

Name\_\_\_\_\_ Student ID\_\_\_\_\_ Department/Year\_\_\_\_\_

## Midterm Examination

Introduction to Computer Networks

Class#: EE 4020, Session#: 02

Fall 2018

10:20-12:10 Thursday

November 8, 2018

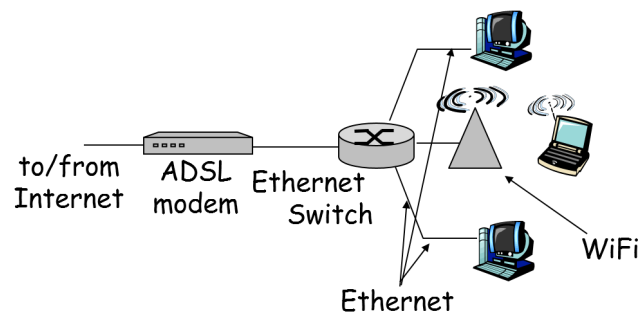
### Prohibited

1. You are not allowed to write down the answers using pencils. Use only black- or blue-inked pens.
2. You are not allowed to read books or any references not on the question sheets.
3. You are not allowed to use calculators or electronic devices in any form.
4. You are not allowed to use extra sheets of papers.
5. You are not allowed to have any oral, visual, gesture exchange about the exam questions or answers during the exam.

### Cautions

1. Check if you get 16 pages (including this title page), 6 questions.
2. Write your **name in Chinese**, student ID, and department/year down on top of the **first** page.
3. There are in total 100 points to earn. You have 100 minutes to answer the questions. Skim through all questions and start from the questions you are more confident with.
4. Use only English to answer the questions. Misspelling and grammar errors will be tolerated, but you want to make sure with those errors your answers will still make sense.
5. If you have any extra-exam emergency or problem regarding the exam questions, raise your hand quietly. The exam administrator will approach you and deal with the problem.

## 1. (Overview)



- (1) Observe an old-fashioned home network such as the above. How many end-system devices are connected to the Internet via a physical wire? And how many via a wireless link? (4%)
- (2) Consider the devices you own personally. Describe each of them and tell us the access network it uses to connect to the Internet. (4%)
- (3) Do you think there will be more devices connecting to the Internet wirelessly in the future? And why do you think so? (4%)
- (4) Do you think the Internet will go entirely wireless in the future? And why or why not? (4%)

### Sample Solution:

- (1) 2, 1
- (2) just tell
- (3) just tell and justify your thoughts
- (4) just tell and justify your thoughts



2. (Overview) A typical highway transportation system allows vehicles to enter the system at the on-ramp, to leave the system at the off-ramp, and to hop from one highway to another at the junctions. If one looks closer, Internet works surprisingly similar. Packets enter the Internet at source nodes. They hop from one link to another at the routers, and exit at the destination.

One interesting design on some highway systems today is the High Occupancy Vehicle (HOV) lanes, which permits only cars of 3 passengers or more to run on. Having fewer cars on the HOV lane, the cars move generally faster. The design is to encourage commuters to car share and increase the efficiency of 'commuter transportation'. Surprisingly again, a packet usually consists of the header part and the data part. In order to be efficient in 'data transportation', one'd also encourage having more data bits on a packet. Now think. How would you implement something similar to the HOV lanes on the Internet so the packets of high data bits can travel faster, therefore encouraging app designers to send bigger packets? Feel free to either use existing functions on the Internet or to invent new mechanisms on your own. (8%)

**Sample Solution:**

Propose a solution and justify.



3. (Overview) Below is the result of tracerouting to www.amazon.com from Polly's office desktop PC. The command is interrupted manually at hop 19.

```
pollyss-iMac:~ pollyoffice$ traceroute www.amazon.com
traceroute to e15316.ci.akamaiedge.net (23.66.45.207), 64 hops max, 52 byte
packets
 1 xdn42o254.ee.ntu.edu.tw (140.112.42.254)  1.895 ms  1.559 ms  3.818 ms
 2 140.112.1.81 (140.112.1.81)  2.095 ms  5.571 ms  2.002 ms
 3 140.112.0.242 (140.112.0.242)  1.413 ms  3.866 ms  1.147 ms
 4 140.112.0.206 (140.112.0.206)  1.727 ms  2.617 ms  2.275 ms
 5 140.112.0.34 (140.112.0.34)  36.857 ms  1.847 ms  1.755 ms
 6 202.169.174.50 (202.169.174.50)  1.930 ms  2.260 ms  2.606 ms
 7 202.169.174.154 (202.169.174.154)  26.885 ms  27.969 ms  30.099 ms
 8 akamai3-lacp-10g.hkix.net (123.255.91.95)  46.697 ms  44.652 ms  42.197
ms
 9 * * *
10 * * *
11 * * *
12 * * *
13 * * *
14 * * *
15 * * *
16 * * *
17 * * *
18 * * *
19 * ^C
pollyss-iMac:~ pollyoffice$
```

- (1) Observe the round-trip delay to 140.112.0.34. The 3 measurements are quite different. Why do you think there is such a difference? (4%)
- (2) Observe the round-trip delay to 140.112.0.242 vs. 140.112.1.81. The delay to the 3<sup>rd</sup> hop seems to be consistently shorter than the 2<sup>nd</sup> hop. Why do you think it's possible the delay to the closer routers being longer? (4%)
- (3) Observe the round-trip delay to 123.255.91.95. Do you think the chance of this router being outside TW is high? Why or why not? (4%)
- (4) Observe the output from hop 9 and on. How would you explain why there are no more measurements coming back? (4%)

**Sample Solution:**

Multiple possible answers. Say one and justify your answer.



4. (Application) We learn in Chapter 2 several application-layer protocols. In particular, HTTP for Web transfer, SMTP for email delivery, IMAP for mailbox access, FTP for file transfer, and DNS for name-IP lookup.

- (1) Which of these 5 protocols run(s) in a client-server model? (3%)
- (2) Which of these 5 protocols send(s) text-based messages? (3%)
- (3) Which of these 5 protocols use(s) out-of-band communication? (3%)
- (4) Which of these 5 protocols cache(s) data to reduce the server loads? (3%)

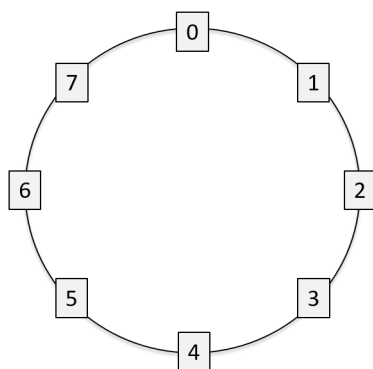
Sample Solution:

- (1) All of them
- (2) All of them
- (3) FTP
- (4) HTTP, DNS. Emails are private and unique. It doesn't make sense to cache. FTP uses mirror sites in case the demand for a particular file is large. Mirrors work like duplicate servers. They are built ahead of time and the data are on the mirror site permanently (not temporarily such as caches)

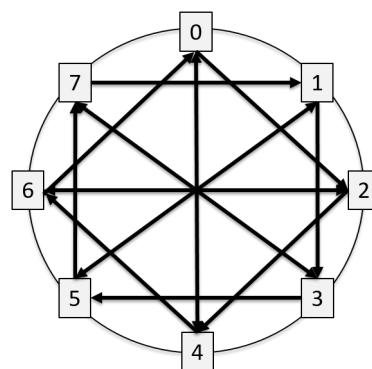


5. (Application) Consider a circular DHT storing the <file-ID, torrent-IPs> mappings for a BitTorrent tracker – PiratesAreHumansToo.tw. The DHT has 8 nodes, whose IDs are 0, 1, 2, ... 7 respectively. All 8 nodes are up and running reliably. The file ID ranges from 0 to arbitrarily large. The hash function is defined as simple as:  $\text{key} = (\text{file-ID} \bmod 8)$ , meaning a <file-ID, torrent-IPs> mapping is stored in the node whose ID = key. Upon receiving a search query for a file (file-ID), each node operates based on a simple algorithm: check if  $\text{node-ID} == \text{key}$ . If so, done and search the local <file-ID, torrent-IPs> entries for the matching file. Otherwise, relay the search query to the successor or the shortcut node that is the closest but not exceeding the key.

The DHTs below depicts (a) the version without any shortcuts (i.e., each node tracks only the successor), (b) the version with 2 shortcuts each to the nodes that are 1/2- and 1/4-circle away



(a) No shortcut



(b) 2 shortcuts 1/2 and 1/4 circle away

- (1) Suppose the file IDs of the queries are uniformly random (equal chance of each file being requested). What is the average number of relays for DHT (a) and (b)? (8%)
- (2) Continue from (1) and consider a DHT(a) of  $2^n$  nodes, where  $n$  is a positive integer. What is the average number of relays for DHT (a)? (8%)
- (3) Continue from (1) and consider a DHT(b) of  $2^n$  nodes, where  $n$  is a positive integer. What is the average number of relays for DHT (a)? (10%)

Sample Solution:

- (1) DHT(a): 0, 1, 2, 3, 4, 5, 6, 7 queries per search for ID=0, 1, 2, 3, ..., 7 (after mod 8)  
 DHT(b): 0, 1, 1, 2, 1, 2, 2, 3 queries per search for ID=0, 1, 2, 3, ..., 7(after mod 8)  
 The average number of relays for DHT (a) and (b) are 3.5 and 1.5 respectively.

- (2) The average number of relays for DHT (a) with  $2^n$  nodes is:  $(2^n - 1)/2$

Total number of queries to search for file ID=0, 1, ...  $2^n - 1 = 0 + 1 + 2 + \dots + 2^n - 1$

Average number of queries per file =  $(0 + 1 + 2 + \dots + 2^n - 1) / 2^n = (2^n - 1) / 2$

- (3) The average number of relays for DHT (a) with  $2^n$  nodes is:  $n/2$

Let total number of relays to search for file ID=0, 1, ... all the way to  $2^n - 1 = f(2^n)$

$$f(2^n) = f(2^{n-1}) + f(2^{n-1}) + 2^{n-1} \quad \# \text{ of relays for the later half} = \text{that of the first half} + 1$$

$$= 2 f(2^{n-1}) + 2^{n-1}$$

$$= 2 [f(2^{n-2}) + f(2^{n-2}) + 2^{n-2}] + 2^{n-1} \quad \# \text{ expand } f(2^{n-1}) \text{ to } f(2^{n-2}) + f(2^{n-2}) + 2^{n-2}$$

$$= 2^2 f(2^{n-2}) + 2^{n-1} + 2^{n-1} = 2^2 f(2^{n-2}) + 2 * 2^{n-1}$$

$$= 2^2 [f(2^{n-3}) + f(2^{n-3}) + 2^{n-2}] + 2 * 2^{n-1} \quad \# \text{ expand } f(2^{n-2}) \text{ to } f(2^{n-3}) + f(2^{n-3}) + 2^{n-2}$$

$$= 2^3 f(2^{n-3}) + 2^{n-1} + 2 * 2^{n-1} = 2^3 f(2^{n-3}) + 3 * 2^{n-1}$$

...

$$= 2^k f(2^{n-k}) + k * 2^{n-1} \quad \# \text{ general form}$$

...

$$= 2^{n-1} f(2^1) + (n-1) * 2^{n-1} \quad \# \text{ when } k=n-1$$

$$= n * 2^{n-1}$$

Average number of queries per file =  $n * 2^{n-1} / 2^n = n/2$

\*\*\*If you are thinking just 2 shortcuts to  $\frac{1}{2}$ - and  $\frac{1}{4}$ -circle away nodes, it'd be  $2^{n-3} + 1/2$ . The approach to count the # of relays is generally to add the # of relays for the nodes in the 1<sup>st</sup> quarter, 2<sup>nd</sup> quarter, and then 2<sup>nd</sup> half.

$$\# \text{relays for the 1}^{\text{st}} \text{ quarter} = 0 + 1 + 2 + \dots + 2^{n-2} - 1 = 2^{n-2}(2^{n-2} - 1) / 2 = 2^{2n-5} - 2^{n-3}$$

$$\# \text{relays for the 2}^{\text{nd}} \text{ quarter} = 1 + 2 + 3 + \dots + 2^{n-2} = 2^{n-2}(2^{n-2} + 1) / 2 = 2^{2n-5} + 2^{n-3}$$

$$\# \text{relays for the 1}^{\text{st}} \text{ half} = 2^{2n-5} - 2^{n-3} + 2^{2n-5} + 2^{n-3} = 2^{2n-4}$$

$$\# \text{relays for the 2}^{\text{nd}} \text{ half} = \# \text{relays for the 1}^{\text{st}} \text{ half} + 2^{n-1} = 2^{2n-4} + 2^{n-1}$$

$$\text{Sum of all} = 2^{2n-4} + 2^{2n-4} + 2^{n-1} = 2^{2n-3} + 2^{n-1}$$

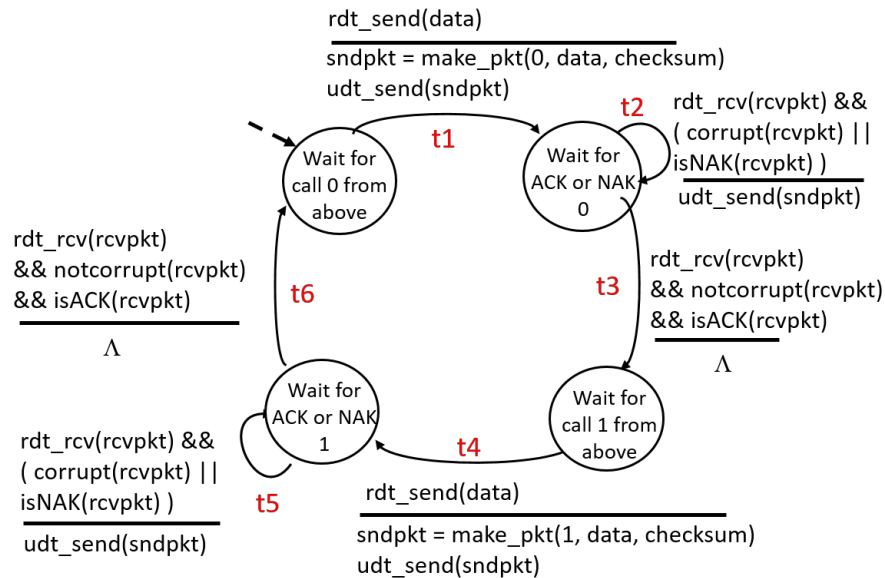
$$\text{Average} = \text{sum} / 2^n = 2^{n-3} + 2^{-1}$$



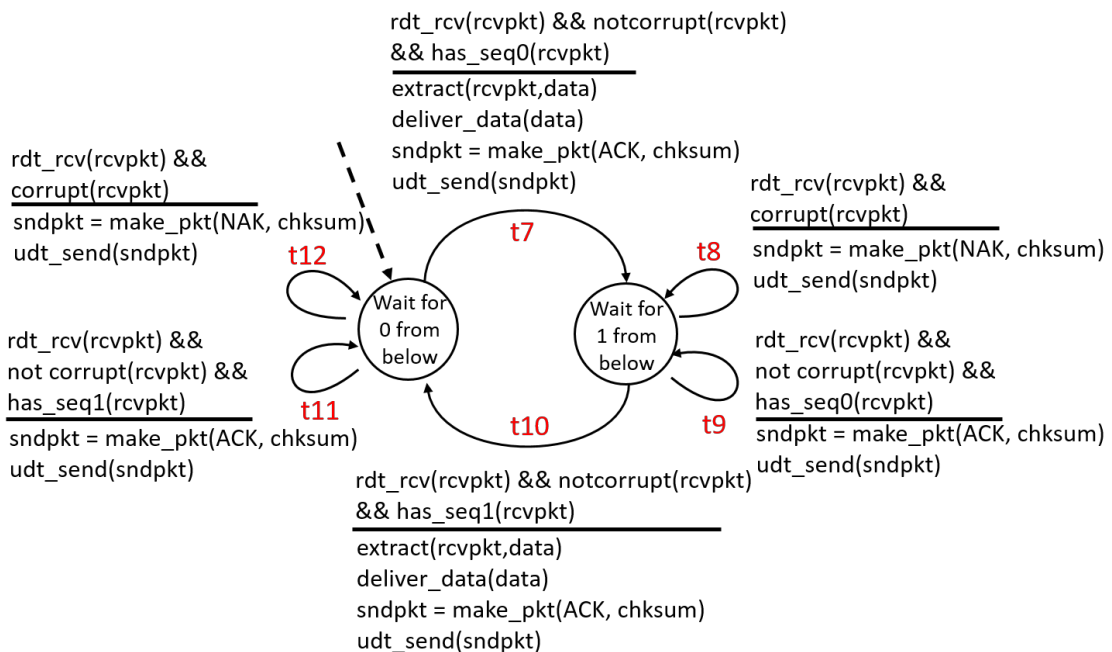


6. (Transport) Provided below are the FSMs of rdt 2.1 sender and receiver. Indicate the order of the transitions (in terms of t1, t2, ..., t12) taking place for the following scenarios.

rdt 2.1 sender:



rdt 2.1 receiver:



- (1) Both sender and receiver start from the initial state. The sender gets one call from above and there is no bit error. (4%)
- (2) Both sender and receiver start from the initial state. The sender gets one call from above and there is a bit error in the data packet. (4%).
- (3) Both sender and receiver start from the initial state. The sender gets one call from above and there is a bit error in the ACK packet coming back. (4%)
- (4) Both sender and receiver start from the initial state. The sender gets a call from above and the ACK packet does not come back at all. (4%)
- (5) Extend rdt 2.1 such that it handles packet losses. (6%)

Sample Solution:

- (1) t1, t7, t3
- (2) t1, t12, t2, t7, t3
- (3) t1, t7, t2, t9, t3
- (4) t1, t7, and then the sender hangs. None of the t2 and t3 defines such an event. rdt 2.1 does not handle cases where there is any packet loss.
- (5) It can be the rdt 3.0 as illustrated in the lecture/textbook or any rdt that would work.



