresponds to a repair packet.

Encoding versus Recoding versus Decoding:
: Encoding is an operation that takes source symbols as input and produces encoding symbols (source or coded symbols) as output. Recoding is an operation that takes encoding symbols as input and produces encoding symbols as output. Decoding is an operation that takes encoding symbols as input and produces source symbols as output.

The terms regarding coding types are defined as follows:

Random Linear Coding (RLC):
: A particular form of linear coding using a set of random coding coefficients. Linear coding performs a linear combination of a set of input symbols (i.e., source and/or coded symbols) using a given set of coefficients and results in a repair symbol.

Block Coding:
: A coding technique wherein the input flow(s) must be first segmented into a sequence of blocks. Encoding and decoding are performed independently on a per-block basis.

Sliding Window Coding or Convolutional Coding:
: A general class of coding techniques that rely on a sliding encoding window. An encoding window is a set of source (and coded in the case of recoding) symbols used as input to the coding operations. The set of symbols change over time, as the encoding window slides over the input flow(s). This is an alternative solution to block coding.

Fixed or Elastic Sliding Window Coding:
: A coding technique that generates coded symbol(s) on the fly, from the set of source symbols 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 time (also known as "Elastic Sliding Window"). For instance, the size may depend on acknowledgments sent by the receiver(s) for a particular source symbol or source packet (received, decoded, or decodable).

The terms regarding low-level coding aspects are defined as follows:

Rank of the Linear System or Degrees of Freedom:
: At a receiver, the number of linearly independent equations of the linear system. It is also known as "Degrees of Freedom". The system may be of "full rank", wherein decoding is possible, or "partial rank", wherein only partial decoding is possible.

Generation or 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 or Block Size:
: With block codes, the number of source symbols belonging to a block. It is equivalent to the number of source packets when there is a single source symbol per source packet.

Coding Coefficient:
With linear coding, this is a coefficient in a certain finite field. This coefficient may be chosen in different ways: for instance, randomly, in a predefined table or using a predefined algorithm plus a seed.

Coding Vector:
A set of coding coefficients used to generate a certain coded symbol through linear coding.

Finite Field:
Finite fields, used in linear codes, have the desired property of having all elements (except zero) invertible for + and *, and no operation over any elements can 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.

2.2. Definitions Related to CCNx/NDN

The terminology regarding CCNx/NDN used in this document is defined in RFC 8793 [16], which was produced by the ICNRG. They are consistent with the relevant documents ([17] [18]).

3. CCNx/NDN Basics

We briefly explain the key concepts of CCNx/NDN. In a CCNx/NDN network, there are two types of packets at the network level: interest and data packet (defined in Section 3.4 of [16]). The term "content object", which means a unit of content data, is an alias to data packet [16]. The ICN consumer (defined in Section 3.2 of [16]) requests a content object by sending an interest that carries the name of the data.

Once an ICN forwarder (defined in Section 3.2 of [16]) receives an interest, it performs a series of lookups. First, it checks if it has a copy of the requested content object available in the cache storage, called Content Store (CS) (defined in Section 3.3 of [16]). If it does, it returns the data, and the transaction is considered to have been successfully completed.

If it does not have a copy of the requested content object in the CS, it performs a lookup of the Pending Interest Table (PIT) (defined in Section 3.3 of [16]) to check if there is already an outgoing interest for the same content object. If there is no such interest, 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 later used to send the content object back, as interest packets do not carry a source field that identifies the consumer. If there is already a PIT entry for this name, it is updated with the incoming interface of this new interest, and the interest is discarded.

After the PIT lookup, the interest undergoes a Forwarding Information Base (FIB) (defined in Section 3.3 of [16]) lookup for selecting an outgoing interface. The FIB lists name prefixes and their corresponding forwarding interfaces in order to send the interest toward a forwarder that possesses a copy of the requested data.

Once a copy of the data is retrieved, it is sent back to the consumer(s) using the trail of PIT entries; forwarders remove the PIT state every time that an interest is satisfied and may store the data in their CS.

Data packets carry some information for verifying data integrity and origin authentication and, in particular, that the data is indeed that which corresponds to the name [19]. This is necessary because authentication of the object is crucial in CCNx/NDN. However, this step is optional at forwarders in order to speed up the processing.

The key aspect of CCNx/NDN is that the consumer of the content does not establish a session with a specific server. Indeed, the forwarder or producer (defined in Section 3.2 of [16]) that returns the content object is not aware of the network location of the consumer, and the consumer 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 completely different networks.

4. NC Basics

While the forwarding node simply relays received data packets in conventional IP communication networks, NC allows the node to combine some data packets that are already received into one or several output packets to be sent. In this section, we simply describe the basic operations of NC. Herein, we focus on RLC in a block coding manner that is well known as a major coding technique.

For simplicity, let us consider an example case of end-to-end coding wherein a producer and consumer respectively perform encoding and decoding for a content object. This end-to-end coding is regarded as a special case of NC. The producer splits the content into several blocks called generations. Encoding and decoding are performed independently on a per-block (per-generation) basis. Let us assume that each generation consists of K original source packets of the same size. When the packets do not have the same size, zero padding is added. In order to generate one repair packet within a certain generation, the producer linearly combines K of the original source packets, where additions and multiplications are performed using a coding vector consisting of K coding coefficients that are randomly selected in a certain finite field. The producer may respond to interests to send the corresponding source packets and repair packets in the content flow (called systematic coding), where the repair packets are typically used for recovering lost source packets.

Repair packets can also be used for performing encoding. If the forwarding nodes know each coding vector and generation identifier (hereinafter referred to as generation ID) of the received repair packets, they may perform an encoding operation (called recoding), which is the most distinctive feature of NC compared to other coding techniques.

At the consumer, decoding is performed by solving a set of linear equations that are represented by the coding vectors of the received source and repair packets (possibly only repair packets) within a certain generation. In order to obtain all the source packets, the consumer requires K linearly independent equations. In other words, the consumer must receive at least K linearly independent data packets (called innovative packets). As receiving a linearly dependent data packet is not useful for decoding, recoding should generate and provide innovative packets. One of the major benefits of RLC is that, even for a small-sized finite field (e.g., $q=2^8$), the probability of generating linearly dependent packets is negligible [9].

5. Advantages of NC and CCNx/NDN

Combining NC and CCNx/NDN can contribute to effective large-scale content/information dissemination. They individually provide 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 that is to be combined [7], while the latter focuses on the content/information itself at the networking layer. Because these approaches are complementary and their combination would be advantageous, it is natural to combine them.

The name-based communication in CCNx/NDN enables consumers to obtain requested content objects without establishing and maintaining end-to-end communication channels between nodes. This feature facilitates the exploitation of the in-network cache and multipath/multisource retrieval and also supports consumer mobility without the need for updating the location information/identifier during handover [1]. Furthermore, the name-based communication intrinsically supports multicast communication because identical interests are aggregated at the forwarders.

NC can enable the CCNx/NDN transport system to effectively distribute and cache the data associated with multipath data retrieval [14]. Exploiting multipath data retrieval and in-network caching with NC contributes to not only improving the cache hit rate but also expanding the anonymity set of each consumer (the set of potential routers that can serve a given consumer) [20]. The expansion makes it difficult for adversaries to infer the content consumed by others and thus contributes to improving cache privacy. Others also have introduced some use cases of the application of NC in CCNx/NDN, such as the cases of content dissemination with in-network caching [21] [22] [23], seamless consumer mobility [24] [25], and low-latency low-loss video streaming [26]. In this context, it is well worth considering NC integration in CCNx/NDN.

6. Technical Considerations

This section presents the considerations for CCNx/NDN with NC in terms of network architecture and protocol. This document focuses on NC when employed in a block coding manner.

6.1. Content Naming

Naming content objects is as important for CCNx/NDN as naming hosts is in the current-day Internet [19]. In this section, two possible naming schemes are presented.

6.1.1. Unique Naming for NC Packets

Each source and repair packet (hereinafter referred to as NC packet) may have a unique name, as each original content object has in CCNx/NDN and as PIT and CS operations typically require a unique name for identifying the NC packet. As a method of naming an NC packet that takes into account the feature of block coding, the coding vector and the generation ID can be used as a part of the content object name. As in [21], when the generation ID is "g-id", generation size is 4, and coding vector is (1,0,0,0), the name could be /CCNx.com/video-A/g-id/1000. Some other identifiers and/or parameters related to the encoding scheme can also be used as name components. For instance, the encoding ID specifying the coding scheme may be used with "enc-id", such as /CCNx.com/video-A/enc-id/g-id/1000, as defined in the FEC Framework (FECFRAME) [27]. This naming scheme is simple and can support the delivery of NC packets with exactly the same operations in the PIT/CS as those for the content objects.

If a content-naming schema, such as the one presented above, is used, an interest requesting an NC packet may have the full name including a generation ID and coding vector (/CCNx.com/video-A/g-id/1000) or only the name prefix including only a generation ID (/CCNx.com/video-A/g-id). In the former case, exact name matching to the PIT is simply performed at data forwarders (as in CCNx/NDN). The consumer is able to specify and retrieve an innovative packet necessary for the consumer to decode successfully. This could shift the generation of the coding vector from the data forwarder onto the consumer.

In the latter case, partial name matching is required at the data forwarders. As the interest with only the prefix name matches any NC packet with the same prefix, the consumer could immediately obtain an NC packet from a nearby CS (in-network cache) without knowing the coding vectors of the cached NC packets in advance. In the case wherein NC packets in transit are modified by in-network recoding performed at forwarders, the consumer could also receive the modified NC packets. However, in contrast to the former case, the consumer may fail to obtain sufficient degrees of freedom (see Section 6.2.3). To address this issue, a new TLV type in an interest message may be required for specifying further coding information in order to limit the NC packets to be received. For instance, this is enabled by specifying the coding vectors of innovative packets for the consumer (also called decoding matrix) as in [14]. This extension may incur an interest packet of significantly increased size, and it may thus be useful to use compression techniques for coding vectors [28] [29]. Without such coding information provided by the interest, the forwarder would be required to maintain some records regarding the interest packets that were satisfied previously (see Section 6.2.3).

6.1.2. Nonunique Naming for NC Packets

An NC packet may have a name that indicates that it is an NC packet and move the coding information into a metadata field in the payload (i.e., the name includes the data type, source, or repair packet). This would not be beneficial for applications or services that may not need to understand the packet payload. Owing to the possibility that multiple NC packets may have the same name, some mechanism is required for the consumer to obtain innovative packets. As described in Section 6.3, a mechanism for managing the multiple innovative packets in the CS would also be required. In addition, extra computational overhead would be incurred when the payload is being encrypted.

6.2. Transport

The pull-based request-response feature of CCNx/NDN is a fundamental principle of its transport layer; one interest retrieves, at most, one data packet. This means that a forwarder or producer cannot inject unrequested data packets on its own initiative. It is believed that it is important that this rule not be violated, as 1) it would open denial-of-service (DoS) attacks, 2) it invalidates existing congestion control approaches following this rule, and 3) it would reduce the efficiency of existing consumer mobility approaches. Thus, the following basic operation should be considered for applying NC to CCNx/NDN. Nevertheless, such security considerations must be addressed if this rule were to be violated.

6.2.1. Scope of NC

An open question is whether a data forwarder can perform in-network recoding with data packets that are being received in transit or if only the data that matches an interest can be subject to NC operations. In the latter case, encoding or recoding is performed to generate the NC packet at any forwarder that is able to respond to the interest. This could occur when each NC packet has a unique name and interest has the full name. On the other hand, if interest has a partial name without any coding vector information or multiple NC packets have the same name, the former case may occur; recoding occurs anywhere in the network where it is possible to modify the received NC packet and forward it. As CCNx/NDN comprises mechanisms for ensuring the integrity of the data during transfer, in-network recoding introduces complexities in the network that needs consideration for the integrity mechanisms to still work. Similarly, in-network caching of NC packets at forwarders may be valuable; however, the forwarders would require some mechanisms to validate the NC packets (see Section 9).

6.2.2. Consumer Operation

To obtain NC benefits (possibly associated with in-network caching), the consumer is required to issue interests that direct the forwarder (or producer) to respond with innovative packets if available. In the case where each NC packet may have a unique name (as described in Section 6.1), by issuing an interest specifying a unique name with g-id and the coding vector for an NC packet, the consumer could appropriately receive an innovative packet if it is available at some forwarders.

In order to specify the exact name of the NC packet to be retrieved, the consumer is required to know the valid naming scheme. From a practical viewpoint, it is desirable for the consumer application to automatically construct the right name components without depending on any application specifications. To this end, the consumer application may retrieve and refer to a manifest [17] that enumerates the content objects, including NC packets, or may use some coding scheme specifier as a name component to construct the name components of interests to request innovative packets.

Conversely, the consumer without decoding capability (e.g., specific sensor node) may want to receive only the source packets. As described in Section 6.1, because the NC packet can have a name that is explicitly different from source packets, issuing interests for retrieving source packets is possible.

6.2.3. Forwarder Operation

If the forwarder constantly responds to the incoming interests by returning non-innovative packets, the consumer(s) cannot decode and obtain the source packets. This issue could happen when 1) incoming interests for NC packets do not specify some coding parameters, such as the coding vectors to be used, and 2) the forwarder does not have a sufficient number of linearly independent NC packets (possibly in the CS) to use for recoding. In this case, the forwarder is required to determine whether or not it can generate innovative packets to be forwarded to the interface(s) at which the interests arrived. An approach to deal with this issue is that the forwarder maintains a tally of the interests for a specific name, generation ID, and the incoming interface(s) in order to record how many degrees of freedom have already been provided [21]. As such a scheme requires state management (and potentially timers) in forwarders, scalability and practicality must be considered. In addition, some transport mechanism for in-network loss detection and recovery [25][26] at a forwarder, as well as a consumer-driven mechanism, could be indispensable for enabling fast loss recovery and realizing NC gains. If a forwarder cannot either return a matching innovative packet from its local content store, nor produce on the fly a recoded packet that is innovative, it is important that the forwarder not simply return a non-innovative packet but instead do a forwarding lookup in its FIB and forward the interest toward the producer or upstream forwarder that can provide an innovative packet. In this context, to retrieve an innovative packet effectively and quickly, an appropriate setting of the FIB and efficient interest-forwarding strategies should also be considered.

In another possible case, when receiving interests only for source packets, the forwarder may attempt to decode and obtain all the source packets and store them (if the full cache capacity are available), thus enabling a faster response to subsequent interests. As recoding or decoding results in an extra computational overhead, the forwarder is required to determine how to respond to received interests according to the use case (e.g., a delay-sensitive or delay-tolerant application) and the forwarder situation, such as available cache space and computational capability.

6.2.4. Producer Operation

Before performing NC for specified content in CCNx/NDN, the producer is responsible for splitting the overall content into small content objects to avoid packet fragmentation that could cause unnecessary packet processing and degraded throughput. The size of the content objects should be within the allowable packet size in order to avoid packet fragmentation in a CCNx/NDN network. The producer performs the encoding operation for a set of the small content objects and the naming process for the NC packets.

If the producer takes the lead in determining what coding vectors to use in generating the NC packets, there are three general strategies for naming and producing the NC packets:

1. Consumers themselves understand in detail the naming conventions used for NC packets and thereby can send the corresponding interests toward the producer to obtain NC packets whose coding parameters have already been determined by the producer.

2. The producer determines the coding vectors and generates the NC packets after receiving interests specifying the packets the consumer wished to receive.
3. The naming scheme for specifying the coding vectors and corresponding NC packets is explicitly represented via a "Manifest" (e.g., FLIC [30]) that can be obtained by the consumer and used to select among the available coding vectors and their corresponding packets and thereby send the corresponding interests.

In the first case, although the consumers cannot flexibly specify a coding vector for generating the NC packet to obtain, the latency for obtaining the NC packet is less than in the latter two cases. For the second case, there is a latency penalty for the additional NC operations performed after receiving the interests. For the third case, the NC packets to be included in the manifest must be pre-computed by the producer (since the manifest references NC packets by their hashes, not their names), but the producer can select which to include in the manifest and produce multiple manifests either in advance or on demand with different coding tradeoffs, if so desired.

A common benefit of the first two approaches to end-to-end coding is that, if the producer adds a signature on the NC packets, data validation becomes possible throughout (as is the case with the CCNx/NDN operation in the absence of NC). The third approach of using a manifest trades off the additional latency incurred by the need to fetch the manifest against the efficiency of needing a signature only on the manifest and not on each individual NC packet.

6.2.5. Backward Compatibility

NC operations should be applied in addition to the regular ICN behavior and should function alongside regular ICN operations. Hence, nodes that do not support NC should still be able to properly handle packets, not only in being able to forward the NC packets but also to cache these packets. An NC framework should be compatible with a regular framework in order to facilitate backward compatibility and smooth migration from one framework to the other.

6.3. In-Network Caching

Caching is a useful technique used for improving throughput and latency in various applications. In-network caching in CCNx/NDN essentially provides support at the network level and is highly beneficial, owing to the involved exploitation of NC for enabling effective multicast transmission [31], multipath data retrieval [21] [24], and fast loss recovery [26]. However, there remain several issues to be considered.

There generally exist limitations in the CS capacity, and the caching policy affects the consumer's performance [32] [33] [34]. It is thus crucial for forwarders to determine which content objects should be cached and which discarded. As delay-sensitive applications often do not require an in-network cache for a long period, owing to their real-time constraints, forwarders have to know the necessity for caching received content objects to save the caching volume. In CCNx, this could be made possible by setting a Recommended Cache Time (RCT) in the optional header of the data packet at the producer side. The RCT serves as a guideline for the CS cache in determining how long to retain the content object. When the RCT is set as zero, the forwarder recognizes that caching the content object is not useful. Conversely, the forwarder may cache it when the RCT has a greater value. In NDN, the TLV type of FreshnessPeriod could be used.

One key aspect of in-network caching is whether or not forwarders can cache NC packets in their CS. They may be caching the NC packets without having the ability to perform a validation of the content objects. Therefore, the caching of the NC packets would require some mechanism to validate the NC packets (see Section 9). In the case wherein the NC packets have the same name, it would also require some mechanism to identify them.

6.4. Seamless Consumer Mobility

A key feature of CCNx/NDN is that it is sessionless, which enables the consumer and forwarder to send multiple interests toward different copies of the content in parallel, by using multiple interfaces at the same time in an asynchronous manner. Through the multipath data retrieval, the consumer could obtain the content from multiple copies that are distributed while using the aggregate capacity of multiple interfaces. For the link between the consumer and the multiple copies, the consumer can perform a certain rate adaptation mechanism for video streaming [24] or congestion control for content acquisition [35].

NC adds a reliability layer to CCNx in a distributed and asynchronous manner, because NC provides a mechanism for ensuring 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 applies to consumer mobility events [24], wherein the consumer could receive additional degrees of freedom with any innovative packet if at least one available interface exists during the mobility event. An interest-forwarding strategy at the consumer (and possibly forwarder) for efficiently obtaining innovative packets would be required for the consumer to achieve seamless consumer mobility.

7. Challenges

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

7.1. Adoption of Convolutional Coding

Several block coding approaches have been proposed thus far; however, there is still not sufficient discussion and application of the convolutional coding approach (e.g., sliding or elastic window coding) in CCNx/NDN. Convolutional coding is often appropriate for situations wherein a fully or partially reliable delivery of continuous data flows is required and especially when these data flows feature real-time constraints. As in [36], on an end-to-end coding basis, it would be advantageous for continuous content flow to adopt sliding window coding in CCNx/NDN. In this case, the producer is required to appropriately set coding parameters and let the consumer know the information, and the consumer is required to send interests augmented with feedback information regarding the data reception and/or decoding status. As CCNx/NDN utilizes the hop-by-hop forwarding state, it would be worth discussing and investigating how convolutional coding can be applied in a hop-by-hop manner and what benefits might accrue. In particular, in the case wherein in-network recoding could occur at forwarders, both the encoding window and CS management would be required, and the corresponding feasibility and practicality should be considered.

7.2. Rate and Congestion Control

The addition of redundancy using repair packets may result in further network congestion and could adversely affect the overall throughput. In particular, in a situation wherein 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 necessary that each content flow cooperatively implement congestion control to adjust the consumed bandwidth [37]. From this perspective, an effective deployment approach (e.g., a forwarder-supported approach that can provide benefits under partial deployment) is required.

As described in Section 6.4, NC can contribute to seamless consumer mobility by obtaining innovative packets without receiving duplicated packets through multipath data retrieval, and avoiding duplicated packets has congestion control benefits as well. It can be challenging to develop an effective rate and congestion control mechanism in order to achieve seamless consumer mobility while improving the overall throughput or latency by fully exploiting NC operations.

7.3. Security

While CCNx/NDN introduces new security issues at the networking layer that are different from the IP network, such as a cache poisoning, pollution attacks, and a DoS attack using interest packets, some security approaches are already provided [19] [38]. The application of NC in CCNx/NDN brings two potential security aspects that need to be dealt with.

The first is in-network recoding at forwarders. Some mechanism for ensuring the integrity of the NC packets newly produced by in-network recoding is required in order for consumers or other forwarders to receive valid NC packets. To this end, there are some possible approaches described in Section 9, but there may be a more effective method with lower complexity and computation overhead.

The second is that attackers maliciously request and inject NC packets, which could amplify some attacks. As NC packets are unpopular in general use, they could be targeted by a cache pollution attack that requests less popular content objects more frequently to undermine popularity-based caching by skewing the content popularity. Such an attack needs to be dealt with in order to maintain the in-network cache efficiency. By injecting invalid NC packets with the goal of filling the CSs at the forwarders with them, the cache poisoning attack could be effectual depending on the exact integrity coverage on NC packets. On the assumption that each NC packet has the valid signature, the straightforward approach would comprise the forwarders verifying the signature within the NC packets in transit and only transmitting and storing the validated NC packets. However, as performing a signature verification by the forwarders may be infeasible at line speed, some mechanisms should be considered for distributing and reducing the load of signature verification in order to maintain in-network cache benefits, such as latency and network-load reduction.

7.4. Routing Scalability

In CCNx/NDN, a name-based routing protocol without a resolution process streamlines the routing process and reduces the overall latency. In IP routing, the growth in the routing table size has become a concern. It is thus necessary to use a hierarchical naming scheme in order to improve the routing scalability by enabling the aggregation of the routing information.

To realize the benefits of NC, consumers need to efficiently obtain innovative packets using multipath retrieval mechanisms of CCNx/NDN. This would require some efficient routing mechanism to appropriately set the FIB and also an efficient interest-forwarding strategy. Such routing coordination may create routing scalability issues. It would be challenging to achieve effective and scalable routing for interests requesting NC packets, as well as to simplify the routing process.

8. IANA Considerations

This document has no IANA actions.

9. Security Considerations

In-network recoding is a distinguishing feature of NC. Only valid NC packets produced by in-network recoding must be requested and utilized (and possibly stored). To this end, there exist some possible approaches. First, as a signature verification approach, the exploitation of multi-signature capability could be applied. This allows not only the original content producer but also some forwarders responsible for in-network recoding to have their own unique signing key. Each forwarder of the group signs a newly generated NC packet in order for other nodes to be able to validate the data with the signature. The CS may verify the signature within the NC packet before storing it to avoid invalid data caching. Second, as a consumer-dependent approach, the consumer puts a restriction on the matching rule using only the name of the requested data. The interest ambiguity can be clarified by specifying both the name and the key identifier (the producer's public key digest) used for matching to the requested data. This KeyId restriction is built in the CCNx design [17]. Only the requested data packet satisfying the interest with the KeyId restriction would be forwarded and stored in the CS, thus resulting in a reduction in the chances of cache poisoning. Moreover, in the CCNx design, there exists the rule that the CS obeys in order to avoid amplifying invalid data; if an interest has a KeyId restriction, the CS must not reply unless it knows that the signature on the matching content object is correct. If the CS cannot verify the signature, the interest may be treated as a cache miss and forwarded to the upstream forwarder(s). Third, as a certificate chain management approach (possibly without certificate authority), some mechanism, such as [39], could be used to establish a trustworthy data delivery path. This approach adopts the hop-by-hop authentication mechanism, wherein the forwarding-integrated hop-by-hop certificate collection is performed to provide suspension certificate chains such that the data retrieval is trustworthy.

Depending on the adopted caching strategy, such as cache replacement policies, forwarders should also take caution when storing and retaining the NC packets in the CS, as they could be targeted by cache pollution attacks. In order to mitigate the cache pollution attacks' impact, forwarders should check the content request frequencies to detect the attack and may limit requests by ignoring some of the consecutive requests. The forwarders can then decide to apply or change to the other cache replacement mechanism.

The forwarders or producers require careful attention to the DoS attacks aimed at provoking the high load of NC operations by using the interests for NC packets. In order to mitigate such attacks, the forwarders could adopt a rate-limiting approach. For instance, they could monitor the PIT size growth for NC packets per content to detect the attacks and limit the interest arrival rate when necessary. If the NC application wishes to secure an interest (considered as the NC actuator) in order to prevent such attacks, the application should consider using an encrypted wrapper and an explicit protocol.

10. Informative References

[1] Jacobson, V., Smetters, D., Thornton, J., Plass, M., Briggs, N., and R. Braynard, "Networking Named Content", Proc. CoNEXT, ACM, DOI 10.1145/1658939.1658941, December 2009, <https://doi.org/10.1145/1658939.1658941>.

[2] Zhang, L., Afanasyev, A., Burke, J., Jacobson, V., Claffy, K., Crowley, P., Papadopoulos, C., Wang, L., and B. Zhang, "Named data networking", ACM SIGCOMM Computer Communication Review, vol. 44, no. 3, DOI 10.1145/2656877.2656887, July 2014, <https://doi.org/10.1145/2656877.2656887>.

[3] Gkantsidis, C. and P. Rodriguez, "Cooperative Security for Network Coding File Distribution", Proc. Infocom, IEEE, DOI 10.1109/INFOCOM.2006.233, April 2006, <https://doi.org/10.1109/INFOCOM.2006.233>.

[4] Cai, N. and R. Yeung, "Secure network coding", Proc. International Symposium on Information Theory (ISIT), IEEE, DOI 10.1109/ISIT.2002.1023595, June 2002, <https://doi.org/10.1109/ISIT.2002.1023595>.

[5] Lima, L., Gheorghiu, S., Barros, J., Medard, M., and A. Toledo, "Secure Network Coding for Multi-Resolution Wireless Video Streaming", IEEE Journal of Selected Area (JSAC), vol. 28, no. 3, DOI 10.1109/JSAC.2010.100409, April 2010, <https://doi.org/10.1109/JSAC.2010.100409>.

[6] Vilea, J., Lima, L., and J. Barros, "Lightweight security for network coding", Proc. ICC, IEEE, DOI 10.48550/arXiv.0807.0610, May 2008, <https://doi.org/10.48550/arXiv.0807.0610>.

[7] Koetter, R. and M. Medard, "An Algebraic Approach to Network Coding", IEEE/ACM Transactions on Networking, vol. 11, no. 5, DOI 10.1109/TNET.2003.818197, October 2003, <https://doi.org/10.1109/TNET.2003.818197>.

[8] Vyetrenko, S., Ho, T., Effros, M., Kliewer, J., and E. Erez, "Rate regions for coherent and noncoherent multisource network error correction", Proc. International Symposium on Information Theory (ISIT), IEEE, DOI 10.1109/ISIT.2009.5206077, June 2009, <https://doi.org/10.1109/ISIT.2009.5206077>.

[9] Wu, Y., Chou, P., and K. Jain, "A comparison of network coding and tree packing", Proc. International Symposium on Information Theory (ISIT), IEEE, DOI 10.1109/ISIT.2004.1365182, June 2004, <https://doi.org/10.1109/ISIT.2004.1365182>.

[10] Ho, T., Medard, M., Koetter, R., Karger, D., Effros, M., Shi, J., and B. Leong, "A Random Linear Network Coding Approach to Multicast", IEEE Trans. on Information Theory, vol. 52, no. 10, DOI 10.1109/TIT.2006.881746, October 2006, <https://doi.org/10.1109/TIT.2006.881746>.

[11] 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, DOI 10.1109/TIT.2010.2054295, September 2010, <https://doi.org/10.1109/TIT.2010.2054295>.

[12] Gkantsidis, C. and P. Rodriguez, "Network coding for large scale content distribution", Proc. Infocom, IEEE, DOI 10.1109/INFCOM.2005.1498511, March 2005, <https://doi.org/10.1109/INFCOM.2005.1498511>.

[13] Seferoglu, H. and A. Markopoulou, "Opportunistic Network Coding for Video Streaming over Wireless", Proc. Packet Video Workshop (PV), IEEE, DOI 10.1109/PACKET.2007.4397041, November 2007, <https://doi.org/10.1109/PACKET.2007.4397041>.

[14] 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, DOI 10.1145/2248361.2248370, June 2012, <https://doi.org/10.1145/2248361.2248370>.

[15]

Adamson, B., Adjih, C., Bilbao, J., Firoiu, V., Fitzek, F., Ghanem, S., Lochin, E., Masucci, A., Montpetit, M-J., Pedersen, M., Peralta, G., Roca, V., Ed., Saxena, P., and S. Sivakumar, "Taxonomy of Coding Techniques for Efficient Network Communications", RFC 8406, DOI 10.17487/RFC8406, June 2018, <https://www.rfc-editor.org/info/rfc8406>.

[16] Wissingh, B., Wood, C., Afanasyev, A., Zhang, L., Oran, D., and C. Tschudin, "Information-Centric Networking (ICN): Content-Centric Networking (CCNx) and Named Data Networking (NDN) Terminology", RFC 8793, DOI 10.17487/RFC8793, June 2020, <https://www.rfc-editor.org/info/rfc8793>.

[17] Mosko, M., Solis, I., and C. Wood, "Content-Centric Networking (CCNx) Semantics", RFC 8569, DOI 10.17487/RFC8569, July 2019, <https://www.rfc-editor.org/info/rfc8569>.

[18] Mosko, M., Solis, I., and C. Wood, "Content-Centric Networking (CCNx) Messages in TLV Format", RFC 8609, DOI 10.17487/RFC8609, July 2019, <https://www.rfc-editor.org/info/rfc8609>.

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

[20] Wu, Q., Li, Z., Tyson, G., Uhlig, S., Kaafar, M., and G. Xie, "Privacy-Aware Multipath Video Caching for Content-Centric Networks", IEEE Journal on Selected Areas in Communications (JSAC), vol. 38, no. 8, DOI 10.1109/JSAC.2016.2577321, June 2016, <https://doi.org/10.1109/JSAC.2016.2577321>.

[21] Saltarin, J., Bourtsoulatze, E., Thomos, N., and T. Braun, "NetCodCCN: a network coding approach for content-centric networks", Proc. Infocom, IEEE, DOI 10.1109/INFOCOM.2016.7524382, April 2016, <https://doi.org/10.1109/INFOCOM.2016.7524382>.

[22] 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, DOI 10.1109/IFIPNetworking.2014.6857127, June 2014, <https://doi.org/10.1109/IFIPNetworking.2014.6857127>.

[23] 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, DOI 10.1016/j.comnet.2016.08.004, August 2016, <https://doi.org/10.1016/j.comnet.2016.08.004>.

[24] 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, DOI 10.1109/ISM.2016.0054, December 2016, <https://doi.org/10.1109/ISM.2016.0054>.

[25] 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. of the 3rd ACM Conference on Information-Centric Networking, DOI 10.1145/2984356.2984361, September 2016, <https://doi.org/10.1145/2984356.2984361>.

[26] Matsuzono, K., Asaeda, H., and T. Turletti, "Low Latency Low Loss Streaming using In-Network Coding and Caching", Proc. Infocom, IEEE, DOI 10.1109/INFOCOM.2017.8057026, May 2017, <https://doi.org/10.1109/INFOCOM.2017.8057026>.

[27] Watson, M., Begen, A., and V. Roca, "Forward Error Correction (FEC) Framework", RFC 6363, DOI 10.17487/RFC6363, October 2011, <https://www.rfc-editor.org/info/rfc6363>.

[28] Thomos, N. and P. Frossard, "Toward One Symbol Network Coding Vectors", IEEE Communications Letters, vol. 16, no. 11, DOI 10.1109/LCOMM.2012.092812.121661, November 2012, <https://doi.org/10.1109/LCOMM.2012.092812.121661>.

[29] Lucani, D., Pedersen, M., Ruano, D., Sørensen, C., Heide, J., Fitzek, F., and O. Geil, "Fulcrum Network Codes: A Code for Fluid Allocation of Complexity", DOI 10.48550/arXiv.1404.6620, April 2014, <https://doi.org/10.48550/arXiv.1404.6620>.

[30] Tschudin, C., Wood, C. A., Mosko, M., and D. Oran, "File-Like ICN Collections (FLIC)", Work in Progress, Internet-Draft, draft-irtf-icnrg-flic-03, November 2021, <https://www.ietf.org/archive/id/draft-irtf-icnrg-flic-03.txt>.

[31] Maddah-Ali, M. and U. Niesen, "Coding for Caching: Fundamental Limits and Practical Challenges", IEEE Communications Magazine, vol. 54, no. 8, DOI 10.1109/MCOM.2016.7537173, August 2016, <https://doi.org/10.1109/MCOM.2016.7537173>.

[32] Perino, D. and M. Varvello, "A reality check for content centric networking", Proc. SIGCOMM Workshop on Information-centric networking (ICN '11), ACM, DOI 10.1145/2018584.2018596, August 2011, <https://doi.org/10.1145/2018584.2018596>.

[33] Podlipnig, S. and L. Böszörmenyi, "A Survey of Web Cache Replacement Strategies", Proc. ACM Computing Surveys, vol. 35, no. 4, DOI 10.1145/954339.954341, December 2003, <https://doi.org/10.1145/954339.954341>.

[34] Rossini, G. and D. Rossi, "Evaluating CCN multi-path interest forwarding strategies", Elsevier Computer Communications, vol. 36, no. 7, DOI 10.1016/j.comcom.2013.01.008, April 2013, <https://doi.org/10.1016/j.comcom.2013.01.008>.

[35] Mahdian, M., Arianfar, S., Gibson, J., and D. Oran, "MIRCC: Multipath-aware ICN Rate-based Congestion Control", Proc. Conference on Information-Centric Networking (ICN), ACM, DOI 10.1145/2984356.2984365, September 2016, <https://doi.org/10.1145/2984356.2984365>.

[36] Tournoux, P., Lochin, E., Lacan, J., Bouabdallah, A., and V. Roca, "On-the-Fly Erasure Coding for Real-Time Video Applications", IEEE Transactions on Multimedia, vol. 13, no. 4, DOI 10.1109/TMM.2011.2126564, August 2011, <https://doi.org/10.1109/TMM.2011.2126564>.

[37] Kuhn, N., Lochin, E., Michel, F., and M. Welzl, "Forward Erasure Correction (FEC) Coding and Congestion Control in Transport", RFC 9265, DOI 10.17487/RFC9265, July 2022, <https://www.rfc-editor.org/info/rfc9265>.

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

[39] Li, R., Asaeda, H., and J. Wu, "DCAuth: Data-Centric Authentication for Secure In-Network Big-Data Retrieval", IEEE Trans. on Network Science and Engineering, vol. 7, no. 1, DOI 10.1109/TNSE.2018.2872049, September 2018, <https://doi.org/10.1109/TNSE.2018.2872049>.

Acknowledgments

The authors would like to thank ICNRG and NWCRG members, especially Marie-Jose Montpetit, David Oran, Vincent Roca, and Thierry Turletti, for their valuable comments and suggestions on this document.

Authors' Addresses

Kazuhisa Matsuzono
National Institute of Information and Communications
Technology
4-2-1 Nukui-Kitamachi, Tokyo
184-8795
Japan
Email: matsuzono@nict.go.jp

Hitoshi Asaeda
National Institute of Information and Communications
Technology
4-2-1 Nukui-Kitamachi, Tokyo
184-8795
Japan
Email: asaeda@nict.go.jp

Cedric Westphal
Huawei
2330 Central Expressway
Santa Clara, California 95050
United States of America
Email: cedric.westphal@futurewei.com,