

## NETWORKS LAB EXERCISE 10

*Name: Jayannthan PT*

*Dept: CSE 'A'*

*Roll No.: 205001049*

### Simulation of Routing protocol

#### Aim:

Write tcl script to simulate the routing protocols in wired networks.

#### Code:

##### Distance Vector Routing Protocol

```
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 n(0) [$ns node]
set n(1) [$ns node]
set n(2) [$ns node]
set n(3) [$ns node]
set n(4) [$ns node]
set n(5) [$ns node]
set n(6) [$ns node]
set n(7) [$ns node]
set n(8) [$ns node]
set n(9) [$ns node]
set n(10) [$ns node]
set n(11) [$ns node]
```

```
for {set i 0} {$i < 8} {incr i} {
$ns duplex-link $n($i) $n([expr $i+1]) 1Mb 10ms DropTail }
$ns duplex-link $n(0) $n(8) 1Mb 10ms DropTail
$ns duplex-link $n(1) $n(10) 1Mb 10ms DropTail
$ns duplex-link $n(0) $n(9) 1Mb 10ms DropTail
$ns duplex-link $n(9) $n(11) 1Mb 10ms DropTail
$ns duplex-link $n(10) $n(11) 1Mb 10ms DropTail
$ns duplex-link $n(11) $n(5) 1Mb 10ms DropTail

$ns duplex-link-op $n(0) $n(1) orient right-down
$ns duplex-link-op $n(1) $n(2) orient right-down
$ns duplex-link-op $n(2) $n(3) orient down
$ns duplex-link-op $n(3) $n(4) orient left-down
$ns duplex-link-op $n(4) $n(5) orient left-down
$ns duplex-link-op $n(5) $n(6) orient left-up
$ns duplex-link-op $n(6) $n(7) orient left-up
$ns duplex-link-op $n(7) $n(8) orient up
$ns duplex-link-op $n(8) $n(0) orient right-up
$ns duplex-link-op $n(11) $n(5) orient up
$ns duplex-link-op $n(9) $n(11) orient right
$ns duplex-link-op $n(10) $n(11) orient left

set udp1 [new Agent/UDP]
$ns attach-agent $n(0) $udp1
set null [new Agent/Null]
$ns attach-agent $n(5) $null
$ns connect $udp1 $null
$udp1 set fid_ 1

set udp2 [new Agent/UDP]
$ns attach-agent $n(1) $udp2
set null [new Agent/Null]
$ns attach-agent $n(5) $null
$ns connect $udp2 $null
$udp2 set fid_ 2

set cbr1 [new Application/Traffic/CBR]
$cbr1 attach-agent $udp1
$cbr1 set type_ CBR
$cbr1 set packet_size_ 1000
$cbr1 set rate_ 1mb
$cbr1 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
```

```

$ns rtproto DV

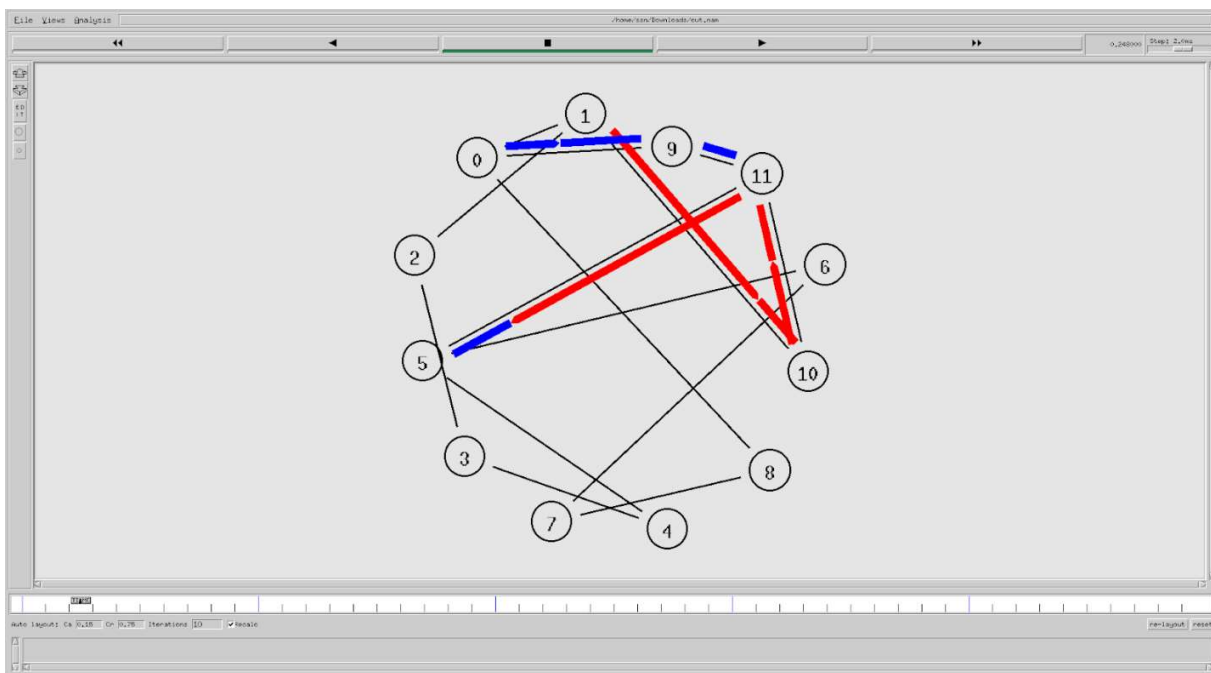
$ns rtmodel-at 1.0 down $n(11) $n(5)
$ns rtmodel-at 2.0 down $n(7) $n(6)
$ns rtmodel-at 2.0 up $n(11) $n(5)
$ns rtmodel-at 3.0 up $n(7) $n(6)

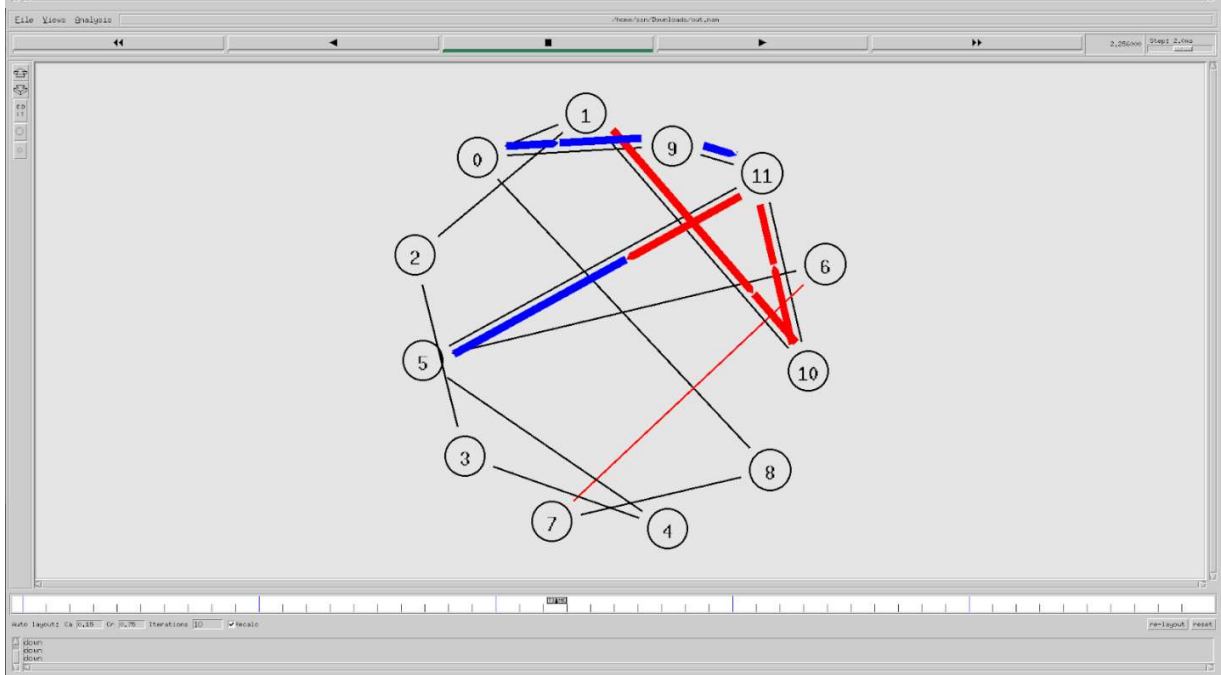
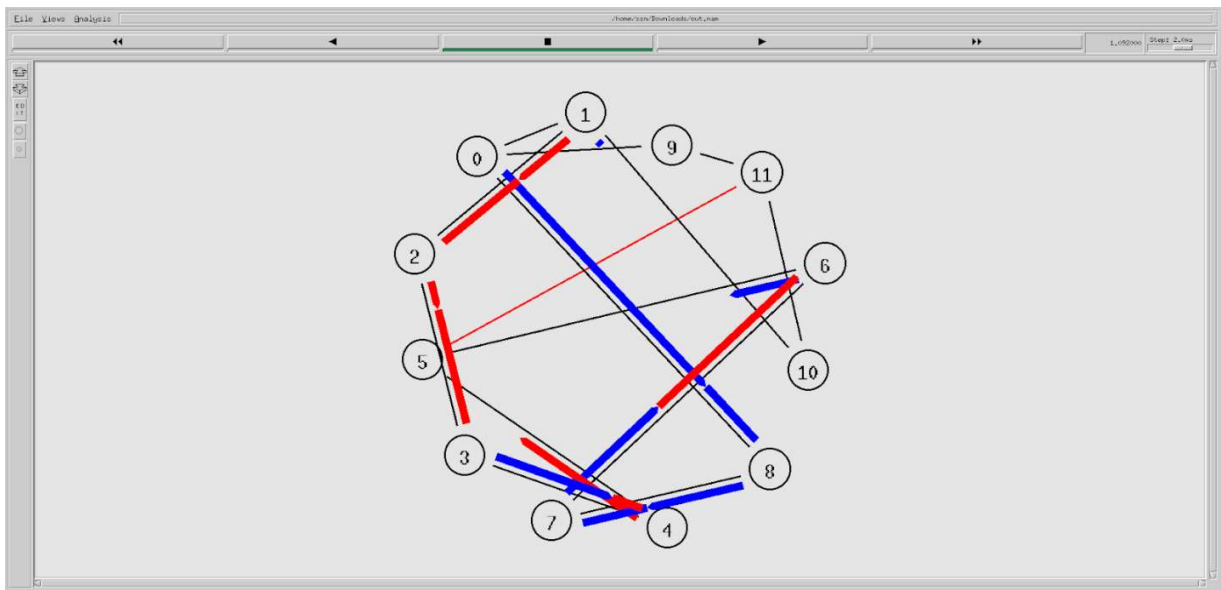
$ns at 0.1 "$cbr1 start"
$ns at 0.2 "$cbr2 start"
$ns at 4.5 "$ns detach-agent $n(0) $udp1 ; $ns detach-agent $n(5) $null ; $ns detach-agent $n(1) $udp2"
$ns at 5.0 "finish"

$ns run

```

## Output:





## Code:

### Link State Routing Protocol

```
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 n(0) [$ns node]
set n(1) [$ns node]
set n(2) [$ns node]
set n(3) [$ns node]
set n(4) [$ns node]
set n(5) [$ns node]
set n(6) [$ns node]
set n(7) [$ns node]
set n(8) [$ns node]
set n(9) [$ns node]
set n(10) [$ns node]
set n(11) [$ns node]

for {set i 0} {$i < 8} {incr i} {
    $ns duplex-link $n($i) $n([expr $i+1]) 1Mb 10ms DropTail }
    $ns duplex-link $n(0) $n(8) 1Mb 10ms DropTail
    $ns duplex-link $n(1) $n(10) 1Mb 10ms DropTail
    $ns duplex-link $n(0) $n(9) 1Mb 10ms DropTail
    $ns duplex-link $n(9) $n(11) 1Mb 10ms DropTail
    $ns duplex-link $n(10) $n(11) 1Mb 10ms DropTail
    $ns duplex-link $n(11) $n(5) 1Mb 10ms DropTail

    $ns duplex-link-op $n(0) $n(1) orient right-down
    $ns duplex-link-op $n(1) $n(2) orient right-down
    $ns duplex-link-op $n(2) $n(3) orient down
    $ns duplex-link-op $n(3) $n(4) orient left-down
    $ns duplex-link-op $n(4) $n(5) orient left-down
    $ns duplex-link-op $n(5) $n(6) orient left-up
    $ns duplex-link-op $n(6) $n(7) orient left-up
    $ns duplex-link-op $n(7) $n(8) orient up
    $ns duplex-link-op $n(8) $n(0) orient right-up
    $ns duplex-link-op $n(11) $n(5) orient up
```

```

$ns duplex-link-op $n(9) $n(11) orient right
$ns duplex-link-op $n(10) $n(11) orient left

set tcp1 [new Agent/TCP]
$ns attach-agent $n(0) $tcp1
set sink [new Agent/TCPSink]
$ns attach-agent $n(5) $sink
$ns connect $tcp1 $sink
$tcp1 set fid_ 1

set tcp2 [new Agent/TCP]
$ns attach-agent $n(1) $tcp2
set sink [new Agent/TCPSink]
$ns attach-agent $n(5) $sink
$ns connect $tcp2 $sink
$tcp2 set fid_ 2

set cbr1 [new Application/Traffic/CBR]
$cbr1 attach-agent $tcp1
$cbr1 set type_ CBR
$cbr1 set packet_size_ 1000
$cbr1 set rate_ 1mb
$cbr1 set random_ false

set cbr2 [new Application/Traffic/CBR]
$cbr2 attach-agent $tcp2
$cbr2 set type_ CBR
$cbr2 set packet_size_ 1000
$cbr2 set rate_ 1mb
$cbr2 set random_ false

$ns rtproto DV

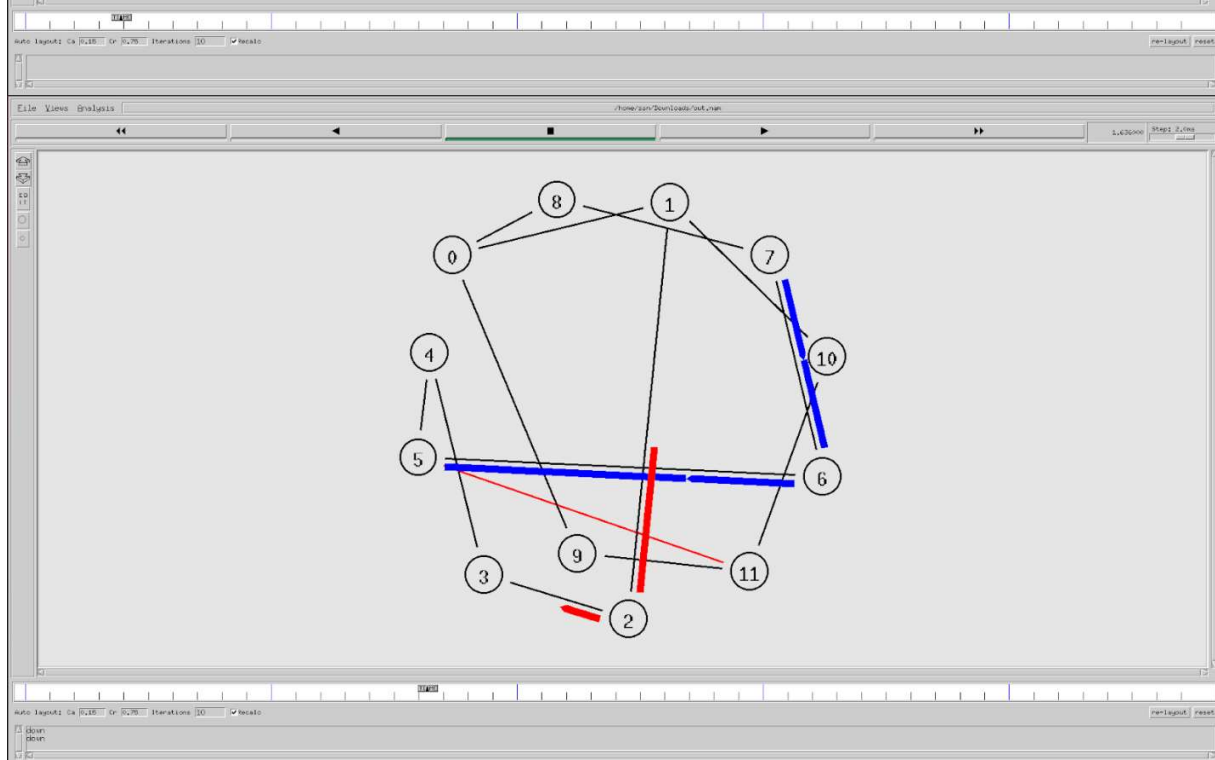
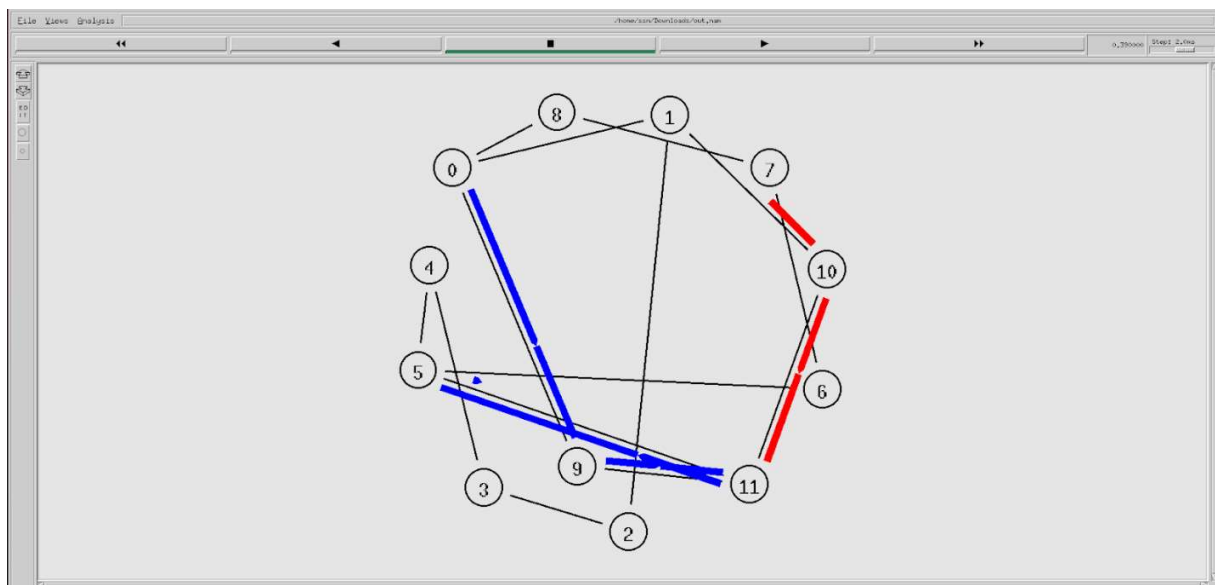
$ns rtmodel-at 1.0 down $n(11) $n(5)
$ns rtmodel-at 2.0 down $n(7) $n(6)
$ns rtmodel-at 2.0 up $n(11) $n(5)
$ns rtmodel-at 3.0 up $n(7) $n(6)

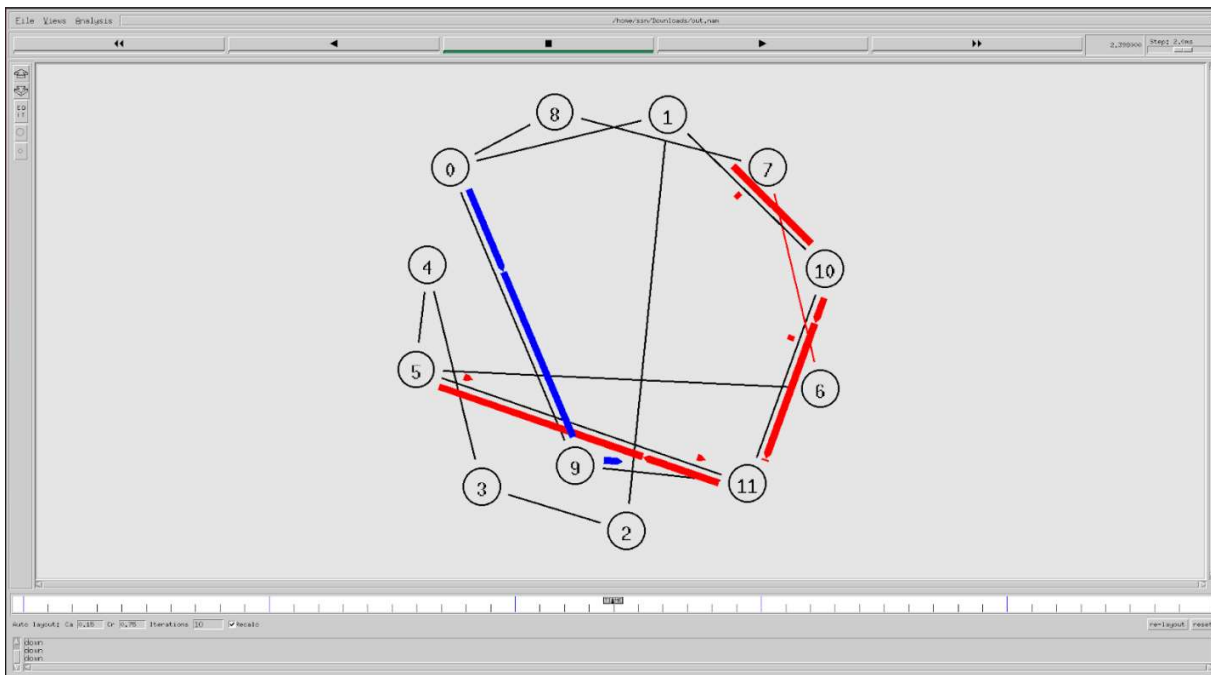
$ns at 0.1 "$cbr1 start"
$ns at 0.2 "$cbr2 start"
$ns at 4.5 "$ns detach-agent $n(0) $tcp1 ; $ns detach-agent $n(5) $sink ; $ns detach-agent
$n(1) $tcp2"
$ns at 5.0 "finish"

$ns run

```

**Output:**





### Learning outcome:

Learnt to implement simulate the simulate the routing protocols in wired networks.

---