2.Echo Server

pyEchoClient.py

```
import socket
HOST = '127.0.0.1'
PORT = 9999
s = socket.socket(socket.AF_INET,
socket.SOCK_STREAM)
s.connect((HOST, PORT))
str ='Hello, world'
b = str.encode('utf-8')
s.send(b)
data = s.recv(1024)
s.close()
print ('Received', data)
```

pyEchoServere.py

```
import socket
HOST = '127.0.0.1'
PORT = 9999
s = socket.socket(socket.AF_INET,
socket.SOCK_STREAM)
s.bind((HOST, PORT))
s.listen(1)
conn, addr = s.accept()
print ('Connected by', addr)
while 1:
 data = conn.recv(1024)
 print(data)
 if not data: break
 conn.send(data)
conn.close()
```

3.TCP client/server

Pyserver.py

```
import socket
def server_program():
 host = socket.gethostname()
 port = 5000
 server_socket = socket.socket()
 server_socket.bind((host, port))
server_socket.listen(2)
conn, address = server_socket.accept()
 print("Connection from: " + str(address))
 while True:
data = conn.recv(1024).decode()
 if not data:
break
 print("from connected user: " +
str(data))
 data = input(' -> ')
 conn.send(data.encode())
 conn.close()
if _name_ == '_main_':
 server_program()
```

pyclient.py

```
import socket
def client_program():
 print('type bye to terminate')
 host = socket.gethostname()
 port = 5000 # socket server port number
 client_socket = socket.socket()
 client_socket.connect((host, port))
 message = input(" -> ")
 while message.lower().strip() != 'bye':
 client_socket.send(message.encode())
 data = client_socket.recv(1024).decode()
 print('Received from server: ' + data
 message = input(" -> ")
client_socket.close()
if _name_ == '_main_':
 client_program()
```

4.Executing remote command

RemoteSender.py

```
import socket
UDP_IP = "localhost"
UDP_PORT = 8080
MESSAGE = "notepad"
print ("message:", MESSAGE)
sock = socket.socket(socket.AF_INET,
socket.SOCK_DGRAM)
sock.sendto(bytes(MESSAGE, "utf-8"),
(UDP_IP, UDP_PORT))
```

RemoteReceiver.py

```
import socket
import wmi
UDP_IP = "localhost"
UDP_PORT = 8080
sock = socket.socket(socket.AF_INET,
socket.SOCK_DGRAM)
sock.bind((UDP_IP, UDP_PORT))
while True
 data, addr = sock.recvfrom(1024)
 str=data.decode("utf-8")
 print ("Received message:", str)
 print(" opening ", str);
conn = wmi.WMI()
pid, returnval=
conn.Win32_Process.Create(CommandLi
ne=str)
```

5.ARP/RARP

ARPClient.py

```
import socket
UDP_IP = "localhost"
UDP_PORT = 8080
MESSAGE = "172.16.1.8"
print ("message:", MESSAGE)
sock = socket.socket(socket.AF_INET,
socket.SOCK_DGRAM)
sock.sendto(bytes(MESSAGE, "utf-8"),
(UDP_IP, UDP_PORT))
```

ARPserver.py

```
import socket
UDP_IP = "localhost"
UDP_PORT = 8080
sock = socket.socket(socket.AF_INET,
socket.SOCK_DGRAM)
sock.bind((UDP_IP, UDP_PORT))
ip=["172.16.1.9","172.16.1.8"]
mac=["6A:08:AA:C2","8A:BC:E3:FA"]
while True:
 data, addr = sock.recvfrom(1024)
 str1 = data.decode('utf-8')
 l = len(data)
 if l != 0:
 print("Received message:", str1)
 break
for x in ip:
 if str1 in x:
 ind=ip.index(str1)
print("the MAC address is: ",mac[ind])
```

7.FTP

Server Coding:

```
import socket
s = socket.socket()
host = socket.gethostname()
port = 8080
s.bind((host,port))
s.listen(1)
print(host)
print("Waiting for any incoming
connection... ")
conn, addr = s.accept()
print(addr, "Has connected to the
server")
filename = input(str("Enter the name of
the file to be transmitted: "))
file = open(filename , 'rb')
file_data = file.read(1024)
conn.send(file_data)
print("File has been transmitted
successfully")
```

Client Coding:

```
import socket
s = socket.socket()
host = input(str("Please enter the host
address of the sender: "))
port = 8080
s.connect((host,port))
print("Connected ... ")
filename = input(str("Please enter a
filename for the incoming file: "))
file = open(filename, 'wb') # Opens a file
for writing only in binary format
file_data = s.recv(1024)
file.write(file_data)
file.close()
print("File has been received
successfully.")
```

8.Encryption/Decryption

Encrypt.py

```
def encrypt(string, shift):
 cipher = ''
 for char in string:
 if char == ' ':
 cipher = cipher + char
 elif char.isupper():
 cipher = cipher + chr((ord(char) + shift -
65) % 26 + 65) else:
 cipher = cipher + chr((ord(char) + shift -
97) % 26 + 97)
 return cipher
text = input("enter string: ")
s = int(input("enter shift number:"))
print("original string: ", text)
print("after encryption: ", encrypt(text,
s))
```

Decrypt.py

```
def encrypt(string, shift):
 cipher = ''
 for char in string:
 if char == ' ':
 cipher = cipher + char
 elif char.isupper():
 cipher = cipher + chr((ord(char) - shift - 65) % 26 + 65)
 else:cipher = cipher + chr((ord(char) - shift - 97) % 26 + 97)return cipher
text = input("enter string: ")
s = int(input("enter shift number:"))
print("original string: ", text)
print("after encryption: ", encrypt(text, s))
```

10.Network Topology

Token Bus

```
set ns [new Simulator]
set nf [open out.nam w]
$ns namtrace-all $nf
proc finish {} {
global ns nf
$ns flush-trace
close $nf
exec nam out.nam &
exit 0}
set n0 [$ns node]
set n1 [$ns node]
set n2 [$ns node]
set n3 [$ns node]
set n4 [$ns node]
set lan0 [$ns newLan "$n0 $n1 $n2 $n3 $n4" 0.5Mb 40ms LL Queue/DropTail MAC/Csma/Cd
Channel]
set tcp0 [new Agent/TCP]
$tcp0 set class_ 1
$ns attach-agent $n1 $tcp0
set sink0 [new Agent/TCPSink]
$ns attach-agent $n3 $sink0
$ns connect $tcp0 $sink0
set cbr0 [new Application/Traffic/CBR]
$cbr0 set packetSize_ 500
$cbr0 set interval_ 0.01
$cbr0 attach-agent $tcp0
$ns at 0.5 "$cbr0 start"
$ns at 4.5 "$cbr0 stop"
$ns at 5.0 "finish"
$ns run
```

Token Ring

```
set ns [new Simulator]
set nf [open out.nam w]
$ns namtrace-all $nf
proc finish {} {
global ns nf
$ns flush-trace
close $nf
exec nam out.nam &
exit 0}
set n0 [$ns node]
set n1 [$ns node]
set n2 [$ns node]
set n3 [$ns node]
set n4 [$ns node]
set n5 [$ns node]
$ns duplex-link $n0 $n1 1Mb 10ms DropTail
$ns duplex-link $n1 $n2 1Mb 10ms DropTail
$ns duplex-link $n2 $n3 1Mb 10ms DropTail
$ns duplex-link $n3 $n4 1Mb 10ms DropTail
$ns duplex-link $n4 $n5 1Mb 10ms DropTail
$ns duplex-link $n5 $n0 1Mb 10ms DropTail
set tcp0 [new Agent/TCP]
$tcp0 set class_ 1
$ns attach-agent $n1 $tcp0
set sink0 [new Agent/TCPSink]
$ns attach-agent $n3 $sink0
$ns connect $tcp0 $sink0
set cbr0 [new Application/Traffic/CBR]
$cbr0 set packetSize_ 500
$cbr0 set interval_ 0.01
$cbr0 attach-agent $tcp0
$ns at 0.5 "$cbr0 start"
$ns at 4.5 "$cbr0 stop"
$ns at 5.0 "finish"
$ns run
```

Token Star :

```
set ns [new Simulator]
set nf [open out.nam w]
$ns namtrace-all $nf
proc finish {} {
global ns nf
$ns flush-trace
close $nf
exec nam out.nam &
exit0}
set n0 [$ns node]
set n1 [$ns node]
set n2 [$ns node]
set n3 [$ns node]
set n4 [$ns node]
set n5 [$ns node]
$n0 shape square
$ns duplex-link $n0 $n1 1Mb 10ms DropTail
$ns duplex-link $n0 $n2 1Mb 10ms DropTail
$ns duplex-link $n0 $n3 1Mb 10ms DropTail
$ns duplex-link $n0 $n4 1Mb 10ms DropTail
$ns duplex-link $n0 $n5 1Mb 10ms DropTail
set tcp0 [new Agent/TCP]
$tcp0 set class_ 1
$ns attach-agent $n1 $tcp0
set sink0 [new Agent/TCPSink]
$ns attach-agent $n3 $sink0
$ns connect $tcp0 $sink0
set cbr0 [new Application/Traffic/CBR]
$cbr0 set packetSize_ 500
$cbr0 set interval_ 0.01
$cbr0 attach-agent $tcp0
$ns at 0.5 "$cbr0 start"
$ns at 4.5 "$cbr0 stop"
$ns at 5.0 "finish"
$ns run
```

11.a)Go Back N Protocol

```
set ns [new Simulator]
set n0 [$ns node]
set n1 [$ns node]
set n2 [$ns node]
set n3 [$ns node]
set n4 [$ns node]
set n5 [$ns node]
$n0 color "purple"
$n1 color "purple"
$n2 color "violet"
$n3 color "violet"
$n4 color "chocolate"
$n5 color "chocolate"
$n0 shape box ;
$n1 shape box ;
$n2 shape box ;
$n3 shape box ;
$n4 shape box ;
$n5 shape box ;
$ns at 0.0 "$n0 label SYS0"
$ns at 0.0 "$n1 label SYS1"
$ns at 0.0 "$n2 label SYS2"
$ns at 0.0 "$n3 label SYS3"
$ns at 0.0 "$n4 label SYS4"
$ns at 0.0 "$n5 label SYS5"
set nf [open goback.nam w]
$ns namtrace-all $nf
set f [open goback.tr w]
$ns trace-all $f
$ns duplex-link $n0 $n2 1Mb 20ms DropTail
$ns duplex-link-op $n0 $n2 orient right-down
$ns queue-limit $n0 $n2 5
$ns duplex-link $n1 $n2 1Mb 20ms DropTail
$ns duplex-link-op $n1 $n2 orient right-up
$ns duplex-link $n2 $n3 1Mb 20ms DropTail
$ns duplex-link-op $n2 $n3 orient right
$ns duplex-link $n3 $n4 1Mb 20ms DropTail
$ns duplex-link-op $n3 $n4 orient right-up
$ns duplex-link $n3 $n5 1Mb 20ms DropTail
$ns duplex-link-op $n3 $n5 orient right-down
Agent/TCP set_nam_tracevar_true
set tcp [new Agent/TCP]
$tcp set fid 1
$ns attach-agent $n1 $tcp
set sink [new Agent/TCPSink]
$ns attach-agent $n4 $sink
$ns connect $tcp $sink
set ftp [new Application/FTP]
$ftp attach-agent $tcp
$ns at 0.05 "$ftp start"
```

```
$ns at 0.06 "$tcp set windowlnit 6"
$ns at 0.06 "$tcp set maxcwnd 6"
$ns at 0.25 "$ns queue-limit $n3 $n4 0"
$ns at 0.26 "$ns queue-limit $n3 $n4 10"
$ns at 0.305 "$tcp set windowlnit 4"
$ns at 0.305 "$tcp set maxcwnd 4"
$ns at 0.368 "$ns detach-agent
$n1 $tcp ; $ns detach-agent $n4 $sink"
$ns at 1.5 "finish"
$ns at 0.0 "$ns trace-annotate
 \"Goback N end\""
$ns at 0.05 "$ns trace-annotate
 \"FTP starts at 0.01\""
$ns at 0.06 "$ns trace-annotate
\"Send 6Packets from SYS1 to SYS4\""
$ns at 0.26 "$ns trace-annotate
 \"Error Occurs for 4th packet
 so not sent ack for the Packet\""
$ns at 0.30 "$ns trace-annotate
\"Retransmit Packet_4 to 6\""
$ns at 1.0 "$ns trace-annotate \"FTP
stops\""
proc finish {} {
global ns nf
$ns flush-trace
close $nf
puts "filtering..."
#exec tclsh../bin/namfilter.tcl
goback.nam
#puts "running nam..."
exec nam goback.nam &
exit 0
}$ns run
```

11.b)Selective Repeat Protocol

```
set ns [new Simulator]
set n0 [$ns node]
set n1 [$ns node]
set n2 [$ns node]
set n3 [$ns node]
set n4 [$ns node]
set n5 [$ns node]
$n0 color "red"
$n1 color "red"
$n2 color "green"
$n3 color "green"
$n4 color "black"
$n5 color "black"
$n0 shape circle ;
$n1 shape circle ;
$n2 shape circle ;
$n3 shape circle ;
$n4 shape circle ;
$n5 shape circle ;
$ns at 0.0 "$n0 label SYS1"
$ns at 0.0 "$n1 label SYS2"
$ns at 0.0 "$n2 label SYS3"
$ns at 0.0 "$n3 label SYS4"
$ns at 0.0 "$n4 label SYS5"
$ns at 0.0 "$n5 label SYS6"
set nf [open Srepeat.nam w]
$ns namtrace-all $nf
set f [open Srepeat.tr w]
$ns trace-all $f
$ns duplex-link $n0 $n2 1Mb 10ms
DropTail
$ns duplex-link-op $n0 $n2 orient right-
down
$ns queue-limit $n0 $n2 5
$ns duplex-link $n1 $n2 1Mb 10ms
DropTail
$ns duplex-link-op $n1 $n2 orient right-
up
$ns duplex-link $n2 $n3 1Mb 10ms
DropTail
$ns duplex-link-op $n2 $n3 orient right
$ns duplex-link $n3 $n4 1Mb 10ms
DropTail
$ns duplex-link-op $n3 $n4 orient right-
up
$ns duplex-link $n3 $n5 1Mb 10ms
DropTail
$ns duplex-link-op $n3 $n5 orient right-
down
Agent/TCP set_nam_tracevar_true
set tcp [new Agent/TCP]
$tcp set fid 1
$ns attach-agent $n1 $tcp
set sink [new Agent/TCPSink]
$ns attach-agent $n4 $sink
$ns connect $tcp $sink
set ftp [new Application/FTP]
$ftp attach-agent $tcp
$ns at 0.05 "$ftp start"
$ns at 0.06 "$tcp set windowlnit 8"
$ns at 0.06 "$tcp set maxcwnd 8"
$ns at 0.25 "$ns queue-limit $n3 $n4 0"
$ns at 0.26 "$ns queue-limit $n3 $n4 10"
$ns at 0.30 "$tcp set windowlnit 1"
$ns at 0.30 "$tcp set maxcwnd 1"
$ns at 0.30 "$ns queue-limit $n3 $n4 10"
$ns at 0.47 "$ns detach-agent $n1
$tcp;$ns detach-agent $n4 $sink"
$ns at 1.75 "finish"
$ns at 0.0 "$ns trace-annotate
\"Select and repeat\""
$ns at 0.05 "$ns trace-annotate
\"FTP starts at 0.01\""
$ns at 0.06 "$ns trace-annotate
\"Send 8Packets from SYS1 to SYS4\""
$ns at 0.26 "$ns trace-annotate
\"Error Occurs in 4th packet \""
$ns at 0.30 "$ns trace-annotate
\"Retransmit Packet_4
 from SYS1 to SYS4\""
$ns at 1.5 "$ns trace-annotate
\"FTP stops\""
proc finish {} {
global ns nf
$ns flush-trace
close $nf
puts "filtering..."
#exec tclsh../bin/namfilter.tcl
srepeat.nam
#puts "running nam..."
exec nam Srepeat.nam &
exit 0}
$ns run
```

12.TCP/IP Data Transfer

```
set ns [new Simulator]
set nf [open out.nam w]
$ns namtrace-all $nf
set tr [open out.tr w]
$ns trace-all $tr
proc finish {} {
global nf ns tr
$ns flush-trace
close $tr
exec nam out.nam &
exit 0
}
set n0 [$ns node]
set n1 [$ns node]
set n2 [$ns node]
set n3 [$ns node]
$ns duplex-link $n0 $n1 10Mb 10ms
DropTail
$ns duplex-link $n1 $n3 10Mb 10ms
DropTail
$ns duplex-link $n2 $n1 10Mb 10ms
DropTail
$ns duplex-link-op $n0 $n1 orient right-
down
$ns duplex-link-op $n1 $n3 orient right
$ns duplex-link-op $n2 $n1 orient right-
up
set tcp [new Agent/TCP]
$ns attach-agent $n0 $tcp
set ftp [new Application/FTP]
$ftp attach-agent $tcp
set sink [new Agent/TCPSink]
$ns attach-agent $n3 $sink
set udp [new Agent/UDP]
$ns attach-agent $n2 $udp
set cbr [new Application/Traffic/CBR]
$cbr attach-agent $udp
set null [new Agent/Null]
$ns attach-agent $n3 $null
$ns connect $tcp $sink
$ns connect $udp $null
$ns rtmodel-at 1.0 down $n1 $n3
$ns rtmodel-at 2.0 up $n1 $n3
$ns rtproto DV
$ns at 0.0 "$ftp start"
$ns at 0.0 "$cbr start"
$ns at 5.0 "finish"
$ns run
```

13.a) Program:DV.tcl

```
set ns [new Simulator]
set nr [open thro.tr w]
$ns trace-all $nr
set nf [open thro.nam w]
$ns namtrace-all $nf
proc finish { } {
 global ns nr nf
 $ns flush-trace
 close $nf
 close $nr
 exec nam thro.nam &
 exit 0
}
for { set i 0 } { $i< 12} { incr i 1 } {
```

```
set n($i) [$ns node]}
for {set i 0} {$i< 8} {incr i} {
$ns duplex-link $n($i) $n([expr $i+1])
1Mb 10ms DropTail }
$ns duplex-link $n(0) $n(8) 1Mb 10ms
DropTail
$ns duplex-link $n(1) $n(10) 1Mb 10ms
DropTail
$ns duplex-link $n(0) $n(9) 1Mb 10ms
DropTail
$ns duplex-link $n(9) $n(11) 1Mb 10ms
DropTail
$ns duplex-link $n(10) $n(11) 1Mb 10ms
DropTail
$ns duplex-link $n(11) $n(5) 1Mb 10ms
DropTail
set udp0 [new Agent/UDP]
$ns attach-agent $n(0) $udp0
set cbr0 [new Application/Traffic/CBR]
$cbr0 set packetSize_ 500
$cbr0 set interval_ 0.005
$cbr0 attach-agent $udp0
set null0 [new Agent/Null]
$ns attach-agent $n(5) $null0
$ns connect $udp0 $null0
set udp1 [new Agent/UDP]
$ns attach-agent $n(1) $udp1
set cbr1 [new Application/Traffic/CBR]
$cbr1 set packetSize_ 500
$cbr1 set interval_ 0.005
$cbr1 attach-agent $udp1
set null0 [new Agent/Null]
$ns attach-agent $n(5) $null0
$ns connect $udp1 $null0
$ns rtproto DV
$ns rtmodel-at 10.0 down $n(11) $n(5)
$ns rtmodel-at 15.0 down $n(7) $n(6)
$ns rtmodel-at 30.0 up $n(11) $n(5)
$ns rtmodel-at 20.0 up $n(7) $n(6)
$udp0 set fid_ 1
$udp1 set fid_ 2
$ns color 1 Red
$ns color 2 Green
$ns at 1.0 "$cbr0 start"
$ns at 2.0 "$cbr1 start"
$ns at 45 "finish"
$ns run
```

13.b)Linkstate routing algorithm

```
set ns [new Simulator]
set nr [open thro.tr w]
$ns trace-all $nr
set nf [open thro.nam w]
$ns namtrace-all $nf
proc finish { } {
global ns nr nf
$ns flush-trace
close $nf
close $nr
exec nam thro.nam &
exit 0
}
for { set i 0 } { $i< 12} { incr i 1 } {
set n($i) [$ns node]}
for {set i 0} {$i< 8} {incr i} {
$ns duplex-link $n($i) $n([expr $i+1])
1Mb 10ms DropTail }
$ns duplex-link $n(0) $n(8) 1Mb 10ms
DropTail
$ns duplex-link $n(1) $n(10) 1Mb 10ms
DropTail
$ns duplex-link $n(0) $n(9) 1Mb 10ms
DropTail
$ns duplex-link $n(9) $n(11) 1Mb 10ms
DropTail
$ns duplex-link $n(10) $n(11) 1Mb 10ms
DropTail
$ns duplex-link $n(11) $n(5) 1Mb 10ms
DropTail
set udp0 [new Agent/UDP]
$ns attach-agent $n(0) $udp0
set cbr0 [new Application/Traffic/CBR]
$cbr0 set packetSize_ 500
$cbr0 set interval_ 0.005
$cbr0 attach-agent $udp0
set null0 [new Agent/Null]
$ns attach-agent $n(5) $null0
$ns connect $udp0 $null0
set udp1 [new Agent/UDP]
$ns attach-agent $n(1) $udp1
set cbr1 [new Application/Traffic/CBR]
$cbr1 set packetSize_ 500
$cbr1 set interval_ 0.005
$cbr1 attach-agent $udp1
set null0 [new Agent/Null]
$ns attach-agent $n(5) $null0
$ns connect $udp1 $null0
$ns rtproto LS
$ns rtmodel-at 10.0 down $n(11) $n(5)
$ns rtmodel-at 15.0 down $n(7) $n(6)
$ns rtmodel-at 30.0 up $n(11) $n(5)
$ns rtmodel-at 20.0 up $n(7) $n(6)
$udp0 set fid_ 1
$udp1 set fid_ 2
$ns color 1 Red
$ns color 2 Green
$ns at 1.0 "$cbr0 start"
$ns at 2.0 "$cbr1 start"
$ns at 45 "finish"
$ns run
```

14.MobileAdhoc Network

```
set val(chan) Channel/WirelessChannel
set val(prop)
Propagation/TwoRayGround
set val(netif) Phy/WirelessPhy
set val(mac) Mac/802_11
set val(ifq) Queue/DropTail/PriQueue
set val(ll) LL
set val(ant) Antenna/OmniAntenna
set val(ifqlen) 50
set val(nn) 3
set val(rp) DSDV
set ns [new Simulator]
set tf [open output.tr w]
$ns trace-all $tf
set tf1 [open output.nam w]
$ns namtrace-all-wireless $tf1 100 100
set topo [new Topography]
$topo load_flatgrid 100 100
create-god $val(nn)
$ns node-config -adhoc Routing $val(rp)
\
-llType $val(ll) \
-macType $val(mac) \
-ifqType $val(ifq) \
-ifqLen $val(ifqlen) \
-antType $val(ant) \
-propType $val(prop) \
-phyType $val(netif) \
-channelType $val(chan) \
-topoInstance $topo \
-agentTrace ON \
-routerTrace OFF \
-macTrace OFF \
-movementTrace OFF
set node0 [$ns node]
set node1 [$ns node]
set node2 [$ns node]
$ns initial_node_pos $node0 10
$ns initial_node_pos $node1 10
$ns initial_node_pos $node2 10
$node0 set X_ 25.0
$node0 set Y_ 50.0
$node0 set Z_ 0.0
$node1 set X_ 50.0
$node1 set Y_ 50.0
$node1 set Z_ 0.0
$node2 set X_ 65.0
$node2 set Y_ 50.0
$node2 set Z_ 0.0
set tcp1 [new Agent/TCP]
$ns attach-agent $node0 $tcp1
set ftp [new Application/FTP]
$ftp attach-agent $tcp1
set sink1 [new Agent/TCPSink]
$ns attach-agent $node2 $sink1
$ns connect $tcp1 $sink1
$ns at 10.0 "$node1 set dest 50.0 90.0
0.0"
$ns at 50.0 "$node1 set dest 50.0 10.0
0.0"
$ns at 0.5 "$ftp start"
$ns at 1000 "$ftp stop"
$ns at 1000 "finish"
proc finish {} {
global ns tf tf1
$ns flush-trace
close $tf
exec nam output.nam &
exit 0}
$ns run
```

15.TCP in Sensor Network

```
set ns [new Simulator]
$ns color 1 Blue
$ns color 2 Red
#Open the Trace files
set file1 [open out.tr w]
set winfile [open WinFile w]
$ns trace-all $file1
set file2 [open out.nam w]
$ns namtrace-all $file2
proc finish {} {
global ns file1 file2
```

```
$ns flush-trace
close $file1
close $file2
exec nam out.nam &
exit 0}
set n0 [$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
$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]
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
$ns at 1.0 "$ftp start"
$ns at 124.0 "$ftp stop"
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\""
$ns at 125.0 "finish"
$ns run
```