Forward Erasure Correction for Short-Message Delay-Sensitive QUIC Connection

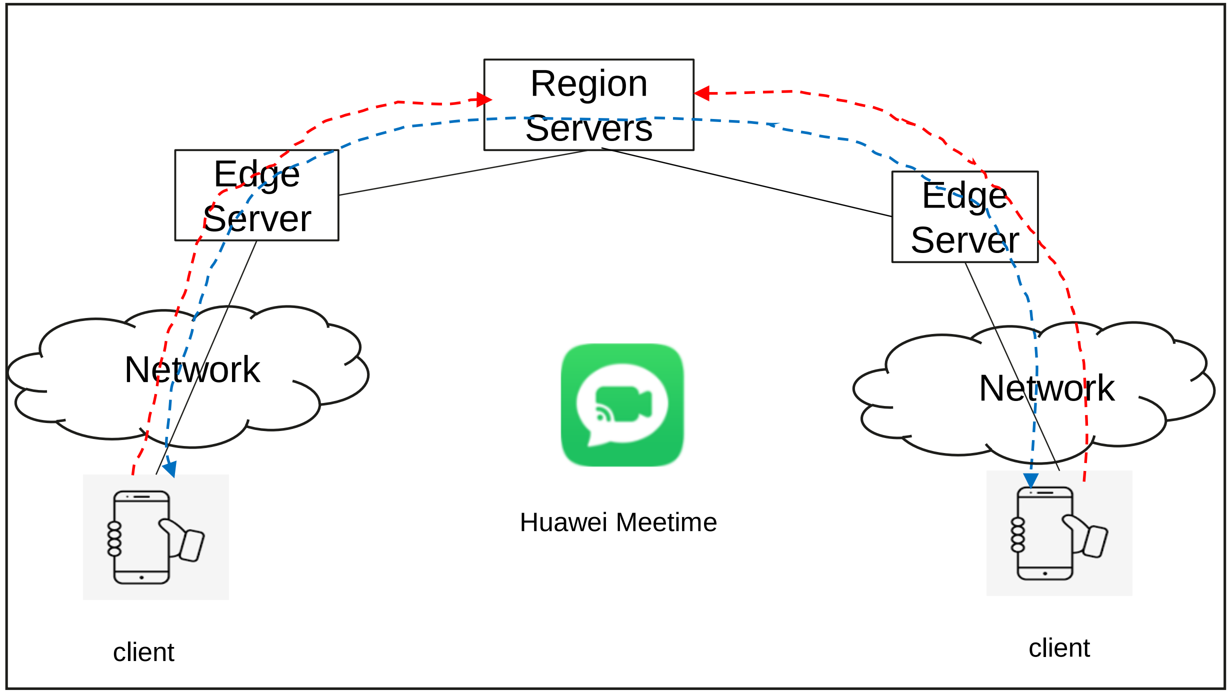
Dmitry Moskvitin,

Evgeny Onegin,

et al

HUAWEI

Background



The Architecture of RTC Services

Control data: Websocket over QUIC

Media data: RTP over UDP

Problem:

> When clients are in the bad network environment, e.g., unstable wireless signal, they will disconnect from the RTC service frequently.

- It has a bad QoE. Users have to often reconnect, and start it over.

> The problem is mainly due to **the loss of most control data**.

> Besides that, **small packets** during short sessions, e.g. user verification session, packets carrying certificates, passwords, fingerprints, emails, SMS codes, etc, will also be **affected by bad network conditions**.

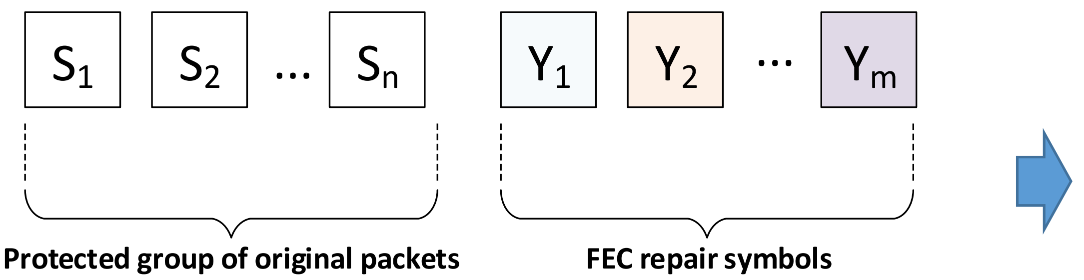
Requirement:

> **Delay sensitive control data should be delivered first, and try to avoid loss as much as possible.**

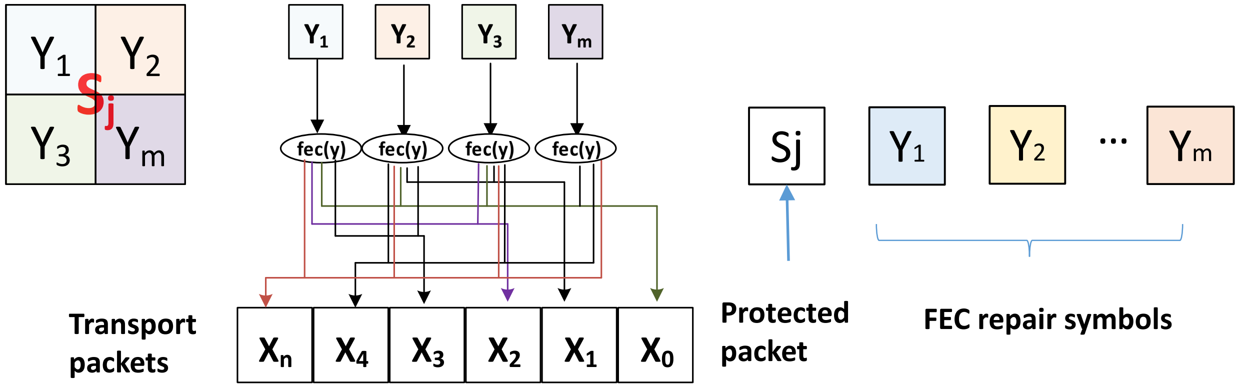
> **FEC is method appropriate for this scenario.**

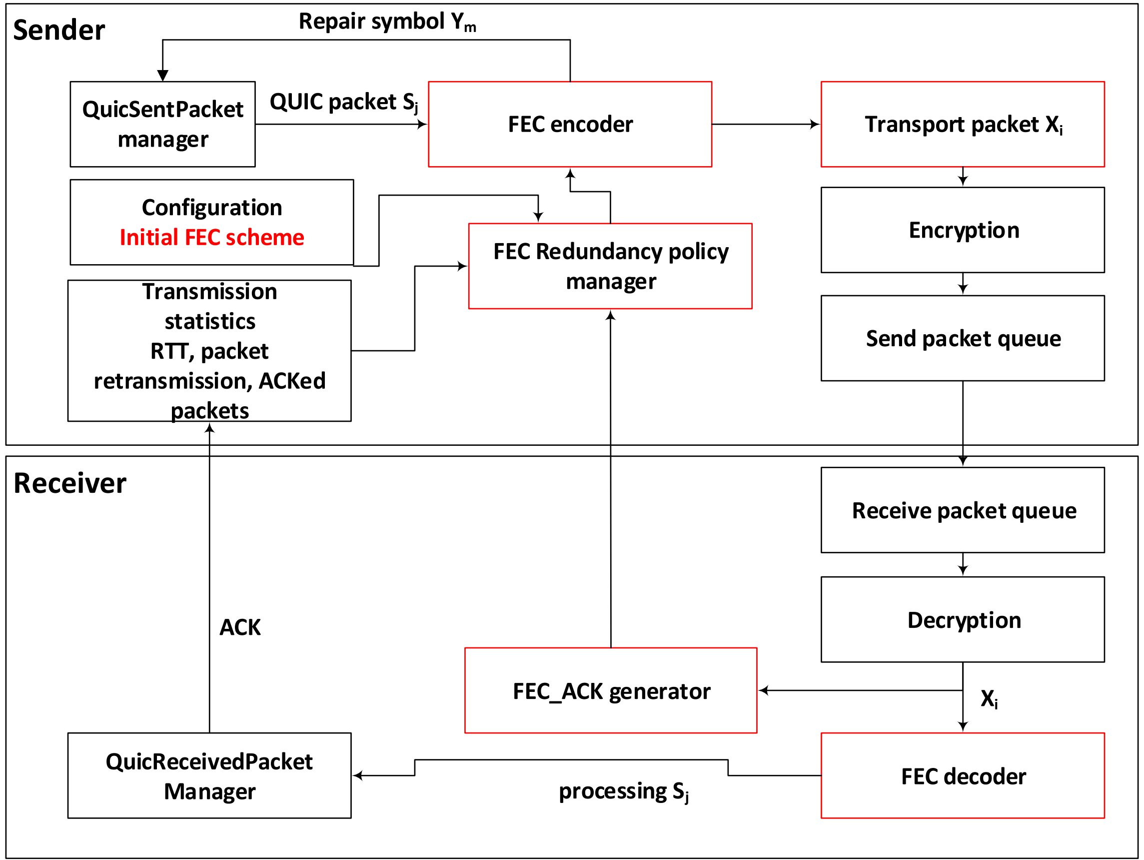
Proposed QUIC FEC Architecture

Traditional FEC



FEC for Short Messages






```
FEC_REPAIR {
    Type (i) = TBD,
    FEC Version (i),
    Packet Number Length (2),
    Protected Packet Number (8..32),
    FEC Meta Data (..),
    Reserved (i) = 0,
    FEC Payload (..)
}
FEC_ACK {
    Type (i) = TBD,
    FEC Latest Restored Packet Number (i),
    FEC Restored Bytes (i),
    FEC Restored Packets (i),
    FEC Restored Packet List Size (i),
    FEC Restored Packet Number (i) ...
}
```

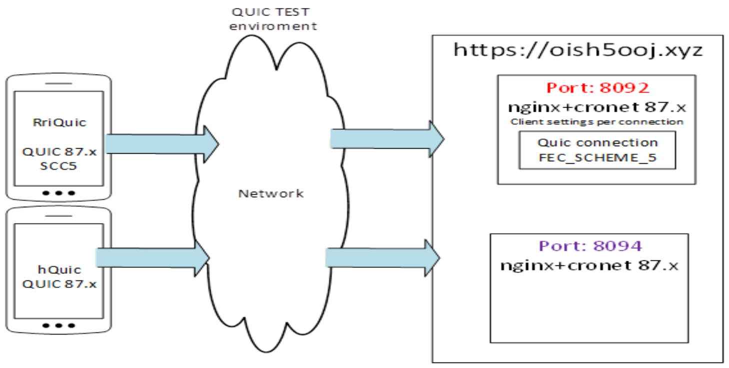
Advantage: It does not modify current QUIC design!

Results from the test

Test From Lab

Test conditions:

- netem server params: Ingress: 50ms delay X% packet loss, Egress: 50ms delay
- Transfer medium: Wi-Fi + cross country



In congested network, the overall performance of version 87 for small packets is improved by **12.25% to 34.04%** on average compared with that of the native version 87.

File size	Protocol type	Packet loss ratio 0% avg e2e time(ms)	Packet loss ratio 3% avg e2e time(ms)	Packet loss ratio 7% avg e2e time(ms)
1KB	Cronet 87 + FEC	296. 81	400. 495	307.99
	Cronet 87	332. 68	452. 92	387. 83
	Enhancemnts	10. 78%	11. 57%	20. 59%
8KB	Cronet 87 + FEC	317. 09	458. 907	320.83
	Cronet 87	332. 99	557. 29	447. 93
	Enhancemnts	4. 77%	17. 65%	28. 37%
16KB	Cronet 87 + FEC	314. 21	503. 321	329.2
	Cronet 87	335. 68	734. 09	409. 85
	Enhancemnts	6. 40%	31. 44%	19. 68%
1M	Cronet 87 + FEC	1091. 2	10158. 519	1607.68
	Cronet 87	1215. 08	15472. 02	2238. 35
	Enhancemnts	10. 20%	34. 34%	28. 18%
2M	Cronet 87 + FEC	1913. 41	19259. 726	2943.16
	Cronet 87	2058. 37	28520. 84	3522. 01
	Enhancemnts	7. 04%	32. 47%	16. 44%
5M	Cronet 87 + FEC	4413. 24	41864. 547	7350.2
	Cronet 87	4511. 16	70281. 02	8993. 82
	Enhancemnts	2. 17%	40. 43%	18. 27%

Data From Implemented Products

Results from Huawei Browser in weak network environment

(10% loss rate, 50ms delay)

Webpage	Metrics (ms)	QUIC without FEC	QUIC with FEC	Improvements
Cold Start	Page loading Time	2783	2651	4.64%
	Content Downloading	166	120	27.71%
Warm Start	Page loading Time	860	670	22.09%
	Content Downloading	750	563	24.93%

Improvement Results After QUIC FEC Implemented in Meetime

Metrics	Improvements
Call Completion Rate	5.71%
Call Completing ratio in 5s	6.34%
Average Delay to Handle in weak network	60.5%
Call Lost Rate in weak network	0.7%

Issues to be discussed

- Does short message have to be protected by FEC or not?
  - > In WAN scenario, RTT is relatively large, and there would be both random loss and burst loss in the network.
  - > If relying on retransmission, delay is a problem; If sending duplicate packets, it is difficult to handle burst loss and waist more bandwidth than FEC.
  - > On the basis of above considerations, we think FEC is a good solution in all respects.
- How does FEC deal with the network congestions?
  - > In order to achieve maximum efficiency, sender should adaptively change FEC redundancy according to network conditions. E.g., in the case of our websocket long connections, FEC redundancy can be more accurately adjusted based on the network information collected from the connections.
- Is it able to support large data transaction?
  - > Yes, can use packet number range to indicate multiple source symbol packets.