CS252 Lab-2

Network Diagnostic Tools

In this lab, we will try out some network diagnostic tools on Unix/Linux. Most of these tools will work on an Apple MAC machine (in case some information for the lab report is not easy to find on MAC osX, then just state that in the report). You may have to install some of the tools on your machine if they are not already installed by default (Google for installation instructions). If you have trouble installing, just post to the MS Team for CS252 and your friends and TAs will help you out.

(i) `ifconfig :` Type "ifconfig" in any terminal window. It will show you a lot of information for each network interface in your machine. You might see a "loopback interface" such as "`lo :`" which is only a virtual interface (not a real network interface), and we will ignore the loopback for now. Identify which of the other interfaces corresponds to either Wired Ethernet or WiFi.

In your report (5 marks), for this interface (either one of Ethernet or WiFi) state what is the IPv4 address (and IPv6 address if any), the hardware address (MAC), and the MTU. You can find out more about ifconfig online
(e.g. http://www.aboutlinux.info/2006/11/ifconfig-dissected-and-demystified.html). State the number of bits used for IPv4 addresses, IPv6 addresses, and hardware address. State what MTU stands for and the units it (MTU) is expressed in. We talked about queues in one of the first lectures. State the transmit queue length on this interface, and give the units this is measured in. (Each team member can give this information about his/her own machine in the report). All team members can give this information from their machines in the report.

(ii) `traceroute <dest host IP or name>:` This tool tries to find out all the IP addresses of layer-3 nodes (actually layer-3 interfaces; remember each router can have many interfaces) on the path from the machine running the traceroute command to the destination host. It uses a field called "TTL" (time-to-live) in the IP header cleverly to do this. TTL is decremented at each layer-3 router on the path, and if it reaches 0, the router drops the packet and send an "error" message to the source node using the ICMP protocol. The source IP address in the ICMP packets reveals the IP address of the interface of the intermediate router. By sending TTL of different values we can thus find out the IP addresses we are after. But not all routers may respond with error messages (this depends on how they are configured). You can read up more about traceroute online.

You can be detectives by trying out traceroute from your own machines to different websites on the Internet. You can also use https://www.uptrends.com/tools/traceroute to run traceroute starting from machines in many cities around the world to any destination.

In your report (15 marks), mention the number of hops (routers) on the paths from your own machines and 5 cities using the URL give above (the cities are of your choice, preferably in different continents) to destinations (i) www.google.com, (ii) www.cnn.com, (iii) www.iitd.ac.in

(note: I have chosen “iitd” rather than “iitb” deliberately; you can try both and see what you find). For each case also state the average round trip time (RTT) to the destination. Also, state if the destination IP address is the same for the same URL, no matter where the source of the traceroute is. In case the IP address is different, do some detective work online to find out why this might be the case and state briefly (in a paragraph or two) what you find out. In case a firewall prevents traceroute from working on your own machines, just mention that in your report.

(iii) `ping <dest host>`: This command sends ICMP packets to the destination and gets back ICMP echo packets. It reports the RTTs observed. A quick way to find if is host is alive. It is possible that a host is configured not to reply to ping packets, so there is no guarantee that the host is dead if you don’t get a reply.

In your report (5 marks), give the average RTT obtained by pinging servers in different continents (hint: government servers are likely to be geographically located in their own countries) from each of your own machines and also using the ping utility at https://ping.eu/ . In case you are behind a firewall, it is possible that the firewall blocks your ping packets, in which case just report that ping does not work from your machine. Comment on what you observe about the RTTs (For example, are pings to servers more if the server is geographically further away?)

(iv) `Iperf` : This is a classic network bandwidth estimation tool used by network researchers. You can use “man iperf” to find out various options on how to use it. Iperf runs between two machines, an iperf server (on which “iperf -s” is run) and an iperf client (on which “iperf -c <server IP or name> <options>” is run). The client sends messages to the server to either start a TCP connection or a UDP stream. TCP does retransmission of packet losses and congestion control, whereas UDP does not do either. Iperf reports the throughput obtained when data is sent from the client to the server, as well as some other information. The iperf server needs to have a public IP address so that it is reachable by a client outside its own network (Note that IP addresses that begin with 10.*.*.* or 192.*.*.* are “private IP addresses” which we briefly discussed in a Live Session). The client need not have a public IP address. (NOTE: iperf will consume data, so don’t run it for too long, else it may use up your Data plan).

For TCP:
(at server) iperf -s
(at client) iperf -c <server IP or server name>
Note: You may have to use CTRL-C at the client machine to stop the iperf experiment

For UDP:
(at server) iperf -s
(at client) iperf -c <server IP or server name> -u -b <UDP data rate> (e.g. -b 10M gives 10Mbps)

Note that there are iperf servers available around the world (see https://iperf.fr for example, from which you can download “Iperf3” code. If you install and run “iperf3” then you will have to replace “iperf” with “iperf3” in the commands given above).

In your report (15 marks), give the TCP bandwidth rates (in Mbps) obtained from at least 2 iperf servers on the Internet. For UDP, run the -b <UDP rate> option for UDP with rates starting with “1M” and then increasing successively by a factor of 2 (i.e. “2M”, “4M”, “8M”) until the observed throughput starts to be less than that the specified UDP rate. Call this last data rate “X”. This means that the network is unable to support the specified data rate X. Compare this UDP rate with the obtained TCP rate. Are they similar, or is one much larger than the other?

Bonus (3 marks): You can try running iperf between your own machines (different team members who are geographically apart). Note that iperf is available on Android as well. You will have to host the server on a machine with a public IP address. You can give results for TCP and UDP in your report.