Coding for low latency (and reliability): Motivation
Coding for low latency

Requirements for low latency and reliability are application driven:

- Video conferencing (150 ms)
- AR/VR (7ms)
- Telesurgery w/ haptic feedback (5 ms)
- Cooperative driving (10 ms)
The distance from Stanford to Boston is ~4320km. The speed of light in vacuum is $300 \times 10^6$ m/s. The speed of light in fibre is roughly 66% of the speed of light in vacuum. The speed of light in fibre is $300 \times 10^6$ m/s $\times 0.66 = 200 \times 10^6$ m/s. The one-way delay to Boston is $4320$ km / $200 \times 10^6$ m/s = 21.6ms.

Related blog:

Coding for low latency

Important facts about latency!

The distance from Stanford to Boston is ~4320km. The speed of light in vacuum is 300 x 10⁶ m/s. The speed of light in fibre is roughly 66% of the speed of light in vacuum. The speed of light in fibre is
300 x 10⁶ m/s x 0.66 = 200 x 10⁶ m/s.
The one-way delay to Boston is
4320 km / 200 x 10⁶ m/s = 21.6ms.

Two facts about networking from Stuart Cheshire's famous rant “It’s the latency, stupid” [1]

#1 Once you have bad latency you are stuck with it
#2 Making more bandwidth is “easy” (we can use this to our advantage)

Two devices need to communicate **reliably with low latency**. How can we achieve this:

**#1 ARQ**: Trading latency for reliability with minimum bandwidth consumption

**#2 Coding**: Trading bandwidth for reliability with minimum latency increase
Coding for low latency

Visualizations:
Is coding the only answer?

- No we can also do other stuff: Edge computing, in path retransmissions (perhaps IETF LOOPS), etc.
- But, it definitely should be part of the toolbox?
Coding for low latency

Demonstration: Online video game streaming

https://youtu.be/Tln1FjK8KI4
Coding for low latency

Work at IETF:

- Others?

Neither mention coding :(
Thanks for you time!
Coding for low latency

Latency + reliability a tough challenge!

Using retransmissions to obtain reliability is latency costly.

Minimum 3x the link latency
Coding for losses + latency

Using coded packets long delays can be avoided. In the above example the receiver is able to recover from the two packet losses when receiving the second coded packet.

If delays are long (e.g. such as in satellite networks) the tradeoff between bandwidth and delay can make sense.

- Lost packets
- \( p_x \) denote the uncoded packet \( x \)
- \( \Sigma p \) denotes a coded packet (linear combination of other packets)
Coding for low-latency (head of line blocking)

$p_x$ denote the uncoded packet $x$

$\Sigma p$ denotes a coded packet (linear combination of other packets)