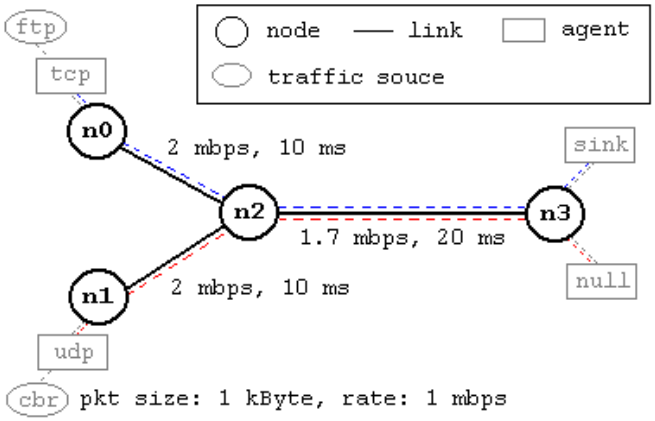
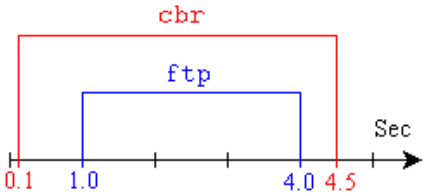
LAB-5, NS2 SIMULATOR

Q1) Find the total number of packets transmitted and received in the network:



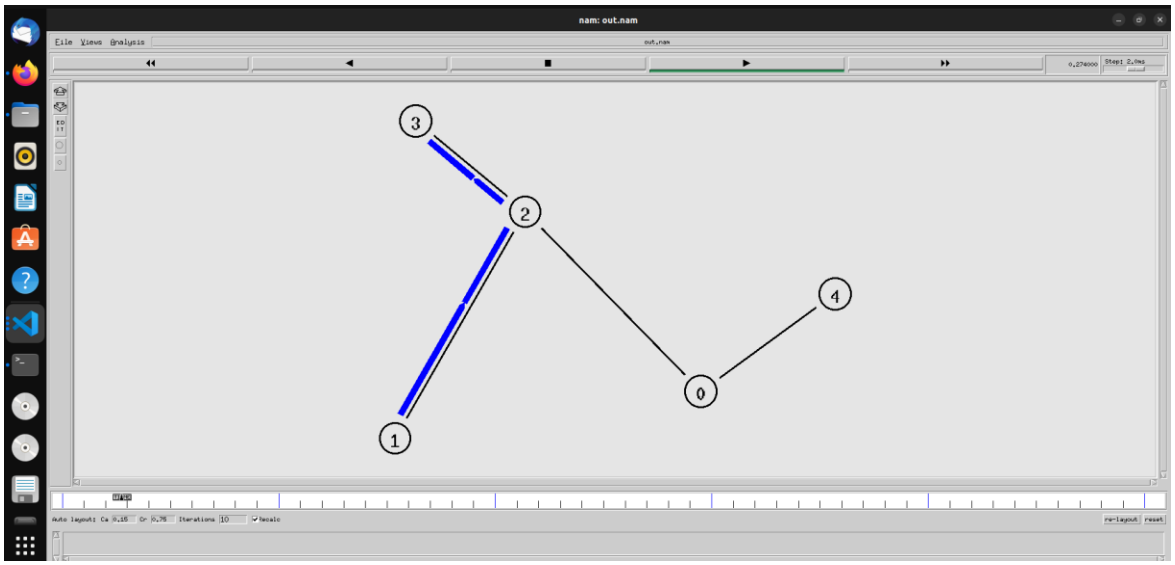


Code:

```
set ns [new Simulator]

$ns color 1 Green
$ns color 2 Blue

set noOfNodes 5

set tf [open trace5.tr w]
$ns trace-all $tf

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

proc finish {} {
    global ns nf tf
    $ns flush-trace
    close $tf
    close $nf
```

```
    exec nam out.nam &
    exit 0
}

for {set i 0} {$i < $noOfNodes} {incr i} {
    set n$i [ $ns node ]
}

$ns duplex-link $n0 $n2 1Mb 10ms DropTail
$ns duplex-link $n1 $n2 1Mb 10ms DropTail
$ns duplex-link $n2 $n3 1Mb 10ms DropTail
$ns duplex-link $n4 $n0 1Mb 10ms DropTail
$ns queue-limit $n2 $n3 1000
set tcp [new Agent/TCP]
$tcp set class_ 2
$ns attach-agent $n0 $tcp
set sink [new Agent/TCPSink]
$ns attach-agent $n3 $sink
$ns connect $tcp $sink
$tcp set fid_ 1

set ftp [new Application/FTP]
$ftp attach-agent $tcp
$ftp set type_ FTP

set udp [new Agent/UDP]
$ns attach-agent $n1 $udp
set null [new Agent/Null]

$ns attach-agent $n3 $null
$ns connect $udp $null
$udp set fid_ 2

set cbr [new Application/Traffic/CBR]
$cbr attach-agent $udp
$cbr set type_ CBR
$cbr set packet_size_ 1000
$cbr set rate_ 1mb
$cbr set random_ false

$ns at 0.1 "$cbr start"
$ns at 1.0 "$ftp start"
$ns at 4.0 "$ftp stop"
$ns at 4.5 "$cbr stop"

$ns at 5.0 "finish"
$ns run
```

Output:



Q2) Simulate the network with varying number of nodes (10,15,20,25,30,40) to compare TCP and UDP protocol in terms of the following performance metrics.

(i) Throughput
(ii) Packet loss
(iii) End-to-end Delay
(iv) Fairness Index

CODE:

```
set ns [new Simulator]

$ns color 1 Green
$ns color 2 Blue

set noOfNodes 5

set tf [open trace5.tr w]
$ns trace-all $tf

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

proc finish {} {
    global ns nf tf
    $ns flush-trace
    close $tf
    close $nf
    exec nam out.nam &
    exit 0
}

for {set i 0} {$i < $noOfNodes} {incr i} {
    set n($i) [ $ns node ]
}


$ns duplex-link $n(0) $n(2) 1Mb 10ms DropTail
$ns duplex-link $n(1) $n(2) 1Mb 10ms DropTail
$ns duplex-link $n(2) $n(3) 1Mb 10ms DropTail

$ns duplex-link $n(4) $n(0) 1Mb 10ms DropTail
#$ns duplex-link $n(5) $n(4) 1Mb 10ms DropTail
#$ns duplex-link $n(6) $n(5) 1Mb 10ms DropTail
#$ns duplex-link $n(10) $n(6) 1Mb 10ms DropTail
for {set i 10} {$i < [expr $noOfNodes - 1]} {incr i} {
    $ns duplex-link $n($i) $n([expr $i + 1]) 1Mb 10ms DropTail
```

```
}


#$ns duplex-link $n(7) $n(1) 1Mb 10ms DropTail
#$ns duplex-link $n(8) $n(7) 1Mb 10ms DropTail
#$ns duplex-link $n(9) $n(8) 1Mb 10ms DropTail


$ns queue-limit $n(2) $n(3) 10


set tcp1 [new Agent/TCP]
$tcp1 set class_ 2
$ns attach-agent $n(0) $tcp1

set tcp2 [new Agent/TCP]
$tcp2 set class_ 2
$ns attach-agent $n(4) $tcp2

# set tcp3 [new Agent/TCP]
# $tcp3 set class_ 2
# $ns attach-agent $n(5) $tcp3

# set tcp4 [new Agent/TCP]
# $tcp4 set class_ 2
# $ns attach-agent $n(6) $tcp4

#set tcp5 [new Agent/TCP]
#$tcp5 set class_ 2
#$ns attach-agent $n(10) $tcp5

for {set i 10} {$i < [expr $noOfNodes - 1]} {incr i} {
    set tcp($i) [new Agent/TCP]
    $tcp($i) set class_ 2
    $ns attach-agent $n($i) $tcp($i)
}

set sink [new Agent/TCPSink]

$ns attach-agent $n(3) $sink
$ns connect $tcp1 $sink
$ns connect $tcp2 $sink
# $ns connect $tcp3 $sink
# $ns connect $tcp4 $sink
#$ns connect $tcp5 $sink

for {set i 10} {$i < [expr $noOfNodes - 1]} {incr i} {
    $ns connect $tcp($i) $sink
}
```

```
$tcp1 set fid_ 1
$tcp2 set fid_ 1

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

set ftp2 [new Application/FTP]
$ftp2 attach-agent $tcp2
$ftp2 set type_ FTP

# set ftp3 [new Application/FTP]
# $ftp3 attach-agent $tcp3
# $ftp3 set type_ FTP

# set ftp4 [new Application/FTP]
# $ftp4 attach-agent $tcp4
# $ftp4 set type_ FTP

#set ftp5 [new Application/FTP]
#$ftp5 attach-agent $tcp5
#$ftp5 set type_ FTP

for {set i 10} {$i < [expr $noOfNodes - 1]} {incr i} {
    set ftp($i) [new Application/FTP]
    $ftp($i) attach-agent $tcp($i)
    $ftp($i) set type_ FTP
    $ns at 1.0 "$ftp($i) start"
    $ns at 4.0 "$ftp($i) stop"
}

set udp [new Agent/UDP]
$ns attach-agent $n(1) $udp

# set udp2 [new Agent/UDP]
# $ns attach-agent $n(7) $udp2

# set udp3 [new Agent/UDP]
# $ns attach-agent $n(8) $udp3

# set udp4 [new Agent/UDP]
# $ns attach-agent $n(9) $udp4

set null [new Agent/Null]

$ns attach-agent $n(3) $null
$ns connect $udp $null
```

```
# $ns connect $udp2 $null
# $ns connect $udp3 $null
# $ns connect $udp4 $null

$udp set fid_ 2

set cbr [new Application/Traffic/CBR]
$cbr attach-agent $udp
$cbr set type_ CBR
$cbr set packet_size_ 1000
$cbr set rate_ 1mb
$cbr set random_ false

# set cbr2 [new Application/Traffic/CBR]
# $cbr2 attach-agent $udp2
# $cbr2 set type_ CBR
# $cbr2 set packet_size_ 1000
# $cbr2 set rate_ 1mb
# $cbr2 set random_ false

# set cbr3 [new Application/Traffic/CBR]
# $cbr3 attach-agent $udp3
# $cbr3 set type_ CBR
# $cbr3 set packet_size_ 1000
# $cbr3 set rate_ 1mb
# $cbr3 set random_ false

# set cbr4 [new Application/Traffic/CBR]
# $cbr4 attach-agent $udp4
# $cbr4 set type_ CBR
# $cbr4 set rate_ 1mb
# $cbr4 set packet_size_ 1000
# $cbr4 set random_ false


$ns at 0.1 "$cbr start"
$ns at 4.5 "$cbr stop"
#$ns at 0.1 "$cbr2 start"
#$ns at 4.5 "$cbr2 stop"
#$ns at 0.1 "$cbr3 start"
#$ns at 4.5 "$cbr3 stop"
#$ns at 0.1 "$cbr4 start"
#$ns at 4.5 "$cbr4 stop"

$ns at 1.0 "$ftp1 start"
$ns at 4.0 "$ftp1 stop"
$ns at 1.0 "$ftp2 start"
$ns at 4.0 "$ftp2 stop"
```

```
# $ns at 1.0 "$ftp3 start"
# $ns at 4.0 "$ftp3 stop"
# $ns at 1.0 "$ftp4 start"
# $ns at 4.0 "$ftp4 stop"
#$ns at 1.0 "$ftp5 start"
#$ns at 4.0 "$ftp5 stop"


$ns at 5.0 "finish"
$ns run
```

PYTHON PLOT FILE:

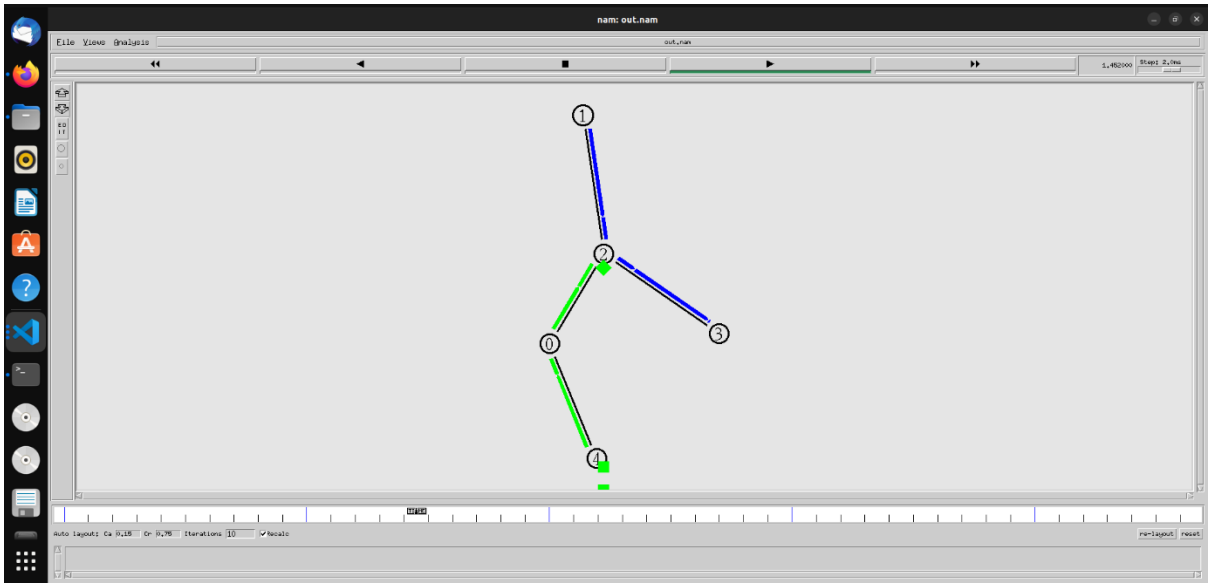
```
from matplotlib import pyplot as plt
def printVal(f,name):
    print(name)
    data = f.read()

    data = data.split("\n")

    startTime = float('inf')
    endTime = float('-inf')
    drop = 0
    dropTCP = 0
    dropUDP = 0
    recv = 0
    recvUDP = 0
    recvTCP = 0
    trans = 0
    transTCP = 0
    transUDP = 0
    delaysTCP = []
    delaysUDP = []
    delay = {}
    delays = []
    recvNode = {}
    recvNodeTCP = {}
    recvNodeUDP = {}
    for dat in data:
        t = dat.split()
        if not t:
            continue

        if t[0] == 'r':
            recvNode[t[3]] = recvNode.get(t[3],0) + 1
            recvNodeTCP[t[3]] = recvNodeTCP.get(t[3],0) + 1
            recvNodeUDP[t[3]] = recvNodeUDP.get(t[3],0) + 1
```

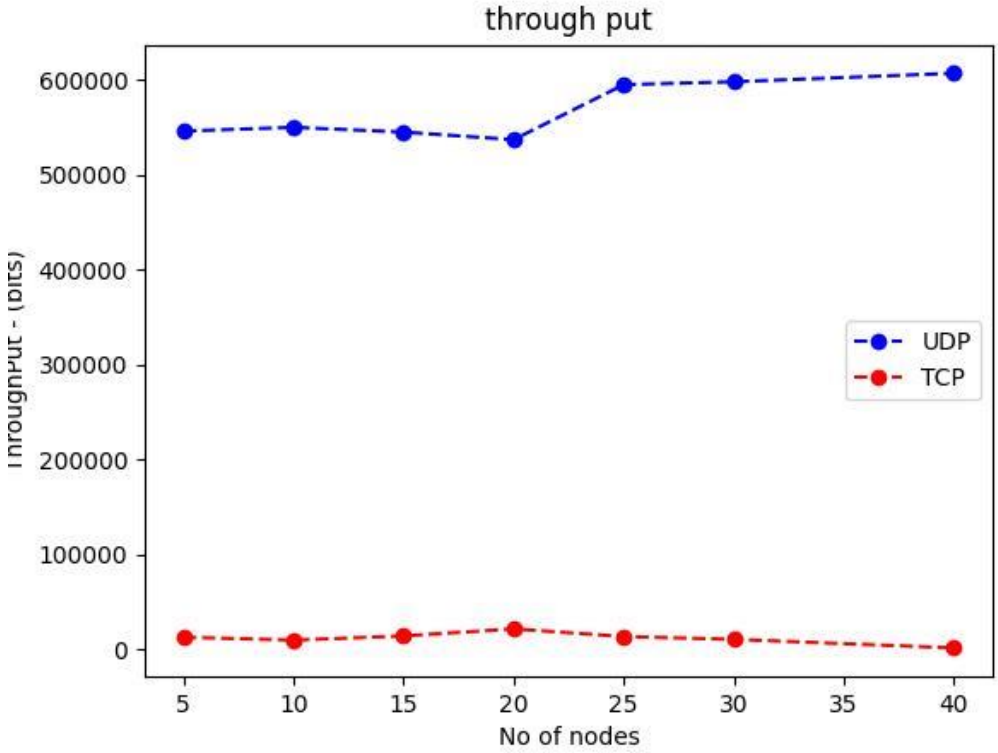
```
            if t[-1] in delay and t[4] == 'cbr':
                delaysUDP.append(float(t[1]) - delay[t[-1]])
            if t[-1] in delay and t[4] == 'tcp':
                delaysTCP.append(float(t[1]) - delay[t[-1]])

        if t[0] == 'r' and t[3] == '3':
            recv += int(t[5])
            if t[4] == "cbr":
                recvUDP += int(t[5])
            if t[4] == "tcp":
                recvTCP += int(t[5])

        if t[0] == '+' and t[2] == "1" and t[4] == "cbr":
            if t[-1] not in delay:
                delay[t[-1]] = float(t[1])
            trans += int(t[5])
            transUDP += int(t[5])

        if t[0] == '+' and t[2] == "0" and t[4] == "tcp":
            if t[-1] not in delay:
                delay[t[-1]] = float(t[1])
            trans += int(t[5])
            transTCP += int(t[5])

        if t[0] == 'd':
            drop += int(t[5])
            if t[4] == 'tcp':
                dropTCP += int(t[5])
            else:
                dropUDP += int(t[5])

        startTime = min(startTime,float(t[1]))
        endTime = max(endTime, float(t[1]))
    if name == '1':
        plt.pie([trans,recv,drop], labels = ["Transmitted", "Received",
"Dropped"])
        print(trans,recv,drop)
        plt.savefig("Q1.png")
    sums = sum(delaysUDP)
    delayUDP = sums/len(delaysUDP)
    delayTCP = sum(delaysTCP)/len(delaysTCP)

    t = []
    for i in recvNodeTCP:
        t.append(recvNodeTCP[i])
    t2 = []
    for i in recvNodeUDP:
        t2.append(recvNodeUDP[i])
```

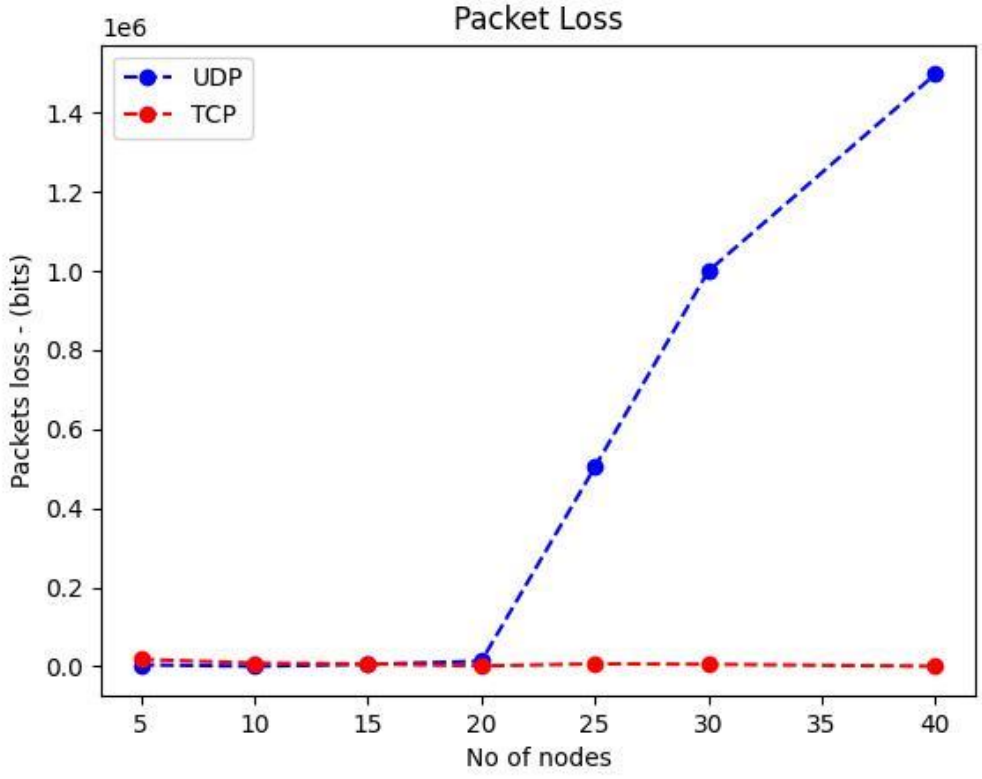
```
    averageT = sum(t)/len(t)
    squaredS = 0
    for i in t:
        squaredS += i**2
    indexTCP = (sum(t)**2)/(squaredS*len(t))

    for i in t2:
        squaredS += i**2
    indexUDP = (sum(t2)**2)/(squaredS*len(t2))

    return dropUDP,dropTCP,recvUDP,recvTCP,delayTCP,delayUDP,indexTCP,indexUDP

X = [5,10,15,20,25,30,40]
Y = []
Y2 =[]
Y3 =[]
Y4 = []
Y5 =[]
Y6 =[]
Y7 =[]
Y8 =[]

f = open("trace5.tr","r")
printVal(f,"1")
for i in X:
    file = (f"trace{i}.tr")
    f = open(file,"r")
    t = printVal(f,str(i))
    Y.append(t[0])
    Y2.append(t[1])
    Y3.append(t[2])
    Y4.append(t[3])
    Y5.append(t[4])
    Y6.append(t[5])
    Y7.append(t[6])
    Y8.append(t[7])

plt.figure()
plt.xlabel("No of nodes")
plt.ylabel("Packets loss - (bits)")
plt.title("Packet Loss")
plt.plot(X,Y,"--bo", label = "UDP")
plt.plot(X,Y2,"--ro", label = "TCP")
plt.legend()
plt.savefig("PacketLossQ2.jpg")

plt.figure()
plt.xlabel("No of nodes")
```

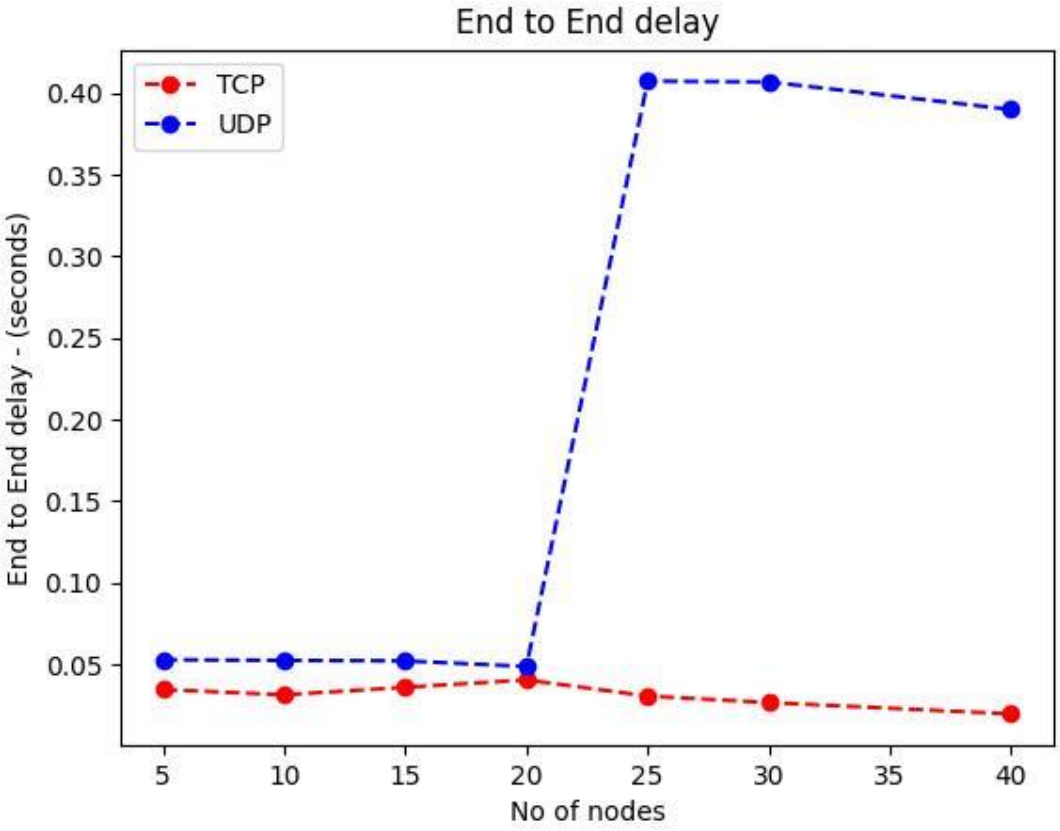
```
plt.ylabel("ThroughPut - (bits)")
plt.title("through put")
plt.plot(X,Y3,"--bo", label = "UDP")
plt.plot(X,Y4,"--ro", label = "TCP")
plt.legend()
plt.savefig("ThroughPutQ2.jpg")

plt.figure()
plt.xlabel("No of nodes")
plt.title("End to End delay")
plt.ylabel("End to End delay - (seconds)")
plt.plot(X,Y5,"--ro", label = "TCP")
plt.plot(X,Y6,"--bo", label = "UDP")
plt.legend()
plt.savefig("EndToEndDelayQ2.jpg")

plt.figure()
plt.xlabel("No of nodes")
plt.title("fairness index")
plt.ylabel("Fairness Index")
plt.plot(X,Y8,"--bo", label = "UDP")
plt.plot(X,Y7,"--ro", label = "TCP")
plt.legend()
plt.savefig("FairnessQ2.jpg")
```

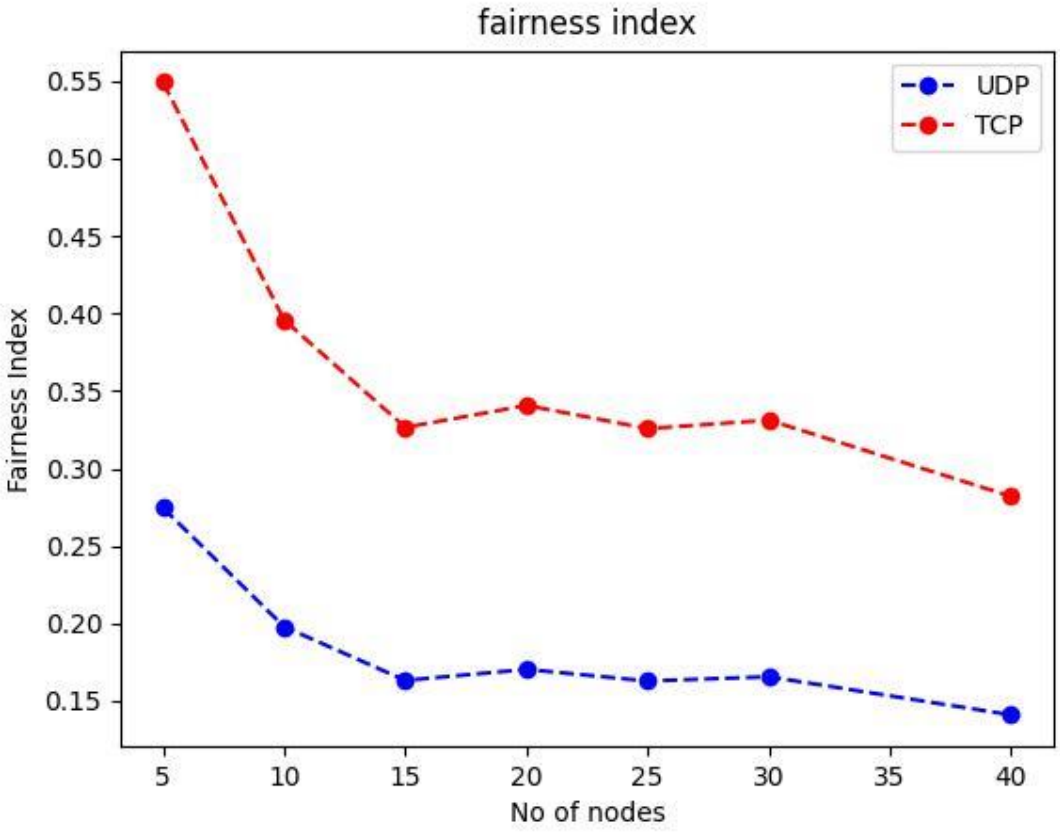
OUTPUT:




through put
600000
500000
400000
300000
200000
100000
0
5
10
15
20
25
30
35
40
No of nodes
UDP
TCP


Packet Loss
1e6
1.4
1.2
1.0
0.8
0.6
0.4
0.2
0.0
Packets loss - (bits)
5
10
15
20
25
30
35
40
No of nodes
UDP
TCP


End to End delay
TCP
UDP
End to End delay - (seconds)
0.40
0.35
0.30
0.25
0.20
0.15
0.10
0.05
5
10
15
20
25
30
35
40
No of nodes


fairness index
UDP
TCP
Fairness Index
0.55
0.50
0.45
0.40
0.35
0.30
0.25
0.20
0.15
5
10
15
20
25
30
35
40
No of nodes