Jason Lai
204995126
Dis 1A

CS 118
May 21, 2020

Homework 6

1 Problem 1

Suppose two packets arrive to two different input ports of a router at exactly the same time. Also suppose there are no other packets anywhere in the router.

(a) Suppose the two packets are to be forwarded to two different output ports. Is it possible to forward the two packets through the switch fabric at the same time when the fabric uses a shared bus?

(b) Suppose the two packets are to be forwarded to two different output ports. Is it possible to forward the two packets through the switch fabric at the same time when the fabric uses switching via memory?

(c) Suppose the two packets are to be forwarded to two different output ports. Is it possible to forward the two packets through the switch fabric at the same time when the fabric uses a crossbar?

a) No, if the fabric is using a shared bus, then only one packet can be forwarded through the switch fabric at a time, since only one packet can cross the bus at a time.

b) No, if the fabric is using switching via memory, then only one packet can be forwarded through the switch fabric at a time, since the processor memory that this method uses also utilizes the system bus, and only one memory read/write can be done at a time over the shared system bus.

c) Yes, if the fabric is using a crossbar, then multiple packets can be forwarded through the switch fabric at a time, since the crossbar switch is non-blocking, so it allows multiple packets to be forwarded in parallel. More specifically, if the router has N input ports and N output ports, the crossbar switch will be comprised of 2N busses (N input busses crossed with N output busses), where a switch controller can direct packets to their destined output ports by closing crosspoints. In this case, two packets can be forwarded through the switching fabric at the same time, because the two packets are using different input and output ports, so they will use different input and output busses.

2 Problem 2

Consider a router that interconnects three subnets: Subnet 1, Subnet 2, and Subnet 3. Suppose all of the interfaces in each of these three subnets are required to have the prefix `224.1.17.0/24`. Also suppose that Subnet 1 is required to support at least 60 interfaces, Subnet 2 is to support at least 90 interfaces, and Subnet 3 is to support at least 8 interfaces. Provide three subnet addresses (of the form `a.b.c.d/x`) that satisfy the constraints. You may use the following link to help verify your result: `http://jodies.de/ipcalc`.

Subnet 1: 224.1.17.0/26

Subnet 2: 224.1.17.64/25

Subnet 3: 224.1.17.192/28

Work:

For Subnet 1, we need at least 60 interfaces, therefore we need 6 bits, which gives us 62 usable, unique IP addresses. This is because 6 bits gives us $2^6 = 64$ addresses, but only 62 are available to be used as host IDs, since 2 addresses (the first and last) are reserved as the network and broadcast addresses. Since we only need 6 bits to cover the range of 60 interfaces supported, we can extend the subnet mask to $32 - 6 = 26$, which gives us slash-26 with the first IP address being 224.1.17.0.

For Subnet 2, we need at least 90 interfaces, therefore we need 7 bits, which gives us 126 usable addresses, as 7 bits gives us $2^7 = 128$ addresses, but only 126 usable due to the two reserved addresses. We can then extend the subnet mask to $32 - 7 = 25$, which gives us slash-25 with the first IP address being 224.1.17.64 (one address after the last address of Subnet 1 i.e. 224.1.17.63).

For Subnet 3, we need at least 8 interfaces, therefore we need 4 bits, which gives us 14 usable addresses, as 4 bits gives us 14 usable addresses. Notice that we don't use 3 bits, which would give us $2^3 = 8$ addresses, because only 6 would be usable as host IDs, due to the two reserved addresses, so we instead use 4 bits to get $2^4 = 16$ addresses. We then extend the subnet mask to $32 - 4 = 28$ with the first IP address being 224.1.17.192 (one address after the last address of Subnet 2 i.e. 224.1.17.191).

3 Problem 3

Consider sending a datagram with total length 2400 B into a link that has an MTU (maximum transmission unit) of 800 B. Suppose the original datagram is stamped with the identification number 421 and all IP headers are 20 bytes.

(a) How many fragments are generated?

(b) What are the values in the various fields (header length, total length, identification, MF flag, and fragment offset) in the IP datagram(s) generated related to fragmentation?

a) 4 fragments are generated.

b)

Work:

a)
If the datagram is 2400 B in length, minus the 20 Byte header, we have 2380 B of data to be sent. Since the MTU is 800 B, we have that each fragments data portion is 780 B in size. So we have: $\frac{2380\text{B}}{776\text{B}} = 3$ with a remainder of 52, so we need 4 fragments. The reason we divide by 776 instead of 780 is because the data is read in as 8 B blocks so that the offset field is set correctly, therefore, the amount of data in each fragment besides the last must be a multiple of 8.

b)

Fragment (in order)	Header Length	Total Length	Identification	MF Flag	Fragment Offset
1	20	796	421	1	0
2	20	796	421	1	97
3	20	796	421	1	194
4	20	72	421	0	291

The first three fragments all have a 776 Byte payload, which is 97 8 Byte blocks each. This means that each offset field starting from 0 at the first block will be incremented by 97 for all subsequent fragments. IP header lengths are given to be 20 Bytes, so the total length of the first three fragments is 796 Bytes. The last fragment has $2380 - (776 \times 3) = 52$ Bytes for the payload, plus the 20 Byte header, to get a total length of 72 Bytes. All four fragments have the same identification number as the original datagram: 421, and the MF flag denotes whether there is more data coming in fragments, so every fragment besides the last one will have an MF flag of 1.

4 Problem 4

Please answer the following questions regarding checksum.

(a) Why is the IP header checksum recalculated at every router?

(b) What is covered by IP checksum and TCP checksum?

a) The IP header checksum is recalculated at every router, since the Time-To-Live field, and possibly the options field, of the datagram will change. The Time-To-Live field will change as this field is decremented by one each time the datagram is processed by a router, thereby changing the value of the checksum. As such, the checksum must be recomputed and stored again at each router.

b) The IP checksum only covers the IP header of the datagram and not the actual data itself, whereas the TCP checksum covers the entire TCP segment, which includes the TCP header and the payload.

5 Problem 5

In this problem we will explore the impact of NATs on P2P applications. Suppose a peer with username Arnold discovers through querying that a peer with username Bernard has a file it wants to download. Also suppose that Bernard and Arnold are both behind a NAT. Try to devise a technique that will allow Arnold to establish a TCP connection with Bernard without application-specific NAT configuration. If you have difficulty devising such a technique, discuss why.

There would be difficulty devising such a technique without using an application-specific NAT configuration, since if Bernard is behind a NAT (as is Arnold), Arnold cannot initiate the three-way handshake to establish a TCP connection with Bernard. This is due to the fact that when Arnold sends the SYN packet to Bernard, the request goes to the NAT address, since Bernard's real IP address and ports are abstracted by the NAT. The SYN packet that arrives at the NAT will be dropped, since there isn't an entry in the NAT translation table for Bernard, since Bernard did not have any outgoing requests, so the NAT will not know who the SYN packet is for, thus establishing the TCP connection fails.