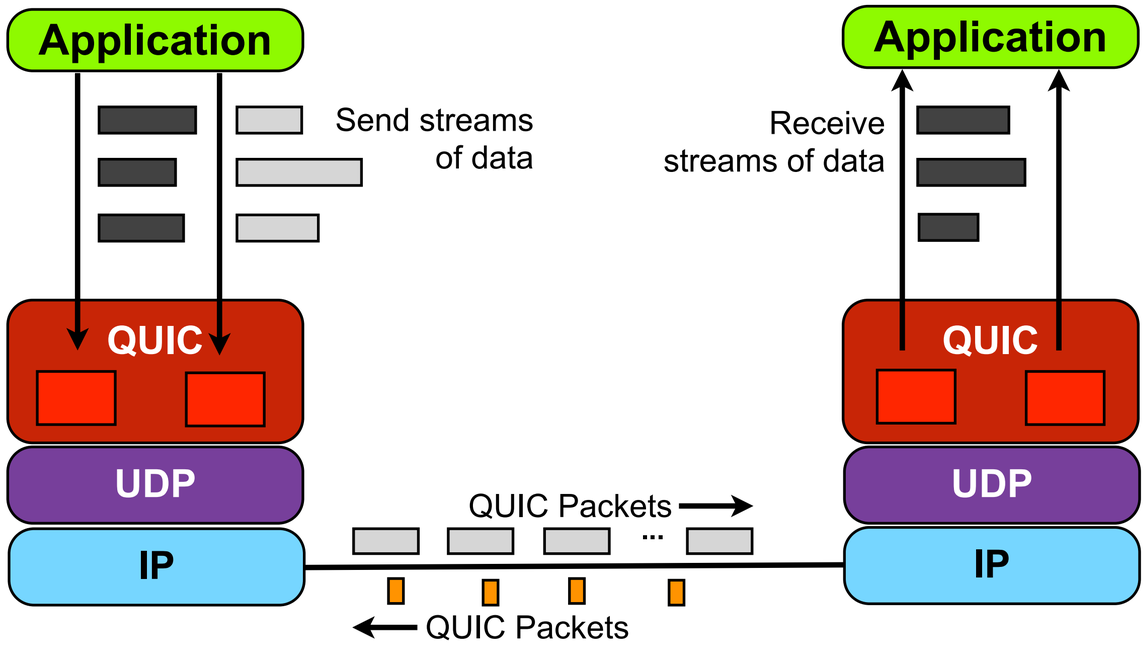
Reliable Data Transfer with QUIC

- QUIC service model
- Multi-streaming
- Limited head-of-line blocking

CC BY ND

QUIC Service Model

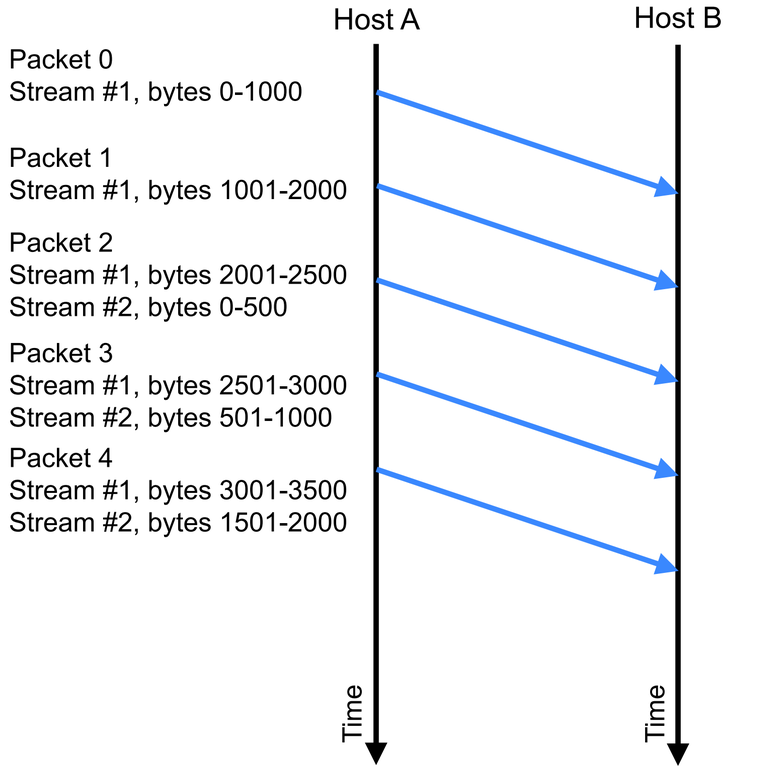


- TCP delivers reliable, ordered, byte stream
- QUIC delivers **several** ordered reliable byte streams within a single connection
  - Each stream is delivered reliably and in-order
  - Order is not preserved between streams within a QUIC connection

Colin Perkins | https://csperkins.org/ | Copyright © 2020 University of Glasgow

QUIC Packets and Sequence Numbers (1/2)

- Each QUIC packet has a **Packet Number**
  - Two packet number spaces
    - Packets sent during initial handshake
    - Packets sent to carry data
  - Within each space, packet number starts at zero, increases by one for each packet sent
    - Counts number of packets sent
    - **Different to TCP**, where the sequence number records the offset within the byte stream
- Each QUIC packet contains one or more frames
  - STREAM frames carry data
  - Each has a stream identifier, and carries the length of the data and its offset from start of stream – closer to TCP sequence number

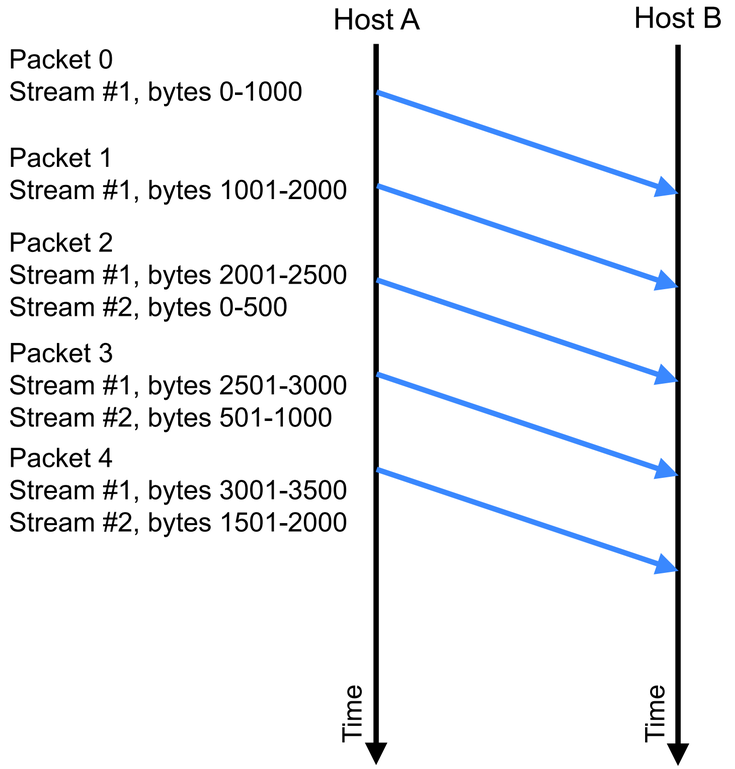


QUIC connection progress, showing packet numbers, stream data

Colin Perkins | https://csperkins.org/ | Copyright © 2020 University of Glasgow

QUIC Packets and Sequence Numbers (2/2)

- QUIC doesn't preserve message boundaries in streams
  - If data written to stream is too big for a packet, it will be split across multiple packets
  - If data written to stream is too small for a packet, it may be delayed and sent with other data to fill complete packet
  - Data from more than one stream might be sent in a single packet
  - If >1 stream has data to send, the QUIC sender can choose how to prioritise and deliver frames from each stream

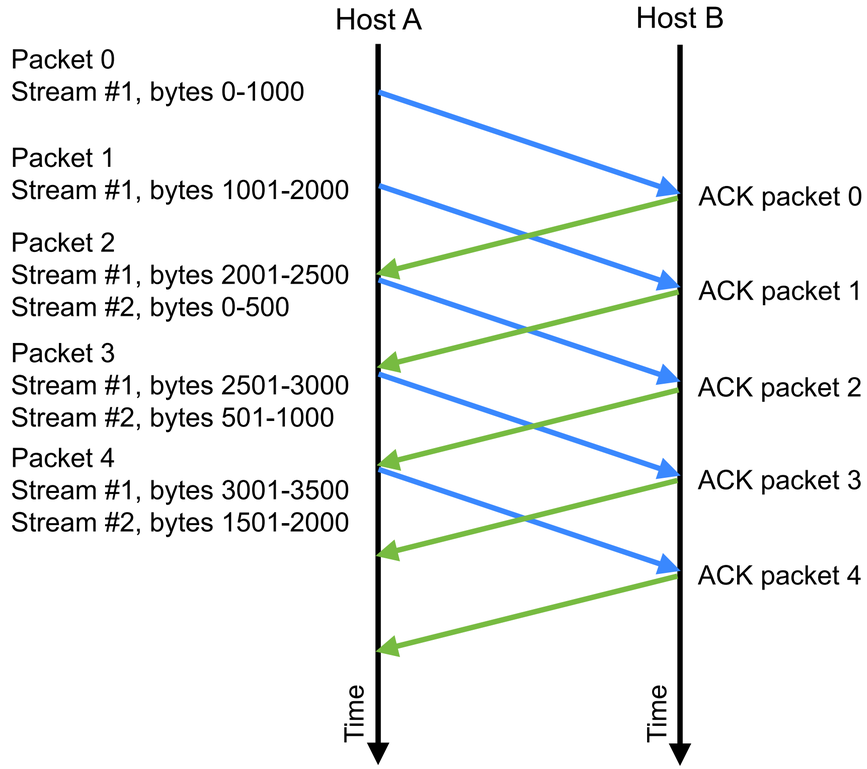


QUIC connection progress, showing packet numbers, stream data

Colin Perkins | https://csperkins.org/ | Copyright © 2020 University of Glasgow

QUIC Acknowledgements (1/2)

- A QUIC receiver sends acknowledgements for **packets** it receives
  - Unlike TCP, it **does not** acknowledge sequence numbers within each stream
  - The sender needs to remember what data from each stream was in each packet, to know what parts of each stream have been received



QUIC connection progress, showing packet numbers, stream data, and acknowledgements

Colin Perkins | https://csperkins.org/ | Copyright © 2020 University of Glasgow

QUIC Acknowledgements (2/2)

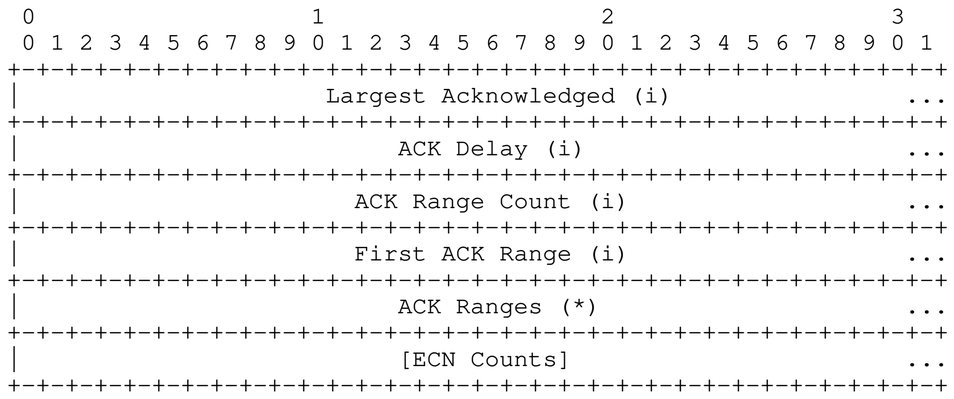
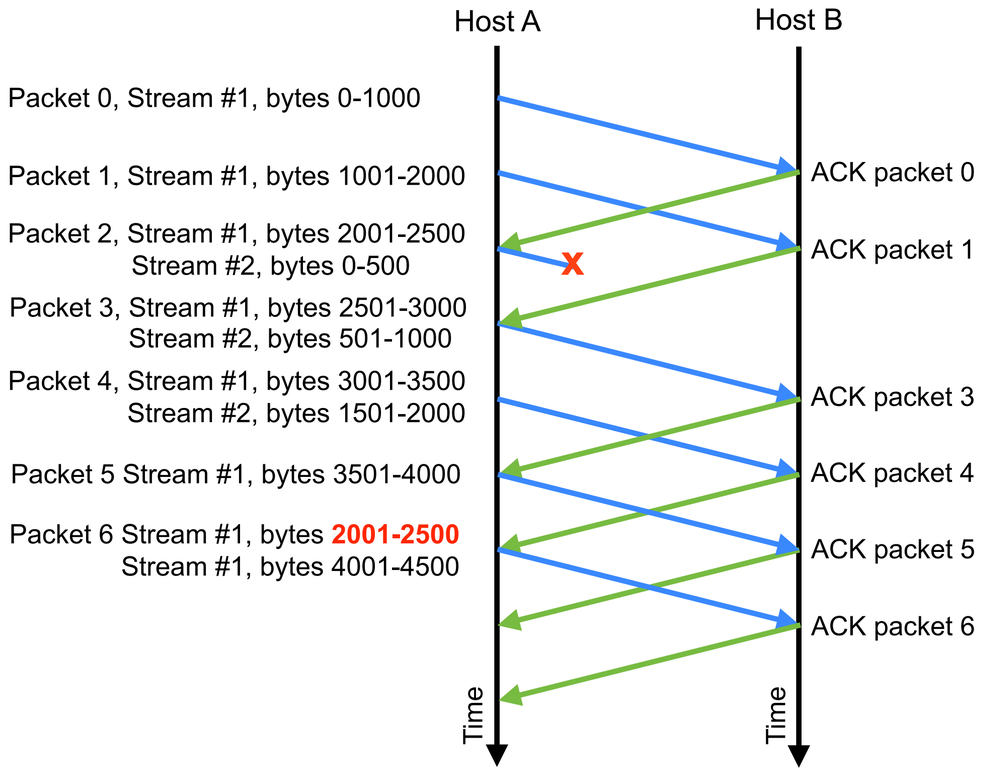


Figure 19: ACK Frame Format

- Acknowledgements sent as reverse path frames
- Acknowledgements can be delayed
  - e.g., send acknowledgement after every second packet is received
  - The **`ACK Delay`** field measures this delay to allow accurate network RTT calculation
- Acknowledgements contain ranges
  - e.g., a single ACK Frame can indicate that packets 1-4 and 6-8 were received
  - Like TCP Selective Acknowledgement extension

Colin Perkins | https://csperkins.org/ | Copyright © 2020 University of Glasgow

QUIC Retransmissions and Loss Recovery



QUIC connection progress, showing packet numbers, stream data, and acknowledgements

- QUIC does **not** retransmit lost packets
  - Every packet has a unique packet number and is only transmitted once
  - Packets declared lost when at least three later packets have been acknowledged
    - Equivalent to TCP triple duplicate ACK
  - Packets declared lost after timeout – like TCP
- QUIC retransmits lost data in new packets
  - Packetisation of STREAM frames into packets may differ when data retransmitted
  - Data ordering between streams not preserved

Colin Perkins | https://csperkins.org/ | Copyright © 2020 University of Glasgow

Head-of-Line Blocking in QUIC

- QUIC can deliver several streams within a single connection
  - Data for each stream is delivered reliability and in-order
    - Whether packet loss affects one or more streams depends how the sender chooses to distribute stream data between packets
      - Each QUIC packet can contain data from one stream, alternating between streams
      - Each QUIC packet can contain data from each stream
      - The specification places no requirements on how streams are split across packets
    - Depending on loss patterns, each stream can independently suffer head-of-line blocking
      - Affects performance **of that stream** in the same way head-of-line blocking affects TCP
  - Order is not preserved between streams within a QUIC connection
    - One stream might be blocked waiting for a retransmission, while other streams continue to deliver data without loss
    - Data from different streams is sent and received independently → careful use of streams can limit duration of head-of-line blocking

Colin Perkins | https://csperkins.org/ | Copyright © 2020 University of Glasgow

Reliable Data Transfer with QUIC

- QUIC delivers **several** ordered reliable byte streams within a single connection
  - Can be treated like several parallel TCP connections → instead of opening several TCP connections to a server, open one QUIC connection and send multiple streams within it
  - Can be treated as a framing device → each stream represents a single object, with end-of-stream signalling the object is complete
  - Order is not preserved between streams within a QUIC connection

- Best practices for use of multiple streams are still evolving

Colin Perkins | https://csperkins.org/ | Copyright © 2020 University of Glasgow

Summary

- Internet → best effort packet delivery
- Data transfer in Internet transports
  - UDP – unreliable but timely; substrate
  - TCP – reliable, ordered, stream
  - QUIC – many reliable, ordered, streams
- Different services for different needs

Colin Perkins | https://csperkins.org/ | Copyright © 2020 University of Glasgow