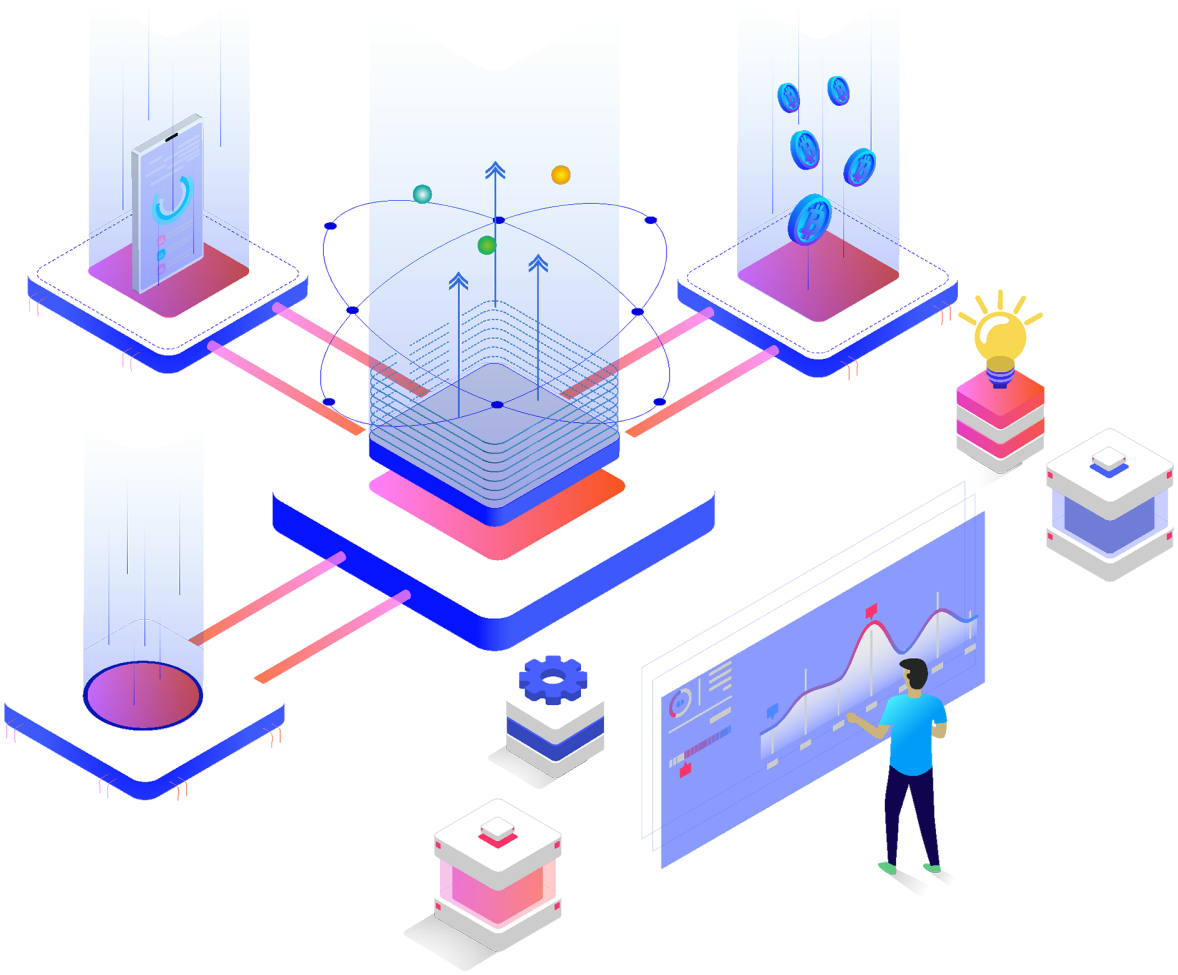
The Effects of Buffer size and Bandwidth in Relation to Packets Dropped.



Fardin Mahid
UCID:30114084

Dec 8, 2023

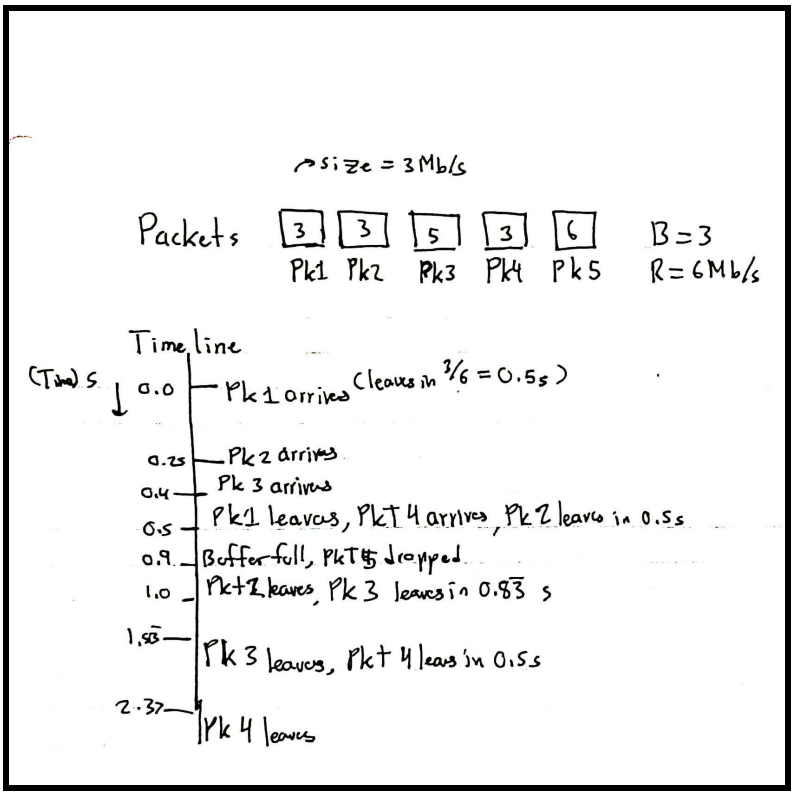
Abstract

The world today is a fast-paced and timely metropolis. There is almost no time for mistakes. Quality of service must be at its very best to serve the ever-needy and ever growing population.We, as humans naturally have desires, and our desires The standard for good internet in the workplace is but one of the many examples of our desires. We want a good, reliable tradeoff between the loss of packets and the speed of our systems . We believe that Mr. Fred's dubious decision of restricting us to only 5 Mb/s of WLAN and his inability to make budget for a better Access Point is restricting us of the best QoS we could possibly have. In this paper, I will go through a simulation to show packet loss compared to two variables: Buffer size and WLAN. I will then present my findings in a neat and ordered conclusion at the end, showing optimal buffer size and bandwidth. I will also show any mistakes in my experiment before presenting my results. Finally, I will answer the question of weather to buy buffer space or bandwidth.

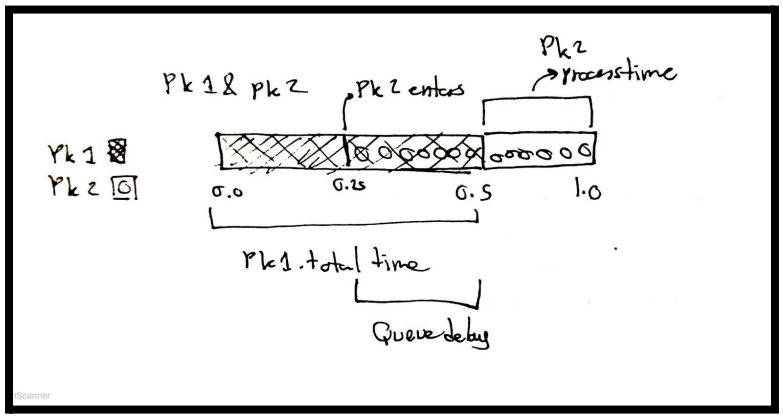
Experiment and code

To test the reliability difference between buffer space and bandwidth, I will run a discrete event simulation to simulate a network and its ability to drop and deliver packets. It has two states, ARRIVAL and DEPARTURE in which it will function. Events are separated by time and each event will either be and ARRIVAL event or a DEPARTURE event. "Packets" are simply read files that contain two numbers per line. These numbers are ordered as 'time' (in seconds) and size (in Bytes). There will be two parts to this experiment, one where we make bandwidth (R = 5 Mb/s) constant and another where we keep the buffer size (B = 100) constant. For each part, we will measure two variables, packet loss percentage and queue wait times. In the first part, we will use the values B = {1,100,200,300,400,500,600,700,800,900,100} and for the second part, we will use values R = {5,6,7,8,9,10} Mb/s. I will go through my thought process and explain my code in more detail below.

Using sample data obtained from the fifth quiz of our Employee training class I created an algorithm to find packet loss and drop rate. I created a timeline of events that occurred to get the final avg packet delay and packet loss. Here is what that looked like:



I then realized that packets leave after processing for a little time. They only leave when the packet in front of them in queue has finished processing. The time it takes for a packet to enter, then leave the buffer, I called *totalTime*. The time it takes for the packet to process in the buffer I called *processTime.* *ProcessTime* is the time it takes for the packet to transmit and depart, and that is found through the equation: $ProcessTime = \frac{pkSize}{bandwidth}$ where pkSize is the size of the incoming packet in bytes. With these two variables having equations I needed one more for the queue delay of each packet. I quickly realized that queue delay is found through simply subtracting the incoming packet's *arrivalTime* with the previous packet's *totalTime.* These findings are illustrated below.



Code snippets for calculation shown here:

```
allPackets[i].totalTime = (i == 0) ? allPackets[i].processTime : allPackets[i -
1].totalTime + allPackets[i].processTime ;
allPackets[i].qTime = (i == 0 || allPackets[i-1].totalTime <=
allPackets[i].arrivalTime) ? 0 : allPackets[i-1].totalTime -
allPackets[i].arrivalTime;
```

Since I didn't know the specific arrival times I added an inline if statement for if the previous packet leaves before the new packet comes in to make sure that I'm not accounting for queue that simply isn't there.

To setup the event, I took all the incoming packets and put them in a Vector datastructure like so:

```
struct Packet{
    int num;
    double arrivalTime;
    double size;
    double processTime;
    double totalTime; //process time + queue wait // in this scenario its basically the total time of the system
    double qTime; // queue wait
};
```

```
Packet p;
while (zoomPacketFile >> p.arrivalTime >> p.size) {
    p.num = totalPacket++;
    p.processTime = p.size / bandWitdth;
    p.totalTime = 0;
    p.qTime = 0;
    allPackets.push_back(p);
}
```

I then organized those packets into an Event queue by adding them all to an Event Vector that was then put into a priority queue:

```
Event eA{ .timeStamp: allPackets[i].arrivalTime, .type: ARRIVAL, .packet: allPackets[i]};
Event eD{ .timeStamp: allPackets[i].totalTime, .type: DEPARTURE, .packet: allPackets[i]};
allEvents.push_back(eA);
allEvents.push_back(eD);
```

eA = EventArrival, eD = eventDeparture.

For every event, I scheduled it as either ARRIVAL or DEPARTURE and checked for the state in a while loop. In ARRIVAL I checked to see if there was space in the buffer, if there was, we added it to a packet queue , if not, we dropped it. In DEPARTURE, we checked to see if the packet was inside the packet queue already. If it isn't we simply assume it was dropped before and do nothing, else we prepare the packet for departure; we take it out of the queue.

```
while (!eventQueue.empty()) {
    Event currentEvent = eventQueue.top();
    eventQueue.pop();

    if (currentEvent.type == ARRIVAL) {
        if (packetQueue.size() >= BUFSIZE1) {
            packetDropped++;
        } else {
            packetQueue.push( x: currentEvent.packet);
        }
    } else if (currentEvent.type == DEPARTURE) {
        if (!packetQueue.empty()) {
            Packet frontPacket = packetQueue.front();
            totalQueueDelay += currentEvent.packet.qTime;
            packetQueue.pop();
            totalDelivered++;
        }
    }
}
```

All needed information which was collected through the process is then neatly displayed in the terminal as shown below:

```
Total delivered : 614272
Total Packets Processed: 689509
Total Packets Dropped: 75237
Average Queue Delay: 3.65589 seconds
drop average for buffer size of 1000 and wLan of 625000 is 0.109117
Process finished with exit code 0
```

AvgQueueDelay is calculated by dividing total queue by total packets delivered.

When trying out the code for yourself:

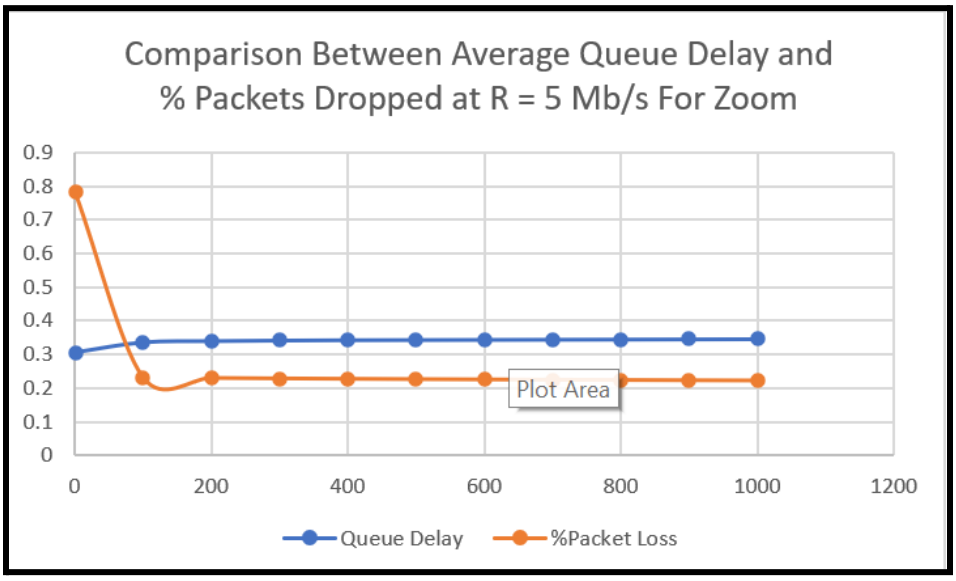
- Change bandwidth on line 9.
- Change buffersize on line 7.
- Change filename on line 62.

Results

Part 1: R = 5 Mb/s (const):

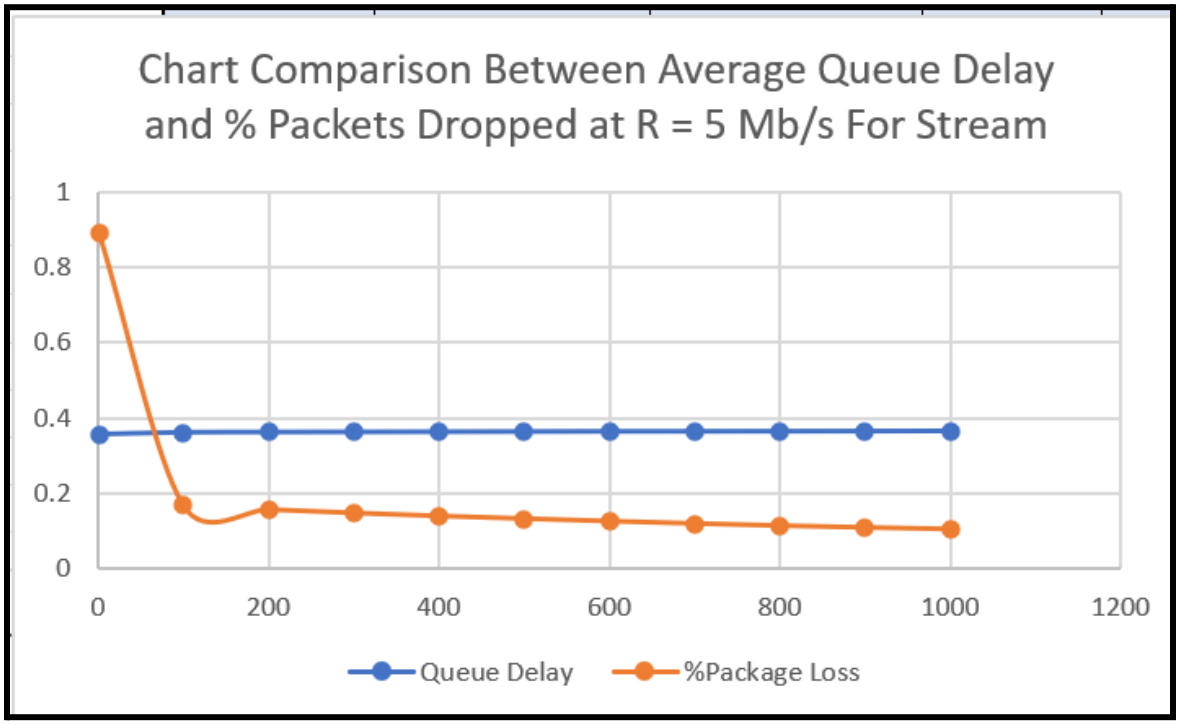
Zoom file:

Keeping bandwith consistent (R = 5Mb/s)				
Zoom File				
Buffer size	Packets Delivered	Packets Dropped	Queue delay(s)	Drop Avg
1	15604	556756	0.304836	0.78103813
100	547000	165741	0.336292	0.23250767
200	548522	164119	0.340285	0.23023227
300	549442	163099	0.342036	0.22880137
400	550157	162284	0.343187	0.22765806
500	550761	161580	0.34369	0.22667046
600	551313	160928	0.34419	0.22575581
700	551813	160328	0.344698	0.22491411
800	552313	159728	0.345204	0.22407241
900	552746	159195	0.34574	0.2233247
1000	553146	158695	0.346313	0.22262328



Stream file:

Keeping Bandwidth consisten				
Stream File				
Buffer size	Packets Delivered	Packets Dropped	Queue dela	Drop Avg
1	73919	615590	3.57497	0.892795
100	569390	120119	3.62153	0.168507
200	577463	112046	3.6281	0.157182
300	583875	105634	3.63172	0.148187
400	589613	99896	3.6355	0.140138
500	594606	94903	3.64075	0.133133
600	599363	90146	3.64509	0.12646
700	603800	85709	3.64602	0.120236
800	607837	81672	3.64848	0.114573
900	611258	78251	3.65139	0.109773
1000	614272	75237	3.65589	0.105545



*NOTE: I think I messed up the calculations somewhere here I'm not sure where. But Queue delay shouldn't be this high (for this one I will decrease the)

Dec 8, 2023

Part 2: B = 100 (Const):

R (Mb/s)	Zoom.txt	Stream.txt
5	Total delivered : 547000 Total Packets Processed: 712841 Total Packets Dropped: 165741 Average Queue Delay: 0.336354 seconds drop average for buffer size of 100 and wLan of 625000 is 0.232508 Process finished with exit code 0	Total delivered : 569390 Total Packets Processed: 689509 Total Packets Dropped: 120119 Average Queue Delay: 3.62513 seconds drop average for buffer size of 100 and wLan of 625000 is 0.174209 Process finished with exit code 0
6	Total delivered : 484734 Total Packets Processed: 712841 Total Packets Dropped: 228007 Average Queue Delay: 0 seconds drop average for buffer size of 100 and wLan of 750000 is 0.319857 Process finished with exit code 0	"C:\Users\Fardin Mahid\Desktop\CPSC 441\Assingment\Assign5\cmake-build-debug\Assign5.exe" Total delivered : 519186 Total Packets Processed: 689509 Total Packets Dropped: 170223 Average Queue Delay: 0.0291926 seconds drop average for buffer size of 100 and wLan of 750000 is 0.246876 Process finished with exit code 0
8	Total delivered : 397851 Total Packets Processed: 712841 Total Packets Dropped: 314890 Average Queue Delay: 0 seconds drop average for buffer size of 100 and wLan of 1e+06 is 0.441739 Process finished with exit code 0	Total delivered : 401151 Total Packets Processed: 689509 Total Packets Dropped: 288258 Average Queue Delay: 0.000114854 seconds drop average for buffer size of 100 and wLan of 1e+06 is 0.418063 Process finished with exit code 0
10	Total delivered : 346090 Total Packets Processed: 712841 Total Packets Dropped: 366651 Average Queue Delay: 0 seconds drop average for buffer size of 100 and wLan of 1.25e+06 is 0.514352 Process finished with exit code 0	Total delivered : 325414 Total Packets Processed: 689509 Total Packets Dropped: 363995 Average Queue Delay: 2.68066e-05 seconds drop average for buffer size of 100 and wLan of 1.25e+06 is 0.527905 Process finished with exit code 0

NOTE: For Some reason, my code gives questionable answers when bandwidth is greater than 5. I was not sure how to fix this. I will say buying a better AP would lead to better QoS but I know that bandwidth would be more worth while.

Conclusion

To conclude my findings. When Bandwidth was fixed at 5 Mb/s it could be seen that having a buffer size of greater than 200 would lead to diminishing returns on % of packets dropped and Queueing delay . so it does not make sense to buy another Access Point to increase buffer size. Having too little buffer size may lead to an increase in packets lost However. It would definitely be favourable if Mr. Fred could buy an Access Point with Buffer size ~200 but it is not really needed. This is only half of the experiment though. Buying Bandwidth makes the queuing delay almost negligent. The zoom file after 5 Mb/s is at 0s of avg queue delay whereas the stream file has a queue delay of almost 0s. The problem with buying bandwidth seems to come from the fact that buying bandwidth leads to a larger loss of packets; 10 Mb/s for both zoomfile and stream file are 4-5x worse than if the buffer size at 5Mb/s were equal to 1000. With this added information. I believe that buying a better Access Point would be more favourable because, while it leads to diminishing returns in QoS packet loss is a lot more manageable. This concludes my findings in this packet loss simulation.