CSEE 4119 Fall 2020
(ungraded) Homework 4

Please refer to the course information in the syllabus for further information and policies regarding late homeworks and collaboration, and the academic honesty policies. Make sure that you observe these policies. Uploading course materials, including questions/answers from this homework, to sites such as CourseHero, Chegg or Github is academic misconduct at Columbia (see pg 10).

Unless otherwise specified, we use 10^3 for "K", 10^6 for "M", and so on to simplify calculations. Please do the same in your answers.

1. IPv4 Datagram, addressing and IPv6 [20 points, parts a-g]

a. [3 points] Suppose you want to send a 2800-byte IPv4 datagram into a link with an MTU of 700 bytes. How many fragments are generated? Please explain your answer.
Answer:
The length of the header is 20, so we can't simply divide 2800 by 700. The actual body size of 2800-byte IPv4 datagram is 2800 - 20 = 2780 bytes, and the actual body size of 900-byte MTU is 700 - 20 = 680 bytes, since 2780/680 = 4.088. 4 fragments are not enough, we need 5 fragments.

+2 Fragments = ceiling[(2800-20)/(700-20)] = 5
+1 Explanation: mention header length 20

b. [3 points]For each of the fragment you got in 1.a, please calculate the fragment flag and offset field inside the datagram.
The fragment flag is 1 except the last one, so the flag for the first four is 1, and the flag of the last one is 0.
The offset of the first one is 0 since it is the beginning one, the following one's offset is 680/8 =85(680 for the MTU body), followed by 170, 255 and 340.

Answer:
+1 for 4 flags, +0.5 for each of 4 offsets
Flags: 1, 1, 1, 1, 0
Offsets: 0, 85, 170, 255, 340

c. [8 points] Given a subnet with IPv4 address whose prefix is 127.0.8.0/23. Design address assignment for each subnet of incoming users. You are not allowed to waste addresses. You need to provide the smallest number of addresses and lowest unassigned addresses for each subnet.

i. [2 points] Design address assignment for the first subnet with 12 users. (Please explain by describing how many addresses are in the subnet total, and how many are free for allocation)
Answer:
Total network prefix: 127.0.8/23
01111111 00000000 0000100X XXXXXXXX
Subnet 1: 16 interfaces
01111111 00000000 00001000 0000XXXX
14 addresses are free for allocation (The first one and last one is reserved by default)
127.0.8.0/28
+1 for correct answer
+1 for explanation

ii. [2 points] Design address assignment for the second subnet with 61 users. (Please explain by describing how many addresses are in the subnet total, and how many are free for allocation)
Answer:
Subnet 2: 64 interfaces
01111111 00000000 00001000 01XXXXXX
62 addresses are free for allocation (The first one and the last one is reserved by default)
127.0.8.64/26
+1 for correct answer
+1 for explanation

iii. [2 points] Design address assignment for the third subnet with 61 users. (Please explain by describing how many addresses are in the subnet total, and how many are free for allocation)
Answer:
Subnet 3: 64 interfaces
01111111 00000000 00001000 10XXXXXX
62 addresses are free for allocation (The first one and last one is reserved by default)
127.0.8.128/26
+1 for correct answer
+1 for explanation

iv. [2 points] Design address assignment for the last subnet with 64 users. (Please explain by describing how many addresses are in the subnet total, and how many are free for allocation)
Answer:
Subnet 4: 128 interfaces considering the first and last address in subnet.
01111111 00000000 00001001 0XXXXXXX
126 addresses are free for allocation (The first one and last one is reserved by default)

127.0.9.0/25
+1 for correct answer
+1 for explanation

d. Please list at least 3 changes from IPv4 to IPv6 and explain the reason for the 3 changes. [6 points]
Answer:
- The length of source and destination address is changed.
  Reason: use longer addresses to get more addresses and satisfy the requirement of increasing demands.
- IPv6 header is fixed-length and IPv4 is variable-length.
  Reason: To streamline IPv6 header processing on the router.
- IPv6 removed options field in IPv4.
  Reason: The header can stay fixed length in order to streamline header processing.
- There is no checksum field in IPv6 header.
  Reason: IPv4 had TTL field that gets decremented at each intermediate router, each router has to recompute the checksum field. Removing the checksum field in IPv6 makes header processing less costly.
- IPv6 does not allow fragmentation and reassembly by the routers.
  Reason: Fragmentation and reassembly are time-consuming and removing it speeds up IP packet forwarding in the network

+1 for each reason and +1 for corresponding explanation

2. Switching and Scheduling [20 points, parts a-g]

a. Suppose there are two packets arriving two different input ports of a router at exactly the same time. Assume there are no other packets anywhere in the router.

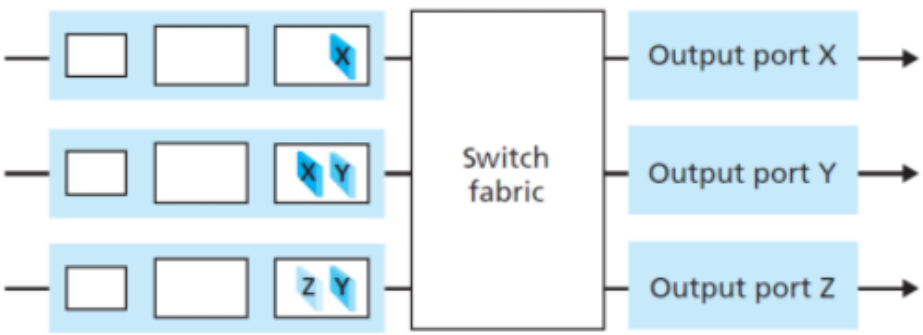
i. [2 points] If these two packets are to be forwarded to two different output ports. Is it possible to forward them at the same time when the fabric uses a *shared bus*? Explain why.
No. Only one packet can be transmitted at a time over a shared bus.
+1 correct answer
+1 correct explanation

ii. [2 points] Similar to question a, two packets are to be forwarded to different output ports. But this time, the fabric uses a *crossbar*, is it possible to forward them at the same time? Explain why.
Yes. As long as these two packets use different input buses and different output buses, they can be forwarded in parel.
+1 correct answer
+1 mentions when packets use different input buses.

iii. [2 points] If these two packets are to be forwarded to the same output port, still with a *crossbar*. Is it possible to forward them at the same time? Explain why.

No. It is impossible to forward two packets to the same output ports, no matter what methods are used by switching fabric.
+1 correct answer
+1 correct explanation

b. Consider the switch shown below. Suppose that all datagrams have the same fixed length, and the switch operates in a slotted, synchronous manner. In one time slot, a datagram can be transferred from an input to an output port. The letter labelled on the datagram means the output port it should be forwarded to.



i. [3 points] If the switch fabric uses a shared bus, what is the number of time slots needed to transfer all the packets from input ports to their output ports? Please figure out the packets forwarded in each time slot.
Five time slots. It takes one time slot for each packets.
+1 correct answer
+2 correct explanation

ii. [3 points] If the switch fabric uses a crossbar, what is the number of time slots needed? Please figure out the packets forwarded in each time slot.

Though there are several possibilities, the only three time slots are needed for all the cases.
There are two possibilities, either of them is correct.

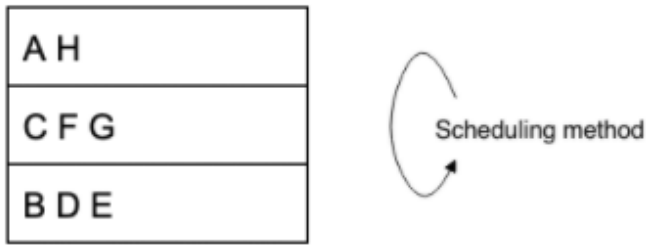
1) In the first time slot, send X in top input queue, Y in middle input queue
The second time slot, send X in the middle, Y in the bottom.
The third time slot, send Z in the bottom.

2) In the first time slot, send X in top input queue, Y in bottom input queue
The second time slot, send Y in the middle, send Z in bottom
The third time slot, X in the middle

+1 correct answer
+2 correct explanation

c. Now let's delve deep in packet scheduling methods. Consider the input queues displayed below. All the packets are classified into three classes. A and H are in class 1,

C, F, G belong to class 2, B, D, E belong to class 3. The arrival time of all packets and the length of each packet are shown in the table below. Suppose it will take 1 unit of time to transmit a packet for each unit of length. Suppose that a packet's transmission will not be interrupted to begin transmission of another packet (i.e. non-preemptive).



Packet	Arrival Time	Length
A	0	8
B	5	6
C	5	10
D	8	9
E	8	8
F	10	6
G	11	10
H	20	8

i. [4 points] If Round Robin is used, please figure out the transmission order of packets, and give your explanation. (Suppose Round Robin starts from class 1, and iterate in the order of 1->2->3->1...)

A C B H F D G E

In round-robin, we take one packet in turn from each non-empty queue at the next output time.

+2 correct answer (+1 if there is a minor mistake)

+2 correct explanation (+1 if there is a minor mistake)

ii. [4 points] If Weighted Fair Queuing is used, we set the weight of both class 1 and class 2 to be 1, and the weight of class 3 to be 2, then what is the order of the transmission order of packets? Please write down in detail how you get your answer. Notice Weighted Fair Queueing is not fully discussed in the textbook, so

you can learn how it works by yourself. (Hint: In weighted fair queuing, packets are selected to be forwarded according to the virtual departure time).
ABDCEHFG or BADCEHFG

+2 correct answer (+1 only for minor calculation error)
+1 only correct answer for simplified equation

+2 correct explanation
+1 only correct explanation for simplified equation

You can find the answer using virtual finish time/ bit by bit round robin/GPS-fluid-model. You can refer to the table and explanation below to figure out how it really works.

Here are some useful links:
https://en.wikipedia.org/wiki/Weighted_fair_queueing
https://intronetworks.cs.luc.edu/current/html/queuing.html

Packet	Arrival Time	Length	Finish time	Output order
A	0	8	17	1
B	5	6	17	2
C	5	10	44	4
D	8	9	34	3
E	8	8	50	5
F	10	6	55	7
G	11	10	65	8
H	20	8	51	6

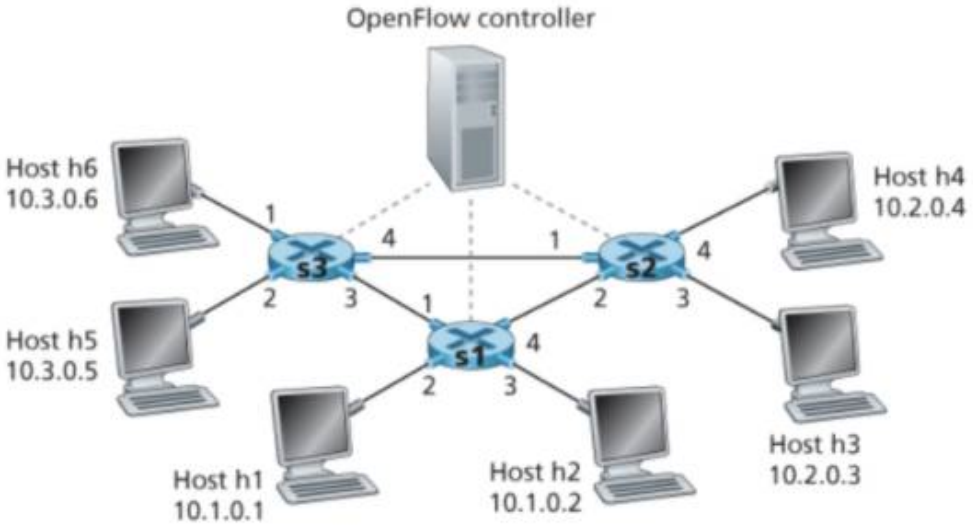
Here is an overview of how the fluid model works for the first 20 units of time:

at 0-5, A will take all the bandwidth
at 5-17, A will take ¼, C will take ¼, B will take ½, so for A, it will need 3/ ¼ = 12 units of time to finish transmitting. As a result, the finish time of A is 5+12=17. For B 6 / ½ =12 units of time necessary as well, so B will also finish transmitting at 17.
at 17-20, D will take ⅔ and C will take ⅓ of the bandwidth.

at 20 H will arrive and start transmitting, making the new bandwidth allocation ¼ for H, ¼ for C and ½ for D.

Packet	Arrival Time	Length	Finish time	Output order
A	0	8	8	1
B	5	6 → 3	8	2
C	5	10	15	4
D	8	9 → 4.5	12.5	3
E	8	8 → 4	16.5	5
F	10	6	21	6
G	11	10	31	8
H	20	8	28	7

3. Generalized Forwarding and OpenFlow [parts a-e, 9 points]



Consider the OpenFlow match-plus-action network shown above (figure 4.30 from textbook chapter 4.4.3). There are three packet switches marked as s1-s3, six hosts marked as h1-h6 and one OpenFlow controller.

Note: For all questions that ask you to design flow table, your rules should apply exactly as described in the goal and should not impact any other traffic that is feasible given the hosts nd switches in the figure (e.g., if the rule says to do something with traffic from h4, it should not

impact traffic from h2). However, you need not consider what happens if additional hosts join (e.g., the rules can impact traffic to/from hosts/IP addresses not shown in the figure).

a.[2 points] Design S2's flow table so that it achieves the following goals:

(1) Datagrams arrived on port 4 and destined to h5 or h6 should be forwarded to port 1.

(2) Datagrams arrived on port 3 and destined to h1 or h2 should be forwarded to port 2.

S2 Flow Table	
Match	**Action**
……	……

Answer:

S2 Flow Table	
Match	**Action**
Ingress Port = 4; IP Dst = 10.3.*.*	Forward (1)
Ingress Port = 3; IP Dst = 10.1.*.*	Forward (2)

B. [1 point] Add one additional rule to S3's flow table so that packets from h5 and h6 destined to h1 or h2 are to be forwarded to S2 and then to S1. Design S3's flow table. (You do not need to take care of S2's flow table for this question)

Answer:

S3 Flow Table	
Match	**Action**
IP Src = 10.3.*.*; IP Dst = 10.1.*.*	Forward (4)

+1 For achieving each design goal

+0.5 If minor mistakes in entry

c.[1 point] Based on the design in (a), what will happen if host 4 sends a datagram to host 3?

Answer:

When S2 receives this datagram, it cannot find a matching entry in the flow table. It can either drop it or send it to the remote controller for processing.
+1 For correct answer ; Answered at least one of these two actions will get full credit; Can also talk about the table-miss flow entry;

[3 points] OpenFlow's match-plus-action is a more general pradiam in which matches can be made over multiple header fields associated with different protocols at different protocol stack layers. Suppose we want to use S2 as a firewall so that:

1. Packets coming from h1 and h2 can be delivered to h3 or h4 (if they don't violate other rules) while packets coming from h5 and h6 will be blocked.
2. Only packets sent to port 80 will be delivered.

Note: You don't need to write down flow table entries for other behaviors of S2.

S2 Flow Table	
Match	**Action**
……	……

Answer:

S2 Flow Table	
Match	**Action**
IP Src = 10.1.*.*; IP Dst = 10.2.0.3; Dst Port = 80; IP Src = 10.1.*.*; IP Dst = 10.2.0.4; Dst Port = 80;	Forward (3) Forward (4)

+1 For including correct IP Src and IP Dst
+1 For including correct Dst Port
+1 For correct actions

[2 points] Suppose due to a business concern, the network administrator wants to make sure that all packets at S2 destined to 10.3.*.* are forwarded to port 2 so installs one entry into S2's flow table:

S2 Flow Table	
Match	**Action**
IP Dst = 10.3.*.*	Forward (2)

Additionally, h3, which is directly connected to S2, prefers to route all of its packets via S3. Therefore, the administrator installs another entry into S2's flow table:

S2 Flow Table	
Match	**Action**
IP Dst = 10.3.*.*	Forward (2)
IP Src = 10.2.0.3	Forward (1)

Now if h3 sends a packet to h5 or h6, this packet will match two entries in S2's flow table. Suppose h3 is absolutely vital, so the administrator will try to meet h3's demand regardless of business concerns. If we can't modify the Match condition in S2's flow table, how can we help accomplish this goal at S2? Hint: There are other components in OpenFlow's flow table entry.

Answer:

We can make use of flow table entry's priority attribute. By assigning higher priority to the second entry, we can make sure h3's demand is always met.

+2 For correct answer