ACN
ASSIGNMENT 01

---

1. **Why do you see a difference in webpage fetch times with short and large router buffers?**

   Answer : Routers with larger buffers can store more data packets and can help absorb short bursts of data and reduce the likelihood of congestion-related packet drops.
   On the other hand, routers with smaller buffers might drop packets more quickly when congestion occurs. This can lead to retransmissions, which would increase the overall time it takes to fetch a webpage. But larger buffers can also introduce additional latency, known as queuing delay, because packets spend more time waiting in the buffer before being processed.
   Thus while large buffers can help in absorbing traffic spikes, it might also lead to increased latency for individual packets, which can accumulate and impact the webpage fetch time, this problem is called "bufferbloat," where packets experience significant delays within the buffer which can lead to inconsistent latency and overall slower network performance.
   With larger buffers, TCP might interpret the extra buffer space as a sign of a less congested network, causing it to send more data before acknowledging the receipt of previously sent data. This can lead to inefficient utilization of the network and increased latency.

2. **Bufferbloat can occur in other places such as your network interface card (NIC). Check the output of ifconfig eth0 on your VirtualBox VM. What is the (maximum) transmit queue length on the network interface reported by ifconfig? For this queue size, if you assume the queue drains at 100Mb/s, what is the maximum time a packet might wait in the queue before it leaves the NIC?**

   Answer: The “txqueuelen” parameter reflects the maximum queue size of the network interface indicating the maximum number of packets that can be queued for transmission at any given time.
   When running the "ifconfig" command and we observe the "txqueuelen" value is 1000, it signifies that the network interface's transmit queue can accommodate up to 1000 packets, positioning them for subsequent delivery.
   Given that we are dealing with an Ethernet interface, where the maximum packet size is typically 1500 bytes, and the transmit queue length is 1000 packets. So total data that can be in the queue = 1500*1000 bytes.
   Given your transmission rate of 100 Mb/s (megabits per second), the calculation proceeds as follows:

   Time = (Total queue capacity in bits) / Transmission Rate

Time = (1,500,000 bytes × 8 bits/byte) / (100 × 10^6 bits/s)
Time = (12,000,000 bits) / (100 × 10^6 bits/s)
Time = 0.12 seconds

3. **How does the RTT reported by ping vary with the queue size? Describe the relation between the two.**

   Answer: As RTT can be defined as the time taken for propagation delay + queuing delay and in our scenario propagation delay is fixed to 6ms thus RTT is proportional to queuing delay(queue size * per packet transmission delay). Thus as the queue size is increased the RTT is increased proportionally.

4. **Identify and describe two ways to mitigate the bufferbloat problem.**

   Solution:
   Bufferbloat is a phenomenon that occurs in computer networks when excessive buffering of data packets causes increased latency and poor network performance. It arises from the well-intentioned but misguided attempt to prevent packet loss during times of network congestion by using large buffers in networking devices such as routers and switches.
   Mitigating bufferbloat involves implementing strategies to prevent or minimize the negative effects of excessive buffering in network devices
   1. Queue size management: Which involves carefully configuring and managing the sizes of buffers in networking devices to prevent excessive buffering and reduce latency. Properly adjusting queue sizes can help balance the need to prevent packet loss during congestion while avoiding the negative effects of bufferbloat.
   2. Active Queue Management (AQM) Algorithms: AQM algorithms actively monitor and manage the length of queues in network devices to prevent them from becoming overly full during congestion. These algorithms help maintain lower and more consistent latency by controlling the delay introduced by buffering. Example: RED (Random Early Discard): RED probabilistically drops packets before the queue becomes too full. By dropping packets preemptively, RED helps signal congestion to sending devices and prevents the buffer from becoming excessively congested.

5. **Describe how and why your results change when you re-run the emulation.**

   Solution:
   1. Dynamic Nature of Traffic Patterns: The variability of traffic patterns is a primary contributor to changes in emulation results. Network traffic is rarely constant, and even subtle shifts in the distribution and timing of incoming packets can lead to different queuing behaviors and latency outcomes.

2. Packet Sizes: Variations in packet sizes can impact queuing behavior. Smaller packets are often processed more quickly and can experience lower queuing delays, while larger packets might contribute to buffer congestion, leading to higher delays and increased potential for packet drops.
3. Background Activities: The presence of other applications and devices sharing the network can create fluctuating levels of load and congestion. These background activities can introduce unpredictability in network performance, affecting the accuracy and consistency of emulation results.