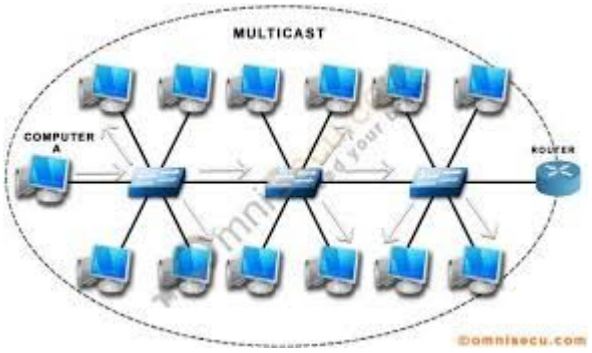
Name: Christine Polly **Roll no: 17**

Experiment No 12

Title :
Demonstration of Multicast on ns2

Theory :
DESCRIPTION:

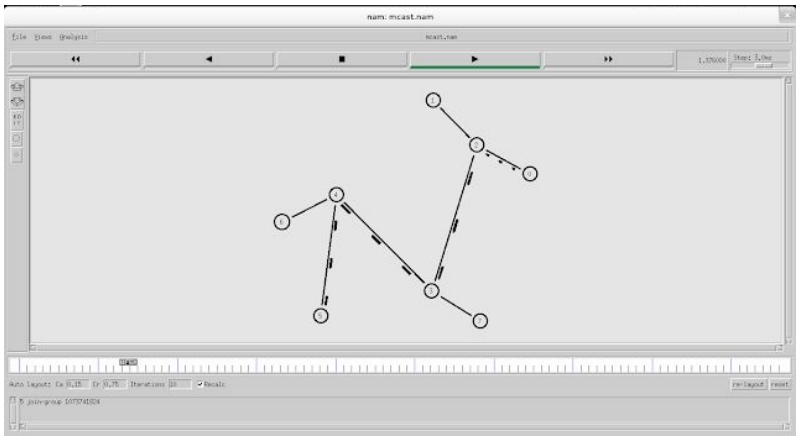
The multicast distribution tree is computed in the ns2 simulation using tcl script to implement the multicast routing strategy. The multicast routing protocols can be run at every node in the network. All nodes are attached to the multicast protocol agents. The nodes can join and leave the multicast group using join-group{} and leave-group{}, procedures respectively. These procedures involve two arguments. The first argument identifies the corresponding node while the second argument represents the group address. Sender transfers data to all the nodes in the multicast group. The code below implements the multicast routing.



Procedure/ Algorithm :
CODE:

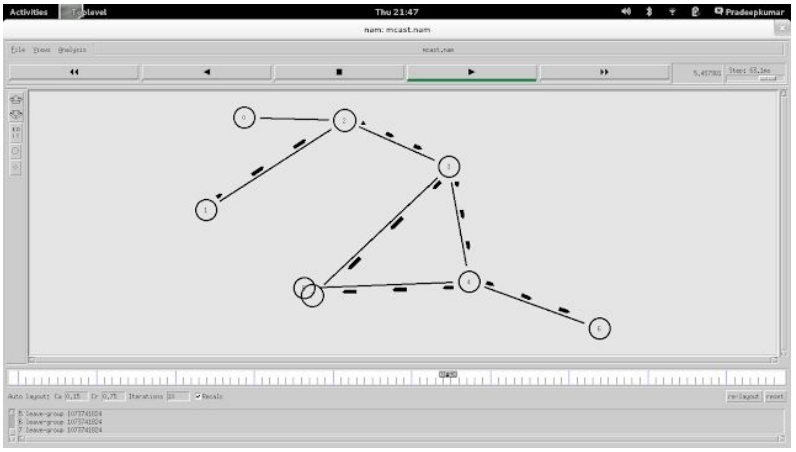
Centralized Multicast routing in ns2

```
#This example is to demonstrate the multicast routing protocol.
set ns [new Simulator -multicast on]
#Turn on Tracing
set tf [open output.tr w]
$ns trace-all $tf

# Turn on nam Tracing
set fd [open mcast.nam w]
$ns namtrace-all $fd

# Create nodes
set n0 [$ns node]
set n1 [$ns node]
set n2 [$ns node]
set n3 [$ns node]
set n4 [$ns node]
set n5 [$ns node]
set n6 [$ns node]
set n7 [$ns node]

# Create links with DropTail Queues
$ns duplex-link $n0 $n2 1.5Mb 10ms DropTail
```



```
$ns duplex-link $n1 $n2 1.5Mb 10ms DropTail
$ns duplex-link $n2 $n3 1.5Mb 10ms DropTail
$ns duplex-link $n3 $n4 1.5Mb 10ms DropTail
$ns duplex-link $n3 $n7 1.5Mb 10ms DropTail
$ns duplex-link $n4 $n5 1.5Mb 10ms DropTail
$ns duplex-link $n4 $n6 1.5Mb 10ms DropTail

# Routing protocol: say distance vector
#Protocols: CtrMcast, DM, ST, BST
#Dense Mode protocol is supported in this example
set mproto DM
set mrthandle [$ns mrtproto $mproto {}]

# Set two groups with group addresses
set group1 [Node allocaddr]
set group2 [Node allocaddr]

# UDP Transport agent for the traffic source for group1
set udp0 [new Agent/UDP]
$ns attach-agent $n0 $udp0
$udp0 set dst_addr_ $group1
$udp0 set dst_port_ 0
set cbr1 [new Application/Traffic/CBR]
$cbr1 attach-agent $udp0

# Transport agent for the traffic source for group2
set udp1 [new Agent/UDP]
$ns attach-agent $n1 $udp1
$udp1 set dst_addr_ $group2
$udp1 set dst_port_ 0
set cbr2 [new Application/Traffic/CBR]
$cbr2 attach-agent $udp1

# Create receiver to accept the packets
set rcvr1 [new Agent/Null]
$ns attach-agent $n5 $rcvr1
$ns at 1.0 "$n5 join-group $rcvr1 $group1"
set rcvr2 [new Agent/Null]
$ns attach-agent $n6 $rcvr2
$ns at 1.5 "$n6 join-group $rcvr2 $group1"


set rcvr3 [new Agent/Null]
$ns attach-agent $n7 $rcvr3
$ns at 2.0 "$n7 join-group $rcvr3 $group1"

set rcvr4 [new Agent/Null]
$ns attach-agent $n5 $rcvr1
```

```
$ns at 2.5 "$n5 join-group $rcvr4 $group2"

set rcvr5 [new Agent/Null]
$ns attach-agent $n6 $rcvr2
$ns at 3.0 "$n6 join-group $rcvr5 $group2"

set rcvr6 [new Agent/Null]
$ns attach-agent $n7 $rcvr3

#The nodes are leaving the group at specified times
$ns at 3.5 "$n7 join-group $rcvr6 $group2"
$ns at 4.0 "$n5 leave-group $rcvr1 $group1"
$ns at 4.5 "$n6 leave-group $rcvr2 $group1"
$ns at 5.0 "$n7 leave-group $rcvr3 $group1"
$ns at 5.5 "$n5 leave-group $rcvr4 $group2"
$ns at 6.0 "$n6 leave-group $rcvr5 $group2"
$ns at 6.5 "$n7 leave-group $rcvr6 $group2"

# Schedule events

$ns at 0.5 "$cbr1 start"
$ns at 9.5 "$cbr1 stop"
$ns at 0.5 "$cbr2 start"
$ns at 9.5 "$cbr2 stop"

#post-processing

$ns at 10.0 "finish"
proc finish {} {
  global ns tf
   $ns flush-trace

   close $tf
   exec nam mcast.nam &
   exit 0
}
$ns set-animation-rate 3.0ms
$ns run
```



The nodes are joining the group before the packet transfer and leaving the group at a specified time.