IRTF Network Coding Working Group
Internet-Draft
Intended status: Experimental
Expires: September 10, 2015

M. Montpetit
MIT
V. Roca
INRIA
J. Detchart
ISAE

March 9, 2015

Dynamic Network Coding
draft-montpetit-dynamic-nc-00

Abstract

This document presents a network coding approach that allows coding decisions to be based on the instantaneous conditions at the network nodes. It uses dynamic rates and coefficients to constantly adapt to local conditions and to allow for provider and application differentiation.

Status of This Memo

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

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

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

This Internet-Draft will expire on September 10, 2015.

Table of Contents

1. Introduction

Network Coding has proven to be an efficient mechanism to provide increased quality of experience for applications depending on Internet Protocols. Current implementations, be them end-to-end or hop-by-hop, depend on a global decision on the type and applicability of the code. However, the heterogeneous nature of IP networks, the differences between transported traffics and the rise of Information Centric Networks (ICN) and multiple in network caches require to define alternatives to current solutions.

Dynamic Network Coding (DNC) intends to use the characteristics of the rising Internet traffic (Internet of Things, streaming, progressive downloads etc.), the use of in-network caching opportunities, customer and policy management and the changing

dynamics on wireless links. These characteristics will be used to adapt the network coding scheme, rate and coefficients to provide adaptive behavior, differentiation and varying quality of experience. In addition, it will also allow to support emerging Internet Architectures such as the ICN that are considering network coding solutions [ICN].

This draft provides the basic elements of such a dynamic code.

2. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

3. Definitions, Notations and Abbreviations

3.1. Definitions

This document uses the following definitions, that are mostly inspired from from [RFC5052], [RFC6363].

-- Editor's note: most of the definitions should be moved to the future NC Architectural document. --

Input Symbol: a unit of data that is provided as an input to the coding process, in a given coding node. It may be a source symbol or an already encoded repair symbol

Output Symbol: a unit of data that is produced as an output of the coding process, in a given coding node

Source Symbol: an original unit of data, before any coding process is applied

Repair Symbol: an Output Symbol that is not a Source Symbol

Code Rate: the ratio between the number of Input Symbols and the number of Output Symbols at given coding node. The Code Rate belongs to a]0; 1] interval. A Code Rate close to 1 indicates that a small number of Output Symbols have been produced during the encoding process

Systematic Code: NC code in which the Source Symbols are part of the Output Symbols

DNC Packet: a packet (e.g., carried as the payload of a UDP datagram) containing a given Output Symbol plus its associated DNC header.

Packet Erasure Channel: a communication path where packets are either dropped (e.g., by a congested router or because the number of transmission errors exceeds the correction capabilities of the physical layer codes) or received. When a packet is received, it is assumed that this packet is not corrupted

To Be Completed

-- Editor's note: Should we consider the possibility of having several symbols per packet? The DNC packet should be updated accordingly in that case. --

3.2. Notations

This document uses the following notations:

CR denotes the Code Rate

To Be Completed

3.3. Abbreviations

This document uses the following abbreviations:

The following definitions are compatible with the FECFRAME framework [RFC6363] and the NC architecture (COMMENT: reference to be added when available).

NC: Network Coding

DNC: Dynamic Network Coding

PRNG: Pseudo-Random Number Generator

TS: Time Stamp

ESI: Encoding Symbol ID

To Be Completed

4. Dynamic Network Coding (DNC) Principles

Network Coding is based on the linear combination of a number of input symbols (at least 1) into a number of output symbols (at least 1), between the ingress and egress of the network. Several such encoding operations can happen in case of in-network re-coding, or there can be a single encoding operation, especially when it is applied end-to-end. In the RLNC [RLNC], Tetrys [I-D.detchart-nwcrg-tetrys], and Dragoncast [I-D.adjih-dragoncast] instances of Network Coding, the linear combination consists in applying a set of coding coefficients to each input symbol of the current encoding window. Here the coding vectors are chosen in a deterministic manner at each encoding node, for instance based on local criteria, or are generated using a PRNG seed chosen locally and carried along with the output symbol.

The DNC proposal extends this process as follows: the set of coding coefficients is determined based on the FEC_Encoding ID, Codepoint, and TS header fields of the various input packets present in the encoding window, plus the local time. The coding decision therefore depends on time, among other pieces of information.

4.1. About the Use of Timestamps in DNC

Each DNC Packet contains timing information: this can be the TS field of the DNC header, or the timestamp field of an NTP header if already present in the packet. This timing information (noted TS hereafter, no matter its origin) is used to determine the coding coefficients in a coding node. When several DNC packets are present in the encoding window, originating from one or several sources, a decision on which TS will be sent downstream in the network must be taken. Options include keeping the oldest TS value, the newest TS value, or generating a local TS value. It is assumed that the granularity of the TS in choosing the coding coefficients would be to the second tin order to react to instantaneous condition.

It is also possible to use the TS in a wider sense, to link to network operations and coding based police management. This includes the determination of the coding window, Code Rate, Galois field, etc.

-- Editor's note: This extension is left for future specifications as it requires closer coordination between network management. policy and coding. --

4.2. About the FEC Encoding ID, Codepoint and Coding Decisions

The FEC Encoding ID is used to identify the type of code (i.e., FEC Scheme) to use at each coding node. This is an integer identifier assigned by IANA. The value of the FEC Encoding ID MAY vary over the time, within the same flow, and/or across the same path between several coding nodes. Different coding decisions can be made by different management entities with different operating constraints (for instance content provider versus network operator).

> -- Editor's note: Having the possibility to change the coding decisions can have major practical technical implications that are not considered for the moment. --

The Codepoint is an opaque value to be used along with the FEC Encoding ID and TS. The {FEC Encoding ID, Codepoint, TS} tuple identifies uniquely the FEC Scheme used and the set of coding coefficients. Examples are provided below on how to do that.

5. Dynamic Network Coding (DNC) Procedures

5.1. Input Symbol Creation

Incoming DNC packets at a coding node are not necessarily of the same fixed size. This size may largely vary over the time, up to a maximum size that is related to the Link MTU and/or Path MTU. This is a problem when Repair Symbols need to be produced by a coding node.

Let us consider the simple case where the FEC Scheme is such that a Repair Symbol necessarily spans the payload of the largest Incoming DNC Packet at the encoding window of the coding node. Let L be equal to the maximum size of the DNC packets in the encoding window plus 2 bytes. The 2 bytes are used to store the original size of the packets. Any DNC packet of size inferior to L-2 bytes MUST be zero padded to achieve the desired length of L bytes. Then any Repair Symbol within the set of Output Symbols is L bytes long. When a Source Symbol is erased and later decoded, the first two bytes of the decoded symbol, that contain the symbol_len field, allows to drop the potential padding.

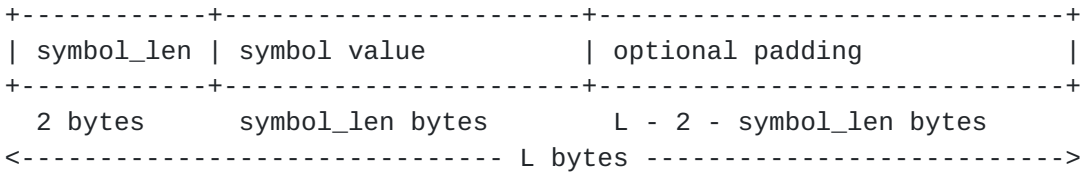

Figure 1: Symbol Format, Case of a Single Flow

-- Editor's note: in presence of multiple flows, an additional FlowID field may be prepended in order to identify the flow this Input Symbol belongs to. The details are TBD. --

5.2. Other Procedures TBD

TBD

For instance it may be interesting to have a feedback flow to enable a receiver to adjust its encoding window according to the received or decoded Source Symbols.

6. Application of Dynamic Network Coding to Various Use-cases

-- Editor's note: This section contains material that may be more appropriate in a future NC Architecture document. --

Several technical aspects need to be considered:

- Intra-flow encoding: (single flow) here all the Source Symbols belong to the same and unique flow;

- Inter-flow encoding: (multiple flows) here the Source Symbols belong to two or more flows. This situation is more complex, in particular because it requires to remember the flow a given source symbol belongs to in case this latter gets lost and is recovered by a decoding node;

- End to end: (single coding node) here there is a single coding node and a single decoding node. However the coding and/or decoding nodes are not necessarily the source and destination nodes. For instance these operations can be performed by middle boxes, or coding may be done in a middle box while decoding is performed at the destination node, or vice-versa. The nature of the coding and decoding nodes should not significantly impact the way DNC operates;

  -- Editor's note: This claim is TBC. --

- In network re-coding: (composability) here several coding nodes exist along the path. This situation significantly impacts the way DNC operates, in particular in terms of signaling. Indeed a decoding node MUST be able to identify the exact manner in which a given Repair Symbol has been generated, recursively since this Repair Symbol MAY be the result of several coding operations, at different coding nodes;

- Multi-source: here several traffic sources exist. They either jointly contribute to the same data flow, all sources sending traffic related to the same content, or they contribute to multiple data flows, sources sending traffic for different contents;

- Multi-paths: here the traffic follows multiple paths. This traffic can be originated from a unique source (e.g., with multi-path TCP, MPTCP), or with multiple sources which is the more general case;

The above taxonomy can be used to identify several types of use-cases. In the following we consider some of the potential use-cases and explain how DNC can be applied. This is in no case an exhaustive list and will be adapted in the future as we get more insights on the code usage.

6.1. Single Flow, Single Source, End-to-end, Single Path or Multi Paths Use-Case

In this use-case, there is a single flow originated by a single source, with intra stream coding that takes place at a single coding node. There can be either a single path or multiple paths, since this situation does not impact the way DNC operates.

This is the simplest use-case, that is very much inline with currently proposed scenarios for end to end streaming.

6.1.1. Example of DNC Implementation for this Use-Case

The following DNC approach / FEC Scheme is appropriate for this simple use-case. However this is only an example and other approaches may be considered, for instance differing in the way the {FEC Encoding ID, Codepoint, TS} tuple is used.

Let us consider a FEC Scheme that defines the G table containing NL lists, each list being populated with NC coefficients over a given Finite Field, say GF(2^^4). This table, G, is contained in the FEC Scheme specification. Each coding and decoding node supports this FEC Scheme (and potentially a certain number of additional FEC Schemes), this latter being identified by an IANA FEC Encoding ID value.

- Encoding process: Let W be the encoding window, containing K Input Symbols (constructed as specified in Section 5.1). It is required that K be at most equal to the number of coefficients in each row of G. If this is not the case, an appropriate number of symbols are removed from window W until this property holds. Let TS be

the timestamp corresponding to the current time at the coding node. Let Codepoint be the current integer value of a counter that is managed sequentially, starting a 0 upon initializing the DNC coding node instance, and wrapping to 0 upon reaching the maximum counter value.

-- Editor's note: The counter field size, in bits, is not specified in this version of the document. 32 bits or 16 bits are two possible values. --

Let r be an integer calculated as r=f(Codepoint, TS, K); where f is a deterministic function that produces an integer in the {0; K-1} range. This function is for instance the result of a hash over {Codepoint, TS} modulo K.

-- Editor's note: The FEC Scheme specification fully specifies this f function. --

The output Repair Symbol is computed as the XOR sum of each Input Symbol in W multiplied by the corresponding coefficient in row r of G (i.e., the first symbol is multiplied by G[r][0], the second symbol is multiplied by G[r][1], etc. til the last symbol of W).

- Transmitted Output Symbol: The FEC Encoding ID is communicated in the DNC packet header. The {Codepoint, TS} tuple can be used to uniquely identify the Repair Symbol produced by the coding process. This tuple is communicated in the DNC packet header. The ESI of each packet currently in W is communicated in the DNC packet header. This information can consist of a list of ESIs, or in the simple case where they are all in sequence, the {first ESI, K} tuple can be communicated, or a sequence of such tuples can be sent in the case where ESIs are mostly in sequence with a limited number of gaps, of the first ESI along with a bit field (e.g., of size 64 bits if K is at most 64) can be communicated.

  -- Editor's note: Many other pieces of information will be transmitted for other features. The DNC header format also remains TBD. --

- Processing at the decoding node: Upon receiving this Repair Symbol, an additional equation is added to the current linear system. The FEC Encoding ID enables the decoding node to identify the FEC Scheme used by the coding node, as well as the G matrix. The {Codepoint, TS} tuple enables the decoding node to identify the set of coding coefficients used by the coding node.

- Decoding process: If the linear system enables it, a decoding process is used to recover an erased Source Symbol. The first two

bytes of the decoded symbol is read and used to identify the initial length of the Source Symbol and to remove padding if needed. The decoded Source Symbol is then communicated to the receiving node (or receiving application).

6.2. Single Flow with In-network Re-Coding

In this use-case, there is a single flow, originated either by a single source or by multiple source for the same content, with in-network re-coding capabilities. There can be either a single path or multiple paths (e.g., if there are multiple sources). There are multiple sub use-cases, among which the following three ones.

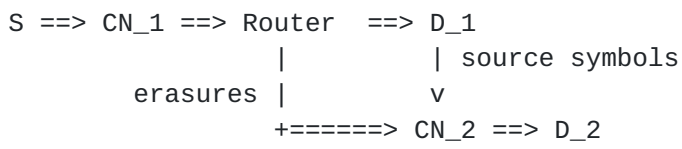

Figure 2

In this first scenario CN_1 et CN_2 are two NC encoders. The flow arriving in CN_2 from the router is impacted by erasures. However this is not the case of the links to destination D_1, that has decoded the packets and this node retransmits the source symbols received or recovered towards D_2. In CN_2 all symbols are recombined and sent to D_2. This could be a scenario that combines wired and wireless paths.

Another scenario is the following, where two sources generate some traffic for the same content:

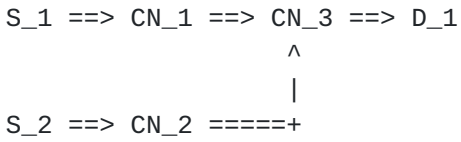

Figure 3

Here S_1 and S_2 are transmitting the same content. The flow coming from S_1 (resp. S_2) is encoded at CN_1 (resp. CN_2), and the two encoded flows are later re-encoded in CN_3, a third NC encoder, on their way to D_1.

Finally, with a single flow passing through wired and wireless paths, the following scenario is likely.

S ==> CN_1 ==> low losses ==> CN_2 ==> high losses ==> D_1

Figure 4

Between CN_1 and CN_2, the network is a wired internet and a high rate code (i.e., adding a limited number of encoded packets) can be used (e.g., it may be a simple XOR of packets as in QUIC [QUIC]). Between CN_2 et D_1 the link can be a WIFI or LTE wireless network, potentially experiencing higher losses. Consequently a higher number of Repair Symbols (i.e., lower code rate) can be needed, with potentially a local feedback loop that enables to adapt the code rate based on the instantaneous conditions observed over that lossy link.

6.3. Other Use Cases

Many use-cases remain TBD, for instance to cover the Peer to peer, complex multipath, or storage use-cases.

7. Acknowledgements

M-J. Montpetit would like to thank Huawei and in particular Shucheng Liu for supporting the work leading to this draft.

8. IANA Considerations

TBD

9. Security Considerations

TBD, see RFC 3552 [RFC3552].

10. References

10.1. Normative References

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

10.2. Informative References

[I-D.adjih-dragoncast] Adjih, C., Cho, S., and E. Baccelli, "Broadcast With Network Coding: DRAGONCAST", draft-adjih-dragoncast-00 (work in progress), July 2013.

[I-D.detchart-nwcrg-tetrys]
jonathan.detchart@isae.fr, j., Lochin, E., Lacan, J., and V. Roca, "Tetrys, an On-the-Fly Network Coding protocol", draft-detchart-nwcrg-tetrys-00 (work in progress), October 2014.

[ICN] Montpetit, M., Wesphal, C., and D. Trossen, "Network coding meets information-centric networking: an architectural case for information dispersion through native network coding", Proceedings of the 1st ACM workshop on Emerging Name-Oriented Mobile Networking Design - Architecture, Algorithms, and Applications (NOM'2012), June 2012.

[QUIC] Roskind, JR., "QUIC: Quick UDP Internet Connections Multiplexed Stream Transport", IETF-88 TSV Area Presentation, November 2013.

[RFC3552] Rescorla, E. and B. Korver, "Guidelines for Writing RFC Text on Security Considerations", BCP 72, RFC 3552, July 2003.

[RFC5052] Watson, M., Luby, M., and L. Vicisano, "Forward Error Correction (FEC) Building Block", RFC 5052, August 2007.

[RFC6363] Watson, M., Begen, A., and V. Roca, "Forward Error Correction (FEC) Framework", RFC 6363, October 2011.

[RLNC] Ho, T., Koetter, R., Medard, M., Karger, D., Effros, M., Shi, J., and B. Leung, "A Random Linear Network Coding Approach to Multicast", IEEE Transactions on Information Theory Vol. 52 No. 10, October 2006.

Appendix A. Appendix: IPR aspects

IPR aspects if any to be provided later

Authors' Addresses

Marie-Jose Montpetit
MIT
Cambridge, MA 02138
USA

Email: mariejo@mit.edu

Vincent Roca
INRIA
655, av. de l'Europe
Inovallee; Montbonnot
ST ISMIER cedex 38334
France

Email: vincent.roca@inria.fr
URI: http://privatics.inrialpes.fr/people/roca/

Jonathan Detchart
ISAE
10, avenue Edouard-Belin
BP 54032
Toulouse CEDEX 4 31055
France

Email: jonathan.detchart@isae.fr