

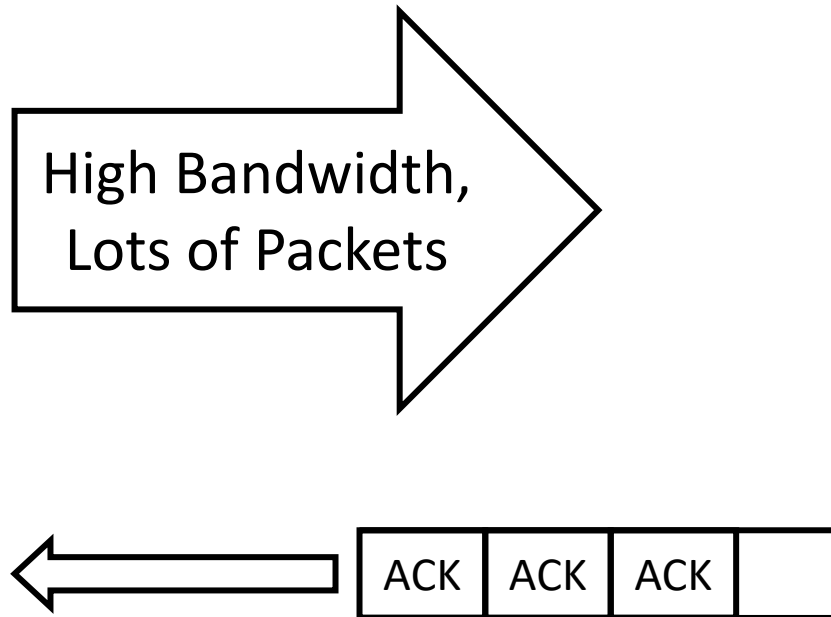
# QUIC Time Stamps, Work on it?

Christian Huitema

IETF 114, Philadelphia, Thursday, July 28, 2022

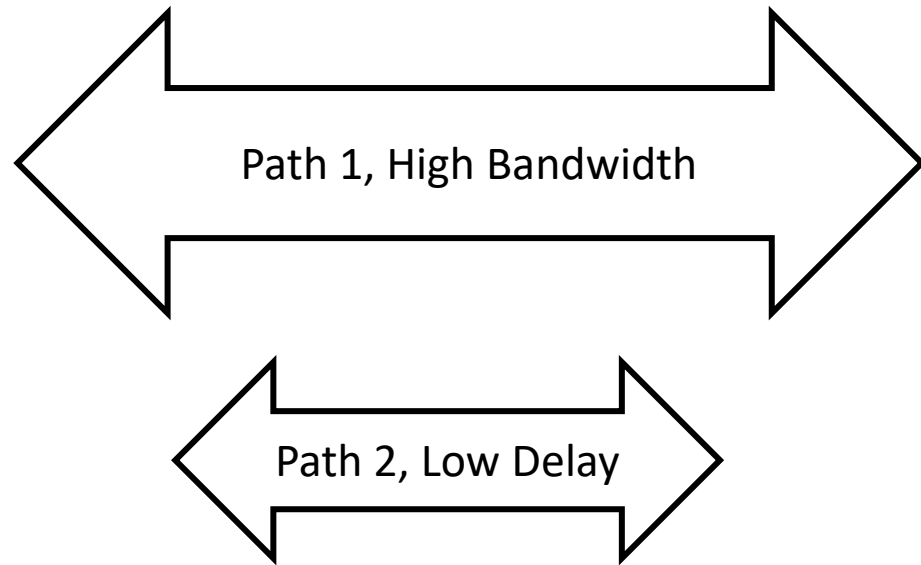
<https://datatracker.ietf.org/doc/draft-huitema-quic-ts/07/>

# Use Case 1, Congested Return Path



- Queue of ACKS on return path
  - Increased RTT
  - Mess up congestion control
  - Mess up retransmissions
- 
- Time stamps fix that
    - Measure One Way Delay
    - Distinguish between forward and return congestion

# Use Case 2, Multipath



- Sending ACK on Short Path
  - Improves responsiveness
  - Better Congestion Control
  - Better Loss Recovery
  - ...
  - But “requires a PhD” for RTT Measurement
- Time stamps fix that:
  - Measure 1 Way Delay for forward and return paths

# Draft QUIC TS

```
TIME_STAMP Frame {  
    Type (i) = TBD,  
    TimeStamp (i),  
}
```

- Time stamp in microseconds, as defined by sender
  - Typically, 0 = start of connection

- enable\_timestamp (TBD)
  1. I would like to receive TIMESTAMP frames
  2. I am able to generate TIMESTAMP frames
  3. I am able to generate TIMESTAMP frames and I would like to receive them

# What next?



- More windmills?
- Or more sunsets?

# Next Step

- Adopt the draft?
- Discuss?
  - Either simple Time Stamp (as is)
  - Or something more complex (change ACK format, on delay per packet)
  - Or something more like NTP
    - Sender Time Stamp
    - Last Received Time Stamp
    - Delay since last received
- Or ?