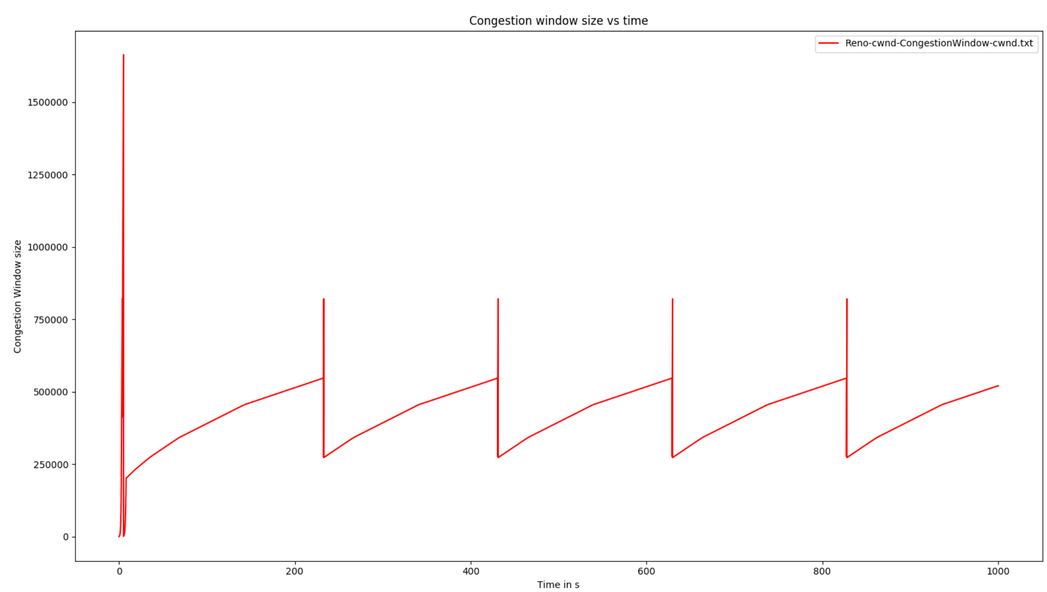
Analysis of TCP congestion control

Ugare Shubham Dipak[1], Rishav Chaurasia[2], and Nikhil Yadala[3]

[1] 140101076
[2] 140101086
[3] 140101083

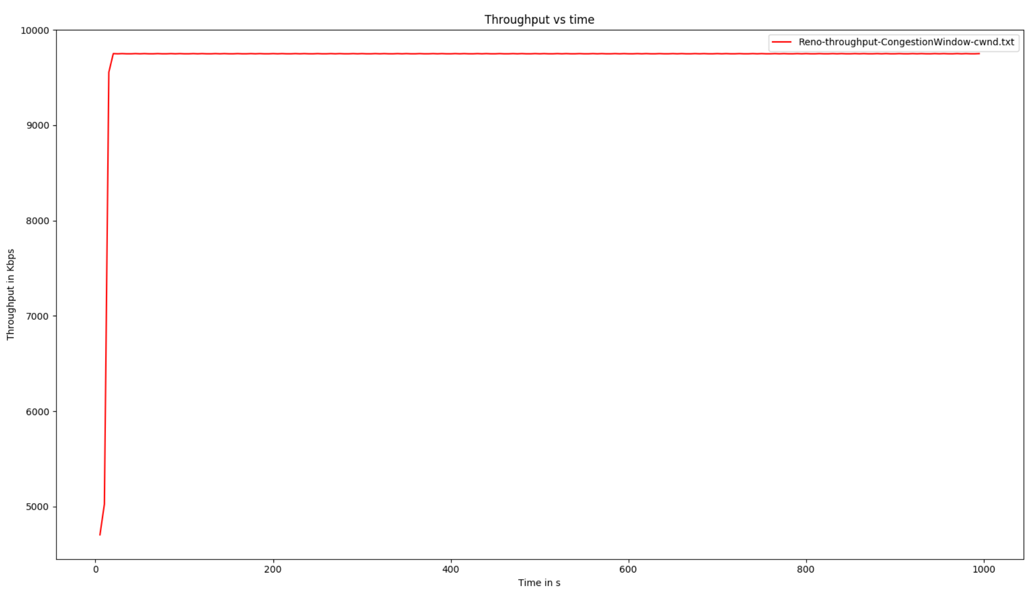
In the assignment we Analyze and compare TCP NewReno, TCP Vegas and TCP Westwood Vegas performance. We will use a Dumbbell topology with two routers R1 and R2 connected by a (10 Mbps, 50 ms) wired link. Each of the routers is connected to 3 hosts H1 to H3 are connected to R1 and H4 to H6 are connected to R2. The hosts are attached with (100 Mbps, 20ms) links. Both the routers use drop-tail queues with queue size set according to bandwidth-delay product. Senders (i.e. H1, H2 and H3) are attached with TCP NewReno, TCP Vegas, and TCP Westwood agents respectively. We choose a packet size of 1.2KB and perform the following task.

1. **We will start only one flow and analyze the throughput over sufficiently long duration (1000 seconds). We have Plotted evolution of congestion window over time. We perform this experiment with flows attached to all the three sending agents.** The reason for choosing the duration of 1000s is so that we can get the results close to the steady state as far as possible. Theoretically, we need to choose a duration much larger than the initial time of unsteady changes. In our case, by trials, we observed that the initial unsteadiness lasts for some 25 seconds, and so 1000s of duration should be enough to get trustworthy results.



   (a) In the above plot the protocol used is TCP new reno. We can observe here a saw tooth shaped graph where the increase and decrease in the congestion window size occurs periodically. As

```
IP Header size is: 20 Bytes.
TCP Header size is: 20 Bytes.
TCP ADU size is: 1169 Bytes.
Bottleneck Bandwidth is -> 9Mbps.
RTT is -> 0.18 seconds.
Queue size is -> 225000 Bytes.

Data for: TCP NewReno.
Flow 1 (10.0.1.1 -> 10.0.5.2)
Tx Bytes:   1242304116.
Rx Bytes:   1241582505.
Number of packets lost due to congestion is 239.
Net Throughput: 9.47252 Mbps.
Net Goodput: 9.06352 Mbps.
```



we can see, at near 200s, the increase in the window is a bit slow. This period corresponds to the "Slow start" stage. In an event of packet loss, the congestion window size immediately decreases (becomes half the previous size) and the slow start stage is again initialised. The event of loss can be triggered due to either the sender reaching threshold limit of window size or a packet loss or when there is a timeout for a packet sent.

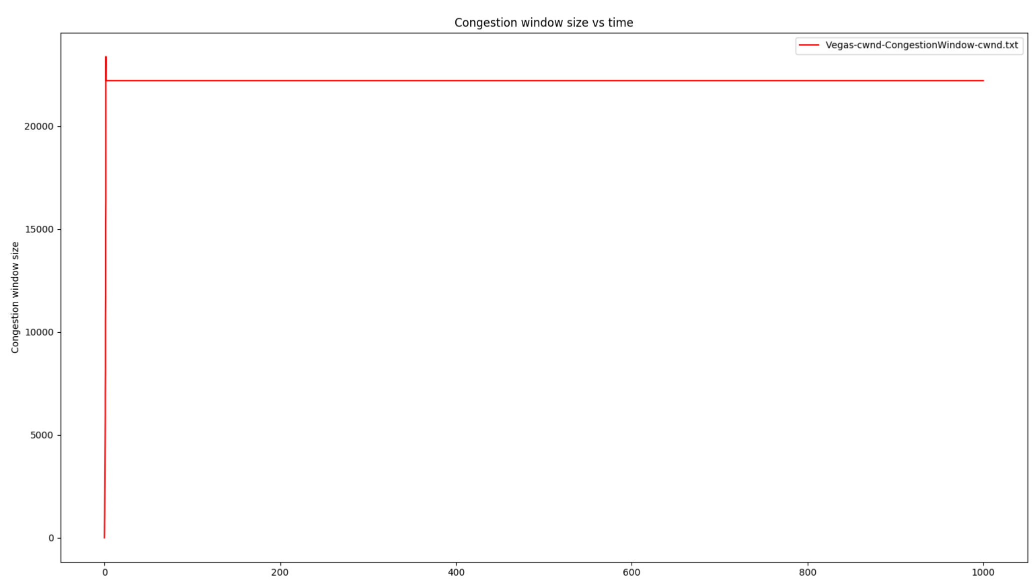
(b) TCP Vegas is a pure delay-based congestion control algorithm implementing a proactive scheme that tries to prevent packet drops by maintaining a small backlog at the bottleneck queue. Vegas continuously samples the RTT and computes the actual throughput a connection achieves using Equation 1 and compares it with the expected throughput calculated in Equation 2. The difference between these 2 sending rates in Equation 3 reflects the amount of extra packets being queued at the bottleneck.

$$actual = cwnd/RTT \tag{1}$$

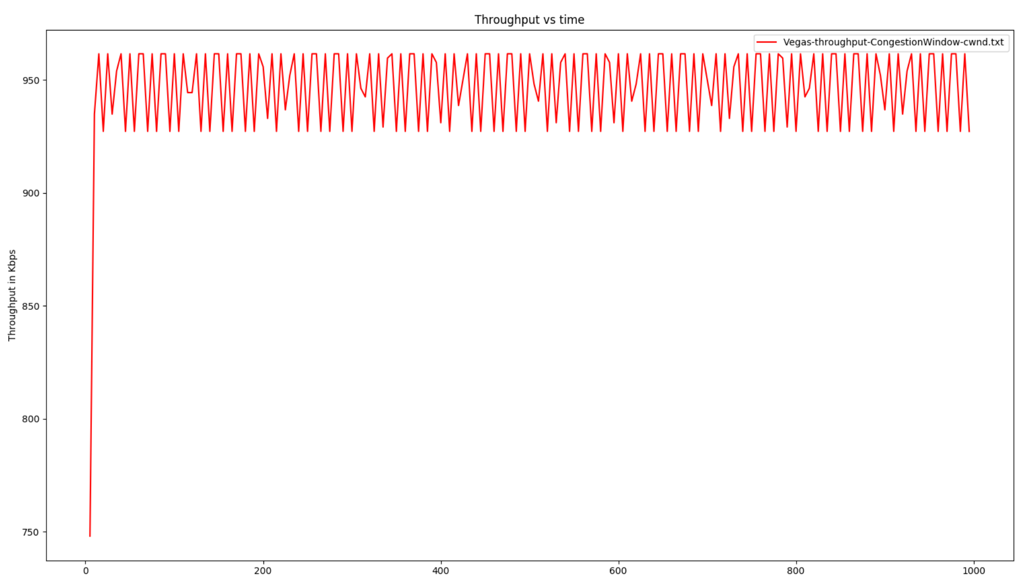
$$expected = cwnd/BaseRTT \tag{2}$$

$$diff = expected - actual \tag{3}$$

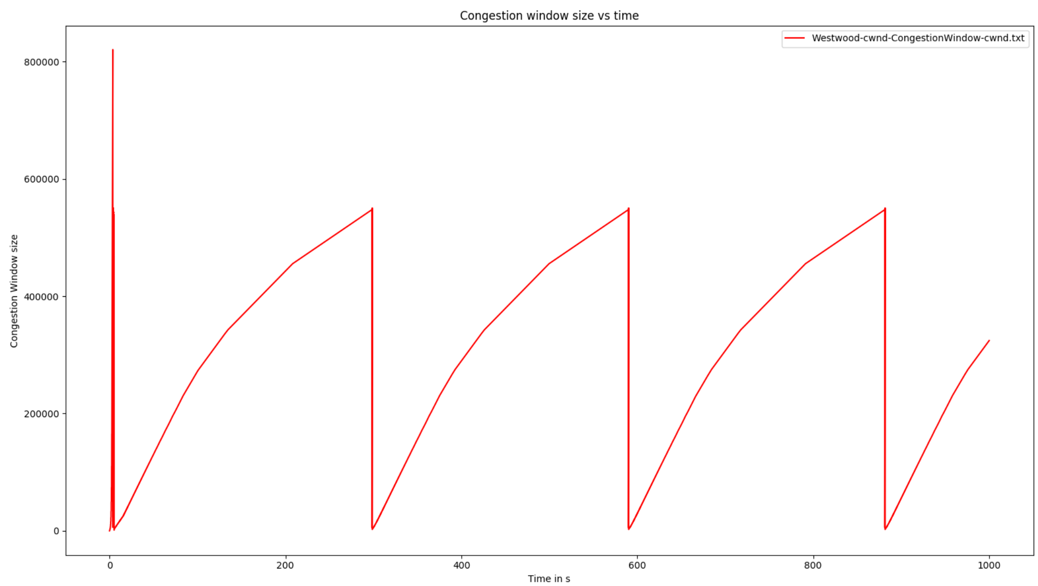
To avoid congestion, Vegas linearly increases/decreases its congestion window to ensure the diff value fall between the 2 predefined thresholds, alpha and beta. diff and another threshold, gamma, are used to determine when Vegas should change from its slow-start mode to linear increase/decrease mode. Following the implementation of Vegas in Linux, we use 2, 4, and 1 as the default values of alpha, beta, and gamma, respectively, but they can be modified through the Attribute system.



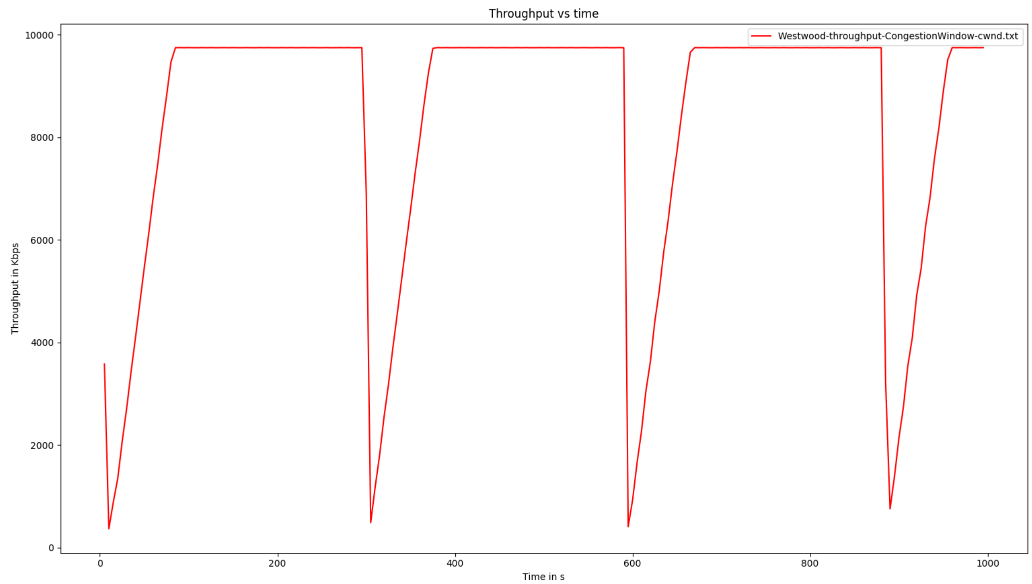
```
IP Header size is: 20 Bytes.
TCP Header size is: 20 Bytes.
TCP ADU size is: 1169 Bytes.
Bottleneck Bandwidth is -> 9Mbps.
RTT is -> 0.18 seconds.
Queue size is -> 225000 Bytes.

Data for: TCP Vegas.
Flow 1 (10.0.2.1 -> 10.0.6.2)
Tx Bytes:   121142844.
Rx Bytes:   121142844.
Number of packets lost due to congestion is 0.
Net Throughput: 0.924247 Mbps.
Net Goodput: 0.884884 Mbps.
```



(c) Westwood and Westwood+ employ the AIAD (Additive Increase/Adaptive Decrease) congestion control paradigm. When a congestion episode happens, instead of halving the cwnd, these protocols try to estimate the networks bandwidth and use the estimated value to adjust the cwnd. While Westwood performs the bandwidth sampling every ACK reception, Westwood+ samples the bandwidth every RTT.



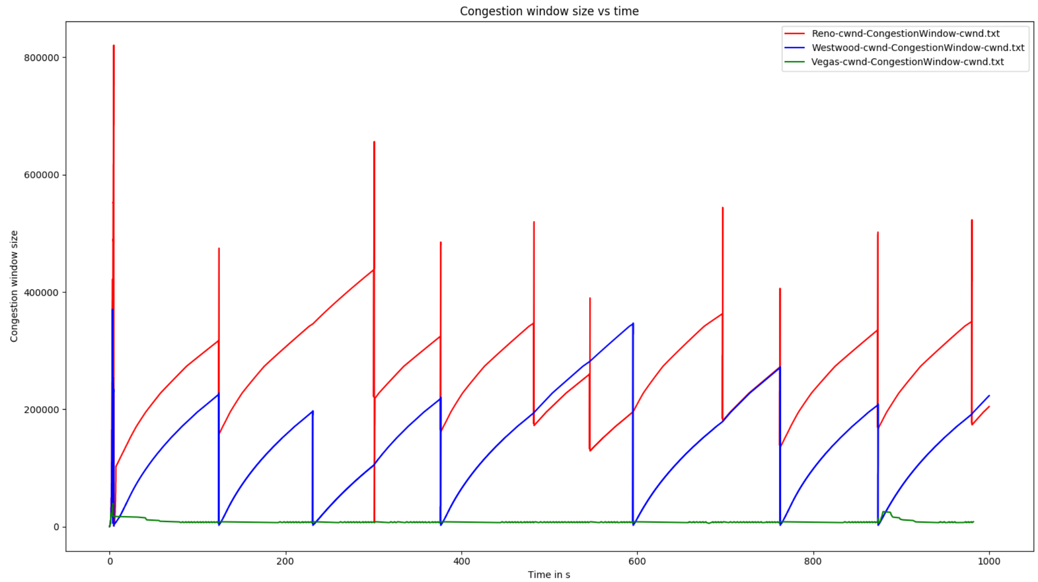

```
IP Header size is: 20 Bytes.
TCP Header size is: 20 Bytes.
TCP ADU size is: 1169 Bytes.
Bottleneck Bandwidth is -> 9Mbps.
RTT is -> 0.18 seconds.
Queue size is -> 225000 Bytes.

Data for: TCP Westwood.
Flow 1 (10.0.3.1 -> 10.0.7.2)
Tx Bytes:   1061117484.
Rx Bytes:   1060601001.
Number of packets lost due to congestion is 238.
Net Throughput: 8.09174 Mbps.
Net Goodput: 7.74308 Mbps.
```

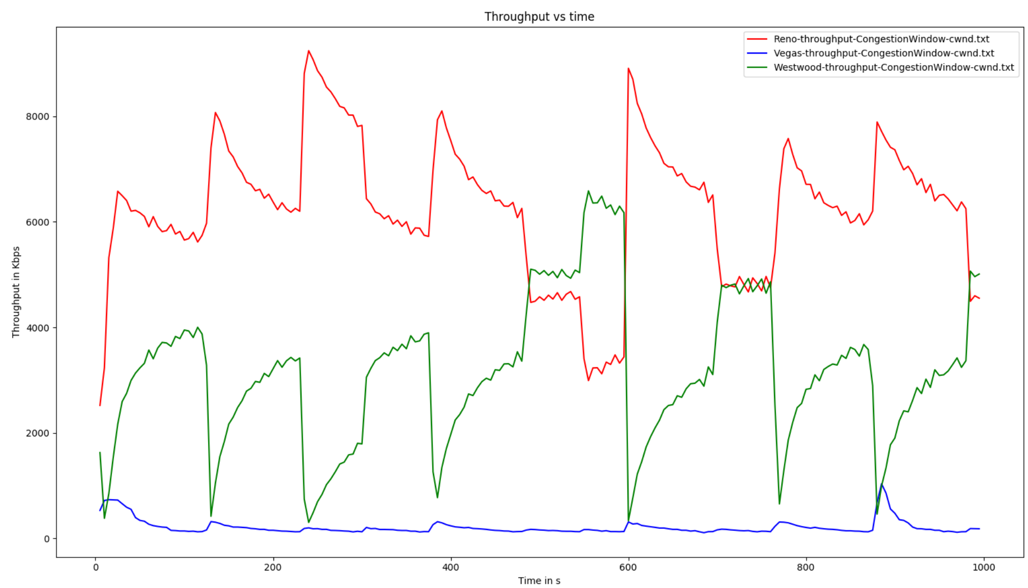


2. **We plot the throughput and congestion window of each flow at steady state and also maximum throughput of each flow**

```
Max throughput from data files are:-
Reno-throughput-withall.txt ->  9243.35 Kbps
Reno-throughput-alone.txt ->  9751.0 Kbps
Vegas-throughput-withall.txt ->  1037.85 Kbps
Vegas-throughput-alone.txt ->  961.55 Kbps
Westwood-throughput-alone.txt ->  9750.9 Kbps
Westwood-throughput-withall.txt ->  6587.68 Kbps
```

In this problem we will run all of the senders simultaneously. The result graphs are as expected. It is well known that even though TCP vegas performs better than TCP Tahoe or Reno in case of heavy packet loss. It gets dominated by these protocols when they are run simultaneously. We find in the graph that as NewReno and Westwood adjust there congestion window size to get more throughput without causing congestion. Throughput of Vegas reduces. Since we have 10 MBps as the channel capacity. Clearly the sum of throughput will always remain lesser than that value.


Congestion window size vs time
Reno-cwnd-CongestionWindow-cwnd.txt
Westwood-cwnd-CongestionWindow-cwnd.txt
Vegas-cwnd-CongestionWindow-cwnd.txt
Congestion window size
Time in s
0
200000
400000
600000
800000
200
400
600
800
1000


Throughput vs time
Reno-throughput-CongestionWindow-cwnd.txt
Vegas-throughput-CongestionWindow-cwnd.txt
Westwood-throughput-CongestionWindow-cwnd.txt
Throughput in Kbps
Time in s
0
2000
4000
6000
8000
200
400
600
800
1000

3. **In this part we will calculate congestion loss and Goodput over the duration of the experiment for each flow.**
   The goodput is measured in application layer while throughput is measured at lower layers. The value of goodput is lesser that throughput since the header size for TCP is not included as transmitted data in application layer.

```
IP Header size is: 20 Bytes.
TCP Header size is: 20 Bytes.
TCP ADU size is: 1169 Bytes.
Bottleneck Bandwidth is -> 10Mbps.
RTT is -> 0.18 seconds.
Queue size is -> 225000 Bytes.

Data for: TCP NewReno.
Flow 1 (10.0.1.1 -> 10.0.5.2)
Tx Bytes:   800668416.
Rx Bytes:   800292348.
Number of packets lost due to congestion is 133.
Net Throughput: 6.10575 Mbps.
Net Goodput: 5.84295 Mbps.

Data for: TCP Vegas.
Flow 2 (10.0.2.1 -> 10.0.6.2)
Tx Bytes:   27385917.
Rx Bytes:   27376149.
Number of packets lost due to congestion is 2.
Net Throughput: 0.208863 Mbps.
Net Goodput: 0.199968 Mbps.

Data for: TCP Westwood.
Flow 3 (10.0.3.1 -> 10.0.7.2)
Tx Bytes:   404318385.
Rx Bytes:   404065638.
Number of packets lost due to congestion is 109.
Net Throughput: 3.08278 Mbps.
Net Goodput: 2.94976 Mbps.
```