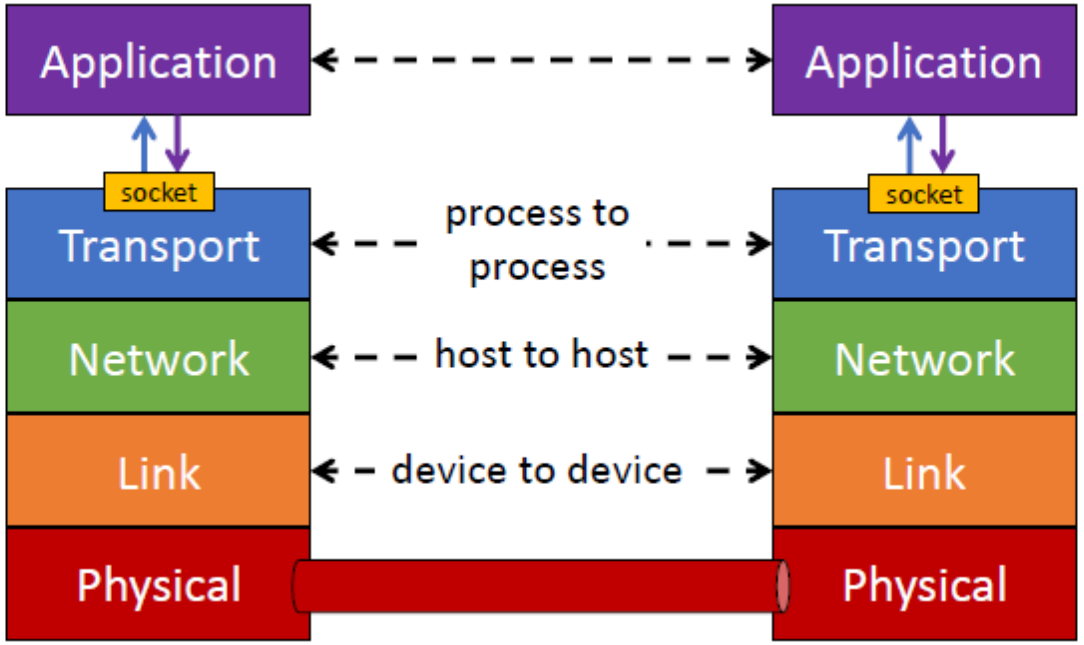
CS2105 Introduction to Computer Networks

Recap

- Application architecture
  - Client Server
    - Server
      - Always – on to wait for incoming requests
      - Provides the requested service to client
    - Client
      - Initiates contacts with the server
      - Requests services from the server
  - P2P
    - No always on server
    - No clear distinction between client and server
      - Any end host can be a client in 1 communication and yet be the server in another
    - Very scalable
  - Hybrid
    - Whatsapp

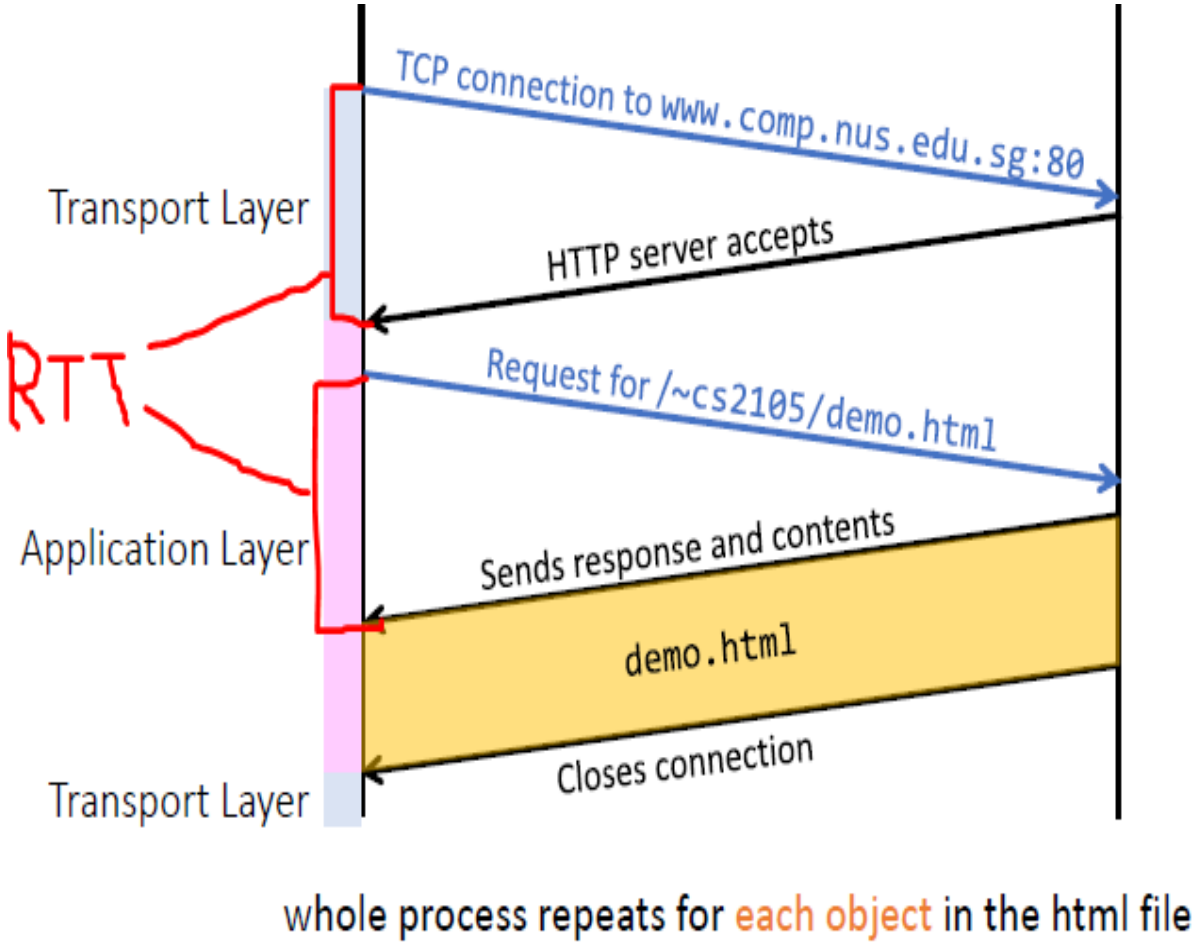
Recap

- Sockets
  - Interface through which a process (application) communicates with the transport layer



Recap

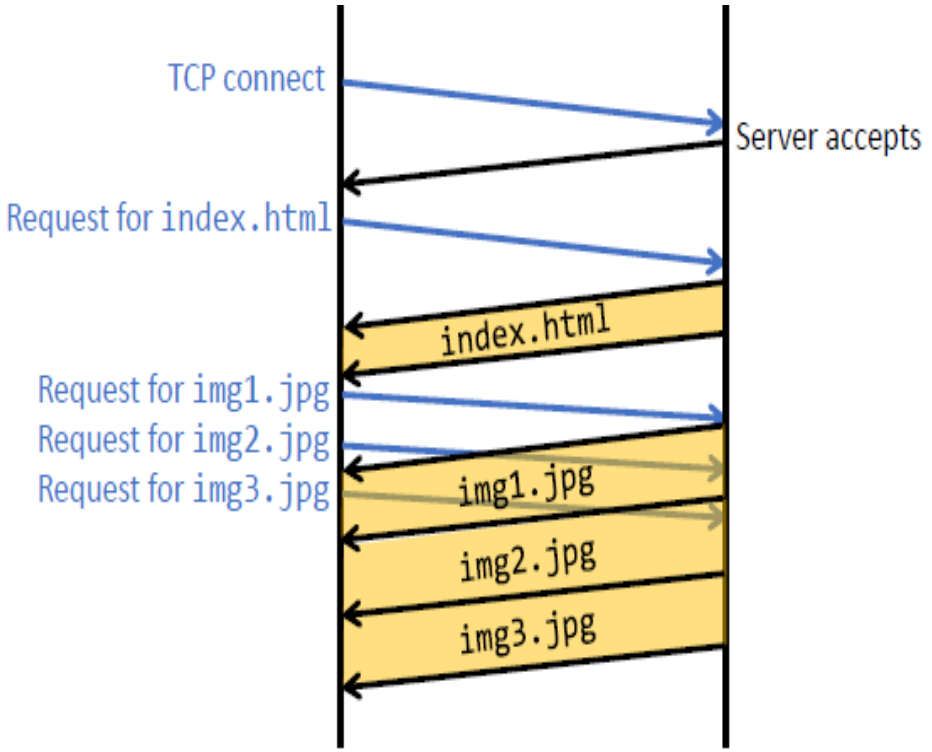
- HTTP 1.0
  - Non persistent
    - For each web resource, a **brand new** TCP connection request has to be sent out
  - Sequential
    - Massive overhead since each file has to be loaded 1 by 1



whole process repeats for each object in the html file

Recap

- HTTP 1.1
  - Persistent
    - Only 1 TCP connection request is required to load **ALL** the relevant files/HTML objects
    - Connection is left open after sending a web object.
    - Less overhead
  - Pipelined
    - New requests can be made even before receiving responses for older requests.
    - HOWEVER, they have to be received back in the order they were requested
      - Head-of-line blocking

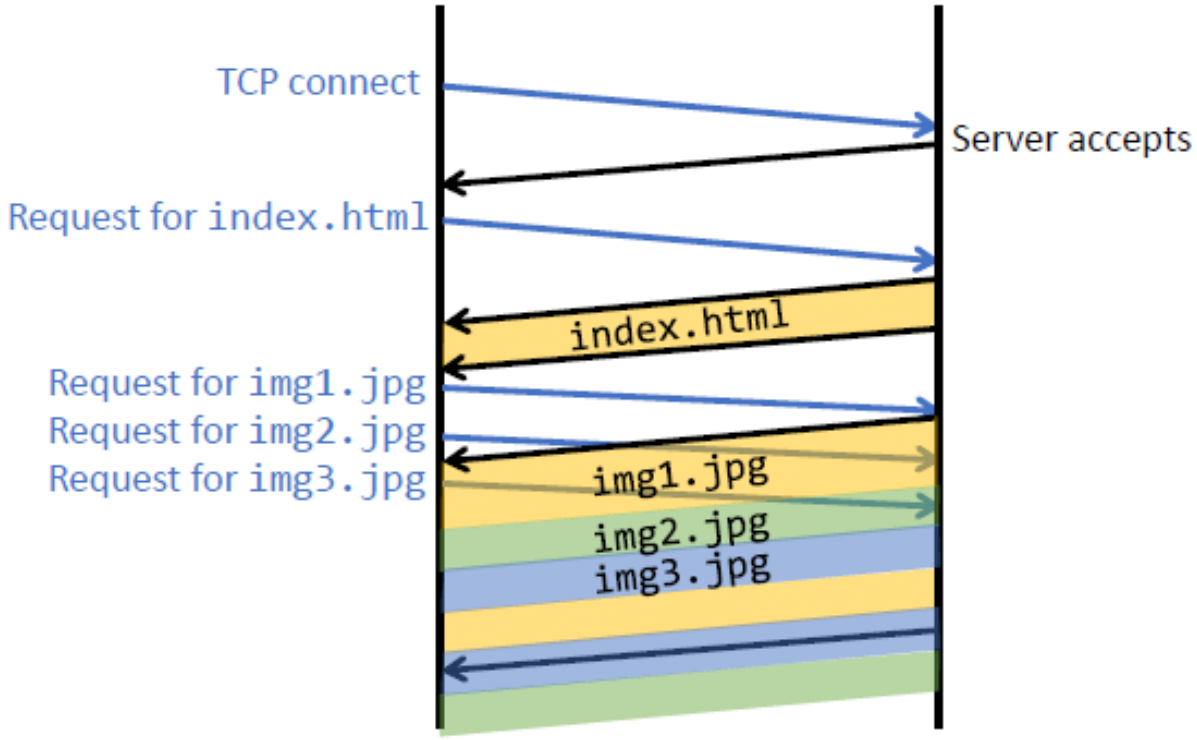


New request made even before receiving response of old requests

NOTE: HTML file is **always** loaded first!

Recap

- HTTP 2.0
  - In addition to being persistent and pipelined, it is **multiplexing**
    - Since fractions of the file can be received at different times, heavy requests will not block server
      - Head of line blocking faced by 1.1 solved



Response can come back in any order, even partially.

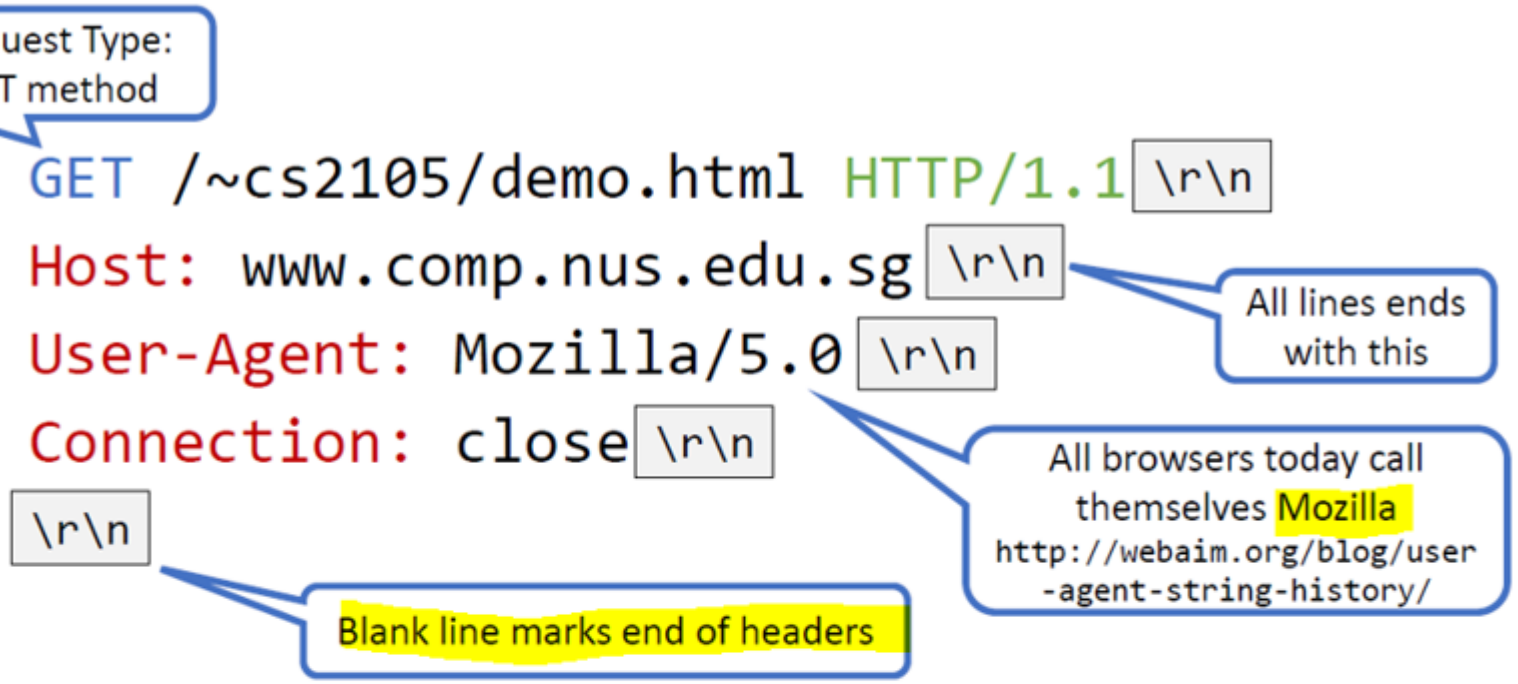
Recap

non-persistent HTTP issues: vs persistent HTTP:

- requires 2 RTTs per object
- OS overhead for each TCP connection
- browsers often open parallel TCP connections to fetch referenced objects

- server leaves connection open after sending response
- subsequent HTTP messages between same client/server sent over the same TCP connection
- client sends requests as soon as it encounters a referenced object (persistent with pipelining)
- as little as one RTT for all the referenced objects

Recap

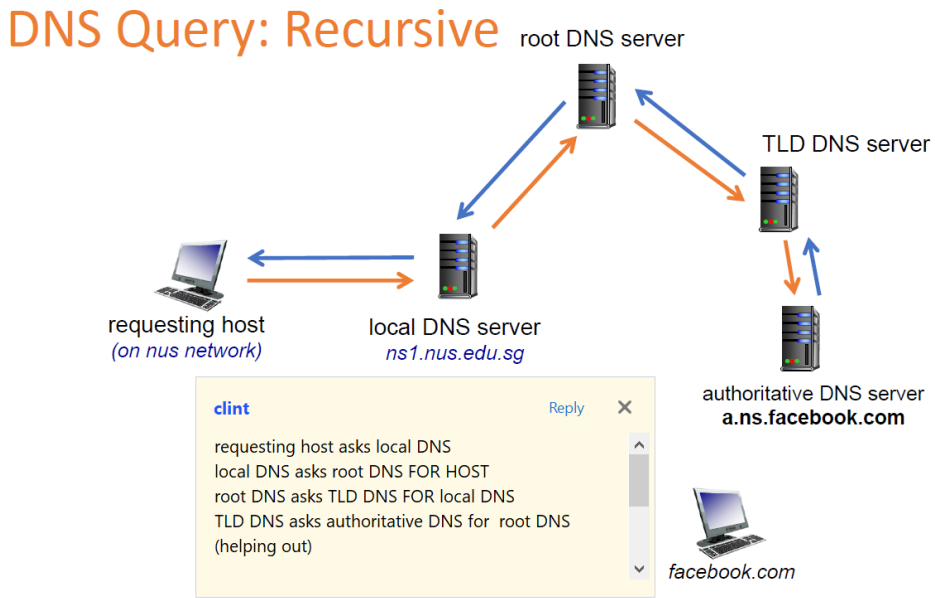


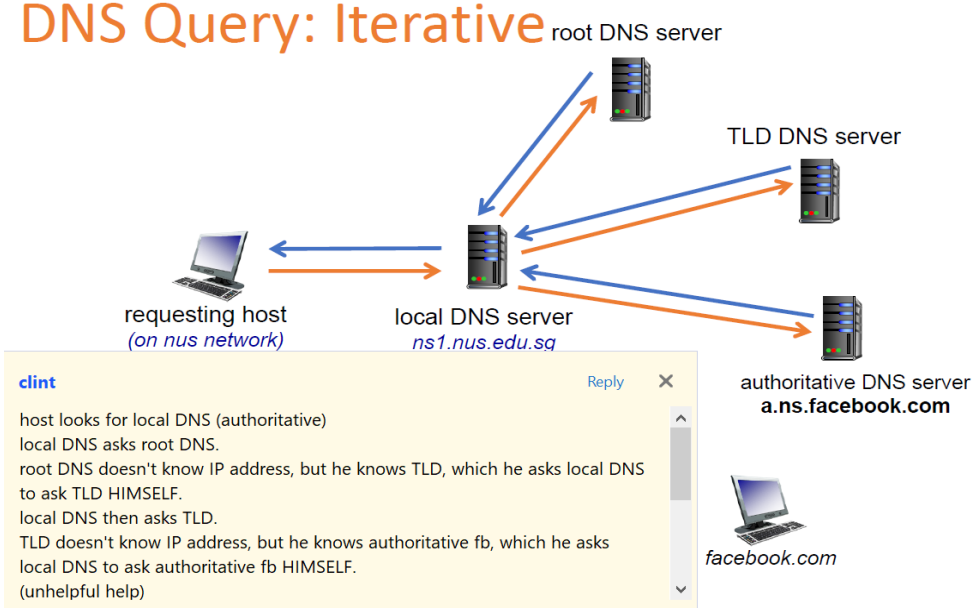
- First line is called the **request line**
  - 3 fields:
    - Method (`GET`, `POST`, `HEAD`, `PUT`, `DELETE`, etc.)
    - `URI` (Uniform Resource Identifier)
    - HTTP version
- Subsequent lines are called **header lines**

- Host: where the object resides (Host + `URI` = **URL**)
- User-agent: the browser type making the request
  - Important to know browser type as servers uses this information to decide which version of the object to send back

Recap

- DNS
  - Runs on UDP. Why?
    - Listens on port 53
  - Application layer protocol
  - **Acts as a contact book to translates a host's name to its corresponding IP address**
  - Recursive vs Iterative





Recap

- DNS
  - Resources are cached as resource records
    - Stored on local DNS servers
  - Every DNS query from a host is first sent to its local DNS server
    - Checks local cache to see if it already has a RR that stores the name to address translation for the query
      - Can find: saves a lot of time; else needs to go through

        root → TLD server → authoritative server
      - Cannot be found: Local DNS server then acts as a proxy to forward query into hierarchy if address is not found in the local cache

RR Format: <name, value, type, TTL>

Type	Name	Value
A (adress)	Hostname	IP Address
NS (name server)	Domain, e.g nus.edu.sg	Hostname of authoritative name server for domain
CNAME (canonical name)	Alias for real name, e.g. www.comp.nus.edu.sg	The real name, e.g. www0.comp.nus.edu.sg
MX (mail exchange)	Domain of email address	Name of mail server managing the domain

Tutorial Questions

Question 1

Consider the following HTTP request message sent by a browser.

```
GET /~cs2105/demo.html HTTP/1.1
Host: www.comp.nus.edu.sg
Connection: keep-alive
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/39.0.2171.99 Safari/537.36
Accept-Encoding: gzip, deflate, sdch
Accept-Language: en-US,en;q=0.8
```

a) What is the URL of the document requested by this browser?
www.comp.nus.edu.sg/~cs2105/demo.html

b) What version of HTTP is this browser running?
HTTP version 1.1

c) Does the browser request a non-persistent or a persistent connection?
The browser requests a persistent connection, as indicated by the header field 'Connection: keep-alive'.

d) What is the IP address of the host on which the browser is running?
IP address is not shown in HTTP message. One would be able to get such information from socket.

Question 2

The text below shows the header of the response message sent from the server in reply to the HTTP GET message in Q1 above. Answer the following questions.

HTTP/1.1 200 OK
Date: Tue, 20 Jan 2015 10:08:12 GMT
Server: Apache/2.4.6 (Unix) OpenSSL/1.0.1h
Accept-Ranges: bytes
Content-Length: 73
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive
Content-Type: text/html

a) Was the server able to successfully find the document or not?
The status code 200 and the phrase OK indicates that the server was able to locate the document successfully.

b) What time did the server send the HTTP response message?
The HTTP response message was formed on Tuesday, 20 Jan 2015 10:08:12 Greenwich Mean Time.

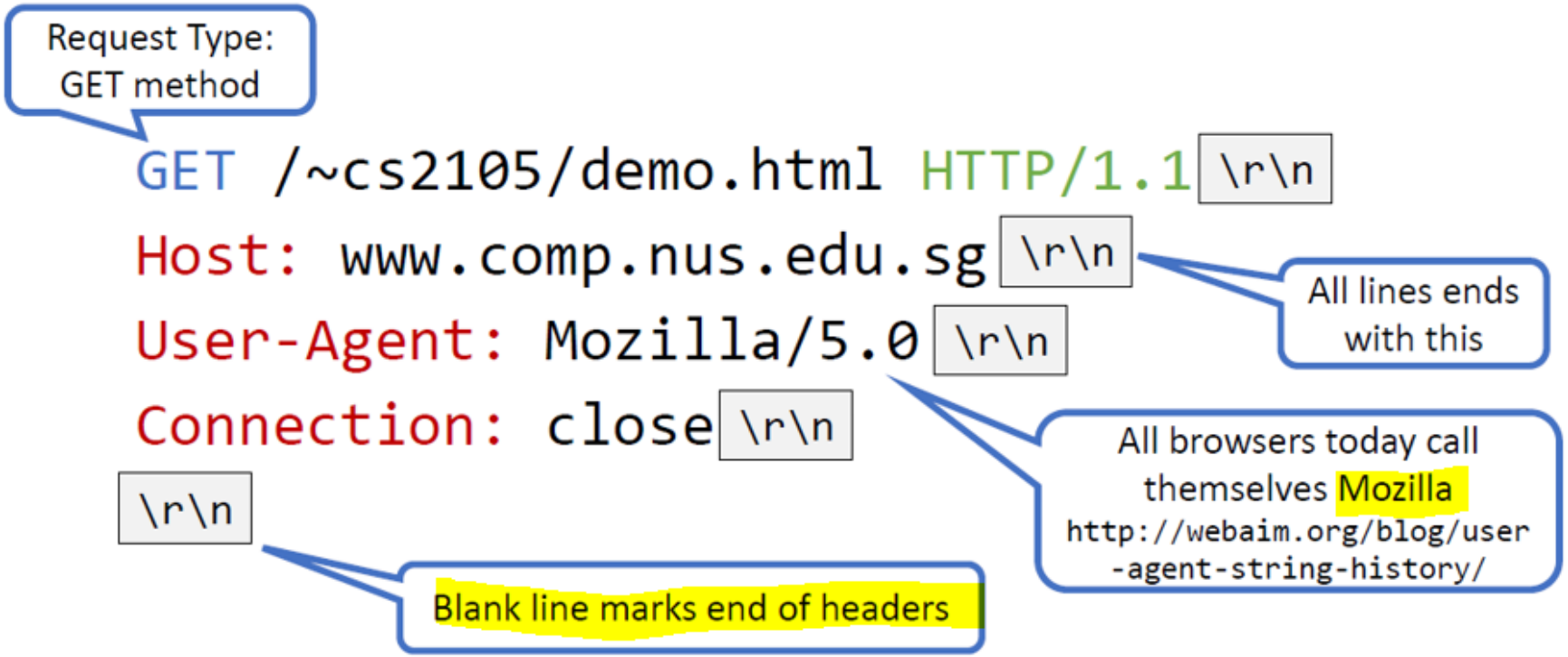
c) How many bytes are there in the document being returned?
There are 73 bytes in the document being returned.

d) Did the server agree to a persistent connection?
The server agreed to a persistent connection, as indicated by the header field 'Connection: Keep-Alive field'.

Question 3a

A user requests a Web page that consists of some text and three images. For this page, the client will send one request message and receive four response messages.

False. Download one object per request.



Question 3b, 3c & 3d

b) Two distinct Web pages (for example, www.mit.edu/research.html and www.mit.edu/students.html) can be sent over the same persistent connection.

True. They are on the same server.

c) The Date: header in the HTTP response message indicates when the object in the response was last modified.

False. Header field 'Date' indicates the server response time. The time the object is last modified is denoted by another header field 'Last-Modified'.

d) HTTP response messages never have an empty message body.

False. E.g. conditional GET whereby browser's cached copy is up-to-date.

Question 4

[Modified from KR, Chapter 2, P7] Suppose within your Web browser, you click on a link to obtain a Web page. The IP address for the associated URL is not cached in your local host, so a DNS lookup is necessary to obtain the IP address.

Suppose that **n** DNS servers are visited before your host receives the IP address from DNS; visiting them incurs an RTT of $\mathbf{D_{DNS}}$ per DNS server.

Further suppose that the Web page associated with the link contains **m** very small objects (in addition to the HTML page). Suppose the HTTP running is **non-persistent and non-parallel**. Let $\mathbf{D_{Web}}$ denote the RTT between the local host and the server of each object.

Assuming zero transmission time of each object, how much time elapses from when the client clicks on the link until the client receives all the objects?

To map from hostname to IP address: n x D_{DNS} (note: DNS is over UDP, so no need to establish connection).

To establish TCP connection and get the HTML page = D_{Web} + D_{Web}

To establish m TCP connections and get all m objects = m x (D_{Web} + D_{Web})

Total time = n x D_{DNS} + (m + 1) x 2 x D_{Web}

Question 5

[Modified from KR, Chapter 2, P8] Referring to the previous question, suppose that **three** DNS servers are visited. Further, the HTML file references **five** very small objects on the same server. Neglecting transmission delay, how much time elapses with:

a) Non-persistent HTTP with no parallel TCP connections?
$3 \times D_{DNS} + (5 + 1) \times 2 \times D_{Web}$ [Note that the 1 is for the HTML file!]

b) Non-persistent HTTP with the browser configured for five parallel connections?
$3 \times D_{DNS} + 2 \times D_{Web} + 2 \times D_{web}$ [for HTML file]

Need to fetch HTML file first ($2 \times D_{Web}$). Subsequently, the rest 5 objects can be fetched in parallel each using a TCP connection ($2 \times D_{Web}$).

c) Persistent HTTP with pipelining?
$3 \times D_{DNS} + 2 \times D_{Web} + D_{web}$ [for HTML file]

Need to fetch HTML file first ($2 \times D_{Web}$). The rest 5 objects can be fetched through the same TCP connection in parallel – no RTT for TCP handshake is needed.

Question 6

Do you know what is DNS cache poisoning? Search online for a real example.

DNS cache poisoning (a kind of DNS spoofing) is a computer hacking attack, whereby rogue DNS records are introduced into a DNS resolver's cache, causing the name server to return an incorrect IP address, diverting traffic to the attacker's computer (or any other computer). For example, DDoS (Distributed Denial of Service Attack) on a particular machine can be achieved via DNS cache poisoning.

Examples:

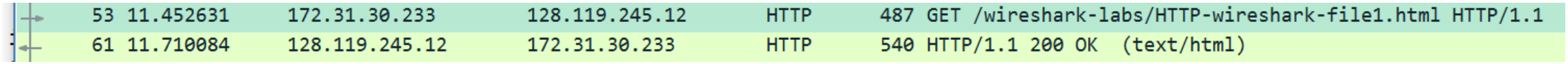
1. DNS Poisoning in China: http://www.howtogeek.com/161808/htg-explains-what-is-dns-cache-poisoning/

2. Angry Bird Website Defaced: https://arstechnica.com/security/2014/01/angry-birds-website-defaced-following-reports-it-enables-government-spying/

Question 8

1. What is the status code returned from the server to your browser?
Ans: 200

2. When was the HTML file that you are retrieving last modified at the server?
Ans: the value is denoted by the header field 'Last-Modified'.



Summary

- Application architecture
- Sockets
- **HTTP 1.0 vs 1.1 vs 2.0 (important!!)**
- HTTP Requests and Responses
- DNS

Extra Questions

Which of the following statements about DNS are true?

i. If the DNS servers are down, you cannot surf the web but you can still use e-mail.

ii. DNS maintains the records of hostnames to IP addresses.

iii. The root servers have to be accessed for every DNS query.

iv. DNS is an application layer protocol.

Extra Questions

A Web server stores a webpage that comprises a base HTML file and 5 images referenced by the base HTML file. The HTML file is 200 bytes and each image is 1,000 bytes. A client is connected to the Web server through a direct link of 1 Mbps. RTT is 100 milliseconds.

Suppose HTTP/TCP headers and control packets are of negligible size; time to close a TCP connection can be omitted. Which of the following correctly calculates the time the client uses a browser to download the webpage from the Web server?

(i) 341.6 milliseconds for persistent HTTP with pipelining

(ii) 741.6 milliseconds for persistent HTTP with no parallel requests (i.e. the next HTTP request is sent after the response for the previous HTTP request is received.)

(iii) 1241.6 milliseconds for non-persistent HTTP with no parallel TCP connections

(iv) 641.6 milliseconds for non-persistent HTTP with maximum 3 parallel TCP connections allowed

Thank you!

Answers:
ii & iv
All are correct!