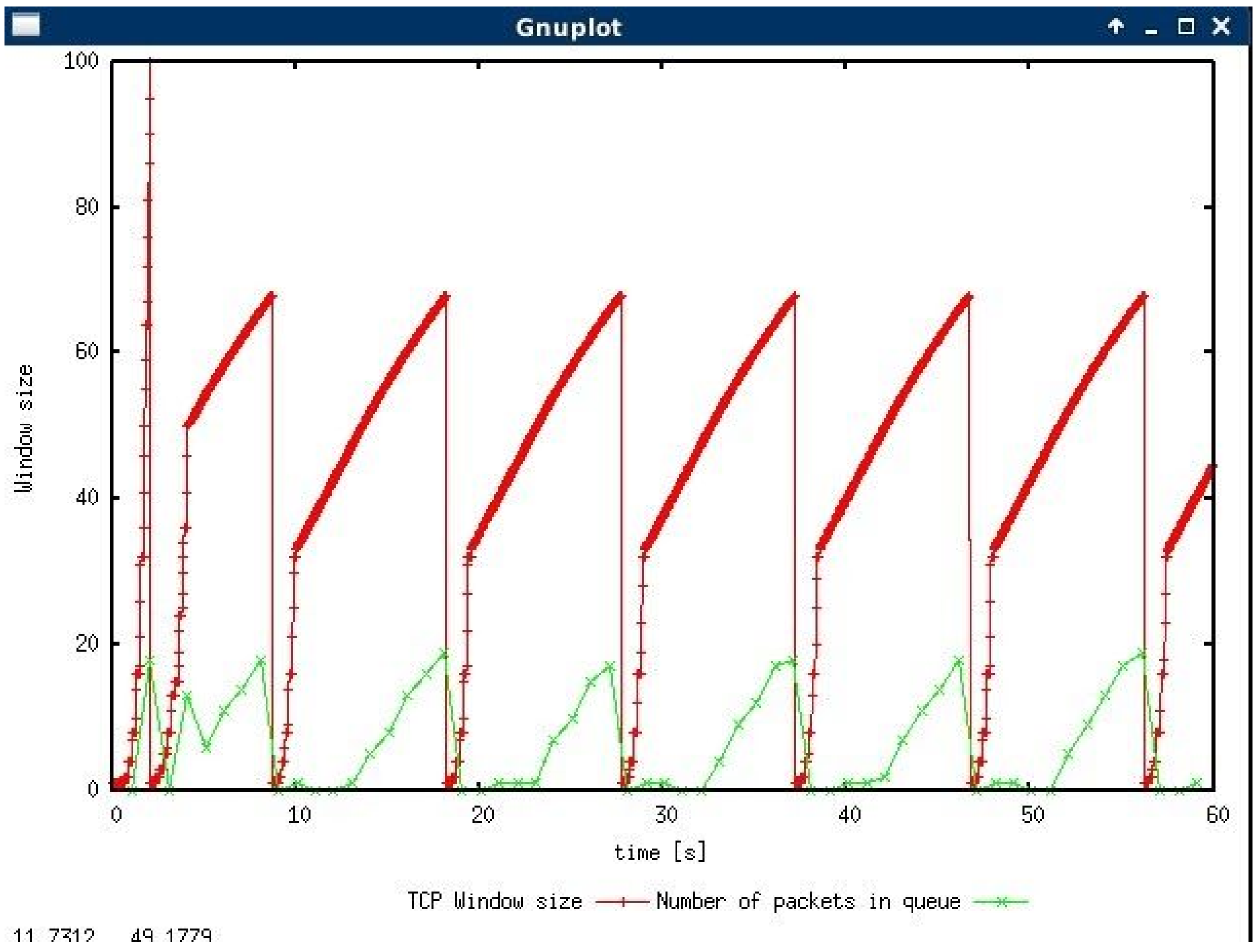
Lab Exercise 5: TCP Congestion Control and Fairness

By John Dao z5258962

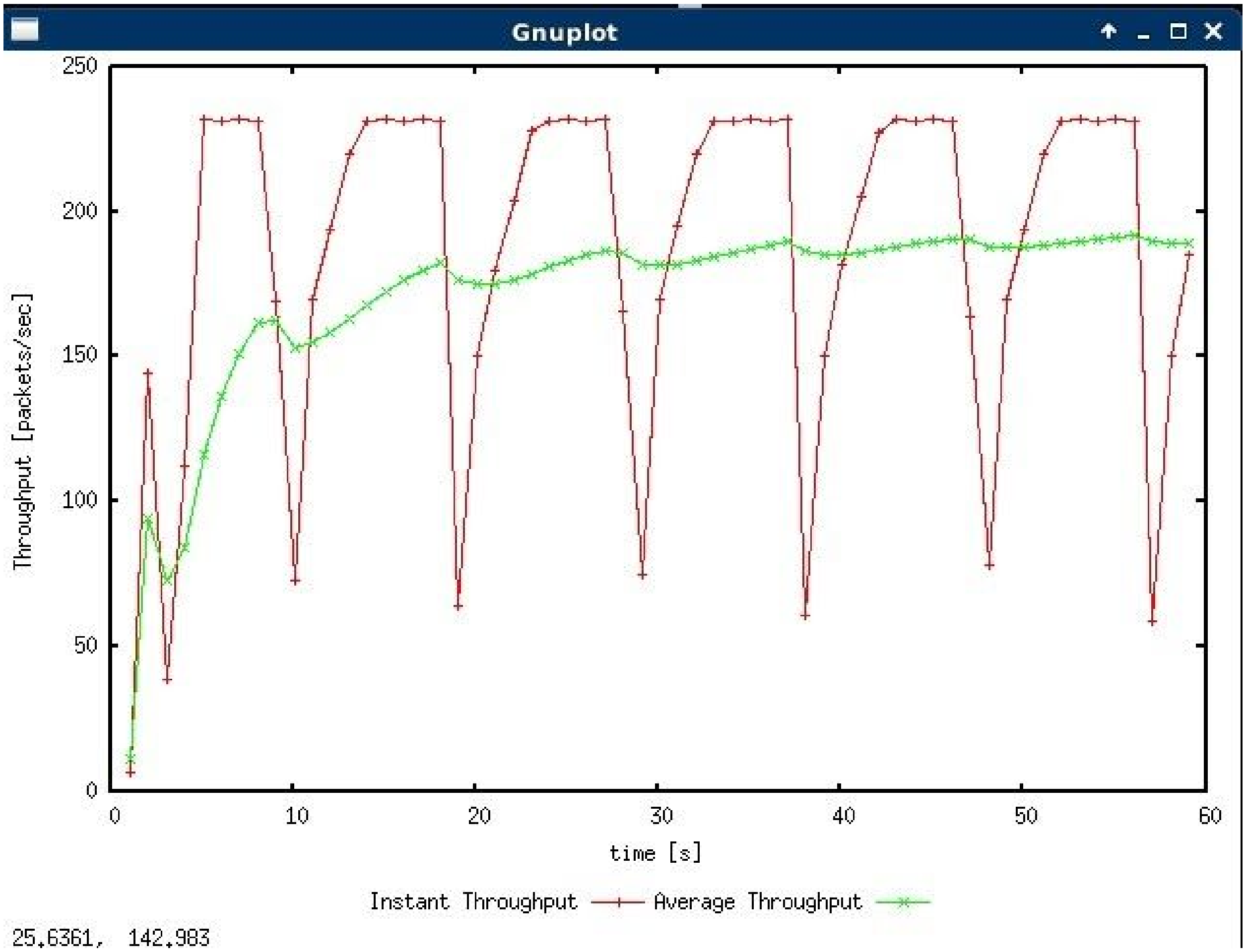
Exercise 1: Understanding TCP Congestion Control using ns-2

Question 1: What is the maximum size of the congestion window that the TCP flow reaches in this case? What does the TCP flow do when the congestion window reaches this value? Why? What happens next? *Include the graph in your submission report.*
Sample Output.



The maximum size of the congestion window is 100. When it reaches this value, the window is set to 1 and the window size is then halved and will then start another slow-start stage as shown in the graph after the window size hits 100.

Q2: From the simulation script we used, we know that the payload of the packet is 500 Bytes. Keep in mind that the size of the IP and TCP headers is 20 Bytes, each. Neglect any other headers. What is the average throughput of TCP in this case? (both in number of packets per second and bps)
Sample Output

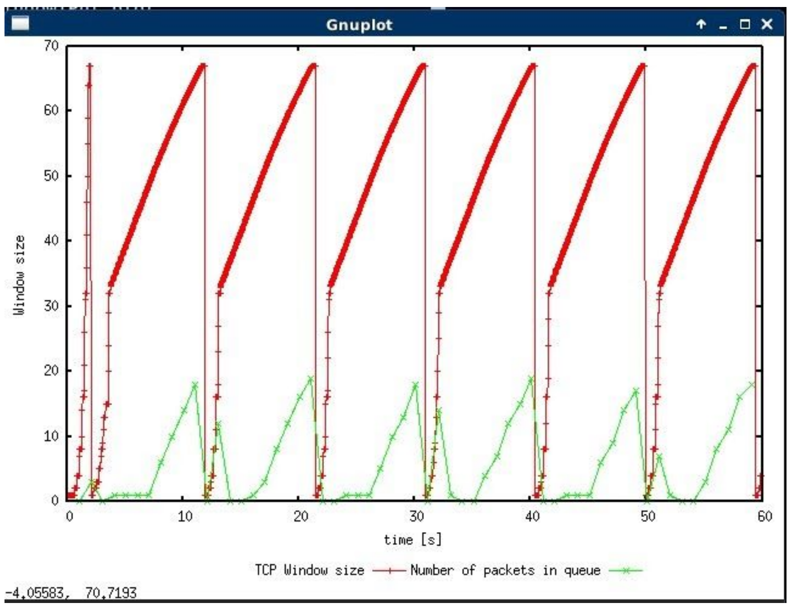


Via the graph, we can decipher that the average throughput is around 190 packets per second. Assuming that the size of the payload is 500 bytes and the IP and TCP headers is 20 Bytes, each.

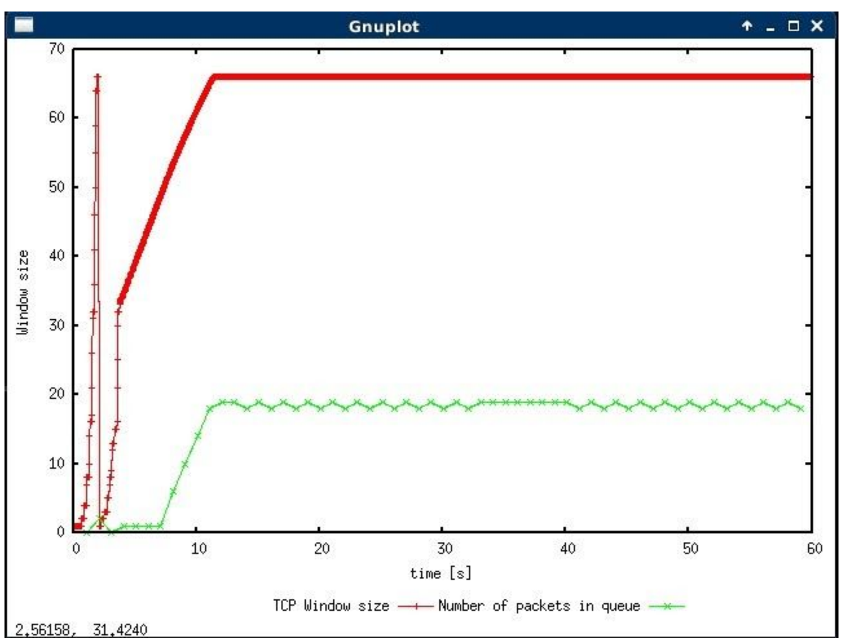
Throughput = amount of data (D) / transmission time (T).
In this case,
(packet only)
Throughput = packet size * throughput packets/sec = 500 * 190 = 95000bytes/s = 760kbps
(including headers)
Throughput = packet size * throughput packets/sec = (500 + 20 + 20) * 190 = 102600 bytes/s = 820.8kbps

Q3: Rerun the above script, each time with different values for the max congestion window size but the same RTT (i.e. 100ms). How does TCP respond to the variation of this parameter? Find the value of the maximum congestion window at which TCP stops oscillating (i.e., does not move up and down again) to reach a stable behaviour. What is the average throughput (in packets and bps) at this point? How does the actual average throughput compare to the link capacity (1Mbps)?

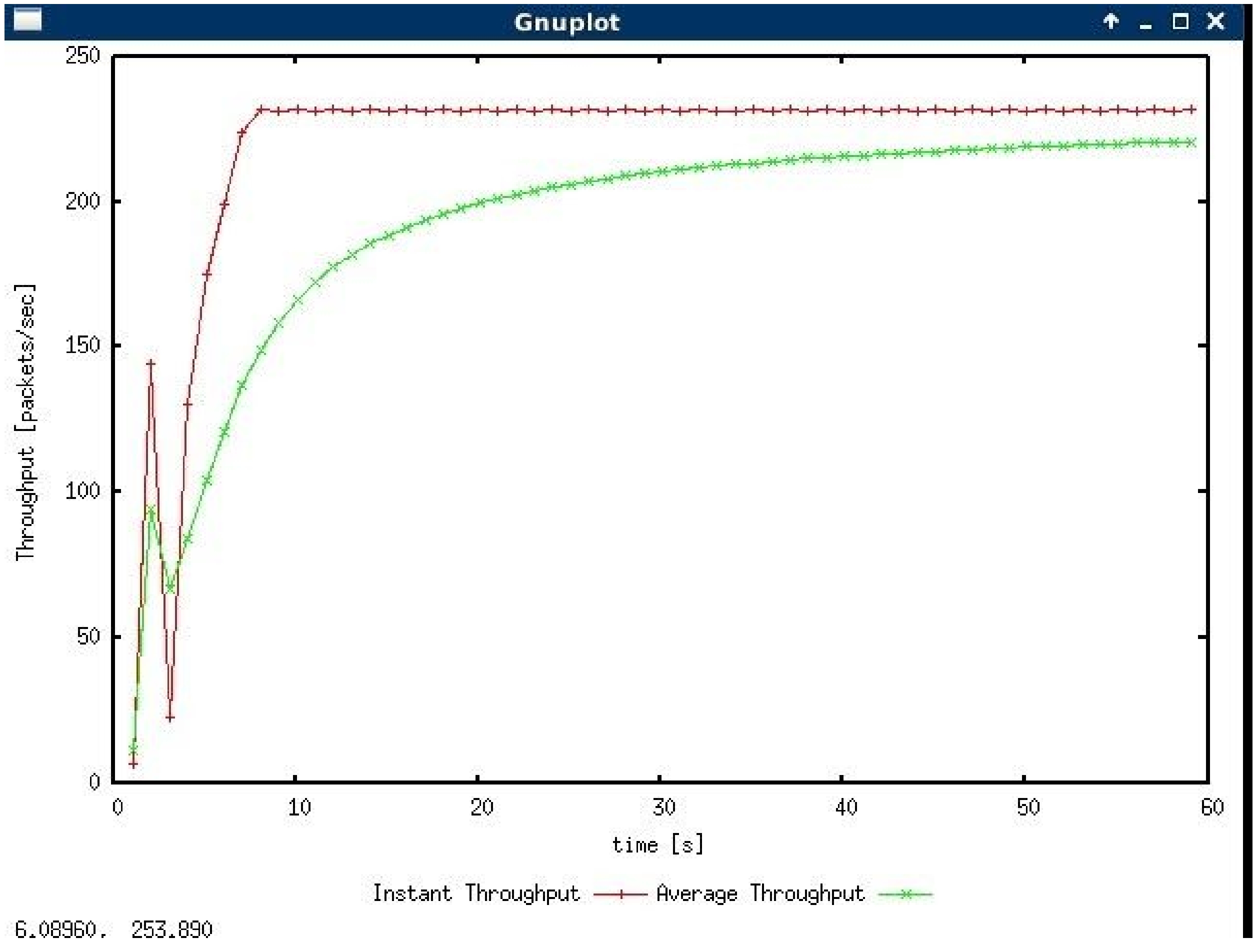
ns tpWindow.tcl 67 100ms



ns tpWindow.tcl 66 100ms



Maximum window size of 66.



Average Throughput at this point is at 220 with it leveling around 230. Eventual average throughput will trend more toward 230 over a longer amount of time.

Throughput = amount of data (D) / transmission time (T).

In this case,

(no headers)

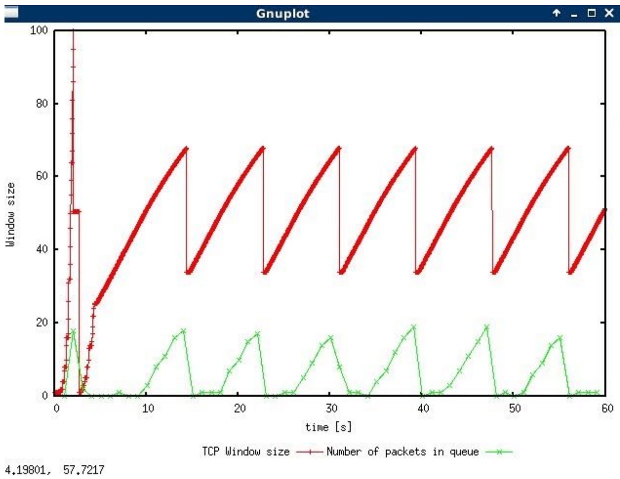
Throughput = packet size * throughput packets/sec = 500 * 220 = 110000bytes/sec = 0.8 mbps

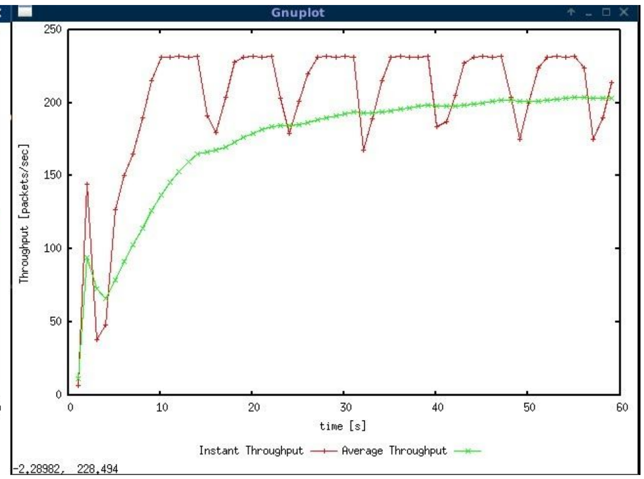
(headers)

Throughput = packet size * throughput packets/sec = 540 * 220 = 110000bytes/sec = 0.95 mbps

AVG throughput is almost identical to the link capacity of 1mbps

Q4: Repeat the steps outlined in Question 1 and 2 (NOT Question 3) but for TCP Reno. Compare the graphs for the two implementations and explain the differences. (Hint: compare the number of times the congestion window goes back to zero in each case). How does the average throughput differ in both implementations?

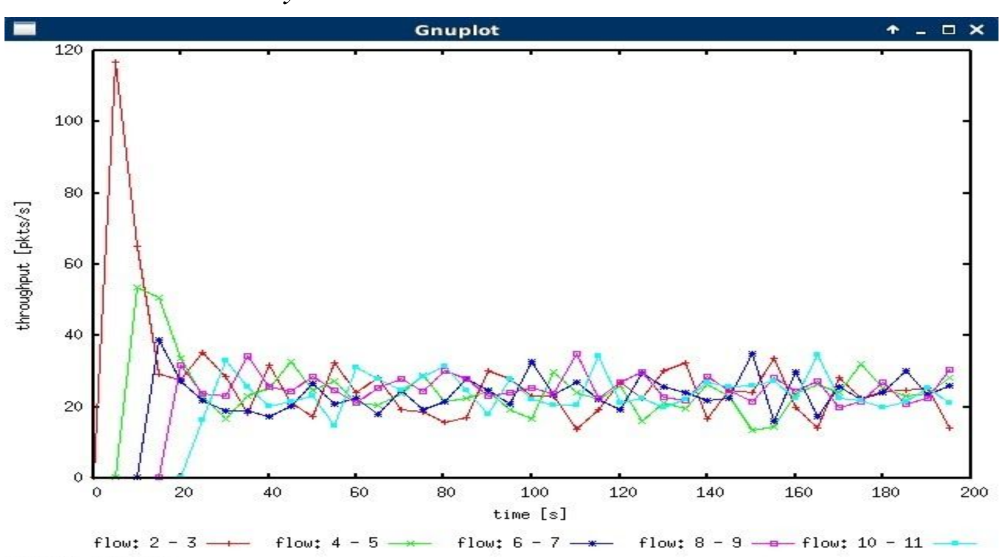




Reno window size only hits zero once during an event of congestion. Average throughput has smaller dips during window reduction as a result and thus has a higher throughput.

Exercise 2: Flow Fairness with TCP

Q1: Does each flow get an equal share of the capacity of the common link (i.e., is TCP fair) ? Explain which observations lead you to this conclusion.



Sample output

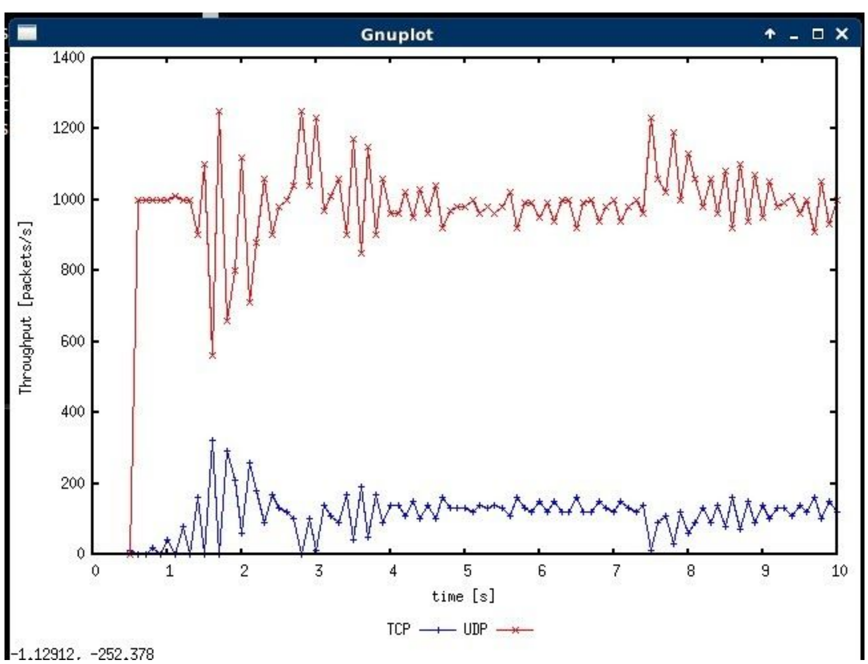
Yes, each flow gets an equal share of the capacity of the common link. This can be observed in the gnuplot. During the initial stages, where during t = 0 -> 20, throughput were relatively uneven and could be considered as unequal, however, as time progressed, especially from t = 40 and onward, throughput between all flows remained within close amounts of each other with no flow receiving an abnormally large throughput compared to the others. Thus we can consider that though initially unequal, the graph clearly shows how TCP is fair in sharing a common link and distributing it across multiple flows.

Q2. What happens to the throughput of the pre-existing TCP flows when a new flow is created? Explain the mechanisms of TCP which contribute to this behaviour. Argue about whether you consider this behaviour to be fair or unfair.

When a new flow is created, the relative throughput of the other flows will decrease to compensate for the introduction of said new flow. This can be dictated in the graph, where t 0 -> 20, as new flows are being introduced. When green is first introduced, the red flow throughput is reduced drastically from around 118 to 63. Similar events occur when blue, purple and cyan are introduced as all prior existing flows' throughputs are decreased to compensate.

Exercise 3: TCP competing with UDP

Q1: How do you expect the TCP flow and the UDP flow to behave if the capacity of the link is 5 Mbps ?



Sample output

UDP does not have congestion control but TCP does. This means that UDP will transmit at scheduled rates and TCP will transmit the rest.

Q2: Why does one flow achieve higher throughput than the other? Try to explain what mechanisms force the two flows to stabilise to the observed throughput.

Since UDP does not have congestion control, throughput must be kept at a constant rate and in this situation, is kept at a high rate due to the window size in a good network environment. This means, due to UDP's nature of not caring about packet loss, it can operate at higher rates without major effect in packets sent/received and thus, TCP throughput rate is kept low.

In the case that the network conditions are poor/unstable, TCP may be preferred and the comparison between the two may be different due to the fact (with TCP being favoured due to possible packet loss)

Q3: List the advantages and the disadvantages of using UDP instead of TCP for a file transfer, when our connection has to compete with other flows for the same link. What would happen if everybody started using UDP instead of TCP for that same reason?

<u>UDP</u>

Pros

- High rate of transmission
- Less packet and header size requirements
- Quicker transmission times as a result

Cons

- Increased packet loss in poor network conditions
- No congestion control can lead to long transmission times.
- Increased File Corruption as a result.

(compared to TCP)

If everyone started utilising UDP instead of TCP in the case where there are many competing flows, we would have a circumstance where a network is overwhelmed. This is due to the pack that UDP has NO congestion control resulting in many of the above cons and thus eliminating reliability within the network. As congestion increases, the network will become increasingly worse with requests actually transmitted (or packets sent) increasing in rate of corruption or received in states of unusability. Thus, it would be detrimental to the user experience when using the network that relies on UDP in events of congestion or competition between flows.