Program 6: Implement an Ethernet LAN using n nodes and set multiple traffic nodes and plot congestion window for different source / destination.

```
set ns [new Simulator]
set tf [open lab3.tr w]
$ns trace-all $tf
set nf [open lab3.nam w]
$ns namtrace-all $nf

set n0 [$ns node]
$n0 color "magenta"
$n0 label "src1" set n1 [$ns node] set n2 [$ns node]
$n2 color "magenta"
$n2 label "src2" set n3 [$ns node]
$n3 color "blue"
$n3 label "dest2" set n4 [$ns node]
set n5 [$ns node]
$n5 color "blue"
$n5 label "dest1"

$ns make-lan "$n0 $n1 $n2 $n3 $n4" 100Mb 100ms LL Queue/DropTail Mac/802_3
/* should come in single line */
$ns duplex-link $n4 $n5 1Mb 1ms DropTail

set tcp0 [new Agent/TCP]
$ns attach-agent $n0 $tcp0
set ftp0 [new Application/FTP]
$ftp0 attach-agent $tcp0
$ftp0 set packetSize_ 500
$ftp0 set interval_ 0.0001
set sink5 [new Agent/TCPSink]
$ns attach-agent $n5 $sink5

$ns connect $tcp0 $sink5

set tcp2 [new Agent/TCP]
$ns attach-agent $n2 $tcp2
set ftp2 [new Application/FTP]
$ftp2 attach-agent $tcp2
$ftp2 set packetSize_ 600
$ftp2 set interval_ 0.001
set sink3 [new Agent/TCPSink]
```

```
$ns attach-agent $n3 $sink3

$ns connect $tcp2 $sink3set file1 [open file1.tr w]

$tcp0 attach $file1
set file2 [open file2.tr w]
$tcp2 attach $file2

$tcp0 trace cwnd_ /* must put underscore ( _ ) after cwnd and no space between them*/
$tcp2 trace cwnd_

proc finish { } { global ns nf tf
$ns flush-trace close $tf
close $nf
exec nam lab3.nam & exit 0
}
$ns at 0.1 "$ftp0 start"
$ns at 5 "$ftp0 stop"
$ns at 7 "$ftp0 start"
$ns at 0.2 "$ftp2 start"
$ns at 8 "$ftp2 stop"
$ns at 14 "$ftp0 stop"
$ns at 10 "$ftp2 start"
$ns at 15 "$ftp2 stop"
$ns at 16 "finish"
$ns run
```

AWK file (Open a new editor using “vi command” and write awk file and save with “.awk” extension)
cwnd:- means congestion window

```
BEGIN {
}
{
if($6 =="cwnd_") /* don’t leave space after writing cwnd_ */
printf("%f\t%f\t\n",$1,$7); /* you must put \n in printf */
} END {
}
```

Steps for execution

1) Open vi editor and type program. Program name should have the extension “ **.tcl** ”

   [root@localhost ~]# vi lab3.tcl

2) Save the program by pressing **“ESC key”** first, followed by **“Shift**

and :” keyssimultaneously and type **“wq”** and press **Enter key**.

3) Open vi editor and type **awk** program. Program name should have the extension **“.awk ”**

```
[root@localhost ~]# vi lab3.awk
```

4) Save the program by pressing **“ESC key”** first, followed by **“Shift and :”** keyssimultaneously and type **“wq”** and press **Enter key**.

5) Run the simulation program

```
[root@localhost~]# ns lab3.tcl
```

6) After simulation is completed run **awk file** to see the output ,

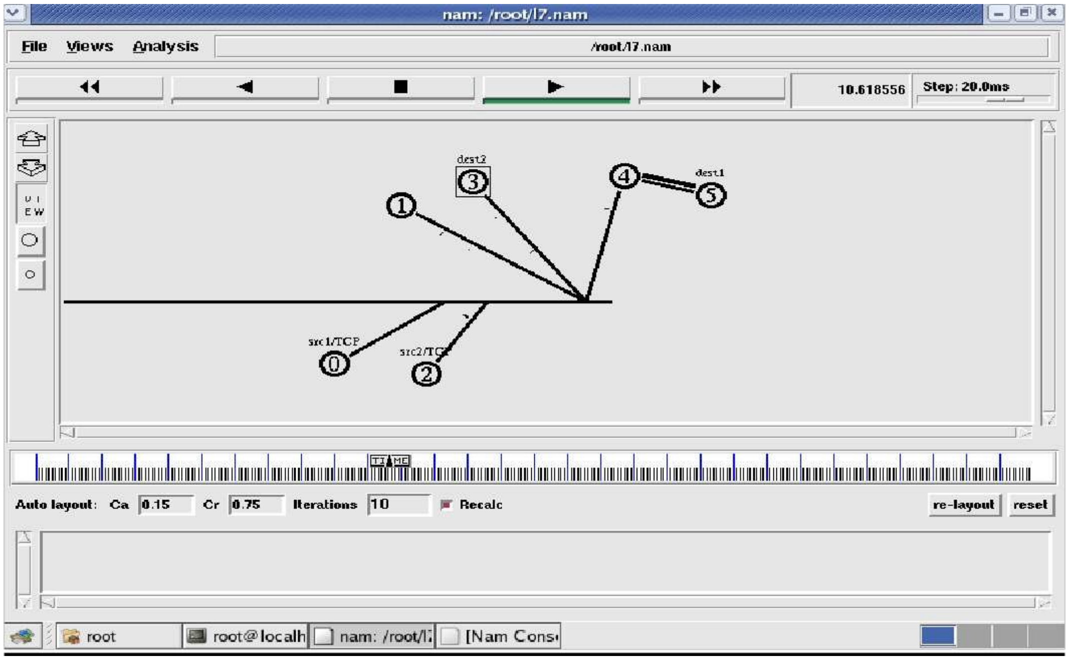
```
i.   [root@localhost~]# awk  –f  lab3.awk  file1.tr  >  a1
ii.  [root@localhost~]# awk  –f  lab3.awk  file2.tr  >  a2
iii. [root@localhost~]# xgraph a1 a2
```

7) Here we are using the congestion window trace files i.e**. file1.tr and file2.tr and weare redirecting the contents of those files to new files say a1** and **a2** using **output** redirection operator (>).

8) To see the trace file contents open the file as ,

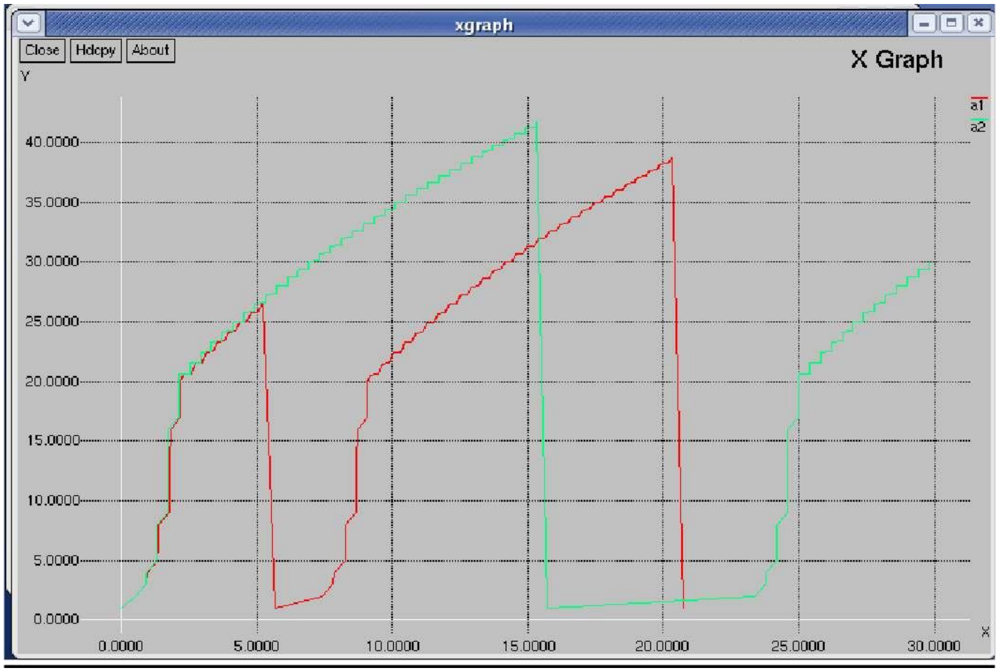
```
[root@localhost~]# vi lab3.tr
```

Topology



Auto layout: Ca 0.15 Cr 0.75 Iterations 10 Recalc

Output:



Explanation of the code:

This code is a simulation script written in the ns-2 network simulator language. It simulates a simple network scenario with FTP (File Transfer Protocol) applications running between two source nodes (src1 and src2) and two destination nodes (dest1 and dest2). The nodes are connected through a LAN and a duplex link. The simulation records various parameters like congestion window size (cwnd) and creates trace files for analysis.

code line by line:

1. set ns [new Simulator]: Creates a new network simulation object.

2. set tf [open lab3.tr w]: Opens a trace file (lab3.tr) for writing. The trace file will store simulation events.

3. $ns trace-all $tf: Enables tracing of all events in the simulation and directs the output to the trace file.

4. set nf [open lab3.nam w]: Opens another trace file (lab3.nam) for the Nam network animator.

5. $ns namtrace-all $nf: Enables tracing of all events for Nam and directs the output to the Nam trace file.

6. Node creation and configuration:

1 Nodes n0, n1, n2, n3, n4, and n5 are created.

2Colors and labels are assigned to nodes.

3A LAN is created using make-lan with a specified bandwidth, delay, link layer, and queue type.

4 A duplex link is created between nodes n4 and n5.

7. TCP agents and applications setup:

1 TCP agents (Agent/TCP) and FTP applications (Application/FTP) are created for source nodes (n0 and n2).

1 TCP sink agents (Agent/TCPSink) are created for destination nodes (n5 and n3).

2 Connections are established between TCP agents and sink agents.

8. Trace setup:

1 Trace files (file1.tr and file2.tr) are opened for writing.

2 Trace events related to congestion window size (cwnd_) are traced for both TCP agents.

9. Event scheduling using $ns at:

1 Events are scheduled at specific simulation times to start and stop FTP applications.

2 The **finish** procedure is scheduled to run at simulation time 16.

10. proc finish { }: Defines a procedure named `finish` that is executed when called.

1 Flushes and closes the trace file for the simulator (lab3.tr).

2 Closes the Nam trace file (lab3.nam).

3 Executes the Nam animator to visualize the simulation.

4 Exits the simulation.

11. $ns run: Initiates the simulation.