**1ST LAB PROGRAM**

**TCL FILE**

set ns [new Simulator]  
set tf [open [lab1.tr](http://lab1.tr/) w]  
$ns trace-all $tf  
set nf [open lab1.nam w]  
$ns namtrace-all $nf  
proc finish { } {  
global ns tf nf  
$ns flush-trace  
exec nam lab1.nam &  
exec awk -f lab1.awk [lab1.tr](http://lab1.tr/) &  
close $tf  
close $nf  
    exit  
}  
set n0 [$ns node]  
set n1 [$ns node]  
set n2 [$ns node]  
$n0 label "TCP source"  
$n2 label "tcp Sink"  
$ns duplex-link $n0 $n2 3Mb 1ns DropTail  
$ns duplex-link $n2 $n1 2Mb 1ns DropTail  
$ns queue-limit $n0 $n2 10  
$ns queue-limit $n2 $n1 10  
set tcp [new Agent/TCP]  
$ns attach-agent $n0 $tcp  
set ftp [new Application/FTP]  
$ftp attach-agent $tcp  
set tcpsink [new Agent/TCPSink]  
$ns attach-agent $n2 $tcpsink  
$ns connect $tcp $tcpsink  
$ns at 0.2 "$ftp start"  
$ns at 2.5 "$ftp stop"  
$ns at 3.0 "finish"  
$ns run

**AWK LAB 1**

BEGIN{  
count = 0  
}  
{  
if ($1 == "d")  
    count ++;  
}  
END{  
printf("Number of packets dropped are: %d/n", count)  
}

**LAB 2   
 TCL**

set val(stop) 10.0 ;  
  
  
set ns [new Simulator]  
set tracefile [open [topo.tr](http://topo.tr/) w]  
$ns trace-all $tracefile  
set namfile [open topo.nam w]  
$ns namtrace-all $namfile  
set n0 [$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 $n2 10Mb 10ms DropTail  
$ns queue-limit $n0 $n2 5  
$ns duplex-link $n1 $n2 100Mb 10ms DropTail  
$ns queue-limit $n1 $n2 50  
$ns duplex-link $n2 $n3 100Mb 10ms DropTail  
$ns queue-limit $n2 $n3 50  
$ns duplex-link $n3 $n4 100Mb 10ms DropTail  
$ns queue-limit $n3 $n4 50  
$ns duplex-link $n3 $n5 0.5Mb 10ms DropTail  
$ns queue-limit $n3 $n5 5  
  
  
$ns duplex-link-op $n0 $n2 orient right-down  
$ns duplex-link-op $n1 $n2 orient right-up  
$ns duplex-link-op $n2 $n3 orient right  
$ns duplex-link-op $n3 $n4 orient right-up  
$ns duplex-link-op $n3 $n5 orient right-down  
  
set tcp0 [new Agent/TCP]  
$ns attach-agent $n0 $tcp0  
  
set sink1 [new Agent/TCPSink]  
$ns attach-agent $n4 $sink1  
$ns connect $tcp0 $sink1  
  
$tcp0 set packetSize\_ 1500  
  
set udp2 [new Agent/UDP]  
$ns attach-agent $n1 $udp2  
set null3 [new Agent/Null]  
$ns attach-agent $n5 $null3  
$ns connect $udp2 $null3  
$udp2 set packetSize\_ 1500  
  
set err [new ErrorModel]  
$ns lossmodel $err $n2 $n3  
$err set rate\_ 0.6  
  
  
set ftp0 [new Application/FTP]  
$ftp0 attach-agent $tcp0  
$ns at 1.0 "$ftp0 start"  
$ns at 10.0 "$ftp0 stop"  
  
set cbr1 [new Application/Traffic/CBR]  
$cbr1 attach-agent $udp2  
$cbr1 set packetSize\_ 1000  
$cbr1 set rate\_ 1.0Mb  
$cbr1 set random\_ null  
  
$ns at 1.0 "$cbr1 start"  
$ns at 10.0 "$cbr1 stop"  
  
proc finish {} {  
global ns tracefile namfile  
$ns flush-trace  
close $tracefile  
close $namfile  
  
exec nam topo.nam &  
exit 0  
}  
  
$ns at $val(stop) "$ns nam-end-wireless $val(stop)"  
$ns at $val(stop) "finish"  
$ns at $val(stop) "puts \"done\" :$ns halt"  
  
$ns run

**AWK LAB 2**

BEGIN{  
ftppack = 0;  
cbrpack = 0;  
ftptime = 0;  
cbrtime = 0;  
}  
{  
if($1 == "r" &&  $3 == "3" && $4 == "4" && ($5 == "tcp"))  
{  
ftppack = ftppack + $6;  
ftptime = $2;  
}  
if($1 == "r" && $3 == "1" && $4 == "2")  
{  
cbrpack += $6;  
cbrtime = $2  
}  
}  
END{  
printf("\n Throughput of FTP = %f Mbps \n", (ftppack/ftptime) \* (8/1000000));  
printf("\n Throughput of CBR = %f Mbps \n", (cbrpack/cbrtime) \* (8/1000000));  
}

**LAB -03 CIRCULAR TOPOLOGY**

**TCL**

set val(stop) 5.0;  
  
set ns [new Simulator]  
  
set tf [open [ring.tr](http://ring.tr/) w]  
$ns trace-all $tf  
set nf [open ring.nam w]  
$ns namtrace-all $nf  
  
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]  
  
$n0 color "red"  
$n6 color "blue"  
  
$ns simplex-link $n0 $n1 3Mb 10ms DropTail  
$ns simplex-link $n1 $n2 100Mb 10ms DropTail  
$ns simplex-link $n2 $n3 100Mb 10ms DropTail  
$ns simplex-link $n3 $n4 5Mb 10ms DropTail  
$ns simplex-link $n4 $n5 5Mb 10ms DropTail  
$ns simplex-link $n5 $n6 5Mb 10ms DropTail  
$ns simplex-link $n6 $n0 5Mb 10ms DropTail  
  
$ns queue-limit $n0 $n1 10  
$ns queue-limit $n1 $n2 100  
$ns queue-limit $n2 $n3 10  
$ns queue-limit $n3 $n4 3  
$ns queue-limit $n4 $n5 3  
$ns queue-limit $n5 $n6 3  
$ns queue-limit $n6 $n0 3  
  
$ns simplex-link-op $n0 $n1 orient right  
$ns simplex-link-op $n1 $n2 orient right-down  
$ns simplex-link-op $n2 $n3 orient down  
$ns simplex-link-op $n3 $n4 orient left  
$ns simplex-link-op $n4 $n5 orient left-up  
$ns simplex-link-op $n5 $n6 orient up  
$ns simplex-link-op $n6 $n0 orient right-up  
  
  
#Set TCP Connection  
set tcp0 [new Agent/TCP]  
$ns attach-agent $n0 $tcp0  
set sink1 [new Agent/TCPSink]  
$ns attach-agent $n6 $sink1  
$ns connect $tcp0 $sink1  
  
  
#Set UDP Connection  
set udp2 [new Agent/UDP]  
$ns attach-agent $n1 $udp2  
set null3 [new Agent/Null]  
$ns attach-agent $n4 $null3  
$ns connect $udp2 $null3  
  
  
#Attach FTP Application over TCP  
set ftp0 [new Application/FTP]  
$ftp0 attach-agent $tcp0  
$ns at 0.3 "$ftp0 start"  
$ns at 10.0 "$ftp0 stop"  
  
#Attach CBR Application over UDP  
set cbr1 [new Application/Traffic/CBR]  
$cbr1 attach-agent $udp2  
$cbr1 set packetSize\_ 1500  
$cbr1 set rate\_ 10.0Mb  
$cbr1 set random\_ null  
  
$ns at 0.3 "$cbr1 start"  
$ns at 10.0 "$cbr1 stop"  
  
$ns color $tcp0 blue  
$ns color $udp2 red  
  
  
  
  
#Finish Procedure  
proc finish {} {  
global ns tf nf  
$ns flush-trace  
close $tf  
close $nf  
#exec awk -f 4.awk [out.tr](http://out.tr/)  
#Execute the nam animation file  
exec nam ring.nam &  
exit 0  
}  
  
$ns at $val(stop) "$ns nam-end-wireless $val(stop)"  
$ns at $val(stop) "finish"  
$ns at $val(stop) "puts \"done\" :$ns halt"  
  
$ns run

**AWK -03**

BEGIN{  
tt = 0;  
tc = 0;  
rt = 0;  
rc = 0;  
}  
{  
if($1=="+" && $5=="tcp")  
{  
  tt=tt+1;  
}  
if($1=="r" && $5=="tcp")  
{  
   rt=rt+1;  
}  
  
if($1=="+" && $5=="cbr")  
{  
  tc=tc+1;  
}  
if($1=="r" && $5=="cbr")  
{  
  rc=rc+1;  
}  
}  
END{  
printf("Number of TCP packets transmitted : %d \n",tt);  
printf("Number of packets received: %d \n",rt);  
printf("number of packets lost: %d \n",tt-rt);  
printf("percentage of packet lost:%f\n",(tt-rt)/tt);  
  
printf("Number of udp packets transmitted : %d \n",tc);  
printf("Number of packets received: %d \n",rc);  
printf("number of packets lost: %d \n",tc-rc);  
printf("percentage of packet lost:%f\n",(tc-rc)/tc);  
  
}

**LAB -04**

**4 NETWORK**

**TCL**

set val(stop) 10.0  
set ns [new Simulator]  
set tf [open [lab5.tr](http://lab5.tr/) w]  
$ns trace-all $tf  
set nf [open lab5.nam w]  
$ns namtrace-all $nf  
  
proc finish { } {  
global nf tf ns  
$ns flush-trace  
close $nf  
close $tf  
exec nam lab5.nam &  
exec awk -f lab5ftp.awk [lab5.tr](http://lab5.tr/) > temp1 &  
exec awk -f lab5telnet.awk [lab5.tr](http://lab5.tr/) > temp2 &  
exec awk -f lab5pareto.awk [lab5.tr](http://lab5.tr/) > temp3 &  
exec awk -f lab5expo.awk [lab5.tr](http://lab5.tr/) > temp4 &  
  
exit  
}  
  
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 $n2 10Mb 10ms DropTail  
$ns queue-limit $n0 $n2 5  
$ns duplex-link $n1 $n2 100Mb 10ms DropTail  
$ns queue-limit $n1 $n2 50  
$ns duplex-link $n2 $n3 100Mb 10ms DropTail  
$ns queue-limit $n2 $n3 50  
$ns duplex-link $n3 $n4 100Mb 10ms DropTail  
$ns queue-limit $n3 $n4 50  
$ns duplex-link $n3 $n5 0.5Mb 10ms DropTail  
$ns queue-limit $n3 $n5 5  
  
  
$ns duplex-link-op $n0 $n2 orient right-down  
$ns duplex-link-op $n1 $n2 orient right-up  
$ns duplex-link-op $n2 $n3 orient right  
$ns duplex-link-op $n3 $n4 orient right-up  
$ns duplex-link-op $n3 $n5 orient right-down  
  
set tcp0 [new Agent/TCP]  
$ns attach-agent $n0 $tcp0  
set sink2 [new Agent/TCPSink]  
$ns attach-agent $n4 $sink2  
$ns connect $tcp0 $sink2  
$tcp0 set packetSize\_ 1500  
  
set tcp1 [new Agent/TCP]  
$ns attach-agent $n1 $tcp1  
set sink3 [new Agent/TCPSink]  
$ns attach-agent $n5 $sink3  
$ns connect $tcp1 $sink3  
$tcp1 set packetSize\_ 1500  
  
set ftp0 [new Application/FTP]  
$ftp0 attach-agent $tcp0  
$ns at 0.1 "$ftp0 start"  
$ns at 5.0 "$ftp0 stop"  
  
set telnet [new Application/Telnet]  
$telnet attach-agent $tcp1  
$ns at 0.1 "$telnet start"  
$ns at 5.0 "$telnet stop"  
  
set udp0 [new Agent/UDP]  
$ns attach-agent $n0 $udp0  
set null0 [new Agent/Null]  
$ns attach-agent $n5 $null0  
$ns connect $udp0 $null0  
$udp0 set packetSize\_ 1500  
  
set udp1 [new Agent/UDP]  
$ns attach-agent $n1 $udp1  
set null1 [new Agent/Null]  
$ns attach-agent $n4 $null1  
$ns connect $udp1 $null1  
$udp1 set packetSize\_ 1500  
  
set pareto [new Application/Traffic/Pareto]  
$pareto attach-agent $udp0  
$ns at 0.2 "$pareto start"  
$ns at 5.0 "$pareto stop"  
  
set exponential [new Application/Traffic/Exponential]  
$exponential attach-agent $udp1  
$ns at 0.2 "$exponential start"  
$ns at 5.0 "$exponential stop"  
  
  
$ns at $val(stop) "$ns nam-end-wireless $val(stop)"  
$ns at $val(stop) "finish"  
$ns at $val(stop) "puts \"done\":$ns halt"  
$ns run

AWK -04

**FTP AWK**

BEGIN{  
sum=0;  
ftppack=0;  
ftptime=0;  
}  
{  
if(($1=="r") && ($4=="4")&&($5=="tcp"))  
{  
printf("%f\t%f\n",sum/$2,$2);  
sum=sum+$6;  
}  
if($1=="r" && $3=="0" && $4=="4" && ($5=="tcp"))  
{  
ftppack=ftppack+$6;  
ftptime=$2;  
}  
}  
END{  
#printf("\n throughput of FTP=%f Mbps\n", (ftppack/ftptime)\*(8/1000000));  
}

**TELNET AWK**

BEGIN{  
sum=0;  
telnetpack=0;  
telnettime=0;  
}  
{  
if(($1=="r") && ($3=="1")&&($5=="tcp"))  
{  
printf("%f\t%f\n",sum/$2,$2);  
sum=sum+$6;  
}  
if($1=="r" && $3=="1" && $4=="5" && ($5=="tcp"))  
{  
telnetpack=telnetpack+$6;  
telnettime=$2;  
}  
}  
END{  
#printf("The throughput of telnet application is %.4f Mbps  
\n",(telnetpack/telnettime)\*(8/1000000));  
}

**PARETO AWK**

BEGIN{  
sum=0;  
p\_pack=0;  
p\_time=0;  
}  
{  
if(($1=="r") && ($4=="5")&&($5=="pareto"))  
{  
printf("%f\t%f\n",sum/$2,$2);  
sum=sum+$6;  
}  
  
}  
END{  
#printf("\n throughput of FTP=%f Mbps\n", (ftppack/ftptime)\*(8/1000000));  
}

**EXPONENTIAL AWK**

BEGIN{  
sum=0;  
exp\_pack=0;  
exp\_time=0;  
}  
{  
if(($1=="r") && ($4=="4")&&($5=="exp"))  
{  
printf("%f\t%f\n",sum/$2,$2);  
sum=sum+$6;  
}  
  
}  
END{  
#printf("\n throughput of FTP=%f Mbps\n", (ftppack/ftptime)\*(8/1000000));  
}

**LAB -05**

**CONGESTION**

set ns [new Simulator]

set tf [open lab7.tr w]

$ns trace-all $tf

set nf [open lab7.nam w]

$ns namtrace-all $nf

set n0 [$ns node]

set n1 [$ns node]

set n2 [$ns node]

set n3 [$ns node]

**$ns make-lan "$n0 $n1 $n2 $n3" 10mb 10ms LL Queue/DropTail  Mac/802\_3**

set tcp0 [new Agent/TCP]

$ns attach-agent $n0 $tcp0

set ftp0 [new Application/FTP]

$ftp0 attach-agent $tcp0

set sink3 [new Agent/TCPSink]

$ns attach-agent $n3 $sink3

**$ns connect $tcp0 $sink3**

set tcp2 [new Agent/TCP]

$ns attach-agent $n2 $tcp2

set ftp2 [new Application/FTP]

$ftp2 attach-agent $tcp2

set sink2 [new Agent/TCPSink]

$ns attach-agent $n1 $sink2

**$ns connect $tcp2 $sink2**

######To trace the congestion window##########

**set file1 [open file1.tr w]**

**$tcp0 attach $file1**

**$tcp0 trace cwnd\_**

$tcp0 set maxcwnd\_ 10

**set file2 [open file2.tr w]**

**$tcp2 attach $file2**

**$tcp2 trace cwnd\_**

**proc finish { } {**

global nf tf ns

$ns flush-trace

exec nam lab7.nam &

close $nf

close $tf

exit 0

**}**

**$ns at 0.1 "$ftp0 start"**

**$ns at 1.5 "$ftp0 stop"**

**$ns at 2 "$ftp0 start"**

**$ns at 3 "$ftp0 stop"**

**$ns at 0.2 "$ftp2 start"**

**$ns at 2 "$ftp2 stop"**

**$ns at 2.5 "$ftp2 start"**

**$ns at 4 "$ftp2 stop"**

**$ns at 5.0 "finish"**

**$ns run**

**AWK Script:**

BEGIN{

#include<stdio.h>

}

{

**if($6=="cwnd\_")**

   printf("%f \t %f \n", $1,$7);

}

**END{**

**puts "DONE"**

}

**To run:**

**ns lab7.tcl**

**awk   –f    lab7.awk    file1.tr>tcp1**

**awk   –f    lab7.awk    file2.tr>tcp2**

**xgraph  –x   “time”  –y   “convalue ”   tcp1   tcp2**

program 5

#===================================  
#        Initialization  
#===================================  
#Create a ns simulator  
set ns [new Simulator]  
  
#Open the NS trace file  
set tracefile [open [out.tr](http://out.tr/) w]  
$ns trace-all $tracefile  
  
#Open the NAM trace file  
set namfile [open out.nam w]  
$ns namtrace-all $namfile  
#Open the window file  
set winFile0 [open WinFile0 w]  
  
#===================================  
#        Nodes Definition  
#===================================  
#Create 4 nodes  
set n0 [$ns node]  
set n1 [$ns node]  
set n2 [$ns node]  
set n3 [$ns node]  
  
#===================================  
#        Links Definition  
#===================================  
#Createlinks between nodes  
$ns duplex-link $n0 $n1 5.0Mb 20ms DropTail  
$ns queue-limit $n0 $n1 50  
$ns duplex-link $n1 $n2 0.5Mb 100ms DropTail  
$ns queue-limit $n1 $n2 5  
$ns duplex-link $n2 $n3 5.0Mb 20ms DropTail  
$ns queue-limit $n2 $n3 50  
  
$ns duplex-link-op $n1 $n2 queuePos 0.5  
  
#Give node position (for NAM)  
$ns duplex-link-op $n0 $n1 orient right  
$ns duplex-link-op $n1 $n2 orient right  
$ns duplex-link-op $n2 $n3 orient right  
  
  
# set TCP variables  
Agent/TCP set nam\_tracevar\_ true  
Agent/TCP set window\_ 20  
Agent/TCP set ssthresh\_ 20  
  
  
#===================================  
#        Agents Definition  
#===================================  
#Setup a TCP connection  
set tcp0 [new Agent/TCP]  
$ns attach-agent $n0 $tcp0  
set sink1 [new Agent/TCPSink]  
$ns attach-agent $n3 $sink1  
$ns connect $tcp0 $sink1  
$tcp0 set packetSize\_ 1500  
  
  
#===================================  
#        Applications Definition  
#===================================  
#Setup a FTP Application over TCP connection  
set ftp0 [new Application/FTP]  
$ftp0 attach-agent $tcp0  
$ns at 0.1 "$ftp0 start"  
$ns at 0.1 "PlotWindow $tcp0 $winFile0"  
$ns at 8.0 "$ftp0 stop"  
  
  
$ns add-agent-trace $tcp0 tcp  
$ns monitor-agent-trace $tcp0  
$tcp0 tracevar cwnd\_  
$tcp0 tracevar ssthresh\_  
  
#===================================  
#        Termination  
#===================================  
#Define a 'finish' procedure  
proc finish {} {  
    global ns tracefile namfile  
    $ns flush-trace  
    close $tracefile  
    close $namfile  
    exec nam out.nam &  
    exit 0  
}  
  
#Plot Window Procedure  
proc PlotWindow {tcpSource file} {  
        global ns  
        set time 0.1  
        set now [$ns now]  
        set cwnd [$tcpSource set cwnd\_]  
        puts $file "$now $cwnd"  
        $ns at [expr $now+$time] "PlotWindow $tcpSource $file"  
}  
  
$ns at 10.0 "finish"  
$ns run  
  
  
  
  
  
Commands to run:  
ns filename.tcl  
xgraph -x "time" -y "congestionvalue" WinFile0