Department of Computer Science

Software Tools Lab :

Exercise 1: Routing

Write a Tcl script that forms a network consisting of 10 nodes, numbered from 1 to 10 , forming a ring topology. The links have a 1024Kbps bandwidth with 10ms delay. Set the routing protocol to AODV/DSR/TORA. Send TCP packets from node 1 to node 5 with the rate of 50 packets/sec (using default packet size). Start   
transmission at 0.02 Simulation Time. Bring down the link between node 2 and node 3 at 0.4 Simulation Time. Finish the transmission at 1.000.

Then run nam to view the results. Find the Network Layer Throughput from Trace File.

Exercise 2 : Routing

Create a network which looks like the following:

                        4 ---------5  
                        |   
                        |   
          0-------------1----------2   
                        |   
                        |   
                        3-------------6

All links are 1 Mbps, 5 ms delay.

Choose DSDV Protocol in the Network Layer and IEEE 802.11 in MAC Layer. Attach UDP packet at node 1 , 4 and 3.

Exercise 3: Analyze Bandwidth Share of TCP and UDP Traffic   
  It is equivalently important to understand the ns-2 output trace format and have some knowledge on how to extract relevant information.  This exercise is dedicated   
to familiarize you with ns-2's output format, in particular by trace-all, and techniques to do trace post-processing.

Topology: Use the [simple.tcl](https://www2.ee.washington.edu/research/nsl/class/566/2001spr/ns/simple.tcl) example discussed in class, but first swap the source/destination of udp1 i.e. make its source be n1 and destination n3.

As a result of running ns on the modified simple.tcl, we obtain an output file "simple.tr". It contains information of all packets enqueued (+), dequeued (-), received (r) and dropped (d).

#simple.tcl

#create scheduler

set ns [new Simulator]

#choose colors

$ns color 0 blue

$ns color 1 red

$ns color 2 green

#turn tracing on

set f [open simple.tr w]

$ns trace-all $f

set nf [open simple.nam w]

$ns namtrace-all $nf

#create topology

set n0 [$ns node]

set n1 [$ns node]

set n2 [$ns node]

set n3 [$ns node]

$ns duplex-link $n0 $n2 5Mb 2ms DropTail

$ns duplex-link $n1 $n2 5Mb 2ms DropTail

$ns duplex-link $n2 $n3 1.5Mb 10ms DropTail

#limit queue size (otherwise unlimited)

#$ns queue-limit $n2 $n3 5

#adjust nam orientation

$ns duplex-link-op $n0 $n2 orient right-up

$ns duplex-link-op $n1 $n2 orient right-down

$ns duplex-link-op $n2 $n3 orient right

#turn nam visualization for n2-n3 queue

$ns duplex-link-op $n2 $n3 queuePos 0.5

#create udp agenet

set udp0 [new Agent/UDP]

#attach agent to node

$ns attach-agent $n0 $udp0

#select packet color for nam

$udp0 set class\_ 0

#create a cbr application

set cbr0 [new Application/Traffic/CBR]

#attach application to agent

$cbr0 attach-agent $udp0

#create another udp/cbr agent/application

set udp1 [new Agent/UDP]

$ns attach-agent $n3 $udp1

$udp1 set class\_ 1

set cbr1 [new Application/Traffic/CBR]

$cbr1 attach-agent $udp1

#create and attach null agents

set null0 [new Agent/Null]

$ns attach-agent $n3 $null0

set null1 [new Agent/Null]

$ns attach-agent $n1 $null1

$ns connect $udp0 $null0

$ns connect $udp1 $null1

#schedule the start of cbr traffic

$ns at 1.0 "$cbr0 start"

$ns at 1.1 "$cbr1 start"

#create and attach a TCP agent source/sink

set tcp [new Agent/TCP]

$tcp set class\_ 2

set sink [new Agent/TCPSink]

$ns attach-agent $n0 $tcp

$ns attach-agent $n3 $sink

$ns connect $tcp $sink

#attach an ftp application to the TCP agent

set ftp [new Application/FTP]

$ftp attach-agent $tcp

#schedule start of ftp application

$ns at 1.2 "$ftp start"

#schedule end of ftp application

$ns at 5.0 "$ftp stop"

#$ns at 5.0 "$ns detach-agent $n0 $tcp ; $ns detach-agent $n3 $sink"

#print the (default) cbr values

puts [$cbr0 set packetSize\_]

puts [$cbr0 set interval\_]

$ns at 6.0 "finish"

proc finish {} {

global ns f nf

$ns flush-trace

close $f

close $nf

#puts "running nam..."

#exec nam simple.nam &

exit 0

}

$ns run