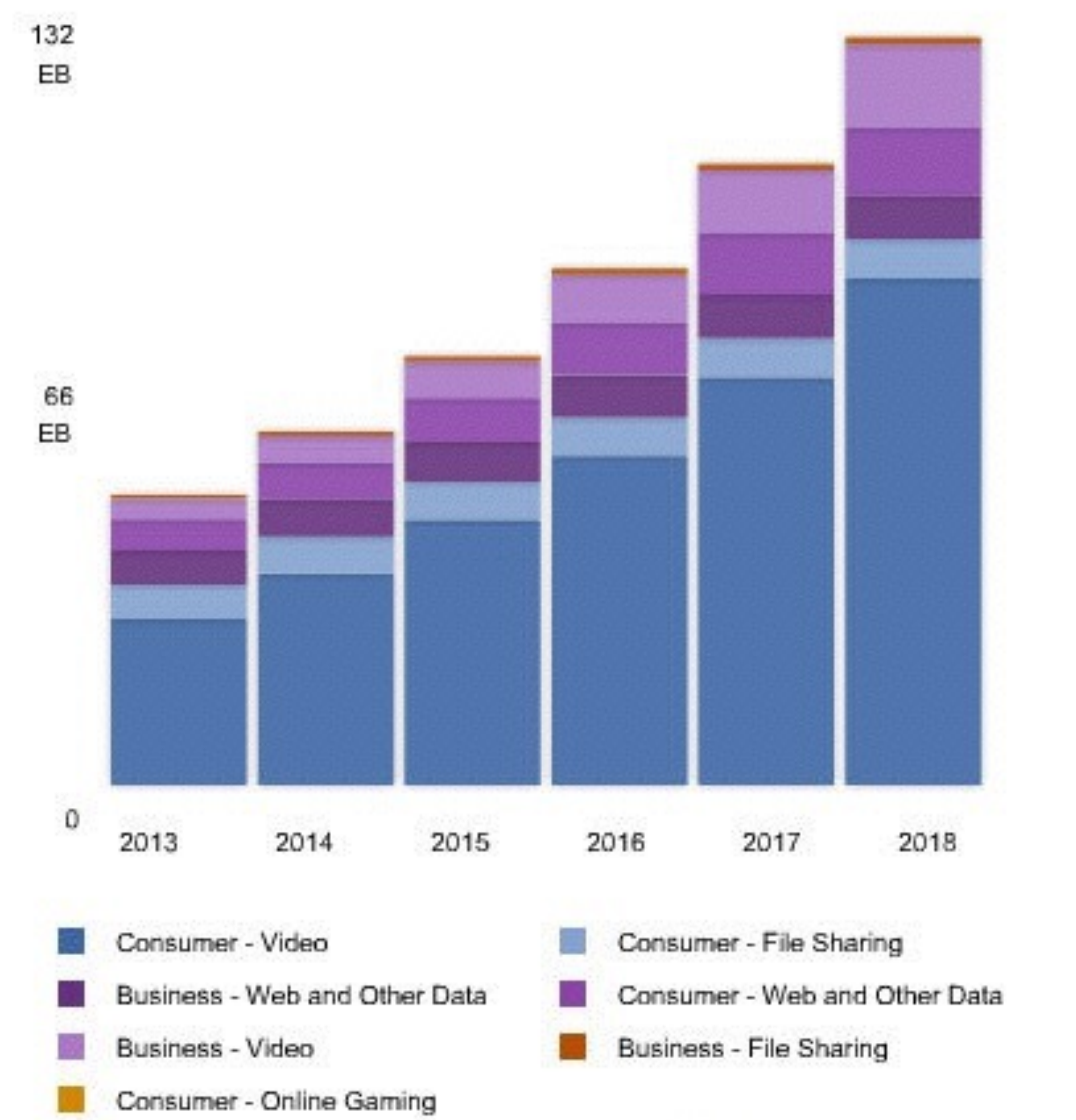
Transport-Layer Support for Interactive Multimedia Applications

Stephen McQuistin
Colin Perkins

Interactive Multimedia Applications

- Multimedia traffic comprises the majority of Internet traffic: 57% in 2013, predicted to grow to 75% by 2018*
- WebRTC standards are likely to increase *interactive* multimedia traffic share
- These applications have strict latency bounds
- Important to consider these bounds throughout the protocol stack: this work focuses on the transport layer



*Cisco VNI, June 2014

Ossification

- (PR-)SCTP and DCCP
- The transport layer has ossified around TCP and UDP: new protocols see very limited deployment
- This ossification is caused by middleboxes in the network: firewalls, NATs, caches ..
- These middleboxes often inspect IP segment payloads, dropping packets with unfamiliar transport headers

TCP vs UDP

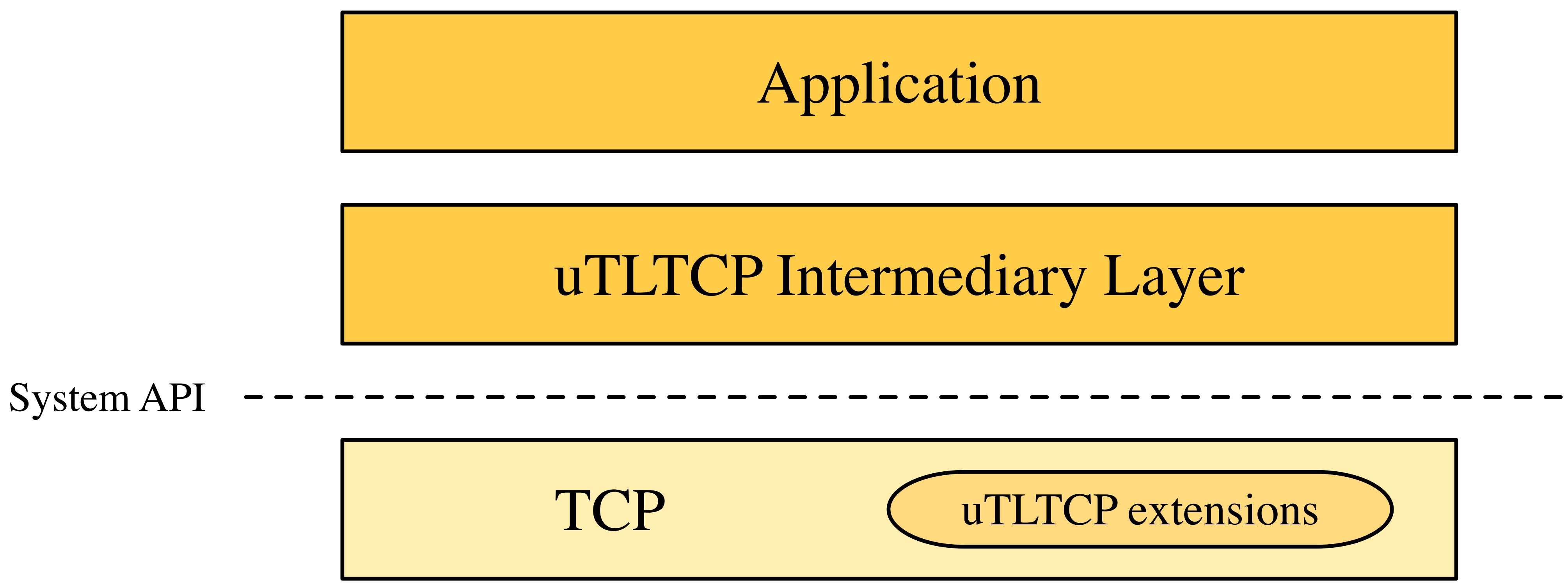
- TCP guarantees reliability and provides congestion control, but introduces delay
- UDP doesn’t introduce delay, but it also doesn’t offer reliability guarantees or congestion control
- While most applications use UDP, this is frequently blocked by enterprise firewalls
- Goal: give applications more control over latency in TCP

Unordered, Time-Lined TCP

- We propose unordered, time-lined, TCP (uTLTCP), a set of modifications to TCP
- Adds three services to TCP: unordered message delivery, time-lines, and dependencies

Architecture

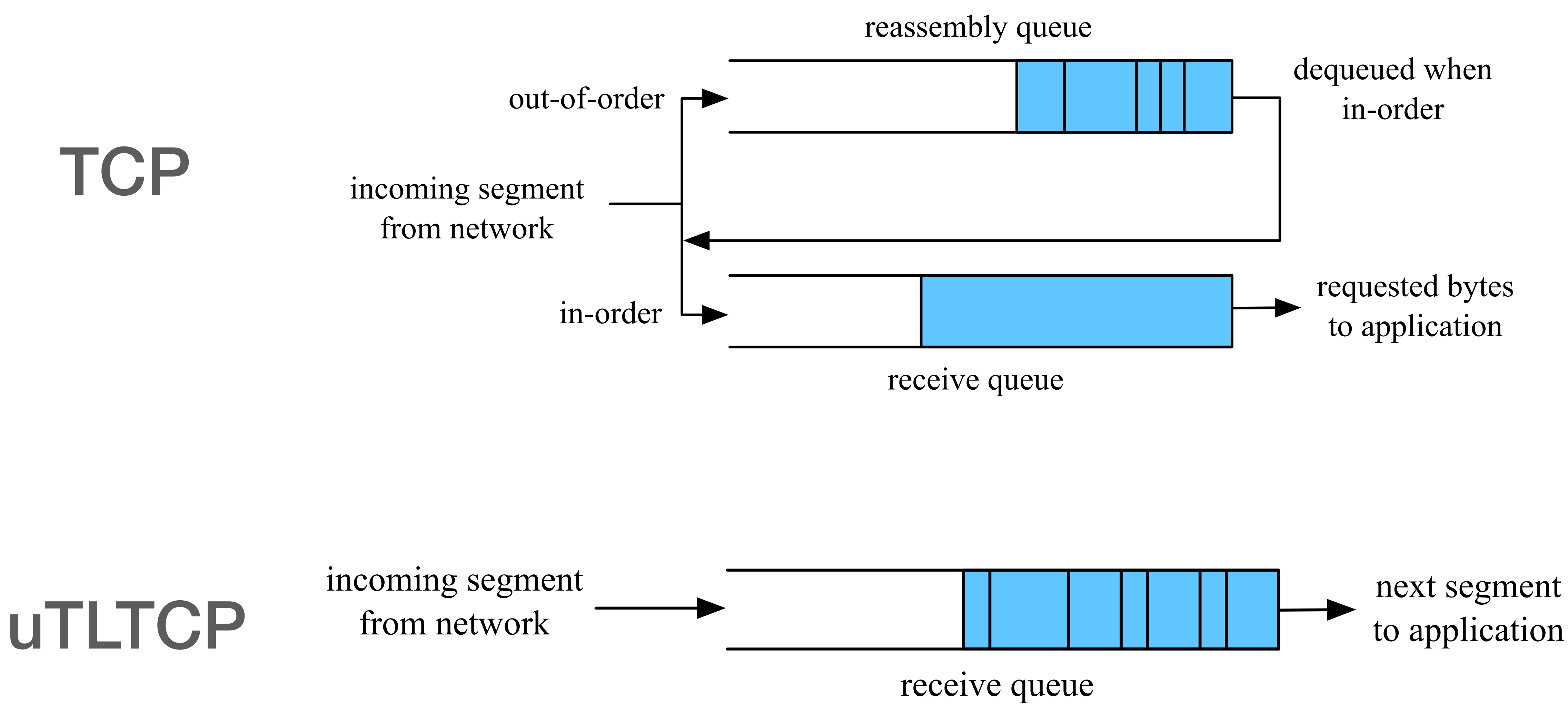
- User-level intermediary layer
- Kernel extensions
- Partial deployment of kernel extensions possible, but intermediary layer needed at both endpoints



Messages and Framing

- Other modifications require partial reliability, and messages are needed to support this
- Applications pass messages to the intermediary layer, which encodes them to be sent over TCP's byte stream
- At the receiver, the kernel extensions remove the in-order delivery buffer; segments are passed to the intermediary layer as they arrive
- The intermediary layer then decodes the segments, and passes the messages to the application
- Other modifications: Nagle algorithm is disabled, and path MTU is exposed

Messages and framing

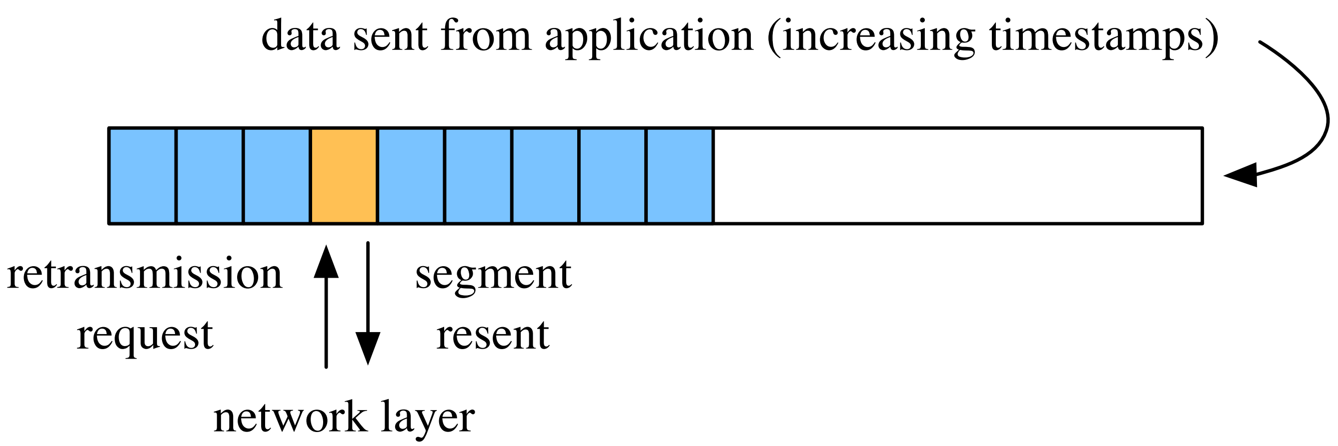


Time-lines and sub-streams

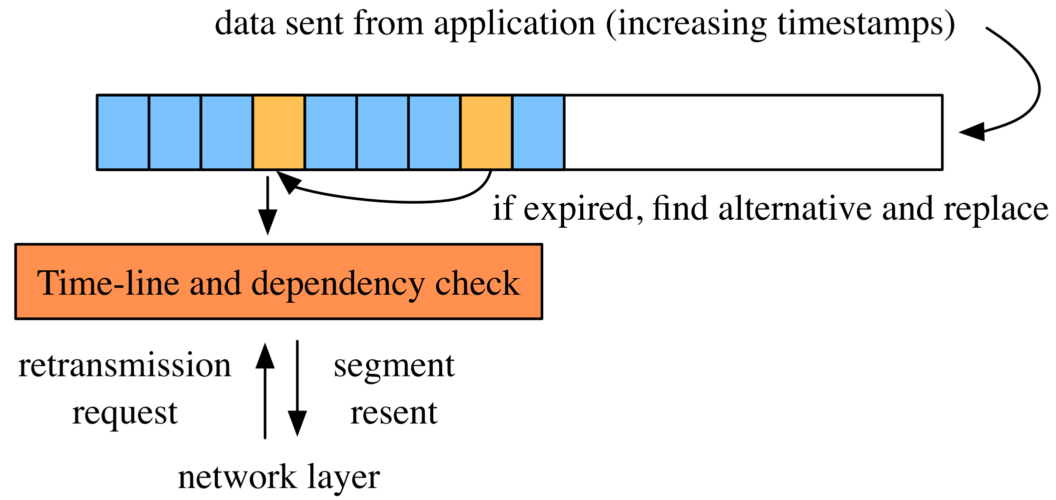
- Applications can specify a deadline by which a message must be received
- This reflects the delivery model needed by interactive applications: a VoIP message, for example, is only useful if it arrives before it needs to be played out
- uTLTCP combines the deadline with a round-trip time (RTT) estimate and play-out delay value to estimate if a message will arrive on time
- If it won't, then it tries to find a replacement in the sending buffer
- Sub-stream support allows non-time-lined data to be multiplexed on the same connection

Time-lines

TCP



uTLTCP



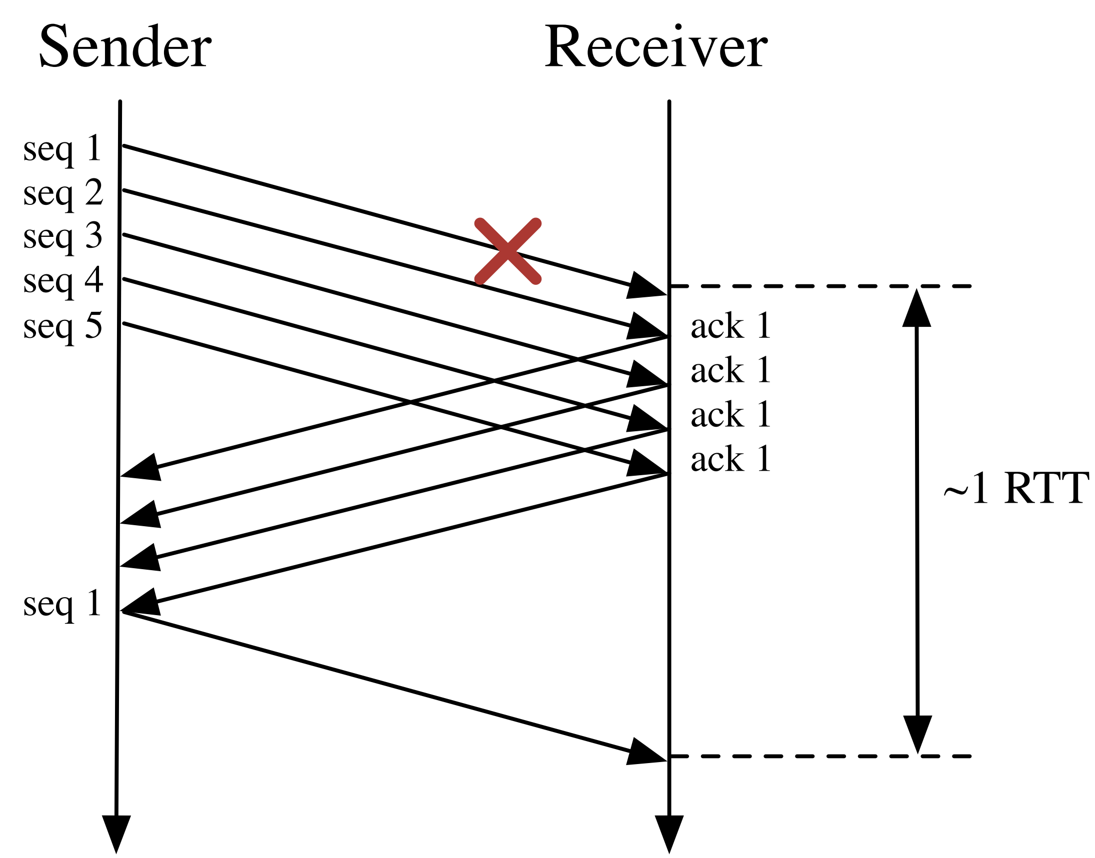
Inconsistent retransmissions:
same TCP sequence number, different payload

Dependencies

- Applications can express dependencies between two messages using sequence numbers
- If a message expires, then its dependents also expire
- A message with an expired dependency will only be sent if no other replacements can be found

Initial Analysis

- It takes approximately 1 RTT for a retransmission to arrive
- This retransmission will only be useful if the play-out buffer is greater than 1 RTT



Deployability

- Only wire-visible change to TCP: inconsistent retransmissions
- Tested by deploying Raspberry Pi devices in homes, and connecting to mobile networks
- All wired providers deliver inconsistent retransmissions successfully
- 3 of the 4 mobile providers tested deliver cached TCP segments, while 1 delivered inconsistent retransmissions

Future work

- Further analysis: when will there be a suitable replacement in the queue?
- Further deployability measurements
- Real-world evaluations to show that the protocol helps in realistic conditions