In the name of Allah

الله

TCP and its Applications
Laboratory Manual

University of Tehran

School of Electrical and Computer Engineering

Computer Network Lab

Dr. Ahmad Khonsari -
a_khonsari@ut.ac.ir

Amir Haji Ali Khamseh'i -
khamse@ut.ac.ir

Sina Kashi pazha -
sina_kashipazha@ut.ac.ir

Mohammad Ali Shahsavand -
mashahsavand@ut.ac.ir

Amirahmad Khordadi -
a.a.khordadi@ut.ac.ir

December 22, 2018

۱ دی ۱۳۹۷

1   Initialization

SSH to your machine[1] use command-line or other solutions[2] with enabled X11-forwarding:
$$\text{\$ \textbf{ssh -X} \texttt{mininet@<your-vm-ip-address>}}$$
Enter `mininet` as password.
Run mininet with default topology:

mininet> **sudo mn**

> hint: you can add link delay to see network process and have better experience. command: **sudo mn** ⋯ **--link tc,delay=10ms**

Telnet terminal

Execute following commands:

h2> **/etc/init.d/xinetd restart**

h1> **wireshark &**

h1> **telnet** *10.0.0.2*

Report

1. Explain TCP connection establishment and termination using the wireshark output.

2. What were the announced MSS values for the two hosts?

3. What happens if there is an intermediate network that has an MTU less than the MSS of each host? See if the DF flag was set in wireshark output.
   *you can change interface **MTU** with* **sudo ifconfig $ETH mtu 1400**

TCP vs UDP

Use **socket**[3] to send a UDP datagram to `h2`:

h2> **socket -u -s 8888**

h1> **wireshark &**

h1> **socket -u -i -n1** *10.0.0.2* **8888**

Save[4] the **wireshark** output for your lab report.
Restart the above **wireshark** command, execute **socket** in the TCP mode:

h2> **socket -s 8888**

h1> **wireshark &**

h1> **socket -i -n1** *10.0.0.2* **8888**

Save the **wireshark** output for your lab report.

---

[1]e.g. 172.18.33.52
[2]Putty, Mobaxterm, . . .
[3]Basic command is sock use alternative socket (linked to sock)
[4]Take snapshot and save image

Report

1. Explain what happened in both the UDP and TCP cases.

2. When a client requests a non-existing server (e.g. *10.0.0.3*), how do UDP and TCP handle this request, respectively?

MSS and MTU

Issue following commands:

h2> **/etc/init.d/xinetd restart**

h1> **wireshark &**

h1> **telnet** *10.0.0.2*

After logging in to the host, type `date` and press the `Enter` key.
Now, in order to generate data faster than the round-trip time of a single byte to be sent and echoed, type any sequence of keys in the **telnet** window very rapidly[5].
<u>Save</u> the **wireshark** output for your lab report.

Report

Answer the following questions, based upon the **wireshark** output saved in the above exercise.

1. What is a delayed acknowledgement? What is it used for?

2. Can you see any delayed acknowledgements in your **wireshark** output?
   If yes, explain the reason. Mark some of the lines with delayed acknowledgements, and submit the **wireshark** output with your report.
   Explain how the delayed ACK timer operates from your **wireshark** output.
   If you dont see any delayed acknowledgements, explain the reason why none was observed.

3. What is the **Nagle** algorithm used for?
   From your **wireshark** output, can you tell whether the Nagle algorithm is enabled or not? Give the reason for your answer. From your **wireshark** output for when you typed very rapidly, can you see any segment that contains more than one character going from your workstation to the remote machine?

IP Segment

Issue following commands:

h2> **wireshark &**

h2> **socket -i -s 7777**

h1> **socket -i -n16** *10.0.0.2* **7777**

Do the same experiment three times.
<u>Save</u> all the **wireshark** outputs for your lab report.

---

[5]for example hold "A" key or write "qwertyuiop" in telnet window.

Report

1. Using one of three **wireshark** outputs, explain the operation of TCP in terms of data segments and their acknowledgements. Does the number of data segments differ from that of their acknowledgements? Compare all the **wireshark** outputs you saved. Discuss any differences among them, in terms of data segments and their acknowledgements.

2. From the **wireshark** output, how many different TCP flags can you see? Enumerate the flags and explain their meanings.
   How many different TCP options can you see? Explain their meanings.

Link Up and Down

Create topology with 100 ms delay for each link and then following below steps:

 h2> **socket -s 8888**

 h1> **wireshark &**

 h1> **socket -i -n1000** *10.0.0.2* **8888**

While the sender is injecting data segments into the network, disconnect the cable connecting the sender to the hub for about ten seconds.
After observing several retransmissions, reconnect the cable:

  mininet> **link s1 h2 down**

         - after ten seconds...

  mininet> **link s1 h2 up**

When all the data segments are sent, save the **wireshark** output for the lab report.

Report

1. Submit the **wireshark** output saved in this exercise.

2. From the **wireshark** output, identify when the cable was disconnected.

3. Describe how the retransmission timer changes after sending each retransmitted packet, during the period when the cable was disconnected.

4. Explain how the number of data segments that the sender transmits at once (before getting an ACK) changes after the connection is reestablished[6].

Fragmentation

Execute the following command, which is similar to the command we used to find out the maximum size of a UDP datagram in previous chapter,

 h2> **wireshark &**

 h2> **socket -s 8888**

 h1> **socket -i -n1 -w***n* *10.0.0.2* **8888**

Let n be larger than the maximum UDP datagram size we found in previous chapter.
As an example, you may use n = 70080.

---

[6]TCP Window visualizer

Report

1. Did you observe any IP fragmentation?

2. If IP fragmentation did not occur this time, how do you explain this compared to what you observed in previous chapter for UDP packets?

Keepalive parameter

Execute **sysctl -A | grep keepalive** to display the default values of the TCP kernel parameters that are related to the TCP keepalive timer.

Report

Answer the above questions.

1. What is the default value of the TCP keepalive timer?

2. What is the maximum number of TCP keepalive probes a host can send?

Linux TCP/IP kernel parameter

Study the manual page of **/sbin/sysctl**. Examine the default values of some TCP/IP configuration parameters that you might be interested in. Examine the configuration files in the **/proc/sys/net/ipv4** directory.

Report

1. Explain what is **sysctl** command for?

2. Explain two arbitrary TCP/IP configuration parameters. What is their default values?

3. Name two arbitrary file in the **/proc/sys/net/ipv4** directory. What is their content?