Jackson Chen (jc4697)
Computer Networks HW4 (Questions 1 and 2)

1. IPv4 Datagram, addressing and IPv6

a. When the router (immediately before the 700 byte MTU link) receives the 2800 byte datagram, it performs IP fragmentation. When the destination host receives the fragments, it reassembles them to the original datagram. During fragmentation, the router splits the datagram is split into **five** smaller datagrams. This is because each fragment can only hold up to 680 bytes of data field data (the header takes up 20 bytes, and the maximum length of the datagram is 700). A total of 2780 bytes need to be sent (since the 2800 byte datagram contains a 20 byte header). Dividing 2780 by 680 requires 5 datagrams to fully encapsulate all of the data.
b. The fragment flag and offset fields for the five datagrams are as follows:
    (1) Fragflag = 1, offset = 0
    (2) Fragflag = 1, offset = 85
    (3) Fragflag = 1, offset = 170
    (4) Fragflag = 1, offset = 255
    (5) Fragflag = 0, offset = 340
c. 01111111 00000000 00001000 00000000 (the underlined are hosts within the subnet)
    i. **127.0.8.0/28**: A total of 16 addresses are in the subnet total, and a total of 14 are free for allocation (the smallest 127.0.8.0 and largest 127.0.8.15 in the subnet are reserved for the network address of the subnet and the broadcast address, respectively[1])
    ii. **127.0.8.16/26**: A total of 64 addresses are in the subnet total, and a total of 62 are free for allocation (the smallest and largest are allocated for the same reasons as above)
    iii. **127.0.8.80/26**: A total of 64 addresses are in the subnet total, and a total of 62 are free for allocation (the smallest and largest are allocated for the same reasons as above)
    iv. **127.0.8.144/25**: A total of 128 addresses are in the subnet total, and a total of 126 addresses are free for allocation (the smallest and largest are allocated for the same reasons as above)
d. Changes:
    - Increases the size of the IP address from 32 to 128 bits: This is because the 32 bit address space was soon to be completely allocated
    - Fixed length 40 byte header: Allows for faster processing of the IP datagram by the router

[1] https://ec2.freesoft.org/CIE/Course/Subnet/103.htm

- No fragmentation allowed (it must be done at the source/destination otherwise a too large packet will be dropped): This speeds up IP forwarding within the network as fragmentation and reassembly are time consuming operations

2. Switching and Scheduling

a.

i. No it is not possible to forward these two packets at the same time using a shared bus. This is because only one packet can cross the bus at a time even if the output ports are different.

ii. Yes it is possible to forward them at the same time since the two packets use different input and output busses (different input ports and different output ports).

iii. No it's not possible to forward them at the same time, since only one packet can be sent over any given bus at a time, and they are both competing to use the same output bus (same output port).

b.

i. A total of **5** time slots will be used. Assume that switch fabric takes packets from the upper most ports first before going to the later ports

Time Slot	Packets
1	X in upper port
2	Y in middle port
3	X in middle port
4	Y in lower port
5	Z in lower port

ii. A total of **3** time slots will be used. Assume that switch fabric resolves conflicts by taking packets from the upper most ports first before going to the later ports

Time Slot	Packets
1	X in upper port, Y in middle port
2	X in middle port, Y in lower port
3	Z in lower port

c. A

i. Overall transmission order: **A, C, B, H, F, D, G, E**

Time for start of	Queue	Packet chosen to be

transmission		transmitted
0	1: A 2: 3:	A (class 1)
8	1: 2: C 3: B D E	C (class 2)
18	1: 2: F G 3: B D E	B (class 3)
24	1: H 2: F G 3: D E	H (class 1)
	1: 2: F G 3: D E	F (class 2)
	1: 2: G 3: D E	D (class 3)
	1: 2: G 3: E	G (class 2)
	1: 2: 3: E	E (class 3)

ii. I am implementing the weight by dividing the length of the packet in order to boost its virtual departure time: finish(i) = max(arrive(i), finish(i-1)) + length/weight)

Overall transmission order: **A, B, D, C, E, F, H, G**

Packet	Arrival Time	Length	Virtual Finish	Output Order
A	0	8	8	1
B	5	3 (halved)	8	2
C	5	10	15	4

D	8	4.5 (halved)	12.5	3
E	8	4 (halved)	16.5	5
F	10	6	21	6
G	11	10	31	8
H	20	8	28	7