NS2 – LAB 3
Eng. Ibrahim Amreya & Dr. Amjad Hawash
Network & Information Security Program.
An-Najah National University.

PERFORMANCE ANALYSIS OF CSMA/CA AND CSMA/CD PROTOCOLS

Objectives:
To create scenario and study the performance of CSMA / CD protocol through simulation

Ethernet is a LAN (Local area Network) protocol operating at the MAC (Medium Access Control) layer. Ethernet has been standardized as per IEEE 802.3. The underlying protocol in Ethernet is known as the CSMA / CD – Carrier Sense Multiple Access / Collision Detection. The working of the Ethernet protocol is as explained below, A node which has data to transmit senses the channel. If the channel is idle then, the data is transmitted. If the channel is busy then, the station defers transmission until the channel is sensed to be idle and then immediately transmitted. If more than one node starts data transmission at the same time, the data collides. This collision is heard by the transmitting nodes which enter into contention phase. The contending nodes resolve contention using an algorithm called Truncated binary exponential back off.

Steps:

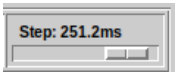
1. Create a simulator object
2. Define different colors for different data flows
3. Open a nam trace file and define finish procedure then close the trace file, and execute nam on trace file.
4. Create six nodes that forms a network numbered from 0 to 5
5. Create duplex links between the nodes and add Orientation to the nodes for setting a LAN topology
6. Setup TCP Connection between n(0) and n(4)
7. Apply FTP Traffic over TCP
8. Setup UDP Connection between n(1) and n(5)
9. Apply CBR Traffic over UDP.
10. Apply CSMA/CD mechanisms and study its performance
11. Schedule events and run the program.

The program:

```
set ns [new Simulator]
#Define different colors for data flows (for NAM)
$ns color 1 Blue
$ns color 2 Red
#Open the Trace files
set file1 [open out.tr w]
set winfile [open WinFile1 w]
$ns trace-all $file1
#Open the NAM trace file
set file2 [open out.nam w]
$ns namtrace-all $file2
#Define a 'finish' procedure
```

```
proc finish {} {
global ns file1 file2
$ns flush-trace
close $file1
close $file2
exec nam out.nam &
exit 0
}
#Create six nodes
set n0 [$ns node]
set n1 [$ns node]
set n2 [$ns node]
set n3 [$ns node]
set n4 [$ns node]
set n5 [$ns node]
$n1 color red
$n1 shape box
#Create links between the nodes
$ns duplex-link $n0 $n2 2Mb 10ms DropTail
$ns duplex-link $n1 $n2 2Mb 10ms DropTail
$ns simplex-link $n2 $n3 0.3Mb 100ms DropTail
$ns simplex-link $n3 $n2 0.3Mb 100ms DropTail
set lan [$ns newLan "$n3 $n4 $n5" 0.5Mb 40ms LL Queue/DropTail MAC/Csma/Cd Channel]
#Setup a TCP connection
set tcp [new Agent/TCP/Newreno]
$ns attach-agent $n0 $tcp
set sink [new Agent/TCPSink/DelAck]
$ns attach-agent $n4 $sink
$ns connect $tcp $sink
$tcp set fid_ 1
$tcp set window_ 8000
$tcp set packetSize_ 552
#Setup a FTP over TCP connection
set ftp [new Application/FTP]
$ftp attach-agent $tcp
$ftp set type_ FTP
#Setup a UDP connection
set udp [new Agent/UDP]
$ns attach-agent $n1 $udp
set null [new Agent/Null]
$ns attach-agent $n5 $null
$ns connect $udp $null
$udp set fid_ 2
#Setup a CBR over UDP connection
set cbr [new Application/Traffic/CBR]
$cbr attach-agent $udp
$cbr set type_ CBR
$cbr set packet_size_ 1000
$cbr set rate_ 0.01mb
```

```
$cbr set random_ false
$ns at 0.1 "$cbr start"
$ns at 1.0 "$ftp start"
$ns at 124.0 "$ftp stop"
$ns at 124.5 "$cbr stop"
# next procedure gets two arguments: the name of the
# tcp source node, will be called here "tcp",
# and the name of output file.
proc plotWindow {tcpSource file} {
global ns
set time 0.1
set now [$ns now]
set cwnd [$tcpSource set cwnd_]
set wnd [$tcpSource set window_]
puts $file "$now $cwnd"
$ns at [expr $now+$time] "plotWindow $tcpSource $file" }
$ns at 0.1 "plotWindow $tcp $winfile"
$ns at 5 "$ns trace-annotate \"packet drop\""
# PPP
$ns at 125.0 "finish"
$ns run
```



-If you feel the simulator too slow, you can increase its speed by increasing step time from the upper-right corner of the simulator.
- You will find the file WinFile1 in the same folder where you saved the TCL file. WinFile1 contains two columns of data: Time and cwnd. In order to plot the data use the command : xgraph WinFile1
You will see a data relation between time (x-axis) and cwnd (y-axis).

Now:

1- What is it meant by cwnd (congestion window)?
2- modify the code above as follows:
  a- change WinFile1 to WinFile2 (in line 7 of the code).
  b- change the delay values in
        $ns duplex-link $n0 $n2 2Mb 10ms DropTail
        $ns duplex-link $n1 $n2 2Mb 10ms DropTail
to become
        $ns duplex-link $n0 $n2 2Mb 20ms DropTail
        $ns duplex-link $n1 $n2 2Mb 20ms DropTail
  c- run the simulator.
3- You will find a new generated file WinFile1
4- Repeat steps 2-3 for the values of 30, 40, 50, 60 ms (each time, change the name of the WinFile to be WinFile2, WinFile3…) in order to have a set of generated values saved in different files.
5- Draw the data appear the WinFiles using Xgraph by executing the command:
        xgraph WinFile WinFile1 WinFile2 WinFile3 WinFile4 WinFile5 WinFile6

6- What do you notice with respect to cwnd?

7- prepare a report by answering question 1 above, snapshot the generated graph, and your discussion of what you got in the graph.