Barret Jackson
66122573
October 23, 2020

<u>COSC 328 Lab 5</u>

1) a) True
b) False; there will be 6 interfaces total in the routers between the source and destination, but the source and destination each have an interface as well.
c) True; the datagram data field is 60 bytes, and each header is 20 bytes, making for 100 bytes total. 40/100 = 40% header overhead
d) True
e) False; the TTL is a counter which is decremented at each router.

2) a) No, it is not possible for two packets to be forwarded from two different input ports to two different output ports with a shared bus since only one packet may move along the shared bus at a time.
b) No, it is not possible for the two packets to be forwarded to two different out put ports if the fabric uses switching via memory since the processor can only perform one read/write operation per clock cycle.
c) No, it is not possible for two packets to be forwarded to the same output port if the fabric uses a crossbar since only one packet can move along a bus at a time.

3) A subnet with prefix 128.119.40.128/26 means that all IP addresses in this subnet must have the same first 26 bits as the given IP; therefore, the subnet mask for the above subnet would be 255.255.255.192, so an example of a valid IP in this subnet would be 128.119.40.136.
If an ISP owned the block of addresses in the subnet 128.119.40.64/26, that means it owns 2^6=64 IP addresses. Dividing this evenly by four subnets gives us 64/4=16 IP addresses (with 16-2=14 possible hosts) per subnet in the ISP, divided in the following blocks:
Subnet 1: 128.119.40.64/28 *Subnet 2:* 128.119.40.80/28
Subnet 3: 128.119.40.96/28 *Subnet 4:* 128.119.40.112/28

4) The subnet prefix for the router 223.1.17/24 has 2^8=256 allowable IP addresses. Since the first subnet needs 60 interfaces, the second 90, and the third 12, for a total of 162, we can safely assign more interfaces per subnet than are strictly needed.

- The first subnet receives 64-2=62 allowable IP addresses, with a network address of 223.1.17.0/26.
- The second subnet gets 128-2=126 allowable IP addresses, with a network address of 223.1.17.128/25.
- The third subnet 16-2=14 IP addresses, with a network address of 223.1.17.192/28.

5) a) Using the given code snippet, 223.1.17.0 is global, 174.3.61.7 is also global, and 192.168.1.1 is private. This checks out when cross-referencing the given IP addresses against the Special Address Ranges (by the way, the link in the assignment takes you to the Python ipaddress module tutorial).
b) i.If we use one more bit to subnet 128.119.40.64/26, the new subnets become 128.119.40.64/27 and 128.119.40.96/27.
ii. Since both subnets are indicated by the first 27 bits, the subnet masks for each will be the same, namely 255.255.255.224
iii. The broadcast address is the highest address within the subnet, so for the first subnet, the broadcast address is 128.119.40.95, and for the second, 128.119.40.127. (The binary representation of the first network address is **1000 0000.0111 0111.0010 1000.010**0 0000. Since first 27 bits stay the same for all

IP addresses in the subnet, the highest address in the subnet is the bold digits followed by all 1s, or **1000 0000.0111 0111.0010 1000.010**1 1111, or 128.119.40.95. Likewise for the second subnet, the address is given by **1000 0000.0111 0111.0010 1000.011**0 0000, and the highest address in the subnet is **1000 0000.0111 0111.0010 1000.011**1 1111, or 128.119.40.127. I realized after doing these calculations that the Python module does all of this for you, so I have included the code I used as well).
iv) Each subnet contains 32 addresses (see include Python code).
c) As I calculated in question 4, the third subnet gets 16-2=14 addresses for usage by hosts, which is backed up by the included Python code.

6) a)

Destination Address Prefix	Link Interface
1110 0000	0
1110 0001 0000 0000	1
1110 0001	2
Otherwise	3

b) i) Since this address does not match any of the prefixes in interfaces 0-2, the datagram would be forwarded to link interface 3.
ii) Although this address matches both links 1 and 2, using longest matching prefix, the datagram would be forwarded to link 1.
iii) The address matches the prefix for link interface 3, so the datagram would be forwarded to interface 3.

7) The IP datagram is 2000 bytes and the MTU is 500 bytes. This means the IP datagram has 1980 bytes of IP payload and 20 bytes of IP header. The datagram will be split as follows:

Fragment	Bytes (in the data field of IP datagram)	Offset (8 byte chunks)
1	480	**0**
2	480	480/8=**60**
3	480	960/8=**120**
4	480	1440/8=**180**
5	60	1920/8=**240**

8) The datagram size is 1500 bytes, and carries IP and TCP headers, each of length 20 bytes. This means 1460 bytes of data can be sent per datagram. The total file size is 5,000,000 bytes, so the total number of datagrams is given by ceiling(5,000,000/1460) = 3425. The first 3424 datagrams will be 1500 bytes each, while the last one will be 5,000,000 - 3424 * 1460 + 40 = 1000 bytes long.