TCP with dynamic FEC For High Delay and Lossy Networks

Simone Ferlin and Ozgu Alay
Simula Research Laboratory, Norway

TCP: Loss detection and recovery - Introduction

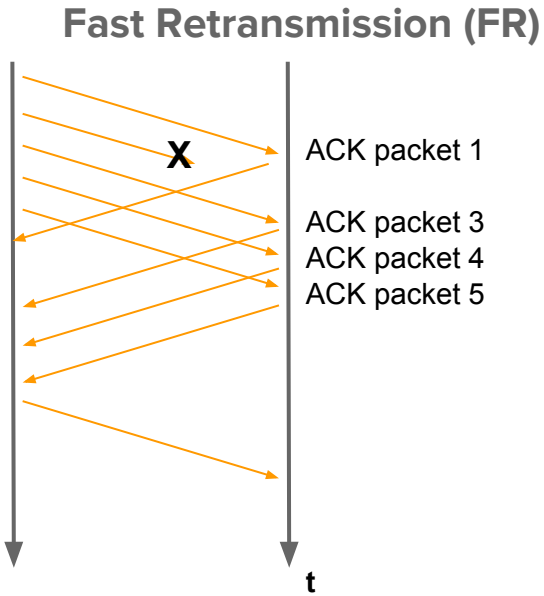
- TCP has been regularly changed over the past years to address performance enhancement under diverse network conditions.

TCP: Loss detection and recovery - Introduction

- TCP has been regularly changed over the past years to address performance enhancement under diverse network conditions.
  - TCP loss recovery mechanisms remain strictly Round Trip Time (RTT) dependent.

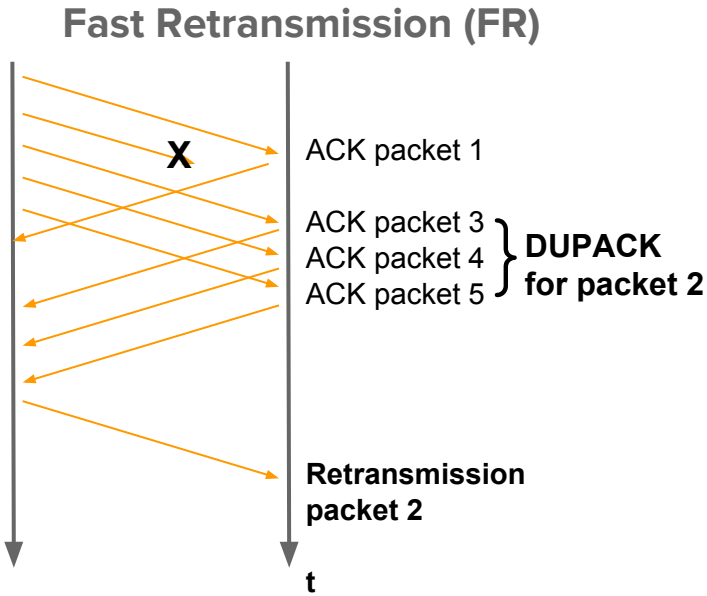
TCP: Loss detection and recovery - Introduction

- TCP has been regularly changed over the past years to address performance enhancement under diverse network conditions.
  - TCP loss recovery mechanisms remain strictly Round Trip Time (RTT) dependent.



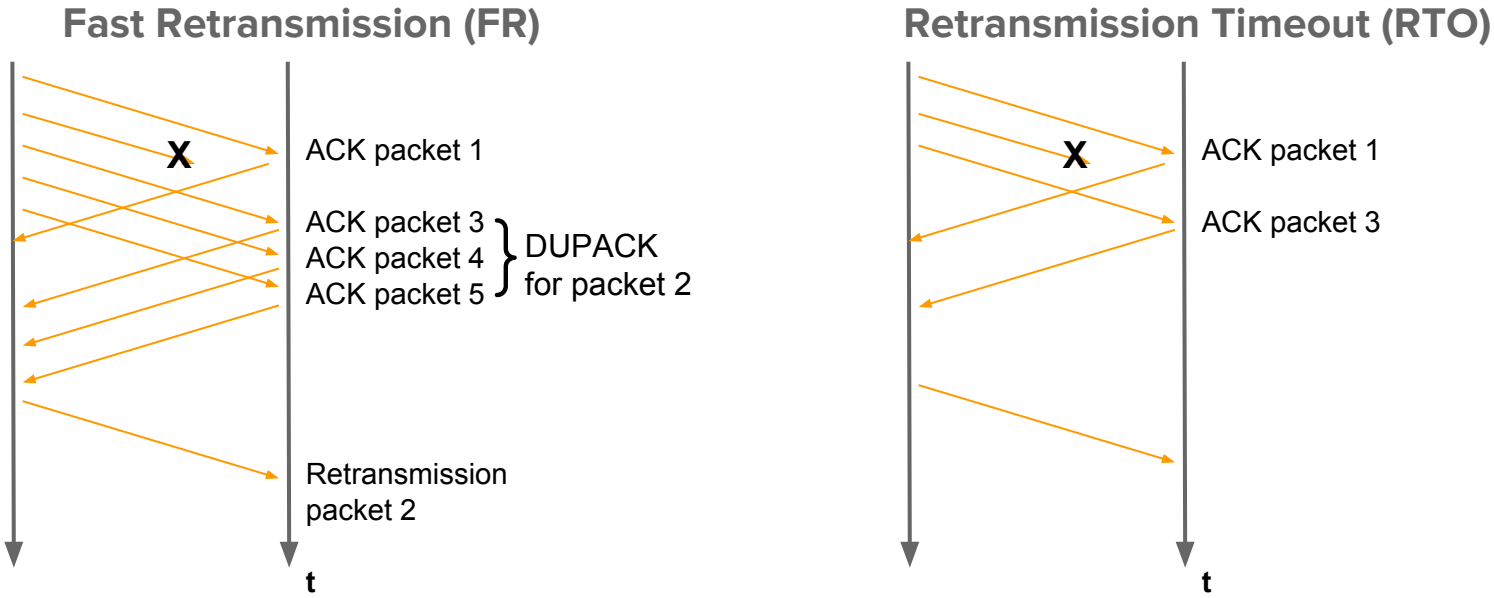
TCP: Loss detection and recovery - Introduction

- TCP has been regularly changed over the past years to address performance enhancement under diverse network conditions.
  - TCP loss recovery mechanisms remain strictly Round Trip Time (RTT) dependent.



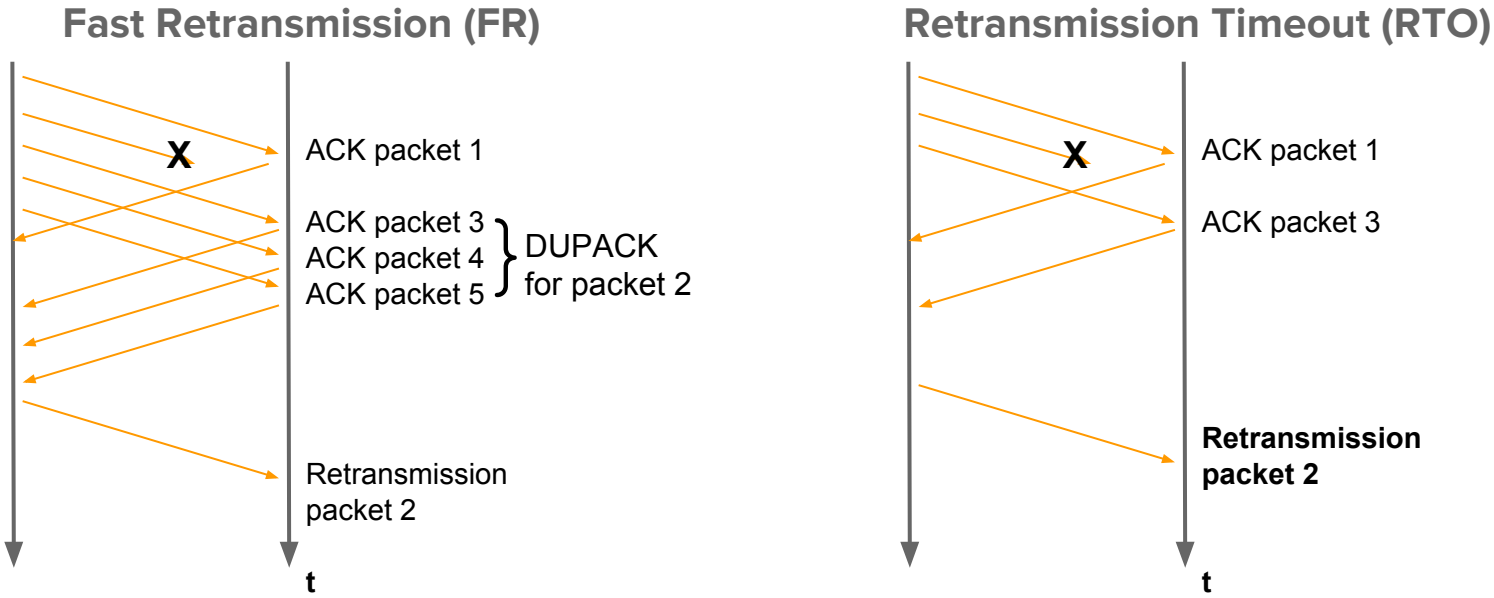
TCP: Loss detection and recovery - Introduction

- TCP has been regularly changed over the past years to address performance enhancement under diverse network conditions.
  - TCP loss recovery mechanisms remain strictly Round Trip Time (RTT) dependent.



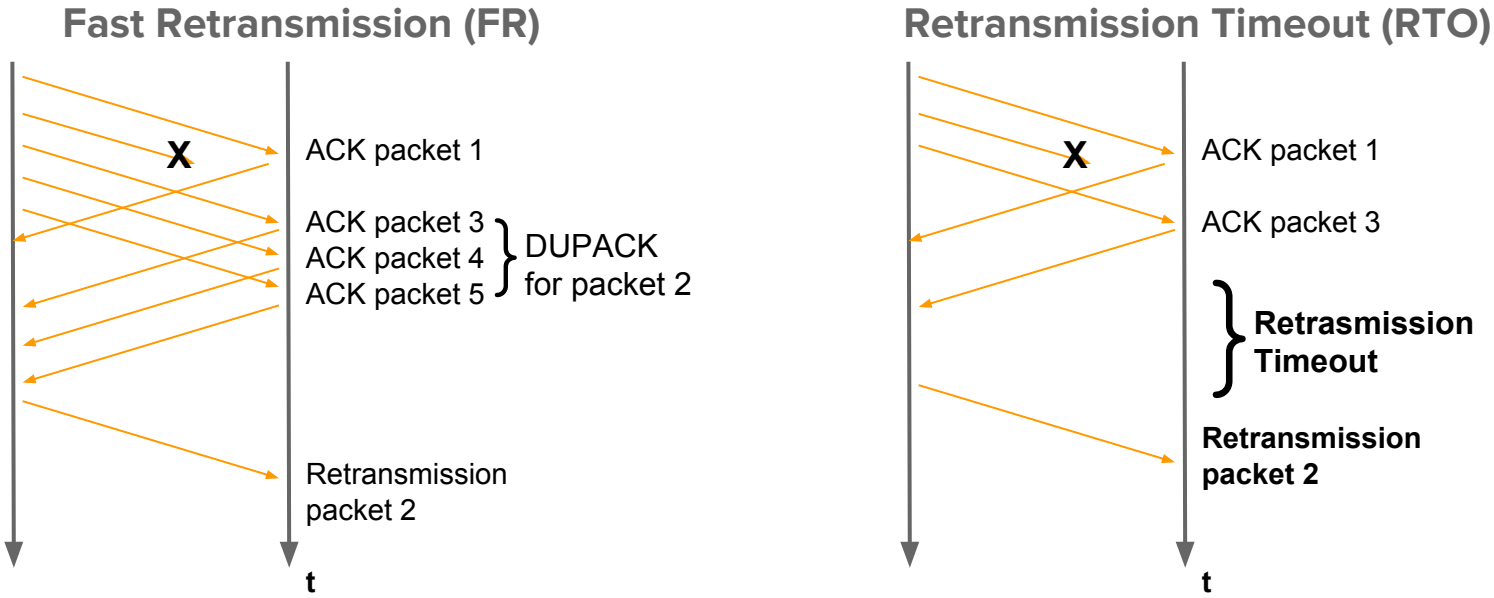
TCP: Loss detection and recovery - Introduction

- TCP has been regularly changed over the past years to address performance enhancement under diverse network conditions.
  - TCP loss recovery mechanisms remain strictly Round Trip Time (RTT) dependent.



TCP: Loss detection and recovery - Introduction

- TCP has been regularly changed over the past years to address performance enhancement under diverse network conditions.
  - TCP loss recovery mechanisms remain strictly Round Trip Time (RTT) dependent.



TCP: Loss detection and recovery - Motivation

- TCP performance over high delay networks is known to be suboptimal.
  - Neworks like these are cellular networks, but also satellite connections;

TCP: Loss detection and recovery - Motivation

- TCP performance over high delay networks is known to be suboptimal.
  - Neworks like these are cellular networks, but also satellite connections;

- TCP's legacy loss detection and recovery remained mostly unchanged.
  - Specially in satellite connections, recovery time and retransmissions are expensive;
  - There are several congestion control algorithms, but no algorithm is suitable for all network;

TCP: Loss detection and recovery - Motivation

- TCP performance over high delay networks is known to be suboptimal.
  - Neworks like these are cellular networks, but also satellite connections;

- TCP's legacy loss detection and recovery remained mostly unchanged.
  - Specially in satellite connections, recovery time and retransmissions are expensive;
  - There are several congestion control algorithms, but no algorithm is suitable for all network;

- TCP-TLP and TCP-IR aid TCP loss detection and recovery - *zero-RTT*.
  - TCP-TLP duplicates data towards the end of the flow to aid short-flows and avoid RTOs;
  - TCP-IR adds Forward Error Correction to TCP including it into the congestion control;

TCP: Loss detection and recovery - Motivation

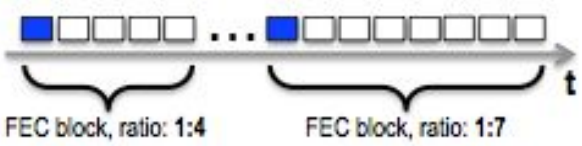
- TCP performance over high delay networks is known to be suboptimal.
  - Neworks like these are cellular networks, but also satellite connections;

- TCP's legacy loss detection and recovery remained mostly unchanged.
  - Specially in satellite connections, recovery time and retransmissions are expensive;
  - There are several congestion control algorithms, but no algorithm is suitable for all network;

- TCP-TLP and TCP-IR aid TCP loss detection and recovery - *zero-RTT*.
  - TCP-TLP duplicates data towards the end of the flow to aid short-flows and avoid RTOs;
  - **TCP-IR adds Forward Error Correction to TCP including it into the congestion control;**

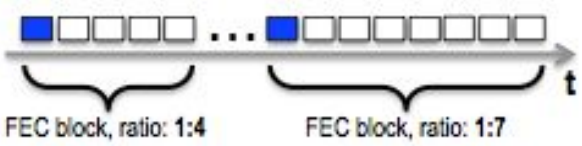
TCP: Forward Error Correction (FEC)

- TCP-IR applies XOR-FEC systematically. It keeps TCP segment structure:



TCP: Forward Error Correction (FEC)

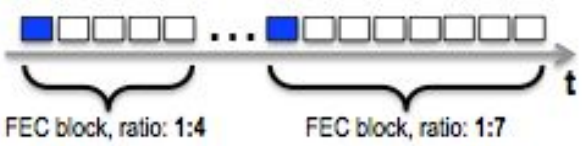
- TCP-IR applies XOR-FEC systematically. It keeps TCP segment structure:



- We build upon TCP-IR using XOR-FEC and extend it to **dynamic FEC (dFEC)**:
  - **TCP with FEC has to respect the Congestion Window (CWND), i.e. congestion control:**

TCP: Forward Error Correction (FEC)

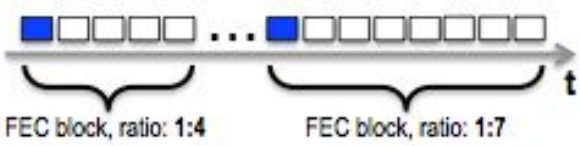
- TCP-IR applies XOR-FEC systematically. It keeps TCP segment structure:



- We build upon TCP-IR using XOR-FEC and extend it to **dynamic FEC (dFEC)**:
  - **TCP with FEC has to respect the Congestion Window (CWND), i.e. congestion control:**
    - Reduce the CWND, e.g., CWND/2, if: 1) FEC is lost or 2) FEC fails;
    - Increase the CWND, e.g., CWND=+1, if 1) FEC succeeds;

TCP: Forward Error Correction (FEC)

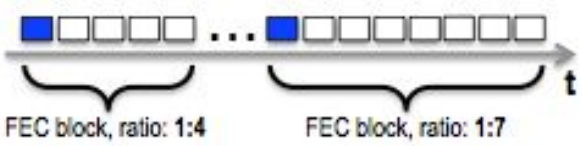
- TCP-IR applies XOR-FEC systematically. It keeps TCP segment structure:



- We build upon TCP-IR using XOR-FEC and extend it to **dynamic FEC (dFEC)**:
  - **TCP with FEC has to respect the Congestion Window (CWND), i.e. congestion control:**
    - Reduce the CWND, e.g., CWND/2, if: 1) FEC is lost or 2) FEC fails;
    - Increase the CWND, e.g., CWND=+1, if 1) FEC succeeds;
  - **TCP with FEC has to adapt FEC dynamically, according to the network conditions:**

TCP: Forward Error Correction (FEC)

- TCP-IR applies XOR-FEC systematically. It keeps TCP segment structure:



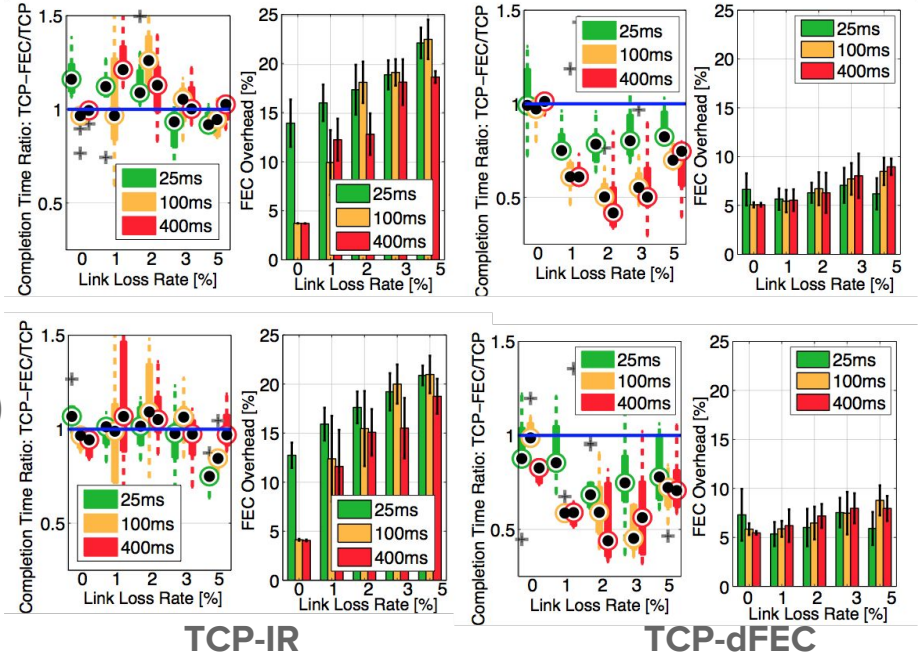
- We build upon TCP-IR using XOR-FEC and extend it to **dynamic FEC (dFEC)**:
  - **TCP with FEC has to respect the Congestion Window (CWND), i.e. congestion control:**
    - Reduce the CWND, e.g., CWND/2, if: 1) FEC is lost or 2) FEC fails;
    - Increase the CWND, e.g., CWND=+1, if 1) FEC succeeds;
  - **TCP with FEC has to adapt FEC dynamically, according to the network conditions:**
    - Steer ***residual losses*** over a period **T**;
    - ***residual losses*** over **N** is taken and compared against a ***target residual loss rate***;
    - The FEC ratio is reduced or increased, following the target, with **1±δ.**

Results for Web Traffic

- TCP-IR vs. TCP-dFEC with HTTP/2.0:
  - 25, 100 and 400 ms RTT
  - Random loss from 0, 1, 2, 3 and 5%

Google (1.080 KiB):

YouTube (3.204 KiB)

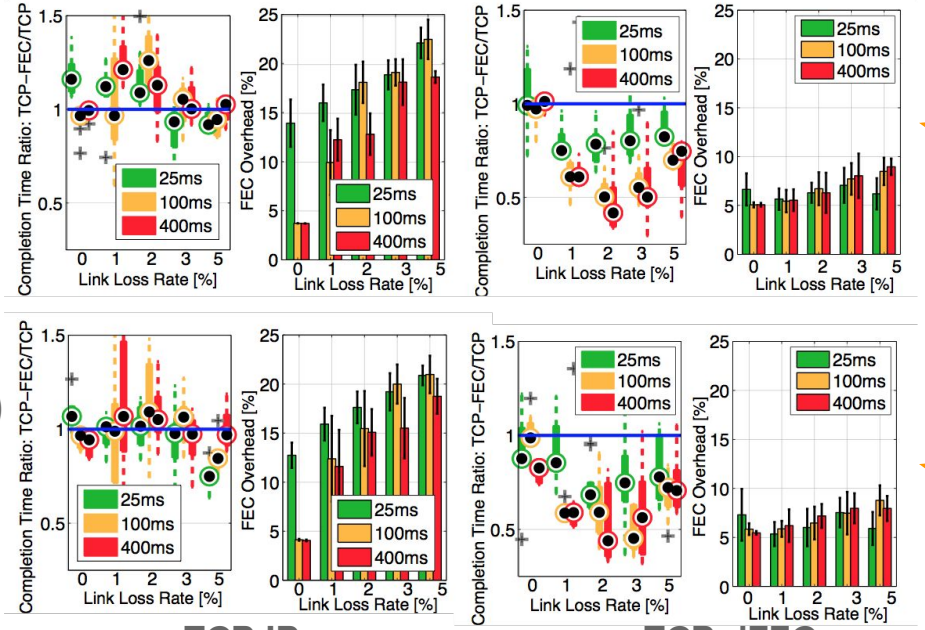


Results for Web Traffic

- TCP-IR vs. TCP-dFEC with HTTP/2.0:
  - 25, 100 and 400 ms RTT
  - Random loss from 0, 1, 2, 3 and 5%

Google (1.080 KiB):

YouTube (3.204 KiB)



TCP-dFEC reduces completion time by up to 50% by an FEC overhead of 10%

Conclusion

- TCP-dFEC is application agnostic and outperforms TCP-IR for various link conditions. For bulk traffic we observed improvements of up to 40%

- Our ongoing work considers extensive analysis of TCP-dFEC under different emulation and include real networks, with low packet loss.

- We plan to investigate how such a *zero-RTT* recovery framework can be incorporated in MPTCP in highly heterogeneous paths.

We keep repeating the question: Is it able to extend and modify TCP?