Lab 6
Transport Layer – TCP Congestion Control
CMPE 150

Lab Report

- Reports must be written and submitted individually as PDFs.

Submission Instructions:

Submit your report on the eCommons by 11:55 PM on the **day of your registered lab section**. Your submission should include an archive (tar or zip) containing your lab report and any other required files. Late/improper submissions will have a 10% grading penalty per day. Late labs will not be accepted after 3 days.

Pre-Lab

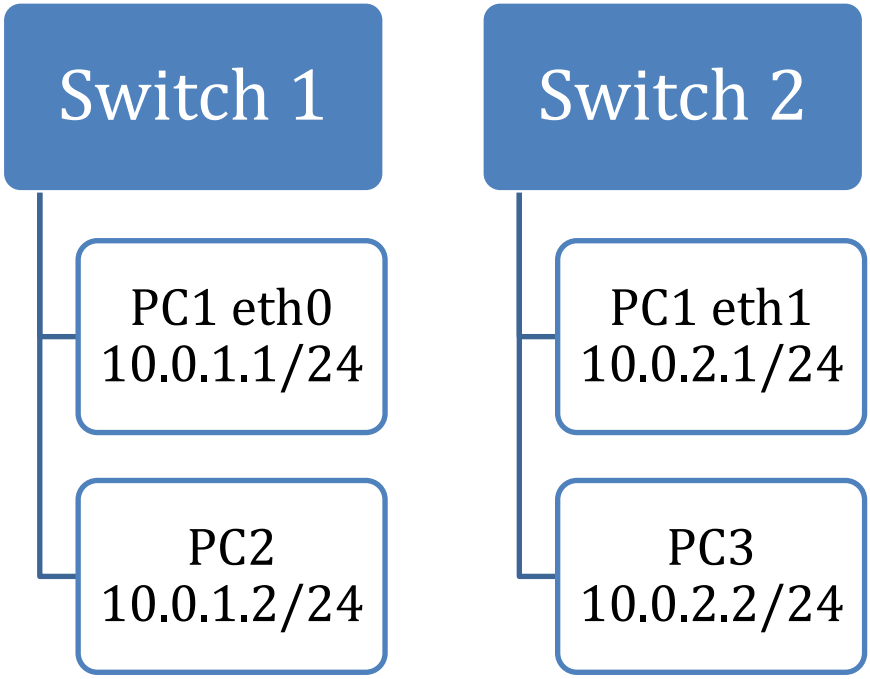
Read about *netem* -
http://www.linuxfoundation.org/collaborate/workgroups/networking/netem

1. Answer the following questions on TCP flow control and congestion control:
    a. Describe the sliding window flow control mechanism used in TCP.
    b. Describe the concepts of slow start and congestion avoidance in TCP.
    c. Explain the concept of fast retransmit and fast recovery in TCP.

Topology Setup

Today we will be using PC1 as a "router" to forward messages between PC2 and PC3.

Connect the following:



On PC1:

ifconfig eth0 10.0.1.1/24
ifconfig eth1 10.0.2.1/24
echo 1 > /proc/sys/net/ipv4/ip_forward (enables packet forwarding)
tc qdisc add dev eth0 root netem delay 1000ms (adds artificial delay)

On PC2:

ifconfig eth0 10.0.1.2/24
ip route add default via 10.0.1.1

On PC3:

ifconfig eth0 10.0.2.2/24
ip route add default via 10.0.2.1

Part A – Slow Start and Congestion Avoidance

We will first begin by observing how tcp traffic flows in a network without any loss.

PC3:

ttcp –rs –l1000 –n1000 –p4444

PC2:

Begin wireshark and set display filter to TCP traffic.
ttcp –ts –l1000 –n1000 –p4444 –D 10.0.2.2

After the application is finished, stop the trace in wireshark.
Select *Statistics-> TCP Stream Graph.* Look at each of the graph types available.
Save each graph by maximizing the window then going to *Applications->Accessories->Take Screen Shot.*

SAVE DATA – Save the wireshark trace and image graphs.

Part B – Packet Loss Effect on TCP Congestion Window

We will now observe how loss affects the TCP congestion window.
PC1:

tc qdisc add dev eth1 root netem loss 5% 25%

(This will cause ~5% of packets to be lost, and each successive probability depends by a quarter on the last one.)
PC3:

ttcp –rs –l1000 –n500 –p4444

PC2:

Begin wireshark and set display filter to TCP traffic.
ttcp –ts –l1000 –n500 –p4444 –D 10.0.2.2

PC1:

After ~300 packets have been observed in Wireshark,
tc qdisc change dev eth1 root netem loss 100%
After ~10 seconds,
tc qdisc change dev eth1 root netem loss 5% 25%

After the application is finished, stop the trace in wireshark.
Select *Statistics-> TCP Stream Analysis.* Look at each of the graph types available.
Save each graph by maximizing the window then going to *Applications->Accessories->Take Screen Shot.*
SAVE DATA – Save the wireshark trace and image graphs.

PLEASE SHUTDOWN AND REMOVE CABLES WHEN FINISHED

Questions

Part A

1. Attach the trace file and images to the report.
2. On the time-sequence graph (tcptrace), identify the regions of slow-start and congestion avoidance.
3. Estimate the size of *ssthresh* (slow start threshold).
4. Estimate the largest observed congestion window size.

Part B

1. Attach the trace file and images to the report.
2. On the time-sequence graph, identify the region where the connection was severed and show identify areas where slow start and congestion avoidance was used.
3. Did you observe any occurrences of TCP "fast recovery"? If so, did it have an observable effect on the congestion window?
4. Estimate the sizes of *ssthresh.* Did they change throughout the trace?
5. Estimate the largest observed congestion window.
6. How did the throughput compared to part A? You may use the throughput graph to support your answer.
7. As far as error recovery, were selective acknowledgement (SACK) used? How can you tell?

General Troubleshooting Tips

If you encounter an error in lab, verify that your machines are configured as they should be (if we are using DNS, then make sure to ping the IP so we can determine wether the issue is DNS or topology related). Check the switch's activity LEDs to verify that a port is transmitting, if the port is not active, then the connection is invalid.

For invalid connections check for:

1. PC is on and eth0/1 is up and configured to the correct IP.

2. Router is on and eth0/1 are configured.

3. Make sure that the cables are secure, pull lightly to verify that they are in place.

If you have a valid connection but still can not ping (and they are all connected on the switch), there may be a problem with the vlan configuration.

1. Either unplug and plug back in the switch.

2. Use the console port on the back of the switch, and use the following commands "S1 > enable", "S1# show vlans", if this command shows a 1 vlan with most of the ports on it, then your switch is configured properly, problem lies in PC/Router IP configure, verify that they are all on the same subnet (/24 generally). Next "S1# configure terminal", then depending on the number of ports you are using in what rang, "S1(conf-t)# interface range Fa0/X - Y" (there is a space between X and Y, where X is the starting port and Y is the ending port), "S1 (inter-range)#switch access vlan 1", places all of the interfaces in that range on the same vlan so that they can communication. Now "end" to make sure you are back at "S1#" and type "S1# show vlan" to verify the correct topology.

If you are using a pod that (lets say for the DNS lab) requires 4 routers, but you only have 3 routers, you can use Fa0/1 on one of the routers to act as another router. Make sure to type "R1# no ip routing" to remove any default routing the router may be carrying out between the two interfaces.

If you have any lab questions related to the hardware feel free to email or post on the forums for help outside of lab.