

#2: Application Layer: HTTP, DNS

Application Architecture

1) Client-server

2) peer-to-peer

3) Hybrid

server

client

- wait for request

- initiates contact with server

- request service from other peers, provide service in return
- distributed control
- v scalable
- harder to manage
- e.g. bitcoin

- e.g. instant messaging
- but location detection is centralised
- ^A online: contacts server to find IP of B
- establishes P2P with B and starts messaging

Transport Service Consideration For Applications:

- Data Integrity
- Throughput
- Timing
- Security

affects

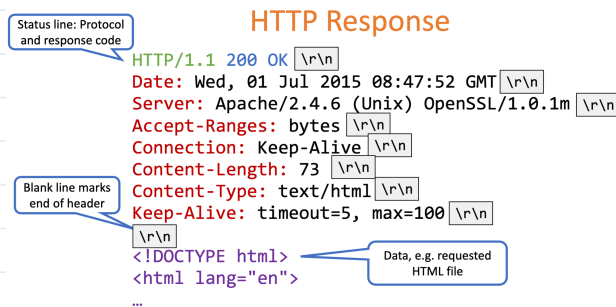
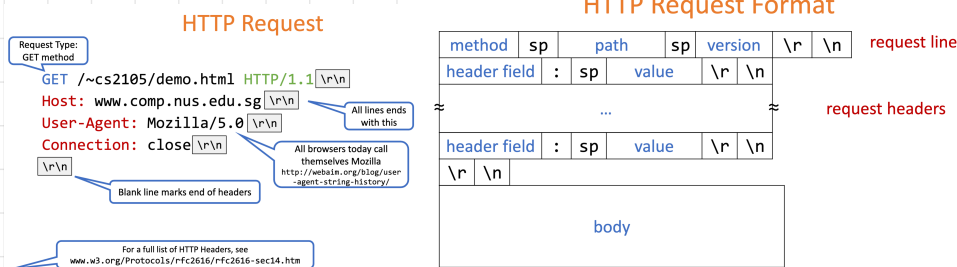
What architecture?

What type of transport service? (TCP/UDP)

how to exchange messages? (protocol?)

HyperText Transfer Protocol (HTTP)

- client/server model



- Uses TCP as a transport service

Round Trip Time (RTT): Time taken for a packet to travel from client to server and go back

* Typical HTTP connection

- 1) Browser establishes TCP connection (1 RTT)
- 2) Load initial webpage (1 RTT)
- 3) Other loaded objects (1 RTT)

~ 1996

HTTP/1.0

- Non-persistent

~ 2015

HTTP/2

- Multiplexing!

~ 1999 HTTP/1.1

- persistent

↳ don't need to reestablish TCP connection for every object

~ 2022

HTTP/3

- uses UDP instead of TCP

pipelining vs sequential

- pipelining allows you to send multiple requests before you receive a response for the previous request you sent

Multiplexing

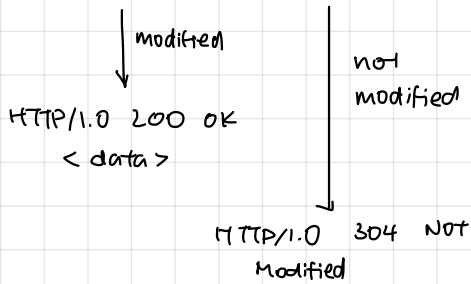
- responses can come back in any order (diff from pipelining!)

HTTP Status Code

- 200: OK
- 301: Moved permanently
- 304: Not Modified
- 403: Forbidden
- 404: Not Found
- 500: Internet Server Error

conditional GET

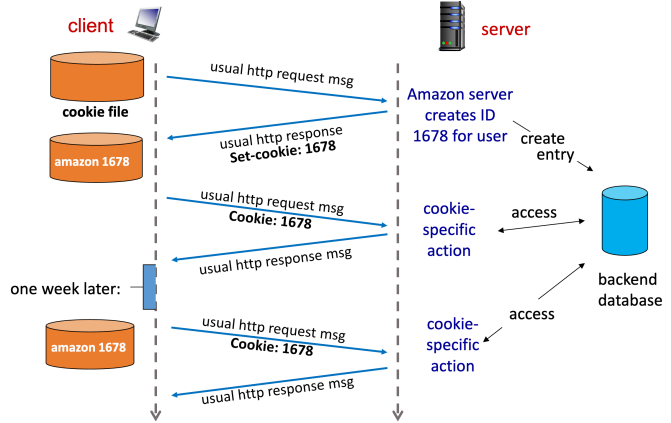
- send HTTP to server with header
If-Modified-Since: <date>



cookies

- HTTP is designed to be stateless
↳ server has no information about past clients
- cookies help maintain state
↳ cookie is saved in your browser,
↳ whenever you send a request, your browser automatically appends the cookie into the header of the http request

Keeping User State with Cookie



Domain Name System

- translates between Hostname to IP address
 ↓
 e.g. www.comp.nus.edu.sg ↓ 137.132.80.57

DNS Resource Record

↳ maps between host names and IP addresses (as well as others)
 ↳ stored as resource record

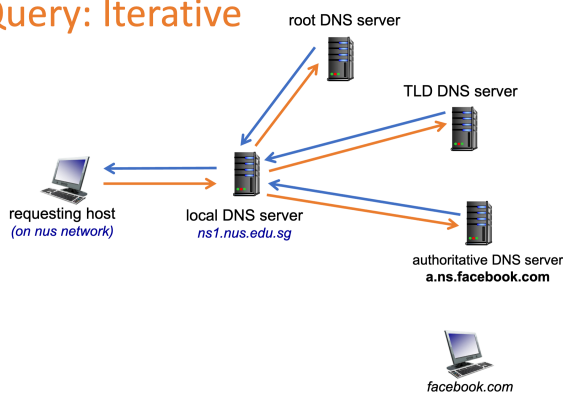
- A record: Hostname → Value
- NS record: Domain → Hostname of authoritative NS for domain
- CNAME: Alias → real hostname
- MX: Domain of email address → Name of mail server managing the domain

* Lots of DNS Records

↳ ∴ DNS stores RR in a distributed, hierarchical Database

* DNS query is an iterative process

DNS Query: Iterative



- Caching at browser → OS → router → ISP
- DNS runs over UDP/53

Cybersec

- DNS Hijacking: compromise NS
- DNS Tunneling: encode data in DNS messages
- DNS Poisoning: intercept DNS queries
- DNSSEC: detect modification to DNS results
- DoT: DNS over TLS, HTTPS (encryption)