```
                        if (rcv_pkt.sq_no==1)
                {  printf("Packet received with seq. no.=1 %d and Packet content is=
%s\n",rcv_pkt.sq_no, rcv_pkt.data);
                                ack_pkt.sq_no = 1;
                        if (sendto(s, &ack_pkt, recv_len, 0, (struct sockaddr*) &si_other, slen) ==
-1)
                  {
                     die("sendto()");
              }
                    state = 0;
                    break;
                  }
                  else if(rcv_pkt.sq_no==0){
                    ack_pkt.sq_no = 0;
                                                    if (sendto(s, &ack_pkt, recv_len, 0, (struct
sockaddr*) &si_other, slen) == -1)
                                                            {
                                                               die("sendto()");
                                                      }
                    break;
                  }

                        }
          }

    }

    close(s);
    return 0;
}
```

---------- Lab 6 ----------

example.tcl

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

#Open the output files
set f0 [open out0.tr w]
set f1 [open out1.tr w]
```

```
set f2 [open out2.tr w]
set nf [open out.nam w]
$ns namtrace-all $nf
set tf [open outall.tr w]
$ns trace-all $tf

#Create 5 nodes
set n0 [$ns node]
set n1 [$ns node]
set n2 [$ns node]
set n3 [$ns node]
set n4 [$ns node]

#Connect the nodes
$ns duplex-link $n0 $n3 1Mb 100ms DropTail
$ns duplex-link $n1 $n3 1Mb 100ms DropTail
$ns duplex-link $n2 $n3 1Mb 100ms DropTail
$ns duplex-link $n3 $n4 1Mb 100ms DropTail

#Define a 'finish' procedure
proc finish {} {
        global f0 f1 f2 nf tf
        #Close the trace and output files
    close $nf
        close $tf
        close $f0
        close $f1
        close $f2
        #Call xgraph to display the results
        exec xgraph out0.tr out1.tr out2.tr -geometry 800x400 &
        #Execute nam on the trace file
    exec nam out.nam &
    exit 0
}


#Define a procedure that attaches a UDP agent to a previously created node
#'node' and attaches an Expoo traffic generator to the agent with the
#characteristic values 'size' for packet size 'burst' for burst time,
#'idle' for idle time and 'rate' for burst peak rate. The procedure connects
#the source with the previously defined traffic sink 'sink' and returns the
#source object.
proc attach-expoo-traffic { node sink size burst idle rate } {
```

```
        #Get an instance of the simulator
        set ns [Simulator instance]

        #Create a UDP agent and attach it to the node
        set source [new Agent/UDP]
        $ns attach-agent $node $source

        #Create an Expoo traffic agent and set its configuration parameters
        set traffic [new Application/Traffic/Exponential]
        $traffic set packetSize_ $size
        $traffic set burst_time_ $burst
        $traffic set idle_time_ $idle
        $traffic set rate_ $rate

    # Attach traffic source to the traffic generator
    $traffic attach-agent $source
        #Connect the source and the sink
        $ns connect $source $sink
        return $traffic
}


#Define a procedure which periodically records the bandwidth received by the
#three traffic sinks sink0/1/2 and writes it to the three files f0/1/2.
proc record {} {
    global sink0 sink1 sink2 f0 f1 f2
        #Get an instance of the simulator
        set ns [Simulator instance]
        #Set the time after which the procedure should be called again
    set time 2
        #How many bytes have been received by the traffic sinks?
    set bw0 [$sink0 set bytes_]
    set bw1 [$sink1 set bytes_]
    set bw2 [$sink2 set bytes_]
        #Get the current time
    set now [$ns now]
        #Calculate the bandwidth (in MBit/s) and write it to the files
    puts $f0 "$now [expr $bw0/$time*8/1000000]"
    puts $f1 "$now [expr $bw1/$time*8/1000000]"
    puts $f2 "$now [expr $bw2/$time*8/1000000]"
        #Reset the bytes_ values on the traffic sinks
    $sink0 set bytes_ 0
    $sink1 set bytes_ 0
```

```
        $sink2 set bytes_ 0
           #Re-schedule the procedure
        $ns at [expr $now+$time] "record"
}


#Create three traffic sinks and attach them to the node n4
set sink0 [new Agent/LossMonitor]
set sink1 [new Agent/LossMonitor]
set sink2 [new Agent/LossMonitor]
$ns attach-agent $n4 $sink0
$ns attach-agent $n4 $sink1
$ns attach-agent $n4 $sink2

#Create three traffic sources
set source0 [attach-expoo-traffic $n0 $sink0 400 8s 1s 100k]
set source1 [attach-expoo-traffic $n1 $sink1 200 2s 1s 200k]
set source2 [attach-expoo-traffic $n2 $sink2 200 2s 1s 300k]

#Start logging the received bandwidth
$ns at 0.0 "record"
#Start the traffic sources
$ns at 10.0 "$source0 start"
$ns at 10.0 "$source1 start"
$ns at 10.0 "$source2 start"
#Stop the traffic sources
$ns at 50.0 "$source0 stop"
$ns at 50.0 "$source1 stop"
$ns at 50.0 "$source2 stop"
#Call the finish procedure after 60 seconds simulation time
$ns at 60.0 "finish"

#Run the simulation
$ns run
```

Example2.tcl

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

#Open the output files
set f0 [open out_0.tr w]
set f1 [open out_1.tr w]
```

```
set nf [open out_x.nam w]
$ns namtrace-all $nf
set tf [open outall_y.tr w]
$ns trace-all $tf

#Create 4 nodes
set n0 [$ns node]
set n1 [$ns node]
set n2 [$ns node]
set n3 [$ns node]

#Connect the nodes
$ns duplex-link $n0 $n2 2Mb 100ms DropTail
$ns duplex-link $n1 $n2 2Mb 100ms DropTail
$ns duplex-link $n2 $n3 2Mb 100ms DropTail

#Define a 'finish' procedure
proc finish {} {
        global f0 f1 nf tf
        #Close the trace and output files
   close $nf
        close $tf
        close $f0
        close $f1

        #Call xgraph to display the results
        exec xgraph out_0.tr out_1.tr -geometry 800x400 &
        #Execute nam on the trace file
      exec nam out_x.nam &
      exit 0
}

#Define a procedure which periodically records the bandwidth received by the
#three traffic sinks sink0/1/2 and writes it to the three files f0/1/2.
proc record {} {
      global sink0 sink1 f0 f1 udp0
        #Get an instance of the simulator
        set ns [Simulator instance]
        #Set the time after which the procedure should be called again
      set time 0.1
        #How many bytes have been received by the traffic sinks?
      set bw0 [$sink0 set bytes_]
```

```
        set bw1 [$sink1 set bytes_]

           #Get the current time
        set now [$ns now]
           #Calculate the bandwidth (in MBit/s) and write it to the files
        puts $f0 "$now [expr $bw0/$time*8/1000000]"
        puts $f1 "$now [expr $bw1/$time*8/1000000]"

           #Reset the bytes_ values on the traffic sinks
        $sink0 set bytes_ 0
        $sink1 set bytes_ 0
           $udp0 set bytes_ 0
           #Re-schedule the procedure
        $ns at [expr $now+$time] "record"
}

#Create UDP-CBR agent for n0
set udp0 [new Agent/UDP]
$ns attach-agent $n0 $udp0

set cbr0 [new Application/Traffic/CBR]
$cbr0 set packetSize_ 800
$cbr0 set rate 0.005
$cbr0 attach-agent $udp0

#Create TCP-FTP agent for n1
set tcp1 [new Agent/TCP]
$ns attach-agent $n1 $tcp1

set ftp1 [new Application/FTP]
$ftp1 attach-agent $tcp1
$tcp1 set packet_size_ 500

#Create three traffic sinks and attach them to the node n4
set sink0 [new Agent/LossMonitor]
set sink1 [new Agent/TCPSink]
$ns attach-agent $n3 $sink0
$ns attach-agent $n3 $sink1

$ns connect $udp0 $sink0
$ns connect $tcp1 $sink1
```

```
#Start logging the received bandwidth
$ns at 0.0 "record"
#Start the traffic sources
$ns at 0.0 "$ftp1 start"
$ns at 5.0 "$cbr0 start"
$ns at 10.0 "$ftp1 stop"
$ns at 10.0 "$cbr0 stop"
#Call the finish procedure after 10 seconds simulation time
$ns at 10.0 "finish"

#Run the simulation
$ns run
```

---------- **Lab 7** ----------

exercise1

```
set ns [new Simulator]

set nf [open out.nam w]
$ns namtrace-all $nf

set f0 [open throughput.tr w]
set f1 [open conw.tr w]

# 3 nodes were created
set n0 [$ns node]
set n1 [$ns node]
set n2 [$ns node]

# Links
$ns duplex-link $n0 $n1 2Mb 100ms DropTail
$ns duplex-link $n1 $n2 2Mb 100ms DropTail

proc finish {} {

        global ns nf f0 f1
        $ns flush-trace

        close $nf
        close $f0
        close $f1

        exec xgraph throughput.tr -geometry 800x800 &
```

```
        exec xgraph conw.tr -geometry 800x800 &

        exec nam out.nam &
        exit 0
}

proc record {} {

        # global variable
        global ns f0 sink2

        # time
        set time .1
        set now [$ns now]

        # bandwidth
        set bw0 [$sink2 set bytes_]

        # Write in the file
        puts $f0 "$now [expr ($bw0/($time*8*1000000))]"
        $sink2 set bytes_ 0

        # Recall
        $ns at [expr $now + $time] "record"
}

proc congest {} {

        global f1 tcp0 ns

        # time
        set time .001
        set now [$ns now]

        set cwnd [$tcp0 set cwnd_]

        puts $f1 "$now [expr $cwnd]"

        $ns at [expr $now + $time] "congest"
}

# TCP Agent
set tcp0 [new Agent/TCP]
```

```
# Agent with node
$ns attach-agent $n0 $tcp0

# Application
set ftp [new Application/FTP]

# Agent with application
$ftp attach-agent $tcp0
$ftp set maxpkts_ 100

# Sink Agent
set sink2 [new Agent/TCPSink]
# Agent with node
$ns attach-agent $n2 $sink2

#Sink with sender
$ns connect $tcp0 $sink2


$ns at 0.0 "record"
$ns at 0.0 "congest"

$ns at 1.0 "$ftp start"
$ns at 30.0 "$ftp stop"
$ns at 30.0 "finish"

$ns run
```

exercise2.tcl

```
set ns [new Simulator]

$ns color 1 Red
$ns color 2 Blue

set nf [open out.nam w]
$ns namtrace-all $nf

set f0 [open throughput1.tr w]
set f1 [open conw1.tr w]
set f2 [open throughput2.tr w]
set f3 [open conw2.tr w]
```

```
# 3 nodes were created
set n0 [$ns node]
set n1 [$ns node]
set n2 [$ns node]
set n3 [$ns node]

# Links
$ns duplex-link $n0 $n1 2Mb 100ms DropTail
$ns duplex-link $n1 $n2 2Mb 100ms DropTail
$ns duplex-link $n3 $n1 2Mb 100ms DropTail

proc finish {} {

        global ns nf f0 f1 f2 f3
        $ns flush-trace

        close $nf
        close $f0
        close $f1
        close $f2
        close $f3

        exec xgraph throughput1.tr throughput2.tr -geometry 800x800 &
        exec xgraph conw1.tr conw2.tr -geometry 800x800 &

        exec nam out.nam &
        exit 0
}

proc record {} {

        # global variable
        global ns f0 f2 sink2 sink1

        # time
        set time 0.75
        set now [$ns now]

        # bandwidth
        set bw0 [$sink1 set bytes_]
        set bw2 [$sink2 set bytes_]
        # Write in the file
```

```
        puts $f0 "$now [expr ($bw0/($time*8*1000000))]"
        puts $f2 "$now [expr ($bw2/($time*8*1000000))]"
        $sink1 set bytes_ 0
        $sink2 set bytes_ 0

        # Recall
        $ns at [expr $now + $time] "record"
}

proc congest {} {

        global f1 f3 tcp0 tcp1 ns

        # time
        set time .1
        set now [$ns now]

        set cwnd1 [$tcp0 set cwnd_]
        set cwnd2 [$tcp1 set cwnd_]

        puts $f1 "$now [expr $cwnd1]"
        puts $f3 "$now [expr $cwnd2]"

        $ns at [expr $now + $time] "congest"
}

# TCP Agent
set tcp0 [new Agent/TCP]
set tcp1 [new Agent/TCP]
# Agent with node
$ns attach-agent $n0 $tcp0
$ns attach-agent $n3 $tcp1


# Application
set ftp1 [new Application/FTP]
set ftp2 [new Application/FTP]

# Agent with application
$ftp1 attach-agent $tcp0
$tcp0 set fid_ 1
#$ftp1 set maxpkts_ 100
$ftp2 attach-agent $tcp1
```

```
$tcp1 set fid_ 2

# Sink Agent
set sink1 [new Agent/TCPSink]
set sink2 [new Agent/TCPSink]

# Agent with node
$ns attach-agent $n2 $sink2
$ns attach-agent $n2 $sink1

#Sink with sender
$ns connect $tcp0 $sink1
$ns connect $tcp1 $sink2

$ns at 0.0 "record"
$ns at 0.0 "congest"

$ns at 1.0 "$ftp1 start"
$ns at 10.0 "$ftp2 start"
$ns at 30.0 "$ftp1 stop"
$ns at 30.0 "$ftp2 stop"
$ns at 30.0 "finish"

$ns run
```

exercise2_1.tcl

```
set ns [new Simulator]

$ns color 1 Red
$ns color 2 Blue

set nf [open out.nam w]
$ns namtrace-all $nf

set f0 [open throughput1.tr w]
set f1 [open conw1.tr w]
set f2 [open throughput2.tr w]
set f3 [open conw2.tr w]

# 3 nodes were created
set n0 [$ns node]
set n1 [$ns node]
```

```
set n2 [$ns node]
set n3 [$ns node]
set n4 [$ns node]

# Links
$ns duplex-link $n0 $n1 2Mb 100ms DropTail
$ns duplex-link $n1 $n2 2Mb 100ms DropTail
$ns duplex-link $n3 $n1 2Mb 100ms DropTail
$ns duplex-link $n4 $n0 2Mb 100ms DropTail

proc finish {} {

        global ns nf f0 f1 f2 f3
        $ns flush-trace

        close $nf
        close $f0
        close $f1
        close $f2
        close $f3

        exec xgraph throughput1.tr throughput2.tr -geometry 800x800 &
        exec xgraph conw1.tr conw2.tr -geometry 800x800 &

        exec nam out.nam &
        exit 0
}

proc record {} {

        # global variable
        global ns f0 f2 sink2 sink1

        # time
        set time 0.5
        set now [$ns now]

        # bandwidth
        set bw0 [$sink1 set bytes_]
        set bw2 [$sink2 set bytes_]
        # Write in the file
        puts $f0 "$now [expr ($bw0/($time*8*1000000))]"
        puts $f2 "$now [expr ($bw2/($time*8*1000000))]"
```

```
        $sink1 set bytes_ 0
        $sink2 set bytes_ 0

        # Recall
        $ns at [expr $now + $time] "record"
}

proc congest {} {

        global f1 f3 tcp0 tcp1 ns

        # time
        set time .001
        set now [$ns now]

        set cwnd1 [$tcp0 set cwnd_]
        set cwnd2 [$tcp1 set cwnd_]

        puts $f1 "$now [expr $cwnd1]"
        puts $f3 "$now [expr $cwnd2]"

        $ns at [expr $now + $time] "congest"
}

# TCP Agent
set tcp0 [new Agent/TCP]
set tcp1 [new Agent/TCP]
# Agent with node
$ns attach-agent $n4 $tcp0
$ns attach-agent $n3 $tcp1


# Application
set ftp1 [new Application/FTP]
set ftp2 [new Application/FTP]

# Agent with application
$ftp1 attach-agent $tcp0
$tcp0 set fid_ 1
#$ftp1 set maxpkts_ 100
$ftp2 attach-agent $tcp1
$tcp1 set fid_ 2
```

```
# Sink Agent
set sink1 [new Agent/TCPSink]
set sink2 [new Agent/TCPSink]

# Agent with node
$ns attach-agent $n2 $sink2
$ns attach-agent $n2 $sink1

#Sink with sender
$ns connect $tcp0 $sink1
$ns connect $tcp1 $sink2

$ns at 0.0 "record"
$ns at 0.0 "congest"

$ns at 1.0 "$ftp1 start"
$ns at 10.0 "$ftp2 start"
$ns at 30.0 "$ftp1 stop"
$ns at 30.0 "$ftp2 stop"
$ns at 30.0 "finish"

$ns run
```

exercise3.tcl

```
set ns [new Simulator]

$ns color 1 Red
$ns color 2 Blue

set nf [open out.nam w]
$ns namtrace-all $nf

set f0 [open throughput1.tr w]
set f1 [open conw1.tr w]
set f2 [open throughput2.tr w]
set f3 [open conw2.tr w]
set tf [open outall.tr w]
$ns trace-all $tf

set total_tcp_bytes 0
set total_udp_bytes 0
```

```
# 3 nodes were created
set n0 [$ns node]
set n1 [$ns node]
set n2 [$ns node]
set n3 [$ns node]

# Links
$ns duplex-link $n0 $n1 2Mb 100ms DropTail
$ns duplex-link $n1 $n2 2Mb 100ms DropTail
$ns duplex-link $n3 $n1 2Mb 100ms DropTail

proc finish {} {

        global ns nf f0 f1 f2 f3
        $ns flush-trace
        global total_udp_bytes total_tcp_bytes
        close $nf
        close $f0
        close $f1
        close $f2
        close $f3
        puts "Avg UDP throughput = "
        puts [expr $total_udp_bytes/29]
        puts "Avg TCP throughput = "
        puts [expr $total_tcp_bytes/29]
        exec xgraph throughput1.tr throughput2.tr -geometry 800x800 &
        exec xgraph conw1.tr -geometry 800x800 &

        exec nam out.nam &
        exit 0
}

proc record {} {

        # global variable
        global ns f0 f2 sink2 sink1
        global total_udp_bytes total_tcp_bytes
        # time
        set time 0.75
        set now [$ns now]

        # bandwidth
        set bw0 [$sink1 set bytes_]
```

```
        set bw2 [$sink2 set bytes_]
        # Write in the file
        puts $f0 "$now [expr ($bw0/($time*8*1000000))]"
        puts $f2 "$now [expr ($bw2/($time*8*1000000))]"
        set total_udp_bytes [expr ($total_udp_bytes+$bw2)]
        set total_tcp_bytes [expr ($total_tcp_bytes+$bw0)]
        #puts "Avg UDP throughput = "
        #puts $total_udp_bytes
        #puts "Avg TCP throughput = "
        #puts $total_tcp_bytes
        $sink1 set bytes_ 0
        $sink2 set bytes_ 0

        # Recall
        $ns at [expr $now + $time] "record"
}

proc congest {} {

        global f1 f3 tcp0 udp0 ns

        # time
        set time .1
        set now [$ns now]

        set cwnd1 [$tcp0 set cwnd_]
        #set cwnd2 [$udp0 set cwnd_]

        puts $f1 "$now [expr $cwnd1]"
        #puts $f3 "$now [expr $cwnd2]"

        $ns at [expr $now + $time] "congest"
}

# TCP Agent
set tcp0 [new Agent/TCP/Reno]
set udp0 [new Agent/UDP]
# Agent with node
$ns attach-agent $n0 $tcp0
$ns attach-agent $n3 $udp0


# Application
```

```
set ftp1 [new Application/FTP]
set cbr1 [new Application/Traffic/CBR]
$cbr1 set packetSize_ 800
$cbr1 set interval_ 0.005


# Agent with application
$ftp1 attach-agent $tcp0
$tcp0 set fid_ 1
#$ftp1 set maxpkts_ 100
$cbr1 attach-agent $udp0
$udp0 set fid_ 2

# Sink Agent
set sink1 [new Agent/TCPSink]
set sink2 [new Agent/LossMonitor]

# Agent with node
$ns attach-agent $n2 $sink2
$ns attach-agent $n2 $sink1

#Sink with sender
$ns connect $tcp0 $sink1
$ns connect $udp0 $sink2

$ns at 0.0 "record"
$ns at 0.0 "congest"

$ns at 1.0 "$ftp1 start"
$ns at 1.0 "$cbr1 start"
$ns at 30.0 "$ftp1 stop"
$ns at 30.0 "$cbr1 stop"
$ns at 30.0 "finish"

#set total_udp_bytes [expr $total_udp_bytes/29]
#set total_tcp_bytes [expr $total_tcp_bytes/29]


$ns run
```

---------- Lab 8 ----------

exercise1.tcl

```
#Create a simulator object
set ns [new Simulator]
$ns color 1 Red
$ns color 2 Blue
$ns color 3 Green

#Open the output files
#cwnd of tcp0
set f0 [open out0.tr w]
 #cwnd of tcp1
set f1 [open out1.tr w]
 #cwnd of tcp2
set f5 [open out5.tr w]
 #throughput of tcp0
set f2 [open out2.tr w]
 #throughput of tcp1
set f3 [open out3.tr w]
 #throughput of tcp1
set f6 [open out6.tr w]
#queue size of qmon
set f4 [open out4.tr w]
set nf [open out.nam w]
$ns namtrace-all $nf
set tf [open outall.tr w]
$ns trace-all $tf

#Create 4 nodes
set n0 [$ns node]
set n1 [$ns node]
set n2 [$ns node]
set n3 [$ns node]

#Connect the nodes
$ns duplex-link $n0 $n1 3Mb 10ms DropTail

$ns duplex-link $n1 $n2 1Mb 10ms RED

set qmon [$ns monitor-queue $n1 $n2 [open qtrace.tr w] 0.03]
[$ns link $n1 $n2] queue-sample-timeout

$ns duplex-link-op $n1 $n2 queuePos 1
```

```
Queue/RED set thresh_ 5
Queue/RED set maxthresh_ 20
#$ns duplex-link $n1 $n2 1Mb 10ms DropTail
$ns queue-limit $n1 $n2 30
#set qmon [$ns monitor-queue $n1 $n2 [open qtrace.tr w] 0.03]
#[$ns link $n1 $n2] queue-sample-timeout

$ns duplex-link $n3 $n1 3Mb 10ms DropTail
#Define a 'finish' procedure
proc finish {} {
        global f0 f1 f2 f3 f4 f5 f6 nf tf
        #Close the trace and output files
    close $nf
        close $tf
        close $f0
        close $f1
        close $f2
        close $f3
        close $f4
        close $f5
        close $f6
        #Call xgraph to display the results
        exec xgraph out0.tr out1.tr out5.tr -geometry 800x400 &
        exec xgraph out2.tr out3.tr out6.tr -geometry 800x400 &
        exec xgraph out4.tr -geometry 800x400 &
        #Execute nam on the trace file
    exec nam out.nam &
    exit 0
}

#Define a procedure which periodically records the bandwidth received by the
#three traffic sinks sink0/1/2 and writes it to the three files f0/1/2.
proc record {} {
    global tcp0 tcp1 tcp2 sink2 sink0 sink1 f0 f1 f2 f3 f4 f5 f6 qmon
        #Get an instance of the simulator
        set ns [Simulator instance]
        #Set the time after which the procedure should be called again
    set time 0.4
        #How many bytes have been received by the traffic sinks?
    set bw0 [$sink0 set bytes_]
        set cwnd0 [$tcp0 set cwnd_]
        set bw1 [$sink1 set bytes_]
        set cwnd1 [$tcp1 set cwnd_]
```

```
        set bw2 [$sink2 set bytes_]
        set cwnd2 [$tcp2 set cwnd_]
        set pkts [$qmon set pkts_]
    #Get the current time
    set now [$ns now]
        #Calculate the bandwidth (in MBit/s) and write it to the files
    puts $f2 "$now [expr $bw0/$time*8/1000000]"
        puts $f0 "$now [expr $cwnd0]"
        puts $f3 "$now [expr $bw1/$time*8/1000000]"
        puts $f1 "$now [expr $cwnd1]"
        puts $f4 "$now [expr $pkts]"
        puts $f5 "$now [expr $cwnd2]"
        puts $f6 "$now [expr $bw2/$time*8/1000000]"
        #Reset the bytes_ values on the traffic sinks
    $sink0 set bytes_ 0
        $sink1 set bytes_ 0
        $sink2 set bytes_ 0
        #Re-schedule the procedure
    $ns at [expr $now+$time] "record"
}

#Create a traffic sinks and attach them to the node n2
set sink0 [new Agent/TCPSink]
$ns attach-agent $n2 $sink0
set sink1 [new Agent/TCPSink]
$ns attach-agent $n2 $sink1
set sink2 [new Agent/TCPSink]
$ns attach-agent $n2 $sink2

set tcp0 [new Agent/TCP]
$ns attach-agent $n0 $tcp0

set tcp2 [new Agent/TCP]
$ns attach-agent $n0 $tcp2

$tcp0 set fid_ 1
$tcp2 set fid_ 3

set traffic0 [new Application/FTP]
$traffic0 attach-agent $tcp0

set traffic2 [new Application/FTP]
$traffic2 attach-agent $tcp2
```

```
#Connect the source and the sink
$ns connect $tcp0 $sink0

$ns connect $tcp2 $sink2

set tcp1 [new Agent/TCP]
$ns attach-agent $n3 $tcp1

$tcp1 set fid_ 2

set traffic1 [new Application/FTP]
$traffic1 attach-agent $tcp1

#Connect the source and the sink
$ns connect $tcp1 $sink1

#Start logging the received bandwidth
$ns at 0.0 "record"
#Start the traffic sources
$ns at 1.0 "$traffic0 start"
$ns at 1.0 "$traffic1 start"
$ns at 1.0 "$traffic2 start"
#Stop the traffic sources
$ns at 60.0 "$traffic0 stop"
$ns at 60.0 "$traffic1 stop"
$ns at 60.0 "$traffic2 stop"
$ns at 65.0 "finish"
#Run the simulation
$ns run
```

exercise2.tcl

```
#Create a simulator object
set ns [new Simulator]
$ns color 1 Red
$ns color 2 Blue
$ns color 3 Green

#Open the output files
#cwnd of tcp0
#set f0 [open out0.tr w]
 #cwnd of tcp1
```

```
#set f1 [open out1.tr w]
 #cwnd of tcp2
#set f5 [open out5.tr w]
 #throughput of tcp0
#set f2 [open out2.tr w]
 #throughput of tcp1
#set f3 [open out3.tr w]
 #throughput of tcp1
#set f6 [open out6.tr w]
#queue size of qmon
#set f4 [open out4.tr w]
set nf [open out.nam w]
$ns namtrace-all $nf
set tf [open outall.tr w]
$ns trace-all $tf

#Create 7 nodes
for {set i 0} {$i < 7} {incr i} {
        set n($i) [$ns node]
}

#Create links between nodes
for {set i 0} {$i < 7} {incr i} {
        $ns duplex-link $n($i) $n([expr ($i+1)%7]) 1Mb 10ms DropTail
}

#Define a 'finish' procedure
proc finish {} {
        global f0 f1 f2 f3 f4 f5 f6 nf tf
        #Close the trace and output files
      close $nf
        close $tf
#       close $f0
#       close $f1
#       close $f2
#       close $f3
#       close $f4
#       close $f5
#       close $f6
        #Call xgraph to display the results
#       exec xgraph out0.tr out1.tr out5.tr -geometry 800x400 &
#       exec xgraph out2.tr out3.tr out6.tr -geometry 800x400 &
#       exec xgraph out4.tr -geometry 800x400 &
```

```
        #Execute nam on the trace file
    exec nam out.nam &
    exit 0
}

#Define a procedure which periodically records the bandwidth received by the
#three traffic sinks sink0/1/2 and writes it to the three files f0/1/2.
proc record {} {
    global tcp0 tcp1 tcp2 sink0 sink1 f0 f1 f2 f3 f4 f5 f6 qmon
        #Get an instance of the simulator
        set ns [Simulator instance]
        #Set the time after which the procedure should be called again
    set time 0.4
        #How many bytes have been received by the traffic sinks?
    #set bw0 [$sink0 set bytes_]
        #set cwnd0 [$tcp0 set cwnd_]
        #set bw1 [$sink1 set bytes_]
        #set cwnd1 [$tcp1 set cwnd_]
        # set bw2 [$sink2 set bytes_]
        #set cwnd2 [$tcp2 set cwnd_]
        #set pkts [$qmon set pkts_]
    #Get the current time
    set now [$ns now]
        #Calculate the bandwidth (in MBit/s) and write it to the files
    #puts $f2 "$now [expr $bw0/$time*8/1000000]"
        #puts $f0 "$now [expr $cwnd0]"
        #puts $f3 "$now [expr $bw1/$time*8/1000000]"
        #puts $f1 "$now [expr $cwnd1]"
        #puts $f4 "$now [expr $pkts]"
        #puts $f5 "$now [expr $cwnd2]"
        #puts $f6 "$now [expr $bw2/$time*8/1000000]"
        #Reset the bytes_ values on the traffic sinks
    $sink0 set bytes_ 0
        $sink1 set bytes_ 0
        #$sink2 set bytes_ 0
        #Re-schedule the procedure
    $ns at [expr $now+$time] "record"
}

#####################  UDP  ################

#Create a traffic sinks and attach them to the node n3
#set sink3 [new Agent/LossMonitor]
```

```
#$ns attach-agent $n(3) $sink3

#set udp0 [new Agent/UDP]
#$ns attach-agent $n(0) $udp0

#$udp0 set fid_ 1

#set traffic0 [new Application/Traffic/CBR]
#$traffic0 set packetSize_ 500
#$traffic0 set interval_ 0.005
#$traffic0 attach-agent $udp0

#Connect the source and the sink
#$ns connect $udp0 $sink3

#####################  TCP  ########################

set sink0 [new Agent/LossMonitor]
set sink1 [new Agent/LossMonitor]

set sink3 [new Agent/TCPSink]
$ns attach-agent $n(3) $sink3

set tcp0 [new Agent/TCP]
$ns attach-agent $n(0) $tcp0

$tcp0 set fid_ 1

set traffic0 [new Application/FTP]
$traffic0 attach-agent $tcp0

#Connect the source and the sink
$ns connect $tcp0 $sink3


#Start logging the received bandwidth
$ns at 0.0 "record"

$ns rtmodel-at 1.5 down $n(1) $n(2) # Nothing from TCP if down at 1 as no syn ack
$ns rtmodel-at 2.0 up $n(1) $n(2)

#Start the traffic sources
$ns at 1.0 "$traffic0 start"
```

```
#Stop the traffic sources
$ns at 5.0 "$traffic0 stop"
$ns at 5.0 "finish"
#Run the simulation
$ns run
```

exercise3.tcl

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

$ns color 1 Red
#$ns color 2 Blue
#$ns color 3 Green

#Open the output files

#cwnd of tcp0
#set f0 [open out0.tr w]
#cwnd of tcp1
#set f1 [open out1.tr w]
#cwnd of tcp2
#set f5 [open out5.tr w]
#throughput of tcp0
#set f2 [open out2.tr w]
#throughput of tcp1
#set f3 [open out3.tr w]
#throughput of tcp1
#set f6 [open out6.tr w]
#queue size of qmon
#set f4 [open out4.tr w]

set nf [open out.nam w]
$ns namtrace-all $nf
set tf [open outall.tr w]
$ns trace-all $tf

#Create 7 nodes
for {set i 0} {$i < 7} {incr i} {
        set n($i) [$ns node]
}

#Create links between nodes
```

```
for {set i 0} {$i < 7} {incr i} {
        $ns duplex-link $n($i) $n([expr ($i+1)%7]) 1Mb 10ms DropTail
}

#Define a 'finish' procedure
proc finish {} {
        global f0 f1 f2 f3 f4 f5 f6 nf tf
        #Close the trace and output files
      close $nf
        close $tf
#       close $f0
#       close $f1
#       close $f2
#       close $f3
#       close $f4
#       close $f5
#       close $f6
        #Call xgraph to display the results
#       exec xgraph out0.tr out1.tr out5.tr -geometry 800x400 &
#       exec xgraph out2.tr out3.tr out6.tr -geometry 800x400 &
#       exec xgraph out4.tr -geometry 800x400 &
        #Execute nam on the trace file
      exec nam out.nam &
      exit 0
}

#Define a procedure which periodically records the bandwidth received by the
#three traffic sinks sink0/1/2 and writes it to the three files f0/1/2.
proc record {} {
      global tcp0 tcp1 tcp2 sink2 sink0 sink1 f0 f1 f2 f3 f4 f5 f6 qmon
        #Get an instance of the simulator
        set ns [Simulator instance]
        #Set the time after which the procedure should be called again
      set time 0.4
        #How many bytes have been received by the traffic sinks?
      set bw0 [$sink0 set bytes_]
        set cwnd0 [$tcp0 set cwnd_]
        set bw1 [$sink1 set bytes_]
        set cwnd1 [$tcp1 set cwnd_]
        set bw2 [$sink2 set bytes_]
        set cwnd2 [$tcp2 set cwnd_]
        set pkts [$qmon set pkts_]
      #Get the current time
```

```
        set now [$ns now]
           #Calculate the bandwidth (in MBit/s) and write it to the files
        puts $f2 "$now [expr $bw0/$time*8/1000000]"
           puts $f0 "$now [expr $cwnd0]"
           puts $f3 "$now [expr $bw1/$time*8/1000000]"
           puts $f1 "$now [expr $cwnd1]"
           puts $f4 "$now [expr $pkts]"
           puts $f5 "$now [expr $cwnd2]"
           puts $f6 "$now [expr $bw2/$time*8/1000000]"
           #Reset the bytes_ values on the traffic sinks
        $sink0 set bytes_ 0
           $sink1 set bytes_ 0
           $sink2 set bytes_ 0
           #Re-schedule the procedure
        $ns at [expr $now+$time] "record"
}

proc rtdump {} {
           global ns
           set now [$ns now]
           puts "Routing table at time $now"
           # Use any one of the following
           #$ns dump-routelogic-nh
           #Table in terms of next hops
           #$ns dump-routelogic-distance
           #Table in terms of distance
}

$ns rtproto DV

Agent/rtProto/DV set advertInterval 1

#####################  UDP  ################

#Create a traffic sinks and attach them to the node n2
#set sink3 [new Agent/LossMonitor]
#$ns attach-agent $n(3) $sink3

#set udp0 [new Agent/UDP]
#$ns attach-agent $n(0) $udp0

#$udp0 set fid_ 1
```

```
#set traffic0 [new Application/Traffic/CBR]
#$traffic0 set packetSize_ 500
#$traffic0 set interval_ 0.02
#$traffic0 attach-agent $udp0

#Connect the source and the sink
#$ns connect $udp0 $sink3

#####################  TCP  ########################

set sink3 [new Agent/TCPSink]
$ns attach-agent $n(3) $sink3

set tcp0 [new Agent/TCP]
$ns attach-agent $n(0) $tcp0

$tcp0 set fid_ 1

set traffic0 [new Application/FTP]
$traffic0 attach-agent $tcp0

#Connect the source and the sink
$ns connect $tcp0 $sink3

$ns cost $n(1) $n(2) 1

#Start logging the received bandwidth
#$ns at 0.0 "record"

$ns rtmodel-at 1.5 down $n(1) $n(2)
$ns rtmodel-at 2.0 up $n(1) $n(2)

#Start the traffic sources
$ns at 1.0 "$traffic0 start"
#$ns at 1.25 "rtdump"
#$ns at 1.75 "rtdump"
#$ns at 3 "rtdump"
#Stop the traffic sources
$ns at 5.0 "$traffic0 stop"
$ns at 5.0 "finish"
#Run the simulation
$ns run
```