                                                           F. Michel
                                                           UCLouvain
                                                           M. Welzl
                                                   University of Oslo
                                                       June 25, 2021

### Coding and congestion control in transport
### draft-irtf-nwcrg-coding-and-congestion-09

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.  FEC coding can help deal with losses
   at the end of transfers 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 a sender to
   detect transfer tail losses (losses that occur at the end of a
   transfer, where, e.g., TCP obtains no more ACKs that would enable it
   to quickly repair the loss via retransmission).  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 is a network 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 the
   loss-based congestion control of TCP; 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 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.

   We consider an end-to-end unicast data transfer with FEC coding in
   the application (above the transport), within the transport or
   directly below the transport.  A typical scenario for the

considerations in this document is a client browsing the web or
watching a live video.

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.  Context

### 2.1.  Separate channels, separate entities

Figure 1 presents the notations that will be used in this document
and introduces the Forward Erasure Correction (FEC) and Congestion
Control (CC) channels.  The Forward Erasure Correction channel
carries repair symbols (from the sender to the receiver) and
information from the receiver to the sender (e.g. signaling which
symbols have been recovered, loss rate prior and/or after decoding,
etc.).  The Congestion Control channel carries network packets from a
sender to a receiver, and packets signaling information about the
network (number of packets received vs. lost, Explicit Congestion
Notification (ECN) marks, etc.) from the receiver to the sender.  The
network packets that are sent by the Congestion Control channel may
be composed of source packets and/or repair symbols.

```
  SENDER                                  RECEIVER

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


  +------+                            +------+
  |      |            source and/or       |      |
  |      | -----    repair symbols  ---->|      |
  | FEC  |                            | FEC  |
  |      |            signaling           |      |
  |      | <---   recovered symbols  ----|      |
  +------+                            +------+
```
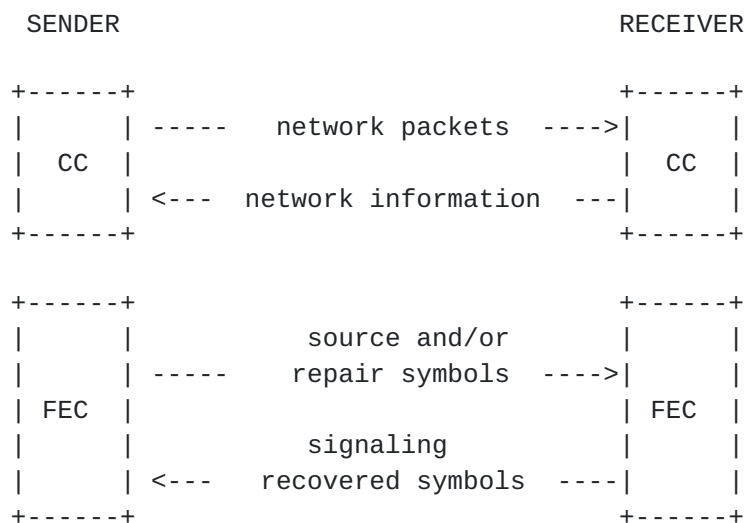
                Figure 1: Notations and separate channels

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      |                |network
  |              |symbols     |                |packets
  +--------------+==>         +----------------+==>
    ^                                 ^
    | signaling                       | network
    | recovered symbols               | information
```
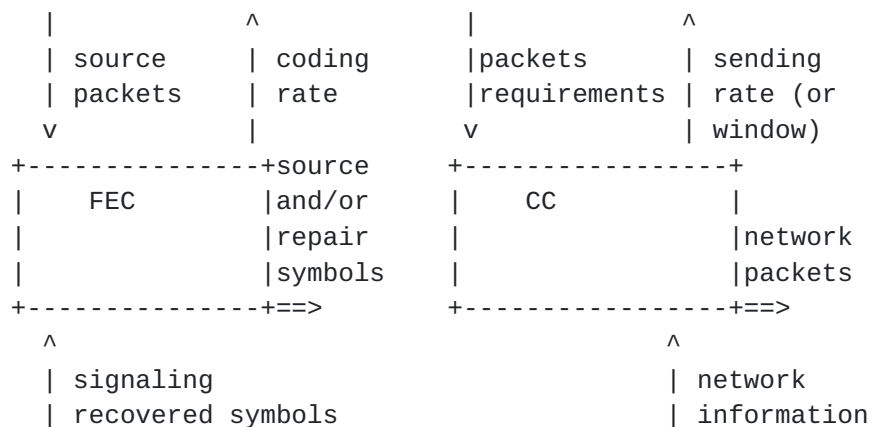
                Figure 2: Separate entities (sender-side)

```
     |                            |
     | source and/or              | network
     | repair symbols             | packets
     v                            v
  +--------------+            +----------------+
  |    FEC       |signaling   |    CC          |
  |              |recovered   |                |network
  |              |symbols     |                |information
  +--------------+==>         +----------------+==>
```
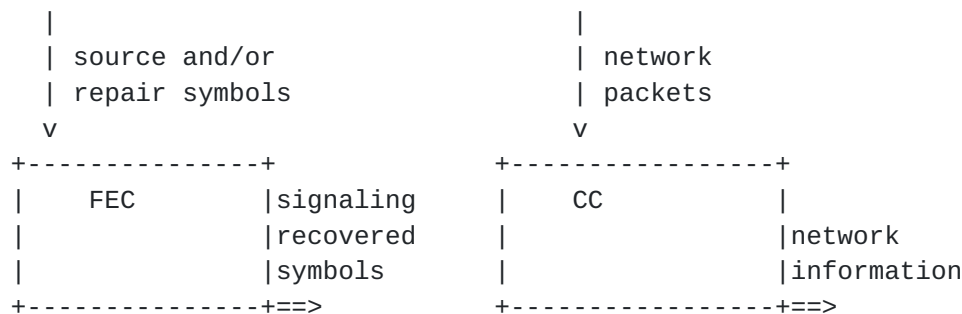
                Figure 3: Separate entities (receiver-side)

   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 recovered 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 (i.e.  proportion of
   transmitted symbols that carry useful data).

o  'network packets' (output data plane for the CC): refers to the
   data that is transmitted by a CC sender to a CC receiver.  The
   network packets may contain source and/or repair symbols.

o  'source and/or repair symbols' (data plane for the FEC): refers to
   the data that is transmitted by a FEC sender to a FEC receiver.
   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 symbols) or a mix of both.

The inputs to FEC (incoming data packets without repair symbols, and
signaling from the receiver about losses and/or recovered 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 recovered blocks may be quite irrelevant to a CC
algorithm.

## 2.2.  Relation between transport layer and application requirements

The choice of the adequate transport layer may be related to
application requirements and the services offered by a transport
protocol [RFC8095]:

o  The transport layer may provide an unreliable transport service
   (e.g.  UDP or DCCP [RFC4340]) or a partially reliable transport
   service (e.g.  SCTP with the partial reliability extension
   [RFC3758] or QUIC with the unreliable datagram extension
   [I-D.ietf-quic-datagram]).  Depending on the amount of redundancy
   and network conditions, there could be cases where it becomes
   impossible to carry traffic.

o  The transport layer may implement a retransmission mechanism to
   guarantee the reliability of a data 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.

**2.3.  Scope of the document concerning transport multipath and multi-streams applications**

   The application layer can be composed of several streams above FEC
   and transport layers instances.  The transport layer can exploit a
   multipath mechanism.  The different streams could exploit different
   paths between the sender and the receiver.  Moreover, a single-stream
   application could also exploit a multipath transport mechanism.  This
   section describes what is in the scope of this document in regards
   with multi-streams applications and multipath transport protocols.

   The different combinations between multi-stream applications and
   multipath transport are the following: (1) one application layer
   stream as input packets above a combination of FEC and multipath
   (Mpath) transport layers (Figure 4), and (2) multiple application
   layer streams as input packets above a combination of FEC and
   multipath (Mpath) or single path (Spath) transport layers (Figure 5).
   This document further details cases I (in Section 3.7), II (in
   Section 4.7) and III (in Section 5.7) illustrated in Figure 4.  Cases
   IV, V and VI of Figure 5 are related to how multiple streams are
   managed by a single transport or FEC layer: this does not directly
   concerns the interaction between FEC and the transport and is out of
   the scope of this document.

```
          CASE I                 CASE II                 CASE III
    +---------------+     +---------------+     +---------------+
    |    Stream 1   |     |    Stream 2   |     |    Stream 3   |
    +---------------+     +---------------+     +---------------+


    +---------------+     +---------------+     +---------------+
    |      FEC      |     |      FEC      |     |Mpath Transport|
    +---------------+     |      in       |     +---------------+
                          |Mpath Transport|
    +---------------+     |               |     +-----+   +-----+
    |Mpath Transport|     |               |     |Flow1|...|FlowM|
    +---------------+     +---------------+     +-----+   +-----+


    +-----+   +-----+     +-----+   +-----+     +-----+   +-----+
    |Flow1|...|FlowM|     |Flow1|...|FlowM|     | FEC |...| FEC |
    +-----+   +-----+     +-----+   +-----+     +-----+   +-----+
```
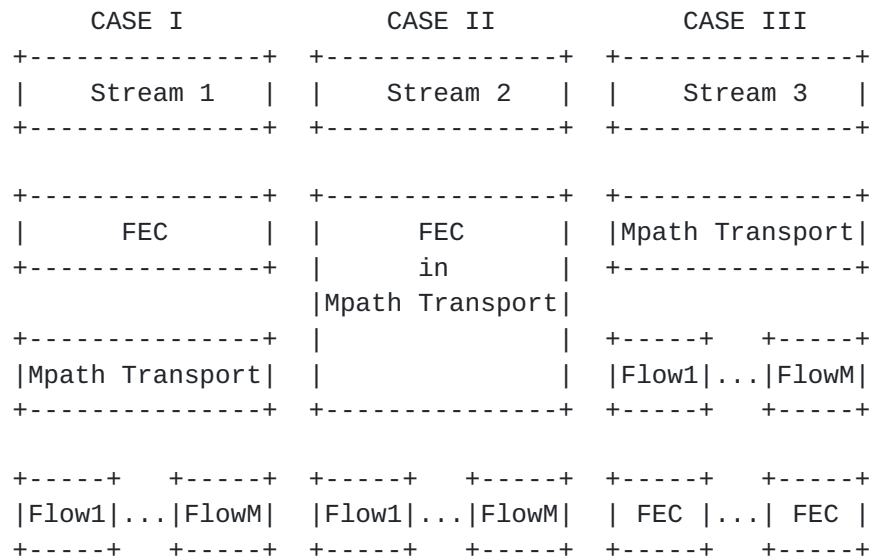
     Figure 4: Transport multipath and single stream applications - in the
                       scope of the document

```
         CASE IV                CASE  V                 CASE VI
   +-------+   +-------+  +-------+   +-------+  +-------+   +-------+
   |Stream1|...|StreamM|  |Stream1|...|StreamM|  |Stream1|...|StreamM|
   +-------+   +-------+  +-------+   +-------+  +-------+   +-------+


   +------------------+   +------------------+   +------------------+
   |                  |   |       FEC        |   | Mpath Transport  |
   |       FEC        |   +------------------+   +------------------+
   |   above/in/below |
   |   Spath Transport|   +------------------+   +------------------+
   |                  |   | Mpath Transport  |   |       FEC        |
   +------------------+   +------------------+   +------------------+


   +------------------+   +-----+        +-----+  +-----+        +-----+
   |       Flow       |   |Flow1|  ...   |FlowM|  |Flow1|  ...   |FlowM|
   +------------------+   +-----+        +-----+  +-----+        +-----+
```
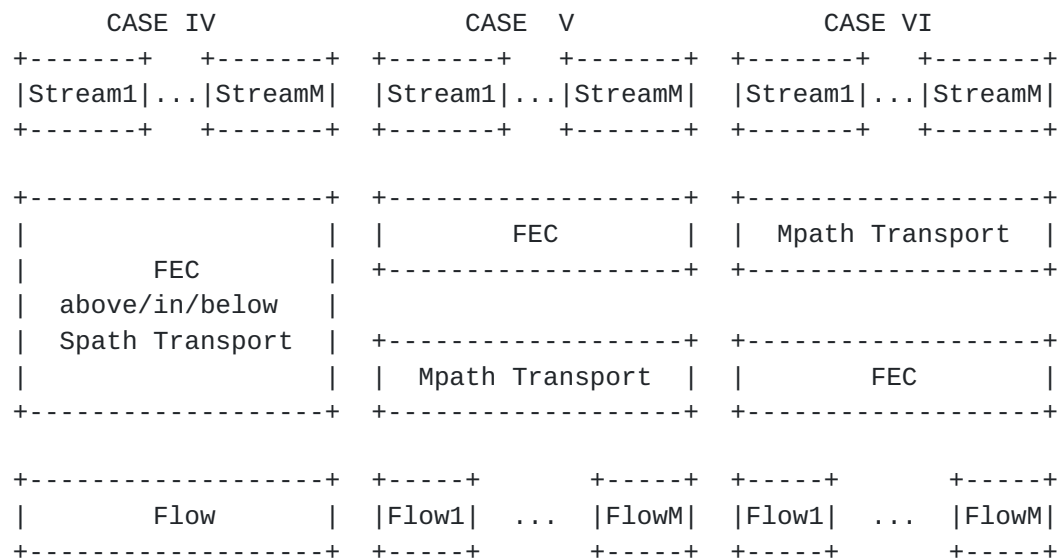
        Figure 5: Transport single path, transport multipath and multi-stream
                  applications - out of the scope of the document

## 2.4.  Types of coding

   [RFC8406] summarizes recommended terminology for Network Coding
   concepts and constructs.  In particular, the document identifies the
   following coding types (among many others):

   o  Block Coding: Coding technique where the input Flow must first be
      segmented into a sequence of blocks; FEC encoding and decoding are
      performed independently on a per-block basis.

   o  Sliding Window Coding: general class of coding techniques that
      rely on a sliding encoding window.

   The decoding scheme may not be able to decode all the symbols.  The
   chance of decoding the erased packets depends on the size of the
   encoding window, the coding rate and the distribution of erasure in
   the transmission channel.  The FEC channel may let the client
   transmit information related to the need of supplementary symbols to
   adapt the level of reliability.  Partial and full reliability could
   be envisioned.

   o  Full reliability: The receiver may hold symbols until the decoding
      of source symbols is possible.  In particular, if the codec does
      not enable a subset of the system to be inverted, the receiver
      would have to wait for a certain minimum amount of repair packets
      before it can recover all the source symbols.

o  Partial reliability: The receiver cannot deliver source symbols
   that could not have been decoded to the upper layer.  For a fixed
   size of encoding window (for Sliding Window Coding) or of blocks
   (for Block Coding) containing the source symbols, increasing the
   amount of repair symbols would increase the chances of recovering
   the erased symbols.  However, this would impact on memory
   requirements, on the cost of encoding and decoding processes and
   on the network overhead.

## 2.5.  Fairness, a policy concern

Traffic from or to different end users may share various types of
bottlenecks.  When such a shared bottleneck does not implement some
form of flow protection, the share of the available capacity between
single flows can help assess when one flow starves the other.

As one example, for residential accesses, the data rate can be
guaranteed for the customer premises equipment, but not necessarily
for the end user.  The quality of service that guarantees fairness
between the different clients can be seen as a policy concern
[I-D.briscoe-tsvarea-fair].

While past efforts have focused on achieving fairness, quantifying
and limiting harm caused by new algorithms (or algorithms with
coding) is more practical [BEYONDJAIN].  This document considers
fairness as the impact of the addition of coded flows on non-coded
flows when they share the same bottleneck.  It is assumed that the
non-coded flows respond to congestion signals from the network.  This
document does not contribute to the definition of fairness at a wider
scale.

## 3.  FEC above the transport

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

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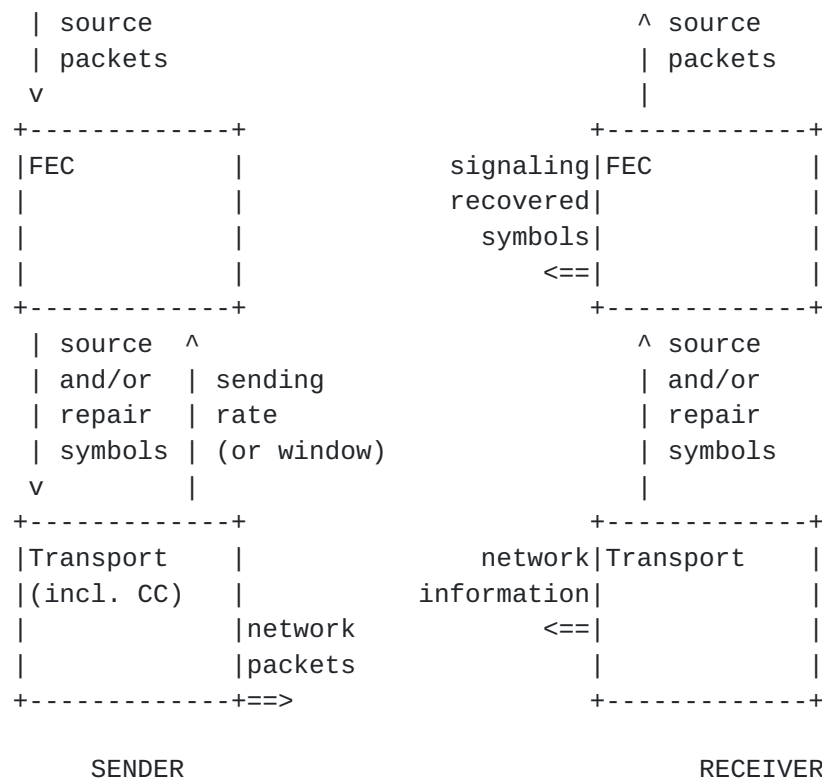
Figure 6: FEC above the transport

Figure 6 presents an architecture where FEC operates on top of the
transport.

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 or rate determined by the CC
mechanism.  This can result in improved quality of experience for
latency sensitive applications such as VoIP or any not-fully reliable
services.

This approach requires that the transport protocol does not implement
a fully reliable in-order data transfer service (e.g., like TCP).
QUIC with unreliable datagram extension [I-D.ietf-quic-datagram] is
an example of a protocol for which this is relevant.  In cases where
QUIC traffic is blocked and a fall-back to TCP is proposed, there is
a risk for bad interactions between TCP's full reliability and coding
schemes.  For reliable transfers, coding usage does not guarantee
better performance; instead, it would mainly reduce goodput.

### [3.1](#). Fairness and impact on non-coded flows

The addition of coding within the flow does not influence the interaction between coded and non-coded flows.  This interaction would mainly depend on the congestion controls associated with each flow.

### [3.2](#). Congestion control and recovered symbols

The congestion control mechanism receives network packets and may not be able to differentiate repair symbols from actual source ones.  The relevance of adding coding at the application layer is related to the needs of the application.  For real-time applications using an unreliable or partially reliable transport, this approach may reduce the number of losses perceived by the application.

### [3.3](#). Interactions between congestion control and coding rates

The coding rate applied at the application layer mainly depends on the available rate or congestion window given by the congestion control underneath.  The coding rate could be adapted to avoid adding overhead when the minimum required data rate of the application is not provided by the congestion control underneath.  When the congestion control allows sending faster than the application needs, adding coding can reduce packet losses and improve the quality of experience (provided that an unreliable or partially reliable transport is used).

### [3.4](#). On useless repair symbols

The discussion depends on application needs.  The only case where adding useless repair symbols does not obviously result in reduced goodput is when the application rate is limited (e.g., VoIP traffic). In this case, useless repair symbols would only impact the amount of data generated in the network.  Extra data in the network can, however, increase the likelihood of increasing delay and/or packet loss, which could provoke a congestion control reaction that would degrade goodput.

### [3.5](#). On partial ordering

Irrespective of the transport protocol, a FEC mechanism does not require to implement a reordering mechanism if the application does not need it.  However, if the application needs in-order delivery of packets, a reordering mechanism at the receiver is required.

## 3.6.  On partial reliability

   The application may require partial reliability.  In this case, the
   coding rate of a FEC mechanism could be adapted based on inputs from
   the application and the trade-off between latency and packet loss.
   Partial reliability impacts the type of FEC and type of codec that
   can be used, such as discussed in Section 2.4.

## 3.7.  On multipath transport

   Whether the transport protocol exploits multiple paths or not does
   not have an impact on the FEC mechanism.

## 4.  FEC within the transport

```
  | source                              ^ source
  | packets                             | packets
  v                                     |
 +------------+              +------------+
 | Transport  |              | Transport  |
 |            |              |            |
 | +---+ +--+ |      signaling| +---+ +--+ |
 | |FEC| |CC| |      recovered| |FEC| |CC| |
 | +---+ +--+ |      symbols| +---+ +--+ |
 |            |            <==|            |
 |            |network      network|            |
 |            |packets   information|            |
 +------------+ ==>              <==+------------+

      SENDER                         RECEIVER
```
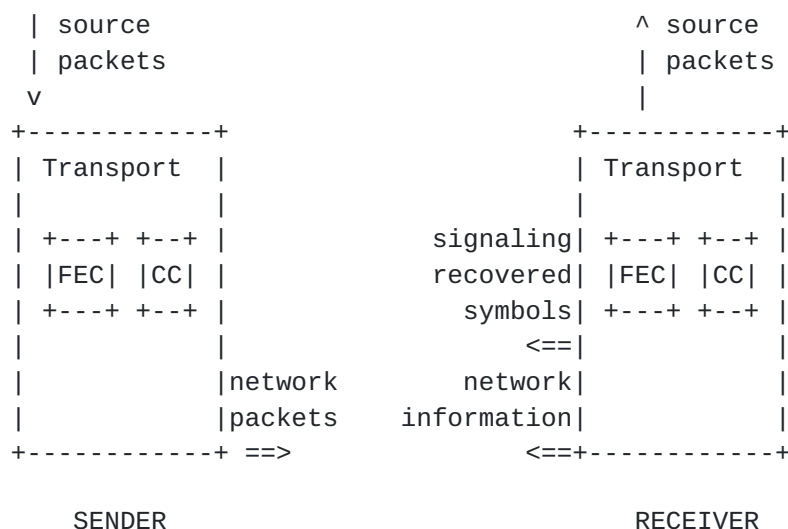
                   Figure 7: FEC in the transport

   Figure 7 presents an architecture where FEC operates within the
   transport.  The repair symbols are sent within what the congestion
   window or calculated rate allows, such as in [CTCP].

   The advantage of this approach is that it allows a joint optimization
   of CC and 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 long
   transfers but the amount of repair symbols can be adapted, e.g.
   depending on the congestion window size.  There is a trade-off
   between 1) the capacity that could have been exploited by application
   data instead of transmitting source packets, and 2) the benefits
   derived from transmitting repair symbols (e.g. unlocking the receive

buffer if it 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.
Sending repair symbols can avoid the silence period between the
transmission of the last packet in the send buffer and 1) firing a
retransmission of lost packets, or 2) the transmission of new
packets.

## 4.1.  Fairness and impact on non-coded flows

The addition of coding within the transport may impact the congestion
control mechanism and hide congestion losses.  Specific interaction
between congestion controls and coding schemes can be proposed (see
Section 4.2, Section 4.3 and Section 4.4).  If no specific
interaction is introduced, the coding scheme may hide congestion
losses from the congestion controller and the description of
Section 5 may apply.

## 4.2.  Congestion control and recovered symbols

The receiver can differentiate between 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.

## 4.3.  Interactions between congestion control and coding rates

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 from losses
earlier than with retransmissions.  The receiver may indicate to the
sender the number of packets that have been received or recovered.
The sender may use this information to tune the coding ratio.  For
example, coupling an increased transmission rate with an increasing
or decreasing coding rate could be envisioned.  A server may use a
decreasing coding rate as a probe of the channel capacity and adapt
the congestion control transmission rate.

## 4.4.  On useless repair symbols

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

## 4.5.  On partial ordering

The application may require in-order delivery of packets.  In this
case, both FEC and transport layer mechanisms should guarantee that
packets are delivered in order.  If partial ordering is requested by
the application, both the FEC and transport could relax the
constraints related to in-order delivery: reordering mechanisms at
the receiver may not be necessary.

## 4.6.  On partial reliability

The application may require partial reliability.  The reliability
offered by FEC may be sufficient, with no retransmission required.
This depends on application needs and the trade-off between latency
and loss.  Partial reliability impacts the type of FEC and type of
codec that can be used, such as discussed in Section 2.4.

## 4.7.  On transport multipath

The sender may adapt the coding rate of each of the single subpaths,
whether the congestion control is coupled or not.  There is an
important flexibility on how the coding rate is tuned depending on
the characteristics of each subpath.

## 5.  FEC below the transport

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


      SENDER                        RECEIVER
```
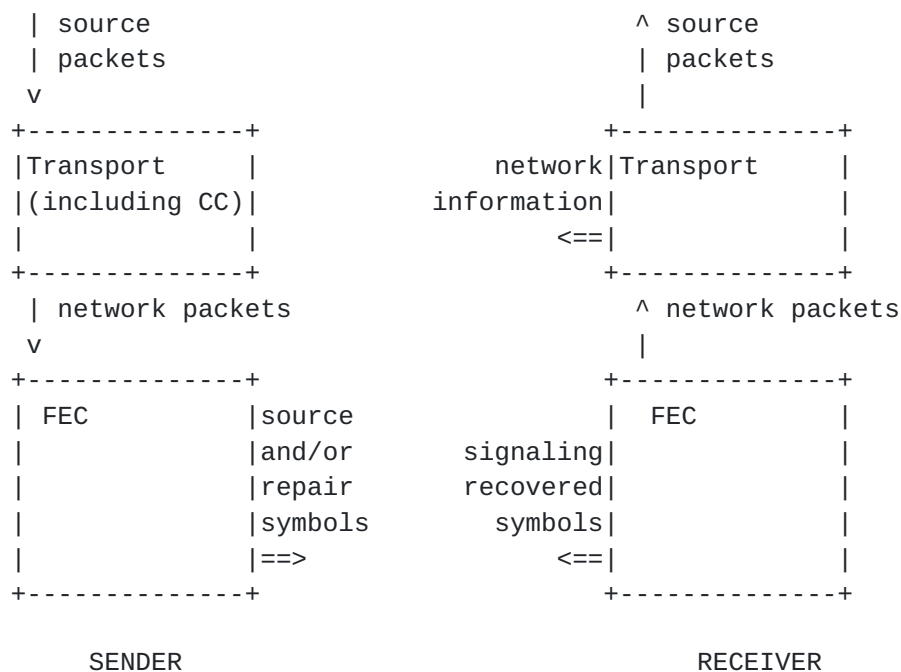
Figure 8: FEC below the transport

Figure 8 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, where it contributes to the
total capacity that a link exposes to upper layers.  This application
of FEC is out of scope of this document.  This includes the use of
FEC on top of a link layer in scenarios where the link is known by
configuration.  In the scenario considered here, the repair symbols
are sent on top of what is allowed by the congestion control.

Including redundancy adds traffic without reducing goodput but incurs
potential fairness issues.  The effective bit-rate is higher than the
CC's computed fair share due to the transmission of repair symbols,
and 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 to add a given percentage of the
congestion window or rate as supplementary symbols, or to send a
fixed amount of repair symbols at a fixed 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 recovered symbols 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.

## 5.1.  Fairness and impact on non-coded flows

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 5.2 and [TENTET]).

## 5.2.  Congestion control and recovered symbols

The congestion control may not be aware of the existence of a coding
scheme underneath it.  The congestion control may behave as if no
coding scheme had been introduced.  The only way for a coding channel
to indicate that symbols have been lost but 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 received, yet congestion has occurred, by using ECN
signaling [TENTET].

## 5.3.  Interactions between congestion control and coding rates

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

## 5.4.  On useless repair symbols

Useless repair symbols only impact the load on the network without
actual gain for the coded flow.  Using feedback signaling, FEC
mechanisms can measure the ratio between actually used and useless
symbols, and adjust the coding rate.

## 5.5.  On partial ordering

The transport above the FEC channel may support out-of-order delivery
of packets: reordering mechanisms at the receiver may not be
necessary.  In cases where the transport requires in-order delivery,
the FEC channel may need to implement a reordering mechanism.
Otherwise, spurious retransmissions may occur at the transport level.

## 5.6.  On partial reliability

The transport or application layer above the FEC channel may require
partial reliability only.  In this case, FEC may provide an
unnecessary service if it is not aware of the reliability
requirements.  Partial reliability impacts the type of FEC and type
of codec that can be used, such as discussed in Section 2.4.

## 5.7.  On transport multipath

The transport may exploit multiple paths without the FEC channel
being aware of it.  This depends on whether FEC is applied to all
subflows or each of the subflows individually.  When FEC is applied
to all the flows, there is a risk for the coding rate to be
inadequate for the characteristics of the individual paths.

6.  Research recommendations and questions

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

6.1.  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: [RFC8680].

   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].

6.2.  Open research questions

   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.

6.2.1.  Parameter derivation

   There is a trade-off related to the amount of redundancy to add, as a
   function of the transport layer protocol and application
   requirements.

   [RFC8095] describes the mechanisms provided by existing IETF
   protocols such as TCP, SCTP or RTP.  [RFC8406] describes the variety
   of coding techniques.  The important level of combinations makes the
   determination of an optimum parameters derivation very complex.  This
   depends on application requirements and deployment context.

   Appendix C of [RFC8681] describes how to tune the parameters for
   target use-case.  However, this discussion does not integrate
   congestion-controlled end points.

   Research question 1 : "Is there a way to dynamically adjust the codec
   characteristics depending on the transmission channel, the transport
   protocol and application requirements ?"

Research question 2 : "Should we apply specific per-stream FEC
mechanisms when multiple streams with different reliability needs are
carried out ?"

## 6.2.2.  New signaling methods and fairness

Recovering lost symbols may hide congestion losses from the
congestion control.  Disambiguate acked packets from rebuilt packets
would help the sender adapt its sending rate accordingly.  There are
opportunities for introducing interaction between congestion control
and coding schemes to improve the quality of experience while
guaranteeing fairness with other flows.

Some existing solutions already propose to disambiguate acked packets
from rebuilt packets [QUIC-FEC].  New signaling methods and FEC-
recovery-aware congestion controls could be proposed.  This would
allow the design of adaptive coding rates.

Research question 3 : "Should we quantify the harm that a coded flow
would induce on a non-coded flow ? How can this be reduced while
still benefiting from advantages brought by FEC ?"

Research question 4 : "If transport and FEC senders are collocated
and close to the client, and FEC is applied only on the last mile,
e.g. to ignore losses on a noisy wireless link, would this raise
fairness issues ?"

Research question 5 : "Should we propose a generic API to allow
dynamic interactions between a transport protocol and a coding scheme
? This should consider existing APIs between application and
transport layers."

## 6.3.  Recommendations and advice for evaluating coding mechanisms

Research Recommendation 1: "From a congestion control point-of-view,
a recovered packet must be considered as a lost packet.  This does
not apply to the usage of FEC on a path that is known to be lossy."

Research Recommendation 2: "New research contributions should be
mapped following the organization of this document (above, below, in
the congestion control) and should consider congestion control
aspects when proposing and comparing FEC coding solutions in
communication systems."

Research Recommendation 3: "When a research work aims at improving
throughput by hiding the packet loss signal from congestion control
(e.g., because the path between the sender and receiver is known to
consist of a noisy wireless link), 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."

## 7.  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.

## 8.  IANA Considerations

This memo includes no request to IANA.

## 9.  Security Considerations

FEC and CC schemes can contribute to DoS attacks.  Moreover, the transmission of signaling messages from the client to the server should be protected and reliable otherwise an attacker may compromise FEC rate adaptation.  Indeed, an attacker could either modify the values indicated by the client or drop signaling messages.

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.

## 10.  Informative References

[BEYONDJAIN]
          Ware (et al.), R., "Beyond Jain's Fairness Index: Setting
          the Bar For The Deployment of Congestion Control
          Algorithms", HotNets '19 10.1145/3365609.3365855, 2019.

[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-06 (work in progress), December 2020.

   [I-D.ietf-quic-datagram]
              Pauly, T., Kinnear, E., and D. Schinazi, "An Unreliable
              Datagram Extension to QUIC", draft-ietf-quic-datagram-02
              (work in progress), February 2021.

   [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>.

   [RFC8095]  Fairhurst, G., Ed., Trammell, B., Ed., and M. Kuehlewind,
              Ed., "Services Provided by IETF Transport Protocols and
              Congestion Control Mechanisms", RFC 8095,
              DOI 10.17487/RFC8095, March 2017,
              <https://www.rfc-editor.org/info/rfc8095>.

   [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>.

   [RFC8680]  Roca, V. and A. Begen, "Forward Error Correction (FEC)
              Framework Extension to Sliding Window Codes", RFC 8680,
              DOI 10.17487/RFC8680, January 2020,
              <https://www.rfc-editor.org/info/rfc8680>.

   [RFC8681]  Roca, V. and B. Teibi, "Sliding Window Random Linear Code
              (RLC) Forward Erasure Correction (FEC) Schemes for
              FECFRAME", RFC 8681, DOI 10.17487/RFC8681, January 2020,
              <https://www.rfc-editor.org/info/rfc8681>.

   [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