
Assignment 10 – Routing Protocol

Date: 04/11/2022

Aim:

To write a TCL script to simulate the routing protocols in wired networks.

Algorithm:

Distance vector routing protocol:

1. Create 12 nodes and the links between the nodes as:
    a. 0 → 8 - 1Mb 10 ms duplex link droptail
    b. 1 → 10 - 1Mb 10 ms duplex link droptail
    c. 0 → 9 - 1Mb 10 ms duplex link droptail
    d. 9 → 11 - 1Mb 10 ms duplex link droptail
    e. 10 → 11 - 1Mb 10 ms duplex link droptail
    f. 11 → 5 - 1Mb 10 ms duplex link droptail
2. Align all nodes properly.
3. Set up UDP connections over 0 and 5, 1 and 5 with flow id, type, packet size, rate, random fields.
4. Set different colors for different flows.
5. Use distance vector routing protocol.
6. Make links 11-5 and 7-6 down for 1 second.
7. Run the simulation for 5 seconds, and show the simulation in network animator and in the trace file.

Link state routing protocol:

1. Create 12 nodes and the links between the nodes as:
    a. 0 → 8 - 1Mb 10 ms duplex link droptail
    b. 1 → 10 - 1Mb 10 ms duplex link droptail
    c. 0 → 9 - 1Mb 10 ms duplex link droptail
    d. 9 → 11 - 1Mb 10 ms duplex link droptail
    e. 10 → 11 - 1Mb 10 ms duplex link droptail

f. 11 → 5 - 1Mb 10 ms duplex link droptail

2. Align all nodes properly.
3. Set up UDP connections over 0 and 5, 1 and 5 with flow id, type, packet size, rate, random fields.
4. Set different colors for different flows.
5. Use link state routing protocol.
6. Make links 11-5 and 7-6 down for 1 second.
7. Run the simulation for 5 seconds, and show the simulation in network animator and in the trace file.

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
```

Link state routing protocol:

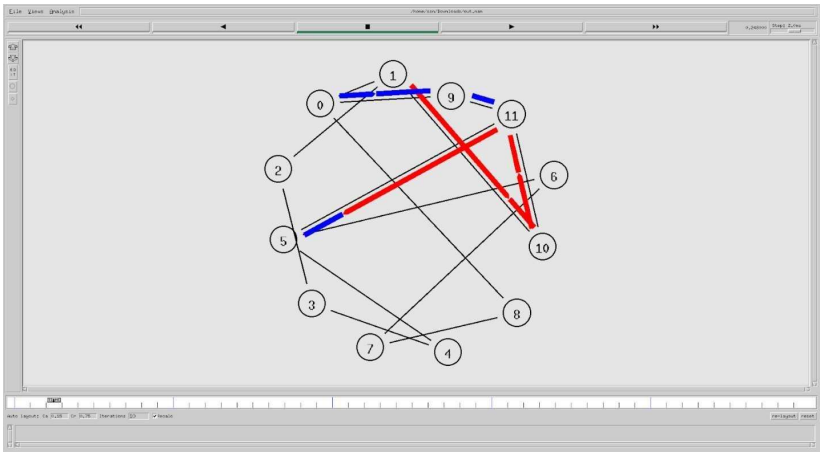
```
set ns [new Simulator]

$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
```

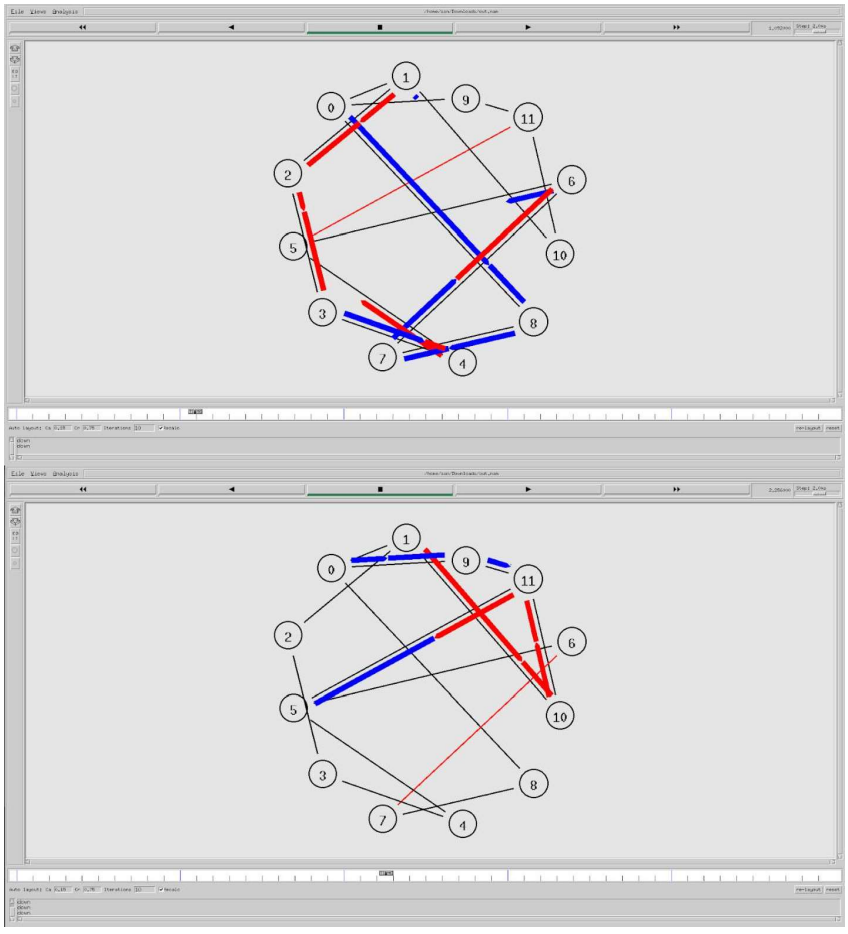
```
    exec nam out.nam
    & exit 0
}

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 udp1 [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) $udp2"
$ns at 5.0 "finish"

$ns run
```

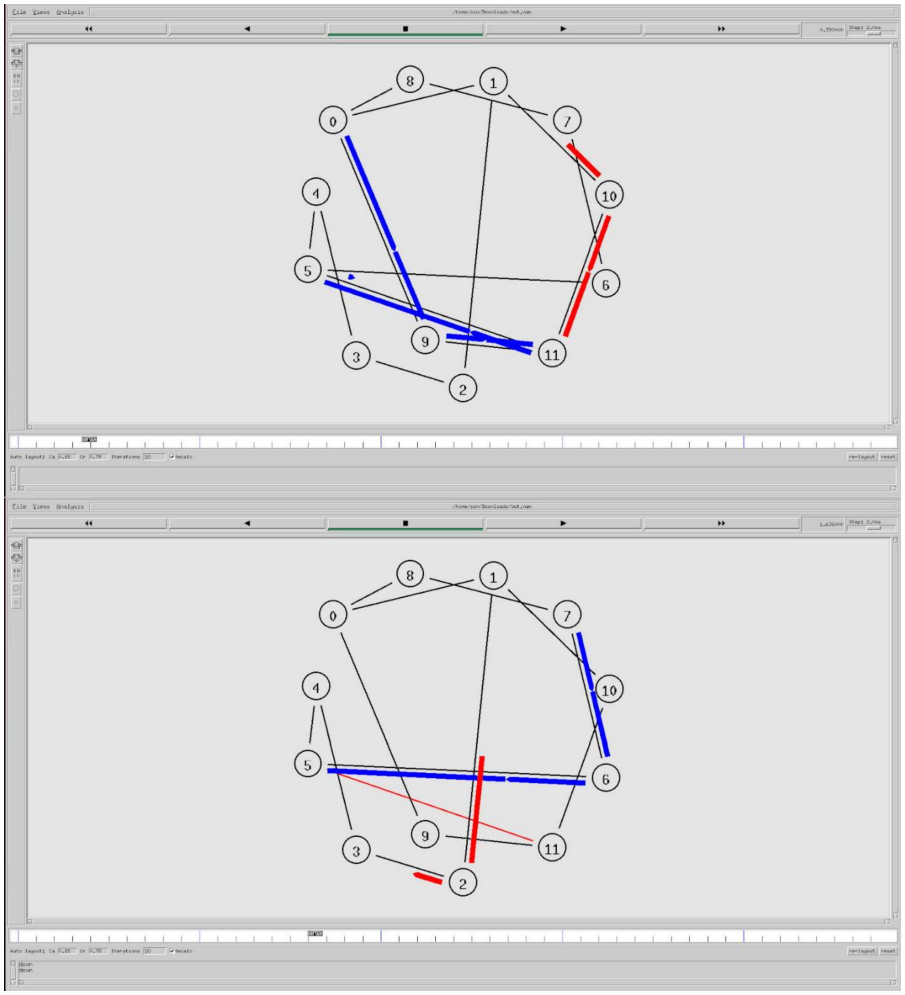
Output:

Distance Vector Routing Protocol:





Link State Routing Protocol:



Learning outcomes:

- Various routing protocols were studied.
- The distance vector routing protocol and the link state routing protocol were understood.
- The implementation of routing protocols using NS2 was understood.