PRACTICAL-7(A)

AIM:

Four different departments (N0, N1, N2, and N3) of an Industry are connected in star topology to create a wired network. The link which is used is a duplex link with the queue size 5. The other parameters of link are listed below.

Link	Bandwidth	Delay	Queue Type
no-n2	10Mbps	10ms	DropTail
n1-n2	10Mbps	10ms	DropTail
n2-n3	5Mbps	10ms	DropTail

Design simple tcl script in NS-2 for transferring FTP traffic having following characteristics.

- Packet Size: 1000

- Rate: 1

- Interval: 150

THEORY:

NS2:

- NS2 stands for Network Simulator Version 2.

- It is an open-source event-driven simulator designed specifically for research in computer communication networks.

- It is a discrete event simulator for networking research.

- It provides substantial support to simulate bunch of protocols like TCP, FTP, UDP, https and DSR.

- It simulates wired and wireless network.

- It is primarily Unix based.

- It uses TCL as its scripting language.

- NS2 consists of two key languages:

  - C++

  - Object-oriented Tool Command Language (OTcl)

- While the C++ defines the internal mechanism (i.e., a backend) of the simulation objects, the OTcl sets up simulation by assembling and configuring the objects as well as scheduling discrete events. The C++ and the OTcl are linked together using TclCL

Nodes

- Nodes can be referenced as a virtual representation of any device present in actual topology.

USES:

- NS2 is a serious rendition of Network Simulator(NS).

- It is generally famous for its administrations to gather and picture virtual organizations for recreation.

- It is better than cisco bundle tracer as every gadget in the association goes about as a hub in NS2 while a particular component qualities should be characterized in cisco(i.e if a hub is switch, switch ,and so forth.).

- There are numerous other comparable highlights which make NS2 more mainstream to utilize.

TOPOLOGY:
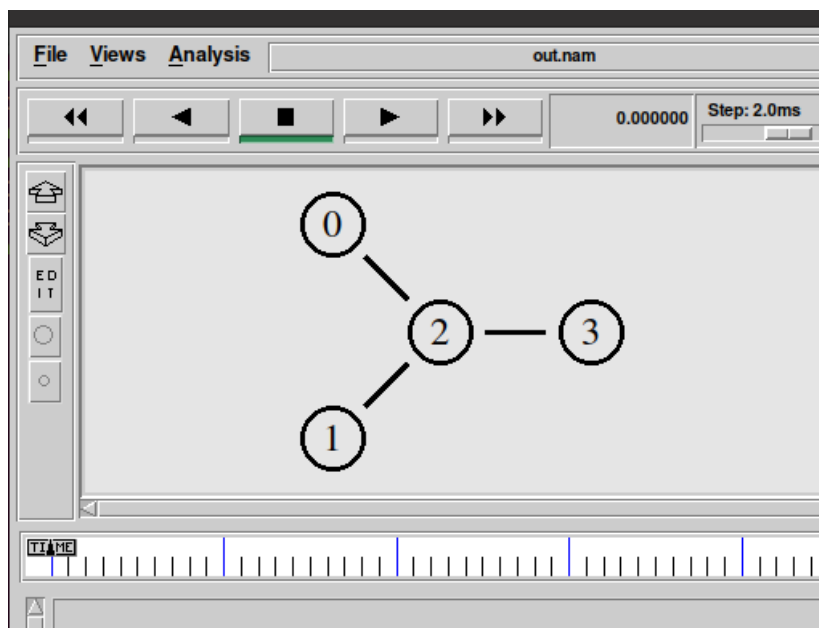
PROGRAM CODE :

```
set ns [new Simulator]

$ns color 1 Blue
$ns color 2 Red

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

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

     exec nam out.nam &
     exit 0
}

set n0 [$ns node]
set n1 [$ns node]
set n2 [$ns node]
set n3 [$ns node]

$ns duplex-link $n0 $n2 10Mb 10ms DropTail
$ns duplex-link $n1 $n2 10Mb 10ms DropTail
$ns duplex-link $n2 $n3 5Mb 10ms DropTail

$ns queue-limit $n2 $n3 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 $n2 $n3 queuePos 0.5
```

```
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
$ftp set packet_size_ 1000
$ftp set rate_ 1mb

set tcp [new Agent/TCP]
$tcp set class_ 1
$ns attach-agent $n1 $tcp
set sink [new Agent/TCPSink]
$ns attach-agent $n3 $sink
$ns connect $tcp $sink
$tcp set fid_ 2



set ftp [new Application/FTP]
$ftp attach-agent $tcp
$ftp set type_ FTP
$ftp set packet_size_ 1000
$ftp set rate_ 1mb

$ns at 0.5 "$ftp start"
$ns at 4.0 "$ftp stop"
$ns at 5.0 "finish"
$ns run
```
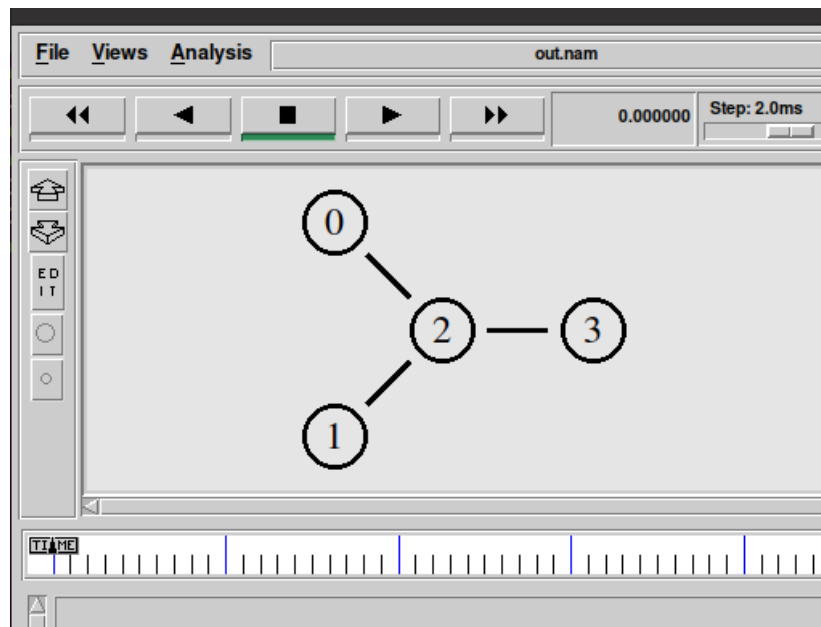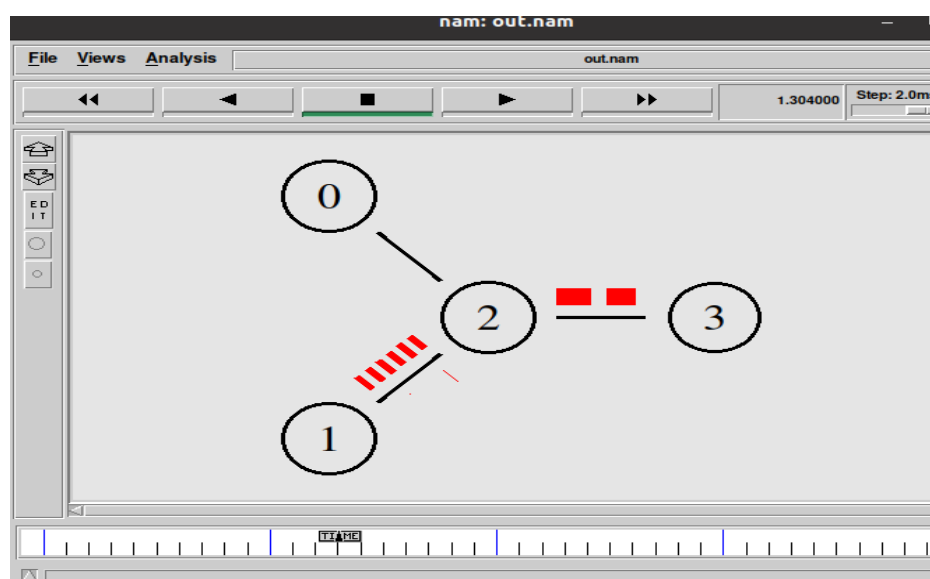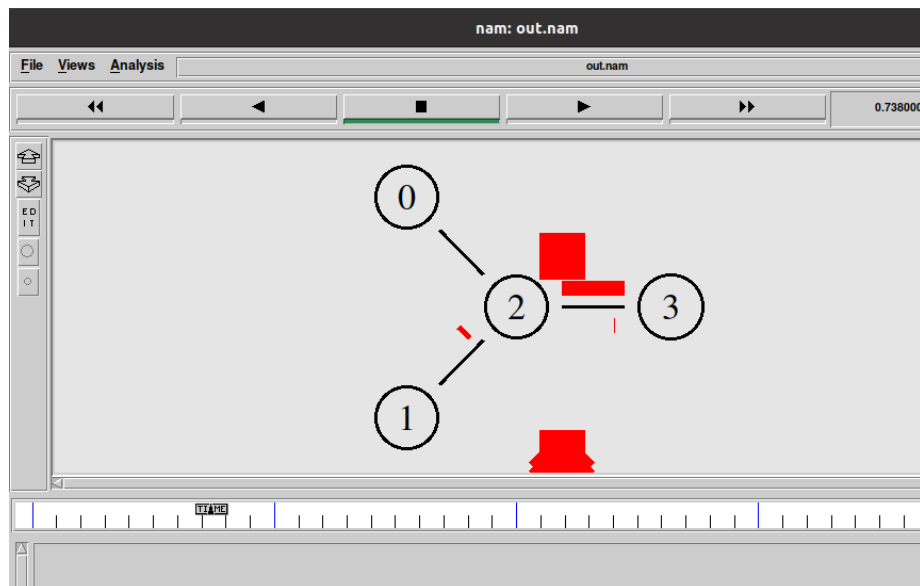
OUTPUT :

- Network Topology

- Data transfer in progress between nodes 1 & 3:

- Packet Drop due to excess packet in queue (i.e more packets than queue capacity) :-

PRACTICAL-7(B)

AIM:

demonstrate various queuing mechanisms and make comparative analysis of various queuing techniques. (using trace file) (DropTail, RED, SFQ and FQ)

THEORY:

SFQ:

- This lining system depends on reasonable lining calculation and proposed by John Nagle in 1987.

- Since it is illogical to have one line for every discussion SFQ utilizes a hashing calculation which isolates the traffic over a set number of lines.

- It isn't so effective than different lines instruments however it additionally requires less computation while being completely reasonable.

- It is classified "Stochastic" because of the explanation that it doesn't really relegate a line for each meeting; it has a calculation which partitions traffic over a confined number of lines utilizing a hashing calculation.

- SFQ allocates an entirely enormous number of FIFO lines**.**

FQ (Fair Queuing):

- It is a lining component that is utilized to permit numerous parcels stream to similarly share the connection limit.

- Switches have different lines for each yield line for each client.

- At the point when a line as accessible as inactive switches filters the lines through cooperative effort and takes first parcel to next line.

- FQ likewise guarantee about the most extreme throughput of the organization.

- For more effectiveness weighted line system is additionally utilized.

RED:

- Random Early Detection (RED) is a congestion avoidance queuing mechanism (as opposed to a congestion administration mechanism) that is potentially useful, particularly in high-speed transit networks.

- Sally Floyd and Van Jacobson projected it in various papers in the early 1990s.

- It is active queue management mechanism.

- It operates on the average queue size and drop packets on the basis of statistics information.

- If the buffer is empty all incoming packets are acknowledged.

- As the queue size increase the probability for discarding a packet also increase.

- When buffer is full probability becomes equal to 1 and all incoming packets are dropped.
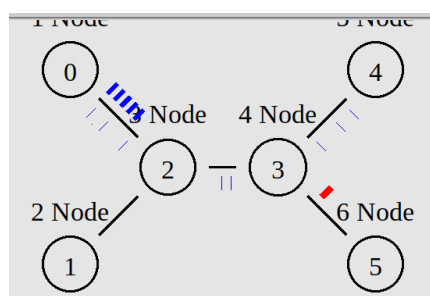
PROGRAM CODE:

SFQ:

```
set ns [new Simulator]
$ns color 1 Blue
$ns color 2 Red
set nf [open prac2.nam w]
$ns namtrace-all $nf
set nr [open prac2.tr w]
$ns trace-all $nr
#Define a 'finish' procedure
proc finish {} {
    global ns nf nr
    $ns flush-trace
    #Close the NAM trace file
    close $nf
    close $nr
    #Execute NAM on the trace file
    exec nam prac2.nam &
    exit 0
}
set n0 [$ns node]
set n1 [$ns node]
set n2 [$ns node]
set n3 [$ns node]
set n4 [$ns node]
set n5 [$ns node]
$n0 shape circle
$n1 shape circle
$n2 shape circle
$n3 shape circle
$n4 shape circle
$n5 shape circle
```

```
$n0 label "1 Node"
$n1 label "2 Node"
$n2 label "3 Node"
$n3 label "4 Node"
$n4 label "5 Node"
$n5 label "6 Node"
$ns duplex-link $n0 $n2 10Mb 10ms SFQ
$ns duplex-link $n1 $n2 10Mb 10ms SFQ
$ns duplex-link $n2 $n3 5Mb 5ms SFQ
$ns duplex-link $n3 $n4 10Mb 10ms SFQ
$ns duplex-link $n3 $n5 10Mb 10ms SFQ
$ns queue-limit $n0 $n2 10
$ns queue-limit $n1 $n2 10
$ns queue-limit $n2 $n3 5
$ns queue-limit $n3 $n4 10
$ns queue-limit $n3 $n5 10
$ns duplex-link-op $n0 $n2 orient down-right
$ns duplex-link-op $n1 $n2 orient up-right
$ns duplex-link-op $n2 $n3 orient right
$ns duplex-link-op $n3 $n4 orient up-right
$ns duplex-link-op $n3 $n5 orient down-right
set tcp [new Agent/TCP]
$tcp set class_ 2
$ns attach-agent $n0 $tcp
set sink [new Agent/TCPSink]
$ns attach-agent $n4 $sink
$ns connect $tcp $sink
$tcp set fid_ 1
set ftp [new Application/FTP]
$ftp attach-agent $tcp
$ftp set type_ FTP
$ftp set packet_size_ 1000
$ftp set rate_ 1mb
$ftp set interval_ 150
```

```
set udp [new Agent/UDP]

$udp set class_ 2

$ns attach-agent $n1 $udp

set sink1 [new Agent/Null]

$ns attach-agent $n5 $sink1

$ns connect $udp $sink1

$udp set fid_ 2

set cbr [new Application/Traffic/CBR]

$cbr attach-agent $udp

$cbr set type_ CBR

$cbr set packet_size_ 1500

$cbr set rate_ 0.05mb

$cbr set interval_ 150

$ns at 1.0 "$ftp start"

$ns at 4.0 "$ftp stop"

$ns at 2.0 "$cbr start"

$ns at 4.5 "$cbr stop"

$nsat5.0"finish"

$ns r
```

OUTPUT:

FQ:

```
set ns [new Simulator]
$ns color 1 Blue
$ns color 2 Red
set nf [open prac3.nam w]
$ns namtrace-all $nf
set nr [open prac3.tr w]
$ns trace-all $nr
#Define a 'finish' procedure
proc finish {} {
    global ns nf nr
    $ns flush-trace
    #Close the NAM trace file
    close $nf
    close $nr
    #Execute NAM on the trace file
    exec nam prac3.nam &
    exec awk -f through.awk prac3.tr &
    exit 0   }
set n0 [$ns node]
set n1 [$ns node]
set n2 [$ns node]
set n3 [$ns node]
set n4 [$ns node]
set n5 [$ns node]
$n0 shape circle
$n1 shape circle
$n2 shape circle
$n3 shape circle
$n4 shape circle
$n5 shape circle
$n0 label "1 Node"
```

```
$n1 label "2 Node"
$n2 label "3 Node"
$n3 label "4 Node"
$n4 label "5 Node"
$n5 label "6 Node"
$ns duplex-link $n0 $n2 10Mb 10ms FQ
$ns duplex-link $n1 $n2 10Mb 10ms FQ
$ns duplex-link $n2 $n3 5Mb 5ms FQ
$ns duplex-link $n3 $n4 10Mb 10ms FQ
$ns duplex-link $n3 $n5 10Mb 10ms FQ
$ns queue-limit $n0 $n2 10
$ns queue-limit $n1 $n2 10
$ns queue-limit $n2 $n3 5
$ns queue-limit $n3 $n4 10
$ns queue-limit $n3 $n5 10
$ns duplex-link-op $n0 $n2 orient down-right
$ns duplex-link-op $n1 $n2 orient up-right
$ns duplex-link-op $n2 $n3 orient right
$ns duplex-link-op $n3 $n4 orient up-right
$ns duplex-link-op $n3 $n5 orient down-right
set tcp [new Agent/TCP]
$tcp set class_ 2
$ns attach-agent $n0 $tcp
set sink [new Agent/TCPSink]
$ns attach-agent $n4 $sink
$ns connect $tcp $sink
$tcp set fid_ 1
set ftp [new Application/FTP]
$ftp attach-agent $tcp
$ftp set type_ FTP
$ftp set packet_size_ 1000
$ftp set rate_ 1mb
$ftp set interval_ 150
set udp [new Agent/UDP]
```
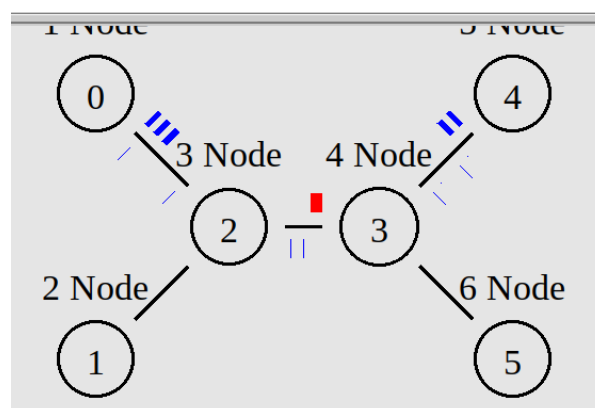
$udp set class_ 2

$ns attach-agent $n1 $udp

set sink1 [new Agent/Null]

$ns attach-agent $n5 $sink1

$ns connect $udp $sink1

$udp set fid_ 2

set cbr [new Application/Traffic/CBR]

$cbr attach-agent $udp

$cbr set type_ CBR

$cbr set packet_size_ 1500

$cbr set rate_ 0.05mb

$cbr set interval_ 150

$ns at 1.0 "$ftp start"

$ns at 4.0 "$ftp stop"

$ns at 2.0 "$cbr start"

$ns at 4.5 "$cbr stop"

$ns at 5.0 "finish"

$ns run

OUTPUT:

Droptail:

```
set ns [new Simulator]
$ns color 1 Blue
$ns color 2 Red
set nf [open prac3.nam w]
$ns namtrace-all $nf
set nr [open prac3.tr w]
$ns trace-all $nr
#Define a 'finish' procedure
proc finish {} {
    global ns nf nr
    $ns flush-trace
    #Close the NAM trace file
    close $nf
    close $nr
    #Execute NAM on the trace file
    exec nam prac3.nam &
    exec awk -f through.awk prac3.tr &
    exit 0   }
set n0 [$ns node]
set n1 [$ns node]
set n2 [$ns node]
set n3 [$ns node]
set n4 [$ns node]
set n5 [$ns node]
$n0 shape circle
$n1 shape circle
$n2 shape circle
$n3 shape circle
$n4 shape circle
$n5 shape circle
$n0 label "1 Node"
$n1 label "2 Node"
```

```
$n2 label "3 Node"
$n3 label "4 Node"
$n4 label "5 Node"
$n5 label "6 Node"
$ns duplex-link $n0 $n2 10Mb 10ms DropTail
$ns duplex-link $n1 $n2 10Mb 10ms DropTail
$ns duplex-link $n2 $n3 5Mb 5ms DropTail
$ns duplex-link $n3 $n4 10Mb 10ms DropTail
$ns duplex-link $n3 $n5 10Mb 10ms DropTail
$ns queue-limit $n0 $n2 10
$ns queue-limit $n1 $n2 10
$ns queue-limit $n2 $n3 5
$ns queue-limit $n3 $n4 10
$ns queue-limit $n3 $n5 10
$ns duplex-link-op $n0 $n2 orient down-right
$ns duplex-link-op $n1 $n2 orient up-right
$ns duplex-link-op $n2 $n3 orient right
$ns duplex-link-op $n3 $n4 orient up-right
$ns duplex-link-op $n3 $n5 orient down-right
set tcp [new Agent/TCP]
$tcp set class_ 2
$ns attach-agent $n0 $tcp
set sink [new Agent/TCPSink]
$ns attach-agent $n4 $sink
$ns connect $tcp $sink
$tcp set fid_ 1
set ftp [new Application/FTP]
$ftp attach-agent $tcp
$ftp set type_ FTP
$ftp set packet_size_ 1000
$ftp set rate_ 1mb
$ftp set interval_ 150
set udp [new Agent/UDP]
$udp set class_ 2
```
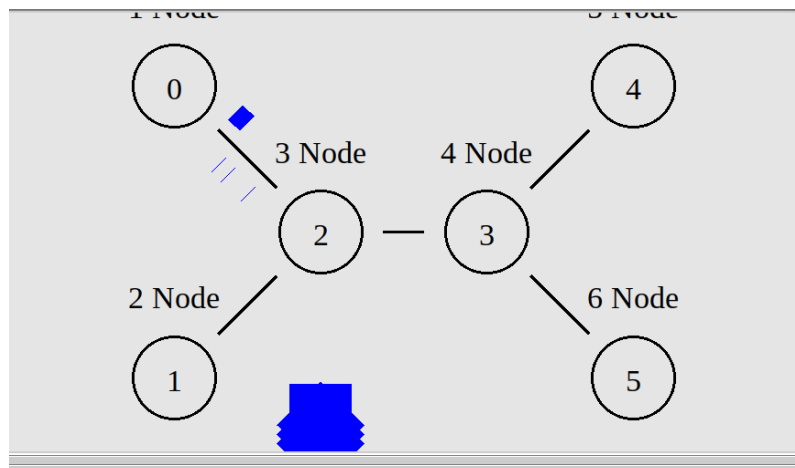
$ns attach-agent $n1 $udp

set sink1 [new Agent/Null]

$ns attach-agent $n5 $sink1

$ns connect $udp $sink1

$udp set fid_ 2

set cbr [new Application/Traffic/CBR]

$cbr attach-agent $udp

$cbr set type_ CBR

$cbr set packet_size_ 1500

$cbr set rate_ 0.05mb

$cbr set interval_ 150

$ns at 1.0 "$ftp start"

$ns at 4.0 "$ftp stop"

$ns at 2.0 "$cbr start"

$ns at 4.5 "$cbr stop"

$ns at 5.0 "finish"

OUTPUT:

RED:

```
set ns [new Simulator]
$ns color 1 Blue
$ns color 2 Red
set nf [open prac3.nam w]
$ns namtrace-all $nf
set nr [open prac3.tr w]
$ns trace-all $nr
#Define a 'finish' procedure
proc finish {} {
    global ns nf nr
    $ns flush-trace
    #Close the NAM trace file
    close $nf
    close $nr
    #Execute NAM on the trace file
    exec nam prac3.nam &
    exec awk -f through.awk prac3.tr &
    exit 0   }
set n0 [$ns node]
set n1 [$ns node]
set n2 [$ns node]
set n3 [$ns node]
set n4 [$ns node]
set n5 [$ns node]
$n0 shape circle
$n1 shape circle
$n2 shape circle
$n3 shape circle
$n4 shape circle
$n5 shape circle
$n0 label "1 Node"
$n1 label "2 Node"
```

```
$n2 label "3 Node"
$n3 label "4 Node"
$n4 label "5 Node"
$n5 label "6 Node"
$ns duplex-link $n0 $n2 10Mb 10ms RED
$ns duplex-link $n1 $n2 10Mb 10ms RED
$ns duplex-link $n2 $n3 5Mb 5ms RED
$ns duplex-link $n3 $n4 10Mb 10ms RED
$ns duplex-link $n3 $n5 10Mb 10ms RED
$ns queue-limit $n0 $n2 10
$ns queue-limit $n1 $n2 10
$ns queue-limit $n2 $n3 5
$ns queue-limit $n3 $n4 10
$ns queue-limit $n3 $n5 10
$ns duplex-link-op $n0 $n2 orient down-right
$ns duplex-link-op $n1 $n2 orient up-right
$ns duplex-link-op $n2 $n3 orient right
$ns duplex-link-op $n3 $n4 orient up-right
$ns duplex-link-op $n3 $n5 orient down-right
set tcp [new Agent/TCP]
$tcp set class_ 2
$ns attach-agent $n0 $tcp
set sink [new Agent/TCPSink]
$ns attach-agent $n4 $sink
$ns connect $tcp $sink
$tcp set fid_ 1
set ftp [new Application/FTP]
$ftp attach-agent $tcp
$ftp set type_ FTP
$ftp set packet_size_ 1000
$ftp set rate_ 1mb
$ftp set interval_ 150
set udp [new Agent/UDP]
$udp set class_ 2
```
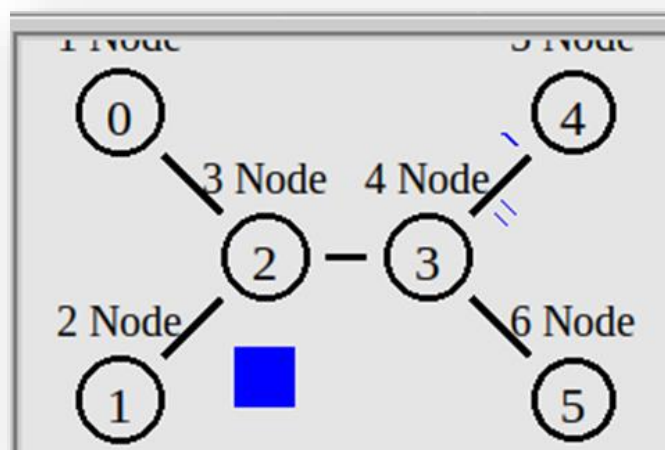
$ns attach-agent $n1 $udp

set sink1 [new Agent/Null]

$ns attach-agent $n5 $sink1

$ns connect $udp $sink1

$udp set fid_ 2

set cbr [new Application/Traffic/CBR]

$cbr attach-agent $udp

$cbr set type_ CBR

$cbr set packet_size_ 1500

$cbr set rate_ 0.05mb

$cbr set interval_ 150

$ns at 1.0 "$ftp start"

$ns at 4.0 "$ftp stop"

$ns at 2.0 "$cbr start"

$ns at 4.5 "$cbr stop"

$ns at 5.0 "finish"

$ns run

OUTPUT:

CONCLUSION:

- By performing the above practicals, we learned the the basic concepts of NS2.

- We also learnt about how to create a topology in NS2.

- We also learnt about how to transfer data.

- We also learnt various queing techniques.