18-756 Project 4

Mingda Zhang

November 12, 2015

1 Simple explanation of my implementation code

I have modified Switch.java files. For input queueing, each time interval the switch will go over every in port to see if there is packet in the queue. If the packet is going to an out port haven't be taken. Then I remove the packet from the input queue and send the packet to the corresponding out port. For output queueing, each time interval I check if the in port has packet. If so, I direct send the packet to the out port. In addition, I have change the ExampleTA.java file to keep update the delay time.

2 TestCase

I have randomly generate 4 set of testcase for each size of switch. All the test files are in the "testcase" folder. In these testcases every source computer will send same number of packet. I have generate testacses with queue length 10,20,50 and 100. The destinations of these packet are randomly generated.

3 Experiment result

Switch buffer size = 20: From the above table we could see that output queueing is not suffering from front of line

Data	2×2		4×4		8×8		16×16		32×32		64×64	
Max Delay	input queue	output queue	input queue	output queue	input queue	output queue	input queue	output queue	input queue	output queue	input queue	output queue
10	15	11	18	13	23	14	26	19	23	17	25	18
20	32	23	36	25	42	27	44	28	47	32	49	32
50	81	58	84	59	96	61	107	68	107	68	123	66
100	144	107	182	122	202	113	215	115	215	128	229	119

problem. Therefore that kind of switch has higher throughput. As for the switch size, I only compare the scenario where each port has same length input queue. As the switcher growing larger the max delay time get larger too.

4 Further Analysis

It is easy to understand the fact that output queueing has larger throughput than input queueing. Let's focus on the size of the switchers. If the packet is uniform distributed. For 2 by 2 switcher the probability of collision is $1/2$. This consistent with our experiment result. The max delay is about 1.5 times of the length of the queue.