MALAVIYA NATIONAL INSTITUTE OF TECHNOLOGY JAIPUR



WIRELESS SENSOR NETWORKS LAB

Submitted By:

Your Name

ID

**ASSIGNMENT 1**

**Objective:** To study about Emulators and Simulators with specific reference to Network Simulator NS-3 and then answer the following questions

**Question 1.** **What is NS-3?**

NS-3 is a discrete-event network simulator targeted primarily for research and educational use. It has been developed to provide an open, extensible network simulation platform for networking research and education. It provides models of how packet data networks work and perform, and provides a simulation engine for users to conduct simulation experiments. Available model set in NS-3 focus on modeling how Internet protocols and networks work, but it is not limited to Internet system; several users use NS-3 to model non-Internet based systems.

**Question 2.** **What is the difference between simulation and emulation? Is simulation or emulation done by NS-3?**

**Emulation: For usage as a substitute**

* Emulation is the process of mimicking the outwardly observable behavior to match an existing target. The internal state of the emulation mechanism does not have to accurately reflect the internal state of the target which it is emulating.
* A Network emulator is used to test the performance of a real network. These can also be used for quality assurance , proof of concept or troubleshooting
* An emulator may use different set of protocols for mimicking the thing being emulated, but the outcome is always same as the original object.

**Simulation: For Analysis and Study**

* Simulation involves modeling the underlying state of the target. The end result off a good simulation is that the simulation model will emulate the target which it is simulating.
* Simulator mimics the activity of something that it is simulating. It appears to be the same as the thing being simulated.

There is a difference in focus. Emulators focus on recreating the behavior of a system, with no regard for how the system functions internally. Simulators focus on modeling the components of a system. You use an emulator when you care mostly about what a system does and a simulator when you care about how it does it.

So as NS-3 models the behavior of a computer network system, it is clearly a simulator and performs Network Simulation.

**Question 3.** **What Does NS-3 Provide?**

**Features of NS-3:**

* Network Simulator ns3 is a Discrete Event (DE) simulator and does provide modelling computer or other networks to analyze their working
* Open source, hence flexibility to user to modify it as required
* developed to provide an open, extensible network simulation platform, for networking research and education
* Highly controlled, reproducible environment to learn about how networks work and to custom tune the network parameters and protocols
* designed as a set of libraries that can be combined together and also with other external software libraries
* primarily used on Linux or macOS systems, although support exists for BSD systems and also for Windows frameworks
* Provides visualization capabilities to analyze statistics and watch network and application events
* Can run simulation in real time over network
* Written entirely in C++, with optional Python bindings , code in either C++ or Python
* Provides logging facility for debugging
* Provides tracing facility
* Supports interoperability and has good memory management

**Conclusion:** We have successfully understood the basics about Network Simulators and Emulators and also read about NS-3 and its features.

**ASSIGNMENT 2**

**Objective:** Design and configure a simple network model. Collect Statistics and Analyze network performance.

**Tasks Performed:** Report the following-

1. Network Topology
2. Scenario (Interaction Between Input and Output)
3. Steps for Topology Creation and Configuration
4. Network Parameters ( Simulation Time , Frames Generated , MAC Protocol Used etc.)
5. Network Performance Metrics ( Throughput, Frames dropped etc. )

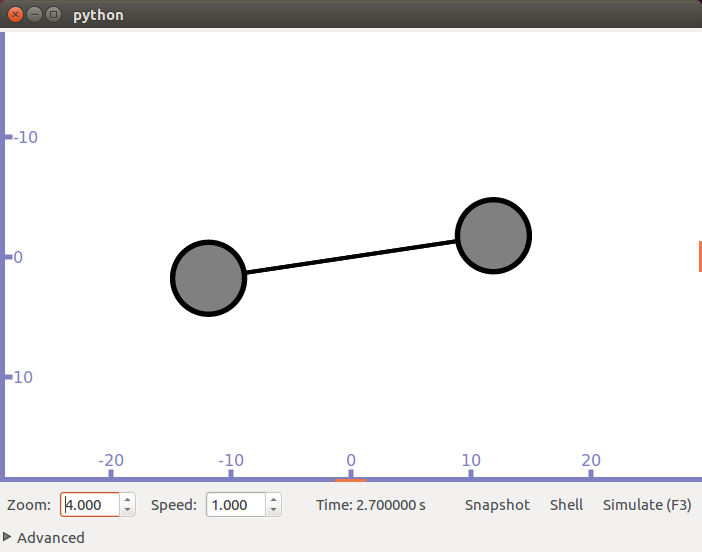
**Steps Involved:**

1. **Network Topology**

Here is an example of a simple point to point topology where two devices/nodes create a network to send or exchange data.

P1 ...............................P2

P2P Link



1. **Scenario**

Two Devices are in a network where one device act as a client and the other as server. The client application is sending data packets to the server which are acknowledged by the server and he client receives acknowledgement response.

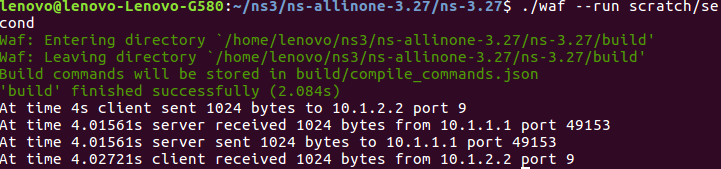
1. **Steps for Topology Creation and Configuration**
2. Import the necessary modules and header files
3. Create a NodeContainer object and then use its Create method to create 2 nodes
4. Create a PointToPointHelper object and set its DeviceAttribute(“Channel Rate”) and also set its ChannelAttribute(“Delay”)
5. Create NetDeviceContainer object install the p2p nodes on it
6. Create an InternetStackHelper and install nodes on stack
7. Create Ipv4AddressHelper and assign addresses to those nodes
8. Using UdpEchoServerHelper create server app on one node
9. Using UdpEchoClientHelper install client on the other node
10. Start and stop time for the client and server apps as required
11. Call Simulator Run and Destroy methods

**Detailed Simulation Parameters:**

* Nodes - 2
* Data Rate - 5 Mbps
* Channel Delay - 2msec
* Base Address - 10.1.1.0,255.255.255.0
* Simulation Time - 10 sec
* Server Start Time - 1 sec
* Server Stop Time - 10 sec
* Client Start Time - 2 sec
* Client Stop Time - 10 sec
* Port Used - 9 , 49153
* Protocol Used - UDP
* Max Datagram Size - 65507 Bytes
* Max No. Of Packets - Variable
* Interval To Send Packets - Variable(in Seconds)
* Packet Size - Variable (In Bytes)
* The Setup is IPv4.

**Results:**

**Output --**



**Network Performance Metrics :**

Here 2 msec delay is for a packet = 0.002 sec but total delay is (2.00369 - 2) sec = 0.00369 sec

Travel/Propagation delay = 0.00169 sec

So , Throughput = (1024\*8)/0.00369\*1024\*1024 = 2.117 Mbps

In this example no packets are dropped.

But , if either client or server stops before entire communication takes place , the packets are dropped or if time interval between packets is more , then frames are dropped.

**Conclusion:** We have successfully configured a simple network topology and analyzed it and also collected various statistics.

**ASSIGNMENT 3**

**Objective:** Create 2 WiFi Networks with separate access points , the access points being in a point to point connection . Send data packets between devices in these networks and also create trace file and analyze it via wireshark or tcpdump

**Tasks Performed:** Report the following-

1. Network Topology
2. Scenario (Interaction Between Input and Output)
3. Steps for Topology Creation and Configuration
4. Network Parameters

**Steps Involved:**

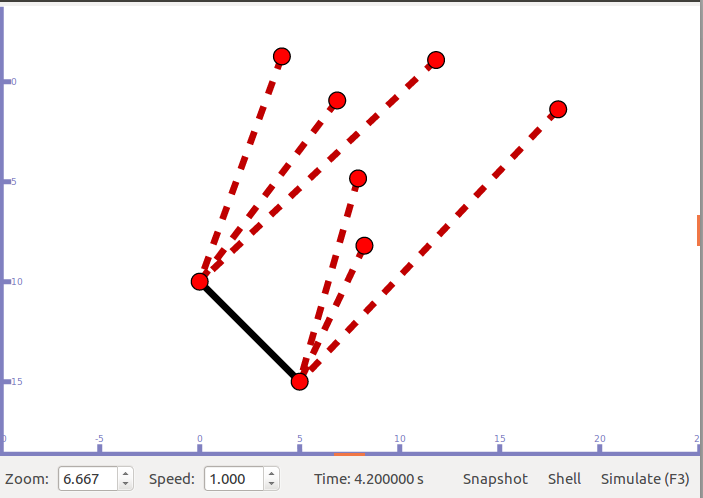
1. **Network Topology**

Here is the topology where two devices/nodes act as Access Points and each of these access points has 3 more station nodes it its range.

Wifi Network 1 Ad Hoc Wifi Network Wifi Network 2

n5 n6 n7 n0 .|......................................................................|. n1 n2 n3 n4

Access Point Access Point



1. **Scenario**

One station device in range of one access point act as server and another station device in range of other access point act as the client. The server and client nodes exchange data packets.

A trace file is also created of this transfer.

1. **Steps for Topology Creation and Configuration**

In this, we create 2 separate wifi networks as follows :

1. Create 2 P2P nodes and define data rate and channel delay using PoinToPoint Helper.
2. Install the P2P nodes . These nodes acts as access points for our 2 wifi networks.
3. Create wifi nodes for both networks.
4. Set wifi Access point nodes from P2P nodes for both networks.
5. Create YansWifiChannelHelper and YansWiFiPhyHelper for both the networks.
6. Create WifiMacHelper for both WiFi networks , assign SSID to them and set MAC type.
7. Install both mobile nodes as well as access points.
8. Set mobility of the Wifi networks , min-max coordinates and bounds.
9. Create InternetStackHelper stack and install all nodes.
10. Create an IPv4 AddressHelper to assign address to each of the trace network nodes simultaneously creating variables for P2P and WiFi interfaces.

Now we send data packet from client in one wifi network to server on another wifi network :

1. Create UDPEchoServerHelper and Install any one wifi node as server.
2. Create UDPEchoClientHelper object and assign address of any node in another wifi network.
3. Set address of server in client object to connect to server.
4. Set attributes from client device like MaxPacket , Interval , Packet Size etc.

To Enable Tracing :

1. EnablePacpall for point to point nodes.
2. EnablePcap for physical YansChannelHelper object

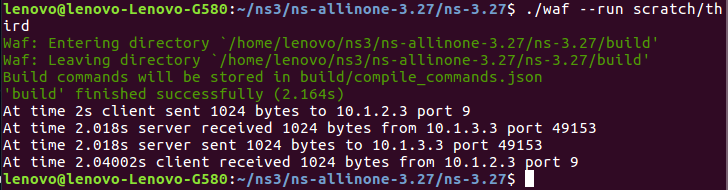
After simulating this model , we will get seperate Pacp files for every node for which Pcap is enabled.

**Detailed Simulation Parameters:**

* Nodes – 2 APs with 3 STA nodes for each
* Access Points - 2
* Data Rate - 5 Mbps
* Channel Delay - 2msec
* Base Address - 10.1.1.0,255.255.255.0
* Simulation Time - as user wishes (variable)
* Server Start Time - 1 sec
* Server Stop Time - 10 sec
* Client Start Time - 2 sec
* Client Stop Time - 10 sec
* Port Used - 9 , 49153
* Protocol Used - UDP
* Max Datagram Size - 65507 Bytes
* Max No. Of Packets - Variable
* Interval To Send Packets - Variable(in Seconds)
* Packet Size - Variable (In Bytes)
* The Setup is IPv4.

**Results:**

**Output -**



**Conclusion:** We have successfully configured the required network topology and analyzed it and also collected various statistics and a saved a pcap file for packet tracing.

**ASSIGNMENT 4**

**Objective:** Create a scenario for hidden & exposed terminal problem. Find out the packet drop in both of the scenarios. Also propose a solution (RTC/CTS mechanism)

**Tasks Performed:** Report the following-

1. Network Topology
2. Scenario (Interaction Between Input and Output) ( with and without RTS/CTS mechanism )
3. Steps for Topology Creation and Configuration
4. Show difference in throughput and packet drop in both the cases
5. Find out throughput less than the without RTS/CTS mechanism
6. Find situation where throughput will be higher even with RTC/CTS
7. Calculate overhead caused by RTS/CTS packets
8. Network Parameters

**Steps Involved:**

1. **Network Topology**

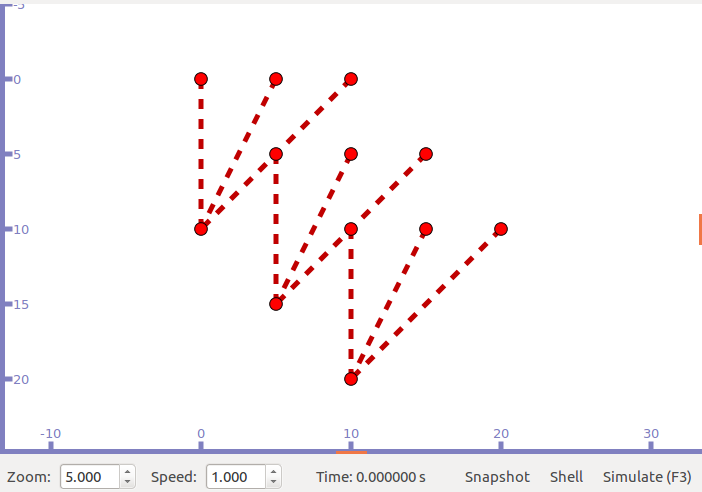
Here is the topology where three devices/nodes act as Access Points and each of these access points has 3 more station nodes it its range

Access Point 1 Ad Hoc 1 Access Point 2 Ad Hoc 2 Access Point 3

n0 ............................................................. n4 ....................................................... n8

n3 n2 n1 n5 n6 n7 n9 n10 n11

Wifi Network 1 Wifi Network 2 Wifi Network 3



1. **Scenario 1 (Hidden Terminal)**

Hidden Terminal problem occurs when a station node is not in range of the sender of data but is in the range of the receiver. When two such unaware nodes start to transmit data to the common receiver data collision occurs. To avoid this problem one can use RTS/CTS control packets.

So we create a scenario where in the above displayed topology, node n1 (client) from wifi network1 and node n9 (client) from wifi network3 send data simultaneously to node n5 (server) in wifi network2.

1. **Steps to create topology and send data**

In this, we create 3 separate wifi networks as follows :

1. Create 3 nodes that will act as access points (using NodeContainer).
2. Create MobilityHelper and set PositionAllocator and mobility model as required ( Here gridposition Allocator , Constant position mobility model).
3. Create 3 WifiStaNodes with 3 nodes in each.
4. Set previously created 3 nodes as AccessPoints Nodes.
5. Create YansWiFiChannelHelper , YansWifiPhyHelper for all 3 seperate networks and set channel to phy helper.
6. Set SSID , RemoteStationManager , Mactype , Probing etc for 3 networks.
7. Install WiFi Phy,Mac and StaNodes to StaDevices net device container.
8. Create 3 ApWiFimac and install on access point devices Net Devices Container.
9. Set previously created mobility helpers to StaNodes and ApNodes.
10. Now create 2 seprate Adhoc networks with AP1-AP2 & AP2-AP3.
11. Create Seprate YansWiFiChannel , WiFiHelper , YansphysicalHelper and WiFimacHelper for these 2 adhoc networks.
12. Set standard (WiFi-Phy-Standard-802.11b).Set RemoteStationManager and mobility model to these 2 new networks.
13. Create InternetStackHelper stack and install all these nodes.
14. Set address & assign them to nodes for each of the network created.
15. Now create an OnOffHelper on ST1 of AP2 & install client OnOffHelper Apps on ST1 of AP1 & ST1 of AP2 both.
16. Set different attributes like Data Rate , Time , Packet Size etc.
17. Create a FlowMonitor to get the FlowStates.
18. Extract & display the stats at the end.
19. Enable the log components for the on-off Helper.
20. Set simulator::Run() & simulator::Destroy() ;.

Adding RTS/CTS Mechanism

1. RTS/CTS can be enabled on setting RTSCTSThreshold to be greater than the packet size
2. To enable Add :

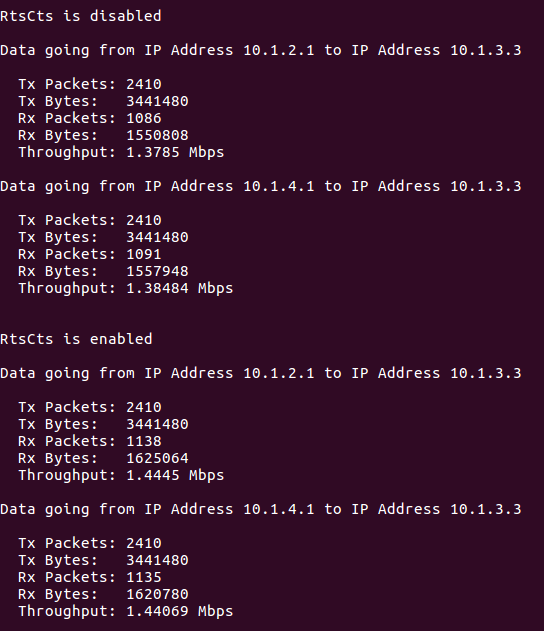
Config :: setdefault ("ns3::WiFiRemoteStationManager::RTSCTSThreshold",CTSThrvalue)

**Detailed Simulation Parameters:**

* RTSCTSThr Value - 2200/10
* Probing Used - Active Probing
* ConstantPosition Mobility model used for ApNodes.
* RandomWalk2dMobility model used for StaNodes.
* Propogation Loss - Fixed for each pair of Nodes & doesnot depend on their actual positions.
* For Data Mode DSSS rate - 2Mbps
* for Control Mode DSSS rate - 1 Mbps
* Packet Size - 1400 bytes

**Results:**

**Output -** Comparison in RTS/CTS enabled & disabled



**RTS/CTS Disabled** **RTS/CTS Enabled**

flow1 flow2 flow1 flow2

**Tx Packets** 2410 2411 2410 2411

**Tx Bytes** 3441480 3442908 3441480 3442908

**Rx Packets** 245 233 425 562

**Rx Bytes** 349860 332724 606900 802536

**Throughput(Mbps)** 0.310987 0.295755 0.539467 0.713365

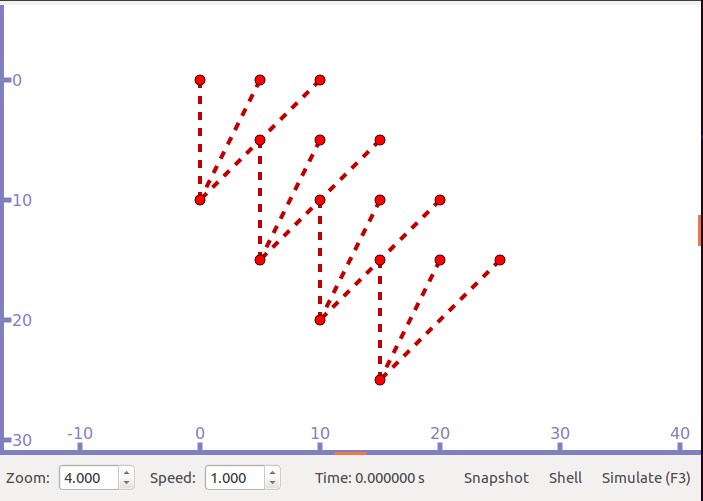
**4. Scenario #2 (Exposed Terminal)**

Exposed Terminal problem occurs when a node is communicating with a node and due to this some other node is unable to communicate with some different node because the former node is in range of one of the nodes which are communicating. The node becomes exposed to its neighbour node. To avoid this problem one can use RTS/CTS control packets.

So we create a scenario where = , a node (client) from wifi network2 send data to a node (server) in wifi network1, and a node (client) from wifi network3 wants to send data to a node (server) in wifi network4.

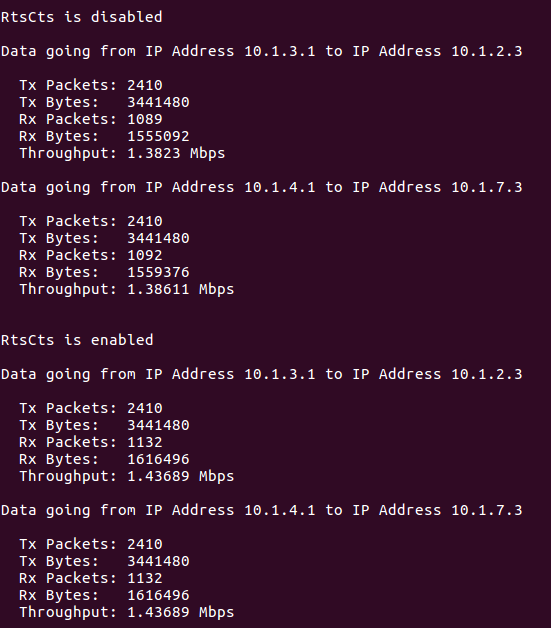
Scenario can be generated following the similar steps as that for hidden terminal problem.

AP1----------AP2------------AP3------------AP4



Using similar simulation parameters as that for hidden terminal scenario.

**Results:**



**Observations:**

* Throughput is less without RTS/CTS mechanism. This is so because with RTS/CTS there are excessive collisions as the stations remain hidden and cannot sense if the channel is free or not.
* Throughput can be lower even when RTS/CTS is applied if there are successive continuous collisions among these RTS/CTS control packets.
* The overhead caused by RTS/CTS packets amounts to the No. of RTS , CTS packets exchanged and the size of these packets.
* Minimum Overhead = (RTS+CTS) \* No. of Packets to be transferred assuming no RTS , CTS collisions.

**Conclusion:** We have successfully configured the required network topology and analyzed it and also collected various statistics and a saved a pcap file for packet tracing.

**ASSIGNMENT 5**

**Objective:** Create a Wireless Sensor Network of 10 Nodes using IEEE 802.15.4 protocol where 9 out of these nodes are sending hello message (ping) to the 10 nodes

**Tasks Performed:** Report the following-

1. Network Topology
2. Scenario (Interaction Between Input and Output)
3. Steps for Topology Creation and Configuration
4. Network Parameters

**Steps Involved:**

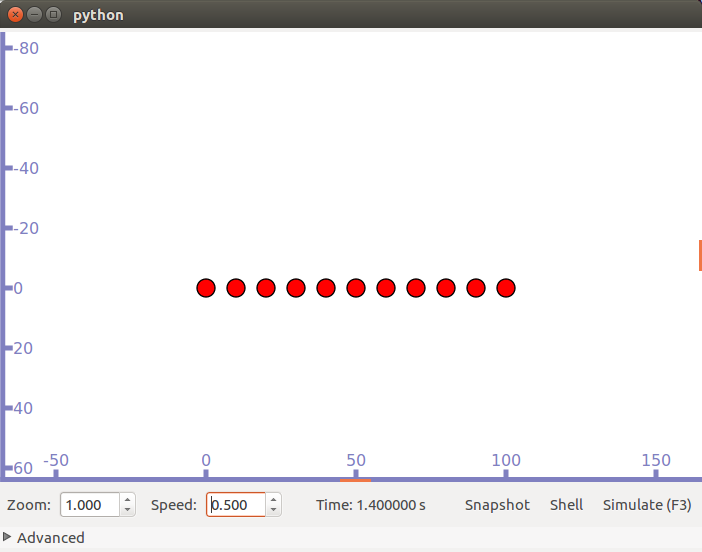
1. **Network Topology**

Here is the topology where ten devices/nodes are in IEEE 802.15.4 Wireless Network

n0 n1 n2 n3 n4 n5 n6 n7 n8 n9

-------------|--------|---------|-------|---------|----------|--------|--------|---------|------|-----------

WSN ( 802.15.4 )



1. **Scenario**

One station n0 acts as the server and all the other nodes will act as clients and will send hello message to node n0

1. **Steps for Topology Creation and Configuration**

In this, we create 2 separate wifi networks as follows :

1. Use NodeContainer object to create 10 nodes
2. Set mobility model for the nodes
3. Create LrWpan Helper object and Install nodes to it
4. Associate a PAN for the network
5. Create Internet Stack Helper , set base address and install nodes on this stack
6. Create 9 Ping6Helper and 9 Application Containers for each of these nodes
7. Set local node and remote node for each of these nodes
8. Also set packet size , max packet count and packet interval
9. Set time to start and stop these apps
10. Call Simulator Run and Destroy

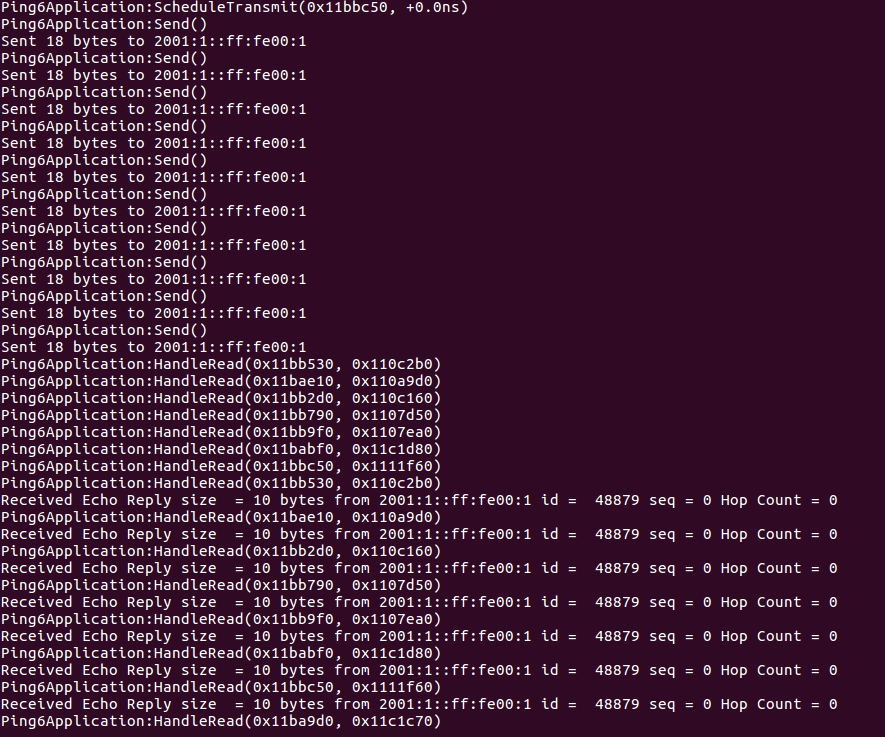
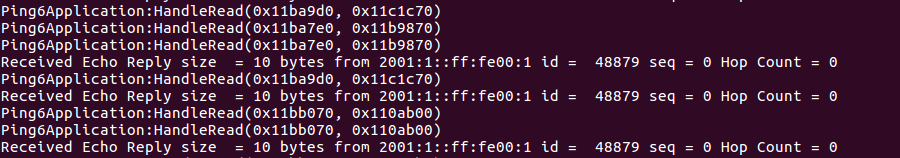
**Detailed Simulation Parameters:**

* Nodes – 10
* Base Address – Ipv6 2001:1::
* Simulation Time - as user wishes (variable)
* Max No. Of Packets - 5
* Interval To Send Packets - 1(in Seconds)
* Packet Size - 10 (In Bytes)
* The Setup is IPv6

**Results:**

**Output -**





**Conclusion:** We have successfully configured the required network topology and analyzed it and also collected various statistics.

**ASSIGNMENT 6**

**Objective:** Use AODV and DSDV for routing in networks and analyze the network and protocol performance.

AODV

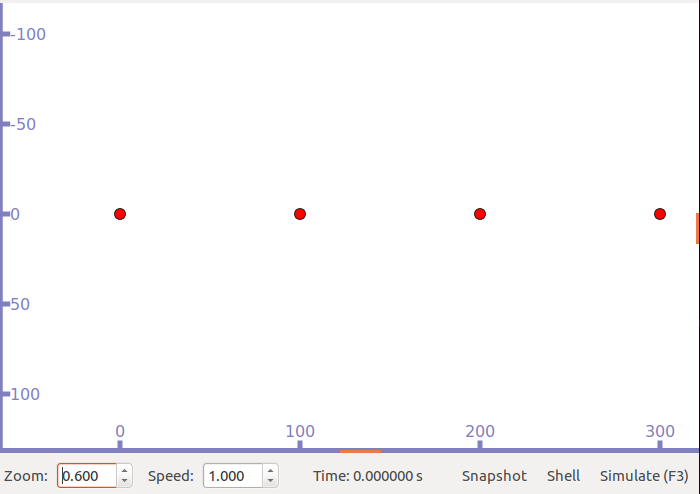
**Tasks Performed:** Report the following-

1. Network Topology
2. Scenario (Interaction Between Input and Output)
3. Steps for Topology Creation and Configuration
4. Network Parameters

**Steps Involved:**

**1. Network Topology**

Here is the topology where 4 nodes are connected



**2. Scenario**

Four nodes are connected in 1 dimension grid topology and leftmost node wants to send data to rightmost node for which it needs to first find a route.

**3. Steps for Topology Creation and Configuration**

In this, we create network as follows :

1. Use NodeContainer object to create 4 nodes placed 50m apart.
2. Set mobility model for the nodes
3. Create YansWifiChannelHelper and YansWiFiPhyHelper for the network.
4. Create WifiMacHelper for network , assign SSID and set MAC type.
5. Install wifi onto nodes.
6. Create InternetStackHelper stack, set aodv routing helper and install all nodes.
7. Create an IPv4 AddressHelper to assign address to each of the nodes.
8. Create ping application with server at rightmost node and a client at leftmost node.
9. Also set parameters like packet size , max packet count and packet interval.
10. Set time to start and stop the application.
11. Call Simulator Run and Destroy.

**Detailed Simulation Parameters:**

* MAC Protocol: 802.11
* Network simulation area: 300m
* Bandwidth: 6Mbps
* Source Node: Node 1 (10.0.0.1)
* Destination Node: Node 4 (10.0.0.4)
* Data Packet Size: 64 bytes
* Number of packets sent in each session: 10
* Simulation length: 10 seconds

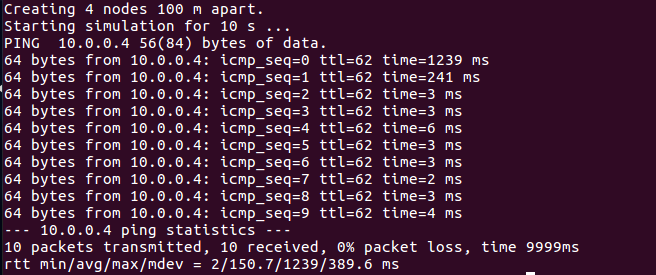
**Results:**

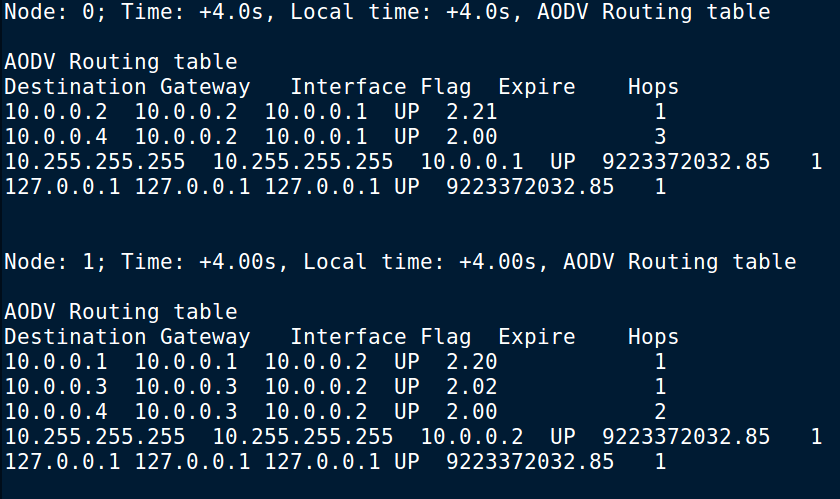
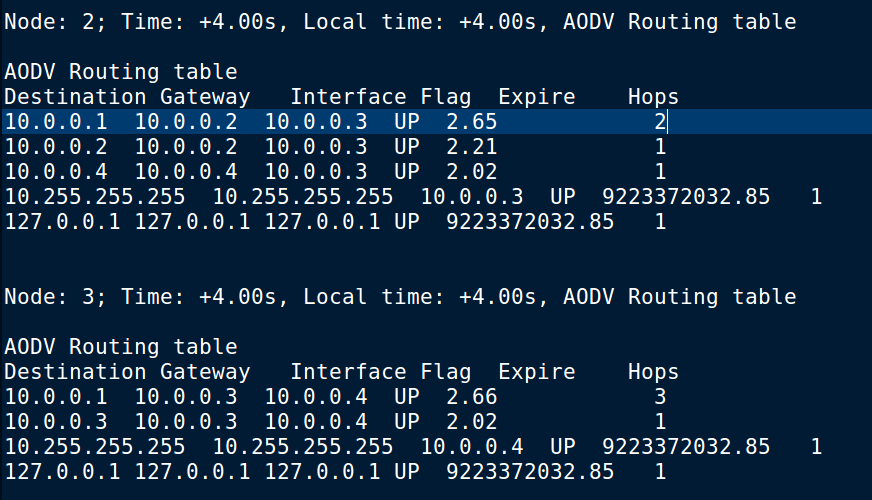
Route Discovery time : 1.217525 s

Packet Loss Performance: 0%

End to end Delay (Round trip times): min/avg/max = 2/150.7/1239 ms

Throughput: 3kbps



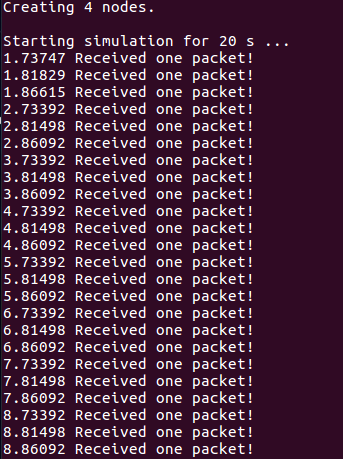


DSDV

Network Topology, scenario and topology creation and configuration is same as that of AODV.

**Detailed Simulation Parameters:**

* MAC Protocol: 802.11
* Bandwidth: 11Mbps
* Source Node: Node 1 (10.0.0.1)
* Destination Node: Node 4 (10.0.0.4)
* Data Packet Size: 1000 bytes
* Number of packets sent in each session: 10
* Simulation length: 20 seconds

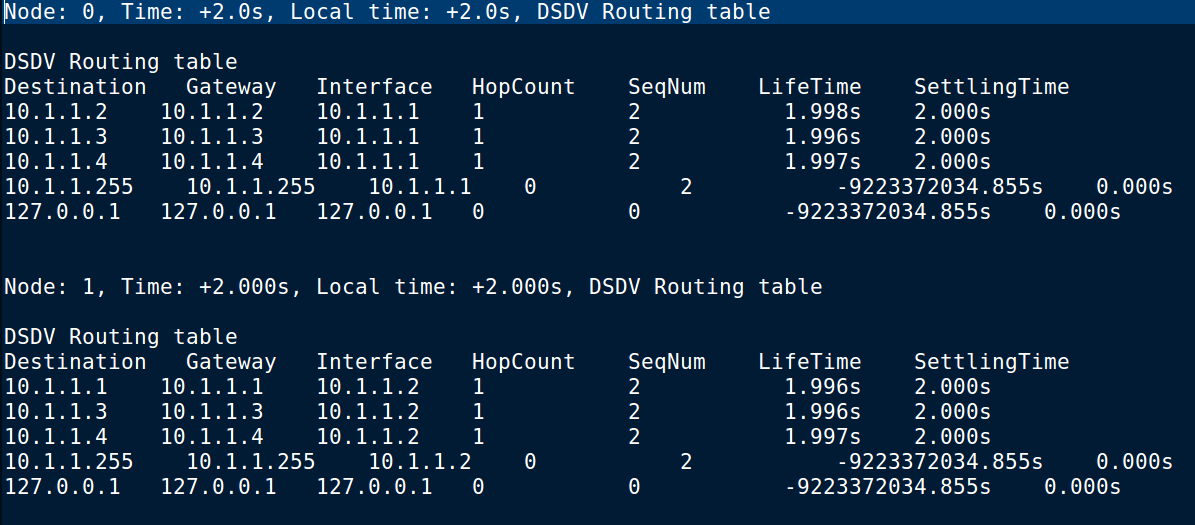


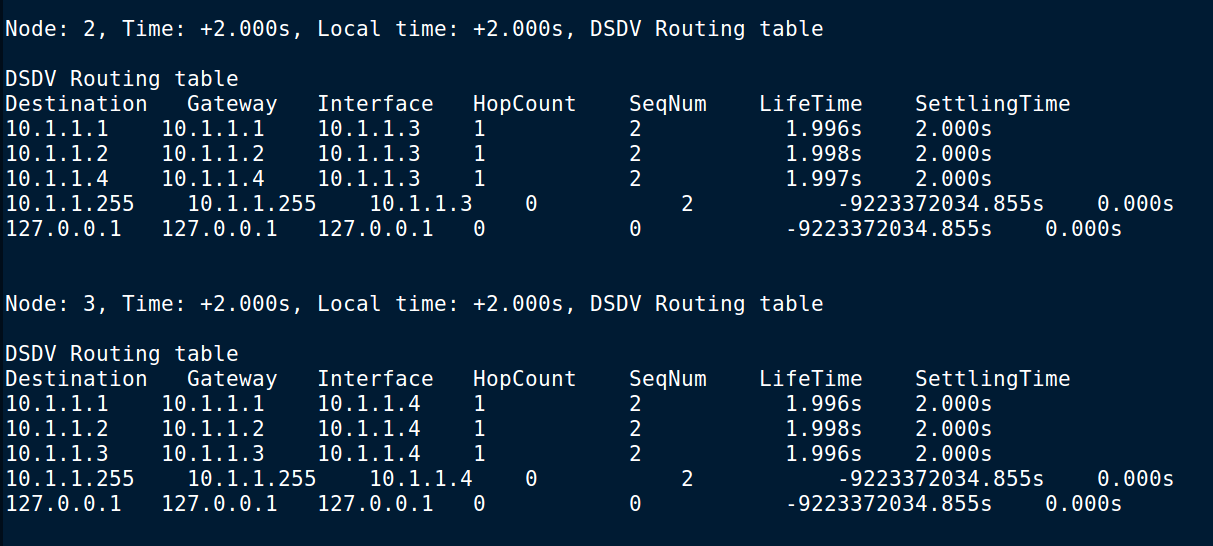
**Results:**

Packet Delivery Ratio : 1

Routing Load: 9

Average End to end Packet Delivery Time: 737.47 ms





**Conclusion:** We have successfully analyzed network performance for AODV and DSDV routing protocols.