UNIVERSITY OF
Southampton

UNIVERSITY OF Southampton

Optimising HTTP

COMP3220 Web Infrastructure

Dr Nicholas Gibbins – nmg@ecs.soton.ac.uk

Physical Limits

Upper limit on speed of communication is the speed of light: c = 300,000,000 m/s

A message sent halfway round the world on a great circle route will take a minimum of 0.067 s to reach its destination

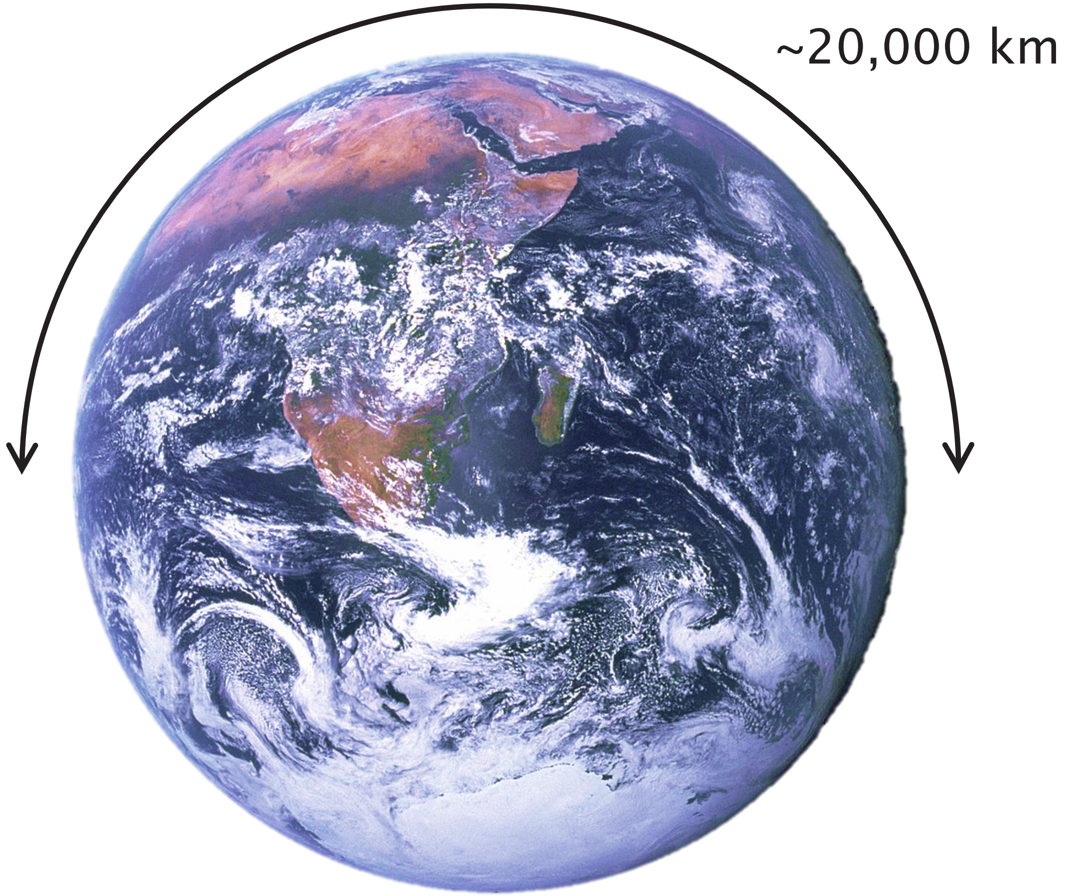


Image: NASA

Network Latency

The time between a source sending a packet and the destination receiving it

Actual speed of communication will be less than the speed of light:

- Optical fibre is typically ~70% of c
- Coaxial cable may be >80% of c

Routers, switches, etc introduce delays

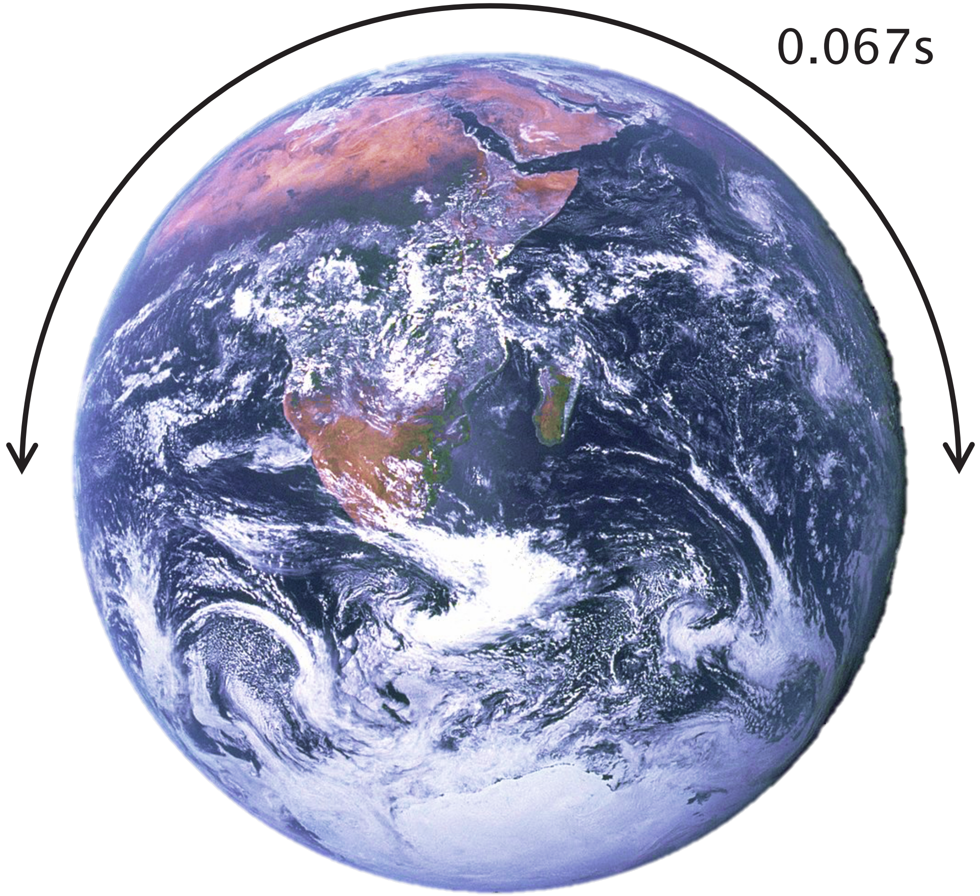


Image: NASA

The Internet Protocol Suite

Application Layer process-to-process	**HTTP** SMTP FTP DNS SSH IMAP POP TLS …
Transport Layer host-to-host	Transmission Control Protocol (TCP) User Datagram Protocol (UDP)
Internet Layer addressing and routing	Internet Protocol (IP)
Link Layer physical transmission	Ethernet 802.11 (WiFi)

HTTP over TCP

HTTP runs on top of TCP (Transmission Control Protocol)

- TCP provides a reliable connection
- Guarantees that packets will eventually reach their destination
- Lost packets are resent

Opening an HTTP connection requires that a TCP connection be opened first

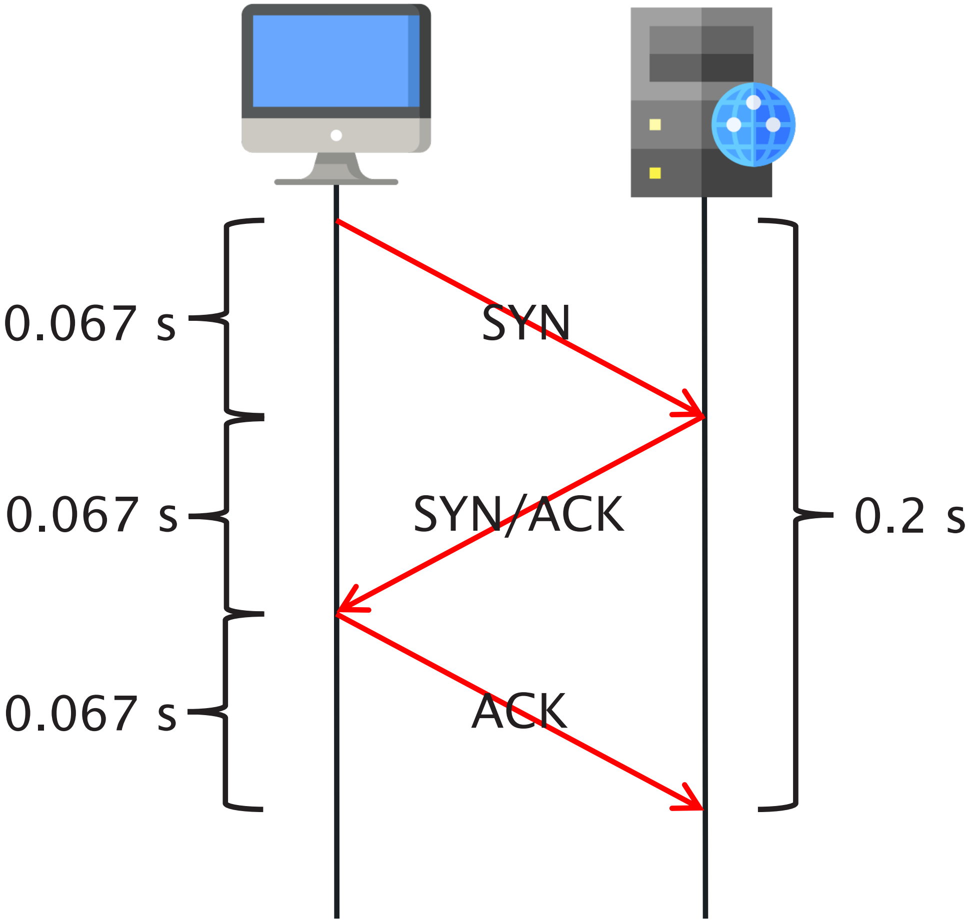
TCP Set-up and Teardown

TCP establishes reliable connection using a three-way handshake

- SYN, SYN/ACK, ACK

Lower limit for the duration of this exchange to a host at the antipodes: 0.2 s

- for just the TCP connection (no HTTP)
- at the speed of light
- without additional delays
- before any additional overhead from the TLS handshake



HTTP/1.0 and earlier

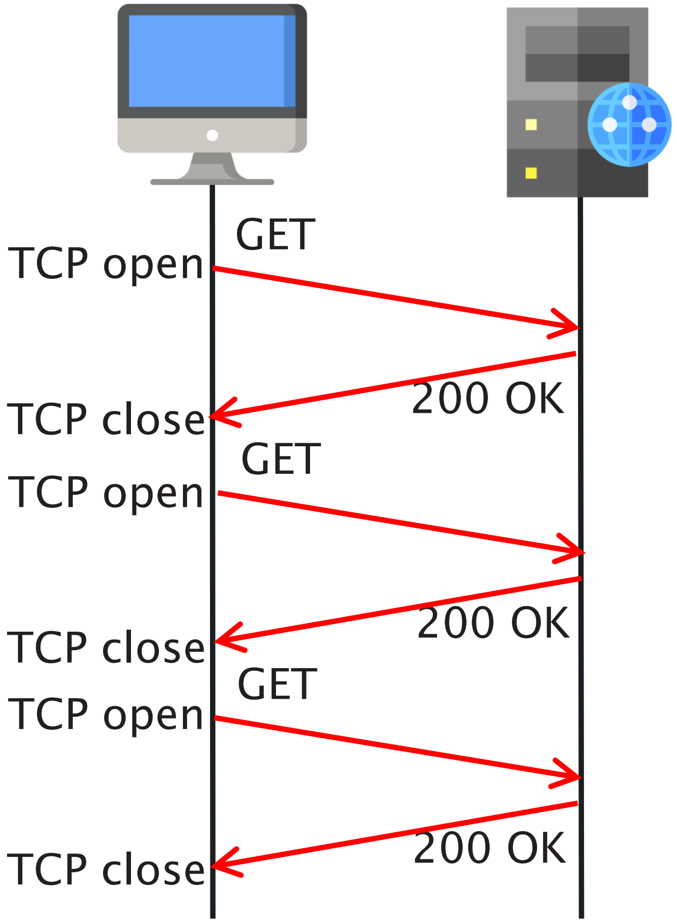
Before HTTP/1.1, each HTTP request used a separate TCP connection

Expensive and time-consuming!

- 3-way handshake for each TCP open
- 4-way handshake for each TCP close
- Latency issues

Some clients open multiple connections at same time

- Increased server load to handle many simultaneous connections



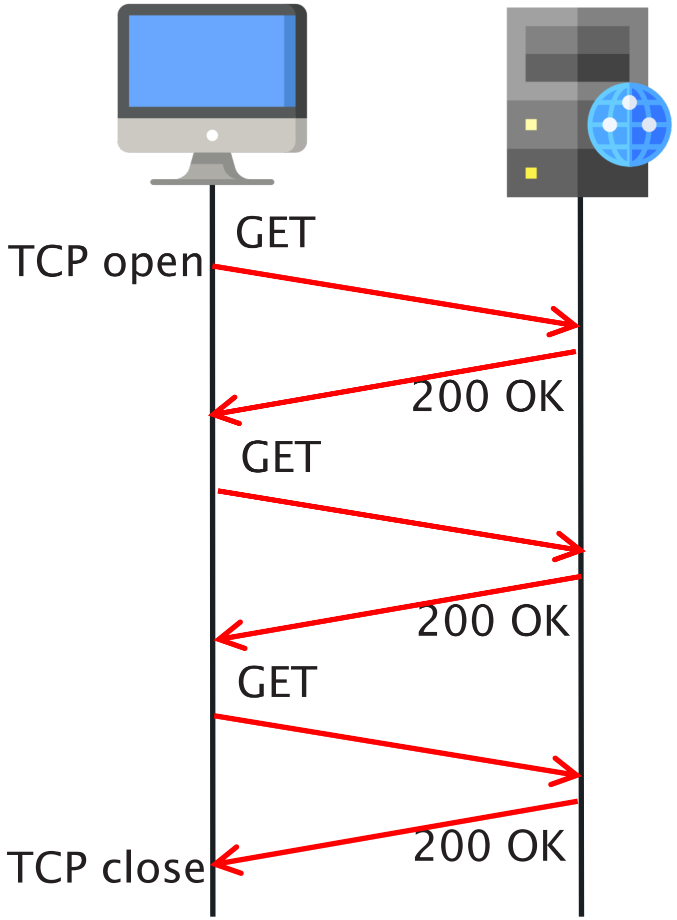
HTTP keep-alive

HTTP/1.1 introduced keep-alive

TCP connections reused for multiple HTTP requests

- one TCP open and TCP close handshake

Still latency issues, but now in the application layer rather than the transport layer

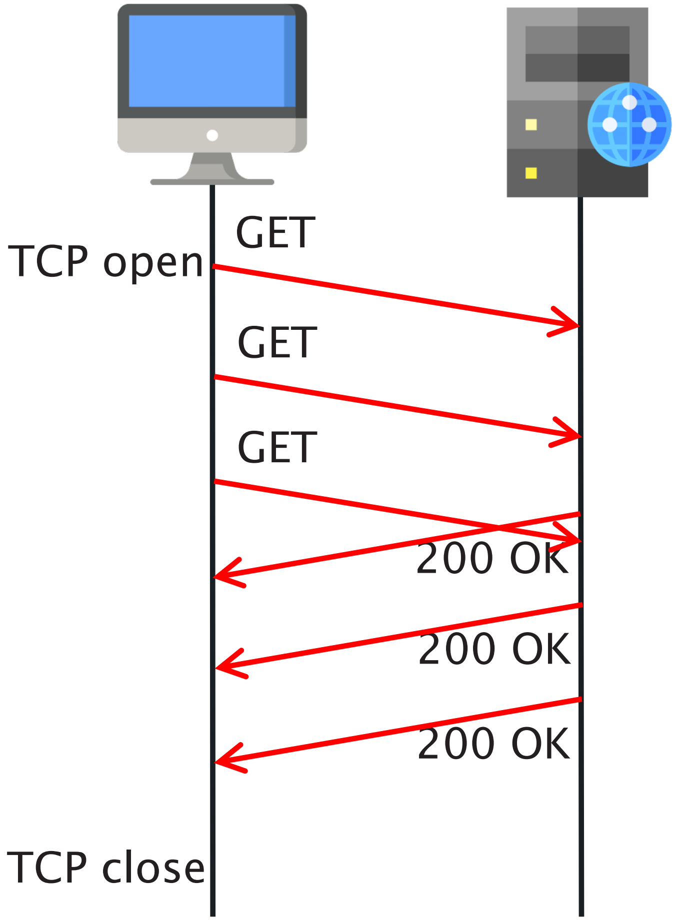


HTTP pipelining

Also available from HTTP/1.1

Pipelining allows multiple requests to be made without waiting for responses

- Server must send responses in same order as received requests
- Reduces application layer latency



Network bandwidth

Latency isn't the only consideration

- There are also limits to the data transfer capacity of the network
- How long will it take to transmit a message?

Content encoding (covered in content negotiation lecture) can be used to compress message bodies

- Smaller message body – less time taken to transmit

HTTP/2

First major HTTP revision since 1997

- Based in part on earlier SPDY proposal from Google
- Preserves existing HTTP semantics (methods, response codes, headers, etc)
- RFC7540 published 14 May 2015

Offers four improvements over HTTP/1.1:

- Multiplexed requests
- Prioritised requests
- Compressed headers
- Server push

Belshe, M. et al (2015) *Hypertext Transfer Protocol Version 2 (HTTP/2)*. RFC7540. Available at: https://tools.ietf.org/html/rfc7540

HTTP/2 prioritised requests

A connection may contain multiple streams - multiplexing

- Each stream consists of a sequence of frames
- Each stream has an associated priority

Frames from higher priority streams sent before those from lower priority streams

- Allows asynchronous stream processing (unlike HTTP/1.1 Pipelining)

HTTP/2 compressed headers

HTTP/1.1 can compress message bodies (using LZW/LZ77/deflate/Brotli)

- `Content-Encoding:` header (see content negotiation lecture)
- Sends headers in plain text

HTTP/2 also provides the ability to compress message headers

HTTP/2 server push

HTTP/1.1 servers only send messages in response to requests

HTTP/2 enables a server to pre-emptively send (or *push*) multiple associated resources to a client in response to a single request.

- Send images, stylesheets, etc before they're requested by the client

HTTP/3

Named as future planned development of HTTP in 2018

- Replaces TCP with QUIC over User Datagram Protocol (UDP)
- Addresses shortcoming of HTTP/2; multiplexes are not visible to TCP's loss recovery mechanism, so a lost or reordered packet stalls all active requests

UDP is an unreliable protocol

- No guarantee of delivery or ordering
- Lightweight, however

QUIC

- Eliminates some issues with TCP (blocking if packets are delayed or lost)
- Supports multiplexed connections over UDP
- Designed to reduce TLS overhead (see next lecture)

Bishop, M. (2020) *Hypertext Transport Protocol Version 3 (HTTP/3).* Internet Draft.
Available at: https://quicwg.org/base-drafts/draft-ietf-quic-http.html

Further Reading

Fielding, R. and Reschke, J. (2014) *Hypertext Transfer Protocol (HTTP/1.1): Message Syntax and Routing*. RFC7230. (see §6)

https://tools.ietf.org/html/rfc7230

Belshe, M. et al (2015) *Hypertext Transfer Protocol Version 2 (HTTP/2).* RFC7540.

https://tools.ietf.org/html/rfc7540

Bishop, M. (2020) *Hypertext Transport Protocol Version 3 (HTTP/3).* Internet Draft.

https://quicwg.org/base-drafts/draft-ietf-quic-http.html

Next Lecture: Securing HTTP