Name_____ Student ID_____ Department/Year_____

2nd Examination

Introduction to Computer Networks (Hybrid)
Class#: EE 4020, Class-ID: 901E31110
Spring 2021

13:20-14:10 Thursday
April 15, 2021

Cautions

1. There are in total 100 points to earn. You have 50 minutes to answer the questions. Skim through all questions and start from the questions you are more confident with.
2. Use only English to answer the questions. Misspelling and grammar errors will be tolerated, but you want to make sure with these errors your answers will still make sense.

1. (ch21, 9pt) YouTube is an on-demand video streaming service (i.e., playback of stored videos). YouTube Live is a live video streaming service. Google Meet is an interactive video streaming service. Recall the QoS requirement for video service and tell which of the following statements are true and which are false.
   (1) YouTube requires no loss. (1pt).
   (2) YouTube Live requires no loss. (1pt).
   (3) Google Meet requires no loss. (1pt).
   (4) YouTube requires 10kbps to 5Mbps bandwidth (1pt).
   (5) YouTube Live requires 10kbps to 5Mbps bandwidth (1pt).
   (6) Google Meet requires 10kbps to 5Mbps bandwidth (1pt).
   (7) YouTube requires 100s ms delay (1pt).
   (8) YouTube Live requires 100s ms delay (1pt).
   (9) Google Meet requires 100s ms delay (1pt).

Sample Solution:

(1)-(3), (7) False

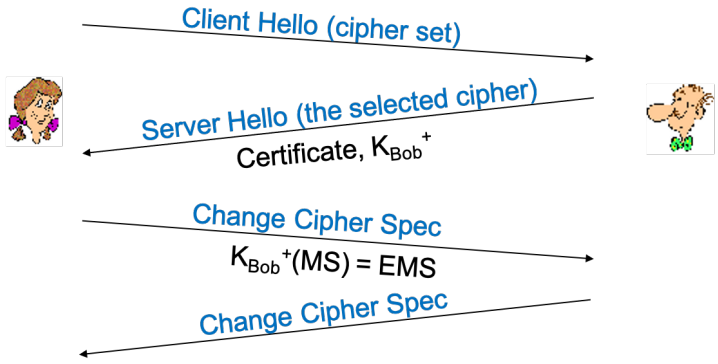
(4)-(6), (8)-(9) True

2. (ch21, 3pt) You are asked to reconsider the use of transport layer protocol for YouTube, YouTube Live, and Google Meet video transmission.
   (1) For YouTube, would you take TCP or UDP? Why? (1pt)
   (2) For YouTube Live, would you take TCP or UDP? Why? (1pt)
   (3) For Google Meet, would you take TCP or UDP? Why? (1pt)

Sample Solution:

One could argue one way or another. Just choose and justify.

3. (ch21, 8pt). The purpose of SSL/TLS is to encrypt the packets transmitted in a TCP connection so the content (often as readable text) will be confidential. To allow one side to decrypt the packets sent from another, the two sides need to established a shared, secret, symmetric key. The handshake phase of SSL/TLS is designed for the very purpose. Illustrated below is the handshake process of TLS 1.2.



In the 1st round trip, the Client Hello message tells the server the set of ciphers available. The Server Hello tells the client the cipher selected (among the set of ciphers suggested in the Client Hello). In the meantime, the server sends its own hostname and the certificate endorsed by a 3rd party authority. The server sends also its public key (i.e., K_{Bob}^{+} in the example above).

In the 2nd round trip, the client generates the symmetric key (i.e., MS) and set the cipher spec accordingly. Before sending the MS out, the client encrypts MS in the server's public key, (i.e., $K_{Bob}^{+}(MS)$). The server receiving the encrypted symmetric key (i.e., EMS) decrypts with its private key and sets its cipher spec accordingly as well. With the Change Cipher Spec message back to the client, the client and server are ready to encrypt and decrypt the forthcoming data using the shared symmetric key.

(1) What is the purpose of the symmetric key (MS) in a TLS connection? (2pt)
(2) What is the purpose of the server sending its public key (K_{Bob}^{+})? (2pt)
(3) What is the purpose of the server sending its hostname and the certificate? (2pt)
(4) The secure version of HTTP, i.e., HTTPS, is enabled by SSL/TLS running on top of TCP. Once the TCP connection is established, TLS handshake will take place before the HTTP messages can be sent confidentially. Now, if it takes 2 RTTs to download a simple web page in HTTP. How many RTTs will be required to download the simple page using HTTPS based on TLS 1.2? (1pt)
(5) Continue from (4). How many RTTs will be required to download the simple page using HTTPS based on TLS 1.3? (1pt)

Sample Solution:

(1) data confidentiality – to encrypt the data in the connection
(2) MS key exchange confidentiality – for the client to encrypt the MS such that it's only decryptable by the server (using server's private key)
(3) Identity theft prevention –to verify the server's identity
(4) 4RTT
(5) 3RTT

4. (ch22, 14pt) HTTP messages are ASCII readable. One can use a remote login service like telnet to emulate sending of HTTP messages and to examine the messages from the server. Let's ssh to the PA server and telnet the Web server hosting the course page – `homepage.ntu.edu.tw`.
   (1) Log in to the PA server with the team's username and password and then create a subdirectory `exam2-<student ID>`, just so that we know you've been there. (2pt)
   (2) Go to the `exam2-<student ID>` subdirectory. telnet to the Web server by this command: `telnet homepage.ntu.edu.tw 80`. Tell from the output the IP address of `homepage.ntu.edu.tw`. (2pt)
   (3) Once connected, one may type up a text message in HTTP Request message format to see how the Web server will respond. Now send the following HTTP Request message. Copy and paste the status line and header lines of the corresponding HTTP Response message here. (2pt)

```
GET /~pollyhuang/ HTTP/1.0

```

   (4) Modify the HTTP Request so that the server will not send the page if it is not modified since the time it is last changed. Tell the HTTP Request message you sent. (2pt)
   (5) Continue from (4). Copy and paste the status line and header lines of the HTTP Response message you received here. (2pt).
   (6) Modify the HTTP Request so that the Web server will return the part of the page in byte range 1000 and 1005. Tell the HTTP Request message you sent. (2pt)
   (7) Continue from (6). Copy and paste the data part of the HTTP Response message you receive here. (2pt).

Sample Solution:

(1) We'll check online.
(2) 140.112.2.140
(3) It should be a "200 OK" response.
(4) Adding a header line: `If-Modified-Since: Mon, 22 Mar 2021 04:59:36 GMT`
(5) It should be a "304 Not Modified" response.
(6) Adding a header line: `Range: bytes=1000-1005`
(7) "`media `".

5. (ch22, 10pt) We have tested HTTP 1.0's behavior in the previous problem set. Let's now observe HTTP 1.1's behavior.
   (1) Send the foloowing HTTP Request message. Copy and paste the status line and header lines of the HTTP Response message here. (2pt)

```
GET /~pollyhuang/ HTTP/1.1

```

   (2) Modify the HTTP Request message as follows. Copy and paste the status line and header lines of the HTTP Response message you receive here. (2pt)

```
GET /~pollyhuang/ HTTP/1.1
Host: homepage.ntu.edu.tw

```

   (3) Why is the `Host:` header line required in HTTP 1.1? (3pt)
   (4) Redo the HTTP Request in 4(3) and 5(2) again. Compare the time to close the connection. You should feel the delay being longer in one case (than the other). Which case is it? And why such a difference? (3pt)

Sample Solution:

(1) It should be a "404 Not Found" response.
(2) It should be a "200 OK" response.
(3) To allow multiple Web servers running on one physical machine. Multiple hostnames mapped to the same IP address.
(4) 6(2) longer. HTTP 1.1 introduces the persistent connection mode. The wait is to keep the connection alive for potential reference object requests.

6. (ch23, 12pt) Email messages are ASCII readable as well. Just like how telnet is useful debugging Web server configuration, we can try EE department's email server out – `cc.ee.ntu.edu.tw`.
   (1) Log in to the PA server with the team's username and password. Go to the `exam2-<student ID>` subdirectory. telnet to the email server by this command: `telnet cc.ee.ntu.edu.tw 25`. Tell, from the output, the IP address of `cc.ee.ntu.edu.tw`. (2pt)
   (2) Type up the following line by line. Hit the enter key at the end of each line. The points will be awarded when the TA receives the email. (4%)

```
HELO blahblahblah
MAIL FROM: <your ntu email address>
RCPT TO: r08942157@ntu.edu.tw
RCPT TO: pollyhuang@ntu.edu.tw
RCPT TO: <your ntu email address>
DATA
To: r08942157@ntu.edu.tw, <your gmail address>
From: <your ntu email address>
Subject: Exam #2 Problem Set 6

Wait a minute. If this works, it means anyone learning about
the existence of cc.ee.ntu.edu.tw can spam anyone else in the
world.
.
QUIT
```

   (3) Would you be able to receive the email at your gmail address? Why? (2pt)
   (4) Would Polly be able to receive the email at her ntu address? Why? (2pt)
   (5) Do you find it easy to pretend as someone else sending emails? Why? (2pt)

Sample Solution:

(1) 140.112.18.7
(2) We'll see if your email arrives.
(3) No. your gmail address is not in the RCPT TO list
(4) Yes. RCPT TO is the command assigning the email recipient.
(5) Feel free to tell what you feel and elaborate how come you feel this way.

7. (ch24, 3pt) Complexity at the edge, is a design principle the Internet engineers often exercise.
   (1) Tell what it means to leave the complexity at the edge. (1pt)
   (2) Tell the benefits of leaving the complexity at the edge. (1pt)
   (3) Recall the protocol designed with the principle in mind. (1pt)

Sample Solution:

(1) Pushing functionality that's yet to evolve in the future to the edge of the Internet.
(2) It is easy to evolve/upgrade the functionality without the need to reboot the core that some parts of the Internet may depend critically on. Or to keep the core simple and therefore fast and reliable.
(3) DNS

8. (ch24, 4pt) DNS is a large distributed database of reference records (RRs). Which of the following are reasons that RRs should not be centrally stored?
   (1) The centralized server will be a single point of failure. One machine down, the entire WWW down. (1pt)
   (2) The traffic volume concentrates at a single point, leading to traffic congestion at the server. (1pt)
   (3) Many of the clients on the Internet will experience long delay connecting to the only server. (1pt)
   (4) It is hard to upgrade or maintain the server when it is the sole machine the Internet relies on. (1pt)

Sample Solution:

(1)-(4)

9. (ch24, 7pt) Try not to jump to the conclusion that the access net must be down when you suddenly have trouble getting any of the Internet services. A DNS server failure might be the cause. To debug, one uses `dig`. Below is a demonstration of how you can test whether the local DNS server is able to get the RR entry for `www.google.com`. You see when the DNS query is successful, `dig` outputs the DNS query and the reply received. Examine the output and see if you can answer the following questions.

```
$ dig www.google.com
; <<>> DiG 9.10.6 <<>> www.google.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 15356
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 512
;; QUESTION SECTION:
;www.google.com.             IN   A

;; ANSWER SECTION:
www.google.com.         218  IN   A    172.217.27.132

;; Query time: 6 msec
;; SERVER: 8.8.8.8#53(8.8.8.8)
;; WHEN: Mon Apr 12 23:45:17 CST 2021
;; MSG SIZE  rcvd: 59
```

   (1) What is the local DNS server's IP address? (1pt)
   (2) What is www.google.com's IP address? (1pt)
   (3) Is the DNS message sent via UDP or TCP? (1pt)
   (4) Go on to the PA server and login with the team's username and password. Create (if you have not yet done so) and go to the `exam2-<student ID>` subdirectory. Store the output of `dig homepage.ntu.edu.tw` to `dig-ntu.txt`. Leave the file there for us to grade. (1pt)
   (5) Continue from (4). Store the output of `dig @8.8.8.8 homepage.ntu.edu.tw` to `dig-google.txt`. Leave the file there for us to grade. (1pt)
   (6) The output of the default local DNS server is different from that of the 8.8.8.8 (a public DNS server provided by Google). Why do you think there's such a difference?

(2pt)

Sample Solution:

(1) 8.8.8.8
(2) 172.217.27.132
(3) UDP
(4) We'll check
(5) We'll check
(6) Will accept anything reasonable. Here is one possibility:
The default local DNS server on the PA server is one of the ntu.edu.tw DNS servers. It tends to cache RRs related to the ntu.edu.tw domain as the users within ntu.edu.tw are more likely to access these RR entries. Relatively speaking, 8.8.8.8 on the other hand is provided by Google as a public service. Access to RRs related to ntu.edu.tw is much less likely.

10. (ch25, 4pt) Consider the following 4 scenarios. Tell whether you think the service runs in a client-server model or the peer-to-peer model and why.
    (1) Students answering each other's questions on the class's Slack workspace. (1pt)
    (2) Home buyers getting a loan from a bank. (1pt)
    (3) Party goers bringing their own dishes to a party. (1pt)
    (4) Sellers on eBay.com bidding items for sale as well. (1pt)

Sample Solution:

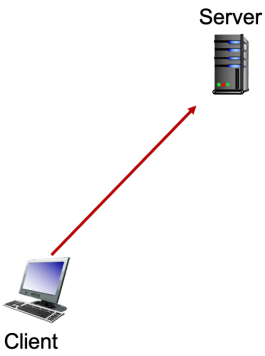
(1) P2P. Students answer questions while they may ask questions.
(2) CS. The bank being the big server allows high-$$ loan.
    Or P2P. The bank is essentially a very resourceful peer, which takes low-$$ loans as well offering interest to savings account holders.
(3) P2P. Participants share food.
(4) P2P. Sellers are also buyers

11. (ch25, 4pt) Recall the minimum distribution time transferring a file from a source to N receivers. The peer-to-peer approach scales better than the client-server approach. How come most of the popular Internet services are still based on the client-server model? Give at least 2 arguments against using the peer-to-peer model.

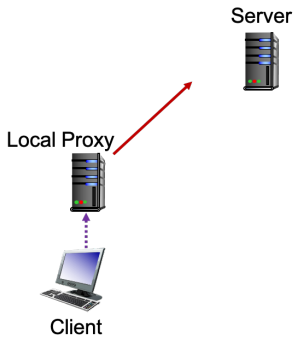
Sample Solution:

- Intermittent connectivity. Some receivers might not be on all the time to help uploading content.
- Selfish peer. Some receivers might leave the p2p net entirely and selfishly.
- NAT traversal required. Some receivers might not be directly reachable.
- Worst-case distribution time unpredictable. The peers come and go. The file transfer time can potentially be longer than in a client-server system. Some users would prefer a system that they know better what to expect.
- Hard to maintain. No one has full control of a P2P system, which makes it harder to debug or optimize.
- Bad image. Some p2p systems are exploited to distribute illegitimate content.
- Any argument that makes sense

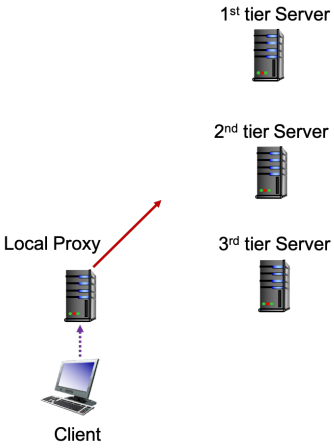
12. (ch26, 6pt) Internet services are inherently distributed systems. Depending on the data volume, access pattern, geographic span, and etc., some of the services can live on a simple, flat structure, but others require a complex, multi-level hierarchy. Depicted below are four common structures.



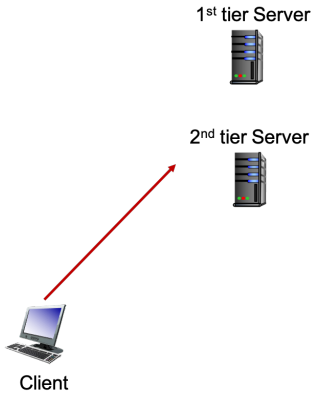
Structure A



Structure B



Structure C



Structure D

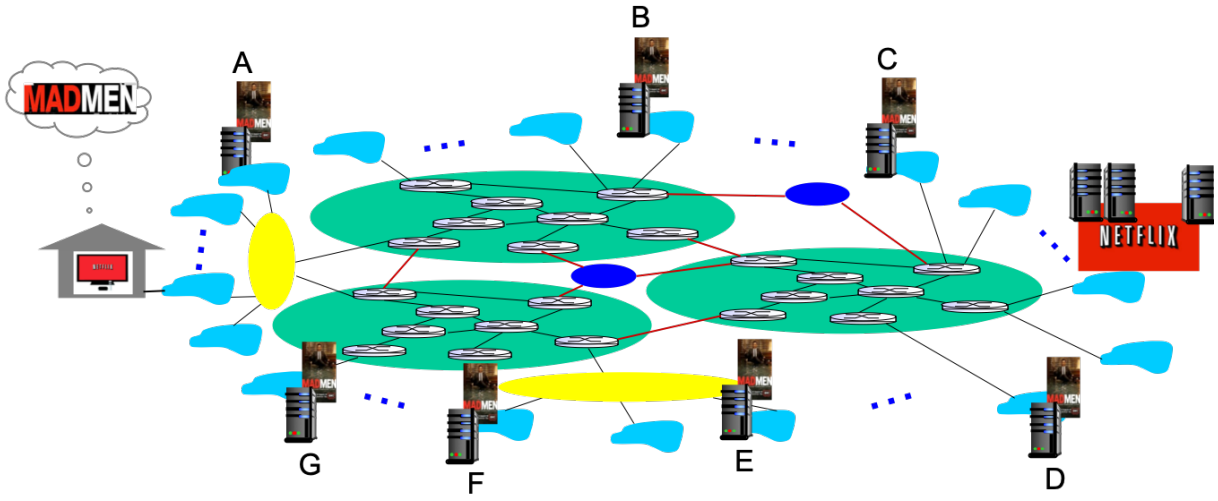
(1) Which structure does FTP operate on? (1pt)
(2) Which structure does the basic HTTP operate on? (1pt)
(3) Which structure does HTTP with proxy server operate on? (1pt)
(4) Which structure does SMTP operate on? (1pt)
(5) Which structure does DNS operate on? (1pt)
(6) Which structure does Netflix operate on? (1pt)

Sample Solution:

A, A, B, B, C, D

13. (ch26, 6pt) Illustrated below is Netflix's CDN. In the scenario, 7 nodes store the movie that the customer is looking for. The client program implements (a) a selection algorithm to select the CDN node first. To determine the stream rate, the client program implements (b) the DASH protocol. In that, the client program selects the best stream within the available bandwidth to the selected node.

Suppose the ping delay from the customer to A, B, C, D, E, F, G are 20, 80, 120, 160, 100, 60, 40ms respectively. The available bandwidth from A, B, C, D, E, F, G are 1, 10, 25, 30, 20, 7, 4 Mbps respectively. The movie is encoded into 4 resolutions 1080p, 720p, 480p, and 360p whose bitrates are 12, 5, 2.5, and 0.8 Mbps respectively.



(1) If (a) implements the shortest delay first, which node will be selected and which resolution video will be streamed? (1%)
(2) If (a) implements the best available bandwidth first, which node will be selected and which resolution video will be streamed? (1%)
(3) If (a) implements the best available bandwidth first with ping delay < 100ms, which node will be selected and which resolution video will be streamed? (1%)
(4) If (a) implements the shortest delay first with video resolution requirement at 1080p, which node will be selected and which resolution video will be streamed? (1%)
(5) If (a) implements the best quality first with ping delay < 100ms, which node will be selected and which resolution video will be streamed? (1%)
(6) If (a) implements the best quality first with ping delay < 100ms and available bandwidth < 5Mbps, which node will be selected and which resolution video will be

streamed? (1%)

Sample Solution:

(1) Node A. 360p.
(2) Node D. 1080p.
(3) Node B. 720p
(4) Node E. 1080p
(5) Node B or F. 720p
(6) Node G. 480p

14. (PA3+PA4, 5pt) We have tested your PA3.go on the PA server with a text file with \n at the end of each line. To display a text file with proper line breaks, it is necessary to append \n at the end of a line. This is conveniently done as we hit the enter key to move to the next line in an editor. Last line of a file is a little tricky however. Appending \n is not necessary because there are no more lines to break. Some editors, therefore, choose to not add \n at the end of last line.

    Go on to the PA server and login with the team's username and password. Create (if you have not yet done so) a subdirectory `exam2-<student ID>`. Copy the original PA3.go and PA4.go over to the `exam2-<student ID>` subdirectory. Rename them PA3-exam2-p14.go and PA4-exam2-p14.go correspondingly. Try rewrite them so that you can handle text files whose last line ends: (1) with \n and (2) without \n.

15. (PA3+PA4, 5pt) Many of you have raised the issue of how a line ends differently in Windows vs. Unix. In Windows, the end of line contains two non-printable characters – \r\n. In Unix, just one – \n. The trouble is that scanner.Scan() ignores both \r and \n when it reads a line. Combined with the assumption that each line ends with only an \n (Unix text file). Polly's PA4 server adds only 1 byte back tracking the file size and therefore underestimates the bytes received for a Windows file.

    Go on to the PA server and login with the team's username and password. Create (if you have not yet done so) a subdirectory `exam2-<student ID>`. Copy the original PA3.go and PA4.go over to the `exam2-<student ID>` subdirectory. Rename them PA3-exam2-p15.go and PA4-exam2-p15.go correspondingly. Try rewrite them so that you can handle text files created in either Windows or Unix.