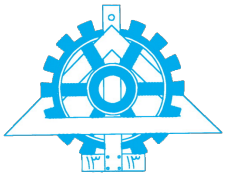
In the name of Allah

بسم الله‌الرحمن الرحیم

TCP and its Applications Laboratory Manual



University of Tehran
دانشگاه تهران

School of Electrical and Computer Engineering
دانشکده مهندسی برق و کامپیوتر

Computer Network Lab
آزمایشگاه شبکه‌های کامپیوتری

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

Amir Haji Ali Khamseh'i - امیر حاجی علی خمسه‌ء
khamse@ut.ac.ir

Sina Kashi pazha - سینا کاشی پزها
sina_kashipazha@ut.ac.ir

Amirahmad Khordadi - امیر احمد خردادی
khordadi@ut.ac.ir

October 19, 2018

۲۷ مهر ۱۳۹۷

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?
   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.

TCP vs UDP

Use **socket**[1] 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 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.

Report

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

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

[1] Basic command is sock use alternative socket (linked to sock)

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 don't 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?

Exercise 4

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.
Save all the **wireshark** outputs for your lab report.

Report

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.

Report

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

Issue following commands:

h2> **socket -s 8888**

h1> **wireshark &**

h1> **socket -i -n1000000** *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

Submit the **wireshark** output saved in this exercise.
From the **wireshark** output, identify when the cable was disconnected.
Describe how the retransmission timer changes after sending each retransmitted packet, during the period when the cable was disconnected.
Explain how the number of data segments that the sender transmits at once (before getting an ACK) changes after the connection is reestablished.

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.

Report

Did you observe any IP fragmentation?
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

Explain what is **sysctl** command for?
Explain two arbitrary TCP/IP configuration parameters. What is their default values?
Name two arbitrary file in the `/proc/sys/net/ipv4` directory. What is their content?