IT304
LAB 7 Analyzing congestion policy, RTT of TCP and working of UDP using NetSim and wireshark.

1 Introduction of Congestion Policy of TCP

A key component of TCP is its congestion-control mechanism. TCP sender perceives that there is little congestion on the path between itself and the destination, then the TCP sender increases its sending rate; if the sender perceives that there is congestion along the path, then the sender reduces its sending rate. But this approach raises three questions. First, how does a TCP sender limit the rate at which it sends traffic into its connection? Second, how does a TCP sender perceive that there is congestion in the network? And third, what algorithm should the sender use to change it's sending rate? The answer is Congestion control policy. The TCP congestion-control mechanism operating at the sender keeps track of an additional variable, the congestion window. The congestion window, denoted cwnd, imposes a constraint on the rate at which a TCP sender can send traffic into the network. Specifically, the amount of unacknowledged data at a sender may not exceed the minimum of cwnd and rwnd.

The TCP Congestion control Algorithm has two major component: slow start, congestion avoidance. Slow start and congestion avoidance are mandatory components of TCP, differing in how they increase the size of cwnd in response to received ACKs.

1. Slow start phase: In this phase, tcp double the cwnd as ack receives up to threshold point. After that It will enter in to congestion avoidance phase where it changed the window in to additive increse mode, where it will increase cwnd by 1 with 1 ackwoledge packet.

2. Congestion avoidance: In this phase two events happens. 1)three duplicate ack 2)Time out. If three duplicate ACKs are received, tcp enters in fast retransmit mode and divide congestion window size by half (instead of adding it by 1), setting the slow start threshold equal to the congestion window, and enter a phase called fast recovery. When time out happens after packet loss, tcp changed congestion window size to 1mss. and enters in fast recovery mode. In fast recovery mode, tCP stores the sequence number of the highest data packet by sending the lost packet.

1.1 Experiment

Perform the following experiment to understand congestion policy of different variants in TCP.

1. Configure scenario as shown in figure1 and implement 2 application for 2 topology.

2. For both nodes C and D, go to properties: TRANSPORT_LAYER and set Congestion Control Algorithm to tahoe. Other Node, Link, and application properties will be same as 2. Enable (Online) Wireshark for the destination node D.

3. For both nodes E and F, go to properties: TRANSPORT_LAYER and set Congestion Control Algorithm to RENO and change according to figure in 3. Enable (Online) Wireshark for the destination Node F.

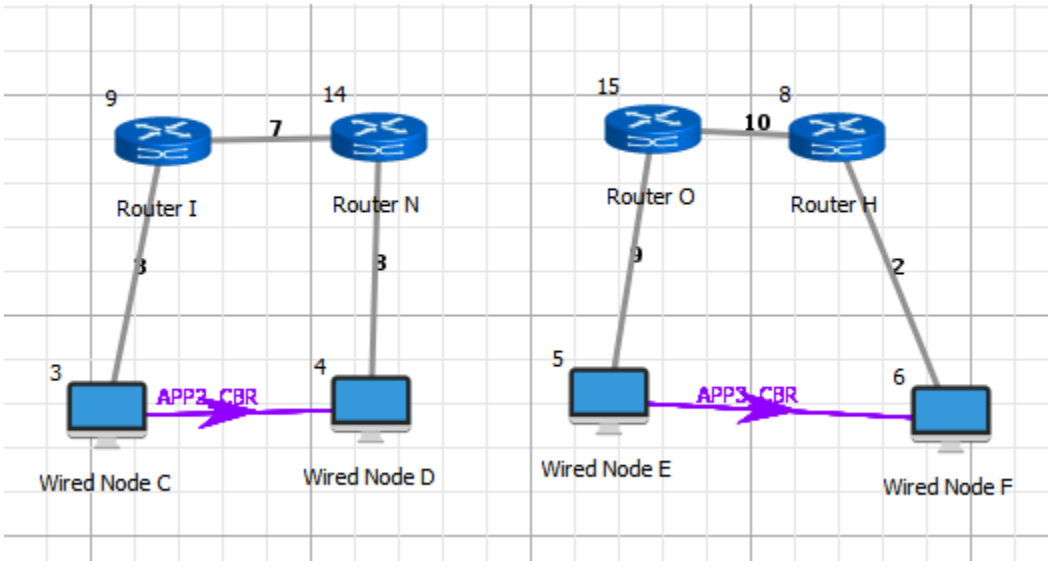

Figure 1: Topology of Congestion policy experiment

Maximum Segment Size	1460 bytes
Node-Router Propagation Delay (both)	5 Micro Sec
Router-Router Propagation Delay	100 Micro Sec
Node-Router Link Speed (both)	20 Mbps
Router-Router Link Speed	100 Mbps
Bit Error Rate (All Links)	10 e(-8)
Application Packet Inter Arrival Time	400 Micro Sec

Figure 2: Application 1 configuration

4. Run the simulation for 30 seconds (packet animation is not required.)
5. Wait for successful termination of simulation and observe 2 wireshark window that opened.
6. For each wireshark, select any row of tcp packet, click on the statistics $--> $ tcp stream graph $--> $ window scaling.
7. Observe the graph and answer the questions.

1.2 Exercise

1. For both the variant, analyze graph of congestion window, answer the following by marking in the graph.
   (a) Identify the event of TCP slow start.
   (b) Identify the event of packet loss and time out.
   (c) Identify the intervals of time when TCP congestion avoidance is operating.
2. What is the difference in congestion control policy of Tahoe and Reno, with respect to congestion avoidance and two events of congestion avoidance phase. Explain briefly in your log book.

Maximum Segment Size	1460 bytes
Node-Router Propagation Delay (both)	5 Micro Sec
Router-Router Propagation Delay	100 Micro Sec
Node-Router Link Speed (both)	20 Mbps
Router-Router Link Speed	100 Mbps
Bit Error Rate (All Links)	10 e(-7)
Application Packet Inter Arrival Time	400 Micro Sec

Figure 3: Application 3 configuration

2 Analyzing fairness of TCP.

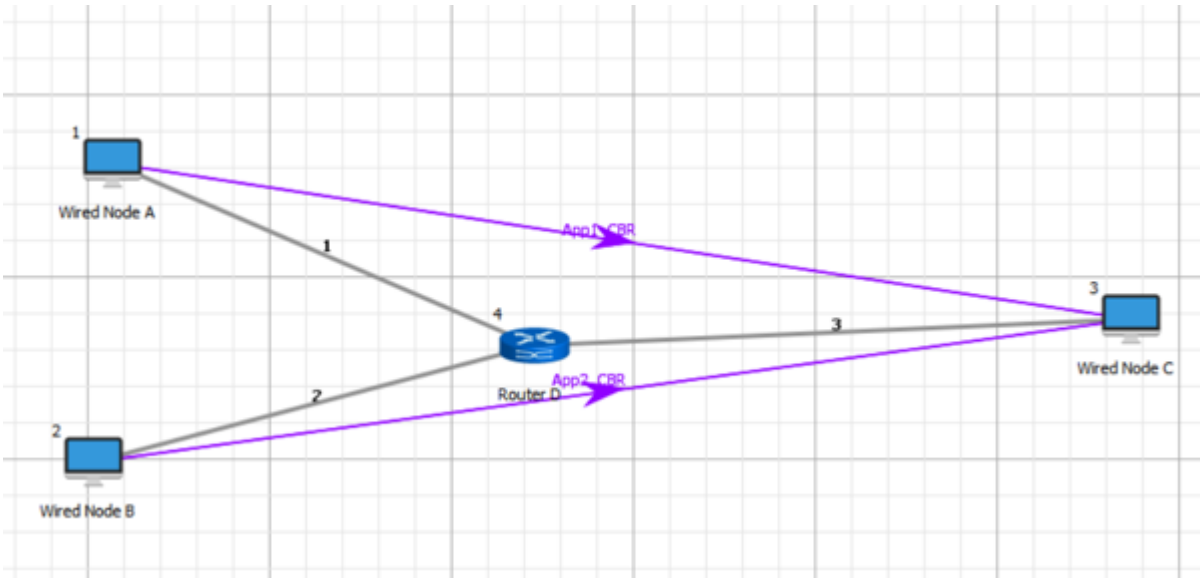

Figure 4: Topology to analyze fairness of TCP

2.1 Experiment

1. Take 3 wired nodes and one router, configure 2 identical CBR applications with default app specification between them as shown in the figure4.
2. Keep link properties as default.
3. Run simulation for 5 seconds.
4. Check throughput for both the applications and write down your observation.

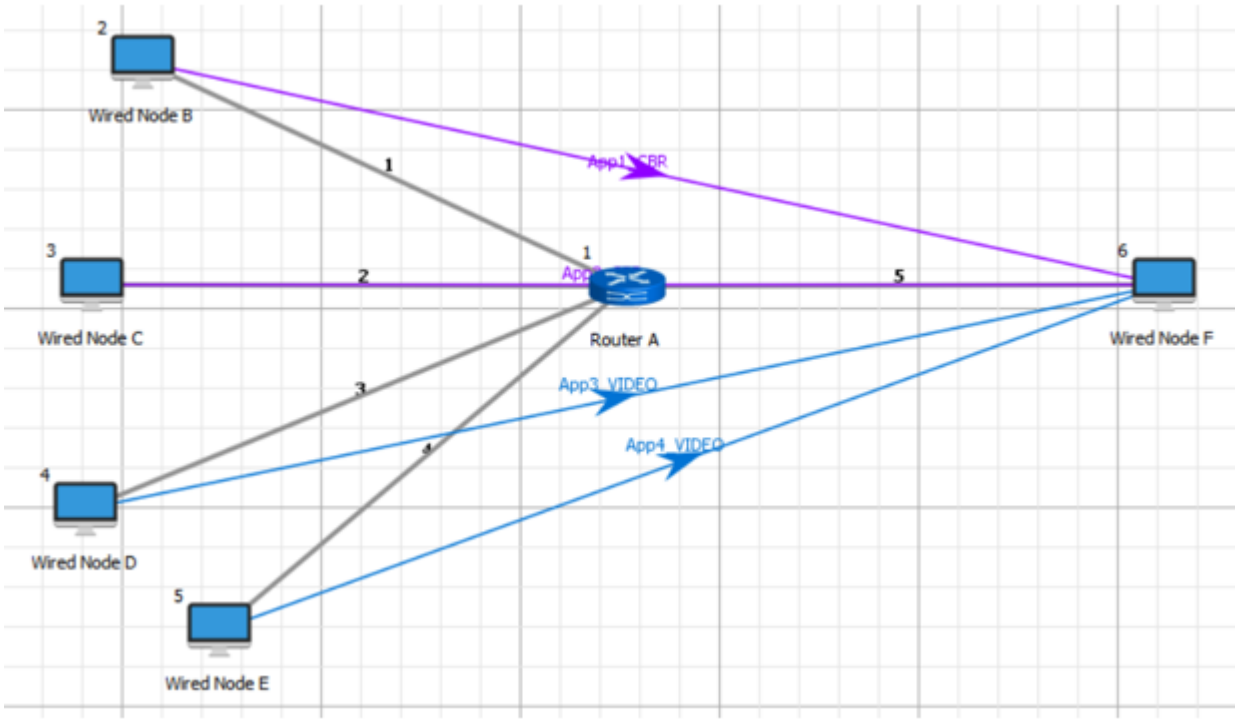

Figure 5: Caption

3 Analyzing throughput.

3.1 Experiment

1. Configure a new network as shown in the figure 5 with 4 wired node, 1 router.
2. Configure two CBR application between node B and F and between C and F with packet size of 500 bytes.
3. Configure two video application between node D and node F and between node E and F with Frame per second 50.
4. Keep all properties of all nodes, router and links as default values.
5. Run the simulation for 10 second.

3.2 Exercise

1. Calculate and Observe average throughput of both the applications (CBR and VIDEO).
2. Observe the delay and throughput metrics in the simulation window and write down your observation.

4 Analysing RTT of TCP using wireshark.

4.1 Exercise:

Answer the following questions, by opening the Wireshark captured packet file tcp-ethereal-trace-1 in http://gaia.cs.umass labs/wireshark-traces.zip (Once you have downloaded the trace, you can load it into Wireshark and view the trace using the File pull down menu, choosing Open, and then selecting the tcp-ethereal-trace-1 trace file.)

4.2 Questions:

1. Consider the TCP segment containing the HTTP POST as the first segment in the TCP connection. What are the sequence numbers of the first six segments in the TCP connection (including the segment containing the HTTP POST)? At what time was each segment sent? When was the ACK for each segment received? Given the difference between when each TCP segment was sent, and when its acknowledgement was received, what is the RTT value for each of the six segments? What is the EstimatedRTT value after the receipt of each ACK?
   Note: Wireshark has a nice feature that allows you to plot the RTT for each of the TCP segments sent. Select a TCP segment in the listing of captured packets window that is being sent from the client to the gaia.cs.umass.edu server. Then select: Statistics− >TCP Stream Graph− >Round Trip Time Graph.

2. What is the length of each of the first six TCP segments?

3. What is the minimum amount of available buffer space advertised at the received for the entire trace? Does the lack of receiver buffer space ever throttle the sender?

4. Are there any retransmitted segments in the trace file? What did you check for (in the trace) in order to answer this question?

5. How much data does the receiver typically acknowledge in an ACK? Can you identify cases where the receiver is ACKing every other received segment.

6. What is the throughput (bytes transferred per unit time) for the TCP connection? Explain how you calculated this value.

5 Analysing UDP protocol using wireshark

5.1 Exercise

1. Start a Wireshark capture.

2. Open a command prompt.

3. Type ipconfig /flushdns and press Enter to clear your DNS name cache.

4. Type nslookup 8.8.8.8 and press Enter to look up the hostname for IP address 8.8.8.8.

5. Close the command prompt.

6. Stop the Wireshark capture.

5.2 Questions

1. Select one UDP packet from your trace. From this packet, determine how many fields there are in the UDP header. (You shouldnt look in the textbook! Answer these questions directly from what you observe in the packet trace.) Name these fields.

2. By consulting the displayed information in Wiresharks packet content field for this packet, determine the length (in bytes) of the UDP header fields.

3. The value in the Length field is the length of what? (You can consult the text for this answer). Verify your claim with your captured UDP packet.

4. What is the maximum number of bytes that can be included in a UDP payload? (Hint: the answer to this question can be determined by your answer to 2. above)

5. What is the largest possible source port number? (Hint: see the hint in 4.)

6. What is the protocol number for UDP? Give your answer in both hexadecimal and decimal notation. To answer this question, youll need to look into the Protocol field of the IP datagram containing this UDP segment.

7. Why we have used DNS commands to capture UDP packets? Do you know any-other method to generate UDP traffic using wireshark? Write your answer in detail.