1. *Write Tcl script to create scenario and study the performance of token ring protocols through simulation. Create 6 nodes that forms a network numbered from 1 to 6. Create duplex links between the nodes to form a Ring Topologywith bandwidth of 100 Mbps and delay of 2ms. Setup TCP Connection between node 1 and node 4. Apply FTP Traffic over TCP.Finish the transmission at 100 sec.*

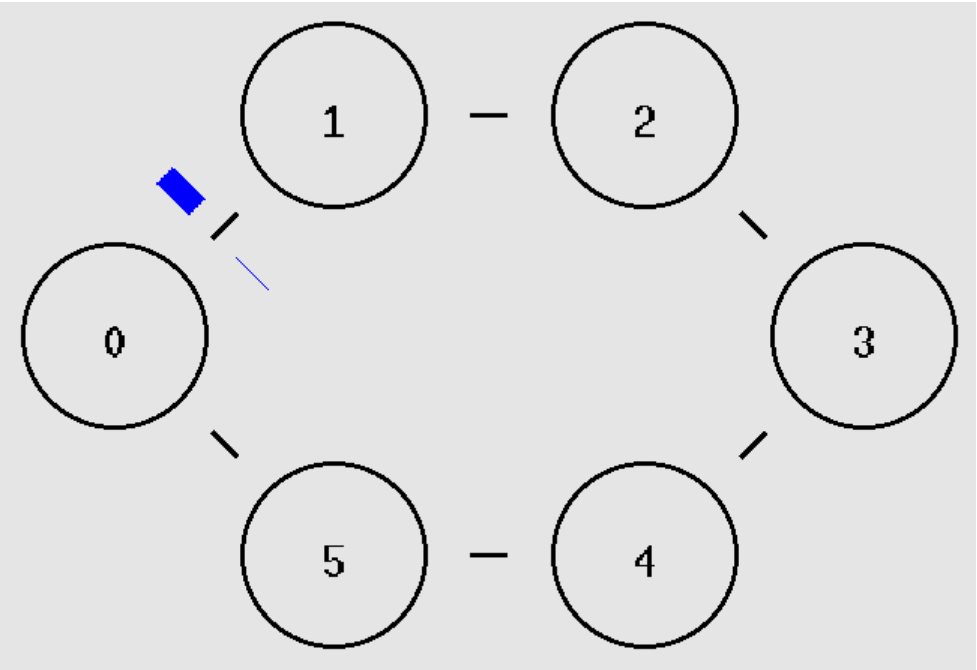
Code:

q1.tcl

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

#Define different colors for data flows
$ns color 1 Blue

#Open the nam trace file
set nf [open q1.nam w]
$ns namtrace-all $nf

#Define a 'finish' procedure
proc finish {} {
    global ns nf
    $ns flush-trace

    #Close the trace file
    close $nf

    #Execute nam on the trace file
    exec nam q1.nam &

    exit 0
}

#Create six nodes
set n1 [$ns node]
set n2 [$ns node]
set n3 [$ns node]
set n4 [$ns node]
set n5 [$ns node]
set n6 [$ns node]

#Create links between the nodes
$ns duplex-link $n1 $n2 100Mb 2ms DropTail
$ns duplex-link $n2 $n3 100Mb 2ms DropTail
$ns duplex-link $n3 $n4 100Mb 2ms DropTail
$ns duplex-link $n4 $n5 100Mb 2ms DropTail
$ns duplex-link $n5 $n6 100Mb 2ms DropTail
$ns duplex-link $n6 $n1 100Mb 2ms DropTail

```

```
41  #Give node position (for NAM)
42  $ns duplex-link-op $n1 $n2 orient right-up
43  $ns duplex-link-op $n2 $n3 orient right
44  $ns duplex-link-op $n3 $n4 orient right-down
45  $ns duplex-link-op $n4 $n5 orient left-down
46  $ns duplex-link-op $n5 $n6 orient left
47  $ns duplex-link-op $n6 $n1 orient left-up
48
49  #Setup a TCP connection
50  set tcp [new Agent/TCP]
51  $tcp set class_ 2
52  $ns attach-agent $n1 $tcp
53
54  set sink [new Agent/TCPSink]
55  $ns attach-agent $n4 $sink
56  $ns connect $tcp $sink
57  $tcp set fid_ 1
58
59  #Setup a FTP over TCP connection
60  set ftp [new Application/FTP]
61  $ftp attach-agent $tcp
62  $ftp set type_ FTP
63
64  #Schedule events for FTP agent
65  $ns at 0.0 "$ftp start"
66  $ns at 95.0 "$ftp stop"
67
68  #Call the finish procedure after 5 seconds of simulation time
69  $ns at 100.0 "finish"
70
71  #Run the simulation
72  $ns run
```

Output:



2) Write a Tcl script that forms a network consisting of 6nodes, numbered from 1 to 6. Each of source and destination has bandwidth of 300 Mbps and delay of 20 ms. Set the bottleneck link bandwidth as 500 sec and delay 10ms. Set the routing protocol to Droptail. Define different colors for different data flows. Send TCP packet from node 1 to node 4and UDP packet from node 5 to 6. Start the TCP data transmission at 1 sec and UDP at 15 sec. Finish the transmission at 100 sec. Then run nam to view the results.

Calculate the following performance metrics using awk script:
a)Throughput
b)Delay
c)Packet loss ratio
d)Jain Fairness index.
e)Plot throughput graph using gnuplot(Tahoe vs Reno)
f)Plot Jain Fairness index graph using gnuplot

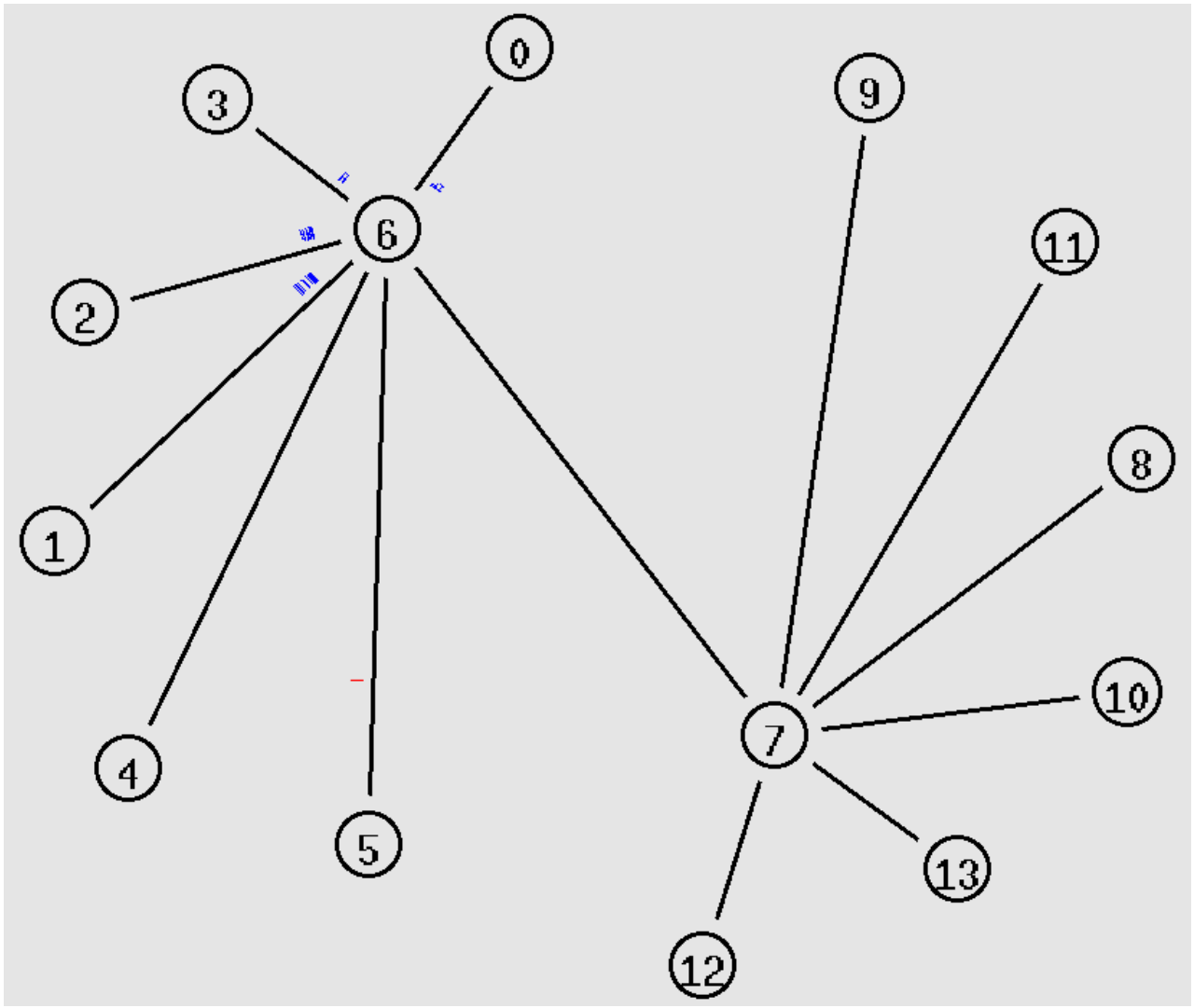
Code:

q2.tcl

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

#Define different colors for data flows
$ns color 1 Blue
$ns color 2 Red

#Open the nam trace file
set nf [open q2.nam w]
$ns namtrace-all $nf

#Define a 'finish' procedure
proc finish {} {
    global ns nf
    $ns flush-trace

    #Close the trace file
    close $nf

    #Execute nam on the trace file
    exec nam q2.nam &

    exit 0
}

#Create fourteen nodes
for {set i 0} {$i < 14} {incr i} {
    set node($i) [$ns node]
}

#Create links between nodes
for {set i 0} {$i < 6} {incr i} {
    $ns duplex-link $node($i) $node(6) 300Mb 20ms DropTail
    $ns duplex-link $node(7) $node([expr $i+8]) 300Mb 20ms DropTail
}
$ns duplex-link $node(6) $node(7) 500Mb 10ms DropTail

```

```
#Setup TCP connections
for {set i 0} {$i < 4} {incr i} {
    set tcp($i) [new Agent/TCP]
    $tcp($i) set class_ 2
    $ns attach-agent $node($i) $tcp($i)

    set sink($i) [new Agent/TCPSink]
    $ns attach-agent $node([expr $i+8]) $sink($i)
    $ns connect $tcp($i) $sink($i)
    $tcp($i) set fid_ 1

    set ftp($i) [new Application/FTP]
    $ftp($i) attach-agent $tcp($i)
    $ftp($i) set type_ FTP
}

for {set i 0} {$i < 2} {incr i} {
    set udp($i) [new Agent/UDP]
    $ns attach-agent $node([expr $i+4]) $udp($i)

    set null($i) [new Agent/Null]
    $ns attach-agent $node([expr $i+12]) $null($i)
    $ns connect $udp($i) $null($i)
    $udp($i) set fid_ 2

    set cbr($i) [new Application/Traffic/CBR]
    $cbr($i) attach-agent $udp($i)
    $cbr($i) set type_ CBR
}

#Schedule events for FTP and CBR agents
for {set i 0} {$i < 4} {incr i} {
    $ns at 1.0 "$ftp($i) start"
}
for {set i 0} {$i < 2} {incr i} {
    $ns at 15.0 "$cbr($i) start"
}

#Call the finish procedure after 5 seconds of simulation time
$ns at 100.0 "finish"

#Run the simulation
$ns run
```

Output:


0
3
6
2
1
4
5
9
11
8
10
7
13
12