PRACTICAL-9

AIM:

Create ring and mesh topology in NS2 and implement distance vector algorithm for dynamic routing. Analyze the behavior for transferring the packet when a link goes down.

THEORY:

Mesh Topology:

- In mesh topology each device is connected to every other device on the network through a dedicated point-to-point link.
- When we say dedicated it means that the link only carries data for the two connected devices only.
- Let's say we have n devices in the network then each device must be connected with (n-1) devices of the network.
- Number of links in a mesh topology of n devices would be n(n-1)/2.

Ring Topology:

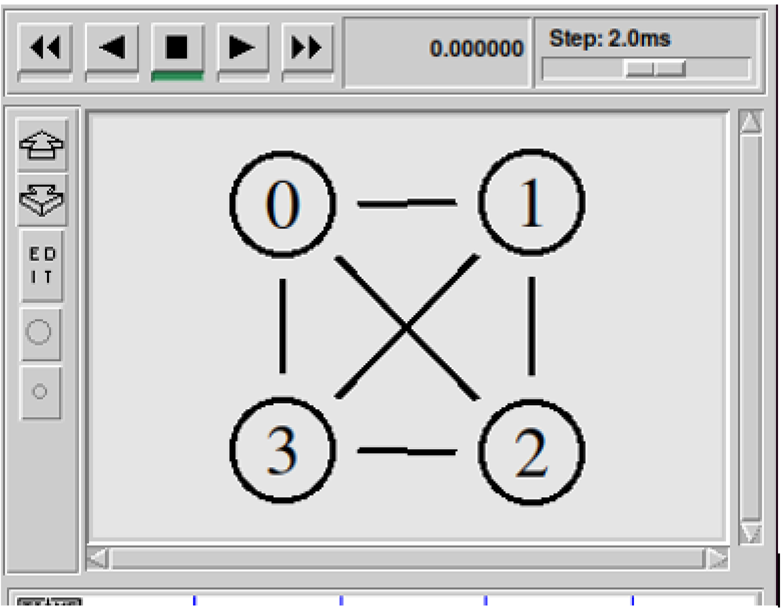
- In ring topology each device is connected with the two devices on either side of it.
- There are two dedicated point to point links a device has with the devices on the either side of it.
- This structure forms a ring thus it is known as ring topology.
- If a device wants to send data to another device then it sends the data in one direction, each device in ring topology has a repeater, if the received data is intended for other device then repeater forwards this data until the intended device receives it.

Dynamic Routing:

- Dynamic routing, also called adaptive routing, is a process where a router can forward data via a different route or given destination based on the current conditions of the communication circuits within a system.
- Dynamic routing, also called adaptive routing, is a process where a router can forward data via a different route or given destination based on the current conditions of the communication circuits within a system.

PRACTICAL IMPLEMENTATION:

MESH TOPOLOGY:



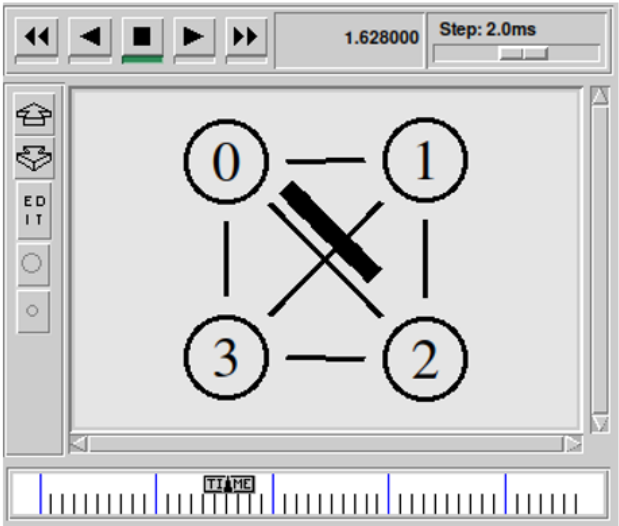
PROGRAM CODE:

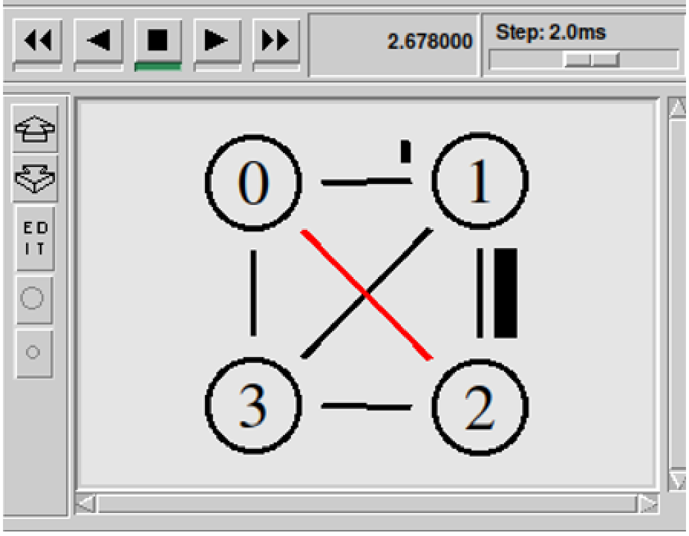
```
set ns [new Simulator]

$ns rtproto DV

setnf [open prac8.nam w]

$ns namtrace-all $nf

proc finish {} {

global ns nf

$ns flush-trace

close $nf

execnam prac8.nam &

exit 0

}

set n0 [$ns node]

set n1 [$ns node]

set n2 [$ns node]

set n3 [$ns node]

$ns duplex-link $n0 $n1 10Mb 10ms DropTail

$ns duplex-link $n1 $n2 10Mb 10ms DropTail

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

$ns duplex-link $n3 $n0 10Mb 10ms DropTail

$ns duplex-link $n0 $n2 10Mb 20ms DropTail

$ns duplex-link $n1 $n3 10Mb 20ms DropTail
```

```
$ns duplex-link-op $n0 $n1 orient right

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

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

$ns duplex-link-op $n3 $n0 orient up

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

$ns duplex-link-op $n1 $n3 orient left-down


settcp [new Agent/TCP]

$tcp set class_ 2

$ns attach-agent $n0 $tcp

set sink [new Agent/TCPSink]

$ns attach-agent $n2 $sink

$ns connect $tcp $sink


set ftp [new Application/FTP]

$ftp attach-agent $tcp

$ftp set type_ FTP

$ftp set packet_sie_ 1000

$ftp set rate_ 1mb


$ns at 1.0 "$ftp start"

$ns rtmodel-at 2.0 down $n0 $n2

$ns rtmodel-at 3.0 up $n0 $n2
```

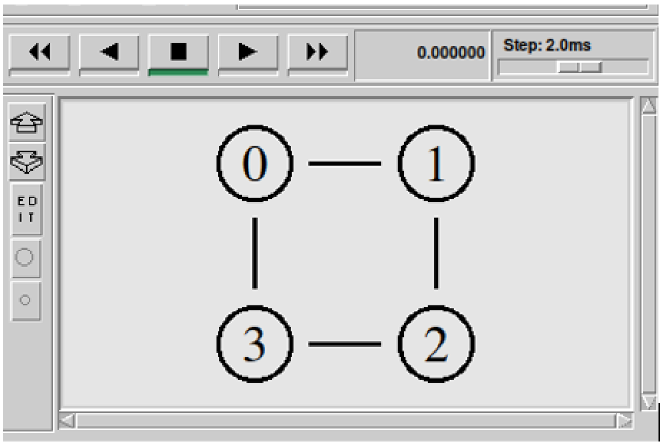
```
$ns at 4.0 "$ftp stop"

$ns at 5.0 "finish"

$ns run
```

OUTPUT:


1.628000
Step: 2.0ms
EDIT
TIME
0
1
2
3


2.678000
Step: 2.0ms
EDIT
0
1
2
3

RING TOPOLOGY:



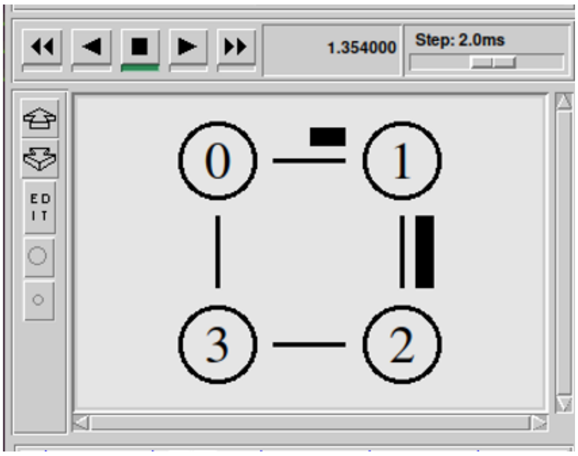
PROGRAM CODE:

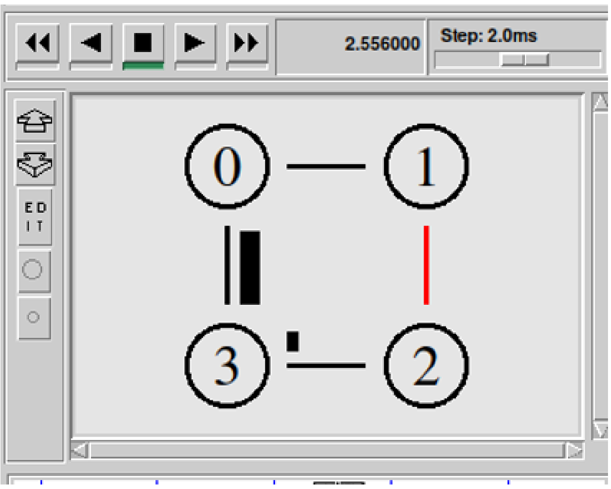
```
set ns [new Simulator]

$ns rtproto DV

setnf [open prac8.nam w]

$ns namtrace-all $nf

proc finish {} {

global ns nf

$ns flush-trace

close $nf

execnam prac8.nam &

exit 0

}
```

```
set n0 [$ns node]
set n1 [$ns node]
set n2 [$ns node]
set n3 [$ns node]
$ns duplex-link $n0 $n1 10Mb 10ms DropTail
$ns duplex-link $n1 $n2 10Mb 10ms DropTail
$ns duplex-link $n2 $n3 10Mb 10ms DropTail
$ns duplex-link $n3 $n0 10Mb 10ms DropTail
$ns duplex-link-op $n0 $n1 orient right
$ns duplex-link-op $n1 $n2 orient down
$ns duplex-link-op $n2 $n3 orient left
$ns duplex-link-op $n3 $n0 orient up
settcp [new Agent/TCP]
$tcp set class_ 2
$ns attach-agent $n0 $tcp
set sink [new Agent/TCPSink]
$ns attach-agent $n2 $sink
$ns connect $tcp $sink
set ftp [new Application/FTP]
$ftp attach-agent $tcp
$ftp set type_ FTP
$ftp set packet_sie_ 1000
$ftp set rate_ 1mb
$ns at 1.0 "$ftp start"
```

```
$ns rtmodel-at 2.0 down $n1 $n2

$ns rtmodel-at 3.0 up $n1 $n2

$ns at 4.0 "$ftp stop"

$ns at 5.0 "finish"

$ns run
```

OUTPUT:





CONCLUSION:

- In this practical, we learned how to create mesh and ring topology in NS2.