Computer Networks Theory + Lab

Summary

Contents

1. Lab Instructions (Document)
2. Action at a Host (Quiz)

« Previous (Lab01: Packet Sniffers) Next (Lab03: Introduction to Socket Programming) »

Lab Instructions

Lab02: Internet Protocol Stack and Metrics

Objective:

1. Get acquainted with some commonly used networking commands and TCP/IP diagnostic tools
2. Dig deeper to understand the concept of layering, encapsulation and multiplexing.
3. Understand how some commonly used metrics like throughput, delay/latency are calculated.

Reference Material:

1. Internet Protocol Stack section videos
2. Goals and Metrics section videos
3. Packet Sniffers and Action at a Host lab videos
4. Wireshark: a. Wireshark Website: http://www.wireshark.org b. Wireshark Documentation: http://www.wireshark.org/docs/ c. Wireshark Wiki:http://wiki.wireshark.org
5. Command “ip” cheat sheet: https://access.redhat.com/sites/default/files/attachments/rh_ip_command_cheatsheet_1214_jcs_print.pdf
6. Docker-setup-Windows: https://docs.google.com/document/d/1IFZyYPkjADFB0kSKzPh0DCvJb9UxQL8vwo9lMH2F6Ow/edit?usp=sharing
7. Docker-setup-MAC: https://docs.google.com/document/d/19-EESZzwKhXxDGCBpM5qzz0EMPgG5Ut8eoZzUKCKvM0/edit?usp=sharing

Requirements:

1. Linux Command Terminal
2. Wireshark
3. Files: sample-cse-iitb.pcapng, types-of-traffic.pcapng , metrics.pcapng, wget.png, ping.png (You can download these files by clicking on the “View Documents” button at the bottom. You can unzip the contents at a terminal via “unzip file-name.zip”)

Instructions:

1. Given the very large class size, we need to automate grading to some extent. To automate grading, we will provide some traces and ask questions based on it. The marks you get here will count towards your final grade.
2. That said, it is extremely important that you learn how to collect the traces yourself also. In many of the exercises below, your first step should be to collect the trace yourself based on the stated goal and do a rough comparison with the provided trace to see if you got it right.
3. While you should answer questions against the provided trace (which is graded), it is also very important that you answer the same questions asked against your own collected trace (though this cannot be graded). This will reinforce your understanding and sometimes you may see interesting quirks that arise when different machines are used for tracing.
4. The grading here is essentially to motivate you to complete the work, we know you can cheat easily. While it is very tempting to cheat, I urge you to resist the temptation. It is your learning that will suffer. Labs really are fun, and we are here to help you with any question you have, short of giving away the answers. You are also very welcome to talk with your friends, discuss with them anything, again other than asking the answers.
5. Remember the goal in these weekly labs is learning. To protect against cheating, we will have a few “proctored” exams (including viva) with good weightage.

Exercise 1: Journey of a Packet at a Host

Goal: To understand layering and demultiplexing, Hari Puttar wants to capture some packets. Unlike Potter, Puttar does not know magic. He has to do it the hard way. He wants to do this under the context of web browsing. So, help him design an experiment that captures only those packets that are exchanged between his machine and IITB CSE web server

Guidance: You can use tcpdump or wireshark with relevant filters to capture packets. You can examine the trace in wireshark. Try to see that you capture only the traffic of this web connection and no other background traffic. You could use a browser (may want to close all tabs, other than one where you will access the CSE web server). You could also use wget to download the url which is cleaner and simpler. And remember to start wireshark/tcpdump before you start generating traffic!

1. If you have not already watched the videos “Action at a Host”, do so.
2. Look at relevant configuration files (/etc/hostname; /etc/hosts; /etc/protocols; /etc/services) and play around with commands host, ip.
    i. Linux/MAC users, this will be straightforward and can be done at terminal.
    ii. Windows users will need the docker setup. Go through instructions provided under references
3. There is a quiz titled “Action at a Host” on BODHITREE. Attempt this quiz. Note this cannot be autograded, but is good practice.
4. There is another quiz on SAFE that is autogradable, attempt the quiz titled “Lab02: Action at a Host -01” on SAFE. This quiz covers aspects beyond this experiment. But this experiment is also included, and is based on the trace “sample-cse-iitb.pcapng”.
5. Like mentioned earlier, collect your own trace and answer the questions against the collected trace.
    i. This does not need any docker setup, you can do it using browser and wireshark.
    ii. The trace provided by us is somewhat older, CSE web server setup has changed since then. You may want to compare/contrast as to what changed. But the quiz is against the older trace.

Exercise 2: Digger deeper into the Internet Protocol Stack

Goal: With the success of the previous experiment, Hari Puttar now wants to capture and examine different types of traffic, basically arp, ICMP (protocol used by ping) and ssh. He wants to capture all of the above in just one single trace. Help him design an experiment to do the same.

Guidance: The provided wireshark trace “types-of-traffic.pcapng” has a variety of traffic generated via the below means. Note, you do not need to understand these protocols at this stage, we will just use this traffic to focus on the concept of layering/multiplexing. You can do the same to generate your own trace. And remember to start wireshark/tcpdump before you start generating traffic!

1. Pinging an IP address (can generate both ARP and ICMP protocol traffic). Example Command: ping www.ee.iitb.ac.in
2. Two SSH sessions but between the same two machines. Example Command: ssh chebrolu@10.129.1.153 (done on two terminals at about same time)

Once you open the trace in wireshark, click on the protocol field to order the packets according to the protocol.

1. Examine the provided trace “types-of-traffic.pcapng” and then attempt the quiz titled “Lab02: Types of Traffic -02 ” on SAFE.
2. Like mentioned earlier, collect your own trace and answer the questions against the collected trace.
    i. No need for docker for this setup as well. Ping and ssh should work from the command line (including in windows, provided you installed wsl as part of docker setup).
    ii. If you are within IITB, you can ping any department web server. Ping may not work as easily if you are outside IITB. But you can always ping your next hop router (to determine nexthop router; in linux/mac: ip route; windows: route PRINT)
    iii. To generate ssh traffic, again if you are within IITB, you can ssh to some of the sl-2 machines (e.g. ssh @sl2-15.cse.iitb.ac.in"">username@sl2-15.cse.iitb.ac.in). If outside, again it is tough, but you can generate some ssh traffic via “ssh -T git@github.com”, though you cannot really login.

Exercise 3: Metrics

Goal: Hari Puttar has heard of the terms throughput and delay and wants to measure them via an actual experiment. To measure throughput, he wishes to download an mp4 file and see how fast the download happens on his network interface. And to measure delay, specifically round-trip delay, he wishes to send a packet to a neighbor and see how long it takes for an acknowledgment to come back for the packet. Design an experiment for him to achieve this.

Guidance:

To measure throughput, one can download a large file from a remote host (located preferably outside your organization since local LAN will be very fast; the bottleneck here would be the link that connects your organization to the outside world) and see how fast the download happens. For this you can do the following.

1. Start wireshark to capture packets on the right interface
2. At a terminal, type “wget http://speedtest.tele2.net/1MB.zip” This downloads a 1 Mega Byte (MB)file. (If it is too fast, you can change the 1MB.zip to 10MB.zip. But be sure that wireshark captures the relevant download.) You can also use browser for this.
3. Once download ends, stop wireshark capture, save the trace and examine it in leisure.

To measure delay, you can use ping. Ping sends a sequence of “request” packets from your machine to a remote host (you are trying to determine if this remote host is up or not) and the remote host may send back “replies”. Based on the time elapsed between when a request was sent to when the corresponding reply was obtained, ping calculates the Round-Trip-Time (RTT) between your machine and the remote host.

1. Start wireshark to capture packets on the right interface
2. At a terminal, type “ping www.ee.iitb.ac.in or ping 10.129.1.153” (Note many machines do not normally reply to pings. You can try out a few local machines using their hostnames or ip-addresses to see if ping works for them. And be sure that wireshark captures the relevant ping.)
3. Once download ends, stop wireshark capture, save the trace and examine it in leisure.

For your convenience, we have done all of the above for your exploration, captured in a single trace “metrics.pcapng”. The screen shots of the command execution are also provided, “wget.png” and “ping.png”. All questions will be based on this trace. But do try to capture your own trace and analyse it also based on the questions asked in the quiz. Some of the earlier explanations regarding ping apply here as well, if you intend to capture this traffic.

Examine the given trace and then attempt the quiz titled “Lab02: Metrics - 03” on SAFE

View Document

Discussion Forum

Search here

Threads : 0

Recent | Earlier | Popularity | Importance | Instructor

Create thread

© 2020 My BodhiTree. All Rights Reserved