RMT
Internet-Draft
Intended status: Informational
Expires: January 31, 2013

V. Roca
A. Roumy
INRIA
B. Sayadi
Alcatel-Lucent, Bell Labs

July 30, 2012

The Need for Extended Forward Erasure Correction (FEC) Schemes: Problem Position

draft-roca-rmt-extended-fec-problem-00

Abstract

This document discusses the limits of [RFC5052]-compliant traditional FEC schemes, and in particular for two use-cases that are not efficiently addressed, namely Unequal Erasure Protection (UEP) of subsets of an object and file bundle protection. This document defines the problem but not the potential solutions. This is left to companion documents.

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 January 31, 2013.

Table of Contents

1. Introduction: traditional FEC Schemes, as per [RFC5052]

The use of Forward Error Correction (FEC) codes is a classic solution to improve the reliability of unicast, multicast and broadcast Content Delivery Protocols (CDP) and applications [RFC3453]. The [RFC5052] document describes a generic framework to use FEC schemes with objects (e.g., files) delivery applications based on the ALC [RFC5775] (e.g. FLUTE [FLUTE]) and NORM [RFC5740] reliable multicast transport protocols.

More specifically, the [RFC5053]/[RFC6330] (Raptor/RaptorQ) and [RFC5170] (LDPC-Staircase) FEC schemes introduce erasure codes based on sparse parity check matrices for object delivery protocols like ALC and NORM. Similarly, the [RFC5510] document introduces Reed-Solomon codes based on Vandermonde matrices for these object delivery protocols. Finally [RFC5445] introduces the Compact No-Code FEC Scheme that is not attached to any FEC code but behaves as if it was the case. This No-Code FEC Scheme can be very useful when an object is small enough to be sent within a single source symbol, since robustness can be achieved by repeating it, relying on a No-Code FEC Scheme for signaling considerations.

Let us consider a source object, submitted by the CDP or application, that needs to be FEC protected by one of the FEC schemes mentioned before. This FEC scheme, the underlying FEC code or the associated codec (i.e. a concrete implementation of the FEC code) defines a maximum number of source symbol that can be considered together for FEC encoding. If the object size, in terms of number of source symbols, is superior to this maximum size, this object is partitioned into multiple smaller blocks, of approximately equal size ([RFC5052], section 9.1, "Block Partitioning Algorithm"). FEC encoding is then applied independently to each block, and repair symbols are produced. The same process is then applied to each source object submitted by the CDP or application.

We see that with these FEC schemes, FEC encoding is directly applied on the source objects. Although natural, this approach leads to serious limits as we will detail. The goal of the present document is to discuss these limits. However this document does not define any solution to actually bypass these limits. This is the role of companion documents, like [GOE-RS] and [GOE-LDPC], that decouples FEC protection from the natural object boundaries, and [UOD-RaptorQ] that relies on a specific packetization technique.

2. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT",

"SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

2.1. Definitions, Notations and Abbreviations

2.1.1. Definitions

This document uses the following terms and definitions. Some of them are FEC scheme specific and are in line with [RFC5052]:

Source Packet: a data packet containing only source symbols, that is sent over the packet erasure channel. Most of the time a source packet will contain a single source symbol.

Repair Packet: a data packet containing only repair symbols, that is sent over the packet erasure channel. Most of the time a repair packet will contain a single repair symbol.

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.

Object: the object (e.g., file) submitted to the CDP by the user.

Source symbol: unit of data used during the encoding process. In this specification, there is always one source symbol per ADU.

Encoding symbol: unit of data generated by the encoding process. With systematic codes, source symbols are part of the encoding symbols.

Repair symbol: encoding symbol that is not a source symbol.

Source block: a block of k source symbols that are considered together for the encoding.

2.1.2. Notations

This document uses the following notations:

k denotes the number of source symbols in a source block.

n denotes the number of encoding symbols generated for a source block.

CR denotes the "code rate", i.e., the k/n ratio.

2.1.3. Abbreviations

This document uses the following abbreviations:

FEC stands for Forward Error (or Erasure) Correction code.

LDPC stands for Low Density Parity Check.

RS stands for Reed-Solomon.

UEP stands for Unequal Erasure Protection.

3. Use-Cases not Correctly Addressed by [RFC5052]-compliant FEC Schemes

We now list two major use-cases, namely UEP and file bundle protection. Those are the two use-cases that any Extended FEC scheme MUST support. Further, optional to support, use-cases may be added.

3.1. Major use-Case 1: Unequal Protection of Parts of an Object

This first use-case defines a situation where some subsets of an object deserve a higher erasure protection than the others. The traditional FEC Schemes, as per [RFC5052], lead to apply the same FEC encoding to all the blocks of a given object, i.e., the whole object is encoded using the same FEC scheme, with the same target code rate, resulting in an equivalent protection.

There is no way to bypass this limit within the traditional [RFC5052]-compliant FEC Schemes. However the CDP or the application may decide to split the original object into say two sub-objects, one per important class, and to submit each sub-object separately to the FEC Scheme. Modifying the CDP specifications, e.g., FLUTE/ALC, to do that is clearly non appropriate: some of these CDP have been widely deployed and modifying deeply the specifications to address this new requirement is not desired. Modifying the application to do that does not require any modification of the specifications, but requires extra logic to manage scattering/gathering within the application. Additionally, each application has to be modified: there is no factorization of the functionality within a common underlying layer, in our case the FEC Scheme.

TODO: concrete example where UEP improves object delivery.

3.2. Major use-Case 2: File Bundle Protection

This second use-case defines a situation where a set of objects need to be reliably transferred within a given CDP session with the same amount of protection. All the receivers are also supposed to recover the full set of objects. With traditional FEC schemes, efficiency problems arise when this set is large and the objects are small (i.e., composed of a few source symbols each). For instance, if we consider the case where each object is composed of a single source symbol and a code rate one half is applied, each object contributes to two encoding symbols (in practice we use a No-Code FEC scheme and repeat each source symbol twice). If the two encoding symbols of an object are lost, a receiver will not be able to recover these erasures.

With traditional FEC schemes, efficiency problems also arise when objects of largely varying sizes need to be transferred within the same session. Indeed, large objects benefit from a higher number of encoding symbols than small ones (since the code rate is kept constant, the higher the number of source symbols, k, the higher the number of encoding symbols, n = k / CR). Therefore large objects will be more robust in front of erasure bursts of a given length than small objects.

In these two examples, interleaving techniques can be used to reduce the probability of losing many symbols of a given object, due to erasure bursts. However, if long packet interleaving strategies can reduce the impacts of packet erasure bursts, they do not solve the fundamental problem.

There is no way to bypass this limit within the traditional [RFC5052]-compliant FEC Schemes. However the CDP or the application may decide to create an archive that contains all the objects and to submit this archive in place of each individual object. Here also, modifying the CDP specifications, e.g., FLUTE/ALC, to do that is clearly non appropriate (same reasons). Similarly, modifying the application to do that requires extra logic, and there is no factorization of the functionality within a common underlying layer.

TODO: concrete example where UEP improves object delivery.

4. Requirements and Good Properties for an Extended FEC Scheme

This section defines requirements (mandatory to support) and properties for any Extended FEC scheme.

4.1. Mandatory to Support Functional Requirements

The following are functional requirements that an Extended FEC scheme MUST provide:

- the use of an Extended FEC scheme MUST NOT require conflicting modifications of the CDP specifications (when applicable). Of course supporting an additional FEC scheme will require some extra logic within the CDP, but the modifications are limited to the implementations of the CDP, without any conflicting impact on the signaling mechanism (or any other mechanism) defined by the CDP.
- it MUST be possible, within a single CDP session, to include objects protected with one or several Extended FEC schemes and at the same time objects protected with one or several traditional FEC schemes. The use of Extended and traditional FEC Schemes MUST not be exclusive with one another.

- an Extended FEC scheme MUST support the use-cases 1 and 2 detailed in Section 3. These two use-cases form the minimum set of functionalities required for any Extended FEC scheme.

4.2. Additional Properties

Several additional properties may be considered. Some of them concern performance. For instance what is the performance in terms of unequal protection (e.g., is the protection differentiation actually achieved?), erasure recovery (e.g., does the Extended FEC scheme perform well in recovering from lost packets?), predictability (e.g., is the Extended FEC scheme behavior predictable or not?).

Some of them concern algorithm agility. For instance does the distribution of objects (i.e., total number and individual size) in a bundle impact performance with the file bundle protection use-case? Or does the Extended FEC scheme enable priority classes to be freely defined, with possible overlapping of data chunks (e.g., can subset1, of highest priority, be either independent from subset2, of lower priority, or at the opposite be part of subset2 so that repair symbols for subset2 also help in recovering erased symbols from subset1?) and gaps (e.g., a very low importance subset of an object might not be FEC encoded at all). Another aspect is the possibility to use different kinds of underlying FEC code in an Extended scheme. Finally, backward compatibility with receivers that do not support the Extended FEC scheme can also be useful in situations where the set of receivers is largely heterogeneous: backward compatibility could enable a legacy receiver to benefit from source symbols even if repair symbols will be disposed, by lack of the required logic to process them.

This list is far from exhaustive and is only for informative purposes (e.g., to compare the pros/cons of several Extended FEC schemes).

5. Security Considerations

TBD

6. Acknowledgments

TBD

7. References

7.1. Normative References

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

7.2. Informative References

[RFC3453] Luby, M., Vicisano, L., Gemmell, J., Rizzo, L., Handley, M., and J. Crowcroft, "The Use of Forward Error Correction (FEC) in Reliable Multicast", RFC 3453, December 2002.

[RFC5445] Watson, M., "Basic Forward Error Correction (FEC) Schemes", RFC 5445, March 2009.

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

[RFC5510] Lacan, J., Roca, V., Peltotalo, J., and S. Peltotalo, "Reed-Solomon Forward Error Correction (FEC) Schemes", RFC 5510, April 2009.

[RFC5170] Roca, V., Neumann, C., and D. Furodet, "Low Density Parity Check (LDPC) Forward Error Correction", RFC 5170, June 2008.

[RFC5053] Luby, M., Shokrollahi, A., Watson, M., and T. Stockhammer, "Raptor Forward Error Correction Scheme", RFC 5053, June 2007.

[RFC6330] Luby, M., Shokrollahi, A., Watson, M., Stockhammer, T., and L. Minder, "RaptorQ Forward Error Correction Scheme for Object Delivery", RFC 6330, August 2011.

[RFC5740] Adamson, B., Bormann, C., Handley, M., and J. Macker, "NACK-Oriented Reliable Multicast (NORM) Transport Protocol", RFC 5740, November 2009.

[RFC5775] Luby, M., Watson, M., and L. Vicisano, "Asynchronous Layered Coding (ALC) Protocol Instantiation", RFC 5775, April 2010.

[FLUTE] Paila, T., Walsh, R., Luby, M., Roca, V., and R. Lehtonen, "FLUTE - File Delivery over Unidirectional Transport", Work in Progress, June 2012.

[GOE-RS] Roca, V., Roumy, A., and S. Bessem, "The Generalized Object Encoding (GOE) Approach for the Forward Erasure Correction (FEC) Protection of Objects and its Application

to Reed-Solomon Codes over GF(2^^8)", Work in progress draft-roca-rmt-goe-fec, July 2012.

[GOE-LDPC]
Roca, V., Roumy, A., and S. Bessem, "The Generalized Object Encoding (GOE) LDPC-Staircase FEC Scheme", Work in progress draft-roca-rmt-goe-fec, July 2012.

[UOD-RaptorQ]
Luby, M. and T. Stockhammer, "Universal Object Delivery using RaptorQ", Work in progress draft-luby-uod-raptorq, September 2011.

Authors' Addresses

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

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

Aline Roumy
INRIA
Campus Universitaire de Beaulieu
RENNES Cedex 35042
France

Email: aline.roumy@inria.fr
URI: http://www.irisa.fr/prive/Aline.Roumy/

Bessem Sayadi
Alcatel-Lucent, Bell Labs
France

Email: bessem.sayadi@alcatel-lucent.com