NWCRG   N. Kuhn
Internet-Draft   CNES
Intended status: Informational   E. Lochin
Expires: May 3, 2021   ENAC
                                                            F. Michel
                                                            UCLouvain
                                                             M. Welzl
                                                    University of Oslo
                                                     October 30, 2020

Coding and congestion control in transport
draft-irtf-nwcrg-coding-and-congestion-04

Abstract

   Forward Erasure Correction (FEC) is a reliability mechanism that is
   distinct and separate from the retransmission logic in reliable
   transfer protocols such as TCP. Using FEC coding can help 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 repair the loss via
   retransmission quickly). Allowing the receiver to recover such
   losses instead of having to rely on a retransmission could 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; since FEC coding repairs such
   losses, blindly applying it may easily lead to an implementation that
   also hides a congestion signal from 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 also to consider congestion
   control aspects when proposing and comparing FEC coding solutions in
   communication systems. This document does not aim at proposing
   guidelines for characterizing FEC coding solutions.

   The proposed document considers an end-to-end unicast data transfer
   with FEC coding at the application (above the transport), within the
   transport or directly below the transport. The typical application
   scenario 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 symbols (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 symbols  ---->|      |
  | FEC  |                              | FEC  |
  |      | <--- info: repaired symbols --|      |
  +------+                              +------+
```
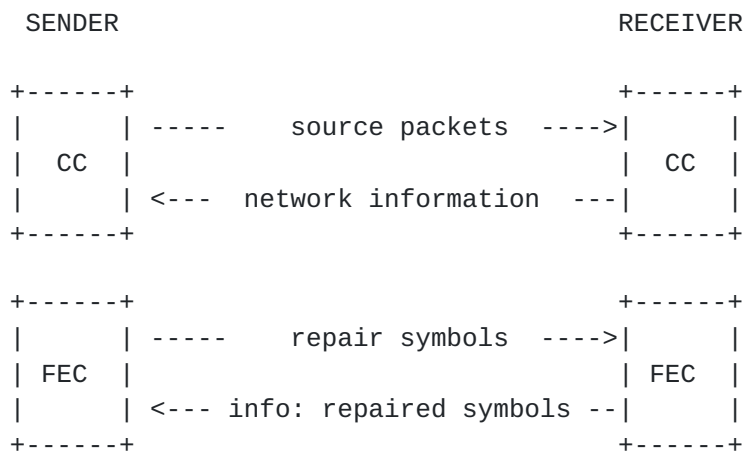
               Figure 1: Notations and separate channels

   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
 |               |symbols   |                 |packets
 +---------------+==>       +-----------------+==>
   ^                                 ^
   | signaling about                 | network
   | losses and/or                   | information
   | repaired symbols
```
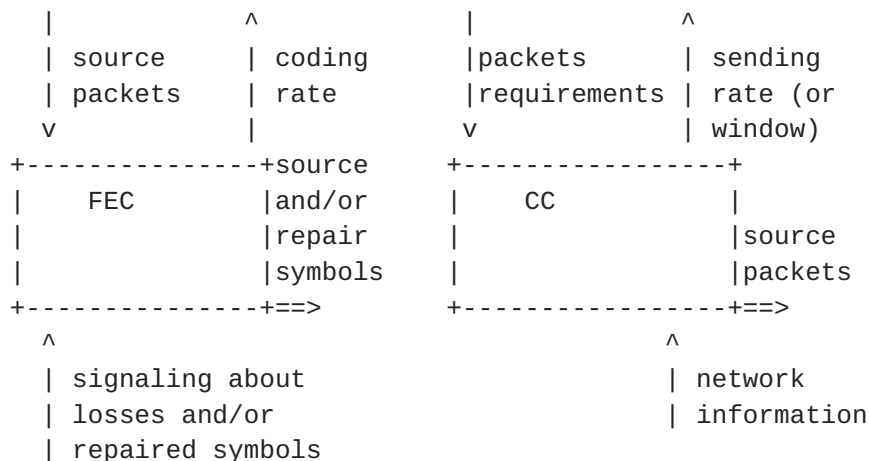
               Figure 2: Separate entities (sender-side)

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

                Figure 3: Separate entities (receiver-side)

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

   o '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).

   o '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.

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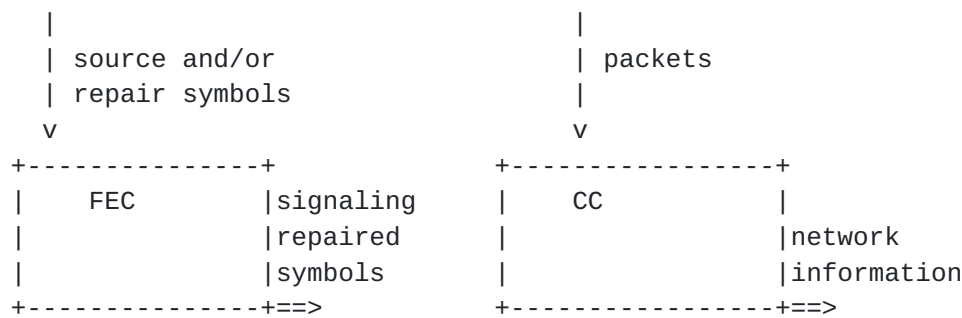   including CC): refers to the rate at which a congestion control
      decides to transmit packets based on 'network information'.

   o 'signaling repaired symbols' (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 by the
      receiver.

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

   o 'source and/or repair symbols' (data plane for both the FEC and
      the CC): refers to the data that is transmitted. The sender can
      decide to send source symbols only (meaning that the coding rate
      is 0), repair symbols only (if the solution decides not to send
      the original source packets) or a mix of both.

   The inputs to FEC (incoming data packets without repair symbols, and
   signaling from the receiver about losses and/or repaired symbols) 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:

o In the case of an unreliable data transfer, the transport layer
   may provide a non-reliable transport service (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
   becomes 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 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|             |
 |             |               symbols|             |
 |             |                   <==|             |
 +-------------+                          +-------------+
    | source   ^                               ^ source
    | and/or   | sending                       | and/or
    | repair   | rate                          | repair
    | symbols  | (or window)                   | symbols
    v          |                               |
 +-------------+                          +-------------+
 |Transport    |source        network|Transport    |
 |(incl. CC)   |and/or      information|             |
 |             |repair              <==|             |
 |             |packets                |             |
 +-------------+==>                       +-------------+

       SENDER                              RECEIVER
```
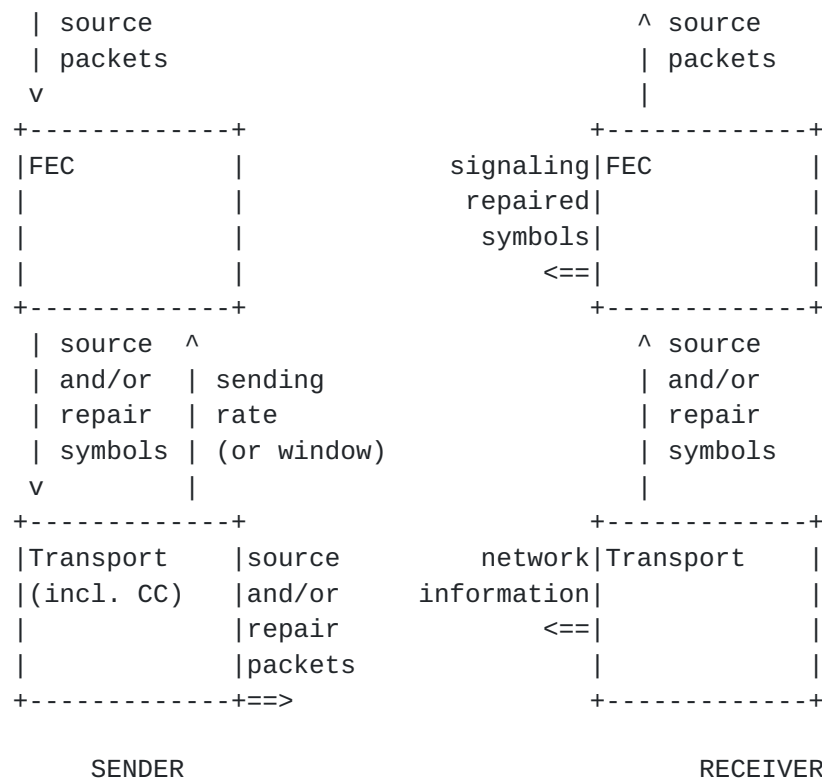
                  Figure 4: FEC above the transport

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

3.2. Discussion

   The advantage of this approach is that the FEC overhead does not
   contribute to congestion in the network. When congestion control is
   implemented at the transport layer, the repair symbols are sent
   following the congestion window. This approach can result in
   improved quality of experience for latency sensitive applications
   such as VoIP.

   This approach requires that the transport protocol does not implement
   a fully reliable data transfer service (e.g., based on lost packet
   retransmission). UDP is an example of a protocol for which this
   approach is relevant. For reliable transfers, coding usage does not
   guarantee better performance and would mainly reduce goodput for
   large file transfers.

   This discussion section is extended in Section 6.

[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         symbols| +---+ +--+ |
  |           |and/or            <==|             |
  |           |repair         network|             |
  |           |packets    information|             |
  +------------+ ==>              <==+------------+

       SENDER                          RECEIVER
```
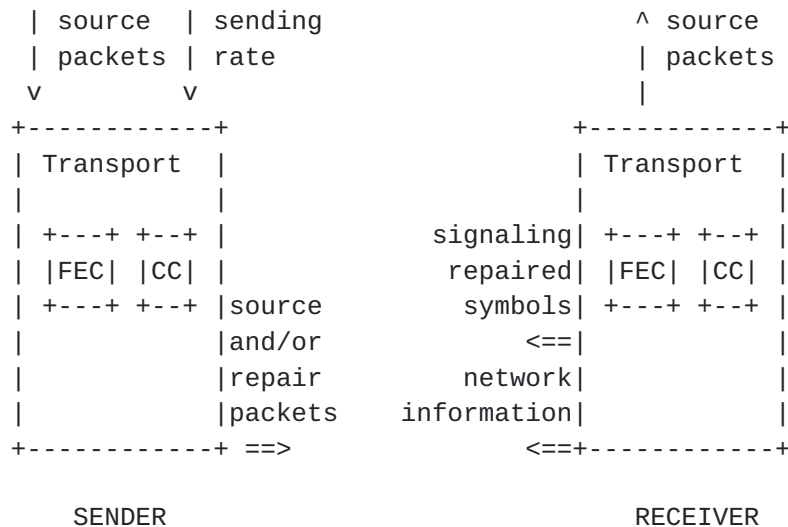
                     Figure 5: FEC in the transport

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

[4.2](#). **Discussion**

   The advantage of this approach allows a joint optimization between
   the CC and the FEC. Moreover, the transmission of repair symbols
   does not add congestion in potentially congested networks but helps
   repair lost packets (such as tail losses).

   For reliable transfers, including redundancy reduces goodput for
   large file transfers but the amount of repair symbols can be adapted,
   e.g. depending on the congestion window size. There is a trade-off
   between the cost in capacity used to transmit source packets and the
   benefits brought out by transmitting repair symbols (e.g. unlocking
   the receive buffer if this is limiting). The coding ratio needs to
   be carefully designed. For small files, sending repair symbols when
   there is no more data to transmit could help to reduce the transfer
   time. In general, sending repair symbols could avoid a silent period
   between the transmission of the last packet in the send buffer and 1)
   firing the retransmission of lost packets, or 2) the transmission of
   new packets.

   This discussion section is extended in Section 6.

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|             |
 |              |symbols    symbols|              |
 |              |==>           <==|               |
 +--------------+                  +--------------+

       SENDER                          RECEIVER
```
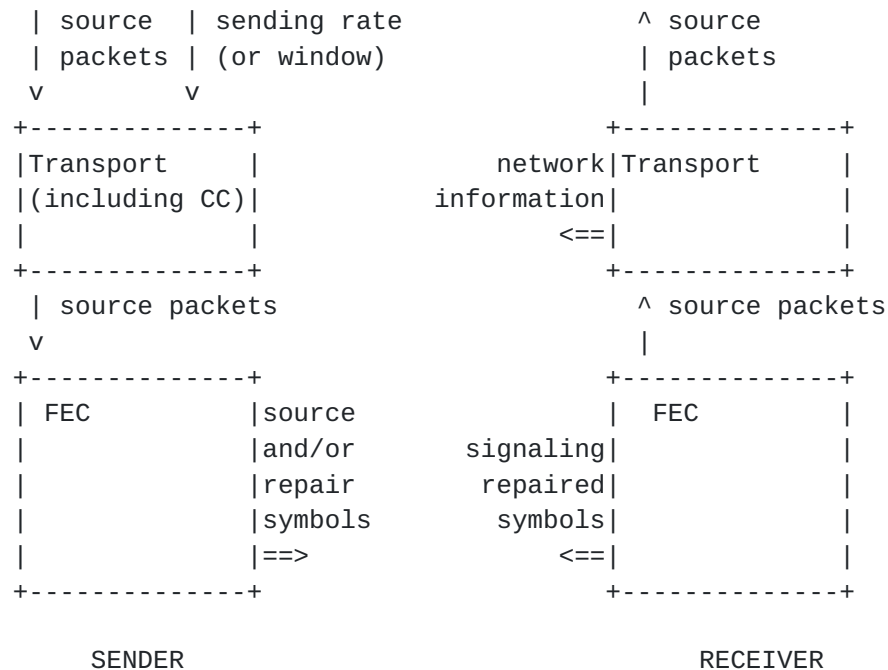
Figure 6: FEC below the transport

Figure 6 presents an architecture where FEC is applied end-to-end
below the transport layer, but above the link layer. Note that it is
common to apply FEC at the link layer, in which it contributes to the
total capacity that a link exposes to upper layers. This application
of FEC is out of scope of this document. In the scenario considered
here, the repair symbols are sent on top of what is allowed by the
congestion control.

5.2. Discussion

In this case, including redundancy adds congestion without reducing
goodput but leads to potential fairness issues. The effective
bitrate is indeed higher than the CC's computed fair share due to the
sending of repair symbols and the losses are hidden from the
transport. This may cause a problem for loss-based congestion
detection, but it is not a problem for delay-based congestion
detection.

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 symbols or sending a given amount of repair symbols at a given rate. The redundancy flow can be decorrelated from the congestion control that manages source packets: a separate congestion control entity could be introduced to manage the amount of repaired packets to transmit on the FEC channel. The separate congestion control instances could be made to work together while adhering to priorities, as in coupled congestion control for RTP media [RFC8699] in case all traffic can be assumed to take the same path, or otherwise with a multipath congestion window coupling mechanism as in Multipath TCP [RFC6356]. Another possibility would be to exploit a lower than best-effort congestion control [RFC6297] for repair symbols.

This discussion section is extended in Section 6.

6. Fairness, redundancy rate and congestion signals

The objective of this section is to further detail some aspects that have been expressed in previous discussion subsections.

6.1. Fairness, a policy concern

The contract between the client and the operator may guarantee a minimum data-rate (e.g. mobile networks). However, for residential accesses, the data-rate can be guaranteed for the customer premises equipment, but not necessarily for the client. The quality of service that guarantees fairness between the different clients can be seen as a policy concern [I-D.briscoe-tsvarea-fair].

While flow level fairness does not embody the actual application level fairness, the share of available capacity between single flows can help assess when one flow starves the other. Clients may share a bottleneck that may not be ruled by a quality of service mechanism, e.g. in case of:

o a mobile network client running several applications;

o two clients on a residential access.

This document considers fairness as an index to quantify the impact of the addition of coded flows on non-coded flows when they share the same bottleneck. This document does not aim at contributing to the

definition of fairness at a wider scale. This document assumes that
the non-coded flows respond to congestion signals from the network.

6.2. Fairness and impact on non-coded flows

6.2.1. FEC above the transport

The addition of coding within the flow does not impact on the
interaction between coded and non-coded flows. This interaction
would mainly depend on the congestion controls embedded in each host.

6.2.2. FEC within the transport

The addition of coding within the flow may impact the congestion
control mechanism and hide congestion losses. Specific interaction
between congestion controls and coding schemes can be proposed (see
Section 6.3, Section 6.4 and Section 6.5). If no specific
interaction is introduced, the coding scheme may hide congestion
losses from the congestion controller and the description of
Section 6.2.3 may apply.

6.2.3. FEC below the transport

In this case, the coding scheme may hide congestion losses from the
congestion controller. There are cases where this can drastically
reduce the goodput of non-coded flows. Depending on the congestion
control, it may be possible to signal to the congestion control
mechanism that there was congestion (loss) even when a packet has
been recovered, e.g. using ECN, to reduce the impact on the non-coded
flows (see Section 6.3.3 and [TENTET]).

6.3. Congestion control and recovered symbols

The objective of this subsection is to describe potential
interactions between the congestion control and the recovered
symbols.

6.3.1. FEC above the transport

The congestion control may not be able to differentiate repair
symbols from actual source packets. The relevance of adding coding
at the application layer is related to the needs of the application.
For real-time applications, this approach may reduce the number of
retransmission. The usage of a non-reliable transport is more
adequate in this case.

6.3.2. FEC within the transport

If the two FEC and CC channels are decoupled, the endpoint may
exploit different protocols for each channel. The channels may be
coupled and one single protocol may be exploited. In both cases, the
receiver can differentiate source packets and repair symbols. The
receiver may indicate both the number of source packets received and
repair symbols that were actually useful in the recovery process of
packets.

6.3.3. FEC below the transport

The congestion control may not know what is going on in the network
underneath and whether a coding scheme is introduced or not. The
congestion control may behave as if no coding scheme is introduced.
The only way for a coding channel to indicate that symbols have been
recovered is to exploit existing signaling that is understood by the
congestion control mechanism. An example would be to indicate to a
TCP sender that a packet has been recovered (i.e., congestion has
occurred), by using ECN signaling [TENTET].

6.4. Interactions between congestion control and coding rates

This section discusses to what extent the interaction between the
congestion control and the coding rates is possible.

6.4.1. FEC above the transport

The coding rate applied at the application layer mainly depends on
the available capacity given by the congestion control underneath.
Adapting the coding rate to the minimum required data rate of the
application may reduce packet losses and improve the quality of
experience.

6.4.2. FEC within the transport

In this case, there is an important flexibility in the trade-off,
inherent to the use of coding, between (1) reducing goodput when
useless repair symbols are transmitted and (2) helping to recover
sooner from transmission and congestion losses. As explained in
Section 6.3.2, the receiver may indicate to the sender the number of
packets that have been received or recovered. The sender may exploit
this information to tune the coding ratio. As one example of
flexibility of this case, coupling an increased transmission rate
with an increasing or decreasing coding rate could be envisioned. A
server may use an increasing coding rate as a probe of the channel
capacity and adapt the congestion control transmission rate.

6.4.3. FEC below the transport

In this case, the coding rate can be tuned depending on the number of
recovered symbols and the rate at which the sender transmits data.
The coding scheme is not aware of the congestion control
implementation, making it hard for the coding scheme to apply the
relevant coding rate.

6.5. On the useless repair symbols

There are cases where useless repair symbols may be transmitted.
These impact on the network load and may reduce the goodput of the
flow without concrete gains.

6.5.1. FEC above the transport

In this case, the discussion depends on application needs. The only
case where adding useless repair symbols does not result in reduced
goodput is when the application needs a limited amount of goodput
(e.g., VoIP traffic). In this case, the useless repair symbols would
only impact the amount of data generated in the network.

6.5.2. FEC within the transport

The sender may exploit the information given by the receiver to
reduce the number of useless repair symbols and the resulting goodput
reduction.

6.5.3. FEC below the transport

In this case, the useless repair symbols only impact the load of the
network without actual gain for the coded flow.

7. Open research questions

This section provides a simplified state-of-the art of the activities
related to congestion control and coding. The objective is to
identify open research questions and contribute to advice when
evaluating coding mechanisms.

7.1. Activities related to congestion control and coding

We map activities related to congestion control and coding with the
organization presented in this document:

o For the FEC above transport case: TBD

   o For the FEC within transport case:
      [I-D.swett-nwcrg-coding-for-quic], [QUIC-FEC], [RFC5109].

   o For the FEC below transport case: [NCTCP],
      [I-D.detchart-nwcrg-tetrys].

7.2. Open research questions

   The research questions should be mapped following the organization of
   this document. In all these three use-cases, open questions remain.
   There is a general trade-off, inherent to the use of coding, between
   (1) reducing goodput when useless repair symbols are transmitted and
   (2) helping to recover from transmission and congestion losses.

   For the FEC above transport case, there is a trade-off related to the
   amount of redundancy to add, as a function of the transport layer
   protocol and application requirements.

   For the FEC within transport case, recovering lost symbols may hide
   congestion losses to the congestion control. Some existing solutions
   already propose to disambiguate acked packets from rebuilt packets
   [QUIC-FEC]. New signalling methods and FEC-recovery-aware congestion
   controls could be proposed.

   For the FEC below transport case, there are opportunities for
   introducing interaction between congestion control and coding schemes
   to improve the quality of experience while guaranteeing fairness with
   other flows. An open question also resides in the relevance of FEC
   when there are multiple streams that exploit the FEC channel.

7.3. Advices for evaluating coding mechanisms

   The contribution to research questions should be mapped following the
   organization of this document. Otherwise, this may lead to wrong
   assumptions on the validity of the proposal and wrong ideas about the
   relevance of coding for a given use case.

   The discussion provided in this document aims at encouraging the
   research community to also consider congestion control aspects when
   proposing and comparing FEC coding solutions in communication
   systems. As one example, this draft proposes discussions on the
   impact of the proposed FEC solution on congestion control, especially
   loss-based congestion control mechanisms. When a research work aims
   at improving the throughput by hiding the packet loss signal from the
   congestion control, the authors should 1) discuss the advantages of
   using the proposed FEC solution compared to replacing the congestion
   control by one that ignores a portion of the encountered losses, 2)

critically discuss the impact of hiding packet loss from the
congestion control mechanism.

8. 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.

9. IANA Considerations

This memo includes no request to IANA.

10. 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.

11. Informative References

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

[I-D.briscoe-tsvarea-fair]
            Briscoe, B., "Flow Rate Fairness: Dismantling a Religion",
            draft-briscoe-tsvarea-fair-02 (work in progress), July
            2007.

[I-D.detchart-nwcrg-tetrys]
            Detchart, J., Lochin, E., Lacan, J., and V. Roca, "Tetrys,
            an On-the-Fly Network Coding protocol", draft-detchart-
            nwcrg-tetrys-05 (work in progress), February 2020.

[I-D.swett-nwcrg-coding-for-quic]
            Swett, I., Montpetit, M., Roca, V., and F. Michel, "Coding
            for QUIC", draft-swett-nwcrg-coding-for-quic-04 (work in
            progress), March 2020.

[NCTCP]   Sundararajan (et al.), J., "Network Coding Meets TCP:
            Theory and Implementation", IEEE
            INFOCOM 10.1109/JPROC.2010.2093850, 2009.

   [QUIC-FEC]
              Michel (et al.), F., "QUIC-FEC: Bringing the benefits of
              Forward Erasure Correction to QUIC", IFIP
              Networking 10.23919/IFIPNetworking.2019.8816838, 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>.

   [RFC5109] Li, A., Ed., "RTP Payload Format for Generic Forward Error
              Correction", RFC 5109, DOI 10.17487/RFC5109, December
              2007, <https://www.rfc-editor.org/info/rfc5109>.

   [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>.

   [RFC6356] Raiciu, C., Handley, M., and D. Wischik, "Coupled
              Congestion Control for Multipath Transport Protocols",
              RFC 6356, DOI 10.17487/RFC6356, October 2011,
              <https://www.rfc-editor.org/info/rfc6356>.

   [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>.

   [RFC8699] Islam, S., Welzl, M., and S. Gjessing, "Coupled Congestion
              Control for RTP Media", RFC 8699, DOI 10.17487/RFC8699,
              January 2020, <https://www.rfc-editor.org/info/rfc8699>.

   [TENTET]   Lochin, E., "On the joint use of TCP and Network Coding",
              NWCRG session IETF 100, 2017.

Authors' Addresses

    Nicolas Kuhn
    CNES

    Email: nicolas.kuhn@cnes.fr

    Emmanuel Lochin
    ENAC

    Email: emmanuel.lochin@enac.fr

    Francois Michel
    UCLouvain

    Email: francois.michel@uclouvain.be

    Michael Welzl
    University of Oslo

    Email: michawe@ifi.uio.no