Department of Information Technology

Academic Term: Jan-May 2021

Class **:** T.E IT Sem -VI
Subject : Sensor Network Lab (ITL604)

Practical No:	**6**
Title:	Implementation of Mobile Network (MANET) using Network Simulator (NS2):
Date of Performance:	
Date of Submission:	
Roll No:	8669
Name of the Student:	Mareena Mark Fernandes

Evaluation:

Sr. No	Rubric	Grade
1	**On time Completion & Submission (2)**	
2	**Output (3)**	
3	**Code Optimization (3)**	
4	**Knowledge of the topic (2)**	
5	**Total (10)**	

Signature of the Teacher **:**

PRACTICAL - 5

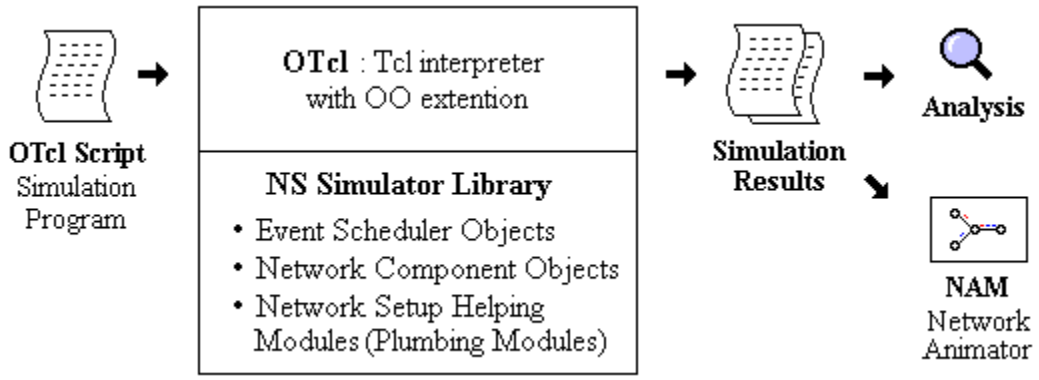
Title: Implementation of Mobile Network using Network Simulator (NS2): Create a Mobile Ad hoc network

Objective: To study Routing in MANET

Pre-Requisite: Basic knowledge of wireless networking

Description:

NS (version 2) is an object-oriented, discrete event driven network simulator developed at UC Berkely written in C++ and OTcl. NS is primarily useful for simulating local and wide area networks. It implements network protocols such as TCP and UPD, traffic source behavior such as FTP, Telnet, Web, CBR and VBR, router queue management mechanism such as Drop Tail, RED and CBQ, routing algorithms such as Dijkstra, and more. NS also implements multicasting and some of the MAC layer protocols for LAN simulations.



Program description:

Each agent keep track of what messages it has seen and only forwards those which it has seen and only forwards those which it hasn't seen before. Each message is of the form "ID:DATA" where ID is some arbitrary message identifier and DATA is the payload. In order to reduce memory usage , the agent store only the message ID.

Steps:

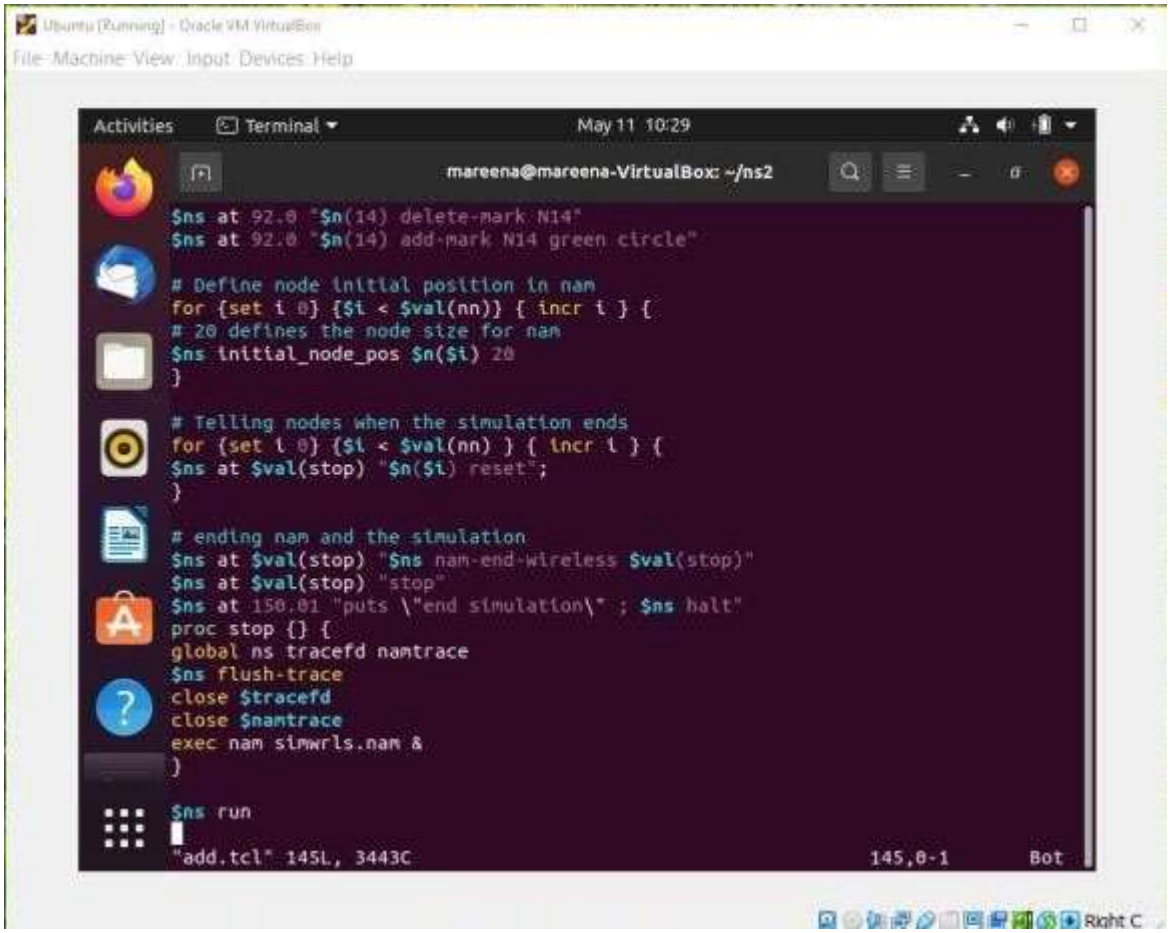
1. Set the following configuration for each node's interface
   - Type of channel - WirelessChannel
   - Type of propagation – TwoRayGround
   - Physical Layer – Wireless
   - Mac Layer – MAC 802.11
   - Type of Queue – DropTail/PriQueue
   - LinkLayer – LL
   - Type of Antenna –OmniAntenna
   - Maximum Packet in Queue - 50
2. Open Trace file in write mode
3. Open NAM file in write mode.

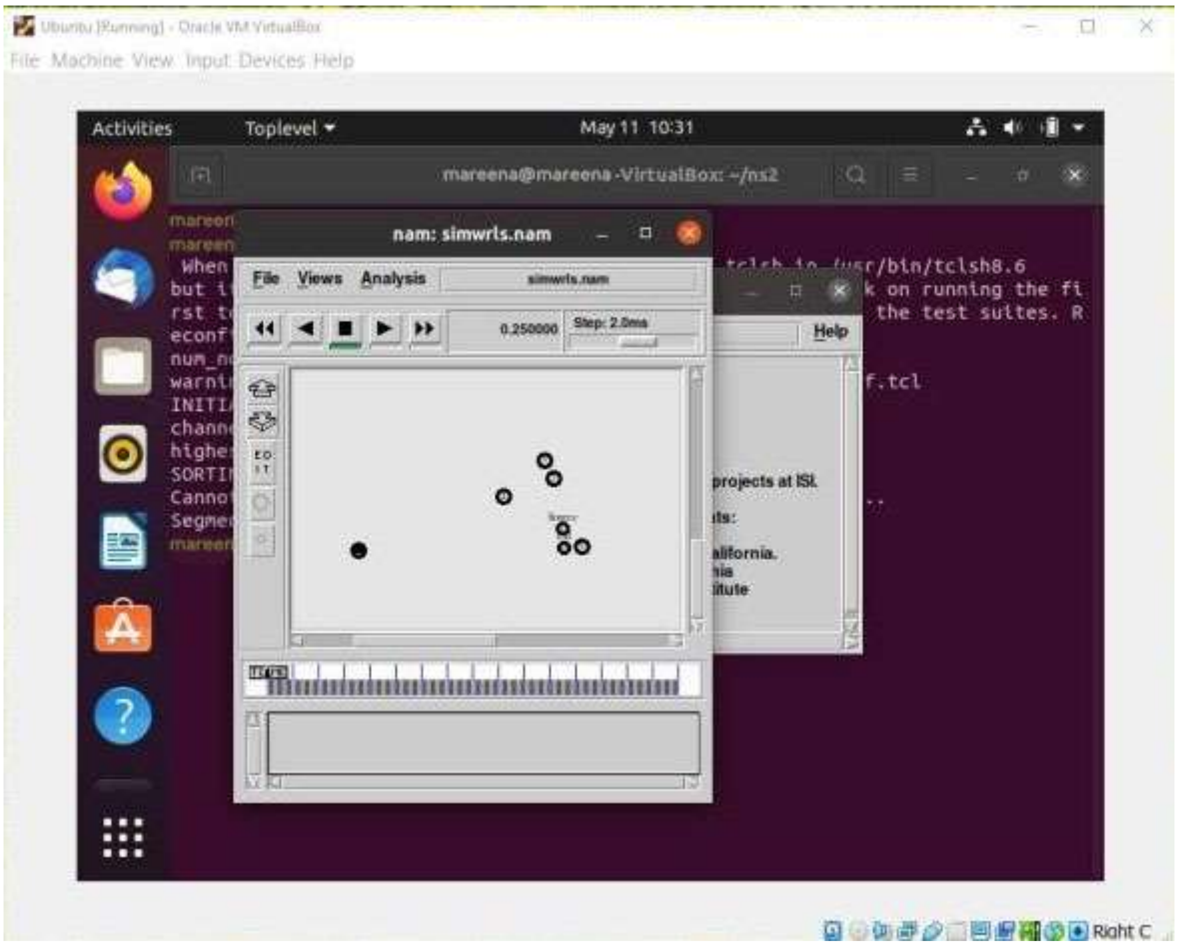
4. Create a topology containing 6 groups each having 4 nodes. Use FlatGrid topology
5. Configure each node using the configuration set in step 1.
6. Create a simple MessagePassing/Flooding agent
7. Create Receive procedure that receives each packet and maintain list of unseen messages
8. Create send procedure that broadcasts message.
9. Create MessagePassing/Flooding agent and attach it with every node.
10. Set up some events.
11. Write finish procedure.

Conclusion: Mobile networks using NS2 has been studied and implemented successfully.

Post Lab Questions:

1. Describe your observations about output.
2. Explain the working of DSDV protocol.


Ubuntu [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities
Terminal
May 11 10:29
mareena@mareena-VirtualBox: ~/ns2
$ns at 92.0 "$n(14) delete-mark N14"
$ns at 92.0 "$n(14) add-mark N14 green circle"
# Define node initial position in nam
for {set i 0} {$i < $val(nn)} { incr i } {
# 20 defines the node size for nam
$ns initial_node_pos $n($i) 20
}
# Telling nodes when the simulation ends
for {set i 0} {$i < $val(nn) } { incr i } {
$ns at $val(stop) "$n($i) reset";
}
# ending nam and the simulation
$ns at $val(stop) "$ns nam-end-wireless $val(stop)"
$ns at $val(stop) "stop"
$ns at 150.01 "puts \"end simulation\" ; $ns halt"
proc stop {} {
global ns tracefd namtrace
$ns flush-trace
close $tracefd
close $namtrace
exec nam simwrls.nam &
}
$ns run
"add.tcl" 145L, 3443C
145,0-1
Bot
Right C


Ubuntu [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities
Toplevel
May 11 10:31
mareena@mareena-VirtualBox: ~/ns2
nam: simwrls.nam
File Views Analysis
simwrls.nam
0.250000
Step: 2.0ms
Help

Post lab

1. Describe your observations about output.

Ans:

At the beginning of the simulation, the layout causes there to be two groups with one group having node 6 as center highlighted in yellow and node 10 being out of range from node 6 is in another group. Once the ftp packet transfer starts, we see the mobile nodes moving around to reach their set destinations. The arrows from source node 3 to sink node 4 represent the data being transferred. As the nodes move around, we see as nodes move out of range of each other they switch groups represented by changing highlights on certain nodes. When source node 3 and sink node 4 are far out of range of each other we see packets are dropped since they cannot reach now without hopping through some other nodes. In DSDV routing protocol, each node keeps updating their current location on the plane along with which other nodes can be reached without hopping multiple routers on regular intervals to the GOD object. Once the GOD table is updated, source node 3 starts redirecting packets through hop node 9 to direct packets to sink node 4 which continues until the end of simulation.

2. Explain the working of DSDV protocol.

Ans:

- Destination Sequenced Distance Vector (DSDV) is a hop-by-hop vector routing protocol requiring each node to periodically broadcast routing updates.
- This is a table-driven algorithm based on modifications made to the Bellman-Ford routing mechanism. Each node in the network maintains a routing table that has entries for each of the destinations in the network and the number of hops required to reach each of them. Each entry has a sequence number associated with it that helps in identifying stale entries. This mechanism allows the protocol to avoid the formation of routing loops. Each node periodically sends updates tagged throughout the network with a monotonically increasing even sequence number to advertise its location.
- New route broadcasts contain the address of the destination, the number of hops to reach the destination, the sequence number of the information received regarding the destination, as well as a new sequence number unique to the broadcast.
- The route labeled with the most recent sequence number is always used. When the neighbors of the transmitting node receive this update, they recognize that they are one hop away from the source node and include this information in
- their distance vectors. Every node stores the “next routing hop” for every reachable destination in their routing table.
- The route used is the one with the highest sequence number i.e., the most recent one. When a neighbor B of A finds out that A is no longer reachable, it advertises the route to A with an infinite metric and a sequence number one greater than the latest sequence number for the route forcing any nodes with B on the path to A, to reset their routing tables.