Name_____ Student ID_____ Department/Year_____

1st Examination

Introduction to Computer Networks (Online)
Class#: EE 4020, Class-ID: 901E31110
Spring 2022

10:20-12:10 Thursday
March 23, 2022

Cautions

1. There are in total 100 points to earn. You have 90 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. (ch11, 3pt) Below are a variety of devices on the Internet today. Which part of the Internet do they belong – (a) edge or (b) core?
   (1) A WiFi AP at home (1pt)
   (2) A backbone router (1pt)
   (3) A Web server (1pt)

Sample Solution:

(1) (a)
(2) (b)
(3) (a)

2. (ch13, 5pt) Recall the packet switching and circuit switching principle, select the keywords that are characteristics of a circuit switched network. (a) Contention, (b) Delay jitter, (c) Call setup, (d) Reservation, (e) Packet loss.

Sample Solution:

(c)(d)

3. (ch13, 2pt) Why does a packet switched network need to store and then forward a packet?

Sample Solution:

In a packet switched network, packets are forwarded over routers towards the destination. The forwarding function depends on the destination address (and potentially other info in the packet) which a router won't see until storing the packet.
(no partial credit)

4. (ch13, 4pt) Tell the tradeoff of sending large vs small packets in a packet switched network.
   (1) What is the benefit of sending large packets? (2pt)
   (2) What is the benefit of sending small packets? (2pt)

Sample Solution:

(1) lower packet header overhead. There are two parts in a packet -- the header that contains the control info and the payload that contains the actual data the networked

applications are sending to each other. The packet header size is fixed. Using a larger packet size gives a higher proportion of bits for data.
(2) shorter file transfer time. When the packet size is small, the time to store and forward the first packet will be short. The subsequent packets will follow in a pipelined way, which do not add more store-and-forward time. This allows a shorter file transfer time overall.
(no partial credit)

5. (ch14, 8pt) `ping` provides another way to investigate the round-trip time (RTT) to a remote machine. The way it works is very simple – sending a request to the remote machine and the remote machine sending back a response. By calculating the difference in the timestamp of the request and response, `ping` shows the measured RTT on the screen. Using the `–c` flag, one can limit the number of request-response probes.

   Let's `ping` to NTU, NCHU, and NCKU's Web servers. These 3 universities are located in northern, central, and southern TW respectively. Let's see if the machines that are geographically closer give also a shorter RTT.

   (1) Login to the PA workstation with your exam account. Create a directory "exam1" and move to the directory. (1pt)
   (2) `ping` to www.ntu.edu.tw and create a file `ntu.txt` to contain the output as follows. You should see 5 RTT measurements in the output file. Leave the file there for grading (1pt).

   ```
   $ ping –c 5 www.ntu.edu.tw > ntu.txt
   ```

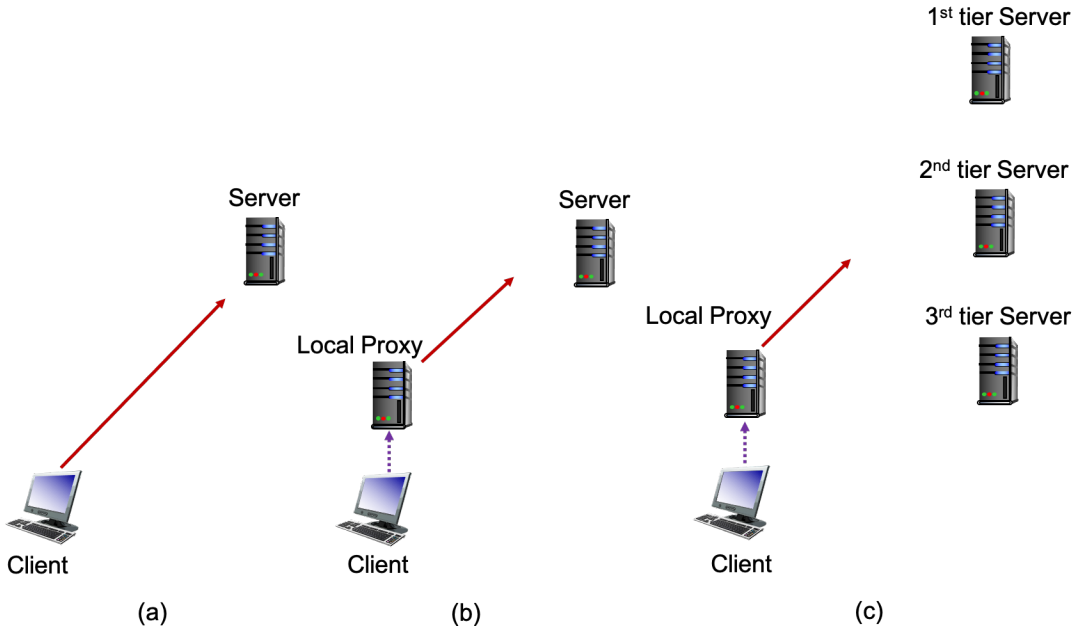
   (3) Tell the average RTT from the PA workstation to www.ntu.edu.tw. (1pt).
   (4) `ping` to www.nchu.edu.tw and create a file `nchu.txt` to contain the output and leave the file there for grading (1pt).
   (5) Tell the average RTT from the PA workstation to www.nchu.edu.tw. (1pt).
   (6) `ping` to www.ncku.edu.tw and create a file `ncku.txt` to contain the output and leave the file there for grading (1pt).
   (7) Tell the average RTT from the PA workstation to www.ncku.edu.tw. (1pt).
   (8) Are the machines geographically closer showing shorter RTTs? (1pt)

Sample Solution:
Will check on the PA workstation
Points for (3)(5)(7) will be credited when they are consistent with the outcome of (2)(4)(6)
Points for (8) will be credited when (2)-(7) complete

6. (ch15, 4pt) Consider the following protocols and tell which of the 4 layers they belong to – (a) application Layer, (b) transport Layer, (c) network Layer, (d) link Layer.
   (1) HTTP
   (2) TCP
   (3) DNS
   (4) Ethernet

Sample Solution:
(a)(b)(a)(d)

7. (ch2, 4pt) We've learned the three design elements in an application layer protocol – the system architecture, socket type, and message formats. Among the 3 system architectures (depicted below) – (a) client-server model without expansion, (b) client-server model with client-end expansion, (c) client-server model with client- and server-end expansion, which one is used by the following protocols:



(1) HTTP
(2) FTP
(3) SMTP
(4) DNS

Sample Solution:

(a)(a)(b)(c)

8. (ch2, 4pt) We've learned the three design elements in an application layer protocol – the system architecture, socket type, and message formats. Among the 2 socket types – (a) TCP and (b) UDP, which one is used by the following protocols:
   (1) HTTP
   (2) FTP
   (3) SMTP
   (4) DNS

Sample Solution:

(a)(a)(a)(b)

9. (ch2, 4pt) We've learned the three design elements in an application layer protocol – the system architecture, socket type, and message formats. Among the 2 message forms (a) readable and (b) non-readable (coded somehow), which one is used by the following protocols:
   (1) HTTP
   (2) FTP
   (3) SMTP
   (4) DNS

Sample Solution:

(a)(a)(a)(a)

10. (ch21, 10pt) Before the data can be encrypted using a master key on a TLS connection, the client needs to send the master key to the server first. To not send the master key without protection, TLS depends on a public key crypto to transmit the master key as a secret. This is achieved by encrypting the master key with the server's public key. This way, only the server (sole holder of the private key) will be able to decrypt the master key.

    RSA is one such public key crypto. Provided two prime numbers p and q, the formula generating the public (e) and private (d) key pair are as follows:

n=p*q
w=(p-1)*(q-1)
Find e and d such that e*d mod w = 1

After key generation, p, q, and d are kept private to the secret receiver. w and e are given to the public so that the sender can use them to encrypt a message using the encryption formula:

secret = (message)^e mod w

Upon receiving the secret, the private key holder can decrypt using a similar formula:

message = (secret)^d mod w

Find the smallest d given the following w and e. Show answer and derivation.
(1) w=60, e=7 (2pt)
(2) w=1584, e=73 (3pt)
(3) w=17316, e=61 (5pt)

Sample Solution:
(1) 43
(2) 217
(3) 6529
Multiple ways to derive the value of smallest d
(Points are credited only when the derivation is presented
Showing derivation but incorrect value may get partial credits
-2 in case your d isn't the smallest)

11. (ch22, 6pt) Consider 2 different connection modes in HTTP – (a) non-persistent + 4 parallel connections and (b) persistent connection without pipelining. Let's assume the transmission time is negligible for simplicity. Let's also assume it takes 40KB to keep a connection on.
(1) Tell the response time in RTT to download a Web page of 1 base html and 4 reference objects using the (a) non-persistent + 4 parallel connections mode. (1pt)
(2) Tell the response time in RTT to download a Web page of 1 base html and 4 reference objects using the (b) persistent without pipelining mode. (1pt)

(3) Tell the peak memory consumption in KB to download a Web page of 1 base html and 4 reference objects using the (a) non-persistent + 4 parallel connections mode. (1pt)
(4) Tell the peak memory consumption in KB to download a Web page of 1 base html and 4 reference objects using the (b) persistent without pipelining mode. (1pt)
(5) Which of the 2 connection modes would you prefer? (1pt) Why? (1pt)

Sample Solution:

(1) Non-persistent + 4 parallel conns: 4RTT
Persistent without pipelining: 6RTT
(2) Non-persistent + 4 parallel conns: 160KB (needing to hold 4 parallel connections)
Persistent without pipelining: 40KB (one connection all the way)
(3) Take your pick and give your justification

12. (ch23-ch24, 10pt) Email is essentially a messaging service, much like Messenger, WhatsApp or LINE. In Email, message receivers are identified by the email address. Through DNS, the sending mail server finds the IP address of the receiving mail server. In WhatsApp, the message receivers are identified by the phone number. DNS, unfortunately, does not resolve phone numbers (to IP addresses of the receiving servers). There must be a lookup service supporting WhatsApp to map the receiver's phone number to the IP address of the receiving server – a databased that stores the (phone#, IP#) mappings and allows querying of the mappings.

(1) Do you think the (phone#, IP#) mappings should be stored centralized in one server? (1pt) Why or why not? (1pt)

Let's try if we can borrow DNS's server-end design and build a 3-level lookup service for WhatsApp – root server, top-level country-code server, and authoritative area-code server. The root server can tell from the country code in the phone# to redirect the mapping query to the top-level country-code server. The top-level country-code server can tell from the area code and redirect the query to the authoritative area-code server.

(2) Would you prefer sending the query recursively or iteratively through the hierarchy? (1pt) Why or why not? (1pt)
(3) Would you prefer caching the top-level country server IP# on the sending server? (1pt) Why or why not? (1pt)
(4) Would you prefer caching the previously requested (phone#, IP#) mappings on the sending server? (1pt) Why or why not? (1pt)

As a user moves from area to area via 4G or WiFi, the IP#s assigned to his/her smartphone can change frequently. Being an "instant" messaging service, the (phone#, IP#) mapping needs to be updated "instantly" as well. Otherwise, the sending server might not be able to find the right IP# to forward the messages instantly.

(5) How frequent would you like the client app on the user's smartphone to send updates of the (phone#, IP#) to the authoritative server? (1pt) And why? (1pt)

Sample Solution:

Take your pick and justify.

13. (ch25, 4pt) Emma joins a BT torrent and connects to 4 peers – Alice, Bob, Cindy, and Danny. The file Emma likes to download is divided into 4 chunks. Each peer tracks its availability of chunks using a vector of 4 bits. When the bit value is 1, the peer holds the chunk on the disk. When the bit value is 0, the chunk is not on the peer's disk. For example, (1, 0, 1, 0) indicates chunk 1 and 3 are available at this particular peer. Provided the vectors of Alice, Bob, Cindy, and Danny below.

Alice:	(1, 1, 1, 1)
Bob:	(1, 0, 1, 0)
Cindy:	(0, 1, 1, 0)
Danny:	(1, 0, 1, 0)

Assume that Alice, Bob, Cindy, and Danny are not interested in the file anymore and stop downloading the remaining chunks. Tell the chunks in the order of Emma's request to complete the file download. (4pt)

Sample Solution:

Chunk 4, chunk 2, chunk 1, and finally chunk 3

14. (PA2, 16pt) Please go on the PA workstation and work under the exam1 directory for this problem set.
    (1) You should see two text files in your home directory – win-test.txt and unix-test.txt. Move the two files to the exam1 directory. (1pt)
    (2) Develop exam1-p14-1.go such that it (1) prompts the user for a filename, (2) reads the file and (3) prints the file size on screen. (2pt)
    (3) Develop exam1-p14-2.go such that it (1) prompts the user for a filename, (2) reads the file and (3) prints the exact file on screen (if there're 10 lines of text in the original file, print the text in 10 lines). (3pt)
    (4) Develop exam1-p14-3.go such that it (1) prompts the user for a filename, (2) reads the file and (3) prints the file in double space on screen (one extra empty line in between the lines in the input file). (2pt)
    (5) Develop exam1-p14-4.go such that it (1) prompts the user for a filename, (2) reads the file and (3) prints the file character by character on the screen, except for two unprintable characters – '\r' and '\n'. When the character is '\r', prints '\' and 'r' on the screen. When the character is '\n', prints '\', 'n', and '\n' on the screen. (6pt)
    (6) Test your exam1-p14-4.go using win-test.txt and unix-test.txt and tell the difference between the output. (2pt)

Sample Solution:

Whatever that works.

Points for (6) will be given only when (5) is completed.

15. (PA3, 16pt) Please go on the PA workstation and work under the exam1 directory for this problem set.
    (1) Develop exam1-p15-1.go such that it connects to the server running on port 11991 and then close the connection. (3pt)
    (2) Develop exam1-p15-2.go such that it connects to the server running on port 11992, sends “hey!\n”, and then closes the connection. (3pt)
    (3) Develop exam1-p15-3.go such that it connects to the server running on port 11993, prompts the user for a short message, sends the message in one line (+’\n’ at the end of the message), and then closes the connection. (4pt)
    (4) Develop exam1-p15-4.go such that it connects to the server running on port 11994, prompts the user for a short message, sends the message in one line (+’\n’ at the end of the message), receives a line of message from the server, prints the line of message from the server on screen, and then closes the connection. (4pt)
    (5) Run exam1-p15-4.go for multiple times sending different short messages and take a guess on what the server on port 11994 is doing. (1pt) What makes you guess so? (1pt)

Sample Solution:

Whatever that works.

Points for (5) will be given only when (4) is completed.