National University of Singapore
School of Computing

CS2105 **Tutorial 2** Question paper

To students:

Due to time constraint, not all the questions will be discussed in class. Your tutor has the discretion to choose the questions to discuss (or you may request your tutor to discuss certain questions). Please go through the rest questions after class.

1. Consider the following HTTP request message sent by a browser.

GET /index.html HTTP/1.1 Version ✓

Host: www.example.org URL www.example.org/index.html.

Connection: keep-alive persistent ✓

User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/71.0.3578.98 Safari/537.36

Accept-Encoding: gzip, deflate

…

a) What is the URL of the document requested by this browser?

b) What version of HTTP is this browser running?

c) Does the browser request a non-persistent or a persistent connection?

d) What is the IP address of the host on which the browser is running?

93.184.216.34. Can't tell → separation of concern.

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

Content-Encoding: gzip

Content-Type: text/html; charset=UTF-8

Date: Wed, 23 Jan 2019 13:50:31 GMT time. ✓

Last-Modified: Fri, 09 Aug 2013 23:54:35 GMT

Connection: Keep-Alive

Content-Length: 606 bytes.

…

a) Was the server able to successfully find the document or not? Yes.

b) What time did the server send the HTTP response message?

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

d) Did the server agree to a persistent connection? Yes.

pipelining → 1 request not 3 objects.

3. True or false?

   a) 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. F

   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. T

   Same server.

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

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

   Conditional GET → Cached copy.

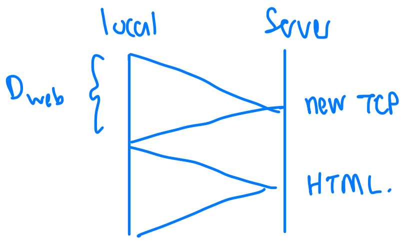
4. **[Modified from KR, Chapter 2, P7]** Suppose within your Web browser, you click on a link to obtain a Web page. The IP address for the associated URL is not cached in your local host, so a DNS lookup is necessary to obtain the IP address.

   DNS over UDP ⇒ no need connection unlike TCP

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

   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 D_{Web} denote the RTT between the local host and the server of each object. Total = $2D_{web}(m+1)$

   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?



Time taken = $nD_{DNS} + 2D_{web}(m+1)$

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? $3D_{DNS} + 12D_{web}$ ✓
   b) Non-persistent HTTP with the browser configured for five parallel connections? $3D_{DNS}$ + ~~$4D_{web}$~~
   c) Persistent HTTP with pipelining? $3D_{DNS} + 3D_{web}$ ✓

Consider transmission time for parallel. in exam.

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

occurs when a threat actor feeds false information into the DNS cache, thereby making a user's web browser return an incorrect response.

an attacker diverts traffic from a legitimate server to a malicious/dangerous server.

In 2010, an Internet service provider outside of China mistakenly configured its DNS servers to fetch information from DNS servers in China. It fetched the incorrect DNS records from China and cached them on its own DNS servers. Other Internet service providers fetched DNS information from that Internet service provider and used it on their DNS servers. The poisoned DNS entries continued to spread until some people in the US were blocked from accessing Twitter, Facebook, and YouTube on their American Internet service providers. The Great Firewall of China had "leaked" outside of its national borders, preventing people from elsewhere in the world from accessing these websites. This essentially functioned as a large-scale DNS poisoning attack. (Source.)

7. Wireshark Introduction

   Wireshark is a tool for observing the messages exchanged between executing protocol entities. It observes messages being sent and received by applications and protocols running on your computer.

   Download and install the Wireshark software:

   - Go to http://www.wireshark.org/download.html and download and install the Wireshark binary for your computer.

   Taking Wireshark for a test run:

   1. Start up your web browser.
   2. Start up the Wireshark software.
   3. To begin packet capture, select the Capture pull down menu and select Interfaces.
   4. Click on Start for the interface on which you want to begin packet capture.
   5. While Wireshark is running, enter the URL: http://gaia.cs.umass.edu/wireshark-labs/INTRO-wireshark-file1.html and have that page displayed in your browser.
   6. After your browser has displayed the INTRO-wireshark-file1.html page, stop Wireshark packet capture by selecting stop in the Wireshark capture window. You now have live packet data that contains all protocol messages exchanged between your computer and other network entities!
   7. Type in "http" (without the quotes, and in lower case – all protocol names are in lower case in Wireshark) into the display filter specification window at the top of the main Wireshark window. Then select Apply. This will cause only HTTP message to be displayed in the packet-listing window.

   Congratulations! You've now completed the Wireshark introduction.

8. Wireshark: HTTP GET/response interaction

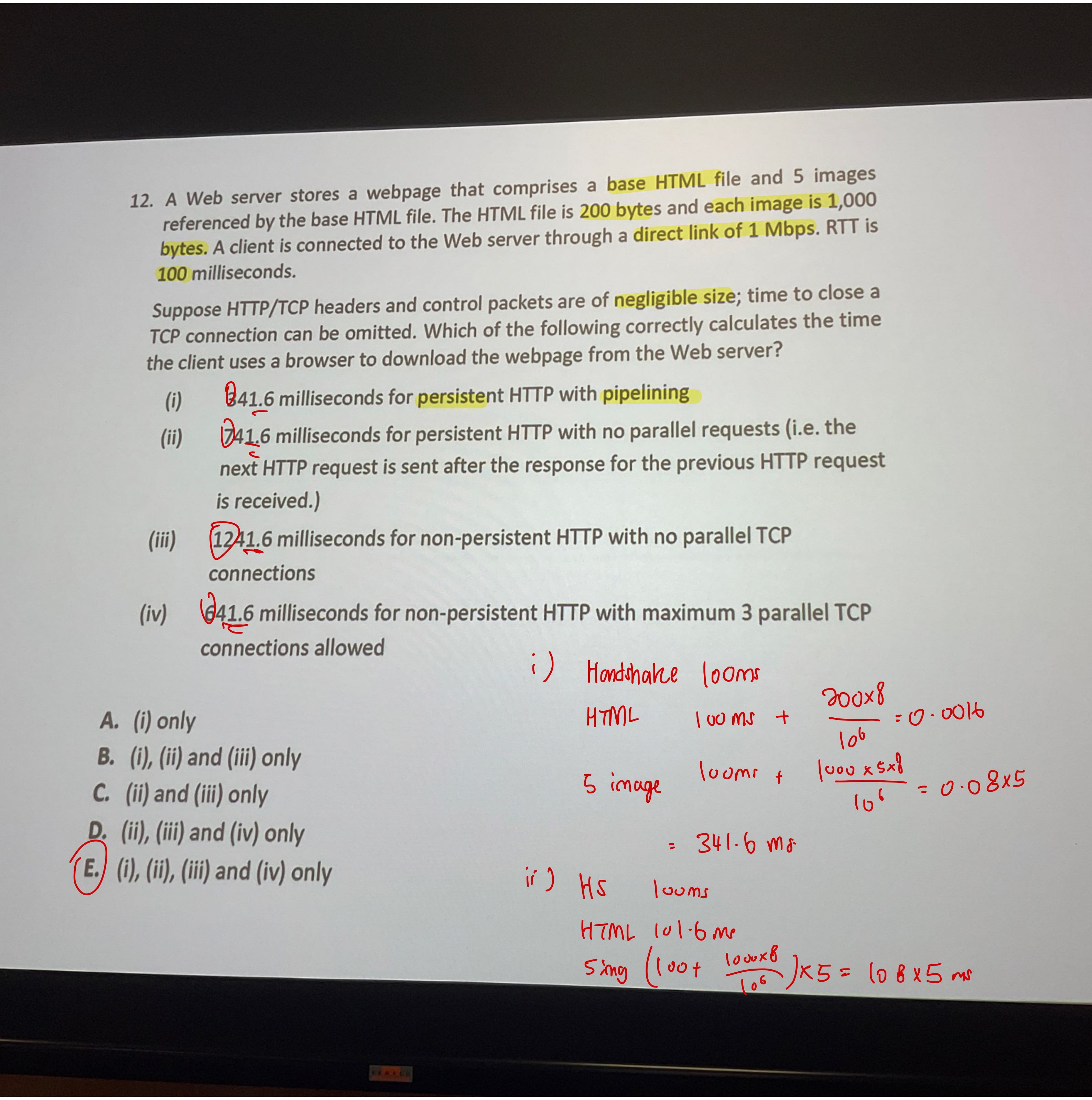
   Let's begin our exploration of HTTP by downloading a very simple HTML file, and contains no embedded objects. Do the following:

   1. Start up your web browser.
   2. Start up the Wireshark packet sniffer. Enter "http" in the display-filter-specification window and begin Wireshark packet capture.
   3. Enter the following to your browser http://gaia.cs.umass.edu/wireshark-labs/HTTP-wireshark-file1.html.
   4. Stop Wireshark packet capture.

   Now answer the following questions:

   1. What is the status code returned from the server to your browser? 200 OK
   2. When was the HTML file that you are retrieving last modified at the server?

      Sun, 28 Aug 01:09:14 GMT.



12. 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

A. (i) only

B. (i), (ii) and (iii) only

C. (ii) and (iii) only

D. (ii), (iii) and (iv) only

[✓ E.] (i), (ii), (iii) and (iv) only

i) Handshake 100ms

HTML 100 ms + $\frac{200 \times 8}{10^6} = 0.0016$

5 image 100ms + $\frac{1000 \times 5 \times 8}{10^6} = 0.08 \times 5$

= 341.6 ms

ii) HS 100ms

HTML 101.6 ms

5 img $\left(100 + \frac{1000 \times 8}{10^6}\right) \times 5 = 108 \times 5$ ms

iii) HTML = 100+100+1.6

5 img : $\left(100 + 100 + \frac{1000 \times 8}{10^6}\right) \times 5$

iv) non-persistent w parallel

HTML : 201.6

3 img : $100 + 100 + \frac{3 \times 1000 \times 8}{10^6}$

2 img : $100 + 100 + \frac{2 \times 1000 \times 8}{10^6}$.