COMPUTER NETWORKS (CS F303)
Topic: Performance of TCP using Wireshark

Learning Objectives:

- We have learned about the working of various application layer protocols (e.g., http, ftp, DNS etc.) in the previous labs (Lab 1 and Lab 2) sessions conducted on Wireshark. In this lab session we will learn about the transport (i.e., TCP) and network layer (IP) protocols.
- Use of Filter Expressions in Wireshark to select the packets of our interest

Filter Expression in Wireshark

Every field in the packet details pane can be used as a filter string, this will result in showing only the packets where this field exists. For example: the filter string: *tcp* will show all packets containing the tcp protocol. Wireshark provides a simple but powerful display filter language that allows you to build quite complex filter expressions. You can compare values in packets as well as combine expressions into more specific expressions.

Apply following display filters to your capture and observe the effect of each on display window:

```
tcp.port eq 80
tcp.port == 80 || tcp.port == 443
ip.addr == 172.22.24.65
ip.src==172.43.54.65 or ip.dst==172.43.54.65 (use IP address values from your capture)
ip.len == 1500
```

Display Filters Comparison Operators

English	C-like	Description and example
eq	==	Equal. `ip.src==10.0.0.5`
ne	!=	Not equal. `ip.src!=10.0.0.5`
gt	>	Greater than. `frame.len > 10`
lt	<	Less than. `frame.len < 128`
ge	>=	Greater than or equal to. `frame.len ge 0x100`
le	<=	Less than or equal to. `frame.len <= 0x20`
contains		Protocol, field or slice contains a value. `sip.To contains "a1762"`
matches	~	Protocol or text field match Perl regualar expression. `http.host matches "acme\.(org|com|net)"`
bitwise_and	&	Compare bit field value. `tcp.flags & 0x02`

Table: 1

Display Filters Logical Operations

English	C-like	Description and example
and	&&	Logical AND. `ip.src==10.0.0.5 and tcp.flags.fin`
or	||	Logical OR. `ip.scr==10.0.0.5 or ip.src==192.1.1.1`
xor	^^	Logical XOR. `tr.dst[0:3] == 0.6.29 xor tr.src[0:3] == 0.6.29`
not	!	Logical NOT. `not llc`

Table: 2

Note: The filter toolbar (Click on Expression button to view this) lets you quickly edit and apply display filters.

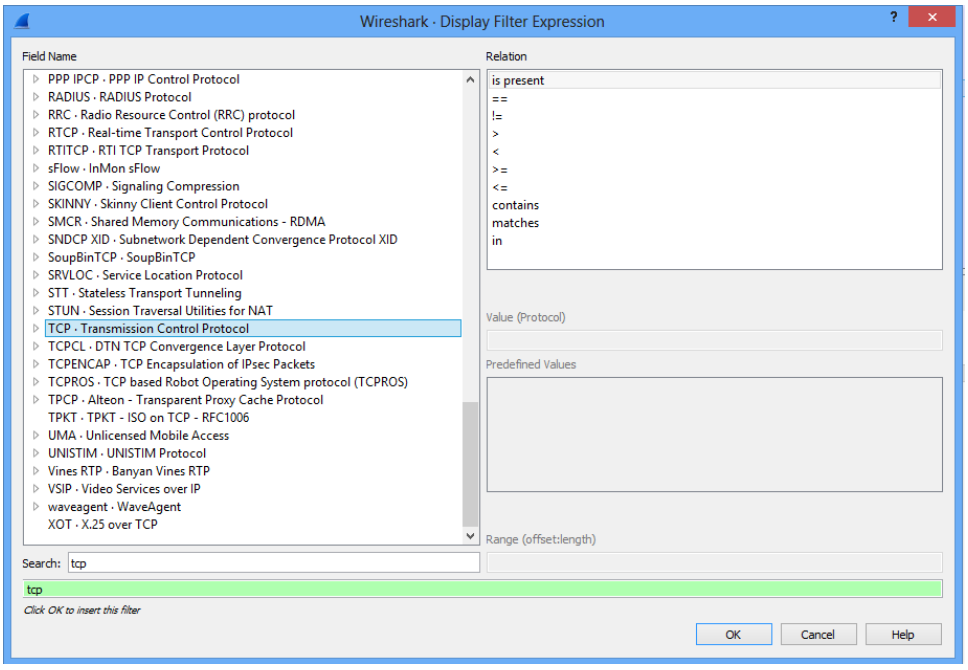

Fig. 1

TCP Protocol Performance and Operational Analysis

This lab exercise is to understand the transport layer protocol i.e. TCP in detail. Capture data packets in promiscuous mode by using Wireshark from the URLs (e.g. http://www.bits-pilani.ac.in, http://www.smc2018.org/, http://www.robio2018.org/ etc.). Apply appropriate filter to visualize only the TCP segments in your capture. Answer the following questions:

Note: Capture some decent amount of packets by clicking various links on the index pages of the URLs to observe the results effectively. Select a TCP stream (use Filter for this) for the questions which are applicable to a particular TCP stream.

1. What is the sequence number of the TCP SYN segment that is used to initiate the TCP connection between the client computer and www.bits-pilani.ac.in? What is it in the segment that identifies the segment as a SYN segment?

2. What is the sequence number of the SYNACK segment sent by www.bits-pilani.ac.in to the client computer in reply to the SYN? What is the value of the Acknowledgement field in the SYNACK segment? How did www.bits-pilani.ac.in determine that value? What is it in the segment that identifies the segment as a SYNACK segment?
Hint: Observe the middle window in your trace for the specified packet

3. The sequence numbers that you observed so far in your trace are the actual sequence numbers? Is it overruling the things which you read in the text? Don't be confused. Are you really wanted to know the actual things? Follow Edit->preferences->protocols->tcp and unmark the Relative Sequence Number and Window Scaling. Now check the sequence numbers in your trace. Did you observe any change? Explain the reason.

4. What is the length of each of the first three TCP segments for http://www.robio2018.org/?

5. Are there any retransmitted segments in the trace file? What did you check for (in the trace) in order to answer this question?

6. What is the throughput (average bytes transferred per unit time) for the TCP connection? (*Select different TCP streams for this observation)*
Hint: *Use statistics->TCP Stream Graph->Throughput graph*

7. What proportion of total TCP segments has their **PSH** flag set**?**
(Hint: Use statistics->flow graph… or You can set appropriate Display filter to know this information..)

TCP congestion control in action:
Let's now examine the amount of data sent per unit time from the client to the server. Rather than (tediously!) calculating this from the raw data in the Wireshark window, we'll use one of Wireshark's TCP graphing utilities - *Time-Sequence-Graph (Stevens)* - to plot out data.
Select a TCP segment in the Wireshark's "listing of captured-packets" window. Then in menu select: *Statistics->TCP Stream Graph-> Time-Sequence-Graph(Stevens)*.

8. Use the *Time-Sequence-Graph (Stevens*) plotting tool to view the sequence number versus time plot of segments sent between the client and the server. Can you identify where TCP's slow start phase begins and ends, and where congestion avoidance takes over?

9. Comment on ways in which the measured data differs from the idealized behavior of TCP that we've studied in the text.

10. Where do you find the (advertised) amount of available buffer space at the receiver (the server)? Does the lack of receiver buffer space ever throttle the sender?

11. What is the overall throughput in bit/s (the average of bits transferred per unit time) for one TCP stream? Explain how you calculate this value.

*****