**Kebi Hong**

**CS 455**

**HW #1**

1. 1). **Network Protocols**: it is similar to a human protocol, except that the entities exchanging messages and taking actions are hardware or software components of some device (for example, computer, smartphone, tablet, router, or other network-capable device). All activity in the Internet that involves two or more communicating remote entities is governed by a protocol. In the case of layered network architecture, there are protocols between each layers and when two layers communicate with each other, they need to follow those protocol, for example, adding a header to the data.

2). **Network Service**: It is the end-to-end transport of packets. When we want to send a message to other people, the computer will encrypt the message when it goes down the layered structure. The encrypted message is sent through the network service, and then the receiver will decrypt the data and get the message.

3). the **service interface** is a connection between different layers and tells the computer how to encrypt the data so that the data can go through layers. The **implementation of service** is not only tell the rule of encrypting but also encrypt the message itself.



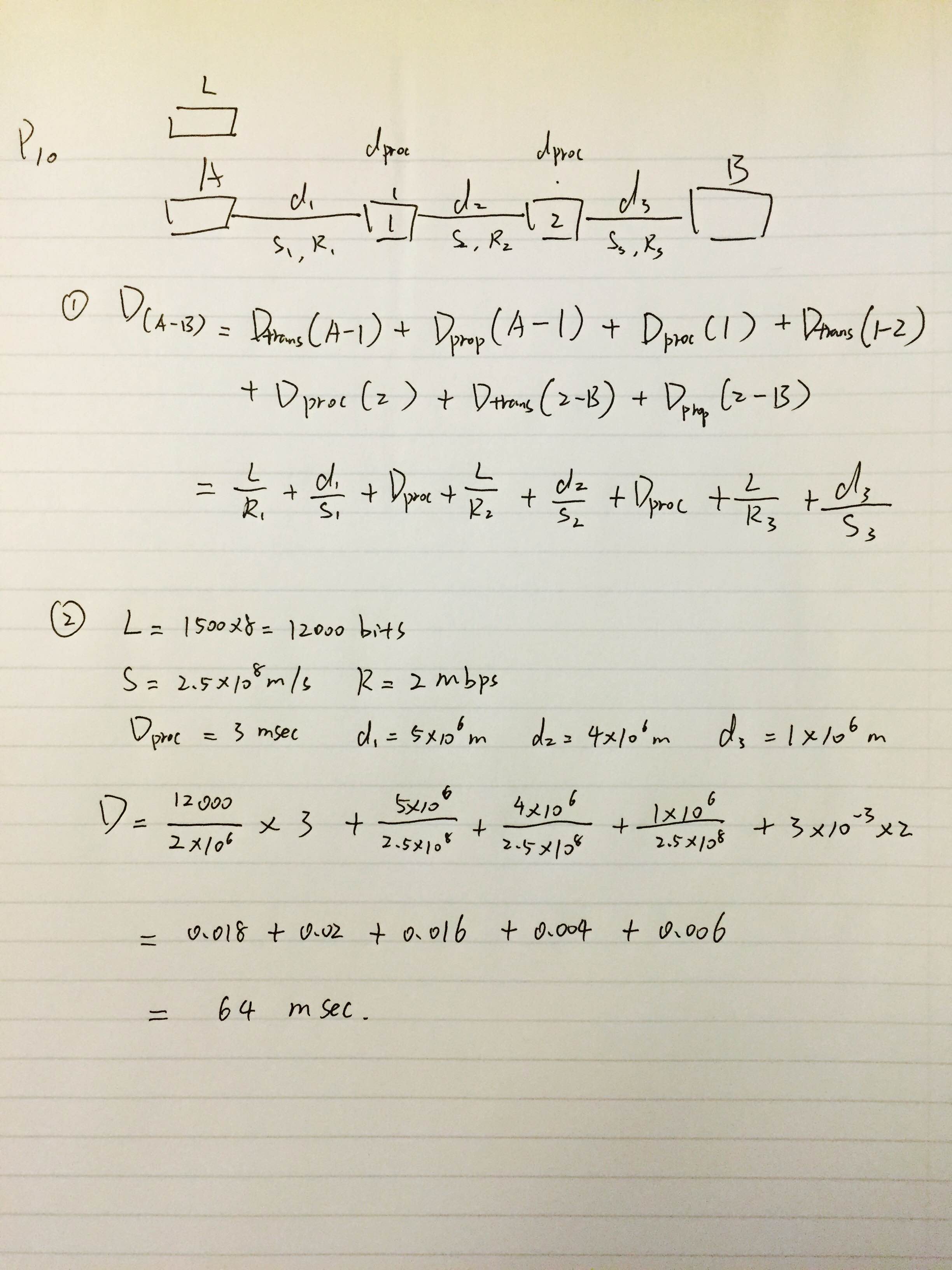
**First Data(at morning)**

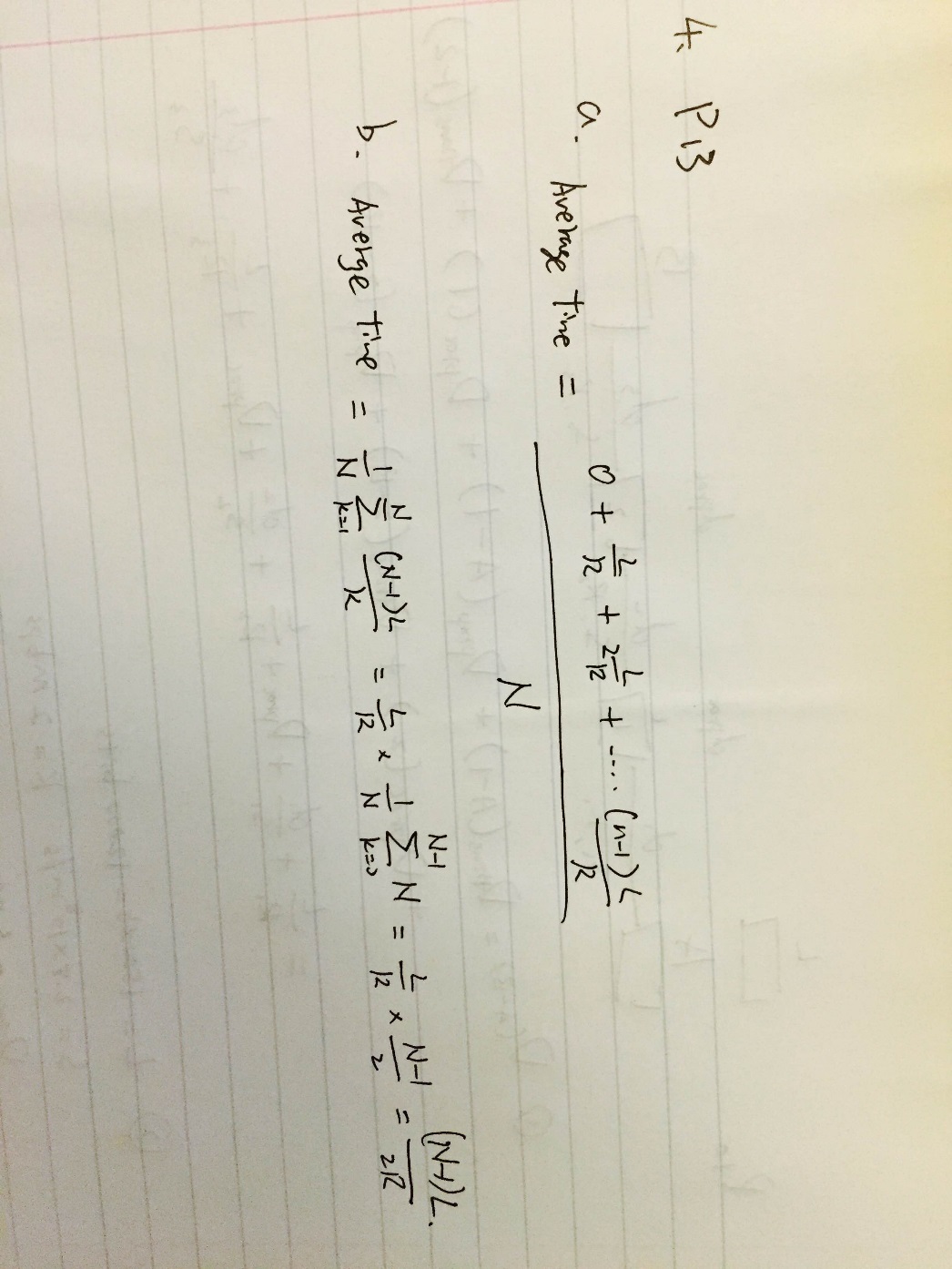
1. PING cs.stanford.edu (171.64.64.64): 56 data bytes
2. 64 bytes from 171.64.64.64: icmp\_seq=0 ttl=51 time=89.961 ms
3. 64 bytes from 171.64.64.64: icmp\_seq=1 ttl=51 time=91.946 ms
4. 64 bytes from 171.64.64.64: icmp\_seq=2 ttl=51 time=92.073 ms
5. 64 bytes from 171.64.64.64: icmp\_seq=3 ttl=51 time=92.377 ms
6. 64 bytes from 171.64.64.64: icmp\_seq=4 ttl=51 time=92.254 ms
7. 64 bytes from 171.64.64.64: icmp\_seq=5 ttl=51 time=92.024 ms
8. 64 bytes from 171.64.64.64: icmp\_seq=6 ttl=51 time=92.508 ms
9. 64 bytes from 171.64.64.64: icmp\_seq=7 ttl=51 time=92.160 ms
10. 64 bytes from 171.64.64.64: icmp\_seq=8 ttl=51 time=92.918 ms
11. 64 bytes from 171.64.64.64: icmp\_seq=9 ttl=51 time=92.773 ms
12. 64 bytes from 171.64.64.64: icmp\_seq=10 ttl=51 time=91.967 ms
13. 64 bytes from 171.64.64.64: icmp\_seq=11 ttl=51 time=89.502 ms
14. 64 bytes from 171.64.64.64: icmp\_seq=12 ttl=51 time=92.346 ms
15. 64 bytes from 171.64.64.64: icmp\_seq=13 ttl=51 time=92.272 ms
16. 64 bytes from 171.64.64.64: icmp\_seq=14 ttl=51 time=92.175 ms
17. 64 bytes from 171.64.64.64: icmp\_seq=15 ttl=51 time=92.138 ms

**Second data (at night)**

1. Qians-MacBook-Pro:~ qianqiao$ ping [cs.stanford.edu](http://cs.stanford.edu/)
2. PING [cs.stanford.edu](http://cs.stanford.edu/) (171.64.64.64): 56 data bytes
3. 64 bytes from [171.64.64.64](http://171.64.64.64/): icmp\_seq=0 ttl=49 time=99.867 ms
4. 64 bytes from [171.64.64.64](http://171.64.64.64/): icmp\_seq=1 ttl=49 time=94.728 ms
5. 64 bytes from [171.64.64.64](http://171.64.64.64/): icmp\_seq=2 ttl=49 time=103.330 ms
6. 64 bytes from [171.64.64.64](http://171.64.64.64/): icmp\_seq=3 ttl=49 time=90.616 ms
7. 64 bytes from [171.64.64.64](http://171.64.64.64/): icmp\_seq=4 ttl=49 time=91.816 ms
8. 64 bytes from [171.64.64.64](http://171.64.64.64/): icmp\_seq=5 ttl=49 time=90.445 ms
9. 64 bytes from [171.64.64.64](http://171.64.64.64/): icmp\_seq=6 ttl=49 time=93.020 ms
10. 64 bytes from [171.64.64.64](http://171.64.64.64/): icmp\_seq=7 ttl=49 time=91.211 ms
11. 64 bytes from [171.64.64.64](http://171.64.64.64/): icmp\_seq=8 ttl=49 time=88.925 ms
12. 64 bytes from [171.64.64.64](http://171.64.64.64/): icmp\_seq=9 ttl=49 time=91.434 ms
13. 64 bytes from [171.64.64.64](http://171.64.64.64/): icmp\_seq=10 ttl=49 time=93.009 ms
14. 64 bytes from [171.64.64.64](http://171.64.64.64/): icmp\_seq=11 ttl=49 time=115.484 ms
15. 64 bytes from [171.64.64.64](http://171.64.64.64/): icmp\_seq=12 ttl=49 time=91.814 ms
16. 64 bytes from [171.64.64.64](http://171.64.64.64/): icmp\_seq=13 ttl=49 time=105.027 ms
17. 64 bytes from [171.64.64.64](http://171.64.64.64/): icmp\_seq=14 ttl=49 time=92.782 ms

By observation, we can see that RTT is longer at evening, because there are more people using it at the same time. Therefore, RTT depends on the degree of crowding of the internet.

**3.** 

**4.** 

**5.**

**a).**

Three round trip delay between source and the router is

D1 = 1.05msec, D2 = 0.58 mesc , D3 = 0.35 msec

Mean Round Trip Delay = (1.05+0.58+0.35)/3   = 0.66msec

=

= 0.3223 msec

**b).**

The number of routes between source and destination is 8. At some certain time in one day, the path may be changed with respect to hours.

**c).**

**d).**

**6.** Distance = 2\*10^7 m R = 2\*10^6 bit/sec

propagation speed = 2.5\*10^8 m/sec

a). Propagation time = 2\*10^7 / 2.5\*10^8 = 0.08 sec

Bandwidth-delay product = R\* Propagation-time

= 2\*10^6 \*0.08 = 1.6 \* 10^5 bits

b). The bandwidth-delay product shows that the maximum number of bits can be hold in the link at the same time. The product is 1.6\*10^5 bits, which is much smaller than the data size 8\*10^5 bits. Therefore, the maximum number of bits that will be in the link in 1.6\*10^5 bits.

c). The maximum number of bits on the transmission line is equal the to product of bandwidth and the delay

d). Length of 1 bit on the transmission line = S/R = 2.5\*10^8/2\*10^6 = 125m/bit. It is longer than a football field

e). width of a bit = propagation speed / transmission rate = S/R

**7.**

a). according to the question, there is only transmission time between each switch, and each transmission delay would be the same.

Time = packet size / transmission rate

= (8 \* 10^6 / 2\* 10^6) \* 3 = 12 sec

b). Time to move first packet = 10,000 / 2\* 10^6 = 0.005 sec

The second packet has to wait 0.005 sec for the first packet to arrive the first switch. Then we start to send out the second packet and it takes 0.005 sec to arrive the first switch.

Therefore, the total time for the second packet to arrive the first switch will be 0.005 + 0.005 = 0.01 sec

c). It takes 0.015 sec for the first packet to fully arrive the destination, and after that, we can receive one packet every 0.005 sec.

Total transmission time = 0.015 + 799 \* 0.005 = 4.01 sec

This is approximately 3 times faster than time used without segmentation.

d). In addition to reducing delay, segmentation can help recover the message when some packets are lost. We can just resend those lost packets, which saves a lot of time.

e). the receiver need to decrypt the message and reassemble packets to get the real message.

**8.** Packet Number = Total byte sent / data size = 1\* 10^6 / data size

Total overhead = 100 Byte \* Packet Number

= 100 \* (1\* 10^6 / data size)

Total size = total overhead + data size

Packet size 1000: 1\* 10^8 / 1000 + 1000 = 1.01\*10^5

Packet size 5000: 1\* 10^8 / 5000 + 5000 = 2.5\*10^4

Packet size 10000: 1\* 10^8 / 10000 + 10000 = 2\*10^4

Packet size 20000: 1\* 10^8 / 20000 + 20000 = 2.5\*10^4

Optimal packet size is 10000 bytes