



Computer Networks LAB

Routing Algorithms using NS2

20.10.2020

V Vishal Bharadwaj

2018103081

Link State Routing:

```
set val(stop) 10.0 ;
```

```
set ns [new Simulator]
```

```
set tracefile [open p1.tr w]
```

```
$ns trace-all $tracefile
```

```
set namfile [open p1.nam w]
```

```
$ns namtrace-all $namfile
```

```
set n0 [$ns node]
```

```
set n1 [$ns node]
```

```
set n2 [$ns node]
```

```
set n3 [$ns node]
```

```
set n4 [$ns node]
```

```
set n5 [$ns node]
```

```
$ns duplex-link $n0 $n1 100.0Mb 10ms DropTail
```

```
$ns queue-limit $n0 $n1 50
```

```
$ns duplex-link $n1 $n3 100.0Mb 10ms DropTail
```

```
$ns queue-limit $n1 $n3 50
```

```
$ns duplex-link $n3 $n5 100.0Mb 10ms DropTail
```

```
$ns queue-limit $n3 $n5 50
```

```
$ns duplex-link $n0 $n2 100.0Mb 10ms DropTail
```

```
$ns queue-limit $n0 $n2 50
```

```
$ns duplex-link $n2 $n4 100.0Mb 10ms DropTail
$ns queue-limit $n2 $n4 50
$ns duplex-link $n4 $n5 100.0Mb 10ms DropTail
$ns queue-limit $n4 $n5 50
$ns duplex-link $n2 $n3 100.0Mb 10ms DropTail
$ns queue-limit $n2 $n3 50
$ns duplex-link $n0 $n3 100.0Mb 10ms DropTail
$ns queue-limit $n0 $n3 50
```

```
$ns duplex-link-op $n0 $n1 orient right-up
$ns duplex-link-op $n1 $n3 orient right
$ns duplex-link-op $n3 $n5 orient right-down
$ns duplex-link-op $n0 $n2 orient right-down
$ns duplex-link-op $n2 $n4 orient right
$ns duplex-link-op $n4 $n5 orient right-up
$ns duplex-link-op $n2 $n3 orient right-up
$ns duplex-link-op $n0 $n3 orient right-up
```

```
$ns cost $n0 $n1 1
$ns cost $n0 $n2 1
$ns cost $n0 $n3 3
```

```
$ns cost $n1 $n0 1
$ns cost $n1 $n3 2
```

```
$ns cost $n2 $n0 1
$ns cost $n2 $n3 2
```

\$ns cost \$n2 \$n4 1

\$ns cost \$n3 \$n2 2

\$ns cost \$n3 \$n1 2

\$ns cost \$n3 \$n0 3

\$ns cost \$n3 \$n5 3

\$ns cost \$n4 \$n2 1

\$ns cost \$n4 \$n5 3

\$ns cost \$n5 \$n3 3

\$ns cost \$n5 \$n4 3

set udp0 [new Agent/UDP]

\$ns attach-agent \$n0 \$udp0

set null1 [new Agent/Null]

\$ns attach-agent \$n5 \$null1

\$ns connect \$udp0 \$null1

\$udp0 set packetSize_ 1500

set cbr0 [new Application/Traffic/CBR]

\$cbr0 attach-agent \$udp0

\$cbr0 set packetSize_ 1000

\$cbr0 set rate_ 1.0Mb

\$cbr0 set random_ null

```
$ns at 1.0 "$cbr0 start"
```

```
$ns at 2.0 "$cbr0 stop"
```

```
$ns rtproto LS
```

```
proc finish {} {
```

```
    global ns tracefile namfile
```

```
    $ns flush-trace
```

```
    close $tracefile
```

```
    close $namfile
```

```
    exec nam p1.nam &
```

```
    exit 0
```

```
}
```

```
$ns at $val(stop) "$ns nam-end-wireless $val(stop)"
```

```
$ns at $val(stop) "finish"
```

```
$ns at $val(stop) "puts \"done\" ; $ns halt"
```

```
$ns run
```

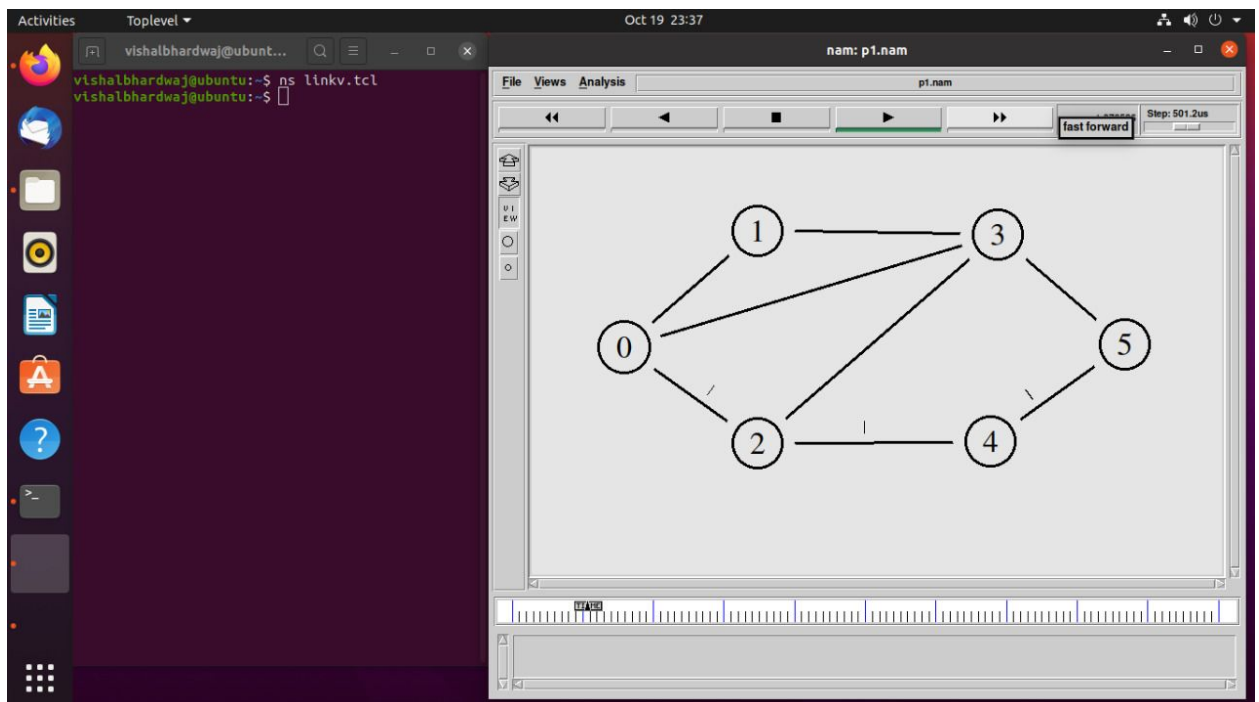
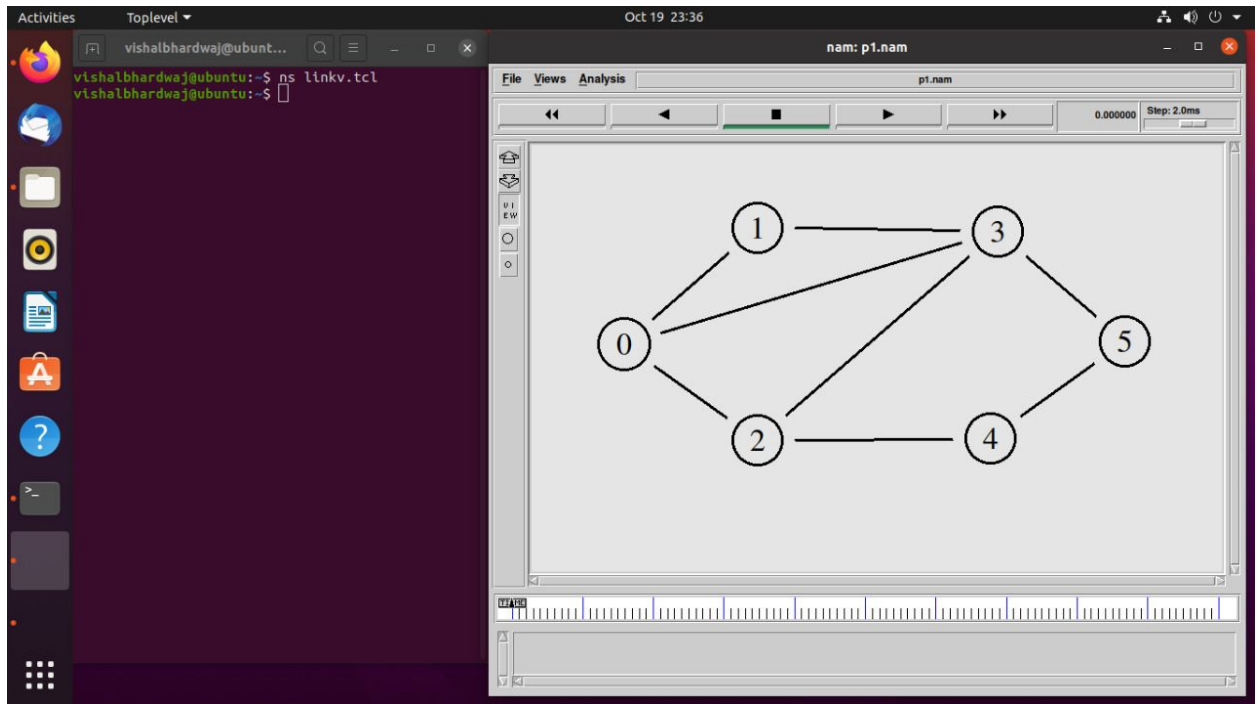
The adjacency matrix representation of nodes and the cost between them:

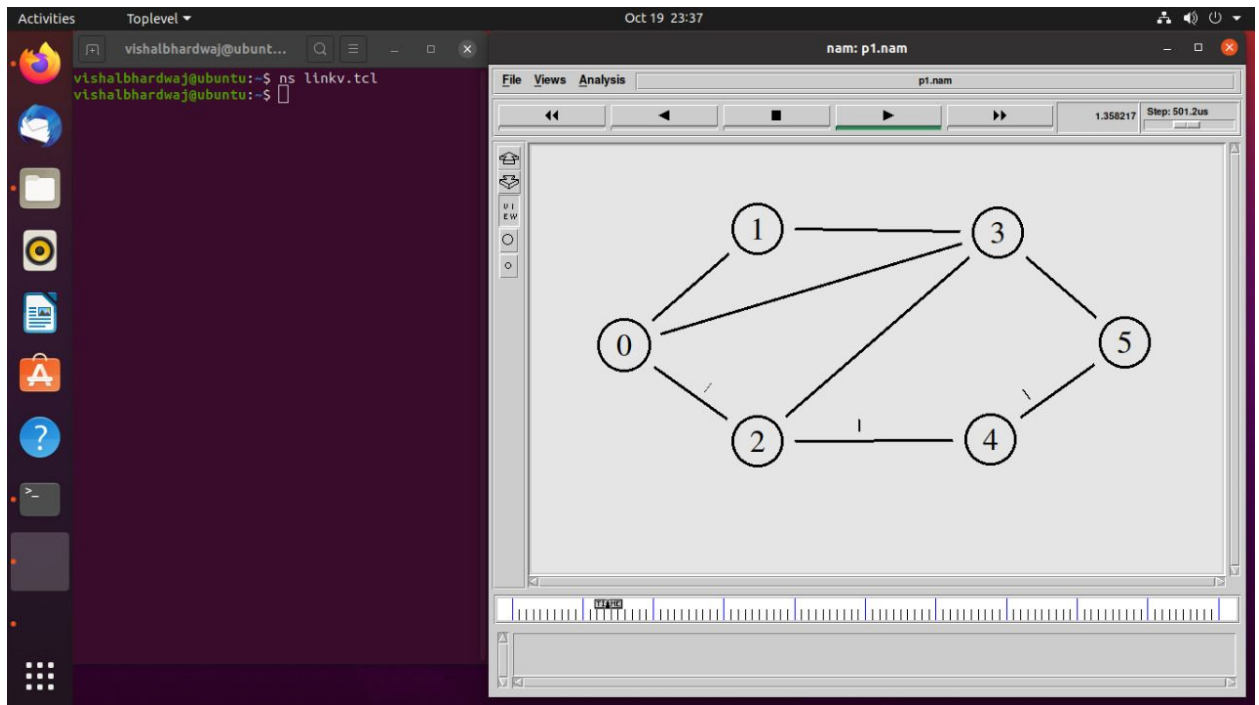
Nodes -->	0	1	2	3	4	5
0	1	1	3	∞	∞	∞
1	1	∞	∞	2	∞	∞
2	1	∞	∞	2	1	∞
3	3	2	2	∞	∞	3
4	∞	∞	1	∞	∞	3
5	∞	∞	∞	3	3	∞

Specifications:

- Bandwidth 100Mbps
- Delay 10ms
- Source : n_0
- Destination: n_5
- Queue: Droptail

Simulation:





So we go from n_0 to n_5 as $n_0 \rightarrow n_1 \rightarrow n_3 \rightarrow n_5 \rightarrow n_6$.

Distance Vector Routing:

```
set ns [new Simulator]
$ns rtproto DV
set nf [open prac3.nam w]
$ns namtrace-all $nf
proc finish {} {
    global nf ns
    $ns flush-trace
    close $nf
    exec nam prac3.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
```

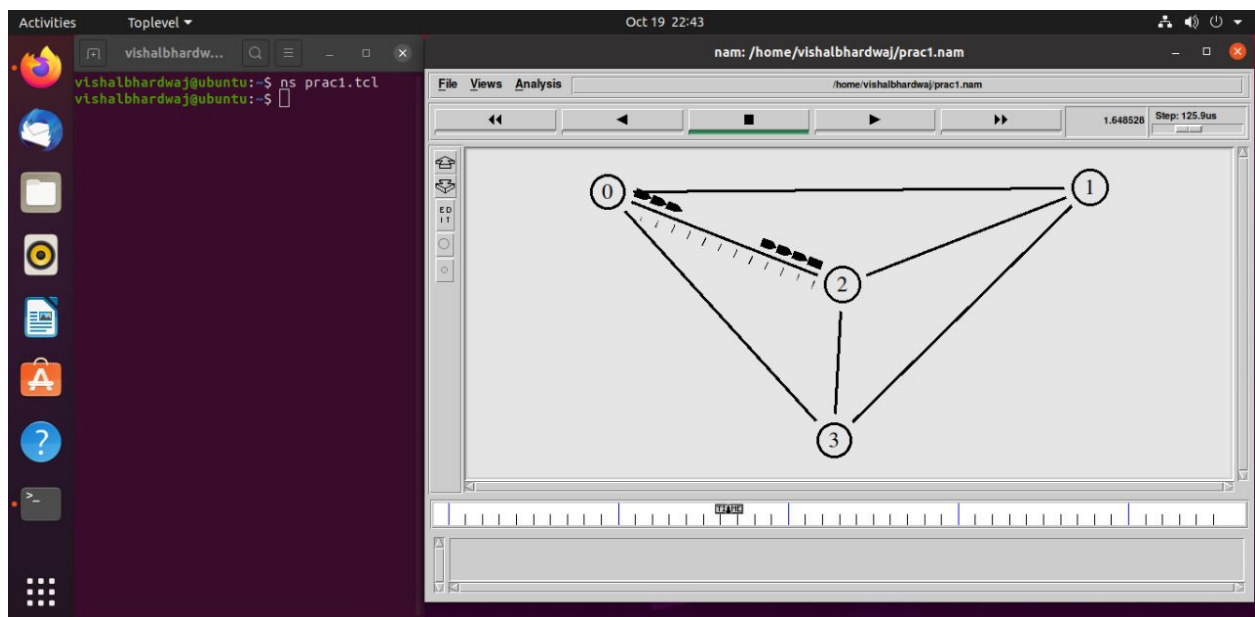
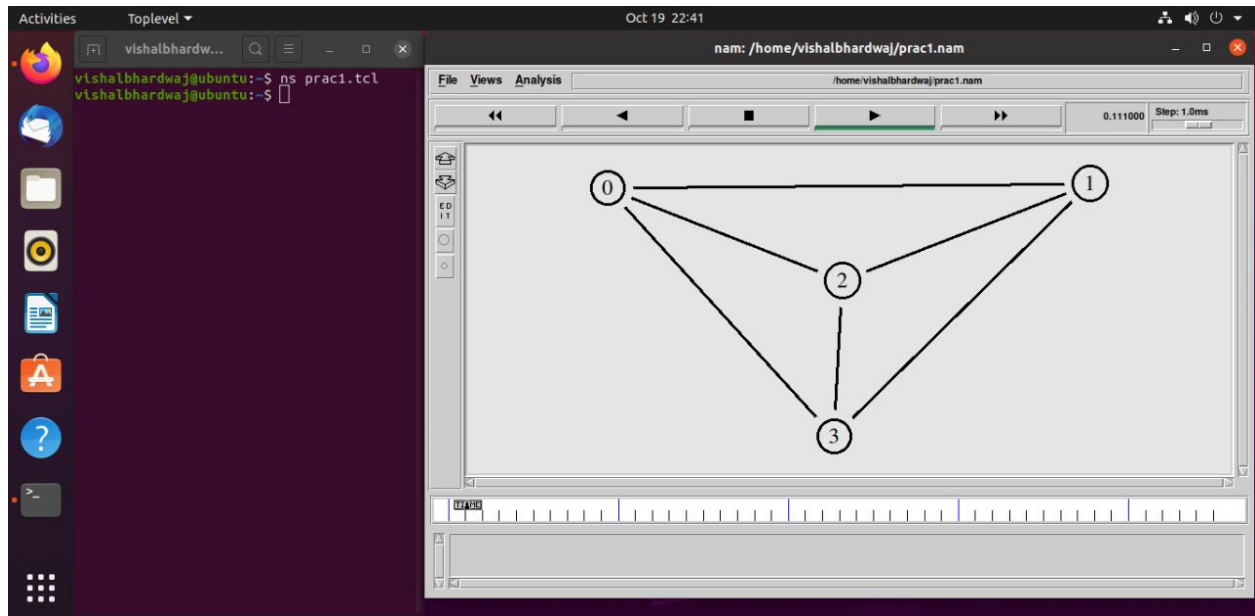


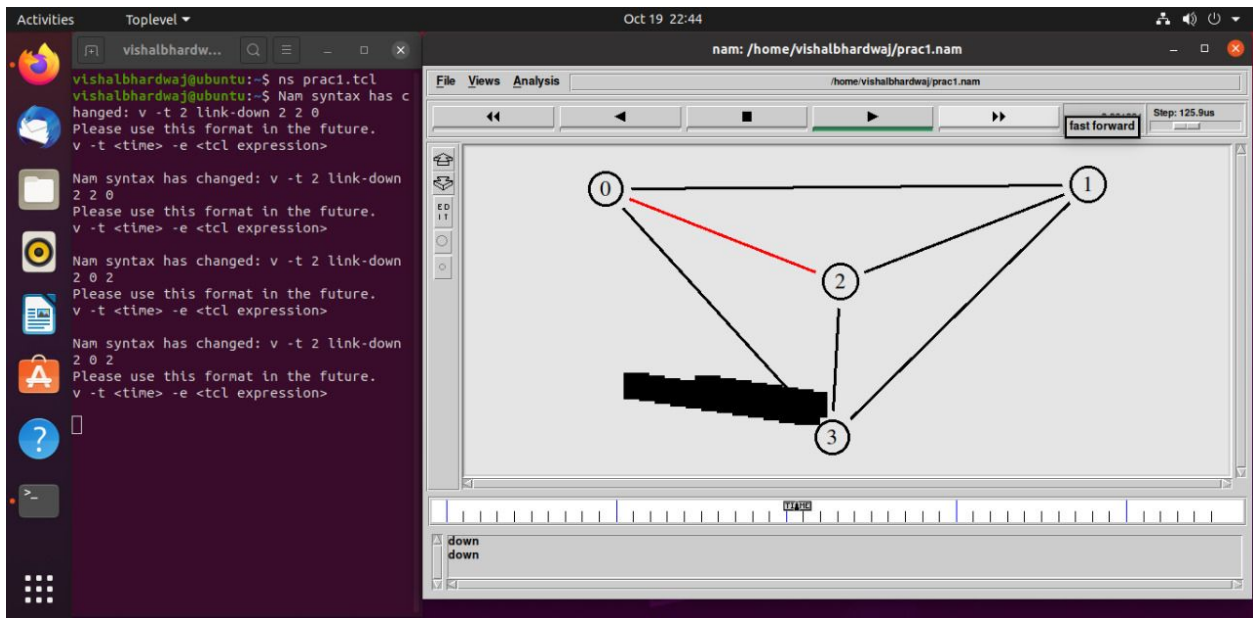
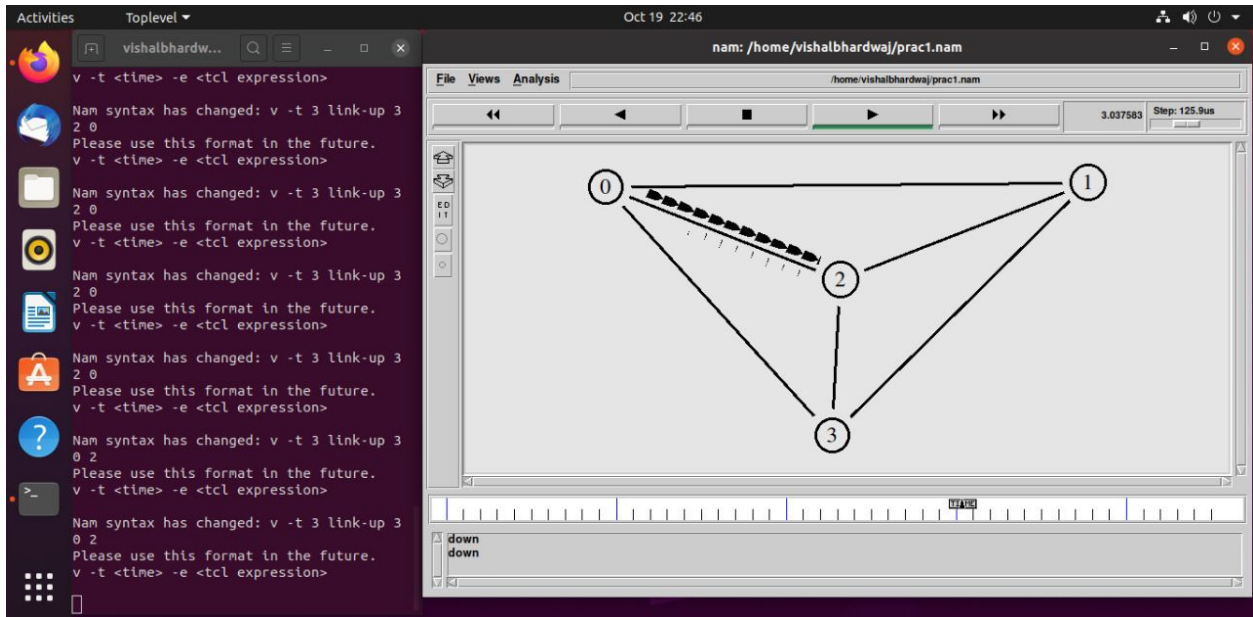
```
$ns duplex-link $n2 $n3 10Mb 10ms DropTail
$ns duplex-link $n3 $n0 10Mb 10ms DropTail
$ns duplex-link $n0 $n2 10Mb 10ms DropTail
$ns duplex-link $n1 $n3 10Mb 10ms DropTail
```

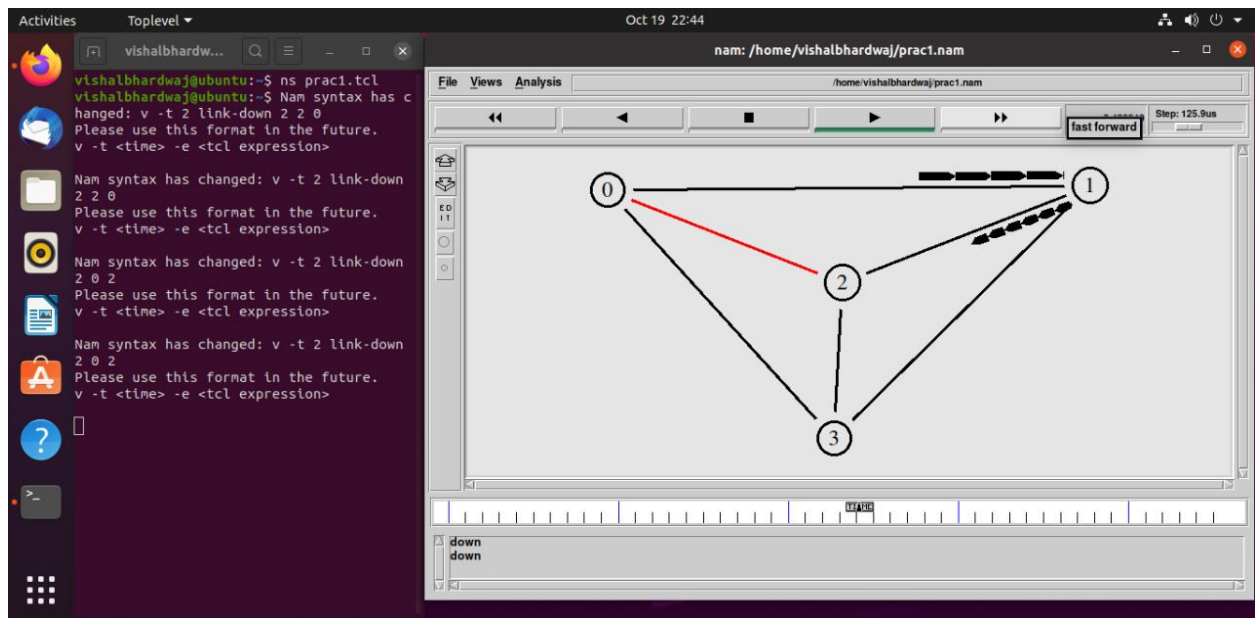
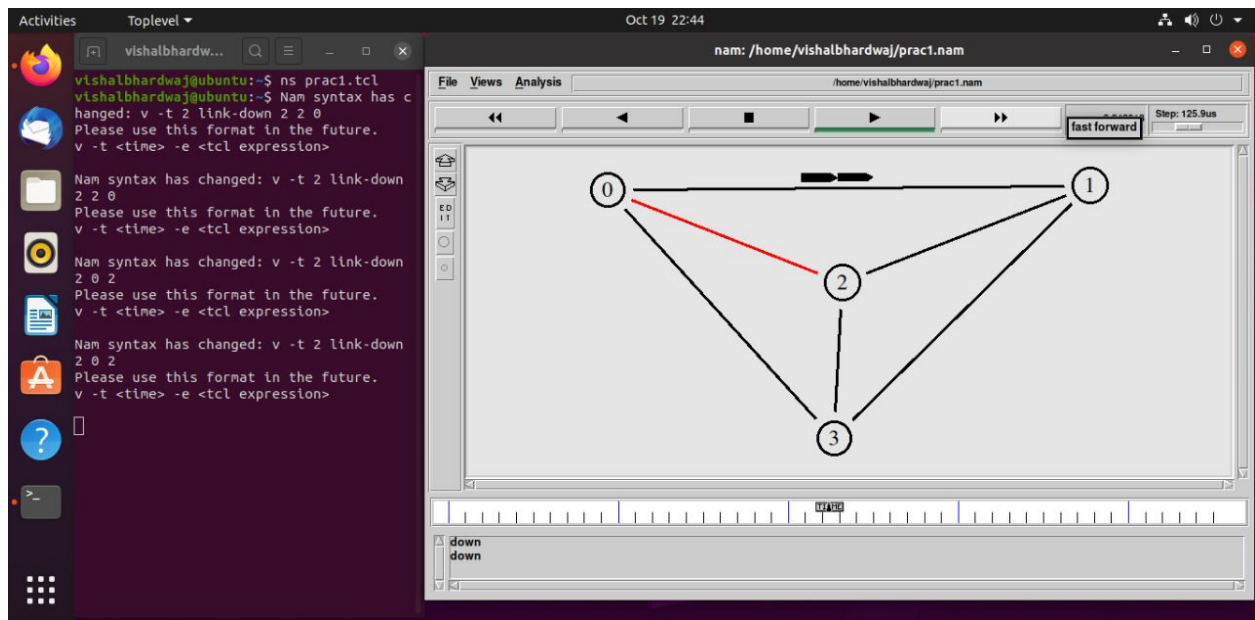
```
$ns duplex-link-op $n0 $n1 orient right
$ns duplex-link-op $n1 $n2 orient down
$ns duplex-link-op $n2 $n3 orient left
$ns duplex-link-op $n3 $n0 orient up
$ns duplex-link-op $n0 $n2 orient right-down
$ns duplex-link-op $n1 $n3 orient left-down
```

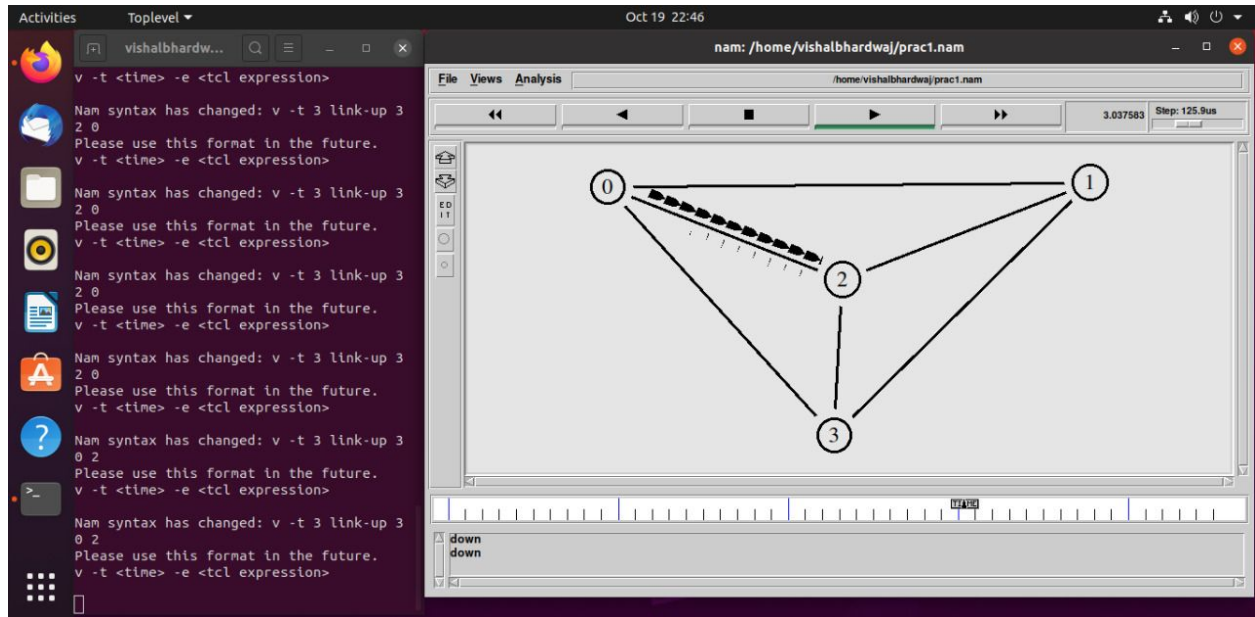
```
set tcp [new Agent/TCP]
$tcp set class_ 2
$ns attach-agent $n0 $tcp
set sink [new Agent/TCPSink]
$ns attach-agent $n2 $sink
$ns connect $tcp $sink
set ftp [new Application/FTP]
$ftp attach-agent $tcp
$ftp set type_ FTP
$ftp set packet_size_ 1000
$ftp set rate_ 1mb
$ns at 1.0 "$ftp start"
$ns rtmodel-at 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
```

Simulation:









Specifications:

- Bandwidth 10Mbps
- Delay 10ms
- Source : n_0
- Destination: n_2
- Queue: Droptail
- Traffic Type: FTP
- Connection :TCP