CPSC 317 COMPUTER NETWORKING

2023W2: Transport – Day 9 – Alternative Transport Protocols

READING

- Reading: Search Google: QUIC SIGCOMM

LEARNING GOALS

Alternate transport protocol designs

- Explain why building a transport protocol on top of UDP is valuable
- Explain the limitations of TCP
- Explain the limitations of HTTP/1.1
- Explain how HTTP/2 (partially) addresses the limitations
- Explain why co-design of application and transport protocols is valuable
- Explain key ideas of the QUIC protocol

REMINDER: TRANSPORT PROTOCOLS

TCP	UDP
Connection	No connection
Byte stream-based	Packets-based
Reliable ordered delivery	Best effort
Flow/Congestion control	Nope
Possible delays	No (transport level) delay

HOW DO WE ACHIEVE RELIABILITY?

- Checksums
- SEQ and ACK numbers (ordering, reliable delivery)
- Timeouts and retransmissions (loss, delays, congestion)
- Windows (flow control)

HYBRID TRANSPORT PROTOCOLS

- DCCP (Datagram Congestion Control Protocol)
  - Does not provide reliable in-order delivery for data
  - Provides application-selected congestion control
  - Useful for multimedia streaming, online gaming

- SCTP (Stream Control Transmission Protocol)
  - Message-based like UDP
  - Multiplex several application “streams”
  - Independent reliable in-order delivery and congestion control for each stream (like TCP)
  - Used for VoIP

MAJORITY OF TODAY'S INTERNET TRAFFIC

- Majority of traffic on the web: HTTP + TLS + TCP

- Lot of traffic suffering from inefficiencies in protocol stack
  - in TCP
  - in HTTP
  - in interaction between HTTP and TCP

PROBLEMS WITH TCP

- 3-way handshake too expensive
  - Worse with added TLS handshake for HTTPS
  - More on TLS later in Module 8

- RTT estimation too inefficient
  - RTT: time between transmitting a segment and receiving an ACK
  - Sequence numbers conflate ordering and reliable delivery
    - Retransmitted segments have the same sequence number as the original
    - Confounds RTT estimates

REVIEW: HTTP/1.1 OVER TCP

Persistent connections and pipelining to reduce connection overheads

Limitation of HTTP/1.1

- Overheads: ASCII headers
  - "Content-Length:" = 15 bytes = 120 bits!

Limitation of HTTP/1.1 over TCP

- Suffers from head-of-the-line blocking
  - A small request blocked behind a large request
  - Loss of any packet prevents all transfers

CLICKER QUESTION

You are in the bank, and there is only one teller. There are 10 customers in the bank, and the one at the front of the line has just started a complicated 10 minute transaction. The other 9 customers (including you) have simple 1 minute transactions.

How long on average will the 10 of you take to get your banking done? Choose the closest answer.

A. 1 minute
B. 5 minutes
C. 15 minutes
D. 20 minutes

CLICKER QUESTION

You are in the bank, and there is only one teller. There are 10 customers in the bank, and the one at the front of the line has just started a complicated 10 minute transaction. The other 9 customers (including you) have simple 1 minute transactions. Suppose the 1st customer lets the other 9 of you go first.

How long on average will the 10 of you take to get your banking done? Choose the closest answer.

A. 1 minute
B. 5 minutes
C. 15 minutes
D. 20 minutes

IMPROVEMENTS IN APPLICATION PROTOCOL

HTTP/2 (no longer called http/2.0)

- Headers are encoded in binary and compressed
  - compressed binary format called QPACK
  - Limitation: not human readable
- Server allowed to push objects without being asked
  - Via “push promises” – reduce latency for objects client likely to ask
  - Client can say “No thanks, I got it already”
- Responses may be reordered w.r.t. requests
  - Limitation: can still prevent transfers if TCP lost packets

REQUIRED IMPROVEMENTS

- Multiplex requests independently (avoid HOL blocking)
- Improve connection handshake protocol
- Reduce congestion control inefficiencies

- Where do we implement changes?

LAYERED ARCHITECTURE CREATES CONSTRAINTS

- May need to change kernel on all devices
- Upper layers cannot control lower layers
  - Some hacks exist: e.g., disabling Nagle's algorithm in TCP

- **Idea:** co-design application and transport protocols
  - Improve performance by combining some of the protocols

ALTERNATIVE: QUIC

- QUIC: Quick UDP Internet Connections
- Development started at Google in 2012
  - Version 1: RFC 9000
  - Version 2: started in Apr 2021; RFC 9369 established in May 2023
- 5-10% of Internet traffic is QUIC as of 2021
- HTTP/2 over QUIC: aka HTTP/3

QUIC KEY FEATURES

- Add reliability on top of UDP
  - Implemented in user space rather than in the kernel

- Could be an alternative to UDP and TCP above IP layer itself
  - Problem: existing network middleboxes cannot handle new protocols

QUIC KEY FEATURES

- Integrates TLS into the transport
- Complete both TCP and TLS handshake in one RTT
  - More on TLS later

TCP VS TLS HANDSHAKES

TCP

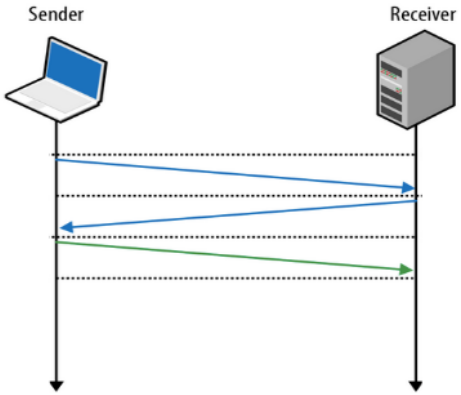
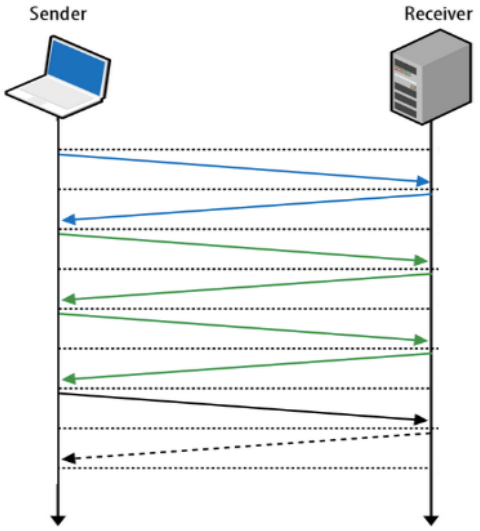


Figure 2-1. Three-way handshake

TCP+TLS



e 4-2. TLS handshake protocol

QUIC HANDSHAKE

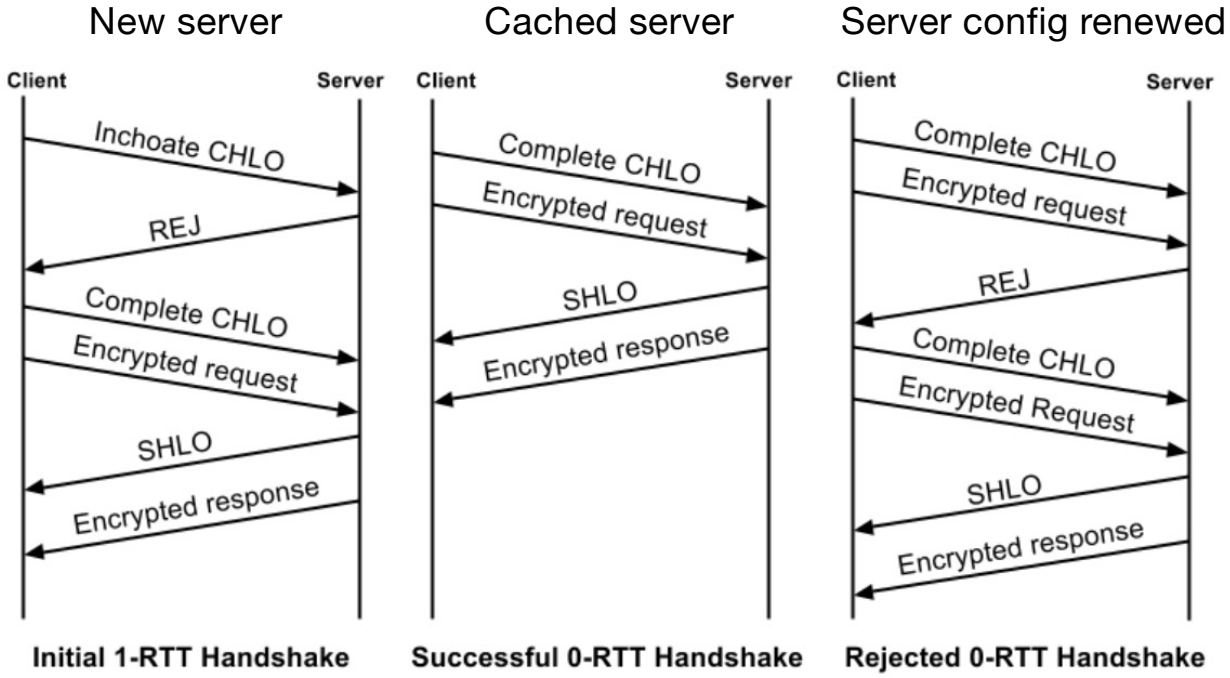


Figure 4: Timeline of QUIC's initial 1-RTT handshake, a subsequent successful 0-RTT handshake, and a failed 0-RTT handshake.

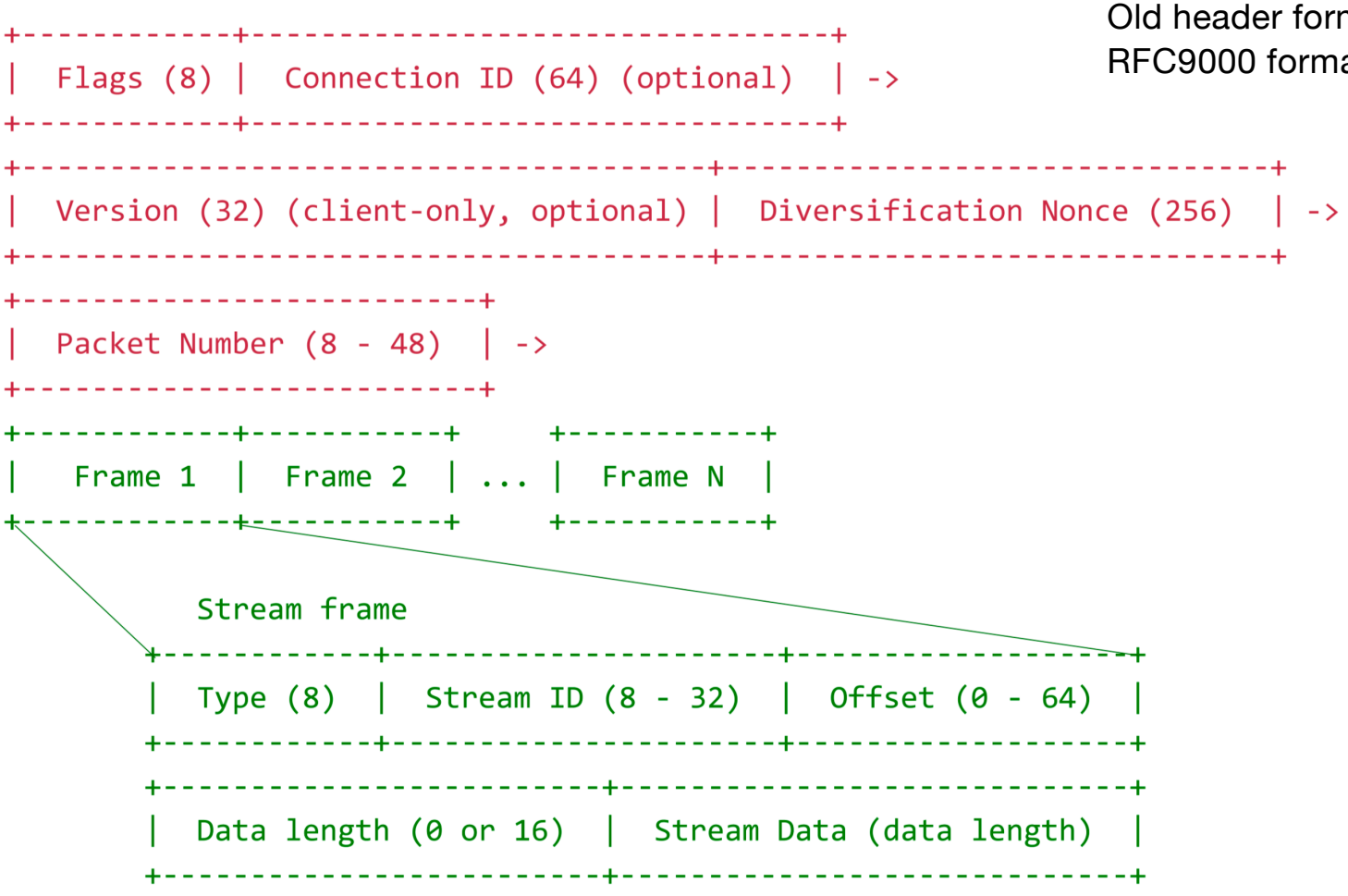
QUIC KEY FEATURES

- Multiple streams within a single connection
- Handle losses, flow control in each stream independently
- Avoid HOL blocking
- Priorities can be set per stream
- Streams are very light-weight
  - Using a new stream identifier creates the stream
  - HTTP/3 uses a different stream for each object

QUIC KEY FEATURES

- Separate sequence numbers for data and frame delivery
- Decouple reliability and ordered delivery
- Better RTT estimation

Old header format
RFC9000 format different



PACKET NUMBERS

- Used for RTT estimation (and congestion detection)
- Every packet gets a new one (including retransmitted ones)
- Starts at 0
- Why is it OK to start at 0 all the time?

Stream Offsets

- Used for reliable delivery
- Like the TCP sequence number
- Starts at 0

QUIC KEY FEATURES

- Mobility
  - Can migrate QUIC connections from one device to another
  - The client-selected connection ID is the key, not the client's host address
  - e.g., from Wi-Fi to cellular

SUMMARY OF TRANSPORT LAYER

- TCP and UDP are the dominant protocols
- Innovation is still possible
  - Lot of innovation driven by performance goals, some by security
- Trends in transport protocols research
  - Changes in TCP congestion management
  - New transport protocols: SCTP, DCCP
  - New transport functionality at the application layer: QUIC