NWCRG N. Kuhn
Internet-Draft CNES
Intended status: Informational E. Lochin
Expires: January 10, 2021 ISAE-SUPAERO
F. Michel
UCLouvain
M. Welzl
University of Oslo
July 9, 2020

Coding and congestion control in transport
draft-irtf-nwcrg-coding-and-congestion-03

Abstract

FEC coding is a reliability mechanism that is distinct and separate from the loss detection of congestion controls. Using FEC coding can be a useful way to deal with transfer tail losses or with networks having non-congestion losses. However, FEC coding mechanisms should not hide congestion signals. This memo offers a discussion of how FEC coding and congestion control can coexist. Another objective is to encourage the research community to also consider congestion control aspects when proposing and comparing FEC coding solutions in communication systems.

This document is the product of the Coding for Efficient Network Communications Research Group (NWCRG). The scope of the document is end-to-end communications: FEC coding for tunnels is out-of-the scope of the document.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

There are cases where deploying FEC coding improves the performance of a transmission. As an example, it may take time for the sender to detect transfer tail losses (losses that occur at the end of a transfer, where e.g. TCP obtains no more ACKs to quickly repair the loss via retransmission). This would improve the experience of applications using short flows. Another example are networks where non-congestion losses are persistent and prevent a sender from exploiting the link capacity.

Coding is a reliability mechanism that is distinct and separate from the loss detection of congestion controls. [RFC5681] defines TCP as a loss-based congestion control; because FEC coding repairs such

losses, blindly applying it may easily lead to an implementation that also hides a congestion signal to the sender. It is important to ensure that such information hiding does not occur.

FEC coding and congestion control can be seen as two separate channels. In practice, implementations may mix the signals that are exchanged on these channels. This memo offers a discussion of how FEC coding and congestion control can coexist. Another objective is to encourage the research community to also consider congestion control aspects when proposing and comparing FEC coding solutions in communication systems. This being said, this document does not aim at proposing guidelines for characterizing FEC coding solutions: comparing FEC Schemes without considering congestion control can be relevant if the goal is to compare those schemes only.

The proposed document considers FEC coding at the transport or application layer for an end-to-end unicast data transfer. The typical application scenario that is considered in the current version of the document is a client browsing the web or watching a live video. This memo may be extended to cases with multiple paths.

This document 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. The document follows the terminology proposed in the taxonomy document [RFC8406].

2. Separate channels, separate entities

Figure 1 presents the notations that will be used in this document and introduces the Congestion Control (CC) and Forward Erasure Correction (FEC) channels. The Congestion Control channel carries source packets from a sender to a receiver, and packets signaling information about the network (number of packets received vs. lost, ECN marks, etc.) from the receiver to the sender. The Forward Erasure Correction channel carries repair packets (from the sender to the receiver) and potential information signaling which packets have been repaired (from the receiver to the sender). It is worth pointing out that there are cases where these channels are not separated.

```
    Sender                                Receiver

   +------+                               +------+
   |      | -----    source packets  ---->|      |
   |  CC  |                               |  CC  |
   |      | <---  network information  ---|      |
   +------+                               +------+

   +------+                               +------+
   |      | -----    repair packets  ---->|      |
   | FEC  |                               | FEC  |
   |      | <--- info: repaired packets --|      |
   +------+                               +------+
```

Figure 1: Notations and separate channels

Inside a host, the CC and FEC entities can be regarded as conceptually separate:

```
     |            ^             |             ^
     | source     | coding      |packets      | sending
     | packets    | rate        |requirements | rate (or
     v            |             v             | window)
   +---------------+source     +-----------------+
   |    FEC        |and/or     |    CC           |
   |               |repair     |                 |source
   |               |packets    |                 |packets
   +---------------+==>        +-----------------+==>
     ^                                       ^
     | signaling about                       | network
     | losses and/or                         | information
     | repaired packets
```

Figure 2: Separate entities (sender-side)

```
     |                                |
     | source and/or                  | packets
     | repair packets                 |
     v                                v
   +---------------+              +-----------------+
   |    FEC        |signaling     |    CC           |
   |               |repaired      |                 |network
   |               |packets       |                 |information
   +---------------+==>           +-----------------+==>
                    repaired packets
```

Figure 3: Separate entities (client-side)

Figure 2 and Figure 3 provide more details than Figure 1. Some elements are introduced:

- 'network information' (input control plane for the transport including CC): refers not only to the network information that is explicitly signaled from the receiver, but all the information a congestion control obtains from a network (e.g. TCP can estimate the latency and the available capacity at the bottleneck).

- 'requirements' (input control plane for the transport including CC): refers to application requirements such as upper/lower rate bounds, periods of quiescence, or a priority.

- 'sending rate (or window)' (output control plane for the transport including CC): refers to the rate at which a congestion control decides to transmit packets, based on 'network information'.

- 'signaling repaired packets' (input control plane for the FEC): refers to the information a FEC sender can obtain from a FEC receiver about the performance of the FEC solution as seen from the receiver.

- 'coding rate' (output control plane for the FEC): refers to the coding rate that is used by the FEC solution.

- 'source and/or repair packets' (data plane for both the FEC and the CC): refers to the packets that are transmitted. There can only be source packets (if the coding rate is 0), repair packets (if the solution decides not to send the original source packets) or a mix of both.

The inputs to FEC (packets to work upon, and signaling from the receiver about losses and/or repaired packets) are distinct from the inputs to CC. The latter calculates a sending rate or window from network information, and it takes the packet to send as input, sometimes along with application requirements such as upper/lower rate bounds, periods of quiescence, or a priority. It is not clear that the ACK signals feeding into a congestion control algorithm are useful to FEC in their raw form, and vice versa - information about repaired blocks may be quite irrelevant to a CC algorithm.

The choice of the adequate transport layer may be related to application requirements:

- In the case of an unreliable data transfer, the transport layer may implement a non-reliable transport layer (e.g. UDP or DCCP [RFC4340] or a partially reliable transport protocol such as SCTP with partial reliability [RFC3758]). Depending on the amount of

redundancy and network conditions, there could be cases where it ends up being impossible to carry traffic.

- o In the case of a reliable data transfer, the transport layer may implement a retransmission mechanism to guarantee the reliability of the file transfer (e.g. TCP). Depending on how the FEC and CC functions are scheduled (FEC above CC, FEC in CC, FEC below CC), the impact of a reliable transport on the FEC reliability mechanisms is different.

3. FEC above the transport

3.1. Flowchart

```
    | source                               ^ source
    | packets                              | packets
    v                                      |
   +-------------+                      +-------------+
   |FEC          |             signaling|FEC          |
   |             |              repaired|             |
   |             |               packets|             |
   |             |                   <==|             |
   +-------------+                      +-------------+
    | source  ^                            ^ source
    | and/or  | sending                    | and/or
    | repair  | rate                       | repair
    | packets | (or window)                | packets
    v         |                            |
   +-------------+                      +-------------+
   |Transport    |source         network|Transport    |
   |(incl. CC)   |and/or     information|             |
   |             |repair             <==|             |
   |             |packets               |             |
   +-------------+==>                   +-------------+

        SERVER                                 CLIENT
```

Figure 4: FEC above the transport

Figure 4 present an architecture where FEC is on top of the transport.

3.2. Discussion

While CC is in principle independent of other transport functions such as reliability, we note that CC is often embedded in reliable transfer protocols (e.g. TCP). This approach may require that the transport protocol does not implement a fully reliable data transfer

service (e.g., based on lost packet retransmission), otherwise this could lead to spurious retransmissions. UDP is an example of a protocol for which this approach is relevant.

The advantage of this approach is that the FEC does not contribute to adding congestion in the network. When a congestion control is implementedd at the transport layer, the repair packets are sent within what the congestion window allows.

4. FEC within the transport

4.1. Flowchart

```
    | source  | sending                    ^ source
    | packets | rate                       | packets
    v         v                            |
   +------------+                      +------------+
   | Transport  |                      | Transport  |
   |            |                      |            |
   | +---+ +--+ |             signaling| +---+ +--+ |
   | |FEC| |CC| |              repaired| |FEC| |CC| |
   | +---+ +--+ |source         packets| +---+ +--+ |
   |            |and/or             <==|            |
   |            |repair         network|            |
   |            |packets    information|            |
   +------------+ ==>               <==+------------+

       SERVER                              CLIENT
```

Figure 5: FEC in the transport

Figure 5 presents an architecture where FEC is within the transport. The repair packets are sent within what the congestion window allows, such as in [CTCP].

4.2. Discussion

The advantage of this approach is that it can enable conjoint optimization between the CC and the FEC. Moreover, the transmission of repair packets does not add congestion in potentially congested networks but helps repair lost packets (such as tail losses).

The drawback of this approach is that it may not result in much gains as opposed to classical retransmission mechanisms and it costs bandwidth that could have been exploited to transmit source packets. The coding ratio needs to be carefully designed.

Examples of the solution would be sending repair packets when there is no more data to transmit or preferably send repair packets instead of the following packets in the send buffer.

5. FEC below the transport

5.1. Flowchart

```
    | source  | sending rate               ^ source
    | packets | (or window)                | packets
    v         v                            |
   +--------------+                      +--------------+
   |Transport     |               network|Transport     |
   |(including CC)|           information|              |
   |              |                   <==|              |
   +--------------+                      +--------------+
    | source packets                       ^ source packets
    v                                      |
   +--------------+                      +--------------+
   | FEC          |source                |  FEC         |
   |              |and/or       signaling|              |
   |              |repair        repaired|              |
   |              |packets        packets|              |
   |              |==>                <==|              |
   +--------------+                      +--------------+

        SERVER                                 CLIENT
```

Figure 6: FEC below the transport

Figure 6 presents an architecture where FEC is below the transport. The repair packets are sent on top of what is allowed by the congestion control.

5.2. Discussion

The advantage of this approach is that it can result in performance gains when there are persistent transmission losses along the path.

The drawback of this approach is that it can induce congestion in already congested networks. The coding ratio needs to be carefully designed.

Examples of the solution could be adding a given percentage of the congestion window as supplementary packets or sending a given amount of repair packets at a given rate. The redundancy flow can be decorrelated from the congestion control that manages source packets: a secondary congestion control could be introduced underneath the FEC

layer. The separate congestion control mechanisms could be made to work together while adhering to priorities, as in coupled congestion control for RTP media [I-D.ietf-rmcat-coupled-cc]. Another possibility would be to exploit a lower than best-effort congestion control [RFC6297] for repair packets.

6. Acknowledgements

Many thanks to Spencer Dawkins, Dave Oran, Carsten Bormann, Vincent Roca and Marie-Jose Montpetit for their useful comments that helped improve the document.

7. IANA Considerations

This memo includes no request to IANA.

8. Security Considerations

FEC and CC schemes can contribute to DoS attacks. This is not specific to this document.

In case of FEC below the transport, the aggregate rate of source and repair packets may exceed the rate at which a congestion control mechanism allows an application to send. This could result in an application obtaining more than its fair share of the network capacity.

9. Informative References

[CTCP] Kim (et al.), M., "Network Coded TCP (CTCP)", arXiv 1212.2291v3, 2013.

[I-D.ietf-rmcat-coupled-cc] Islam, S., Welzl, M., and S. Gjessing, "Coupled congestion control for RTP media", draft-ietf-rmcat-coupled-cc-09 (work in progress), August 2019.

[RFC3758] Stewart, R., Ramalho, M., Xie, Q., Tuexen, M., and P. Conrad, "Stream Control Transmission Protocol (SCTP) Partial Reliability Extension", RFC 3758, DOI 10.17487/RFC3758, May 2004, <https://www.rfc-editor.org/info/rfc3758>.

[RFC4340] Kohler, E., Handley, M., and S. Floyd, "Datagram Congestion Control Protocol (DCCP)", RFC 4340, DOI 10.17487/RFC4340, March 2006, <https://www.rfc-editor.org/info/rfc4340>.

[RFC5681] Allman, M., Paxson, V., and E. Blanton, "TCP Congestion Control", RFC 5681, DOI 10.17487/RFC5681, September 2009, <https://www.rfc-editor.org/info/rfc5681>.

[RFC6297] Welzl, M. and D. Ros, "A Survey of Lower-than-Best-Effort Transport Protocols", RFC 6297, DOI 10.17487/RFC6297, June 2011, <https://www.rfc-editor.org/info/rfc6297>.

[RFC8406] 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>.

Authors' Addresses

Nicolas Kuhn
CNES

Email: nicolas.kuhn@cnes.fr

Emmanuel Lochin
ISAE-SUPAERO

Email: emmanuel.lochin@isae-supaero.fr

Francois Michel
UCLouvain

Email: francois.michel@uclouvain.be

Michael Welzl
University of Oslo

Email: michawe@ifi.uio.no