What’s wrong with HTTP?

- TCP connection “warmup”
  - ... persistent connections

- Slow down due to TCP error control actions
  - ... parallel connections

- Head-of-line blocking (big requests ahead of small ones)
  - ... SPDY and HTTP/2

- Cost of initiating (secure) transport
  - ... QUIC
An open protocol developed by Google
- precursor to HTTP/2, deprecated as HTTP/2 was ratified

Goal: reduce latency and improve security
- through multiplexing, prioritization, and compression
- server push

Does not replace HTTP/1.1
- implemented using TLS’s Next Protocol Negotiations (NPN)
HTTP/2

- Based on SPDY

- Key features (mostly the same as for SPDY)
  - multiplexing
  - compression
  - server push

- RFC 7450 (May 2015)
  - supported by major browsers
  - fast-tracked standardization with heavy Google involvement
TCP is bad and there isn’t much we can do about it…

- so, let’s throw it away and build a new transport/application layer protocol (in the application layer) on top of UDP
- and, improve on TCP / TLS / HTTP/2 combination

UDP-Based Secure and Reliable Transport for HTTP/2

Goals:
- reduce connection establishment time (esp. of repeated ones)
- multiplexing
- improve congestion control
- FEC
- connection migration
Router Architecture (review)

Input buffers

Routing table

Route processor

Output buffers

Switching fabric

Packet path

Control

Header info (destination address)
Switching Fabric

- N inputs and M outputs (often M = N)

- **Solutions**
  - bus
  - crossbar
  - multistage interconnection network (MIN)

- **Cost / benefit analysis**
  - number of crosspoints (switches)
  - latency
  - permissible requests / cross-section bandwidth
Elementary Switch

- 2 x 2 switch

- Multiplexing equivalency
  - space
  - time
  - wavelength
Costs...

- For simplicity, assuming $M = N$
- **Bus**
  - $O(N)$
- **Crossbar**
  - $O(N^2)$
- **Multistage Interconnection Networks**
  - Typically $O(N \log(N))$ or $O(N \log^2(N))$
Better than $O(N^2)$...

- What is lost?
  - output and internal conflicts

- **Clos** network
  - Beneš network

- **Delta** networks (self-routing networks)
  - Banyan, Baseline, Omega, Flip

- **Batcher** sorting network