Coding for low latency (and reliability): Motivation
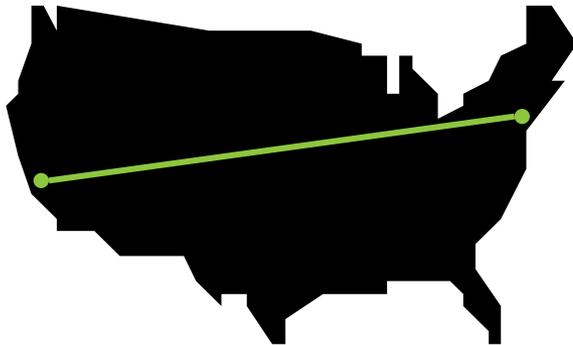
steinwurf

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)

Coding for low latency

Important facts about latency!
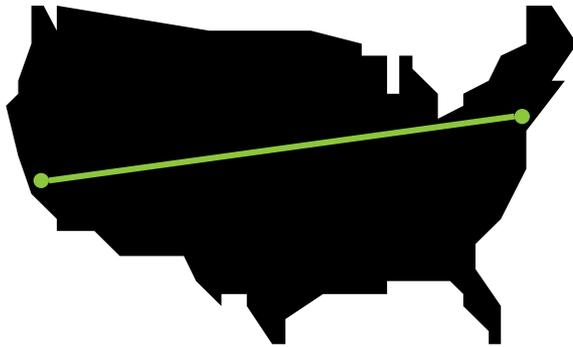

**4320 km == 21.6 ms
One way!**

The distance from Stanford to Boston is ~4320km.
The speed of light in vacuum is 300×10^6 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
300×10^6 m/s x 0.66 = 200×10^6 m/s.
The one-way delay to Boston is
4320 km / 200×10^6 m/s = 21.6ms.

Related blog:
http://steinwurf.com/blog/2018-06-01-delay.html

Coding for low latency

Important facts about latency!

**4320 km == 21.6 ms
One way!**

The distance from Stanford to Boston is ~4320km.
The speed of light in vacuum is 300×10^6 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
300×10^6 m/s x 0.66 = 200×10^6 m/s.
The one-way delay to Boston is
4320 km / 200×10^6 m/s = 21.6ms.

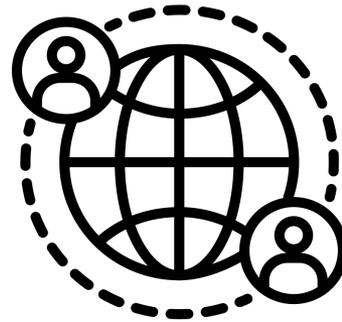Two facts about networking from Stuart Cheshire's famous rant "It's the latency, stupid" [1]

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

[1] http://www.stuartcheshire.org/rants/Latency.html

Coding for low latency

Our problem statement:

Two devices needs to communicate **reliably with low latency**. How can we achieve this:
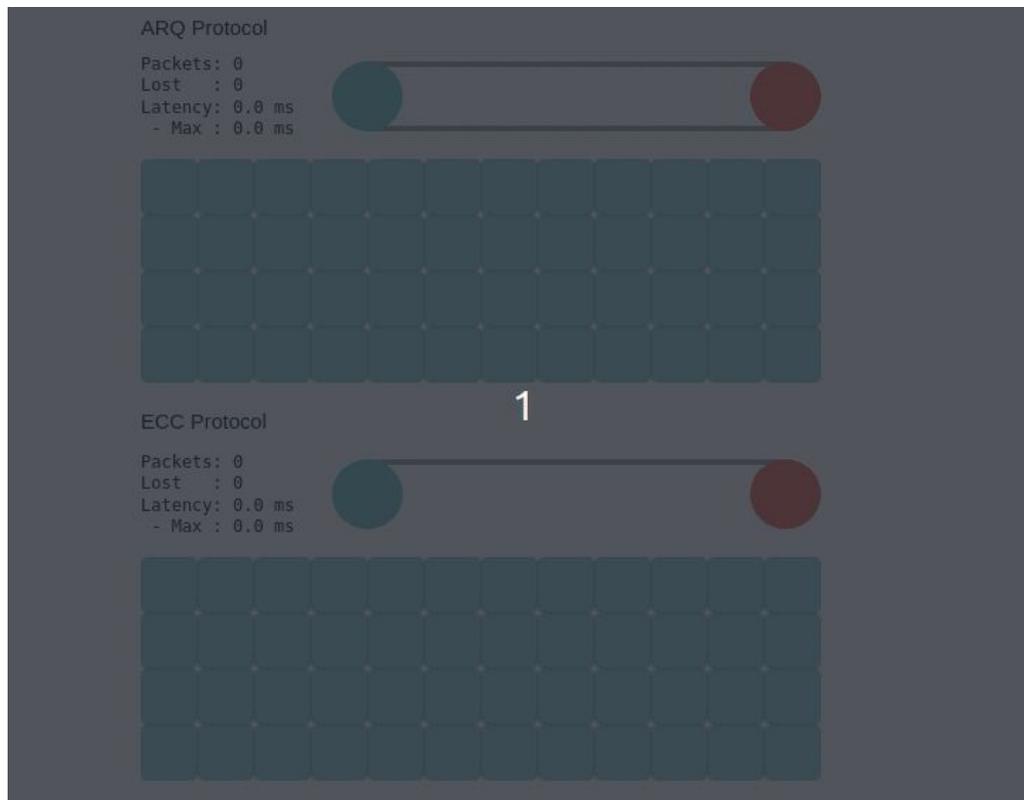
#1 ARQ: Trading latency for reliability with minimum bandwidth consumption

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

Coding for low latency

Visualizations:

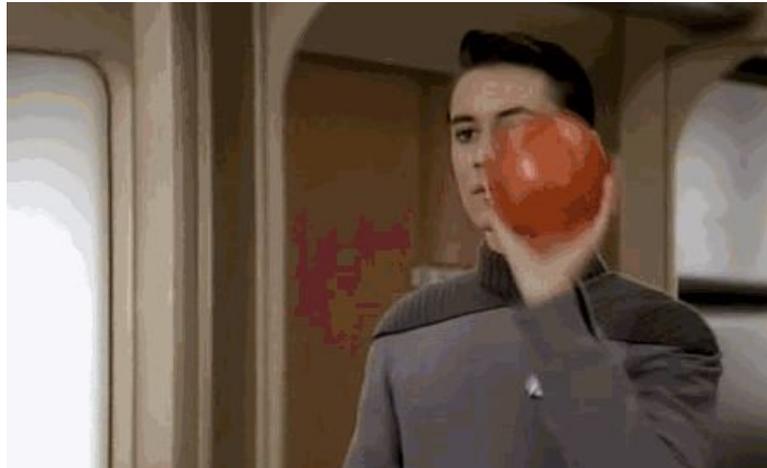http://steinwurf.com/blog/2018-06-01-delay.html

Coding for low latency

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

Coding for low latency

Work at IETF:

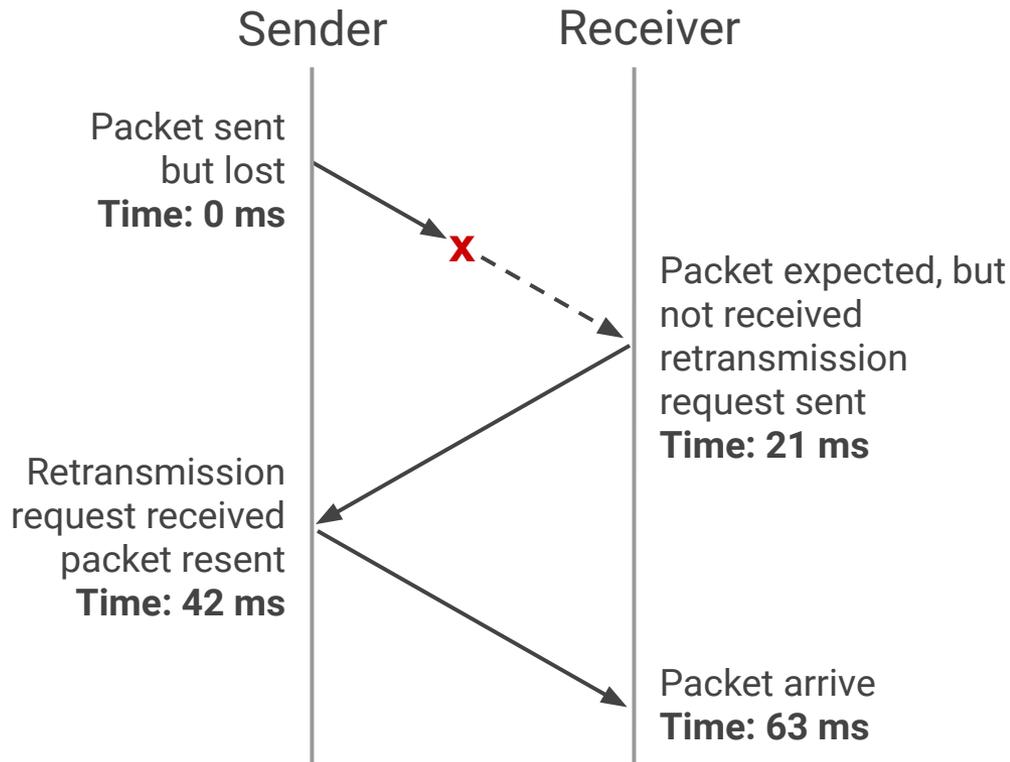- https://tools.ietf.org/html/draft-dunbar-e2e-latency-arch-view-and-gaps-02
- https://www.ietf.org/archive/id/draft-arkko-arch-low-latency-02.txt
- Others?

Neither mention coding :(

Thanks for you time!

Additional information

Coding for low latency

Latency + reliability a tough challenge!

Sender   Receiver

Packet sent
but lost
Time: 0 ms

X

Packet expected, but
not received
retransmission
request sent
Time: 21 ms

Retransmission
request received
packet resent
Time: 42 ms

Packet arrive
Time: 63 ms

Using retransmissions to obtain reliability is latency costly.

Minimum 3x the link latency

Coding for losses + latency

Source

Channel

p$_8$ | Σp | p$_7$ | p$_6$ | | | p$_3$ | p$_2$ | Σp | p$_1$
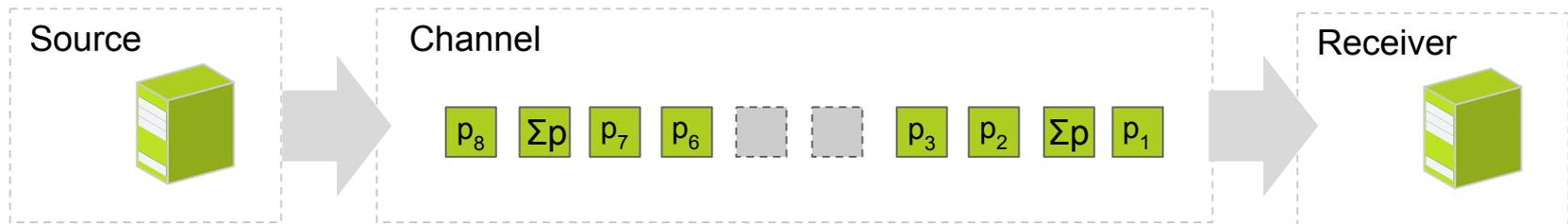
Receiver

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

Σp denotes a coded packet (linear combination of other packets)

Coding for low-latency (head of line blocking)
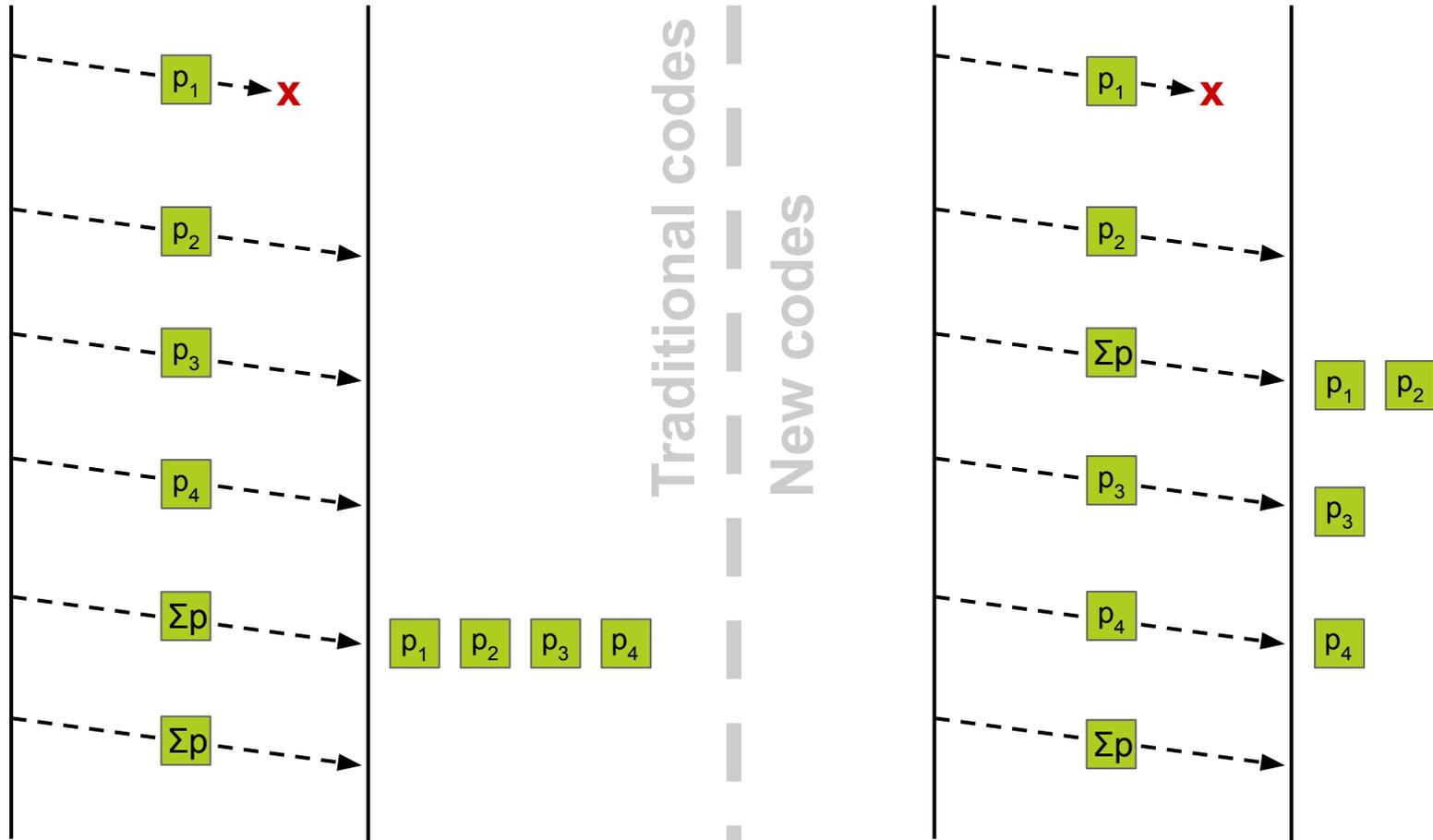
Traditional codes

New codes

p_x denote the uncoded packet *x*

Σp denotes a coded packet (linear combination of other packets)

Related blog:
http://steinwurf.com/blog/2018-06-01-delay.html