
Stream: Internet Research Task Force (IRTF)
RFC: [9265](#)
Category: Informational
Published: July 2022
ISSN: 2070-1721
Authors: N. Kuhn E. Lochin F. Michel M. Welzl
 CNES *ENAC* *UCLouvain* *University of Oslo*

RFC 9265

Forward Erasure Correction (FEC) Coding and Congestion Control in Transport

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 (NWCRCG). 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

This document is not an Internet Standards Track specification; it is published for informational purposes.

This document is a product of the Internet Research Task Force (IRTF). The IRTF publishes the results of Internet-related research and development activities. These results might not be suitable for deployment. This RFC represents the consensus of the Network Coding for Efficient Network Communications Research Group of the Internet Research Task Force (IRTF). Documents approved for publication by the IRSG are not candidates for any level of Internet Standard; see Section 2 of RFC 7841.

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at <https://www.rfc-editor.org/info/rfc9265>.

Copyright Notice

Copyright (c) 2022 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document.

Table of Contents

1. Introduction
2. Context
 - 2.1. Fairness, Quantifying and Limiting Harm, and Policy Concerns
 - 2.2. Separate Channels, Separate Entities
 - 2.3. Relation between Transport Layer and Application Requirements
 - 2.4. Scope of the Document Concerning Transport Multipath and Multistream Applications
 - 2.5. Types of Coding
3. FEC above the Transport
 - 3.1. Fairness and Impact on Non-coded Flows
 - 3.2. Congestion Control and Recovered Symbols
 - 3.3. Interactions between Congestion Control and Coding Rates
 - 3.4. On Useless Repair Symbols
 - 3.5. On Partial Ordering at FEC Level
 - 3.6. On Partial Reliability at FEC Level
 - 3.7. On Multipath Transport and FEC Mechanism
4. FEC within the Transport
 - 4.1. Fairness and Impact on Non-coded Flows
 - 4.2. Interactions between Congestion Control and Coding Rates
 - 4.3. On Useless Repair Symbols
 - 4.4. On Partial Ordering at FEC and/or Transport Level
 - 4.5. On Partial Reliability at FEC Level

- 4.6. On Transport Multipath and Subpath FEC Coding Rate
 - 5. FEC below the Transport
 - 5.1. Fairness and Impact on Non-coded Flows
 - 5.2. Congestion Control and Recovered Symbols
 - 5.3. Interactions between Congestion Control and Coding Rates
 - 5.4. On Useless Repair Symbols
 - 5.5. On Partial Ordering at FEC Level with In-Order Delivery Transport
 - 5.6. On Partial Reliability at FEC Level
 - 5.7. FEC Not Aware of Transport Multipath
 - 6. Research Recommendations and Questions
 - 6.1. Activities Related to Congestion Control and Coding
 - 6.2. Open Research Questions
 - 6.2.1. Parameter Derivation
 - 6.2.2. New Signaling Methods and Fairness
 - 6.3. Recommendations and Advice for Evaluating Coding Mechanisms
 - 7. IANA Considerations
 - 8. Security Considerations
 - 9. Informative References
- Acknowledgements
- Authors' Addresses

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 and the loss detection of congestion controls are two distinct and separate reliability mechanisms. Since FEC coding repairs losses, blindly applying FEC may easily lead to an implementation that also hides a congestion signal from the sender. It is important to ensure that such hiding of information does not occur, because loss may be the only congestion signal available to the sender (e.g., TCP [RFC5681]).

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 to also consider congestion control aspects when proposing and comparing FEC coding solutions in communication systems. This document does not aim to propose guidelines for characterizing FEC coding solutions.

We consider three architectures for end-to-end unicast data transfer:

- with FEC coding in the application (above the transport) (Section 3),
- within the transport (Section 4), or
- directly below the transport (Section 5).

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 (NWCRCG); it is not an IETF product nor a standard. The document follows the terminology proposed in the taxonomy document [RFC8406].

2. Context

2.1. Fairness, Quantifying and Limiting Harm, and Policy Concerns

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 [FLOW-RATE-FAIRNESS].

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.

2.2. Separate Channels, Separate Entities

Figures 1 and 2 present the notations that will be used in this document and introduce the Forward Erasure Correction (FEC) and Congestion Control (CC) channels. The FEC 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 CC 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 [RFC3168], etc.) from the receiver to the sender. The network packets that are sent by the CC channel may be composed of source packets and/or repair symbols.

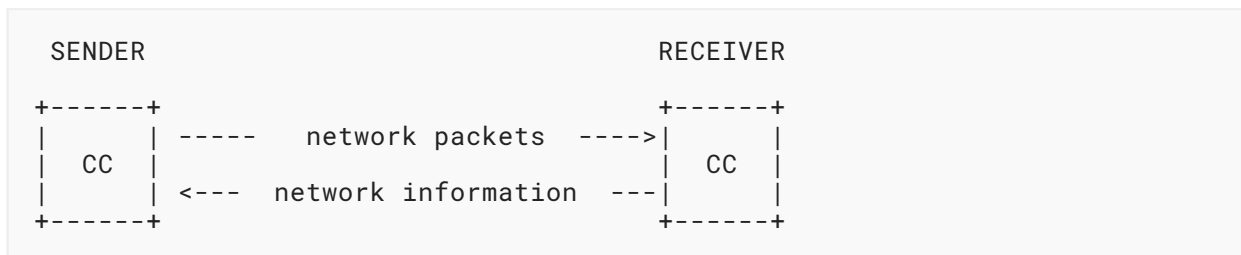


Figure 1: Congestion Control (CC) Channel

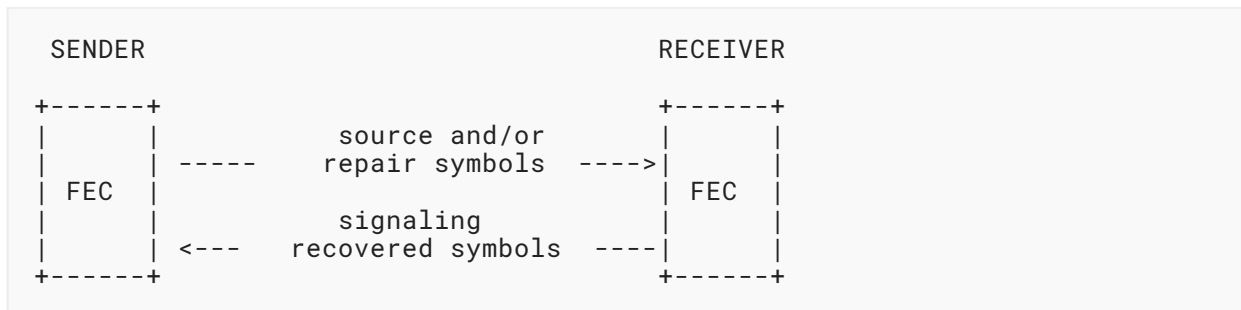


Figure 2: Forward Erasure Correction (FEC) Channel

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

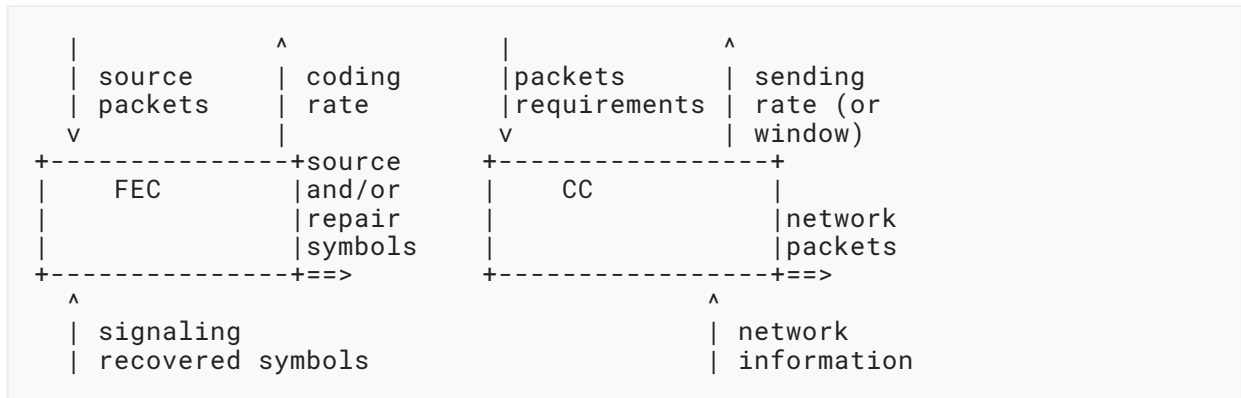


Figure 3: Separate Entities (Sender-Side)

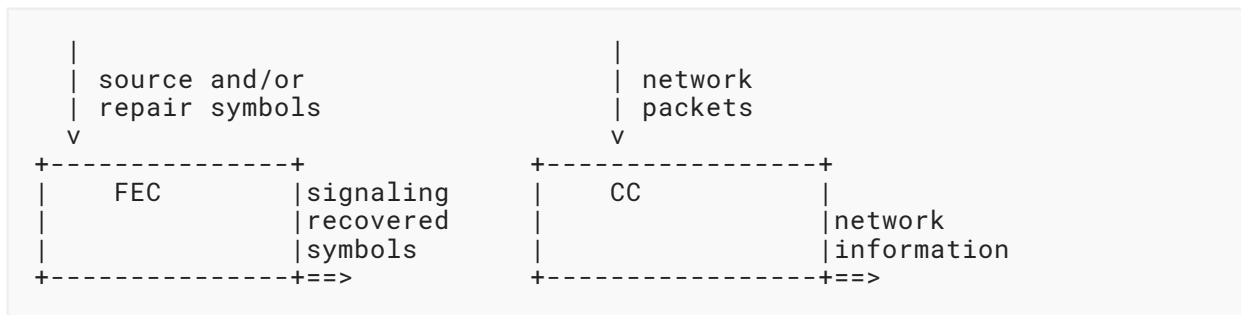


Figure 4: Separate Entities (Receiver-Side)

Figures 3 and 4 provide more details than Figures 1 and 2. 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.
- '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 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.
- '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).

'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.

'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.

2.3. 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](#)]:

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 ([Section 3](#)), FEC in CC ([Section 4](#)), and FEC below CC ([Section 5](#))), the impact of reliable transport on the FEC reliability mechanisms is different.

The transport layer may provide an unreliable transport service (e.g., UDP or the Datagram Congestion Control Protocol (DCCP) [[RFC4340](#)]) or a partially reliable transport service (e.g., the Stream Control Transmission Protocol (SCTP) with the partial reliability extension [[RFC3758](#)] or QUIC with the unreliable datagram extension [[RFC9221](#)]). Depending on the amount of redundancy and network conditions, there could be cases where it becomes impossible to carry traffic. This is further discussed in [Section 3](#) where a "FEC above CC" case is assessed and in [Sections 4](#) and [5](#) where "FEC in CC" and "FEC below CC" are assessed, respectively.

2.4. Scope of the Document Concerning Transport Multipath and Multistream Applications

The application layer can be composed of several streams above FEC and transport-layer 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 with regard to multistream applications and multipath transport protocols.

The different combinations between multistream 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 5) and (2) multiple application-layer streams as input packets above a combination of FEC and multipath (Mpath) or single path (Spath) transport layers (Figure 6). This document further details cases I (in Section 3.7), II (in Section 4.6), and III (in Section 5.7) as illustrated in Figure 5. Cases IV, V, and VI of Figure 6 are related to how multiple streams are managed by a single transport or FEC layer; this does not directly concern the interaction between FEC and the transport and is out of the scope of this document.

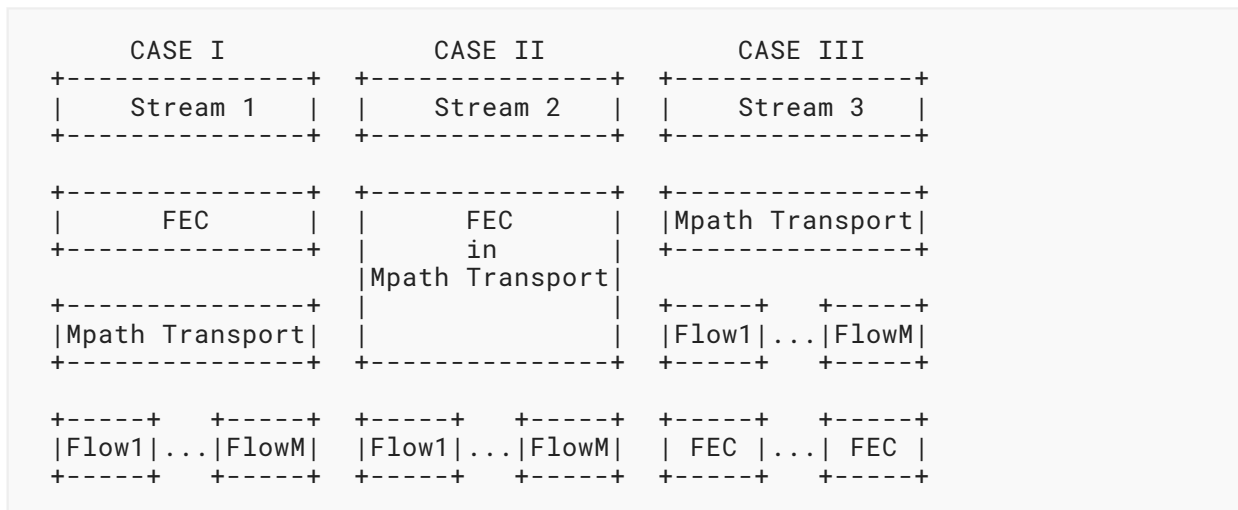


Figure 5: Transport Multipath and Single-Stream Applications - in the Scope of the Document

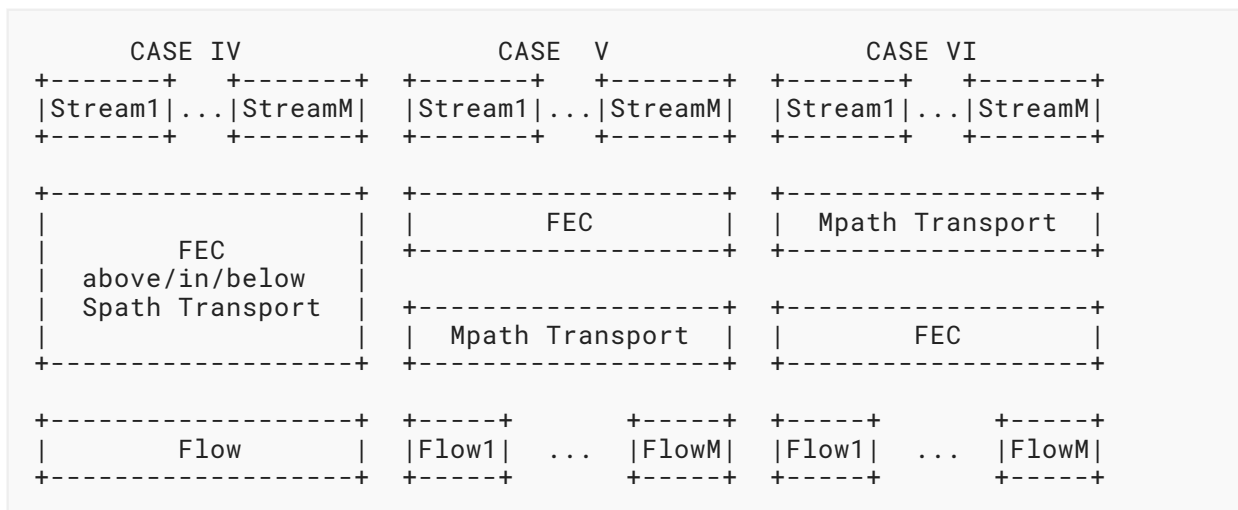


Figure 6: Transport Single Path, Transport Multipath, and Multistream Applications - out of the Scope of the Document

2.5. 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):

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.

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.

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.

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 have an impact on memory requirements, the cost of encoding and decoding processes, and the network overhead.

3. FEC above the Transport

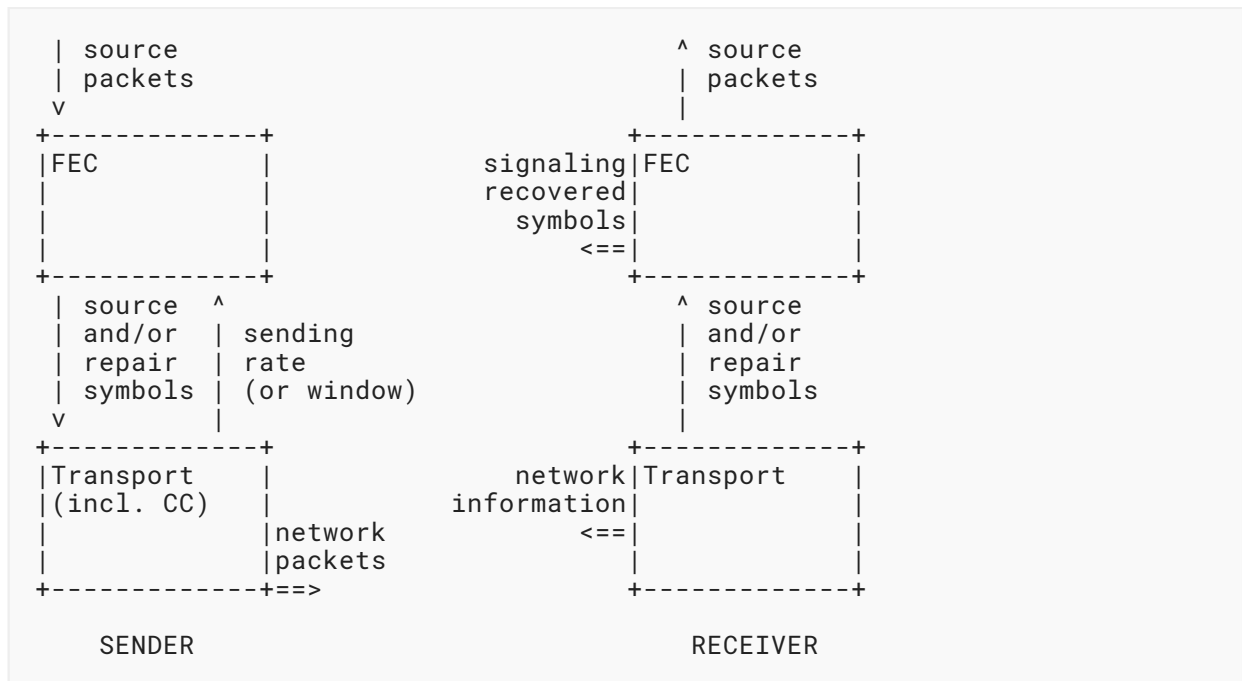


Figure 7: FEC above the Transport

Figure 7 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, because the repair symbols are sent following the congestion window or rate determined by the CC mechanism. This can result in an improved quality of experience for latency-sensitive applications such as Voice over IP (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 the unreliable datagram extension [RFC9221] is an example of a protocol for which this is relevant. In cases where the partially reliable transport is blocked and a fallback to a reliable transport is proposed, there is a risk for bad interactions between reliability at the transport level 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. This differentiation requires a transport protocol to provide more than the services described in [RFC8095], such as specifically indicating

what information has been repaired. 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 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 at FEC Level

Irrespective of the transport protocol, a FEC mechanism does not require implementing 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 at FEC Level

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 2.5](#).

3.7. On Multipath Transport and FEC Mechanism

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

4. FEC within the Transport

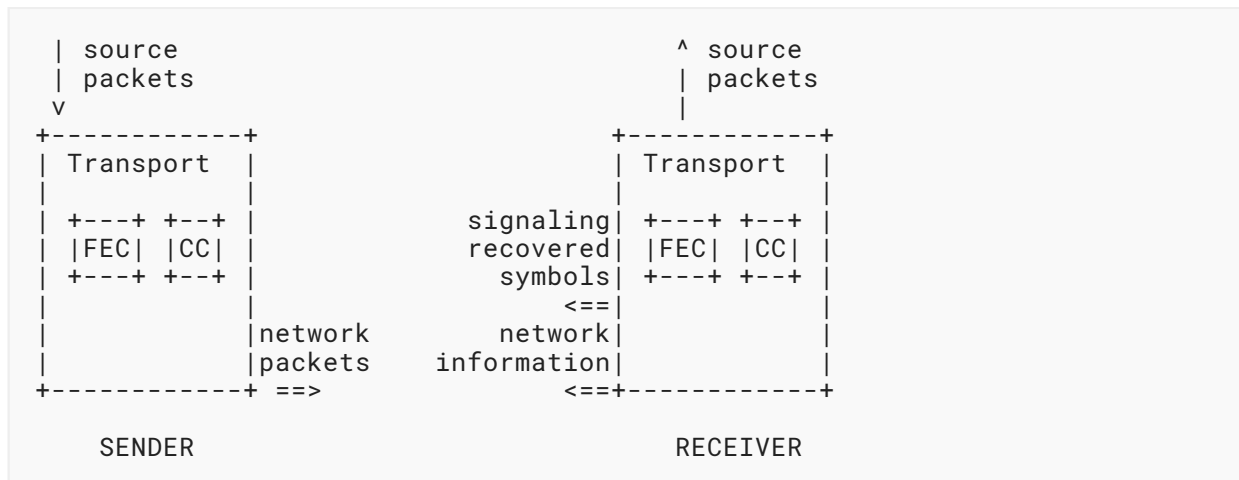


Figure 8: FEC in the Transport

Figure 8 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). This joint optimization is the key to prevent flows to consume the whole available capacity. The amount of repair traffic injected should not lead to congestion. As denoted in [FEC-CONGESTION-CONTROL], an increase of the repair ratio should be done conjointly with a decrease of the source sending rate.

The drawback of this approach is that it may require specific signaling and transport services that may not be described in [RFC8095]. Therefore, development and maintenance may require specific efforts at both the transport and the coding levels, and the design of the solution may end up being complex to suit different deployment needs.

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.

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.

4.1. Fairness and Impact on Non-coded Flows

Specific interaction between congestion controls and coding schemes can be proposed (see Sections 4.2 and 4.3). 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. Interactions between Congestion Control and Coding Rates

The receiver can differentiate between source packets and repair symbols. The receiver may indicate both the number of source packets received and the repair symbols that were actually useful in the recovery process of packets. The congestion control at the sender can then exploit this information to tune congestion control behavior.

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.

4.3. On Useless Repair Symbols

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

4.4. On Partial Ordering at FEC and/or Transport Level

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; partial ordering impacts both the congestion control and the type of FEC and type of codec that can be used.

4.5. On Partial Reliability at FEC Level

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 2.5.

4.6. On Transport Multipath and Subpath FEC Coding Rate

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

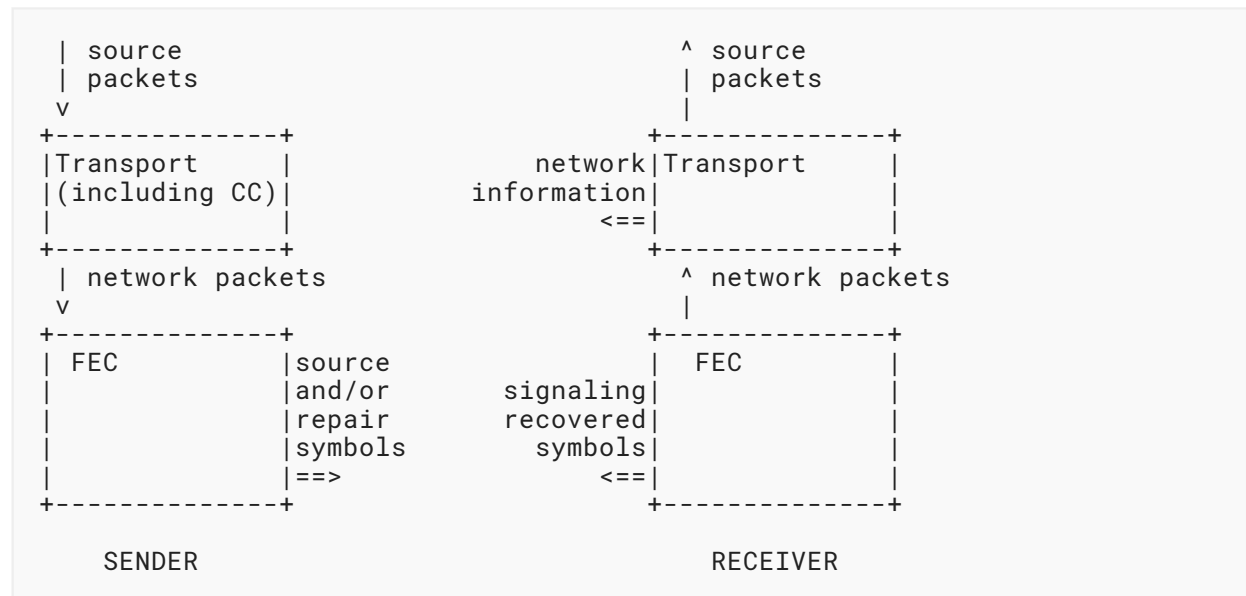


Figure 9: FEC below the Transport

Figure 9 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 on one or more of the links that make up the end-to-end path. The application of FEC at the link layer contributes to the total capacity that a link exposes to upper layers, but it may not be visible to either the end-to-end sender or the receiver, if the end-to-end sender and receiver are separated by more than one link; this is therefore out of scope for 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 not visible to the end-to-end congestion controller and may be 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 the number of symbols that were actually used and the number of symbols that were useless, and adjust the coding rate.

5.5. On Partial Ordering at FEC Level with In-Order Delivery Transport

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 at FEC Level

The transport or application layer above the FEC channel may require partial reliability only. FEC may provide an unnecessary service unless it is aware of the reliability requirements. Partial reliability impacts the type of FEC and codec that can be used, such as discussed in [Section 2.5](#).

5.7. FEC Not Aware of Transport Multipath

The transport may exploit multiple paths without the FEC channel being aware of it. If FEC is aware that multiple paths are in use, FEC can be applied to all subflows as an aggregate, or to each of the subflows individually. If FEC is not aware that multiple paths are in use, FEC can only be applied to each subflow individually. When FEC is applied to all the flows as an aggregate, the varying characteristics of the individual paths may lead to 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:

For the FEC above transport case: [\[RFC8680\]](#)

For the FEC within transport case: [\[CODING-FOR-QUIC\]](#), [\[QUIC-FEC\]](#), and [\[RFC5109\]](#)

For the FEC below transport case: [\[NCTCP\]](#) and [\[TETRYS\]](#)

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 number 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 a 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. Disambiguating 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, and 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 and 2) critically discuss the impact of hiding packet loss from the congestion control mechanism."

7. IANA Considerations

This document has no IANA actions.

8. 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.

9. Informative References

- [BEYONDJAIN]** Ware, R., Mukerjee, M. K., Seshan, S., and J. Sherry, "Beyond Jain's Fairness Index: Setting the Bar For The Deployment of Congestion Control Algorithms", HotNets '19: Proceedings of the 18th ACM Workshop on Hot Topics in Networks, DOI 10.1145/3365609.3365855, November 2019, <<https://doi.org/10.1145/3365609.3365855>>.
- [CODING-FOR-QUIC]** Swett, I., Montpetit, M., Roca, V., and F. Michel, "Coding for QUIC", Work in Progress, Internet-Draft, draft-swett-nwcr-g-coding-for-quic-04, 9 March 2020, <<https://datatracker.ietf.org/doc/html/draft-swett-nwcr-g-coding-for-quic-04>>.
- [CTCP]** Kim, M., Cloud, J., ParandehGheibi, A., Urbina, L., Fouli, K., Leith, D., and M. Medard, "Network Coded TCP (CTCP)", arXiv: 1212.2291v3, DOI 10.48550/arXiv.1212.2291, April 2013, <<https://doi.org/10.48550/arXiv.1212.2291>>.

- [FEC-CONGESTION-CONTROL]** Singh, V., Nagy, M., Ott, J., and L. Eggert, "Congestion Control Using FEC for Conversational Media", Work in Progress, Internet-Draft, draft-singh-rmcat-adaptive-fec-03, 20 March 2016, <<https://datatracker.ietf.org/doc/html/draft-singh-rmcat-adaptive-fec-03>>.
- [FLOW-RATE-FAIRNESS]** Briscoe, B., "Flow Rate Fairness: Dismantling a Religion", Work in Progress, Internet-Draft, draft-briscoe-tsvarea-fair-02, 11 July 2007, <<https://datatracker.ietf.org/doc/html/draft-briscoe-tsvarea-fair-02>>.
- [NCTCP]** Sundararajan, J., Shah, D., Médard, M., Jakubczak, S., Mitzenmacher, M., and J. Barros, "Network Coding Meets TCP: Theory and Implementation", Proceedings of the IEEE (Volume: 99, Issue: 3), DOI 10.1109/JPROC.2010.2093850, March 2011, <<https://doi.org/10.1109/JPROC.2010.2093850>>.
- [QUIC-FEC]** Michel, F., De Coninck, Q., and O. Bonaventure, "QUIC-FEC: Bringing the benefits of Forward Erasure Correction to QUIC", DOI 10.23919/IFIPNetworking.2019.8816838, May 2019, <<https://doi.org/10.23919/IFIPNetworking.2019.8816838>>.
- [RFC3168]** Ramakrishnan, K., Floyd, S., and D. Black, "The Addition of Explicit Congestion Notification (ECN) to IP", RFC 3168, DOI 10.17487/RFC3168, September 2001, <<https://www.rfc-editor.org/info/rfc3168>>.
- [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>>.

[RFC9221]

Pauly, T., Kinnear, E., and D. Schinazi, "An Unreliable Datagram Extension to QUIC", RFC 9221, DOI 10.17487/RFC9221, March 2022, <<https://www.rfc-editor.org/info/rfc9221>>.

[TENTET]

Lochin, E., "On the joint use of TCP and Network Coding", NWCRG Session, IETF 100, November 2017, <<https://datatracker.ietf.org/meeting/100/materials/slides-100-nwcrg-07-lochin-on-the-joint-use-of-tcp-and-network-coding-00>>.

[TETRYS]

Detchart, J., Lochin, E., Lacan, J., and V. Roca, "Tetrys, an On-the-Fly Network Coding protocol", Work in Progress, Internet-Draft, draft-detchart-nwcrg-tetrys-08, 17 October 2021, <<https://datatracker.ietf.org/doc/html/draft-detchart-nwcrg-tetrys-08>>.

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.

Authors' Addresses

Nicolas Kuhn

CNES

Email: nicolas.kuhn.ietf@gmail.com**Emmanuel Lochin**

ENAC

Email: emmanuel.lochin@enac.fr

François Michel

UCLouvain

Email: francois.michel@uclouvain.be**Michael Welzl**

University of Oslo

Email: michawe@ifi.uio.no