ECE 358: Computer Networks
Spring 2018
Project 2: Data Link Layers and ARQ Protocol

Date of submission: Nov 16, 2018

Submitted by: David (Yan Liang) Qian
Student ID: 20568467
Waterloo Email: ylqian@uwaterloo.ca

Marks Received:
Marked by:

Question 1

Explanation of simulator

I implemented a GBN sender/receiver since it is the generalization of ABP to larger window sizes. I simply set window_size = 1 for this experiment.

The sender has a buffer size of 1. It repeatedly pulls data from the upper layer to send. The SEND() function mentioned in the description is in simulation.py and passed to the simulator as udt_send_fn(). The simulator sends all the packets in the buffer (only one packet) and sets a timeout based on when it was sent (current_time + transmission delay). The packet passes through an unreliable channel to receiver, which returns an ACK through the channel to generate an event in the event queue. The sender then processes the next event in the event queue. If it is a faulty ack, it goes on the process the next event until it gets a good ACK or timeout. If it gets a good ACK or timeout, it clears the buffer and adds another packet (only if good ACK), purges timeout in event queue, and restarts the send process for the packet in the buffer.

Results

Here are the throughput (bytes/second) results of the experiment:

delta/prop	2t = 10ms			2t = 500ms		
	BER=0.0	BER=1e-5	BER=1e-4	BER=0.0	BER=1e-5	BER=1e-4
2.5	119305	102664	29690	2984	2543	724
5	119305	91502	18780	2984	2207	398
7.5	119305	82843	13416	2984	1970	285
10	119305	75559	10377	2984	1765	222
12.5	119305	68298	8603	2984	1606	180

Discussion

We see that with BER=0, the throughput doesn't change regardless because there are errors, and no bandwidth is wasted on retransmission. We see that as error increases, the throughput decreases rapidly since much of the bandwidth is wasted on retransmission. We see that for higher propagation delay, the throughput decreases. This is obvious as it takes more time to send the packet and so even though utilization is 100%, throughput decreases.

Question 2

Explanation of simulator

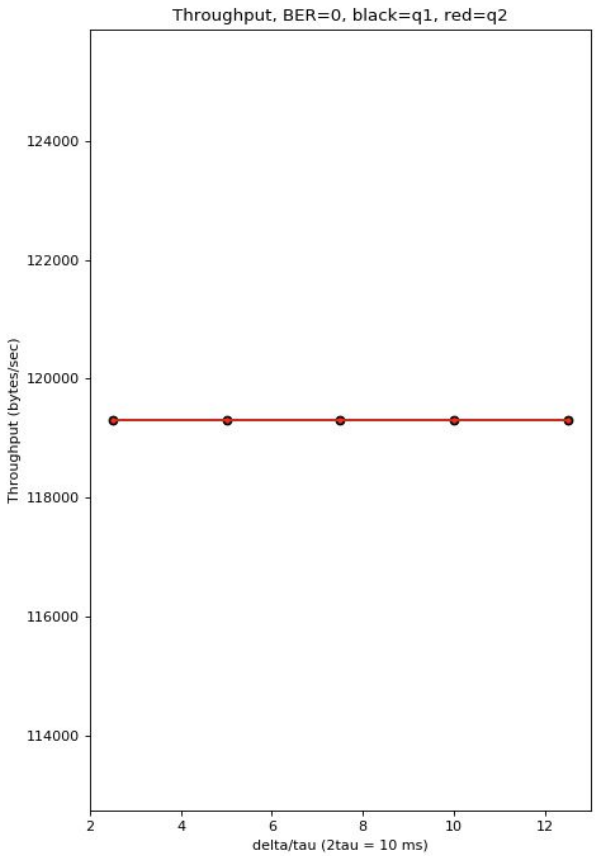
This is the same simulator as described in part one with one small different. Now, whenever we get a corrupted/wrong seq no ack, we immediately set p = 0 (pointer to next packet to send) and resend the only packet in the buffer instead of waiting for a timeout.

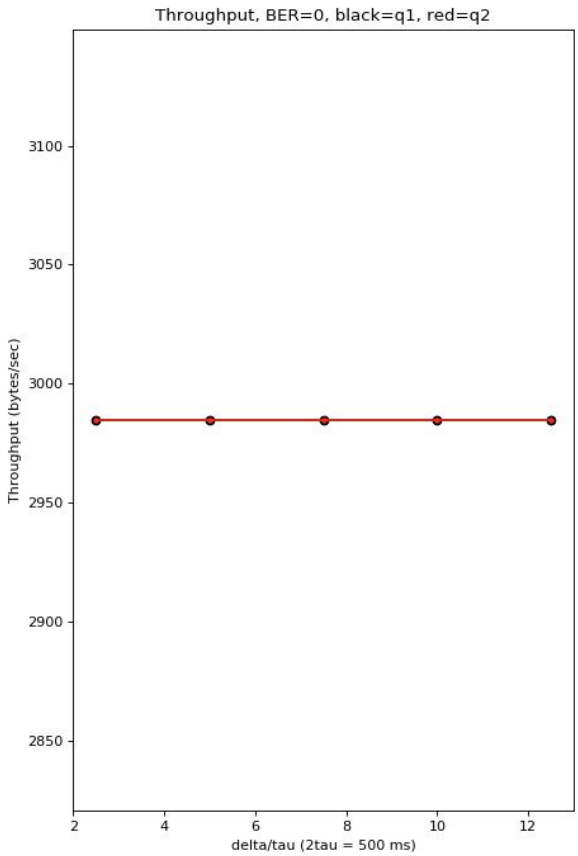
Results

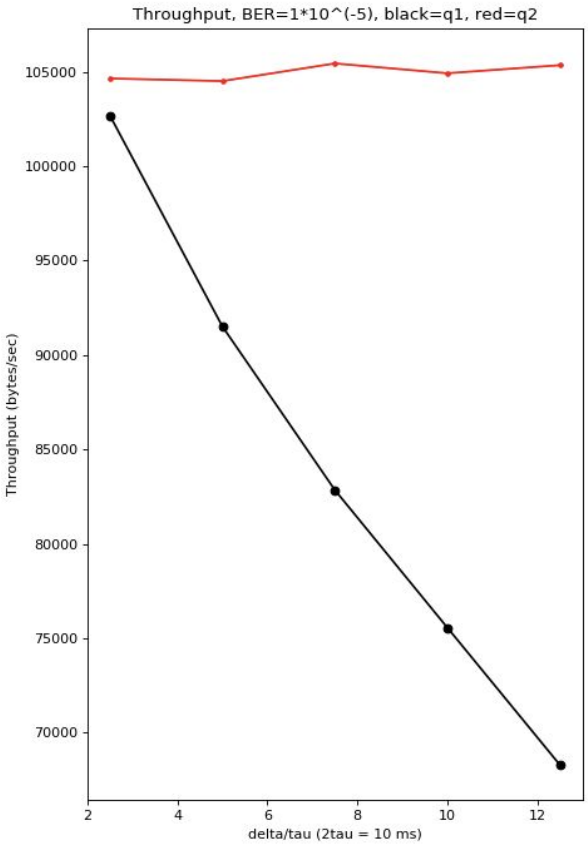
Here are the throughput results of the experiment:

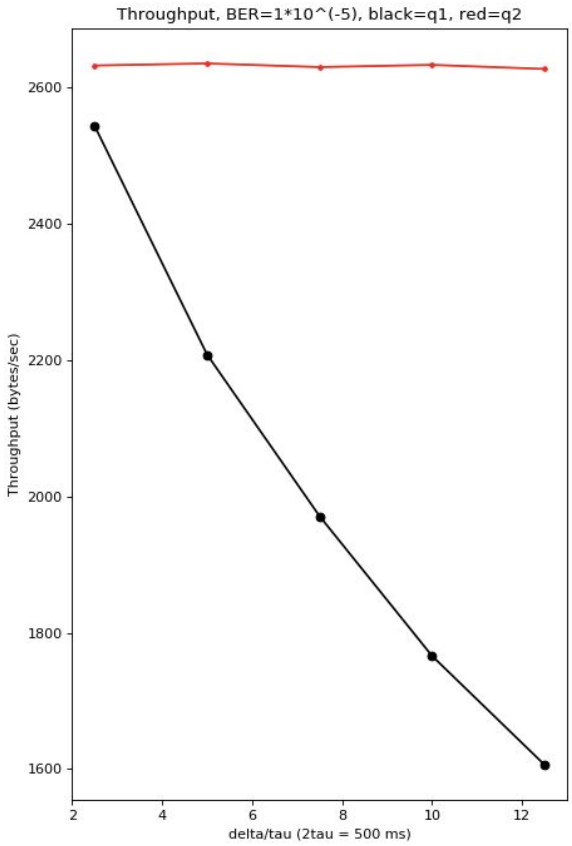
delta/prop	2t = 10ms			2t = 500ms		
	BER=0.0	BER=1e-5	BER=1e-4	BER=0.0	BER=1e-5	BER=1e-4
2.5	119305	104636	33749	2984	2631	850
5	119305	104499	33880	2984	2635	834
7.5	119305	105431	32892	2984	2629	824
10	119305	104912	32274	2984	2632	825
12.5	119305	105338	32235	2984	2626	815

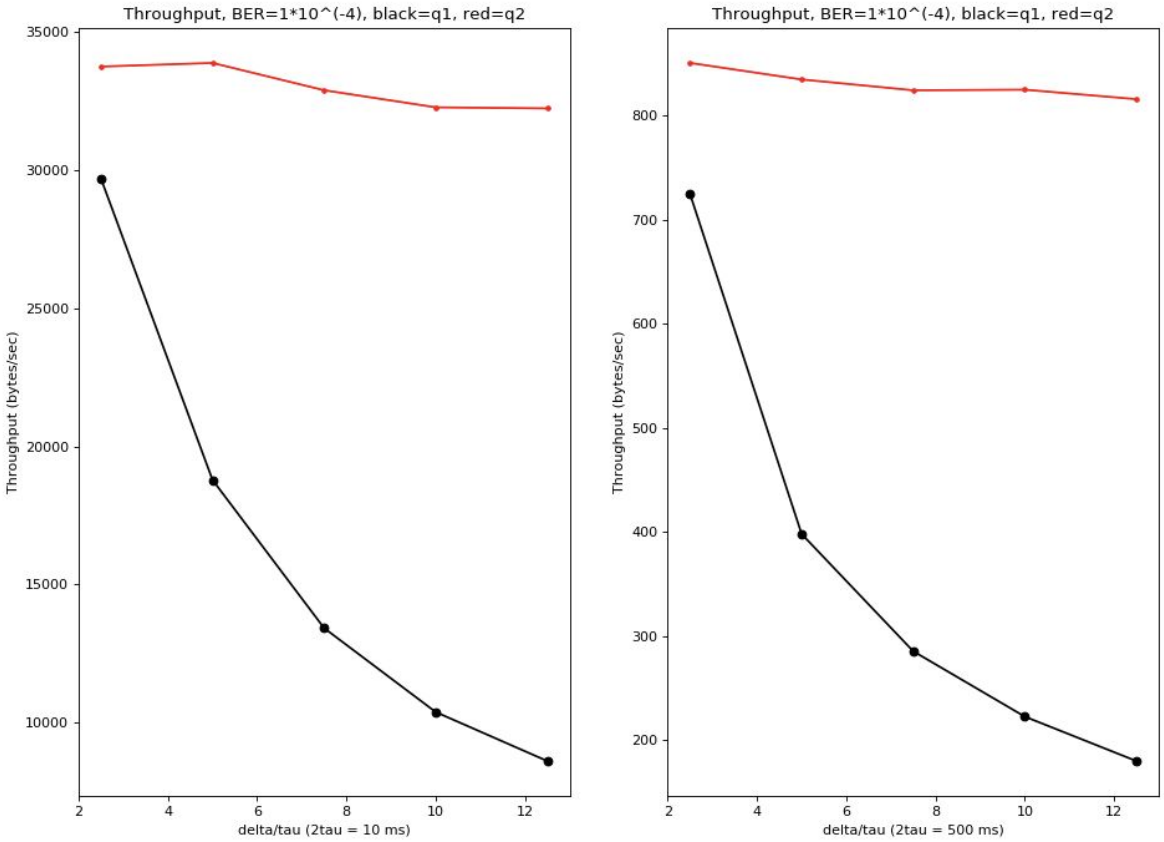
Here are the graphs comparing ABP with ABP_NAK:


Throughput, BER=0, black=q1, red=q2
Throughput (bytes/sec)
124000
122000
120000
118000
116000
114000
2
4
6
8
10
12
delta/tau (2tau = 10 ms)


Throughput, BER=0, black=q1, red=q2
Throughput (bytes/sec)
3100
3050
3000
2950
2900
2850
2
4
6
8
10
12
delta/tau (2tau = 500 ms)


Throughput, BER=1*10^(-5), black=q1, red=q2
Throughput (bytes/sec)
105000
100000
95000
90000
85000
80000
75000
70000
2
4
6
8
10
12
delta/tau (2tau = 10 ms)


Throughput, BER=1*10^(-5), black=q1, red=q2
Throughput (bytes/sec)
2600
2400
2200
2000
1800
1600
2
4
6
8
10
12
delta/tau (2tau = 500 ms)



Question 3

Explanation of simulator

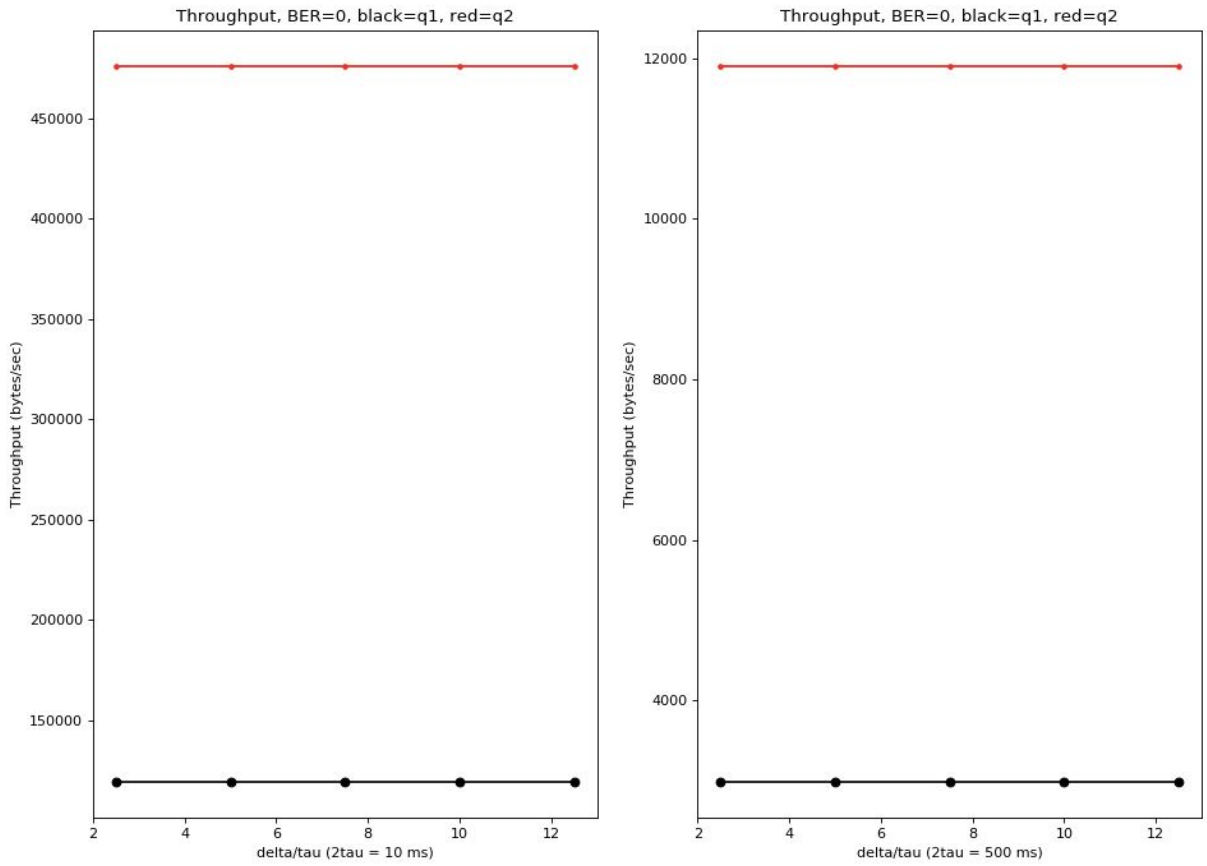
Instead of having a window size of 1, we now have a window size of 4. The process is to fill the buffer with data from the upper layer using fill_buffer(). We then have next_packet_to_send_idx pointing to the next packet to send. We send according this pointer and we increment it every time. Between sends, we pull from the event queue of ALL events that happening during transmission. We process all events during transmission one by one, and we continue with the next send. We do so until we send enough packets.

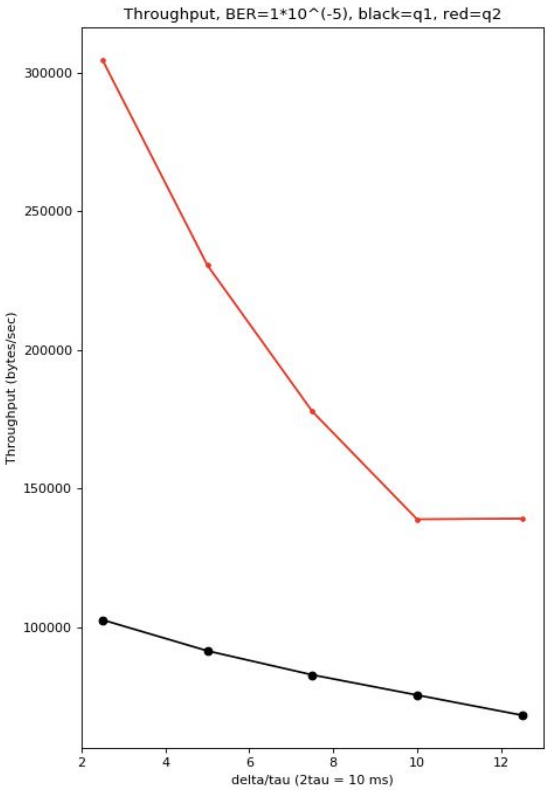
Here are the throughput results of the experiment:

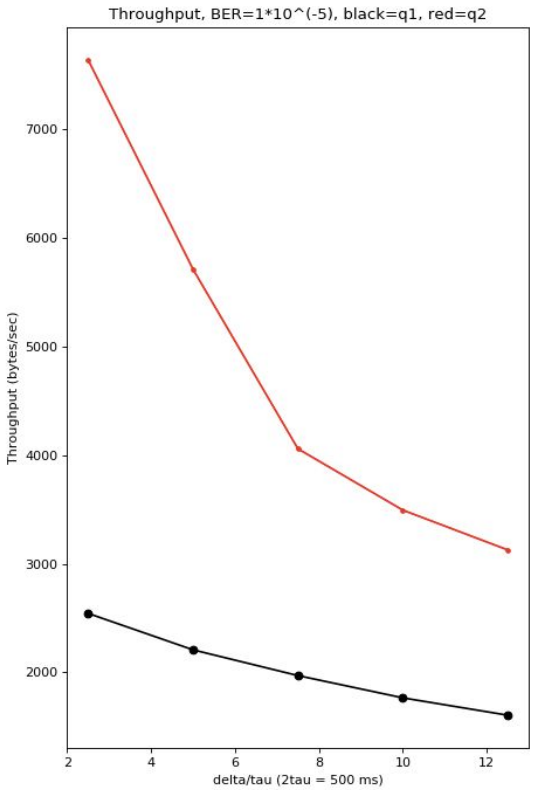
delta/prop	2t = 10ms			2t = 500ms		
	BER=0.0	BER=1e-5	BER=1e-4	BER=0.0	BER=1e-5	BER=1e-4

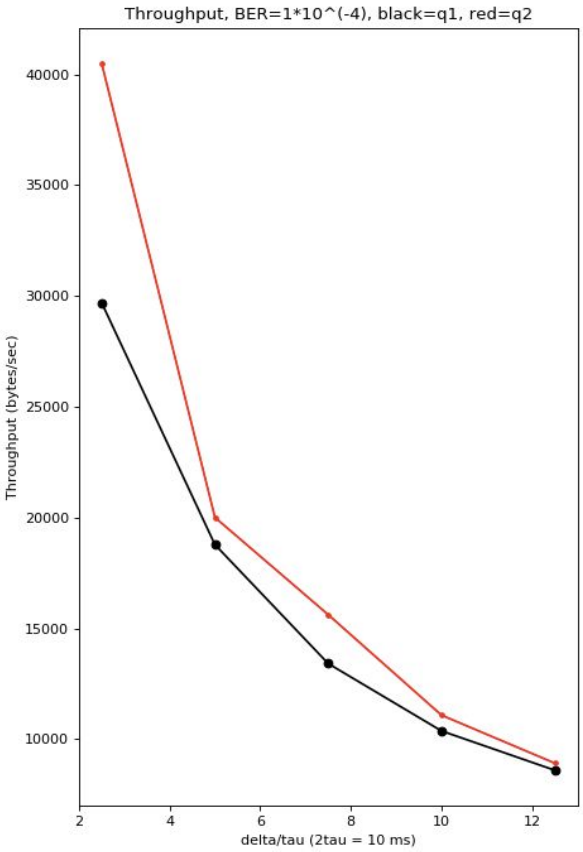
2.5	475794	304496	40491	11902	7635	985
5	475794	230514	20009	11902	5709	478
7.5	475794	177830	15622	11902	4058	303
10	475794	138931	11087	11902	3493	239
12.5	475794	139176	8923	11902	3127	193

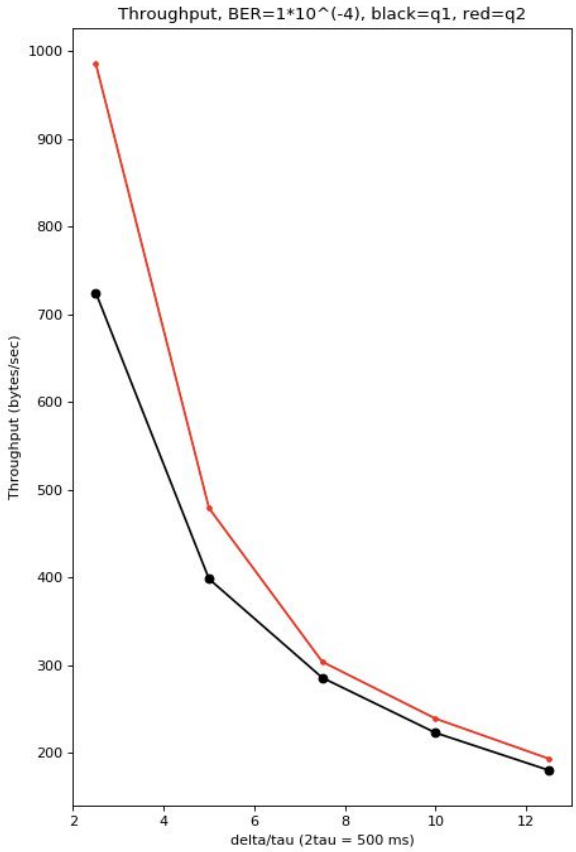
Here are the graphs comparing ABP with GBN:(red q3)




Throughput, BER=1*10^(-5), black=q1, red=q2
300000
250000
200000
150000
100000
Throughput (bytes/sec)
2
4
6
8
10
12
delta/tau (2tau = 10 ms)


Throughput, BER=1*10^(-5), black=q1, red=q2
7000
6000
5000
4000
3000
2000
Throughput (bytes/sec)
2
4
6
8
10
12
delta/tau (2tau = 500 ms)


Throughput, BER=1*10^(-4), black=q1, red=q2
Throughput (bytes/sec)
delta/tau (2tau = 10 ms)
40000
35000
30000
25000
20000
15000
10000
2
4
6
8
10
12


Throughput, BER=1*10^(-4), black=q1, red=q2
Throughput (bytes/sec)
delta/tau (2tau = 500 ms)
1000
900
800
700
600
500
400
300
200
2
4
6
8
10
12