Lab 2: transport layer TCP/UDP

How to write the report: Each student may work individually or in a group of up to three members. If working in a group, all students must submit the same PDF file that includes all exercises and the names of every group member. You have to run the provided examples and use them to illustrate the underlying concepts. For the exercises, analyze the observed behavior, describe your findings, include relevant images, and document the steps you followed to obtain your results.

Example

The goal of this lab is to study the behavior of the TCP protocol using the NS-2 (Network Simulator 2) running in Windows. It is a discrete event network simulator used to simulate network protocols, topologies, traffic patterns, and link behaviors. Written in C++, NS-2 uses a simple network description language called TCL. It is written in C++ and it supports visualization using NAM (Network Animator).

The typical NS-2 workflow is:

1. Create simulator
2. Create nodes
3. Create agents
4. Create applications
5. Create links → duplex-link or simplex-link
6. Schedule events → start/stop apps, monitor variables
7. Run simulation → ns script_name.tcl
8. Visualize → nam out.nam
9. Analyze trace files

The following `example.tcl` script defines a simple network topology consisting of **two nodes connected via a point-to-point link**. It simulates a **TCP connection** between the nodes, with an **FTP application generating traffic** from the source to the sink. The FTP flow runs until **0.7 seconds** in the simulation.

With the following `example.tcl` script, do the following:

- Add comments to the code to understand the purpose of each command
- Visualize the network traffic using **NAM**
- Examine the trace file and plot sender's congestion window size as a function of the time.

```
set ns [new Simulator]
set ftrace [open trace.tr w]
set nf [open out.nam w]
$ns namtrace-all $nf

proc finish {} {
    global ns ftrace nf
    $ns flush-trace
    close $nf
    close $ftrace
    exit 0
}
proc tracewindow {} {
    global tcp0 ftrace
    set ns [Simulator instance]
    if {![info exists tcp0]} { return }
    set time 0.001
    set now [$ns now]
    set now [format "%.3f" $now]
    set cwnd [$tcp0 set cwnd_]
    puts $ftrace "$now $cwnd"
    $ns at [expr $now+$time] {tracewindow}
}
Agent/TCP set packetSize_ 1500
Agent/TCP set maxcwnd_ 30

set n0 [$ns node]
set tcp0 [new Agent/TCP]
$ns attach-agent $n0 $tcp0
set ftp [new Application/FTP]
$ftp attach-agent $tcp0

set n1 [$ns node]
set tcp1 [new Agent/TCPSink]
$ns attach-agent $n1 $tcp1

$ns duplex-link $n0 $n1 10Mb 10ms DropTail
$ns connect $tcp0 $tcp1

$ns at 0.0 {$ftp start}
$ns at 0.7 {$ftp stop}
$ns at 0.0 {tracewindow}
$ns at 1.0 {finish}
$ns run
```

Study of the congestion window behavior

Ex1

Considering the above example do:

- Modify the point-to-point FTP communication with 10Mb bandwidth, 40ms delay
- Run the simulation for 3 seconds and the observation interval of 0.01
- Modify the trace procedure in order to track sequence number, congestion window, and ack
- Write trace results in a .csv ‘;’ separated

Ex2

Modify the script from Exercise 1 so that it accepts, in this order, the congestion window, bandwidth, and propagation delay as input parameters. Then, create an automated Bash script capable of running multiple simulations with different parameter configurations. Assuming ideal conditions, no packet loss, evaluate the average throughput obtained and plot the relationship with the changing parameter. Would it possible to forecast this behavior and can you define a general formula for average receiver throughput as a function of maximum congestion window size, round-trip time (RTT), and link capacity?

Scenario 1:

- 10 Mb bandwidth
- Propagation delay 40ms
- Receiver congestion window: 1, 5, 10, 15, 20, 25, and 30 packets

Scenario 2:

- 10 Mb bandwidth
- Receiver congestion window 30 packets
- Propagation delay: 2.5, 5, 10, 15, 30 and 40ms

Scenario 3:

- Receiver congestion window 30 packets
- Propagation delay 40ms
- Bandwidth: 1, 5, 10, 15, 20, 25, and 30Mb.

Ex3

The maximum congestion window size is typically limited to 65 kB. However, this value may impose significant constraints on high-speed networks. To illustrate this, consider a reference network with a span of 2000 km, a link capacity of 155.5 Mb/s, and a propagation speed of 200,000 km/s, where transmission delays can be considered negligible. Under these conditions, what are the potential implications of the 65 kB congestion window limit, and what measures could be implemented to overcome this limitation?

Ex4

Develop a TCL script to model a network topology composed of three nodes: a source node (A), a router (R), and a destination node (B). Configure all nodes with identical TCP parameters, setting the maximum congestion window size and the slow start threshold to 64 packets. The router should be configured with a limited queue size, capable of storing up to 20 packets. Conduct a simulation to trace the evolution of the sender's congestion window as a function of time. Finally, analyze and interpret the results obtained from the trace, discussing the observed congestion control behavior in relation to TCP's mechanisms.