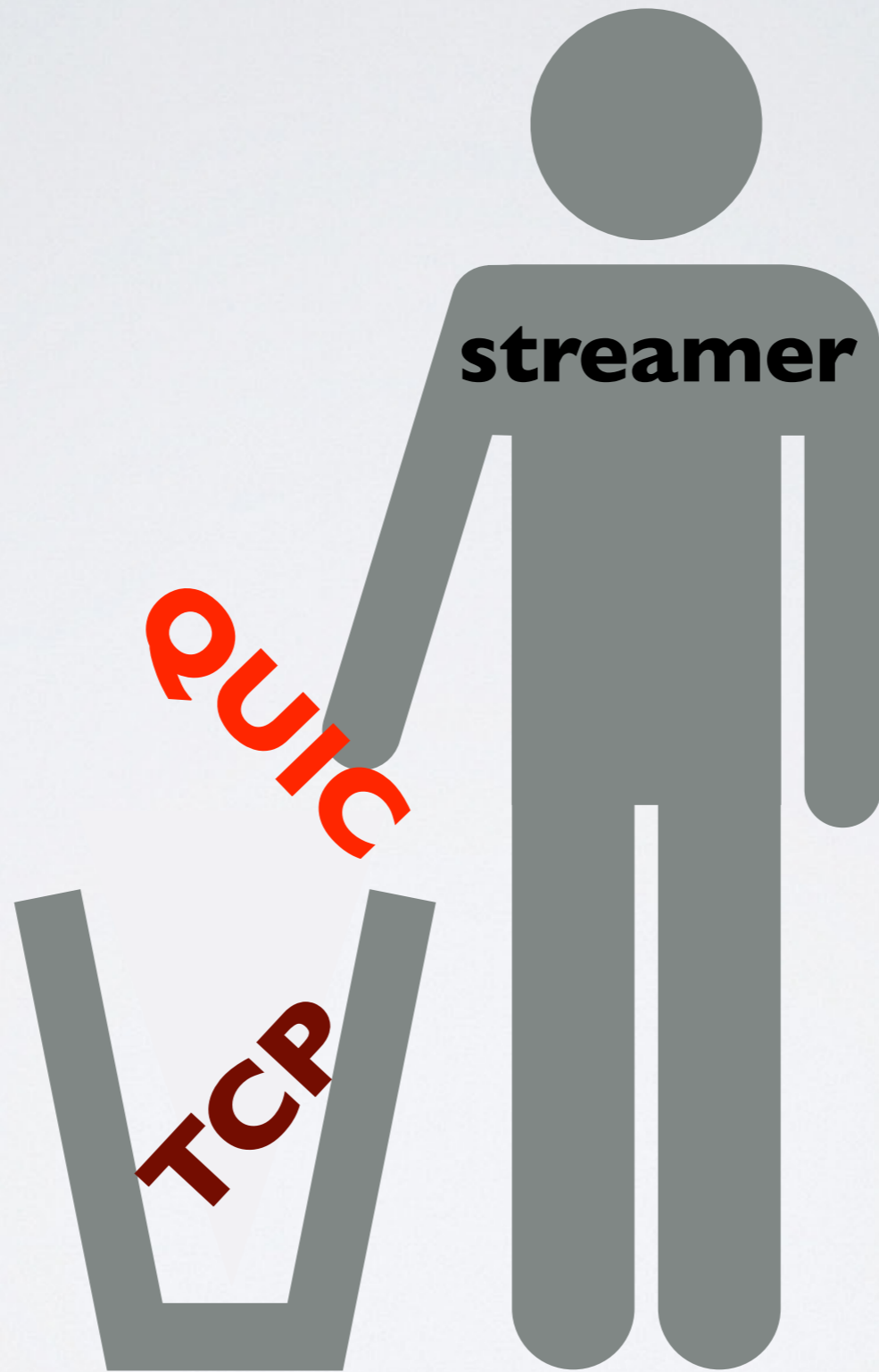




REAL-TIME MEDIA OVER QUIC

Mirko Palmer





STREAMING LANDSCAPE

- Video is the largest contributor to Internet traffic with **75%** by 2017^[1]
- Predominantly transported via CDNs hence HTTP over TCP
 - “Intelligence” has to be on the client side

[1] Cisco Visual Networking Index: Forecast and Trends, 2017–2022

QUIC LANDSCAPE

- QUIC contributing already 6% of the Internet traffic^[2]
 - Google delivers more than 40% of its content via QUIC^[3]
- CDNs support QUIC^[4]

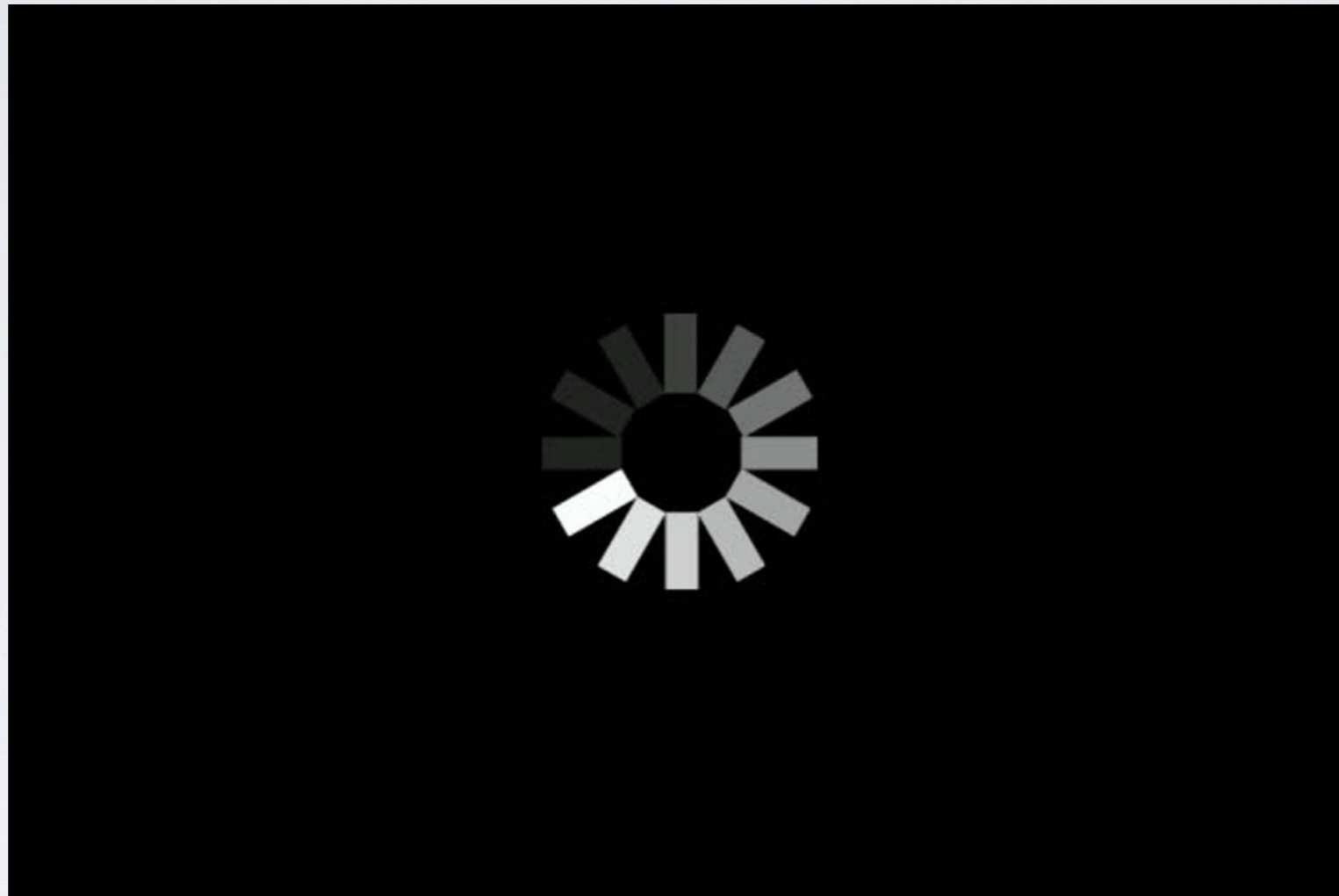
[2] The QUIC Transport Protocol: Design and Internet-Scale Deployment. In Proceedings of ACM SIGCOMM '17, 2017

[3] A First Look at QUIC in the Wild. In Passive and Active Measurement, 2018

[4] Akamai: Introducing QUIC for Web Content, Oct. 2018



WHY?



QUIC still (only) allows reliable transfers!

QUIC inherits TCP's problems

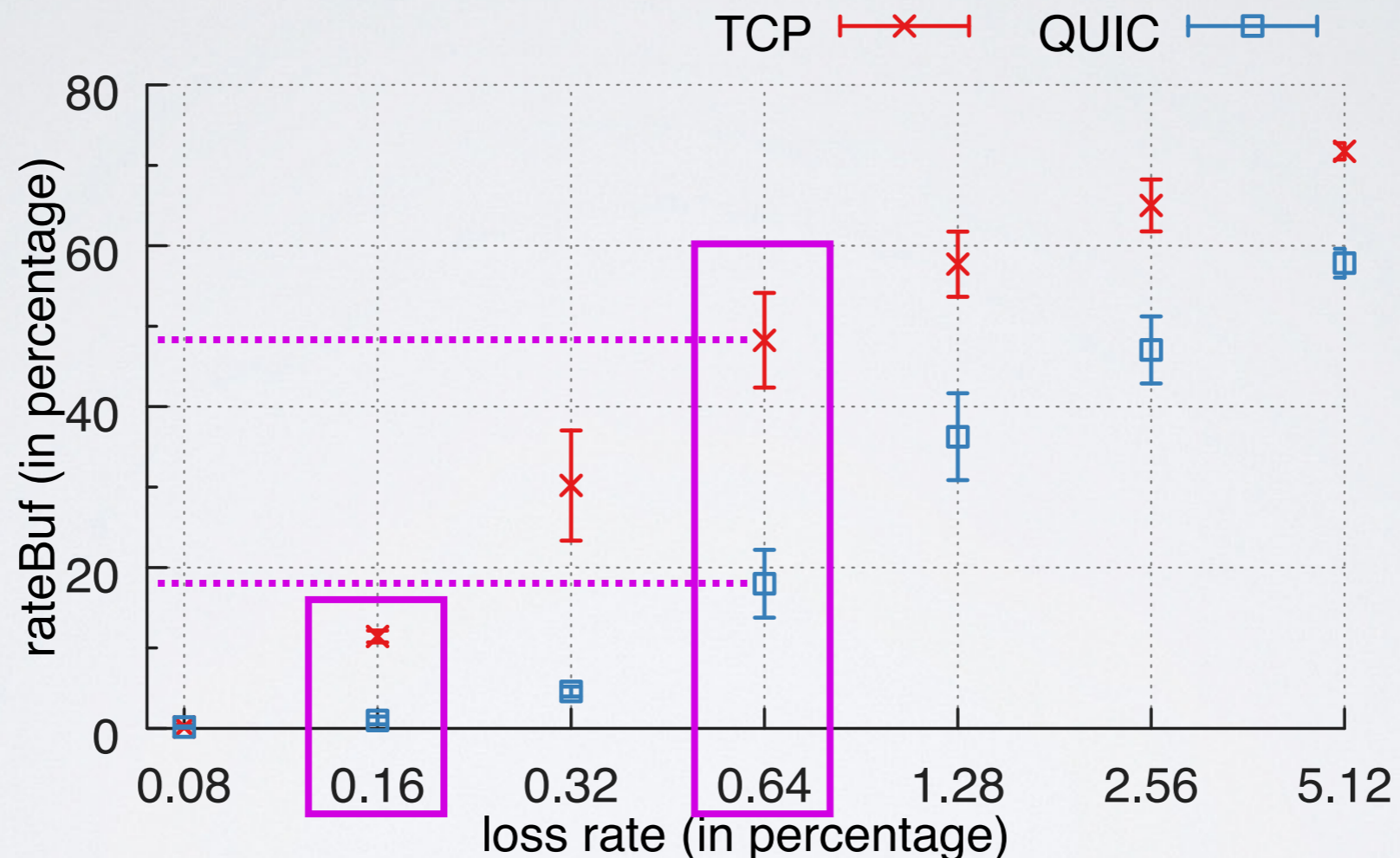
STREAMING EXPERIMENT

Setup



STREAMING EXPERIMENT

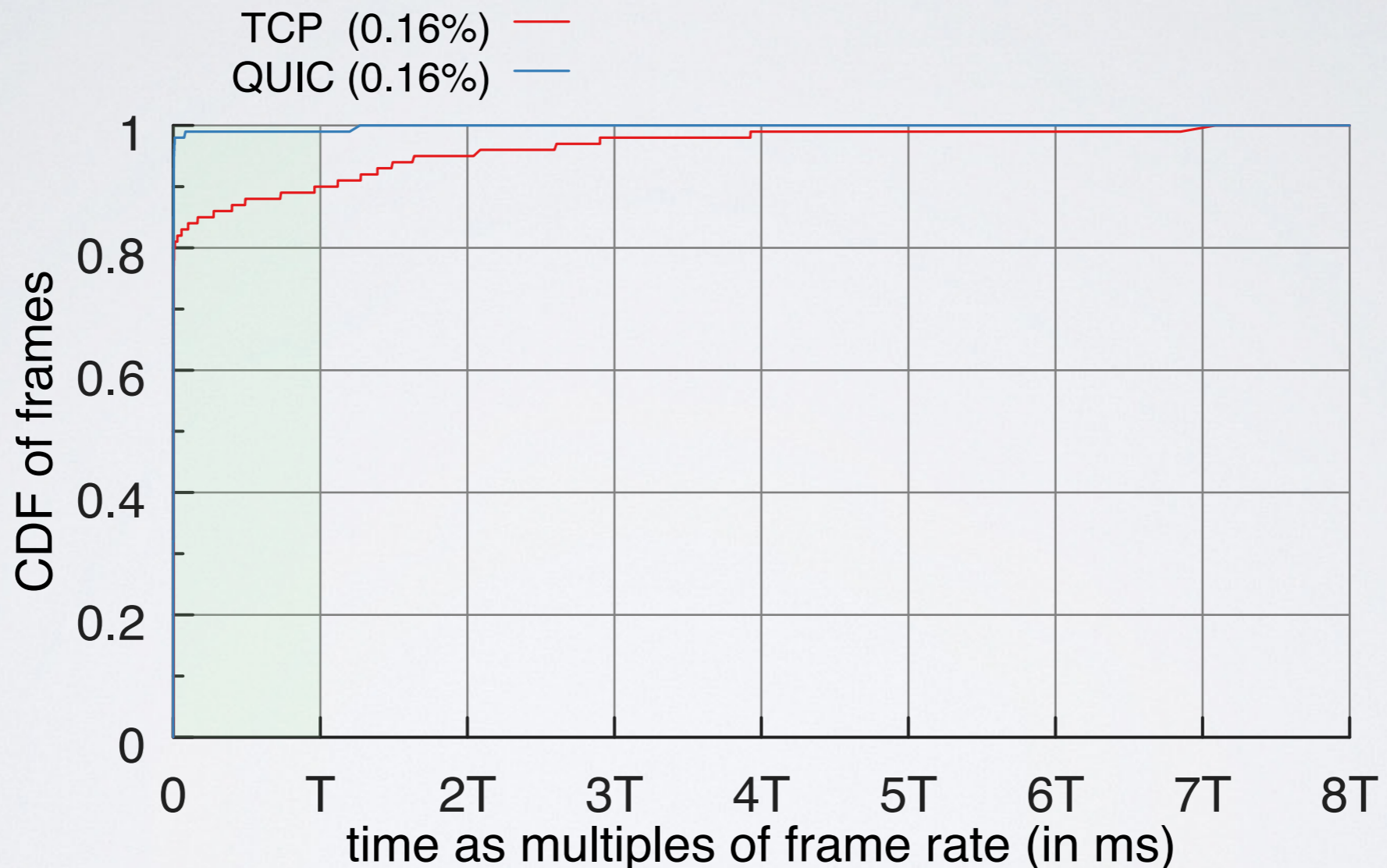
Stalling Events



- TCP has terrible performance even under low loss rates
- QUIC **should** solve stalling issues, yet it doesn't

STREAMING EXPERIMENT

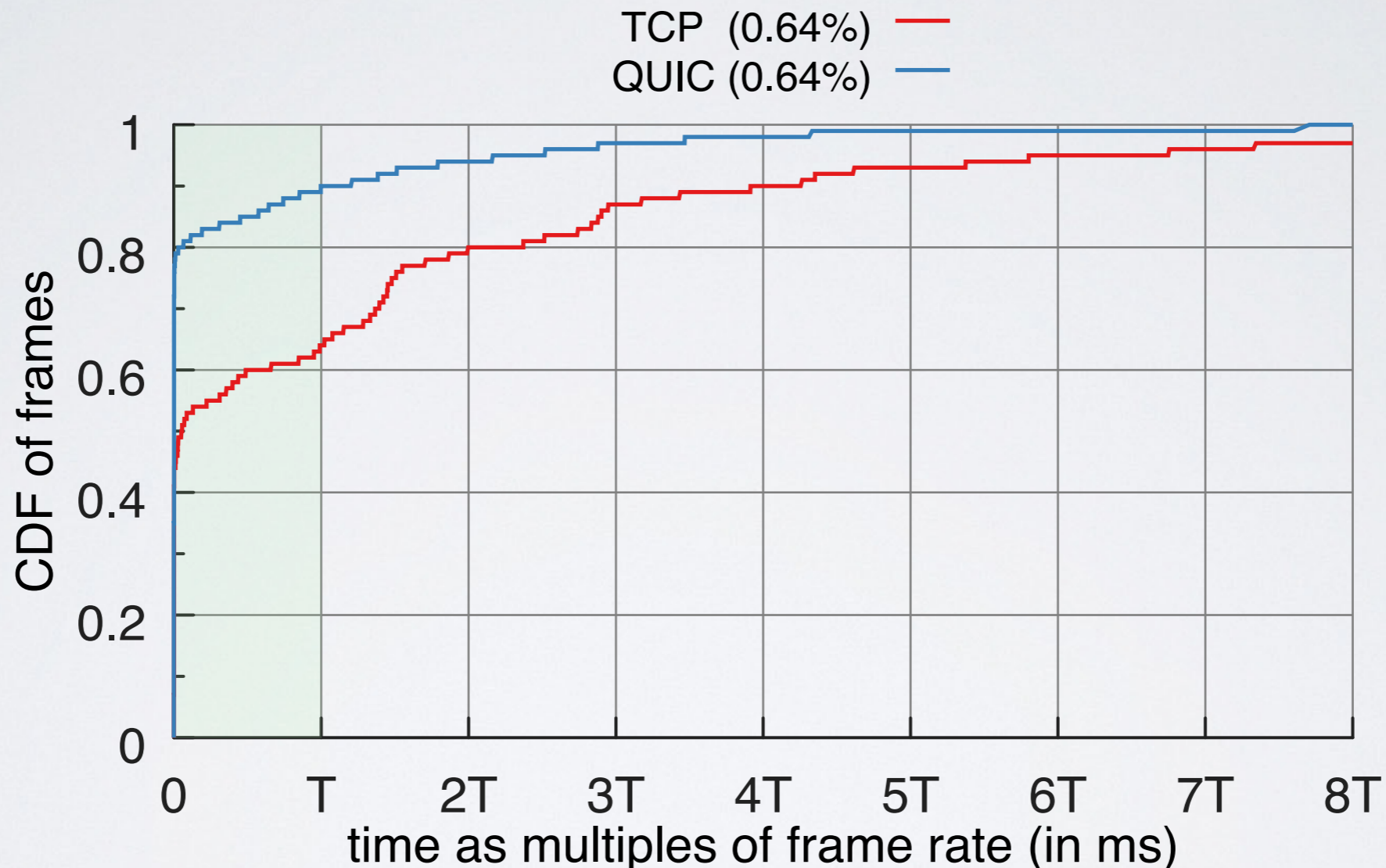
Stalling Durations



- TCP suffers even at very low loss rate
- QUIC performance quite good

STREAMING EXPERIMENT

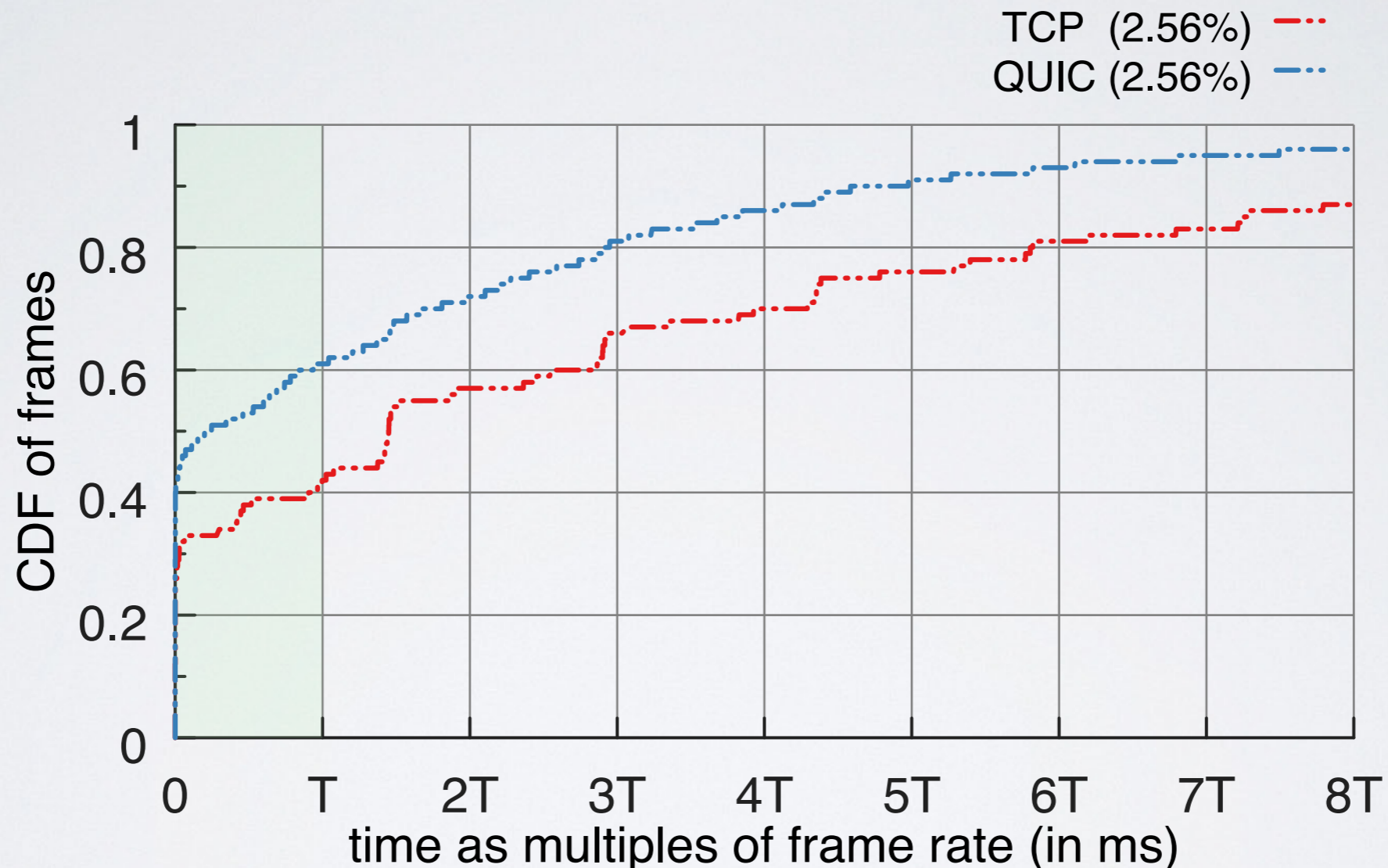
Stalling Durations



- QUIC performance degrades quickly with increased loss
- TCP suffers **even more**

STREAMING EXPERIMENT

Stalling Durations



- TCP and QUIC **practically unusable** under high loss

VIDEO STREAMING

TCP

- TCP encounters Head-of-Line blocking
 - Retransmitted data often too late to be useful
- TCP is a reliable transport protocol
 - Unavoidable retransmission of outdated data

VIDEO STREAMING

QUIC

- **QUIC** encounters HoL blocking (on chunk basis)
 - Retransmitted data often too late to be useful
- **QUIC** is a reliable transport protocol
 - Unavoidable retransmission of outdated data

QUIC inherits TCP's problems

THE QUIC FIX

Unreliable Streams

- No retransmissions—instead send new data
- Do not touch QUIC's ACK mechanism
- Unreliable streams are flow- and congestion-controlled
- Capability has to be negotiable during handshake
- Our fix is an **extension** not a modification

OTHER FIXES

Partially Reliable Message Streams for QUIC

- IETF draft by Akamai folks
- Server indicates expired stream offset
 - ‘Will not retransmit data “older” than given offset’
- Offset is given by server side
 - Challenging for client to specify retransmission offsets

OTHER FIXES

An Unreliable Datagram Extension to QUIC

- IETF draft by Google & Apple folks
- Add UDP like streams to QUIC
- Similar to our approach but not flow-controlled
 - No congestion control for UDP-like frames
 - No Indication on stream end

OTHER FIXES

Real-time Audio-Visual Media Transport over QUIC

- Work by Colin Perkins and Jörg Ott
- Map RTP functionality to the QUIC world
- Introduce new frame type RT_Streams
- Different approach—more suited for video conferencing

OUR QUIC FIX

What do we do differently?

OUR QUIC FIX

What do we do differently?

We try to modify QUIC as little as possible!

OUR QUIC FIX

Unreliable Streams

- Reuse StreamID space for ease of implementation
 - Using different Frame type is a good idea
- Indicate end-of-stream explicitly
 - Why is that a good thing?

BEYOND THE FIX

Challenges - Multiplexing

- Transmitting video completely unreliably is undesirable



BEYOND THE FIX

Challenges - Multiplexing

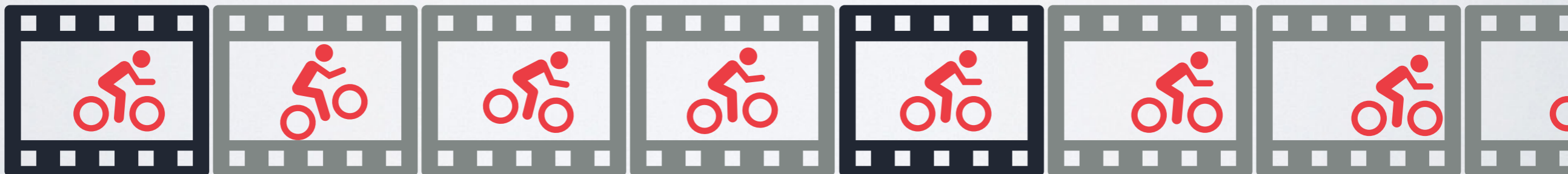
- Transmitting video completely unreliably is undesirable
- How do you decide which part needs reliability?



BEYOND THE FIX

Challenges - Multiplexing

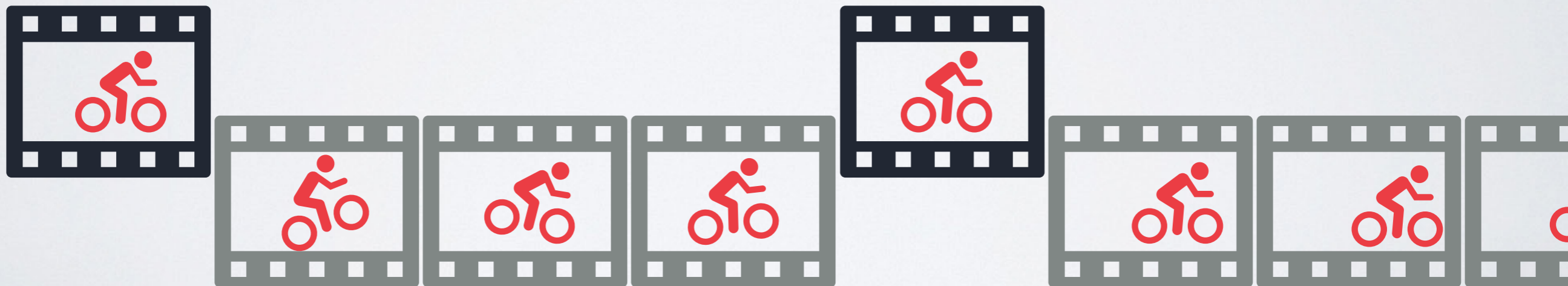
- One approach: Transfer Key Frames reliably
 - Rest of the video is less “important”



BEYOND THE FIX

Challenges - Multiplexing

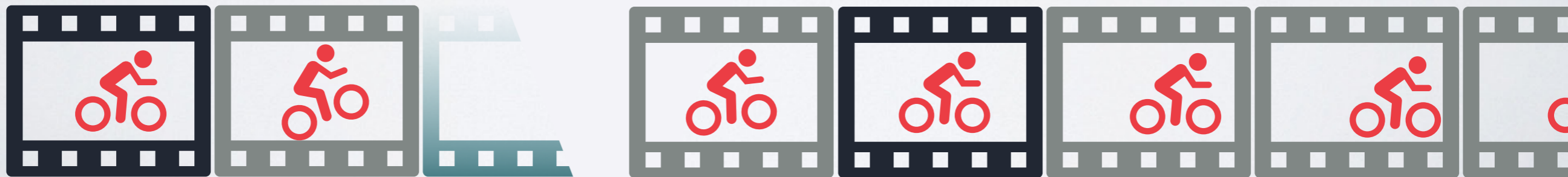
- One approach: Transfer Key Frames reliably
 - Rest of the video is less “important”



BEYOND THE FIX

Challenges - Filling Holes

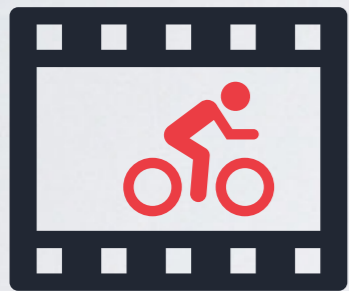
- How do we fix the “holes” in lossy QUIC streams?
 - Tell the application that there was a hole
 - Mask the hole with padding data



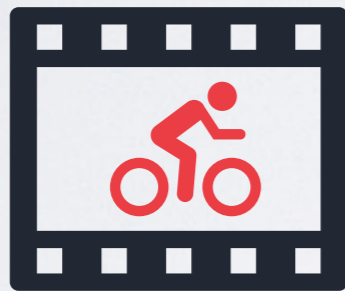
BEYOND THE FIX

Challenges - Demultiplexing Streams

- How do we reassemble the video?
- QUIC streams are not “related” to each other



frame 1



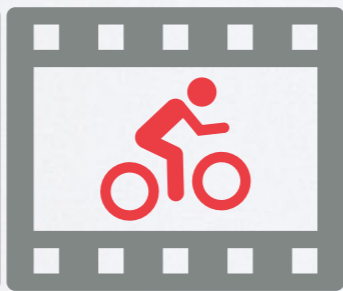
frame 5



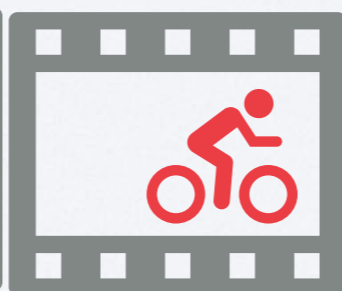
2



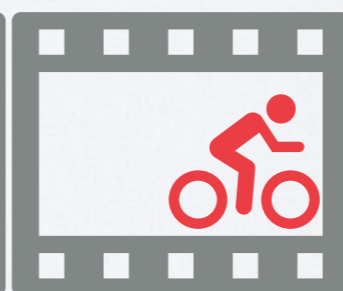
3



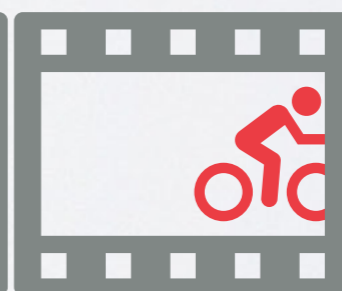
4



6



7

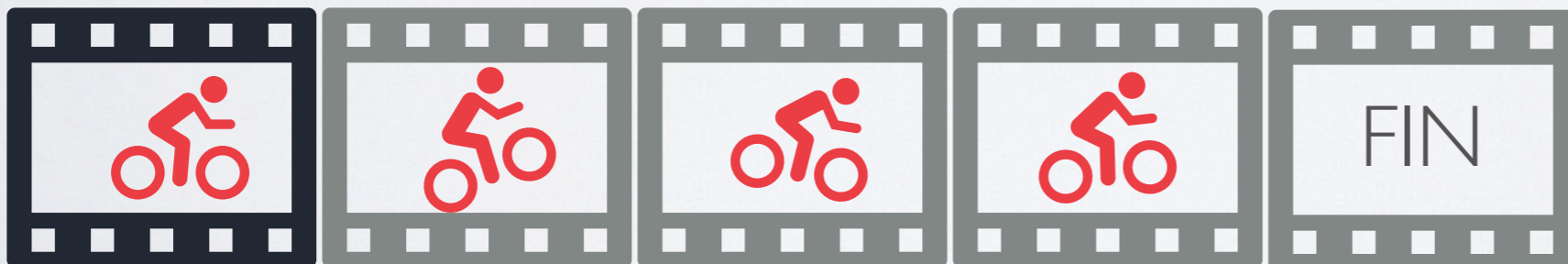


8

BEYOND THE FIX

Challenges - the FIN bit

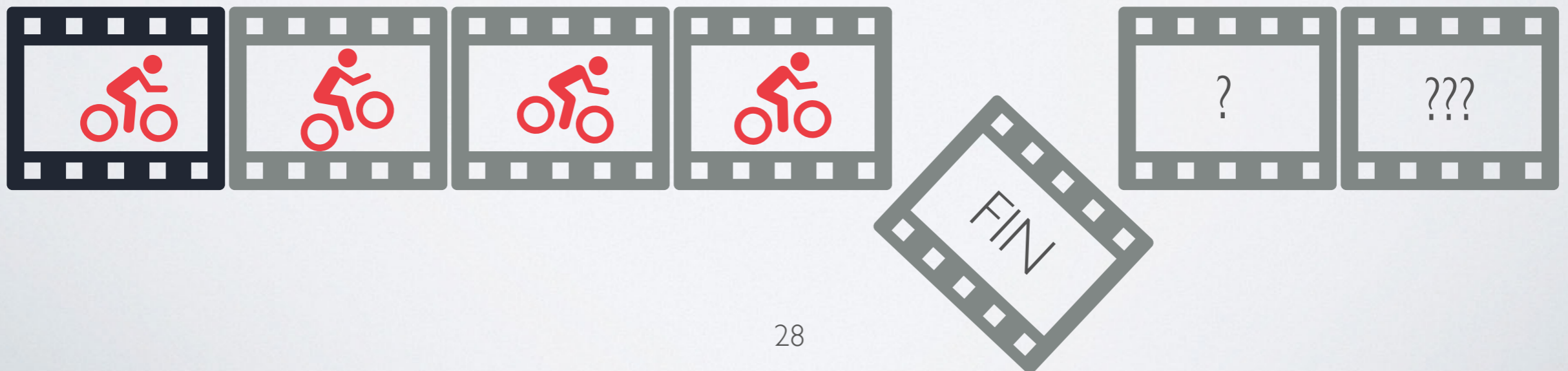
- When can we close a stream?
 - Reception of the FIN bit!



BEYOND THE FIX

Challenges - the FIN bit

- What if the frame containing the FIN is lost?
 - Without modifying QUIC: just send that frame reliably!



SUMMARY & OUTLOOK

- Streaming video over vanilla QUIC is not the solution
- Introducing partial reliable transfers for video is non-trivial
- Extending QUIC with unreliable streams is one solution
- A full end-to-end setup is part of ongoing work