Internet Research Task Force (IRTF)
Request for Comments: 8406
Category: Informational
ISSN: 2070-1721

B. Adamson
NRL
C. Adjih
INRIA
J. Bilbao
Ikerlan
V. Firoiu
BAE Systems
F. Fitzek
TU Dresden
S. Ghanem
Independent
E. Lochin
ISAE - Supaero
A. Masucci
Orange
M-J. Montpetit
Independent
M. Pedersen
Aalborg University
G. Peralta
Ikerlan
V. Roca, Ed.
INRIA
P. Saxena
AnsuR Technologies
S. Sivakumar
Cisco
June 2018

Taxonomy of Coding Techniques for Efficient Network Communications

Abstract

This document summarizes recommended terminology for Network Coding concepts and constructs. It provides a comprehensive set of terms in order to avoid ambiguities in future IRTF and IETF documents on Network Coding. This document is the product of the Coding for Efficient Network Communications Research Group (NWCRG), and it is in line with the terminology used by the RFCs produced by the Reliable Multicast Transport (RMT) and FEC Framework (FECFRAME) IETF working groups.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

This document is the product of and represents the collaborative work and consensus of the Coding for Efficient Network Communications Research Group (NWCRG); it is not an IETF product and is not a standard. In 2017, the document was discussed during three audio conferences, each of them gathering 6 to 8 key experts; it was co-edited and subjected to an RG Last Call. The general feeling was that the document was ready. Additional information about Network Coding may be found on these NWCRG pages: <https://irtf.org/nwcrg> and <https://datatracker.ietf.org/rg/nwcrg/about/>.

The literature on Network Coding research and system design, including IETF documentation, led to a rich set of concepts and constructs. This document collects terminology used in the domain, both outside and inside IETF, provides concise definitions, and introduces a high-level taxonomy. Its primary goal is to be useful to IETF and IRTF activities. It is also in line with the terminology already used by the RFCs produced by the Reliable Multicast Transport (RMT) and FEC Framework (FECFRAME) IETF working groups, in particular [RFC5052], [RFC5740], [RFC5775], [RFC6363], and [RFC6726]. This document is also related to IETF work being done in the PAYLOAD and TSVWG WGs (in particular, the extension of FECFRAME to support Sliding Window Codes and the Random Linear Coding (RLC) sliding window FEC scheme) and past work in the AVTCORE and MMUSIC WGs. Note that in the definitions, the "(IETF)" tag indicates that the associated term is already used in IETF documents (Internet-Drafts and RFCs).

This document focuses on packet transmissions and losses. These losses will typically be triggered by various types of networking issues and/or impairments (e.g., congested routers or intermittent wireless connectivity). The notion of "packet" itself is multiform, depending on the target use case and the notion of network (e.g., in which layer of the protocol stack does the coding middleware operate?). For instance, a "packet" may be a data unit to be carried as a UDP payload because the coding middleware is located between the application and UDP. In another configuration, coding may be applied within an overlay network and the notion of "packet" will be totally different. In any case, the goals of Network Coding can be to improve the network throughput, efficiency, latency, and scalability, as well as to provide resilience to partition, attacks, and eavesdropping (NWCRG charter). Both End-to-End Coding and systems that also perform recoding within intermediate forwarding nodes are considered in this document.

This document does not consider physical-layer transmission issues, physical-layer codes, or error detection: if low-layer error codes detect but fail to correct bit errors, or if an upper-layer checksum (e.g., within IP or UDP) identifies a corrupted packet, then the packet is supposed to be dropped.

2. General Definitions and Concepts

This section provides general definitions and concepts that are used throughout this document.

Packet Erasure Channel:
A communication path where packets are either dropped or received without any error. This type of packet drop is referred to as an "erasure" or "loss". The term "channel" must be understood as a generic term for any type of communication technology (e.g., an Ethernet link, a WiFi network, or a full path between two nodes over the Internet). As opposed to the "Erasure" channels, "Error" channels are where one or multiple bit errors may happen during a packet transmission. These "Error" channels are out of scope.

Erasure Correcting Code (ECC) or (IETF) Forward Erasure Correction (FEC):
A code for the Packet Erasure Channel (only). These codes are also called "Application-Level FECs" to highlight that they have been designed for use within the higher layers of the protocol stack to protect against packet losses. As opposed to ECCs/FECs, "Error" correction codes are capable of identifying the presence of bit errors and perhaps correcting them. The "Error" correction codes are out of scope.

End-to-End Coding:
A system where coding is performed at the source or (coding) middlebox, and decoding is performed at the destination(s) or (decoding) middlebox. There is no recoding operation at intermediate nodes. This is the approach followed in the FLUTE/ALC [RFC6726] [RFC5775], NORM [RFC5740], and FECFRAME [RFC6363] protocols.

Network Coding:
A system where coding can be performed at the source as well as at intermediate forwarding nodes (all or a subset of them). End-to-End Coding is regarded as a special case of Network Coding. Depending on the use case, additional assumptions can be made: for instance, the destination knowing the Coding Nodes’ topology and coding operations can help during decoding operations.

Packet versus Symbol:
   Generally speaking, a Packet is the unit of data that is sent in the Packet Erasure Channel, while a Symbol is the unit of data that is manipulated during the encoding and decoding operations.

Original Payload, Uncoded Payload, Systematic Symbol, or (IETF) Source Symbol:
   A unit of data originating from the source that is used as input to encoding operations.

Coded Payload, Coded Symbol, or (IETF) Repair Symbol:
   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 Repair Symbols. When there is a single Repair Symbol per Repair Packet, a Repair Symbol corresponds to a Repair Packet.

Input Symbol and Output Symbol:
   A unit of data that is used as input to an encoding operation or that is generated as output of an encoding operation. At a recoding node, Repair Symbols are also part of the Input Symbols. With Systematic Coding, Source Symbols are also part of the Output Symbols.

(IETF) Encoding Symbol:
   A Source or a Repair Symbol.

(En)coding versus Recoding versus Decoding:
   (En)coding is an operation that takes Source Symbols as input and produces Encoding Symbols as output. Recoding is an operation that takes Encoding Symbols as input and produces Encoding Symbols as output. Decoding is an operation takes Encoding Symbols as input and produces Source Symbols as output.

(IETF) Source Packet:
   A packet originating from the source that contributes to one or more Source Symbols. For instance, an 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.

(IETF) Repair Packet:
   A packet containing one or more Repair Symbols.

Figure 1 illustrates the relationships between packets (what is sent in the Packet Erasure Channel) and symbols (what is manipulated during encoding and decoding operations) in case of a Systematic

Coding at a Coding Node that performs Encoding (rather than Recoding). FEC decoding procedures are similarly performed in the reverse order.

```
           Source Packet
                 |
                 | Source Packet to Source Symbols transform
                 | (one or more symbols per packet)
                 v
           Source Symbols
                 |
                 v Input Symbols
      +----------------------+
      |     FEC encoding     |
      +----------------------+
         | Output Symbols |
         v                v
   Source Symbols   Repair Symbols
         |                |
         |                | symbol-to-packet transform
         |                | (one or more symbols per packet)
         v                v
   Source Packet    Repair Packet
```

Figure 1: Packet and Symbol Relationships at a Coding Node That Performs Encoding (Rather Than Recoding)

Source Node:
A node that generates one or more Source Flows.

Coding Node:
A node that performs FEC Encoding or Recoding operations. It may be an end host or a middlebox (Encoding case), or a forwarding node (Recoding case).

(IETF) Flow:
A stream of packets logically grouped.

(IETF) Source Flow:
A flow of Source Packets coming from an application on a given host and to which FEC encoding is to be applied, potentially along with other Source Flows. Depending on the use case, Source Flows may come from the same application, from different applications on the same host, or from different applications on different hosts.

(IETF) Repair Flow:
A flow containing Repair Packets after FEC encoding.

3. Taxonomy of Code Uses

This section discusses the various ways of using coding, without going into coding details.

Source Coding versus Channel Coding:
(see Figure 2) When both terms are used, "Source Coding" usually refers to compression techniques (e.g., audio and video compression) within the upper application that generates the Source Flow. "Channel Coding" refers to FEC encoding in order to improve transmission robustness, for instance, within the lower physical layer (out of scope of this document) or as part of Network Coding. These terms should not be confused with "FEC coding within the Source Node" and "FEC recoding within an intermediate Coding Node", respectively.

```
   raw data flow from camera     ^              video flow display
               |                 |                      ^
               v                 | upper                |
   +------------------------+    |           +-------------------------+
   |     source coding      |    | applica-  |  source (de)coding      |
   |(e.g., mpeg compression)|    | tion      |(e.g., mpg decompression)|
   +------------------------+    v           +-------------------------+
               |                                        ^
               v                                        |
   +------------------------+    ^           +-------------------------+
   | network/AL-FEC coding  |    | middle-   | network/AL-FEC coding   |
   |  (e.g., RLC encoding)  |    | ware      |  (e.g., RLC decoding)   |
   +------------------------+    v           +-------------------------+
               |                                        ^
               v                                        |
   +------------------------+    ^           +-------------------------+
   |     packetization      |    |           |    depacketization      |
   |     (e.g., UDP/IP)     |    | communi-  |     (e.g., UDP/IP)      |
   +------------------------+    | cation    +-------------------------+
               |                 |                      ^
               v                 | layers               |
   +-----------------------+     |           +-------------------------+
   |       PHY layer       |     |           |       PHY layer         |
   |    (channel coding)   |     |           |   (channel decoding)    |
   +-----------------------+     v           +-------------------------+
               |                                         ^
               |          source + repair traffic        |
               +-----------------------------------------+
```

Figure 2: Example of End-to-End Flow Manipulation with Network Coding

Figure 2 shows Network Coding between the application and UDP layers (as with RMT or FECFRAME architectures). Other architectures are possible, for instance, with Network Coding below the transport layer to allow recoding within the network.

Intra-Flow Coding or Single-Source Network Coding:
Process where incoming packets to the Coding Node belong to the same flow.

Inter-Flow Coding or Multi-Source Network Coding:
Process where incoming packets to the Coding Node belong to different flows.

Single-Path Coding:
Network Coding over a route that has a single path from the source to each destination(s). In case of multicast or broadcast traffic, this route is a tree. Coding may be done end to end and/or at intermediate forwarding nodes.

Multi-Path Coding:
Network Coding over a route that has multiple (at least partially) disjoint paths from the source to each given destination. Coding may be done end to end and/or at intermediate forwarding nodes.

4. Coding Details

4.1. Coding Types

This section provides a high-level taxonomy of coding techniques. Technical details are discussed in subsequent sections.

Linear Coding:
Linear combination of a set of Input Symbols (i.e., Source and/or Repair Symbols) using a given set of coefficients and resulting in a Repair Symbol. Many linear codes exist that differ from the way coding coefficients are drawn from a Finite Field of a given size.

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

Adaptive Linear Coding:
Linear Coding that utilizes cross-layer adaptation. For instance, an adaptive coding scheme may adapt the generation and transmission of Repair Packets according to the channel variations over time, accounting for the predictive loss of degrees of freedom due to erasures.

Block Coding:
Coding technique where the input Flow(s) must first be segmented into a sequence of blocks; FEC encoding and decoding are performed independently on a per-block basis. The term "Chunk Coding" is sometimes used, where a "Chunk" denotes a block.

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 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 the 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).

Systematic Coding:
A coding technique where Source Symbols are part of the output Flow generated by a Coding Node.

Rateless and Non-rateless Coding:
Rateless Coding can generate an unlimited number of Repair Symbols (in practice, this number can be limited by practical considerations or because of use-case requirements) from a given set of Source Symbols, meaning that the code rate is null. RLC codes are an example of Rateless Codes. Alternately, Non-rateless Coding usually has a predefined maximum number of Repair Symbols that can be generated from a given set of Source Symbols.

4.2. Coding Basics

This section discusses and defines low-level coding aspects.

Code Rate:
In case of a Block Code, the Code Rate is the k/n ratio between the number of Source Symbols, k, and the number of Source plus Repair Symbols, n. With a Sliding Window Code, the Code Rate is defined similarly over a certain time interval, since the Code Rate may change dynamically. By definition, the Code Rate is such that: 0 < Code Rate <= 1. A Code Rate close to 1 indicates that a small number of Repair Symbols have been produced during the encoding process and vice versa.

(En)coding Window:
A set of Source (and Repair in the case of recoding) Symbols used as input to the coding operations. The set of symbols will typically change over time, as the Coding Window slides over the input Flow(s).

(En)coding Window Size:
The number of Source (and Repair in case of recoding) Symbols in the current Encoding Window. This size may change over the time.

Payload Set:
The set of Source and Repair Symbols available (i.e., received or previously decoded) at the receiver and used during FEC decoding operations.

Decoding Window:
The set of Source Symbols (only) that are considered in the current linear system of a receiver, independently of the fact these Source Symbols have been received, decoded, or lost. The Decoding Window will typically change over time, as transmissions and decoding progress, and may be different for different receivers of a session where content is multicast or broadcast.

Decoding Window Size:
The number of Source Symbols (only) in the current Decoding Window. This size may change over time.

Rank of a Payload Set or Rank of the Linear System:
At a receiver, number of linearly independent members of a Payload Set, or equivalently 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" where decoding is possible or "partial rank" where only partial decoding is possible.

Seen Payload or Seen Symbol:
A Source Symbol is Seen when the receiver can compute a linear combination with this symbol and Source Symbols that are strictly more recent (i.e., with logically higher Encoding Symbol Identifiers). Otherwise, the Source Symbol is considered as "Unseen".

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 Matrix or Generator Matrix:
A matrix G that transforms the set of Input Symbols X into a set of Repair Symbols: Y = X * G. Defining a Generator Matrix is typical with Block Codes. The set of Input Symbols X can consist only of Source Symbols (e.g., with End-to-End Coding) or can consist of Source and Repair Symbols (e.g., with recoding in an intermediate node).

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 Repair Symbol through Linear Coding. The number of nonzero coefficients in the Coding Vector defines its density.

Finite Field, Galois Field, or Coding Field:
Finite Fields, used in Linear Codes, have the desired property of having all elements (except zero) invertible for the + and * operators, 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. The 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 or Coding 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.

Feedback:
Feedback information sent by a decoding node to a Coding Node (or from a receiver to a source 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 FEC decoding statistics, the list of available Source Packets received or decoded (acknowledgement), the list of lost Source Packets that should be retransmitted (negative acknowledgement), or a number of additional Repair Symbols needed to have a Full Rank Linear System.

4.3. Coding in Practice

This section discusses practical aspects. Indeed, a practical solution must specify the exact manner in which encoding and decoding are performed but also detail all the peripheral aspects, for instance, how an encoder informs a decoder about the parameters used to generate a certain Repair Packet (signaling).

(IETF) FEC Scheme:
A specification that defines a particular FEC code as well as the additional protocol aspects required to use this FEC code. In particular, the FEC Scheme defines in-band (e.g., information contained in Source and Repair Packet header or trailers) and out-of-band (e.g., information contained in an SDP description) signaling needed to synchronize encoders and decoders.

Payload Index or (IETF) Encoding Symbol Identifier (ESI):
An identifier of a Source or Repair Symbol. With Block Coding, each symbol of a given block is identified by a unique ESI value. With Sliding Window Coding, a continuous Source Flow and a limited field size to hold the ESI, wrapping to zero is unavoidable and the same integer value will be reused several times.

(IETF) FEC Payload ID:
Information that identifies the contents of a packet with respect to the FEC Scheme. The FEC Payload ID of a packet containing Source Symbol(s) is usually different from that of a packet containing Repair Symbol(s). The FEC Payload ID typically contains at least an ESI.

Coding Vector and Encoding Window Signaling:
With Sliding Window Codes, the FEC Payload ID of a Repair Packet contains information needed and sufficient to identify the Coding Vector and Coding Window. Concerning the Coding Vector, this may consist of a full list of Coding Coefficients (that may or may not be compressed), or a piece of information (e.g., a seed) that can be used to generate the list of Coding Coefficients thanks to a predefined algorithm known by encoders and decoders (e.g., a Pseudorandom Number Generator, or PRNG) or an ESI that points to a given entry in a Generator Matrix in case of a Block Code. Concerning the Coding Window, this may consist of the full list of ESI of symbols in the Coding Window (that may or may not be compressed) or the ESI of the first Source Symbol along with their number (assuming there is no gap).

5. IANA Considerations

This document has no IANA actions.

6. Security Considerations

This document introduces a recommended terminology for Network Coding and therefore does not contain any security considerations. This does not mean that Network Coding systems do not have any security implication.

7. Informative References

[RFC5052] Watson, M., Luby, M., and L. Vicisano, "Forward Error Correction (FEC) Building Block", RFC 5052, DOI 10.17487/RFC5052, August 2007, <https://www.rfc-editor.org/info/rfc5052>.

[RFC5740] Adamson, B., Bormann, C., Handley, M., and J. Macker, "NACK-Oriented Reliable Multicast (NORM) Transport Protocol", RFC 5740, DOI 10.17487/RFC5740, November 2009, <https://www.rfc-editor.org/info/rfc5740>.

[RFC5775] Luby, M., Watson, M., and L. Vicisano, "Asynchronous Layered Coding (ALC) Protocol Instantiation", RFC 5775, DOI 10.17487/RFC5775, April 2010, <https://www.rfc-editor.org/info/rfc5775>.

[RFC6363] 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>.

[RFC6726] Paila, T., Walsh, R., Luby, M., Roca, V., and R. Lehtonen, "FLUTE - File Delivery over Unidirectional Transport", RFC 6726, DOI 10.17487/RFC6726, November 2012, <https://www.rfc-editor.org/info/rfc6726>.

Authors' Addresses

Brian Adamson
NRL
United States of America

Email: brian.adamson@nrl.navy.mil

Cedric Adjih
INRIA
France

Email: cedric.adjih@inria.fr

Josu Bilbao
Ikerlan
Spain

Email: jbilbao@ikerlan.es

Victor Firoiu
BAE Systems
United States of America

Email: victor.firoiu@baesystems.com

Frank Fitzek
TU Dresden
Germany

Email: frank.fitzek@tu-dresden.de

Samah A. M. Ghanem
Independent

Email: samah.ghanem@gmail.com

Emmanuel Lochin
ISAE - Supaero
France

Email: emmanuel.lochin@isae-supaero.fr

Antonia Masucci
Orange
France

Email: antoniamaria.masucci@orange.com

Marie-Jose Montpetit
Independent
United States of America

Email: marie@mjmontpetit.com

Morten V. Pedersen
Aalborg University
Denmark

Email: mvp@es.aau.dk

Goiuri Peralta
Ikerlan
Spain

Email: gperalta@ikerlan.es

Vincent Roca (editor)
INRIA
France

Email: vincent.roca@inria.fr

Paresh Saxena
AnsuR Technologies
Norway

Email: paresh.saxena@ansur.es

Senthil Sivakumar
Cisco
United States of America

Email: ssenthil@cisco.com