

# **Wireless Sensor Network Graded Lab 1 Report**

Done By:

Prasanna Natarajan

1410110298

1)

a)

Code for simulation:

```
#Create a simulator object
set ns [new Simulator]

#Define different colors for data flows (for NAM)
$ns color 1 Blue
$ns color 2 Red
$ns color 3 Green

#Open the NAM trace file
set nf [open out.nam w]
$ns namtrace-all $nf

set nf1 [open out.tr w]
$ns trace-all $nf1
proc finish {} {
    global ns nf nf1
    #Close the NAM trace file
    close $nf
    #Close the trace-all file
    close $nf1
    #Execute NAM on the trace file
    exec nam out.nam &
    exit 0
}

#Create seven nodes
set n0 [$ns node]
set n1 [$ns node]
set n2 [$ns node]
set n3 [$ns node]
set n4 [$ns node]

#Create links between the nodes
$ns duplex-link $n0 $n3 2Mb 5ms DropTail
$ns duplex-link $n1 $n3 2Mb 5ms DropTail
$ns duplex-link $n2 $n3 2Mb 5ms DropTail

$ns duplex-link $n3 $n4 600Kb 10ms DropTail

#Set Queue Size of link (n3-n4) to 10
$ns queue-limit $n3 $n4 10

#Give node position (for NAM)
$ns duplex-link-op $n3 $n0 orient left-up
$ns duplex-link-op $n3 $n2 orient left-down
$ns duplex-link-op $n3 $n1 orient left
```

```
$ns duplex-link-op $n4 $n3 orient left

#Monitor the queue for link (n3-n4). (for NAM)
$ns duplex-link-op $n3 $n4 queuePos 0.5

#Setup a TCP connection
set tcp [new Agent/TCP]
$tcp set class_ 1
$tcp set window_ 8000
$tcp set packetSize_ 552
$ns attach-agent $n0 $tcp
set sink [new Agent/TCPSink]
$ns attach-agent $n4 $sink
$ns connect $tcp $sink
$tcp set fid_ 1

#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 $n4 $null
$ns connect $udp $null
$udp set fid_ 2

set udp1 [new Agent/UDP]
$ns attach-agent $n2 $udp1
$ns connect $udp1 $null
$udp1 set fid_ 3

#Setup a CBR over UDP connection
set cbr0 [new Application/Traffic/CBR]
$cbr0 attach-agent $udp
$cbr0 set type_ CBR
$cbr0 set interval_ 1
$cbr0 set packet_size_ 200
$cbr0 set rate_ 500kb
$cbr0 set random_ false

#Setup a CBR over UDP connection
set exp1 [new Application/Traffic/Exponential]
$exp1 attach-agent $udp1
$exp1 set type_ Exponential
```

```

$exp1 set packetSize_ 200
$exp1 set burst_time_ 2
$exp1 set idle_time_ 1
$exp1 set rate_ 500kb
$exp1 set random_ false

#Schedule events for the CBR and FTP agents

$ns at 0.5 "$ftp start"
$ns at 1   "$cbr0 start"
$ns at 1.5 "$exp1 start"
$ns at 2.5 "$cbr0 stop"
$ns at 3.5 "$ftp stop"
$ns at 4.5 "$exp1 stop"

$ns at 5 "finish"

$ns run

```

Code for plotting/Counting number of packet drops:

```

BEGIN {
    packetDrops = 0;
    x=0.5;
    recv4 = 0;
}

{
    action = $1;
    time = $2;
    from = $3;
    to = $4;
    type = $5;
    pktsize = $6;
    flow_id = $8;
    src = $9;
    dst = $10;
    seq_no = $11;
    packet_id = $12;
    if(action == "d"){
        packetDrops++;
    }
    if (from == "3" && to == "4"){
        if(time < x){
            recv4++;
        }else {
            recv4 = recv4/0.5;
            printf("%f %f\n", x, recv4);
            x = x + 0.5;
            recv4 = 0;
        }
    }
}

```

```

    }
}

END {

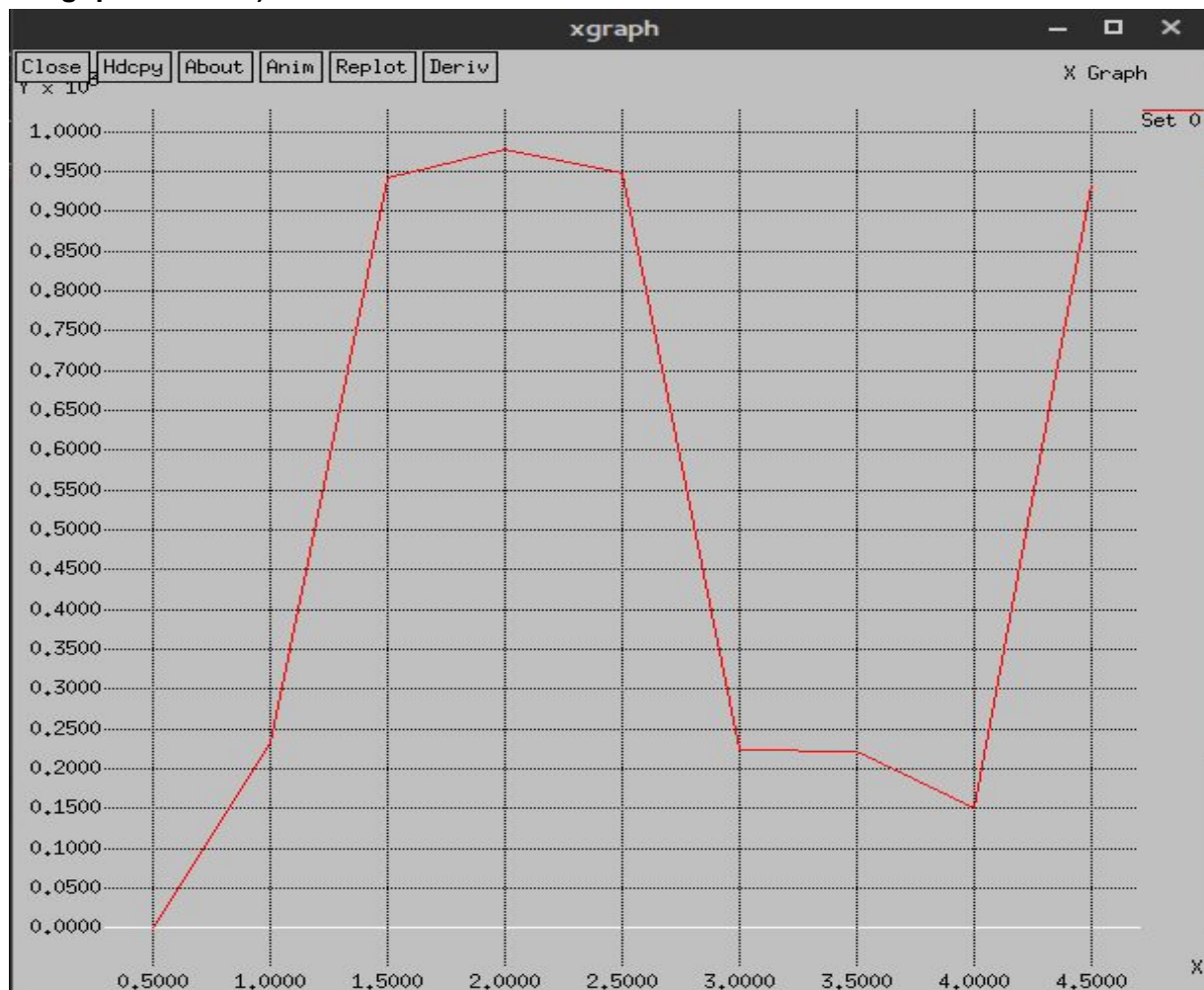
printf("num drops = %d\n",packetDrops);

}

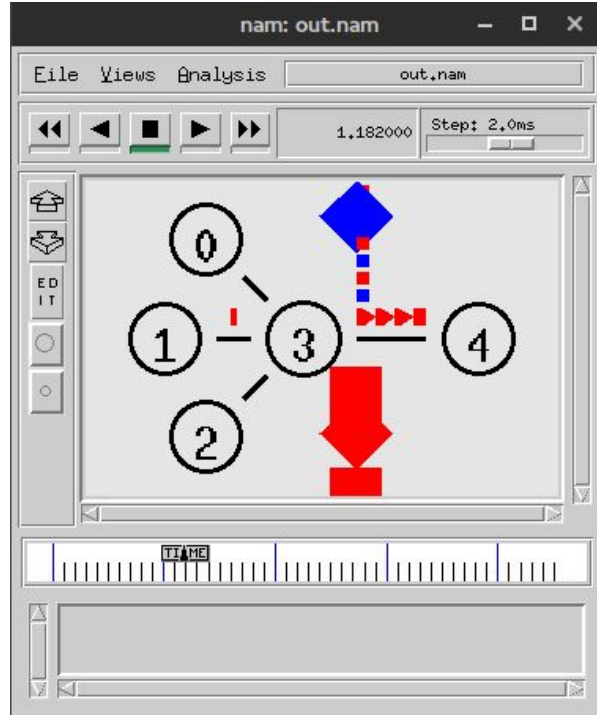
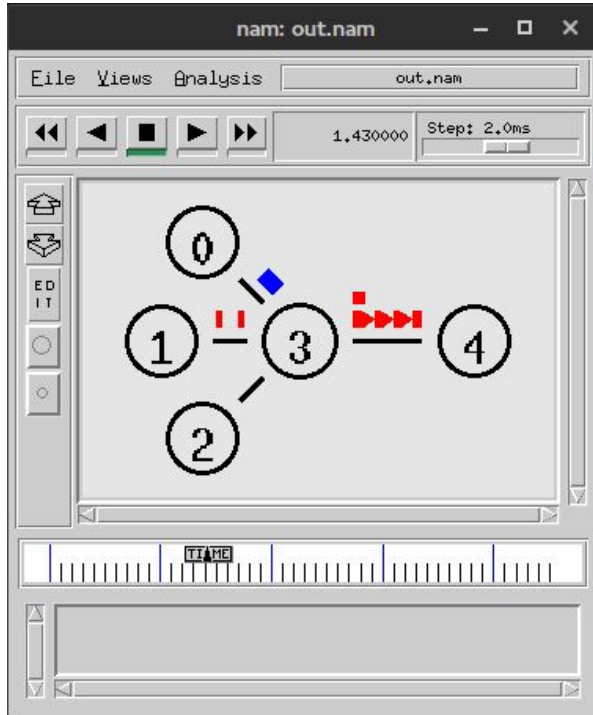
prasanna (master *) lab3 $ gawk -f plot.awk out.tr
num drops = 73
prasanna (master *) lab3 $

```

**Plot (Throughput vs Time)**



## Screenshots:



## b) Error Module

Code:

```
#Create a simulator object
set ns [new Simulator]

#Define different colors for data flows (for NAM)
$ns color 1 Blue
$ns color 2 Red
$ns color 3 Green

#Open the NAM trace file
set nf [open out.nam w]
$ns namtrace-all $nf

set nf1 [open out.tr w]
$ns trace-all $nf1
proc finish {} {
    global ns nf nf1
    #Close the NAM trace file
    close $nf
    #Close the trace-all file
    close $nf1
    #Execute NAM on the trace file
```

```

        exec nam out.nam &
        exit 0
    }

#Create seven nodes
set n0 [$ns node]
set n1 [$ns node]
set n2 [$ns node]
set n3 [$ns node]
set n4 [$ns node]

#Create links between the nodes
$ns duplex-link $n0 $n3 2Mb 5ms DropTail
$ns duplex-link $n1 $n3 2Mb 5ms DropTail
$ns duplex-link $n2 $n3 2Mb 5ms DropTail
$ns duplex-link $n3 $n4 600Kb 10ms DropTail

#Set Queue Size of link (n3-n4) to 10
$ns queue-limit $n3 $n4 10

#Give node position (for NAM)
$ns duplex-link-op $n3 $n0 orient left-up
$ns duplex-link-op $n3 $n2 orient left-down
$ns duplex-link-op $n3 $n1 orient left
$ns duplex-link-op $n4 $n3 orient left

#Monitor the queue for link (n3-n4). (for NAM)
$ns duplex-link-op $n3 $n4 queuePos 0.5

#Setup a TCP connection
set tcp [new Agent/TCP]
$tcp set class_ 1
$tcp set window_ 8000
$tcp set packetSize_ 552
$ns attach-agent $n0 $tcp
set sink [new Agent/TCPSink]
$ns attach-agent $n4 $sink
$ns connect $tcp $sink
$tcp set fid_ 1

#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 $n4 $null

```

```

$ns connect $udp $null
$udp set fid_ 2

set udp1 [new Agent/UDP]
$ns attach-agent $n2 $udp1
$ns connect $udp1 $null
$udp1 set fid_ 3
#Setup a CBR over UDP connection
set cbr0 [new Application/Traffic/CBR]
$cbr0 attach-agent $udp
$cbr0 set type_ CBR
$cbr0 set interval_ 1
$cbr0 set packet_size_ 200
$cbr0 set rate_ 500kb
$cbr0 set random_ false

#Setup a CBR over UDP connection
set exp1 [new Application/Traffic/Exponential]
$exp1 attach-agent $udp1
$exp1 set type_ Exponential
$exp1 set packetSize_ 200
$exp1 set burst_time_ 2
$exp1 set idle_time_ 1
$exp1 set rate_ 500kb
$exp1 set random_ false

#Setting up error module
# create a loss_module and set its packet error rate to 10 percent
set em [new ErrorModel]
$em set rate_ 0.1

# optional: set the unit and random variable
$em unit pkt
$em ranvar [new RandomVariable/Uniform]

# set target for dropped packets
$em drop-target [new Agent/Null]

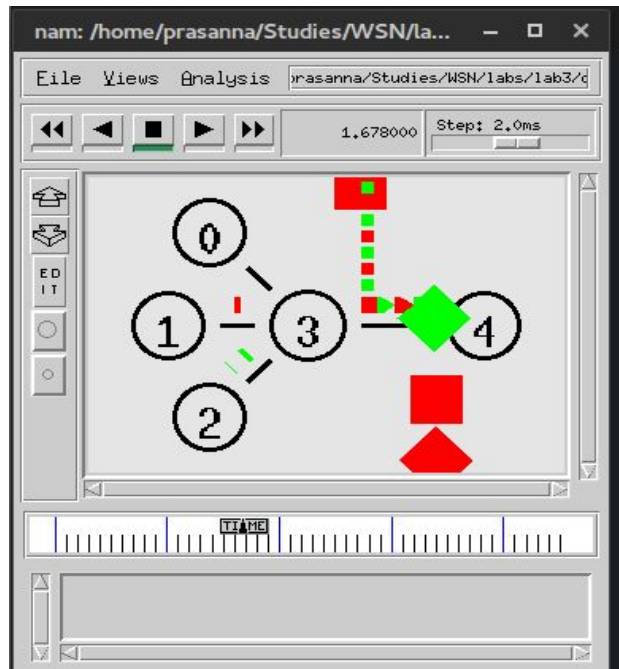
#Attaching error module to link 3-4
$ns link-lossmodel $em $n3 $n4
$ns at 0.5 "$ftp start"
$ns at 1 "$cbr0 start"
$ns at 1.5 "$exp1 start"
$ns at 2.5 "$cbr0 stop"
$ns at 3.5 "$ftp stop"
$ns at 4.5 "$exp1 stop"
$ns at 5 "finish"

$ns run

```



Screenshot after error module:



Plot (Throughput vs Time) - same awk code (as in the previous subpart) was used to plot it.



### c) TCP Reno:

Code:

```
#Create a simulator object
set ns [new Simulator]

#Define different colors for data flows (for NAM)
$ns color 1 Blue
$ns color 2 Red
$ns color 3 Green

#Open the NAM trace file
set nf [open out.nam w]
$ns namtrace-all $nf

set nf1 [open out.tr w]
$ns trace-all $nf1
proc finish {} {
    global ns nf nf1
    #Close the NAM trace file
    close $nf
    #Close the trace-all file
    close $nf1
    #Execute NAM on the trace file
    exec nam out.nam &
    exit 0
}

#Create seven nodes
set n0 [$ns node]
set n1 [$ns node]
set n2 [$ns node]
set n3 [$ns node]
set n4 [$ns node]

#Create links between the nodes
$ns duplex-link $n0 $n3 2Mb 5ms DropTail
$ns duplex-link $n1 $n3 2Mb 5ms DropTail
$ns duplex-link $n2 $n3 2Mb 5ms DropTail

$ns duplex-link $n3 $n4 600Kb 10ms DropTail

#Set Queue Size of link (n3-n4) to 10
$ns queue-limit $n3 $n4 10

#Give node position (for NAM)
$ns duplex-link-op $n3 $n0 orient left-up
$ns duplex-link-op $n3 $n2 orient left-down
$ns duplex-link-op $n3 $n1 orient left
$ns duplex-link-op $n4 $n3 orient left
```

```
#Monitor the queue for link (n3-n4). (for NAM)
$ns duplex-link-op $n3 $n4 queuePos 0.5
```

```
#Setup a TCP connection
```

```
set tcp [new Agent/TCP/Reno]
```

```
$tcp set class_ 1
```

```
$ns attach-agent $n0 $tcp
```

```
set sink [new Agent/TCPSink]
```

```
$ns attach-agent $n4 $sink
```

```
$ns connect $tcp $sink
```

```
$tcp set fid_ 1
```

```
#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 $n4 $null
```

```
$ns connect $udp $null
```

```
$udp set fid_ 2
```

```
set udp1 [new Agent/UDP]
```

```
$ns attach-agent $n2 $udp1
```

```
$ns connect $udp1 $null
```

```
$udp1 set fid_ 3
```

```
set cbr0 [new Application/Traffic/CBR]
```

```
$cbr0 attach-agent $udp
```

```
$cbr0 set type_ CBR
```

```
$cbr0 set interval_ 1
```

```
$cbr0 set packet_size_ 200
```

```
$cbr0 set rate_ 500kb
```

```
$cbr0 set random_ false
```

```
set exp1 [new Application/Traffic/Exponential]
```

```
$exp1 attach-agent $udp1
```

```
$exp1 set type_ Exponential
```

```
$exp1 set packetSize_ 200
```

```
$exp1 set burst_time_ 2
```

```
$exp1 set idle_time_ 1
```

```
$exp1 set rate_ 500kb
```

```
$exp1 set random_ false
```

```
$ns at 0.5 "$ftp start"
```

```
$ns at 1 "$cbr0 start"
```

```
$ns at 1.5 "$exp1 start"
```

```
$ns at 2.5 "$cbr0 stop"
```

```
$ns at 3.5 "$ftp stop"
```

```
$ns at 4.5 "$exp1 stop"
```

```

$ns at 5 "finish"
proc plotWindow {tcpSource outfile} {
    global ns

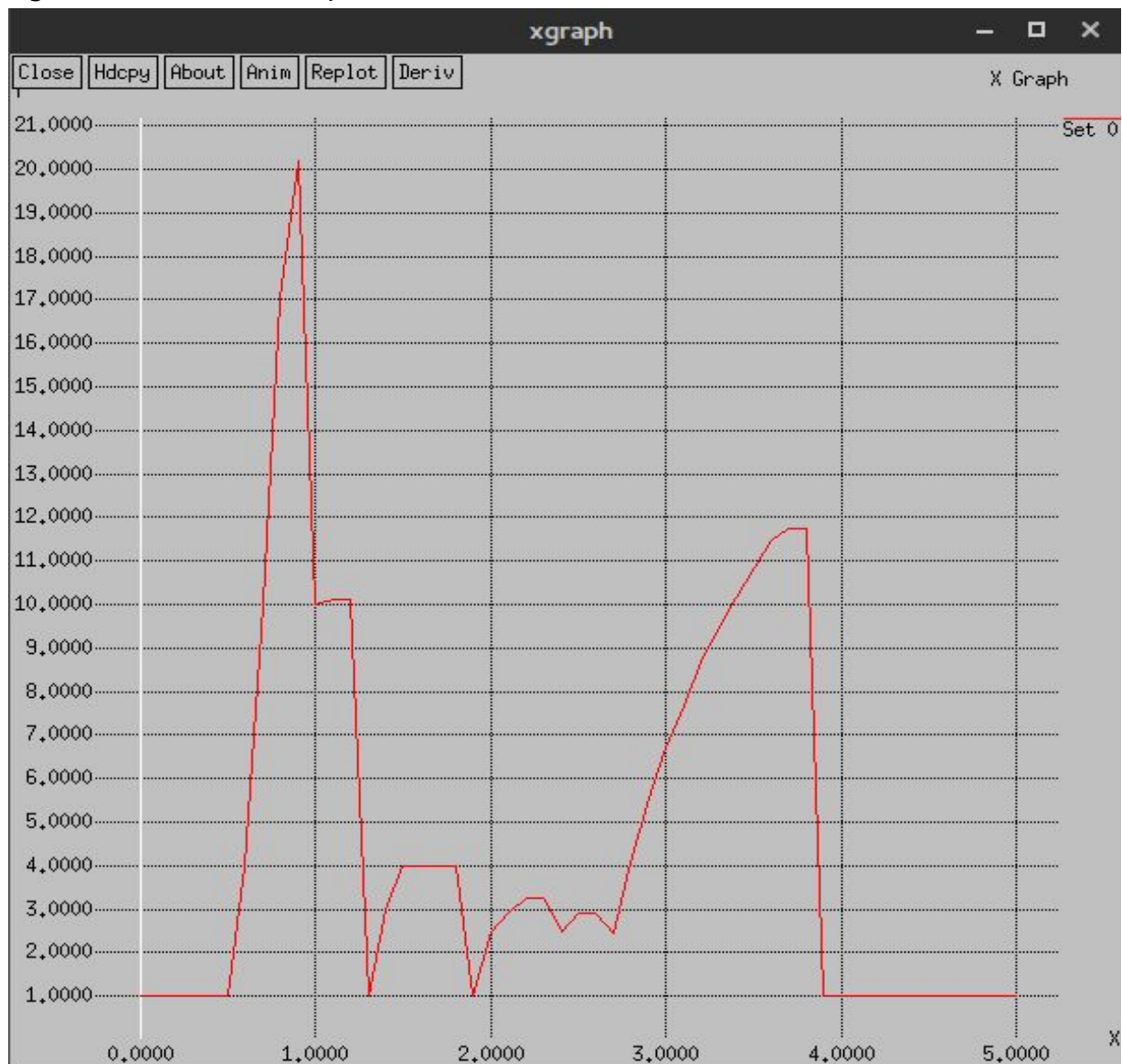
    set now [$ns now]
    set cwnd [$tcpSource set cwnd_]

    puts $outfile "$now $cwnd"

    $ns at [expr $now+0.1] "plotWindow $tcpSource $outfile"
}
$ns at 0.0 "plotWindow $tcp stdout"
$ns run

```

**Plot (congestion window vs time):**



## TCP Tahoe:

Code:

```
#Create a simulator object
set ns [new Simulator]

#Define different colors for data flows (for NAM)
$ns color 1 Blue
$ns color 2 Red
$ns color 3 Green

#Open the NAM trace file
set nf [open out.nam w]
$ns namtrace-all $nf

set nf1 [open out.tr w]
$ns trace-all $nf1
proc finish {} {
    global ns nf nf1
    #Close the NAM trace file
    close $nf
    #Close the trace-all file
    close $nf1
    #Execute NAM on the trace file
    exec nam out.nam &
    exit 0
}

#Create seven nodes
set n0 [$ns node]
set n1 [$ns node]
set n2 [$ns node]
set n3 [$ns node]
set n4 [$ns node]

#Create links between the nodes
$ns duplex-link $n0 $n3 2Mb 5ms DropTail
$ns duplex-link $n1 $n3 2Mb 5ms DropTail
$ns duplex-link $n2 $n3 2Mb 5ms DropTail
$ns duplex-link $n3 $n4 600Kb 10ms DropTail
$ns queue-limit $n3 $n4 10

#Give node position (for NAM)
$ns duplex-link-op $n3 $n0 orient left-up
$ns duplex-link-op $n3 $n2 orient left-down
$ns duplex-link-op $n3 $n1 orient left
$ns duplex-link-op $n4 $n3 orient left

#Monitor the queue for link (n3-n4). (for NAM)
$ns duplex-link-op $n3 $n4 queuePos 0.5
```

```
#Setup a TCP connection
set tcp [new Agent/TCP]
$tcp set class_ 1
$tcp set window_ 8000
$tcp set packetSize_ 552
$ns attach-agent $n0 $tcp
set sink [new Agent/TCPSink]
$ns attach-agent $n4 $sink
$ns connect $tcp $sink
$tcp set fid_ 1

#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 $n4 $null
$ns connect $udp $null
$udp set fid_ 2

set udp1 [new Agent/UDP]
$ns attach-agent $n2 $udp1
$ns connect $udp1 $null
$udp1 set fid_ 3
set cbr0 [new Application/Traffic/CBR]
$cbr0 attach-agent $udp
$cbr0 set type_ CBR
$cbr0 set interval_ 1
$cbr0 set packet_size_ 200
$cbr0 set rate_ 500kb
$cbr0 set random_ false

# Set Up a CBR over UDP connection
set exp1 [new Application/Traffic/Exponential]
$exp1 attach-agent $udp1
$exp1 set type_ Exponential
$exp1 set packetSize_ 200
$exp1 set burst_time_ 2
$exp1 set idle_time_ 1
$exp1 set rate_ 500kb
$exp1 set random_ false
$ns at 0.5 "$ftp start"
$ns at 1 "$cbr0 start"
$ns at 1.5 "$exp1 start"
$ns at 2.5 "$cbr0 stop"
```

```

$ns at 3.5 "$ftp stop"
$ns at 4.5 "$exp1 stop"
$ns at 5 "finish"
proc plotWindow {tcpSource outfile} {
    global ns

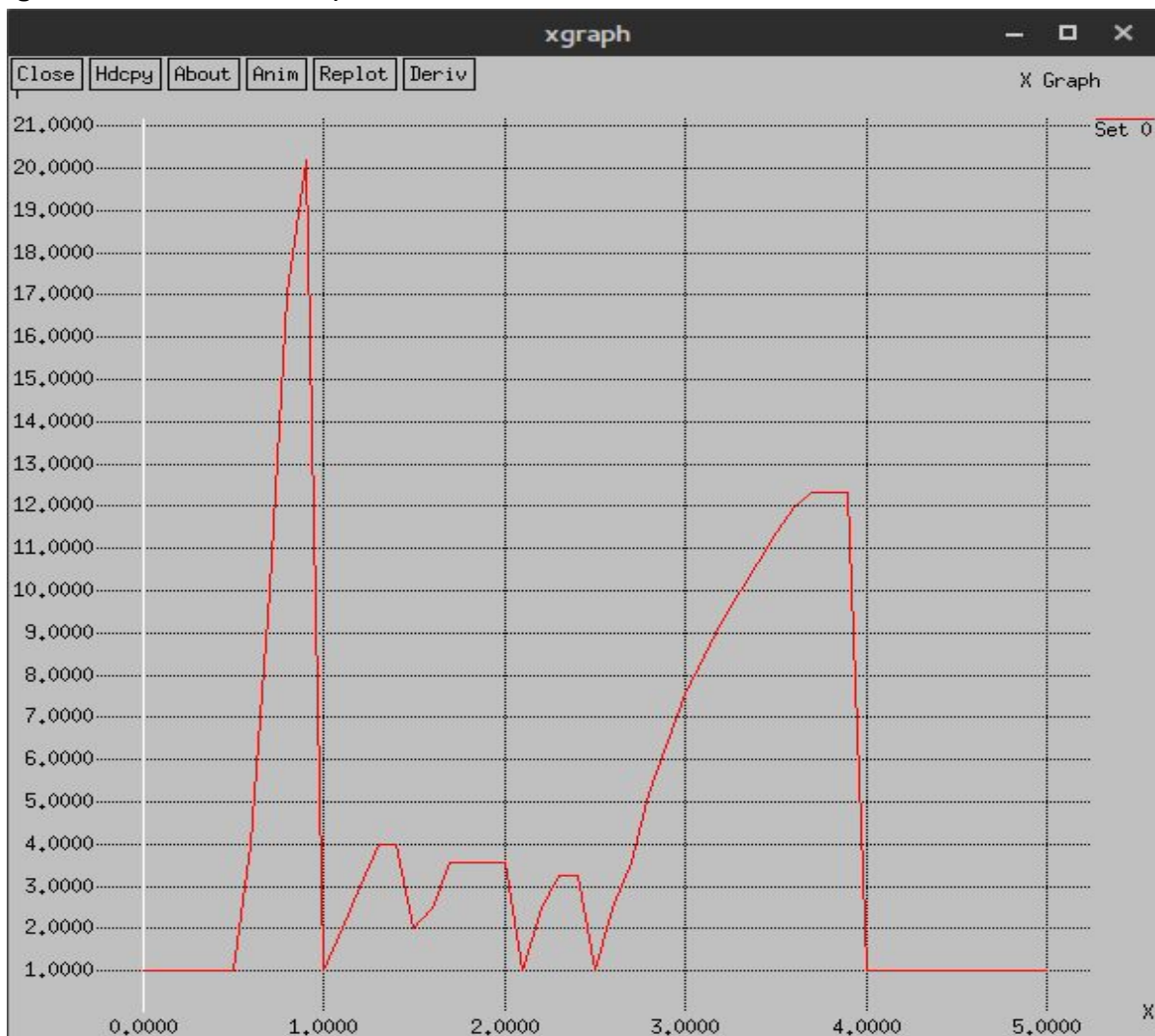
    set now [$ns now]
    set cwnd [$tcpSource set cwnd_]

    puts $outfile "$now $cwnd"

    $ns at [expr $now+0.1] "plotWindow $tcpSource $outfile"
}
$ns at 0.0 "plotWindow $tcp stdout"
$ns run

```

**Plot (congestion window vs time):**



### Comparison of graphs:

The two graphs are as expected. Both of them drop half the height if packet loss happens. The major difference in the two graphs is the peaks, in TCP Tahoe the peaks are mostly flat unlike in TCP Reno where some of them are curved upwards. This is because of the fact that TCP Reno uses fast-retransmit mechanism when it gets 3 ACK's unlike Tahoe which uses slow-start.

### d) Simulate an ethernet connection in n4

```
#Create a simulator object
set ns [new Simulator]

#Define different colors for data flows (for NAM)
$ns color 1 Blue
$ns color 2 Red
$ns color 3 Green

#Open the NAM trace file
set nf [open out.nam w]
$ns namtrace-all $nf

set nf1 [open out.tr w]
$ns trace-all $nf1
proc finish {} {
    global ns nf nf1
    #Close the NAM trace file
    close $nf
    #Close the trace-all file
    close $nf1
    #Execute NAM on the trace file
    exec nam out.nam &
    exit 0
}

#Create seven 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]
set n6 [$ns node]
set n7 [$ns node]

#Create links between the nodes
$ns duplex-link $n0 $n3 2Mb 5ms DropTail
$ns duplex-link $n1 $n3 2Mb 5ms DropTail
$ns duplex-link $n2 $n3 2Mb 5ms DropTail

$ns duplex-link $n3 $n4 600Kb 10ms DropTail
```



```
#Set Queue Size of link (n3-n4) to 10
```

```
$ns queue-limit $n3 $n4 10
```

```
#Give node position (for NAM)
```

```
$ns duplex-link-op $n3 $n0 orient left-up
```

```
$ns duplex-link-op $n3 $n2 orient left-down
```

```
$ns duplex-link-op $n3 $n1 orient left
```

```
$ns duplex-link-op $n4 $n3 orient left
```

```
#Monitor the queue for link (n3-n4). (for NAM)
```

```
$ns duplex-link-op $n3 $n4 queuePos 0.5
```

```
#Setup a TCP connection
```

```
set tcp [new Agent/TCP]
```

```
$tcp set class_ 1
```

```
$ns attach-agent $n0 $tcp
```

```
set sink [new Agent/TCPSink]
```

```
$ns attach-agent $n4 $sink
```

```
$ns connect $tcp $sink
```

```
$tcp set fid_ 1
```

```
#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 $n4 $null
```

```
$ns connect $udp $null
```

```
$udp set fid_ 2
```

```
set udp1 [new Agent/UDP]
```

```
$ns attach-agent $n2 $udp1
```

```
$ns connect $udp1 $null
```

```
$udp1 set fid_ 3
```

```
#Setup a CBR over UDP connection
```

```
set cbr0 [new Application/Traffic/CBR]
```

```
$cbr0 attach-agent $udp
```

```
$cbr0 set type_ CBR
```

```
$cbr0 set interval_ 1
```

```
$cbr0 set packet_size_ 200
```

```
$cbr0 set rate_ 500kb
```

```
$cbr0 set random_ false
```

```
set exp1 [new Application/Traffic/Exponential]
```

```
$exp1 attach-agent $udp1
```

```
$exp1 set type_ Exponential
```

```
$exp1 set packetSize_ 200
```

```
$exp1 set burst_time_ 2
```

```
$exp1 set idle_time_ 1
```

```
$exp1 set rate_ 500kb
```

```
$exp1 set random_ false
```

```
set l [$ns newLan "$n4 $n5 $n6 $n7" 0.5Mb 40ms LL Queue/DropTail MAC/Csma/Cd Channel]
```

```
$ns at 0.5 "$ftp start"
```

```
$ns at 1 "$cbr0 start"
```

```
$ns at 1.5 "$exp1 start"
```

```
$ns at 2.5 "$cbr0 stop"
```

```
$ns at 3.5 "$ftp stop"
```

```
$ns at 4.5 "$exp1 stop"
```

```
$ns at 5 "finish"
```

```
$ns run
```

Screenshot:

