NetSim®

Accelerate Network R & D

Mobile Ad hoc Networks (MANETs)

A Network Simulation & Emulation Software

By

TETCOS LLP

The information contained in this document represents the current view of TETCOS LLP on the issues discussed as of the date of publication. Because TETCOS LLP must respond to changing market conditions, it should not be interpreted to be a commitment on the part of TETCOS LLP, and TETCOS LLP cannot guarantee the accuracy of any information presented after the date of publication.

This manual is for informational purposes only.

The publisher has taken care in the preparation of this document but makes no expressed or implied warranty of any kind and assumes no responsibility for errors or omissions. No liability is assumed for incidental or consequential damages in connection with or arising out of the use of the information contained herein.

Warning! DO NOT COPY

Copyright in the whole and every part of this manual belongs to TETCOS LLP and may not be used, sold, transferred, copied or reproduced in whole or in part in any manner or in any media to any person, without the prior written consent of TETCOS LLP. If you use this manual you do so at your own risk and on the understanding that TETCOS LLP shall not be liable for any loss or damage of any kind.

TETCOS LLP may have patents, patent applications, trademarks, copyrights, or other intellectual property rights covering subject matter in this document. Except as expressly provided in any written license agreement from TETCOS LLP, the furnishing of this document does not give you any license to these patents, trademarks, copyrights, or other intellectual property. Unless otherwise noted, the example companies, organizations, products, domain names, e-mail addresses, logos, people, places, and events depicted herein are fictitious, and no association with any real company, organization, product, domain name, email address, logo, person, place, or event is intended or should be inferred.

Rev 15.1 (V), Oct 2026, TETCOS LLP. All rights reserved.

All trademarks are the property of their respective owner.

Contact us at

TETCOS LLP

214, 39th A Cross, 7th Main, 5th Block Jayanagar,

Bangalore - 560 041, Karnataka, INDIA.

Phone: +91 80 26630624

E-Mail: sales@tetcos.com

Visit: www.tetcos.com

Contents

1 Introduction **6**

2 Simulation GUI **7**

2.1 NetSim MANET Network Setup . . . 8

2.1.1 Deployment Architecture . . . 8

2.2 Fast Configuration . . . 9

2.2.1 Device Placement . . . 9

2.3 Create Scenario . . . 12

2.3.1 Wireless Node . . . 12

2.3.2 Bridge Node . . . 12

2.3.3 Link . . . 12

2.4 Set Node, Link and Application Properties . . . 13

2.5 Enable Packet Trace, Event Trace (Optional) . . . 16

2.6 Enable protocol specific logs and plots . . . 16

2.7 GUI Configuration Parameters . . . 17

2.8 Run Simulation . . . 25

3 Model Features **26**

3.1 Ad hoc On Demand Distance Vector Routing . . . 26

3.2 Dynamic Source Routing (DSR) . . . 27

3.2.1 Using Link-Layer Acknowledgements . . . 27

3.2.2 Using Network-Layer Acknowledgements . . . 27

3.2.3 Why do we see continuous RREQ and RREP packets in DSR? . . . 28

3.3 AODV/DSR Metrics . . . 29

3.4 Zone Routing Protocol (ZRP) . . . 29

3.5 Optimized Link State Routing Protocol (OLSR) . . . 31

3.5.1 One Hop neighbor . . . 31

3.5.2 Two Hop neighbor . . . 31

3.5.3 Multipoint Relays (MPR) . . . 31

3.5.4 Topology discovery . . . 32

3.5.5 TC Message Format . . . 32

3.5.6 Core Functionality of OLSR . . . 32

3.5.7 Link Sensing . . . 33

3.5.8 Neighbor detection . . . 33

3.5.9 MPR Selection . . . 33

3.5.10 TC message . . . 34

3.5.11 Route Calculation . . . 34

3.6 Optimized Link State Routing Protocol Version 2 (OLSRv2) . . . 34

3.6.1 Evolution from OLSR Version 1 . . . 34

3.6.2 NHDP – Neighbor Discovery Protocol . . . 35

3.6.3 Message Structure . . . 37

3.6.4 Message Header Fields . . . 37

3.6.5 Address Blocks and TLVs . . . 37

3.6.6 Message Body Structure . . . 37

3.7 OLSRv2 – Protocol . . . 37

3.7.1 Protocol Overview . . . 38

3.7.2 Control Messages . . . 38

3.7.3 Control Message Details . . . 38

3.7.4 Configuring OLSRv2 in NetSim . . . 39

3.8 Energy model in MANETs . . . 42

3.9 Mobility models in NetSim . . . 43

3.9.1 Random Walk Mobility Model . . . 43

3.9.2 Random Waypoint Mobility Model . . . 43

3.9.3 Group Mobility Model . . . 43

3.9.4 Pedestrian Mobility Model . . . 44
3.9.5 File Based Mobility Model . . . 44
3.10 Omni and sector (directional) antennas . . . 45
3.10.1 Limitations . . . 47
3.11 How are packet collisions modelled in NetSim? . . . 47
3.12 How is packet error decided in NetSim . . . 48
3.13 Broadcast in MANETs . . . 48
3.14 Radio measurement log file and plots . . . 49
4 Featured Examples **51**
4.1 AODV Routing . . . 51
4.1.1 Network Settings . . . 52
4.2 AODV Route Discovery and Route Maintenance . . . 54
4.2.1 AODV Expanding Ring Search . . . 54
4.2.2 AODV G/D Flag Scenario . . . 61
4.2.3 AODV RERR Route Maintenance . . . 66
4.3 ZRP-IARP . . . 72
4.3.1 Network Settings . . . 72
4.3.2 Output . . . 73
4.4 ZRP-IERP . . . 73
4.4.1 Network Settings . . . 74
4.4.2 Output . . . 75
4.5 MANET-OLSR . . . 75
4.5.1 Part-1: MPR Table Formation in NetSim . . . 75
4.5.2 Part-2: NDP and TC Messages in OLSR . . . 80
4.6 Comparative analysis of AODV and OLSR in static and mobility scenarios . . . 83
4.6.1 Introduction . . . 83
4.6.2 Simulation Setup . . . 84
4.6.3 Network Configuration . . . 84
4.6.4 Scenario Setup . . . 85
4.6.5 Steps for Control Traffic Overhead Calculation . . . 86
4.6.6 Results and Discussion . . . 89
4.6.7 Impact of Varying Velocity on Routing Protocol Performance . . . 93
4.6.8 Velocity cases . . . 93
4.7 Multiple MANETS . . . 95
4.7.1 Network Settings . . . 96
4.7.2 Output . . . 96
4.8 Energy consumption in MANETs . . . 96
4.8.1 Network Setup . . . 97
4.8.2 Energy Consumption . . . 99
4.8.3 Energy consumption calculations . . . 99
4.8.4 Case#1: Results and discussion . . . 103
4.8.5 Case #2: Results and discussion . . . 103
4.8.6 Case #3: Results and discussion . . . 105
4.9 Application throughput variation with node mobility . . . 106
4.9.1 Network Settings . . . 107
4.9.2 Output . . . 110
4.9.3 Inference . . . 110
4.10 Performance of range based pathloss model in MANETs . . . 111
4.10.1 Introduction . . . 111
4.10.2 Case 1: Range Based Pathloss. Nodes Within Transmission Range. . . . 112
4.10.3 Output and Plot . . . 112
4.10.4 Case 2: Range Based Pathloss. Nodes Beyond Transmission Range. . . . 113
4.10.5 Output and Plot . . . 114

4.10.6 Case 3: Range Based Pathloss with Node Mobility. 114
4.10.7 Output and Plot 115
4.10.8 Case 4: Range Based Pathloss with Multi-hop Communication. 116
4.10.9 Output and Plot 117
4.10.10 Case 5: Range Based Pathloss with Multi-hop Communication and Node Mobility 118
4.10.11 Output and Plot 120
4.10.12 Case 6: Range Based Pathloss. Nodes within range and beyond range. Broadcast Application 121
4.10.13 Case 7: Range Based Pathloss. Transmission Failure due to Interference. 123
4.10.14 Case 8: Range Based Pathloss. Carrier sense (CS) blocking due to neighboring transmissions. 124
4.11 OLSRv2 Multi-cluster MANET 125
4.11.1 Scenario Overview 125
4.11.2 Network Topology – Phase 1 (Baseline) 127
4.11.3 Simulation Phases 129
4.11.4 Features Exercised 133
4.11.5 Traffic Flows 134
4.11.6 Simulation Results – Phase by Phase 135
4.11.7 Running the Example in NetSim 136
4.11.8 Conclusion 137
5 References **138**
6 Latest FAQs **138**

1 Introduction

In NetSim's Internetworks library, wireless nodes connect to an Access Point (AP) while in NetSim's LTE and 5G libraries, the mobile nodes (UEs) connect to base stations (eNBs, gNBs). There is no such association between the wireless stations and any fixed infrastructure in MANETs.

A Mobile Ad hoc Network (MANET) is an autonomous system of mobile nodes. In such networks, information transport services are built over a set of arbitrarily located nodes, which are possibly mobile. Every node behaves both like a mobile host and as a wireless router. There are many obvious applications for such networks, including emergency communications, vehicular communications, and military applications.

Such networks have dynamic (sometimes rapidly changing), random, multi-hop topologies that are composed of relatively bandwidth-constrained wireless links. MANETs must therefore support efficient operation in mobile wireless environments by incorporating routing functionality into mobile nodes. Note that such multi-hop networks exploit spatial reuse; transmissions can occur simultaneously on links that are sufficiently separated in space.

In NetSim MANETs, data packets are sent between source-destination pairs by multi-hop relaying. The MANETs in NetSim may operate in isolation or may have bridge nodes to interface with other networks (wired networks or even other MANETs).

NetSim MANET library supports the following protocols.

- Layer 3 Unicast Routing
  - Dynamic Source Routing (DSR)
  - Ad hoc On demand Distance Vector Routing (AODV)
  - Optimized Link state Routing (OLSR)
  - Zone Routing Protocol (ZRP)
- MAC / PHY (interfaced from NetSim Internetworks library)
  - 802.11 a, b, g, n, ac, p, and e

NetSim MANETs component can be interfaced with:

- NetSim Component 6 (IOT) module to run 802.15.4 in MAC/PHY
- NetSim Component 9 (VANETs) module to run IEEE 1609 WAVE in MAC/PHY
- NetSim TDMA Radio Networks (Add on) to run TDMA/DTDMA in MAC/PHY

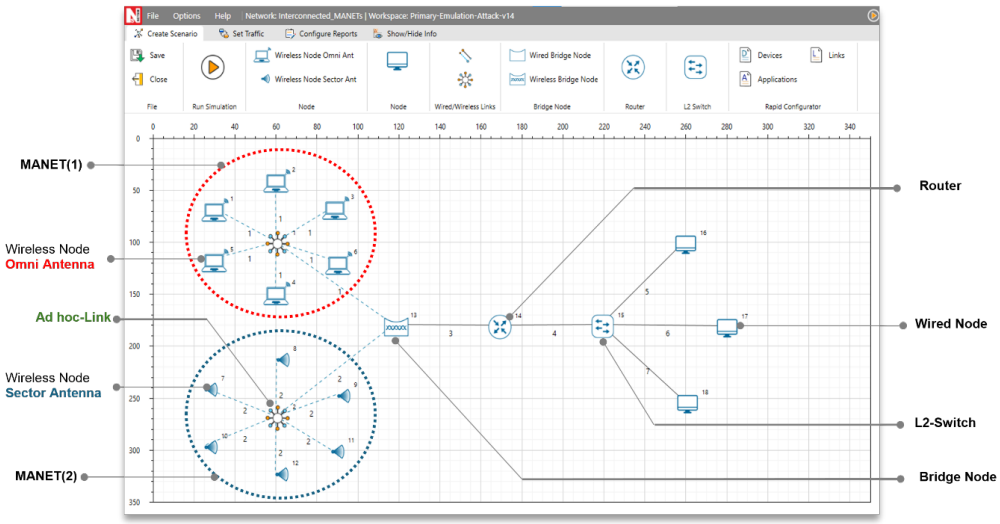


Figure 1-1: *A typical MANET Network scenario in NetSim. The topology shows multiple MANETs connected via a bridge node, and connected to an external network through a router*

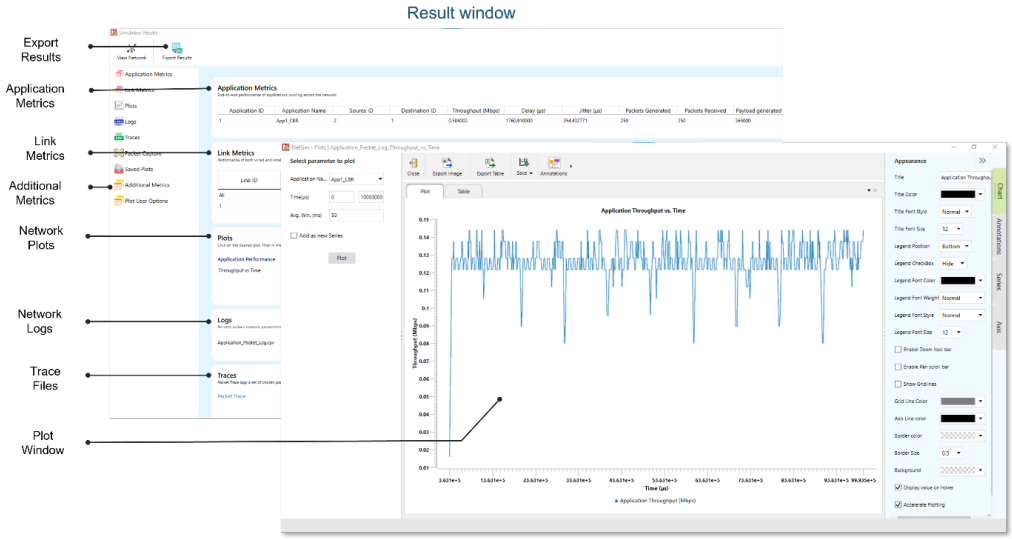


Figure 1-2: *The Result dashboard and Plot window shown in NetSim after completion of simulation*

The digital communication system employed, the transmit power used, and the radio propagation characteristics of the environment determine the "links" in the network.

Performance analysis of wireless ad hoc networks is a challenging task because such analysis must consider the interactions between the wireless physical layer, radio propagation, multiple access, random topology, routing, and the characteristics of the application that generates the traffic carried by the network. Therefore, unlike wired and fixed-topology networks, understanding and optimizing the performance of MANETs is a difficult undertaking owing to the complex interaction between the various "layers" of the network.

2 Simulation GUI

In the Main menu select New Simulation → Mobile Ad hoc networks as shown in Figure 2-1.

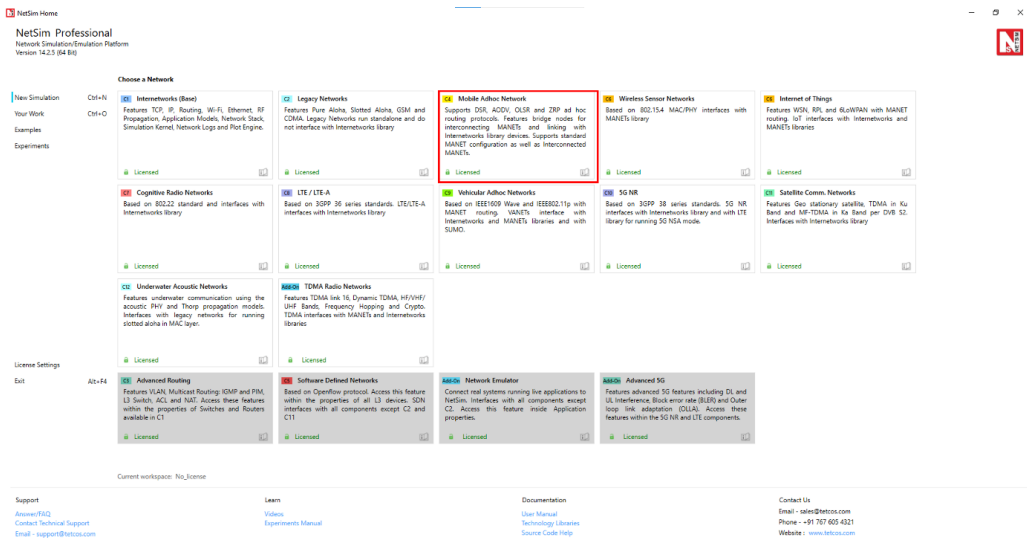


Figure 2-1: *NetSim Home Screen*

2.1 NetSim MANET Network Setup

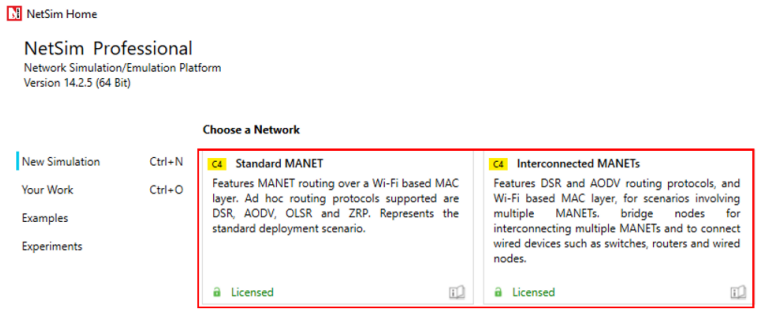


Figure 2-2: *MANET Network Setup window.*

2.1.1 Deployment Architecture

The deployment options have been grouped into 2 categories. Standard MANET option where the scenario comprises only wireless nodes without any bridge node. Interconnected MANETs option allows users to connect two or more MANETs using a bridge node.

Standard MANET: In Standard MANET, a network scenario can be created using ad-hoc links and wireless nodes. Standard MANET supports DSR, AODV, OLSR and ZRP Routing protocols.

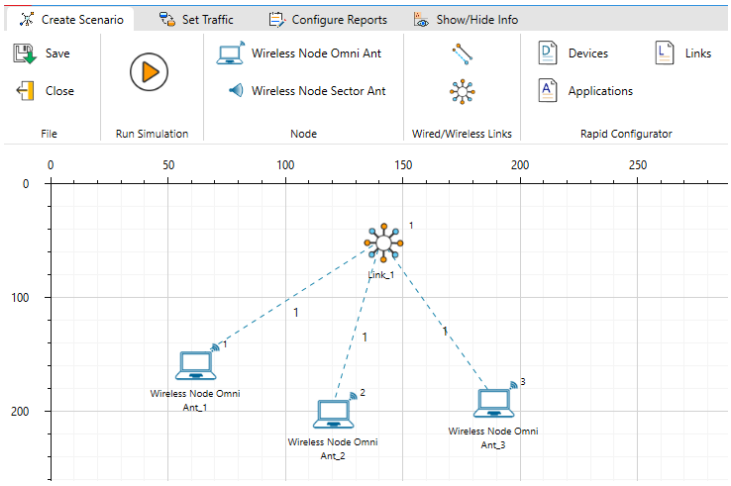


Figure 2-3: *A Single MANET scenario.*

Interconnected MANETs: In Interconnected MANETs, a network scenario can be created using multiple MANETs, bridge nodes for interconnecting multiple MANETs, and wired devices like switches, routers, and nodes. Interconnected MANETs support only DSR and AODV routing protocols.

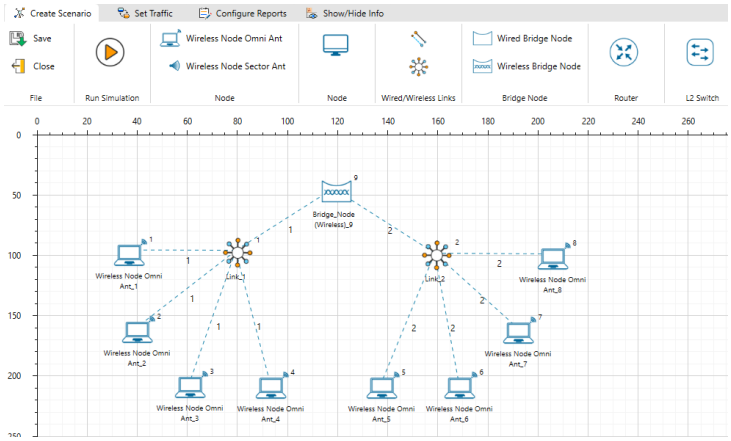


Figure 2-4: *Interconnected (Multiple) MANET Scenario*

2.2 Fast Configuration

When Mobile Ad Hoc Networks is selected, NetSim opens the Configure Mobile Ad Hoc Network (MANET) window, where the user can choose Automatic Placement or Manual Click-and-Drop and then select the placement method.

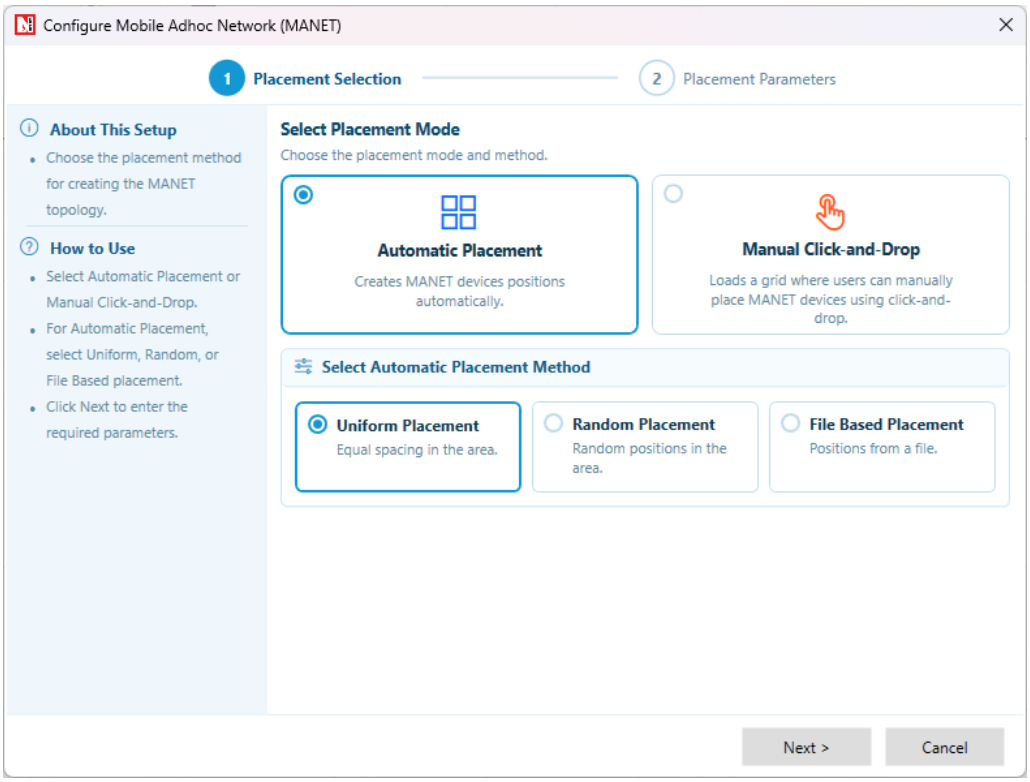


Figure 2-5: *Configure Mobile Ad Hoc Network (MANET) window*

2.2.1 Device Placement

Automatic Placement

- Uniform Placement: Devices are placed uniformly with equal gap between the devices in area based on the side length. This requires users to specify the number of devices as square numbers, such as 1, 4, 9, 16, etc.

- Random Placement: Devices are placed randomly in the grid environment within the area based on side length.
- File Based Placement: In order to place devices in user defined locations file-based placement options can be used. The file has the following general format:
- `<DEVICE NAME>,<DEVICE TYPE>,<X COORDINATE>,<Y COORDINATE>`

Where,

DEVICE NAME – The name assigned to the device.

DEVICE TYPE – The unique device identifier specific to each type of device in NetSim.

Table 2-1: *Fast Configuration window supports different networks.*

NETWORK	DEVICE TYPE
MANET	Wireless Node Omni Antenna Wireless Node Sector Antenna Wireless Bridge Node Wireless Bridge Node Wired Node Router L2 Switch
WSN	Sensors Sink node
IOT	IoT Sensors Gateway Wired Node IoT Router Access Point L2 Switch

X COORDINATE – The X-coordinate value of the device.

Y COORDINATE – The Y-coordinate value of the device.

Eg: MANET File-Based Placement.txt

```
Wireless Node Omni Antenna, WirelessNode,100,150
Wireless Node Omni Antenna, WirelessNode,150,100
Wireless Node Omni Antenna, WirelessNode,100,100
Wireless Node Omni Antenna, WirelessNode,50,50
```

Open NetSim, and in the Main menu, select New Simulation ▷ Mobile Ad Hoc Networks. Select File Based Placement option under Automatic Placement and give the path of the text file as shown in Figure 2-6 below.

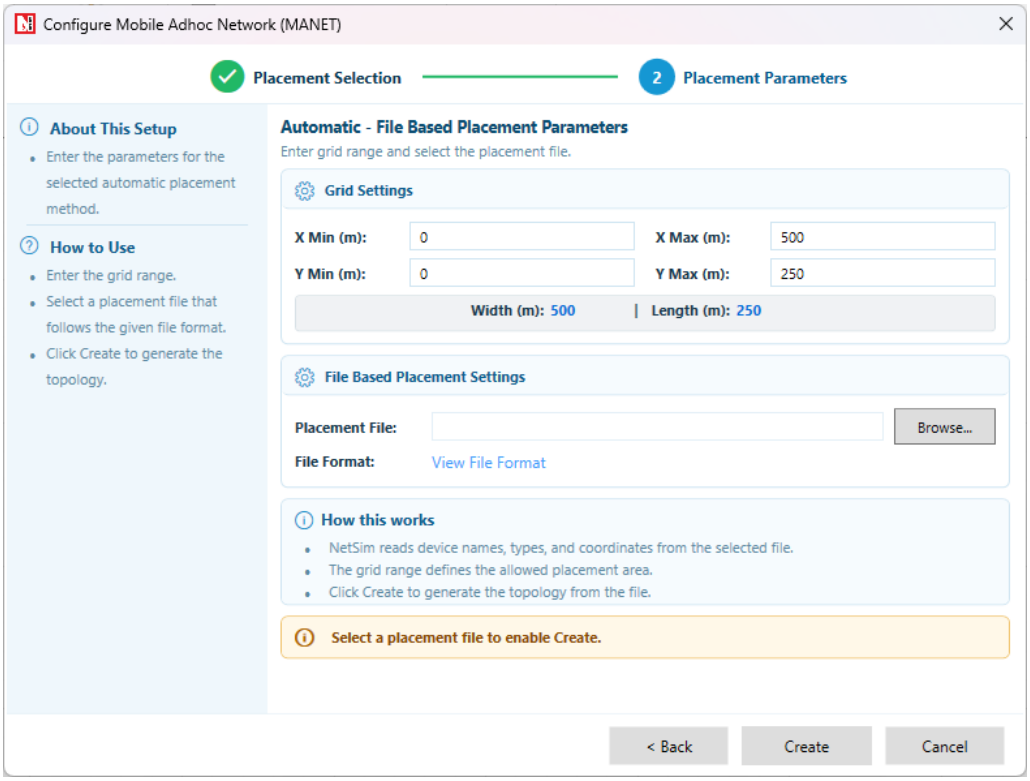


Figure 2-6: *Device placement strategy to File based placement*

After providing the path, clicking on OK will display the MANET network shown below, where all devices are placed as per the positions given in the text file as shown in Figure 2-7.

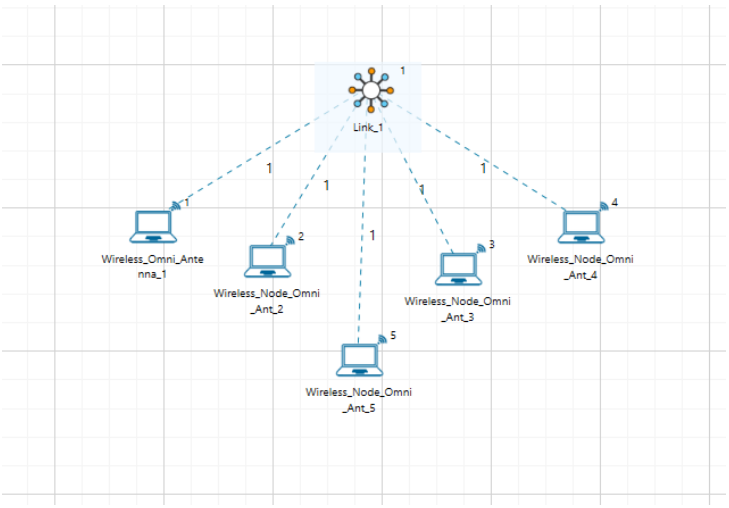


Figure 2-7: *Network Topology*

Number of Devices: It is the total number of devices that are to be placed in the grid environment. It should be a square number in case of uniform placement.

Manually Via Click and Drop

Selecting this option will load a link in the grid environment, where users can add devices by clicking and dropping them as required.

2.3 Create Scenario

2.3.1 Wireless Node

A MANET consists of mobile platforms – simply referred to as "wireless nodes" in NetSim – which are free to move about arbitrarily. They are IP addressable devices. Wireless Nodes in NetSim MANETs library act as both end-nodes and routers. These nodes make routing decisions using the IP fabric.

In NetSim MANETs, each node can have only one wireless interface.

2.3.2 Bridge Node

Bridge Node acts as a bridge/interface/gateway between multiple MANETs. Packets from one MANET can be routed to another MANET via a Bridge Node as shown in Figure 2-8.

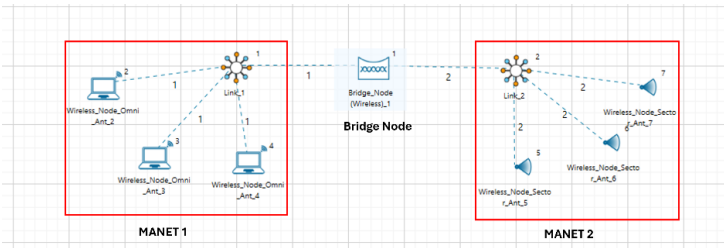


Figure 2-8: *Multiple MANETs connected via a Bridge node*

Each Bridge Node has 24 interfaces. When connecting bridge nodes to one another or to routers, care should be taken to ensure that the static routes are set.

NetSim supports Wired and Wireless Bridge Nodes as shown in Figure 2-9/Figure 2-10.

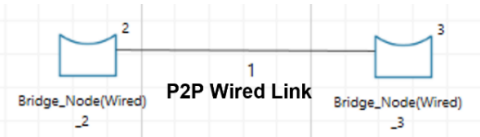


Figure 2-9: *Two wired bridge nodes can be connected to each other using P2P wired links*

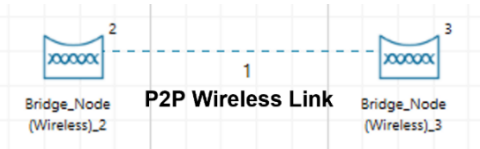


Figure 2-10: *Two wireless bridge nodes can be connected to each other using P2P wireless links*

A wired bridge node cannot be connected to a wireless bridge node and vice versa.

2.3.3 Link

At a given point in time, depending on the nodes' positions and their transmitter and receiver coverage patterns, transmission power levels and interference levels, wireless connectivity in the form of a random multi-hop graph or "ad hoc" network exists between the nodes. This ad hoc topology may change with time as the nodes move or adjust their transmission and reception parameters. Ad hoc links are used in NetSim to visually represent this connection of devices on an ad hoc basis as shown in Figure 2-11.

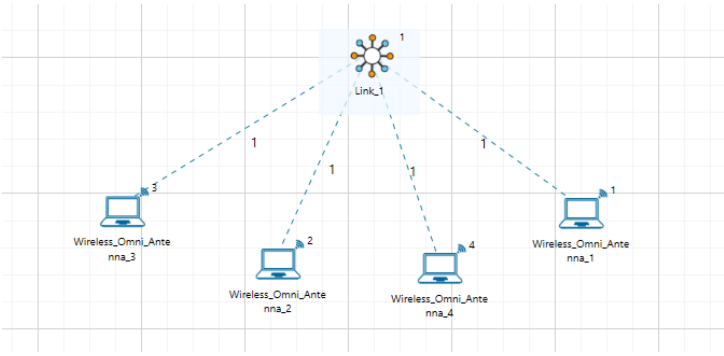


Figure 2-11: *Mobile Ad hoc Network*

Wireless links generally have significantly lower capacity than their hardwired counterparts. In addition, the realized throughput of wireless communications after accounting for the effects of multiple access, fading, noise, and interference conditions, etc. is often much less than a radio's maximum transmission rate.

Connecting ad hoc links is a one step process, by default one ad hoc link will be present in the grid. Users need to drop wireless nodes from the create scenario tab which gets automatically connected to ad hoc link.

An ad hoc link is a multipoint-to-multipoint link. When connected to a device its interface is set to 0 in NetSim.

2.4 Set Node, Link and Application Properties

Click on the appropriate node or link to open a right-side property panel. Then modify the parameters according to the requirements.

- Global Properties: Certain properties are global in nature, i.e., changing properties in one node will automatically reflect in the others in that network.
- In case of MANETs, in Wireless Nodes, Routing Protocol in Network Layer is global and all user editable properties in Datalink Layer, Physical Layer and Power are Local.
- The following are the main properties of wireless node omni ant in Datalink and Physical layers as shown in Figure 2-12/Figure 2-13.

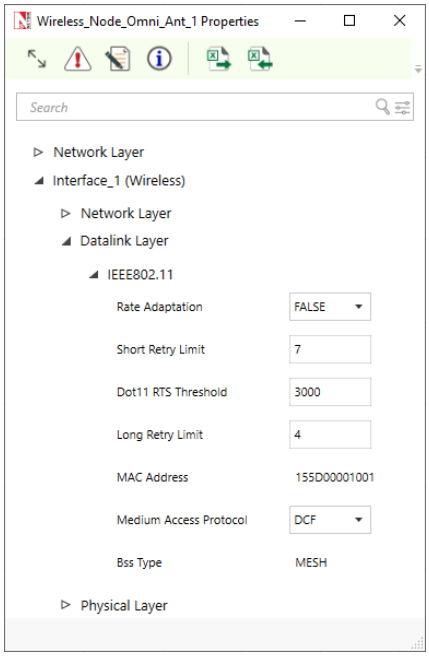


Figure 2-12: *Datalink layer properties window for wireless node omni ant*

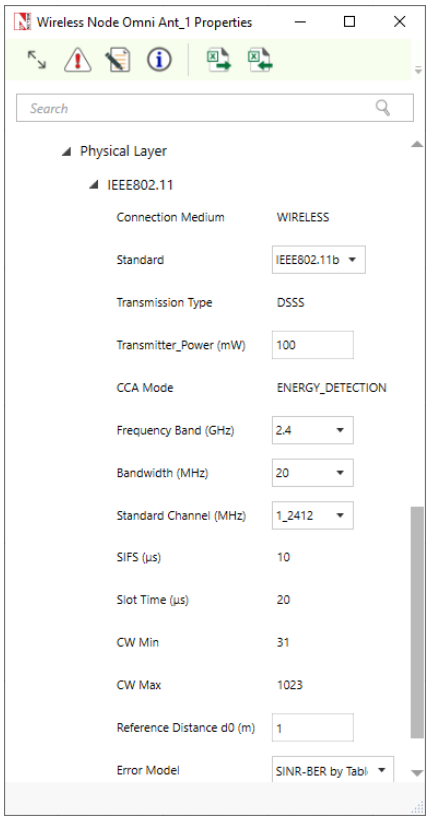


Figure 2-13: *Physical layer properties window for wireless node omni ant.*

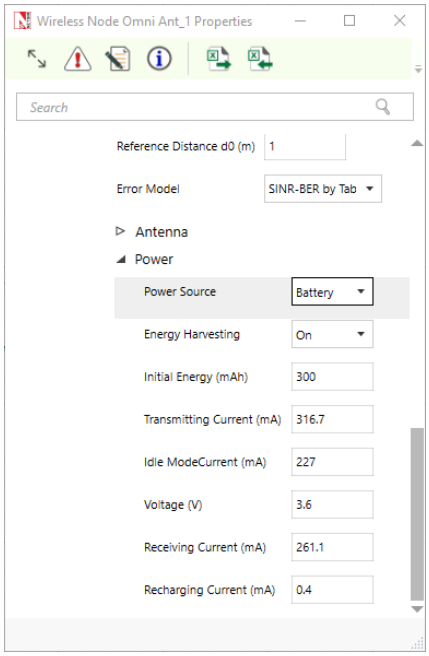


Figure 2-14: *Battery Model for Wireless node omni ant.*

Configure an application between any two nodes by selecting any application from the Set Traffic tab. Clicking on the application will open a right-side property panel where you can set the properties according to the requirement.

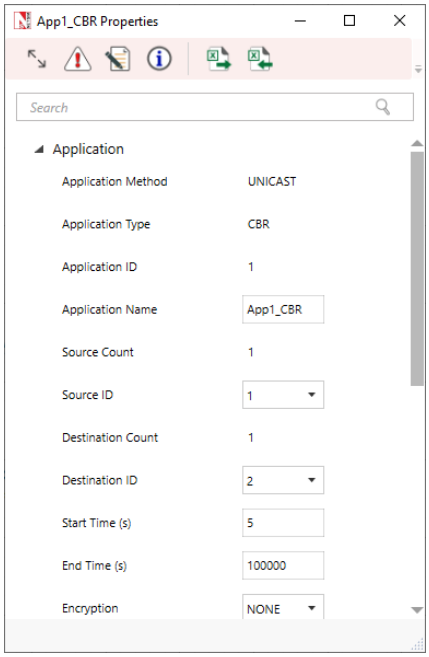


Figure 2-15: *Application icon and Configuration Window*

NOTE: For MANET networks the application start time should be a minimum of 5s, since this amount of time is required for convergence of OLSR/ZRP.

2.5 Enable Packet Trace, Event Trace (Optional)

To enable packet trace and event trace click on the configure reports tab in the ribbon on the top as shown below. For detailed help about the packet and event trace, please refer to sections 8.4 and 8.5 in the User Manual.

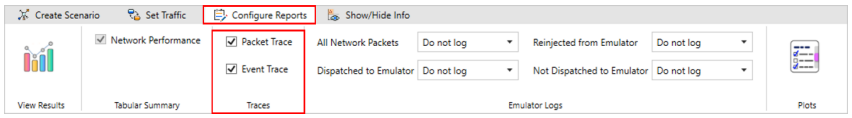


Figure 2-16: *Enable Packet Trace, Event Trace & Plots options on top ribbon.*

2.6 Enable protocol specific logs and plots

NetSim provides protocol-specific logs for MANET libraries, which users can enable before running a simulation. These can be enabled by clicking on configure reports in top ribbon ▷ clicking on plots ▷ choosing as desired, and running the simulation.

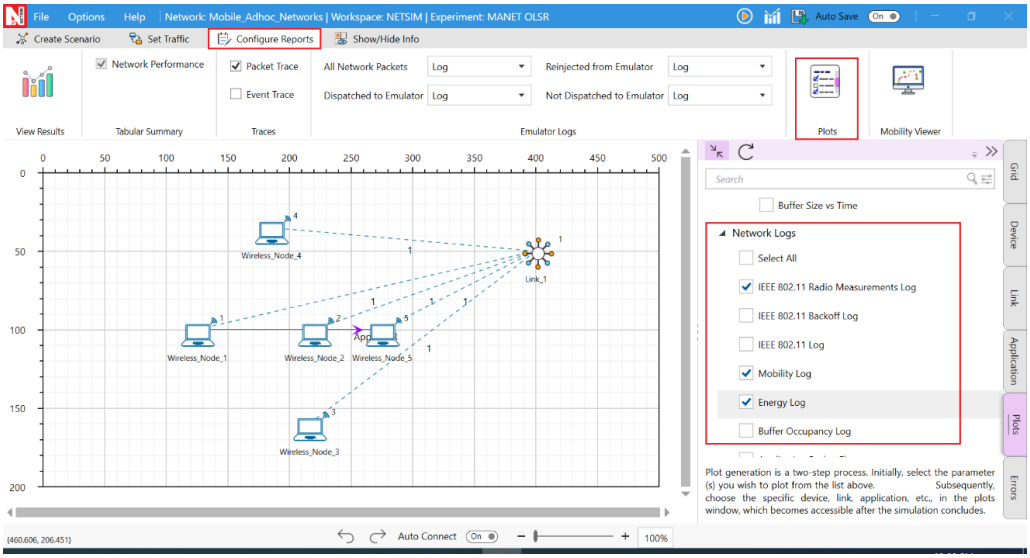


Figure 2-17: *Enabling the Network logs in MANET*

Similarly, users can enable the plots for Wi-Fi radio measurements and energy.

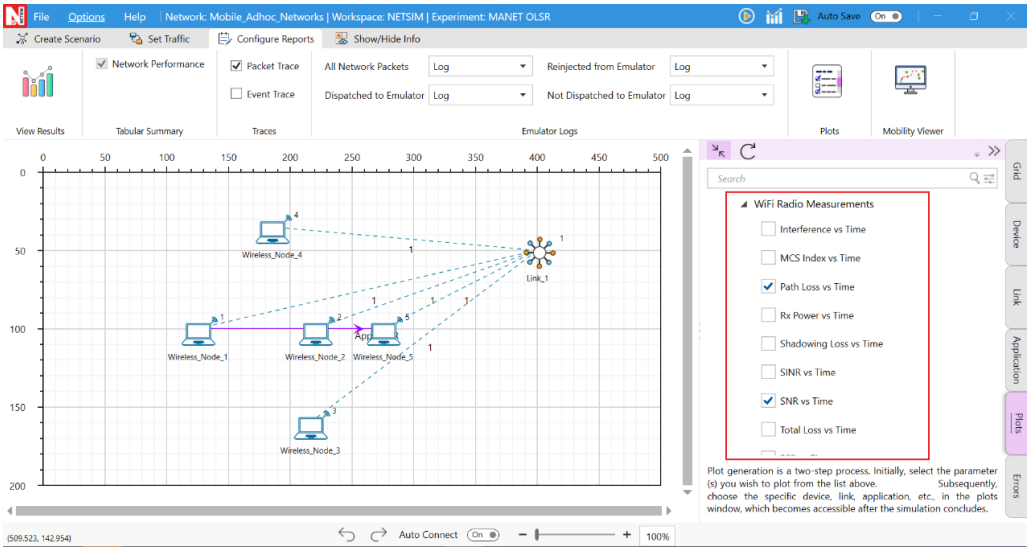


Figure 2-18: *Enabling the Plots in MANET*

2.7 GUI Configuration Parameters

Table 2-2: *Datalink layer, Network layer and Physical layer properties for Wireless Node Omni and Sector Ants.*

Parameter	Scope	Range	Description
Interface(Wireless) – Datalink Layer			
Rate Adaptation	Global (Standard MANETs) / Per MANET (Interconnected MANETs)	False	The algorithm is similar to Receiver based Auto Rate (RBAR) algorithm. In this, the PHY rate gets set based on the target PEP (packet error probability) for a given packet size. The adaptation is termed as "False" since the rate is pre-determined as per standard and there is no subsequent adaptation.
		Minstrel	Rate adaptation algorithm implemented in Linux.
		Generic	The algorithm is similar to the Auto Rate Fallback (ARF) algorithm. In this algorithm (i) rate goes up one step for 20 consecutive packet successes, and (ii) rate goes down one step after 3 consecutive packet failures.
Short Retry Limit	Local	1 to 255	Determines the maximum number of transmission attempts of a frame. The length of MPDU is less than/equal to Dot11 RTS Threshold value, made before a failure condition is indicated.
Long Retry Limit	Local	1 to 255	Determines the maximum number of transmission attempts of a frame. The length of MPDU is greater than Dot11 RTS Threshold value, made before a failure condition is indicated.
Dot11 RTS Threshold	Local	0 to 4692480	The size of packets (or A-MPDU if applicable) above which RTS/CTS (Request to Send / Clear to Send) mechanism gets triggered.

Continued on next page

Table 2-2: *(continued)*

Parameter	Scope	Range	Description
MAC Address	Fixed	Auto Generated	The MAC address is a unique value associated with a network adapter. This is also known as hardware address or physical address. This is a 12-digit hexadecimal number (48 bits in length).
Physical Type	Global	DSSS	Direct Sequence Spread Spectrum. The physical type of parameter is set to DSSS if the standard selected is IEEE802.11b.
		OFDM	Orthogonal Frequency Division Multiplexing is utilized as a digital multi-carrier modulation method. The physical type of parameter is set to OFDM if the standard selected is IEEE802.11a, g and p.
		HT	Operates in frequency bands 2.4GHz or 5GHz band. The physical type parameter is set to HT if the standard selected is IEEE802.11n.
		VHT	The physical type parameter is set to VHT if the standard selected is IEEE802.11ac.
Medium Access Protocol	Local	DCF	DCF is the process by which CSMA/CA is applied to Wi-Fi networks. DCF defines four components to ensure devices share the medium equally: Physical Carrier Sense, Virtual Carrier Sense, Random Back-off timers, and Interframe Spaces (IFS). DCF is used in non-QoS WLANs.

Continued on next page

Table 2-2: *(continued)*

Parameter	Scope	Range	Description
Medium Access Protocol	Local	EDCAF	QoS was introduced in 802.11e and is achieved using enhanced distributed channel access functions (EDCAFs). EDCA provides differentiated priorities to transmitted traffic, using four different access categories (ACs). With EDCA, high-priority traffic has a higher chance of being sent than low-priority traffic: a station with high priority traffic waits a little less before it sends its packet, on average, than a station with low priority traffic. **Access Categories under EDCAF:** AC_BK, AC_BE, AC_VI and AC_VO. **CWmin (Slots):** This attribute specifies the value of the minimum size of the window that is used by an AP for a particular AC for generating a random number for the backoff. The value of this attribute is such that dot11EDCATableCWmin could always be expressed in the form of $2^X - 1$, where X is an integer. The default value for this attribute is CWmin. Range is 0 to 255. **CWmax (Slots):** This attribute specifies the value of the maximum size of the window that is used by an AP for a particular AC for generating a random number for the backoff. The value of dot11EDCATableCWmax is such that it could always be expressed in the form of $2^X - 1$, where X is an integer. The default value for this attribute is CWmax. Range is 0 to 65535. **AIFSN (Slot):** This attribute specifies the number of slots, after a SIFS duration, that the STA for a particular AC senses the medium is idle before transmitting or executing a backoff. The default value for this attribute is 7. Range is 2 to 15. **MAX TXOP:** This attribute specifies the maximum number of microseconds of an EDCA TXOP for a given AC. The default value for this attribute is 0 for all PHYs. Range is 0 to 65535, with reference to the 802.11 standard. **MSDU Lifetime:** This attribute specifies, in TU, the maximum duration an MSDU for a given AC would be retained by the MAC before it is discarded. Range is 0 to 4294967295 TU.

Continued on next page

Table 2-2: *(continued)*

Parameter	Scope	Range	Description
OCBA Activated	Local	True or False	This parameter determines the type of standard to be chosen for the OFDM physical type. The standard is set to IEEE802.11p if OCBA is True. The standard is set to IEEE802.11a and g if OCBA is False.
BSS Type	Fixed	Auto Generated	The BSS type is fixed to Infrastructure mode. The wireless device can communicate with each other or with a wired network.
Interface(Wireless) – Physical Layer			
Protocol	Fixed	IEEE 802.11	Defines the MAC and PHY specifications like IEEE802.11a/b/g/n/ac/p for wireless connectivity for fixed, portable and moving stations within a local area.
Connection Medium	Fixed	Auto Generated	Defines how the devices are connected or linked to each other.
Standard	Global (Standard MANETs) / Per MANET (Interconnected MANETs)	IEEE 802.11 a/b/g/n/ac/p	Refers to a family of specifications developed by IEEE for WLAN technology. The IEEE standards supported in NetSim are IEEE 802.11 a, b, g, n, ac and p. 802.11a provides up to 54 Mbps in the 5GHz band. 802.11b provides 11 Mbps in the 2.4GHz bands. 802.11g provides 54 Mbps transmission over short distances in the 2.4 GHz band. 802.11n adds up MIMO. 802.11ac provides support for wider channels and beamforming capabilities. 802.11p provides support to Intelligent Transportation Systems.
Transmission Type	Fixed	DSSS	Direct Sequence Spread Spectrum (DSSS), a radio transmission technique that spreads a narrowband signal across a wider carrier frequency band. Each transmission is assigned a 10-bit pseudorandom binary code sequence, which comprises a series of ones and zeros in a seemingly random pattern known to both the transmitter and receiver.
		OFDM	Orthogonal Frequency Division Multiplexing (OFDM), a frequency-division multiplexing (FDM) scheme utilized as a digital multi-carrier modulation method. A large number of closely-spaced orthogonal sub-carriers are used to carry data. The data is divided into several parallel data streams or channels, one for each sub-carrier. Each sub-carrier is modulated with a conventional modulation scheme such as Quadrature Amplitude Modulation or Phase Shift Keying at a low symbol rate, maintaining total data rates similar to conventional single-carrier modulation schemes in the same bandwidth.

Continued on next page

Table 2-2: *(continued)*

Parameter	Scope	Range	Description
		HT	High Throughput (HT). HT stands for High Throughput. The IEEE 802.11 HT STA operates in frequency bands 2.4 GHz / 5 GHz band.
		VHT	Very High Throughput (VHT). VHT stands for Very High Throughput. The IEEE 802.11 VHT STA operates in frequency bands below 6 GHz excluding the 2.4 GHz band. Most VHT features, among other benefits, increase the maximum throughput achievable between two VHT STAs over that achievable using HT (High Throughput) features alone.
Transmit Power	Local	0 to 1000	Transmitted signal power. Note that the transmit power is not split among the antennas. This value is applied to each antenna in a multi-antenna transmitter. Unit is mW.
CCA Mode	Fixed	Auto generated	A mechanism to determine whether a medium is idle or not. Includes Carrier sensing and energy detection.
Frequency Band	Local	2.4, 5, 5.9 (Depends on the standard chosen)	The centre frequency of the band at which the device is operating. Unit is GHz.
Bandwidth	Local	5,10, 20, 40, 60, 80, 160 (Depends on the standard chosen)	Bandwidth is a range of frequencies occupied by radio communication signal to carry most of its energy. Unit is MHz.
Standard Channel	Local	Depends on the standard chosen	The channel options defined in the standards. The options would also depend on the frequency band if the standard supports multiple bands.
SIFS	Fixed	Auto Generated	The time interval required by a wireless device in between receiving a frame and responding to the frame. Unit is microseconds.
Slot Time	Fixed	Auto Generated	Time is quantized as slots in Wi-Fi. Unit is microseconds.
Guard Interval	Local	400 and 800	Guard Interval is intended to avoid signal loss from multipath effect. Unit is nanoseconds.
MCS Selection	Local	Auto Rate Fallback, Fixed	MCS selection in Wi-Fi impacts data rates and efficiency. Auto Rate Fallback adapts the MCS based on signal quality. Fixed MCS locks the MCS. Default Value: Auto Rate.
Data MCS	Local	802.11b: 0–3, 802.11a/g/p: 0–7, 802.11n: 0–7, 802.11ac: 0–9 (MCS 9 not available for 20MHz in VHT)	Allows selection of the MCS value for different Wi-Fi standards. Determines the modulation and coding scheme. Default Value: 0.
Data PHY Rate (Mbps)	Local	Determined by selected Data MCS and Wi-Fi standard	Shows the physical layer data rate based on the chosen modulation and coding scheme.

Continued on next page

Table 2-2: *(continued)*

Parameter	Scope	Range	Description
CW Min	Fixed	Auto generated	The minimum size of the Contention Window in units of slot time. The CW min is used by the MAC to calculate the back off time for channel access during a carrier sense.
CW Max	Fixed	Auto generated	The maximum size of the Contention Window in units of slot time. The CW is doubled progressively when collisions occur.
Error Model	Local	SINR-BER-By-Table, SINR-BER-By-Formula	Specifies how the Bit Error Rate (BER) is calculated: BER is determined based on predefined tables mapping SINR to BER. BER is calculated using mathematical formulas that account for the modulation and coding schemes used, based on the SINR value.
Antenna Height	Local	0 to 100m	It is used in the pathloss calculation in the following models: Cost231 Hata Urban, Cost231 Hata SubUrban, Hata Urban, Hata SubUrban and Two Ray. This parameter has no effect when using any of the other pathloss models. Default:0.0 m.
Antenna Gain	Local	0 to 1000 dB	A relative measure of an antenna's ability to direct or concentrate radio frequency energy in a particular direction or pattern. The measurement is typically measured in dBi (Decibels relative to an isotropic radiator).
Antenna Type	Fixed	Omnidirectional or Sector antenna	NetSim supports two types of Antenna, Omnidirectional and Sector Antennas.
Wireless Node Sector Ant			
Antenna Model	Fixed	–	Currently NetSim only supports 2D passive antenna per 3GPP TR 37.840.
Orientation Angle	Local	0°–360°	NetSim implements a 2D parabolic sector antenna as per 3GPP TR 37.840. The boresight angle denotes the direction of maximum gain, or the highest radiated power. The angle is defined to start at 0 from the positive X-axis. If positive Y points downward, the angle increases on clockwise rotation from the positive X-axis. If positive Y points upward, the angle increases in an anti-clockwise direction from the positive X-axis. The units for the boresight angle are in degrees.
Element Gain (dB)	Local	−50 to +50	This is the maximum directional gain of the radiation element (in dB). The default value is 8 dBi.
Front to Back Ratio (dB)	Local	10–40	The ratio of power gain between the front and rear of a directional antenna.
Beam Width	Local	0–90	The 3 dB, or half power, beamwidth of the antenna is defined as the angular width of the radiation pattern, between points 3 dB down from maximum beam level (beam peak).
Power Model			

Continued on next page

Table 2-2: *(continued)*

Parameter	Scope	Range	Description
Power Source	Local	Main Line or Battery	MANETs communicate with each other using battery power. By default, the power model is set to Main Line, which represents a general-purpose alternating current (AC) electric power supply. The power model is user-configurable, with adjustable properties.
Energy Harvesting	Local	On or Off	Energy harvesting is the process of deriving energy from external sources (e.g., solar power, thermal energy, wind energy, and kinetic energy), capturing it, and storing it for use in small, wireless autonomous devices, such as those in wearable electronics and wireless sensor networks. NetSim supports an abstract energy harvesting model in which a specified amount of energy (calculated from the recharging current and specified voltage) is periodically added to the remaining energy of the node to replenish the battery. This feature can be turned on or off.
Initial Energy	Local	0.001–1000 mAh	A node has an initial value which is the level of energy the node has at the beginning of the simulation.
Transmitting Current	Local	0–5000 mA	In the Transmitting mode (Tx mode), the node consumes energy to transfer packets or data. The amount of energy consumed in this mode depends on the number of packets sent by the node; the greater the number of packets, the more energy is consumed.
Idle Mode Current	Local	0–500 mA	In idle mode, a node doesn't transmit or receive data but still listens to the wireless medium for potential packets and new nodes. This consumes less energy than sending or receiving, as no active communication occurs.
Voltage	Local	0–10 V	Voltage is a measure of the energy carried by the charge.
Receiving Current	Local	0–1000 mA	In the Receiving mode (Rx mode), the nodes are actively listening to the incoming data, it consumes the energy as it receives the data from the sender.
Recharging Current	Local	0–20 mA	Recharging Current refers to the flow of electric charge supplied to a battery during the recharging process.
Network Layer			

Continued on next page

Table 2-2: *(continued)*

Parameter	Scope	Range	Description
Routing Protocol	Global	DSR, AODV, ZRP, OLSR	AODV (Ad hoc on Demand Distance Vector) is an on-demand routing protocol for wireless networks that uses traditional routing tables to store routing information. AODV uses timers at each node and expires the routing table entry after the route is not used for a certain time. DSR (Dynamic Source Routing) is a routing protocol for wireless mesh networks. It is similar to AODV, in that it forms a route on-demand when a transmitting computer requests one. However, it uses source routing instead of relying on the routing table at each intermediate device. ZRP (Zone Routing Protocol) is a hybrid Wireless Networking routing protocol that uses both proactive and reactive routing protocols when sending information over the network. ZRP divides the entire network into zones of variable size. Every node in the network has a zone associated to it. OLSR (Optimized Link State Routing Protocol) is a proactive link-state routing protocol, which uses hello and topology control (TC) messages to discover and then disseminate link state information throughout the mobile ad hoc network.
DSR Routing Protocol			
ACK Type	Global	Link Layer ACK or Network Layer ACK	The user can enable either Link Layer ACK (Layer 2 ACK) or Network Layer ACK (Layer 3 ACK). Link Layer ACK uses MAC layer acknowledgment for route maintenance, while Network Layer ACK uses DSR acknowledgment for route maintenance. For more details, refer to sections 3.2.1 and 3.2.2 of the MANET Technology Library.
AODV Routing Protocol			
Hello Message	Global	Enable/Disable	Hello messages are periodic broadcasts used to maintain local connectivity and discover neighbors, ensuring that nodes are aware of each other's presence. Enabled: You will observe Hello packets being sent and received in the simulation. Disabled: Hello packets are not transmitted or received. Without HELLO messages, AODV's route discovery (RREQ/RREP) remains the same. However, route maintenance shifts from proactive local link sensing (via HELLO) to reactive link break detection.
ZRP and OLSR Routing Protocol			

Continued on next page

Table 2-2: *(continued)*

Parameter	Scope	Range	Description
Hello Interval	Global	1–100 s	Hello interval parameter is used for neighbor discovery process. This parameter determines how frequently Hello messages are sent out and also how frequently a neighbor table will be updated.
Refresh Interval	Global	1–100 s	Refresh interval is the duration after which each active node periodically refreshes routes to itself.
IARP	Fixed	–	Intra-zone Routing Protocol (IARP) is used by a node to communicate with the interior nodes of its zone. It proactively maintains routes to destinations within the local routing zone, limited by the zone radius.
TC Interval	Global	1–100 s	Topology Control (TC) messages are sent at the TC interval. This parameter defines how frequently TC messages are generated and exchanged to advertise topology information in the network.
Zone radius	Global	2–225 m	ZRP divides the entire network into zones. The radius of these zones is defined by the Zone radius parameter. It determines the extent of the local routing zone maintained proactively by IARP.

2.8 Run Simulation

Click on Run Simulation icon on the top toolbar.

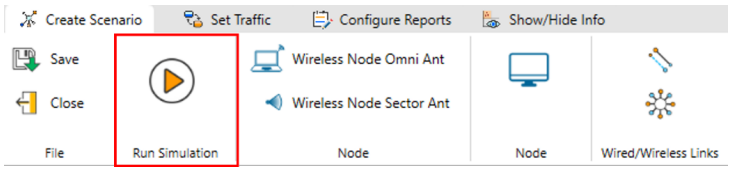


Figure 2-19: *Run Simulation option on top ribbon*

Set the Simulation Time and click on OK.

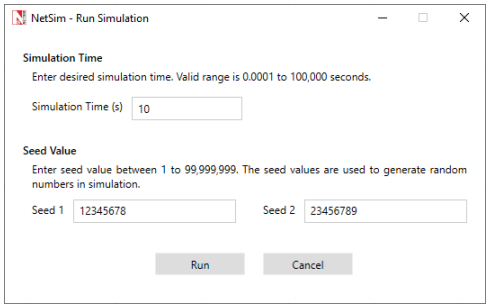


Figure 2-20: *Run Simulation window*

NOTE: MANET implementation in NetSim:

- If a user wants to implement an HTTP application among nodes, TCP must be enabled in source node as TCP is set to disable by default.

- OLSR is a proactive link-state routing protocol. It uses Hello and topology control (TC) messages to discover and then disseminate link state information throughout the mobile ad hoc network.
- Individual nodes use this topology information to compute next hop destinations for all nodes in the network using shortest hop forwarding paths. For topology control (TC) messages to disseminate throughout, it requires 5 or more seconds depending upon the network size. In general, it is (5.5 secs + Tx Time × network size).
- Hence, when simulating OLSR in MANET the Application Start Time must be greater than 5s (preferably greater than 10s) because in OLSR Topology Control (TC) messages start at 5s. Once the TC messages are sent, some further time will be required for OLSR to find the route. This can be done by setting the "Start Time" parameter in Application properties.

3 Model Features

3.1 Ad hoc On Demand Distance Vector Routing

The Ad Hoc On-Demand Distance Vector (AODV) algorithm enables dynamic, self-starting, multi-hop routing between participating mobile nodes wishing to establish and maintain an ad hoc network. AODV allows mobile nodes to respond to link breakages and changes in network topology in a timely manner. When links break, AODV causes the affected set of nodes to be notified so that they can invalidate the routes using the lost link. AODV in NetSim can run in layer 3 over MAC/PHY protocols such as 802.11, 802.15.4, TDMA and DTDMA.

The features of AODV implemented in NetSim are:

Broadcast Route Discovery Mechanism, and Route Maintenance: AODV accomplishes this with the help of RREQ and RREP. The packet trace file in NetSim helps to view and understand how AODV accomplishes this mechanism using RREQ and RREP packets.

Timer - based states: Routing table states expire when the route remains inactive for a certain period of time. This is established by the use of the lifetime field in the routing table and RREP packets. The lifetime field can be observed in Wireshark under the AODV RREP packets.

Sequence numbers: The source sequence number is a monotonically increasing number maintained by each source node. It is used by other nodes to determine whether the information contained in the packet sent by the source node is new. The destination sequence number is created by destination and includes it with the routing information sent to the requesting nodes. Usage of destination sequence number ensures loop freedom. The route with the greatest sequence number is used by requesting nodes to reach their destination. This also prevents routing loops and avoids using inactive or old routes. The source and destination sequence numbers can be viewed in Wireshark under the RREQ and RREP packets in AODV protocols. Some important fields of the RREQ format that can be viewed in NetSim using Wireshark are listed in Table 3-1 below.

Table 3-1: *Important fields of the RREQ format viewed in NetSim using Wireshark*

Field	Description
Hop Count	The number of hops from the Originator IP Address to the node handling the request.
Destination IP Address	The IP address of the destination for which a route is desired.
Destination Sequence Number	The latest sequence number received in the past by the originator for any route towards the destination.
Originator IP Address	The IP address of the node which originated the Route Request.
Originator Sequence Number	The current sequence number to be used in the route entry pointing toward the originator node.

Some important fields of the RREP format that can be viewed in NetSim using Wireshark are listed in Table 3-2 below.

Table 3-2: *Important fields of the RREP format viewed in NetSim using Wireshark*

Field	Description
Hop count	The number of hops from the Originator IP Address to the Destination IP Address.
Destination IP Address	The IP address of the destination for which a route is supplied.
Destination Sequence Number	The destination sequence number associated to the route.
Originator IP Address	The IP address of the node which sends the RREQ.
Lifetime	The time in milliseconds for which nodes receiving the RREP consider the route to be valid.

3.2 Dynamic Source Routing (DSR)

The Dynamic Source Routing (DSR) protocol is a simple and efficient routing protocol designed specifically for multi-hop wireless ad hoc networks of mobile nodes. Using DSR, the network is completely self-organizing and self-configuring, requiring no existing network infrastructure or administration.

Network nodes cooperate to forward packets for each other, enabling communication over multiple "hops" between nodes that are not directly within wireless transmission range of one another. As nodes move, join, or leave the network, and as wireless transmission conditions (e.g., interference sources) change, all routing is automatically determined and maintained by the DSR protocol.

Since the number or sequence of intermediate hops needed to reach any destination may change at any time, the resulting network topology can be dynamic and rapidly evolving.

In NetSim, DSR operates at Layer 3 over MAC/PHY protocols such as 802.11, 802.15.4, TDMA, and DTDMA.

3.2.1 Using Link-Layer Acknowledgements

If the MAC protocol in use provides feedback as to the successful delivery of a data packet (such as is provided for unicast packets by the link-layer acknowledgement frame defined by IEEE 802.11), then the use of the DSR Acknowledgement Request and Acknowledgement options is not necessary. If such link-layer feedback is available, it should be used instead of any other acknowledgement mechanism for Route Maintenance, and the node should not use either passive acknowledgements or network-layer acknowledgements for Route Maintenance.

When using link-layer acknowledgements for Route Maintenance, the retransmission timing and the timing at which retransmission attempts are scheduled are generally controlled by the particular link layer implementation in use in the network. For example, in IEEE 802.11, the link-layer acknowledgement is returned after a unicast packet as a part of the basic access method of the IEEE 802.11 Distributed Coordination Function (DCF) MAC protocol; the time at which the acknowledgement is expected to arrive and the time at which the next retransmission attempt (if necessary) will occur are controlled by the MAC protocol implementation.

3.2.2 Using Network-Layer Acknowledgements

When a node originates or forwards a packet and has no other mechanism of acknowledgement available to determine reachability of the next-hop node in the source route for Route Maintenance, that node should request a network-layer acknowledgement from that next-hop node. To do so, the node inserts an Acknowledgement Request option in the DSR Options Header in the packet. The Identification field in

that Acknowledgement Request option must be set to a unique value over all packets recently transmitted by this node to the same next-hop node.

When using network-layer acknowledgements for Route Maintenance, a node should use an adaptive algorithm in determining the retransmission timeout for each transmission attempt of an acknowledgement request. For example, a node should maintain a separate round-trip time (RTT) estimate for each node to which it has recently attempted to transmit packets, and it should use this RTT estimate in setting the timeout for each retransmission attempt for Route Maintenance.

While simulating certain network configurations, users may see that packets received are more than packets sent. This is because:

- This is being measured as part of our UDP protocol metrics in layer 4 in the source and in the destination.
- Let us say UDP protocol at source node A sends a datagram. At the MAC - WLAN sends the frame and starts a retransmission timer.
- If no Ack is received within this timer period, it would initiate a re-transmission (consider cases where the WLAN Ack has a collision or is errored).
- As the destination, the MAC (WLAN) layer would send up to UDP both the first packet it received and the re-transmitted packet it received.
- UDP protocol in the destination would count both the packets received.

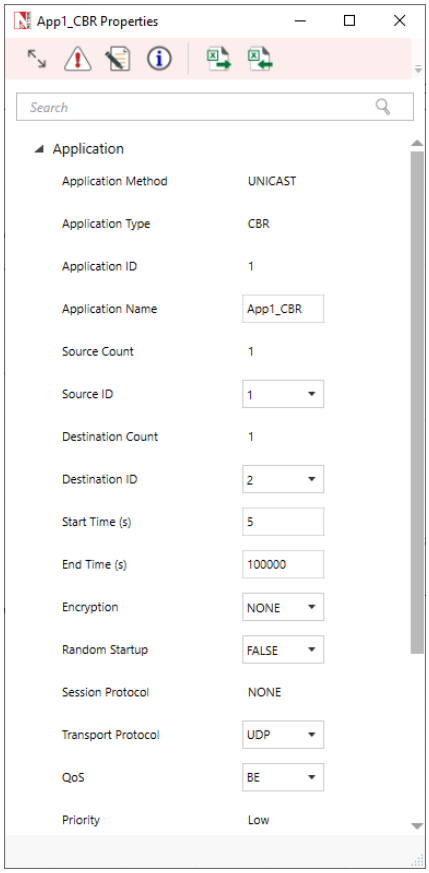


Figure 3-1: *Application Configuration window*

3.2.3 Why do we see continuous RREQ and RREP packets in DSR?

When the Data packet is to be transmitted, in DSR, within the NETWORK OUT Event, the DSR-packet-processing function is called. This initiates Route discovery, if no route to destination exists. Once discovery is complete, DSR switches to Route maintenance mode.

Route Maintenance is the mechanism by which a source node S is able to detect, while using a source route to some destination node D, if the network topology has changed such that it can no longer use its route to D because a link along the route no longer works.

In Route Maintenance mode, DSR checks for ACKs for every data packet sent. If the ACK for a data packet is not received within a specified time (as defined by the Maintenance Time Out) then Route Error (RERR) gets triggered. This trigger leads to removal of Route entries from the Route Cache thereby causing Route discovery to be initiated again.

A typical simulation scenario where Route discovery, Route Error, Route discovery cycle is repeated, is when the network has multiple applications generating data packets at the exact same time. This can cause packet collisions in the MAC layer. Since no packet is received no ACK is sent. The non-receipt of ACK within the specified time causes RERR to get triggered.

3.3 AODV/DSR Metrics

AODV or DSR Metrics table will be part of the results dashboard if routing protocol in the network layer is set to either AODV or DSR in at least one device in the simulated network scenario.

Table 3-3: *Parameter Description for AODV/DSR Metrics table*

Parameter	Description
Device Id	It is the unique ID of the wireless node
RREQ sent	It is the number of Route Request packets sent by wireless node during Route Discovery process
RREQ forwarded	It is the number of Route Request packets forwarded by wireless node during Route Discovery process
RREP sent	It is the number of Route Reply packets sent by wireless node when route is found during Route Discovery process
RREP forwarded	It is the number of Route Reply packets forwarded by a wireless node when route is found during Route Discovery process
RERR sent	It is the total number of Route Error packets sent by a wireless node during Route Maintenance
RERR forwarded	It is the total number of Route Error packets forwarded by a wireless node during Route Maintenance
Packet originated	It is the total number of packets originated in a source node
Packet transmitted	It is the total number of packets transmitted by a source node and intermediate device (LoWPAN Gateway or sink node)
Packet dropped	It is the total number of packets dropped by a wireless node

3.4 Zone Routing Protocol (ZRP)

The ZRP is based on two procedures:

1. Intrazone Routing Protocol (IARP) and
2. Interzone Routing Protocol (IERP).

Through the use of the IARP, each node learns the identity of and the (minimal) distance to all the nodes in its routing zone. The actual IARP is not specified and can include any number of protocols, such as the derivatives of Distance Vector Protocol (e.g., Ad Hoc On-Demand Distance Vector, Shortest Path First (e.g., OSPF). In fact, different portions of an ad hoc network may choose to operate based on different choices of the IARP protocol. Whatever the choice of IARP is, the protocol needs to be modified to ensure that the scope of this operation is restricted to the zone of the node in question.

Note that as each node needs to learn the distances to the nodes within its zone only, the nodes are updated about topological changes only within their routing zone. Consequently, in spite of the fact that a network can be quite large, the updates are only locally propagated.

IERP

While IARP finds routes within a zone, IERP is responsible for finding routes between nodes located at distances larger than the zone radius. IERP relies on border casting. Border casting is possible as any node knows the identity and the distance to all the nodes in its routing zone by the virtue of the IARP protocol.

The IERP operates as follows: The source node first checks whether the destination is within its routing zone. (Again, this is possible as every node knows the content of its zone). If so, the path to the destination is known and no further route discovery processing is required. If, on the other hand, the destination is not within the source's routing zone, the source border casts a route request (referred to here as a "request") to all its peripheral nodes. Now, in turn, all the peripheral nodes execute the same algorithm: check whether the destination is within their zone. If so, a route reply (referred to here as a "reply") is sent back to the source indicating the route to the destination. If not, the peripheral node forwards the query to its peripheral nodes, which, in turn, execute the same procedure. An example of this Route Discovery procedure is demonstrated in Figure 3-1 below.

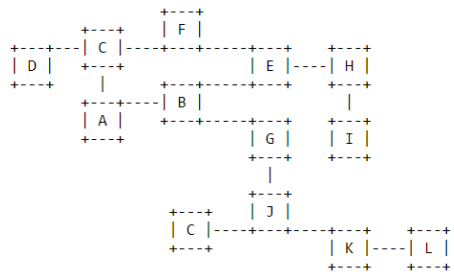


Figure 3-2: *Interzone Routing Protocol Discovery procedure*

Node A has a datagram to node L. Assume routing zone radius of 2. Since L is not in A's routing zone (which includes B, C, D, E, F, G), A bordercasts a routing request to its peripheral nodes: D, F, E, and G. Each one of these peripheral nodes check whether L exists in their routing zones. Since L is not found in any routing zones of these nodes, the nodes border cast the request to their peripheral nodes. In particular, G border casts to K, which realizes that L is in its routing zone and returns the requested route (L-K-G-A) to the query source, namely A.

IARP

In Zone Routing, the Intra zone Routing Protocol (IARP) proactively maintains routes to destinations within a local neighborhood, which we refer to as a routing zone. More precisely, a node's routing zone is defined as a collection of nodes whose minimum distance in hops from the node in question is no greater than a parameter referred to as the zone radius. Note that each node maintains its own routing zone. An important consequence is that the routing zones of neighboring nodes overlap.

An example of a routing zone (for node A) of radius 2 is shown in Figure 3-2 below.

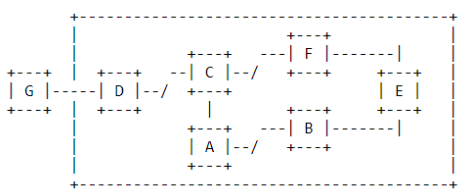


Figure 3-3: *Intra zone Routing Protocol Discovery procedure*

In this example, nodes B through F are within the routing zone of node A, while node G is outside node A's routing zone. Node E can be reached via two paths from A: one with a length of 2 hops and the other with a length of 3 hops. Since the minimum hop count is less than or equal to 2, E is included in A's routing zone.

Peripheral Nodes are nodes whose minimum distance to a given node is exactly equal to the zone radius. Thus, in the example above, nodes D, F, and E are the peripheral nodes of node A.

To construct a routing zone, a node must first identify its neighbors. A neighbor is a node with which direct (point-to-point) communication can be established and is, therefore, one hop away. Neighbor identification can be provided by MAC protocols, such as in polling-based protocols, or by a separate Neighbor Discovery Protocol (NDP).

Neighbor discovery typically involves the periodic broadcasting of "HELLO" beacons. The reception (or quality of reception) of these beacons is used to determine the status of the connection to the beaconing neighbor.

This neighbor discovery information forms the basis of the Intra zone Routing Protocol (IARP). IARP can be derived from globally proactive link-state routing protocols that offer a complete view of network connectivity.

3.5 Optimized Link State Routing Protocol (OLSR)

OLSR is developed for mobile ad hoc networks. It operates as a table driven, proactive protocol, i.e., exchanges topology information with other nodes of the network regularly. Each node selects a set of its neighbor nodes as "multipoint relays" (MPR). In OLSR, only nodes, selected as such MPRs, are responsible for forwarding control traffic, intended for diffusion into the entire network. MPRs provide an efficient mechanism for flooding control traffic by reducing the number of transmissions required.

Nodes selected as MPRs also have a special responsibility for declaring link state information in the network. Indeed, the only requirement for OLSR to provide shortest path routes to all destinations is that MPR nodes declare link-state information for their MPR selectors. Additional available link-state information may be utilized, e.g., for redundancy.

Nodes which have been selected as multipoint relays by some neighbor node(s) announce this information periodically in their control messages. This allows a node to announce to the network that it has reachability to the nodes which have selected it as an MPR. In route calculation, MPRs are used to form the route from a given node to any destination in the network. Furthermore, the protocol uses the MPRs to facilitate efficient flooding of control messages in the network.

3.5.1 One Hop neighbor

Node X is considered a neighbor of node Y when node Y is within communication range of node X, meaning there is an established link between OLSR interfaces on both nodes that enables direct communication.

3.5.2 Two Hop neighbor

A 2-hop neighbor of node X is a node that is not directly connected to X, but is connected to one of X's direct neighbors.

3.5.3 Multipoint Relays (MPR)

Multipoint Relays (MPRs) are a smart technique used to reduce the number of broadcast messages in a wireless network. Normally, when a node sends out a broadcast message (like link information), all its neighbors may forward it again, which creates a lot of unnecessary traffic and wastes bandwidth. MPRs solve this by selecting only a few special neighbors to forward those messages.

Each node chooses some of its 1-hop neighbors as its MPRs. These MPRs are chosen in such a way that they can reach all the 2-hop neighbors. This means fewer nodes need to retransmit the message to spread it across the network.

The chosen MPR nodes will forward broadcast messages from that node.

The rest of the neighbors receive the message but do not retransmit it.

This reduces the number of duplicate messages in the network and saves battery and bandwidth.

Also, each node keeps track of which neighbors have selected it as their MPR, using the HELLO messages that are exchanged regularly.

3.5.4 Topology discovery

The link-sensing and neighbor-detection parts of the protocol provide each node with a list of neighbors it can directly communicate with. Combined with the Packet Format and Forwarding part, this enables an optimized flooding mechanism through MPRs. Using this, topology information is disseminated throughout the network.

3.5.5 TC Message Format

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|              ANSN             |           Reserved            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|               Advertised Neighbor Main Address                |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|               Advertised Neighbor Main Address                |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                             ...                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 3-4: *TC Message Format*

The TC message consists of the following fields:

ANSN (Advertised Neighbor Sequence Number):

- A 16-bit field used to indicate the freshness of the TC information.

Reserved:

- A 16-bit field reserved for future use; always set to 0.

Advertised Neighbor Main Addresses:

- These are one or more addresses of the neighboring nodes that the sender can reach and wants to inform others about. Usually, these are the nodes that have selected the sender as their MPR.

3.5.6 Core Functionality of OLSR

OLSR's core working is made up of several key parts:

Packet Format and Forwarding

All control messages (like HELLO, TC, etc.) follow a common packet format.

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|         Packet Length         |    Packet Sequence Number     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Message Type |     Vtime     |         Message Size          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                      Originator Address                       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Time To Live |   Hop Count   |    Message Sequence Number    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
:                            MESSAGE                            :
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Message Type |     Vtime     |         Message Size          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                      Originator Address                       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Time To Live |   Hop Count   |    Message Sequence Number    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
:                            MESSAGE                            :
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
:                                                               :
         (etc.)
```

Figure 3-5: *Packet Format*

OLSR uses a smart way to flood (send to all nodes) control messages across the network using a technique called MPR (MultiPoint Relay) to reduce unnecessary retransmissions.

3.5.7 Link Sensing

Each node sends HELLO messages at regular intervals through its network interfaces.

These messages help a node detect which of its neighbors are reachable (link sensing).

This creates a local link set—a list of links to neighboring nodes.

3.5.8 Neighbor detection

Neighbor detection populates the neighborhood information base and concerns itself with nodes and node main addresses. The mechanism for neighbor detection is the periodic exchange of HELLO messages.

A node maintains a set of neighbor tuples based on the link tuples. This information is updated according to changes in the Link Set. The Link Set keeps the information about the links, while the Neighbor Set keeps the information about the neighbors. There is a clear association between those two sets, since a node is a neighbor of another node if and only if there is at least one link between the two nodes.

The "Originator Address" of a HELLO message is the main address of the node that emitted the message. Upon receiving a HELLO message, a node should first update its Link Set and then update its Neighbor Set.

3.5.9 MPR Selection

Each node selects a few special neighbors called MPRs (MultiPoint Relays).

These MPRs are chosen in such a way that if they forward a message, it will reach all 2-hop neighbors (neighbors of neighbors).

This reduces the number of transmissions during flooding.

Nodes advertise their selected MPRs in their HELLO messages.

3.5.10 TC message

In order to build the topology information base, each node, which has been selected as MPR, broadcasts Topology Control (TC) messages. TC messages are flooded to all nodes in the network and take advantage of MPRs. MPRs enable a better scalability in the distribution of topology information.

The list of addresses can be partial in each TC message (e.g., due to message size limitations, imposed by the network), but parsing of all TC messages describing the advertised link set of a node MUST be complete within a certain refreshing period (TC INTERVAL). The information diffused in the network by these TC messages will help each node calculate its routing table.

When the advertised link set of a node becomes empty, this node SHOULD still send (empty) TC-messages during the duration equal to the "validity time" (typically, this will be equal to TOP HOLD TIME) of its previously emitted TC-messages, in order to invalidate the previous TC-messages. It SHOULD then stop sending TC-messages until some node is inserted in its advertised link set.

A node MAY transmit additional TC-messages to increase its reactiveness to link failures. When a change to the MPR selector set is detected and this change can be attributed to a link failure, a TC-message SHOULD be transmitted after an interval shorter than TC INTERVAL.

3.5.11 Route Calculation

After collecting enough link information through HELLO and TC messages, each node calculates the best route to all other nodes.

This is usually done using shortest-path algorithms like Dijkstra which finds the minimum-cost path from the source node to all other nodes based on the link metrics (e.g., hop count, ETX).

3.6 Optimized Link State Routing Protocol Version 2 (OLSRv2)

The Optimized Link State Routing Protocol Version 2 (OLSRv2) is a proactive link-state routing protocol designed specifically for Mobile Ad Hoc Networks (MANETs). It is the successor to the original OLSR protocol defined in RFC 3626 and introduces several improvements to enhance routing efficiency and scalability.

OLSRv2 retains the fundamental principles of OLSR while introducing major enhancements such as support for link metrics beyond hop count and simplified message structures.

3.6.1 Evolution from OLSR Version 1

OLSRv2 builds upon OLSR with the following key improvements:

- Support for advanced link metrics instead of simple hop count
- More flexible and efficient message encoding
- Reduced control message overhead

These improvements make OLSRv2 more suitable for dynamic and large-scale wireless networks.

Before understanding OLSRv2, it is essential to understand the protocol on which it is built, namely the Neighbor Discovery Protocol (NHDP). OLSRv2 does not perform neighbor discovery by itself; instead, it relies on NHDP to provide accurate and up-to-date information about neighboring nodes and link conditions.

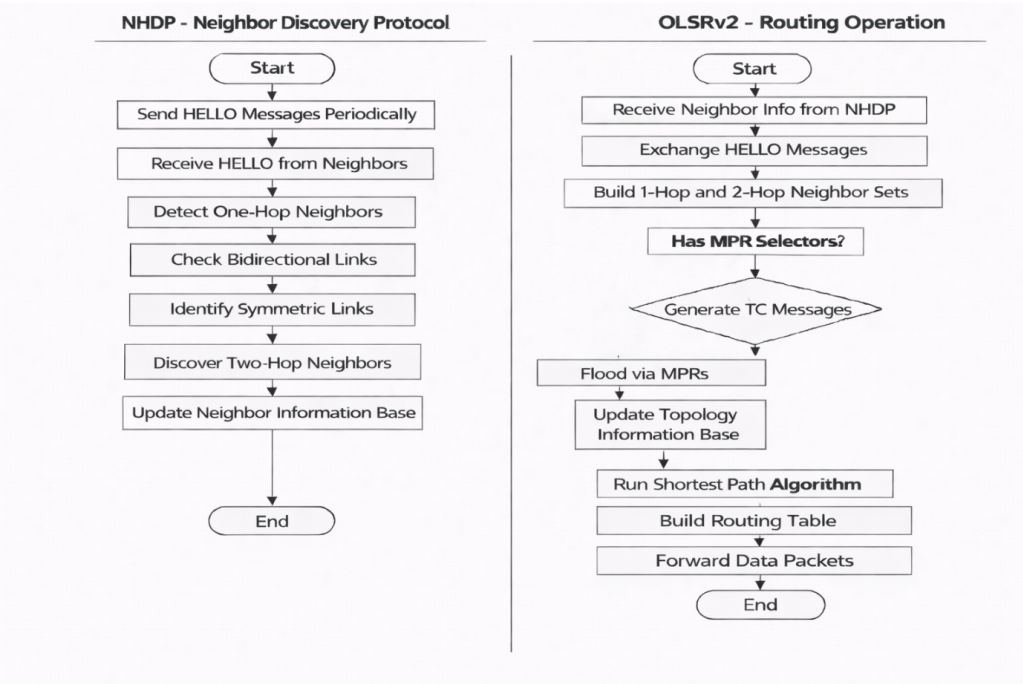


Figure 3-6: *NHDP and OLSRv2 Working Flow Diagram*

3.6.2 NHDP – Neighbor Discovery Protocol

Overview

The Neighbor Discovery Protocol (NHDP) is designed to discover and maintain information about neighboring nodes in a Mobile Ad Hoc Network. It operates through periodic exchange of local HELLO messages, allowing each node to determine the presence, connectivity, and link characteristics of its neighbors.

NHDP enables nodes to identify both:

- One-hop neighbors (directly reachable nodes)
- Symmetric two-hop neighbors (nodes reachable through one intermediate neighbor)

NHDP messages are formatted and transmitted using the standardized MANET packet format defined in RFC 5444.

Objectives of NHDP

The primary objective of NHDP is to maintain accurate and up-to-date information about local network connectivity in a dynamic wireless environment.

Specifically, NHDP performs the following functions:

- Detects directly reachable one-hop neighbors
- Identifies symmetric two-hop neighbors
- Verifies bidirectional communication links
- Tracks link establishment and failures
- Maintains updated neighbor information over time

NHDP Information Bases

NHDP maintains structured data repositories known as Information Bases, which store neighbor and link information. These databases provide essential inputs to MANET routing protocols such as OLSRv2 and are continuously updated based on received HELLO messages.

Link Quality and Hysteresis

NHDP uses a link quality mechanism to determine whether a link between two nodes should be considered usable or unreliable. Instead of immediately accepting or rejecting a link, the protocol evaluates the quality of the link using hysteresis thresholds.

The following interface parameters are:

HYST_ACCEPT – The link quality threshold at or above which a link becomes usable.

HYST_REJECT – The link quality threshold below which a link becomes unusable.

INITIAL_QUALITY – The initial quality value assigned to a newly identified link.

INITIAL_PENDING – If set to true, a newly identified link is considered pending and is not usable until the link quality reaches or exceeds the HYST_ACCEPT threshold.

Parameter Constraints:

$0 \leq \text{HYST_REJECT} \leq \text{HYST_ACCEPT} \leq 1$

$0 \leq \text{INITIAL_QUALITY} \leq 1$

If link quality is not updated, then $\text{INITIAL_QUALITY} \geq \text{HYST_ACCEPT}$.

If $\text{INITIAL_QUALITY} \geq \text{HYST_ACCEPT}$, then $\text{INITIAL_PENDING} = \text{false}$.

If $\text{INITIAL_QUALITY} < \text{HYST_REJECT}$, then $\text{INITIAL_PENDING} = \text{true}$.

One-Hop and Two-Hop Neighbor Discovery

NHDP discovers neighboring nodes through periodic HELLO message exchanges.

- When a node receives a HELLO message, it records the sender as a one-hop neighbor.
- Initially, the link is considered unidirectional.
- The link becomes symmetric when both nodes report each other.

Two-hop neighbors are discovered indirectly through neighbor information included in HELLO messages. When a node receives a HELLO message containing the addresses of another node's neighbors, it identifies those nodes as two-hop neighbors reachable through the intermediate node.

Thus, NHDP continuously maintains updated information about one-hop and two-hop neighboring nodes, enabling routing protocols to accurately understand local network topology.

Generalized MANET Packet Format

The generalized MANET packet format defines a standardized method for exchanging routing information in MANETs. It allows multiple protocol messages to be carried within a single packet transmission, improving efficiency.

The format uses:

- Address Blocks for compact representation
- Type-Length-Value (TLV) structures for flexibility

3.6.3 Message Structure

A message consists of two main parts:

- Message Header
- Message Body

The Message Header contains control information, while the Message Body carries protocol-specific data.

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Message Type  |1|1|1|1|  MAL  |         Message Size          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                     Originator Address                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   Hop Limit   |   Hop Count   |    Message Sequence Number    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|                       Message Body                            |
|                                                               |
|                               +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 3-7: *Generalized MANET Message Structure*

3.6.4 Message Header Fields

The Message Header includes:

- Message Type
- Message Size
- Originator Address
- Hop Limit
- Hop Count
- Message Sequence Number

3.6.5 Address Blocks and TLVs

Address Blocks allow multiple addresses to be represented efficiently. Each Address Block may include TLVs describing link status, link quality, and neighbor attributes.

3.6.6 Message Body Structure

The Message Body begins with a Message TLV Block followed by pairs of Address Blocks and Address TLV Blocks containing routing information.

3.7 OLSRv2 – Protocol

OLSRv2 operates by maintaining network topology information to proactively compute routes to all known destinations.

Its operation is based on two frameworks:

Neighbor Discovery using NHDP

Message Formatting using RFC 5444

3.7.1 Protocol Overview

The main efficiency of OLSRv2 comes from Multipoint Relays (MPRs).

Types of MPRs

Flooding MPRs: A node selects a subset of its symmetric one-hop neighbors as Flooding MPRs. These are the only neighbors allowed to retransmit its broadcast control messages. This creates an efficient, reduced flooding mechanism that minimizes overhead while ensuring messages are disseminated throughout the network.

Routing MPRs (MPR Selectors): A node maintains a set of neighbors that have selected it as an MPR. These are known as its MPR selectors. In OLSRv2, only the links to these MPR selectors are advertised by a node in its Topology Control (TC) messages. This is a second major optimization, as it significantly reduces the amount of topological information that must be flooded.

3.7.2 Control Messages

OLSRv2 uses two primary control messages:

HELLO Messages

Exchanged periodically and locally between neighboring nodes. They are responsible for link sensing, neighbor discovery, and signaling MPR selections. The information obtained from HELLO messages populates the Interface Information Base and Neighbor Information Base.

Topology Control (TC) Messages

Generated periodically only by nodes that have at least one MPR selector (i.e., nodes that have been selected as an MPR by one or more neighbors). These messages are flooded throughout the entire network using Flooding MPRs. TC messages contain the node's MPR selector set, effectively declaring the links that form the network topology. The information from TC messages populates the Topology Information Base.

OLSRv2 Information Bases

OLSRv2 maintains the following information bases:

- Local Information Base
- Interface Information Base
- Neighbor Information Base
- Topology Information Base
- Routing Set
- Received Message Information Base

3.7.3 Control Message Details

HELLO Messages

HELLO messages are generated periodically and include:

- Local addresses
- Neighbor status information
- Link metrics

TC Messages

TC messages are generated by nodes having at least one MPR selector. They contain:

- Originator address
- Advertised neighbor set
- Validity time

MPR Selection

Selecting MPRs

The goal is to select a minimal set of one-hop neighbors such that all symmetric two-hop neighbors are reachable.

Selection depends on:

- Link symmetry
- Willingness
- Link metrics

3.7.4 Configuring OLSRv2 in NetSim

Selecting OLSRv2 as Routing Protocol

To enable OLSRv2 in NetSim:

1. Open the network scenario in NetSim.
2. Right-click on a wireless node.
3. Select Properties to open the Node Properties window.
4. Navigate to the Network Layer section.
5. Locate the Routing Protocol dropdown.
6. Select OLSRv2.

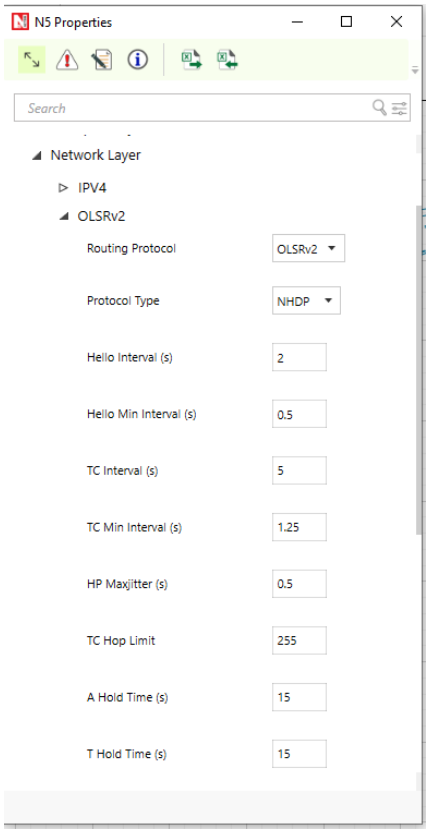


Figure 3-8: *Selecting OLSRv2 in Node Properties*

Configuring OLSRv2 Parameters

After selecting OLSRv2 as the routing protocol, several protocol parameters can be configured in NetSim to control neighbor discovery, topology dissemination, and link quality evaluation.

HELLO Message Parameters

Hello Interval (Default: 2 seconds)
Defines the time interval between consecutive HELLO message transmissions. A smaller value allows faster neighbor detection but increases control overhead.

Hello Minimum Interval – hello_min_interval (Default: 0.5 second)
Specifies the minimum allowed time between two HELLO message transmissions.

Maximum HELLO Jitter – HP_Maxjitter (Default: 0.5 seconds)
Introduces a small random delay before transmitting HELLO messages to avoid synchronization of transmissions among nodes.

Link Quality and Hysteresis Parameters

Link Quality Mode (Default: Disabled / Hop-count based routing)
Enables the use of link quality metrics instead of hop-count metrics.

Initial Quality – INITIAL_QUALITY (Default: 1.0)
Defines the starting quality value assigned to a newly detected link before sufficient observations are made.

Hyst Accept – hyst_accept (Default: 0.8)
Threshold above which the link becomes usable.

Hyst Reject – hyst_reject (Default: 0.3)
Threshold below which the link becomes unusable.

Initial Pending – INITIAL_PENDING (Default: false)
Represents the initial pending state used in link-quality calculations before the link state becomes stable.

Topology Control (TC) Parameters

TC Interval – tc_interval (Default: 5 seconds)
Defines how frequently Topology Control messages are generated.

TC Minimum Interval – tc_min_interval (Default: 1.25 seconds)
Specifies the minimum time between two TC transmissions.

Maximum TC Jitter – TP_maxjitter (Default: 0.5 seconds)
Random delay applied before TC message transmission to reduce collisions.

TC Hop Limit – tc_hop_limit (Default: 255)
Defines the maximum number of hops a TC message can travel in the network.

Information Base Hold Times

Topology Hold Time – T_hold_time (Default: 15 seconds)
Specifies how long topology information received from TC messages remains valid.

Advertised Neighbor Hold Time – A_hold_time (Default: 15 seconds)
Defines how long advertised neighbor information is retained in the topology database.

MPR and Routing Parameters

Willingness (Flooding and Routing Willingness) (Default: WILL_DEFAULT = 7)
Indicates a node's readiness to act as a Multipoint Relay (MPR). Higher willingness increases the probability of being selected as an MPR.

Configuring Wireless Link Properties

Since OLSRv2 relies on neighbor discovery, configuring wireless range is essential.

Steps:

1. Right-click on the wireless link.
2. Select Properties.
3. Under Medium Property, configure:
    - Channel Characteristics: Pathloss
    - Pathloss Model: Range Based
    - Range (m): Set as required

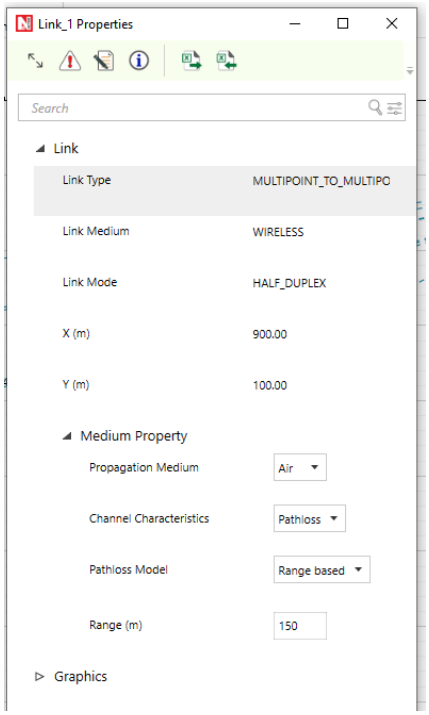


Figure 3-9: *Wireless Link Configuration for OLSRv2*

3.8 Energy model in MANETs

A Mobile Ad-hoc Network (MANET) consists of self-organizing wireless mobile nodes deployed without a centralized infrastructure. Nodes communicate with each other using a limited battery supply. NetSim supports DSR, AODV, ZRP, and OLSR as the network layer protocols, considering transmission power and residual battery power. This work aims to increase the network lifetime while reducing energy consumption and minimizing end-to-end delay.

Energy is a very important part in ad-hoc networks because all nodes in this network communicate with each other, consuming power in the process. The primary source of power is the battery of the mobile node. A node consumes energy in various modes, including sending/transmitting mode, receiving mode, and idle mode.

Transmitting mode: In this Tx mode, the node consumes energy to transfer packets or data. The amount of energy consumed in this mode depends on the number of packets sent by the node; the greater the number of packets, the more energy is consumed.

Receiving mode: In the Rx mode, the node actively listens for incoming data, consuming energy as it receives data from the sender.

Idle mode: In idle mode, a node neither transmits nor receives data packets. However, it still needs to listen to the wireless medium for potential packet reception and check for the addition of any new nodes to the network. As a result, the energy consumed in idle mode is less than that consumed in transmitting or receiving mode, since no actual communication occurs during idle mode.

In NetSim, default energy model settings are as follows:

- Transmitting current (mA): 316.7
- Receiving current (mA): 261.1
- Idle mode current (mA): 227
- Voltage (V): 3.6

The power model is disabled by default. It can be enabled in physical layer of wireless node Omni Ant.

Click on a wireless node's properties to open the right-side property panel, then navigate to Interface 1(Wireless) ▷ Physical layer.

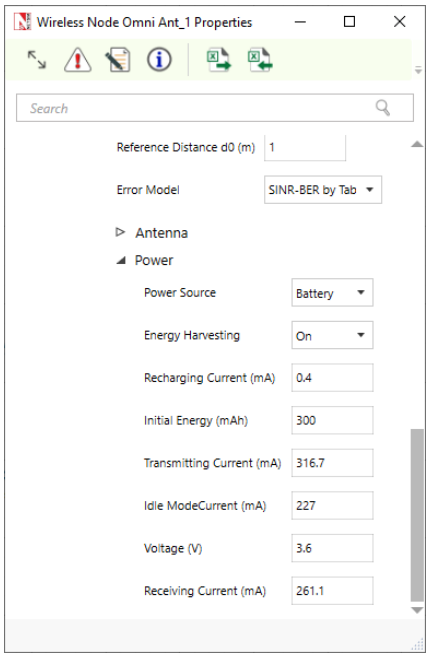


Figure 3-10: *Power Model in MANET.*

3.9 Mobility models in NetSim

The change in position of a user over time can be configured using mobility models. Such models are frequently used in simulation to evaluate the performance of mobile wireless systems. The mobility models available in NetSim are:

3.9.1 Random Walk Mobility Model

This is a simple mobility model based on random directions and speeds. In this mobility model, a mobile node moves from its current location to a new location by randomly choosing a direction. The speed can be set in the GUI. The calculation interval determines how often a new choice is made. For example, a 1s calculation interval means that a new position is chosen every 1s.

3.9.2 Random Waypoint Mobility Model

This model includes pause time between changes in direction and/or speed. A mobile node begins by staying in one location for a certain period of time (i.e., a pause time). Once this time expires, the mobile node chooses a random destination in the simulation area. The mobile node then travels toward the newly chosen destination at the selected speed. Upon arrival, the mobile node pauses for a specified PauseTime before starting the process again. Note that the movement calculation is done every Calculation Interval. Therefore, it is recommended to set the PauseTime as an integral multiple of Calculation Interval.

3.9.3 Group Mobility Model

This model describes the behavior of mobile nodes as they move together, i.e., the devices having common group id will move together.

3.9.4 Pedestrian Mobility Model

This model is applicable to each node (local parameter), and the configuration parameters are:

- Pedestrian Max Speed (m/s) (Range: 0.0 to 10.0. Default: 3.0)
- Pedestrian Min Speed (m/s) (Range: 0.0 to 10.0. Default: 1.0)
- Pedestrian stop probability (Range: 0 to 1)
- Pedestrian stop duration (s). (Range: 1 to 10000)

In this model it is assumed that the pedestrian stops at traffic lights. The stop probability represents the probability of encountering a traffic light. This is checked for every calculation interval. Once stopped, the pedestrian waits for a duration equal to stop duration for the light to turn green. A new direction is chosen randomly after every stop with θ (angle between new direction and current direction) taking values of 0, 90, 180, 270. These θ values represent the pedestrians continuing in the same direction, taking a left, making a U-turn, and taking a right, respectively.

A new speed is chosen randomly after event stop. Min speed $\leq$ Speed $\leq$ Max speed

The maximum number of stops and starts is 10.

3.9.5 File Based Mobility Model

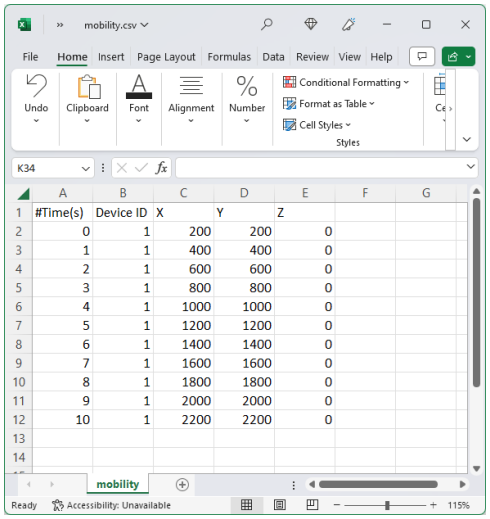
In the File Based Mobility model, users can write their own custom mobility models and define the movement of the mobile users. The content should be per the NetSim Mobility File Format explained below. Users can also generate the mobility files using external tools like SUMO (Simulation of Urban Mobility), VanetMobiSim etc.

The NetSim Mobility File setting and format is explained below:

Step 1: Right click to open Properties as a new window of Wireless Node (MANET) or UE (LTE Networks). Then properties ▷ Position Properties ▷ Mobility model as File Based Mobility and click on Open Mobility file.

Step 2: Open the CSV file and write the node mobility in the format shown below.

```
<TIME IN SECONDS>,<DEVICE ID>,X,Y,Z
```



#Time(s)	Device ID	X	Y	Z
0	1	200	200	0
1	1	400	400	0
2	1	600	600	0
3	1	800	800	0
4	1	1000	1000	0
5	1	1200	1200	0
6	1	1400	1400	0
7	1	1600	1600	0
8	1	1800	1800	0
9	1	2000	2000	0
10	1	2200	2200	0

Figure 3-11: *Mobility.csv file with inputs added via MS Excel*

When using spreadsheet software commas (,) are included automatically between the entries in each column. When using text editors such as notepad, users will have to manually include comma (,) between each entry.

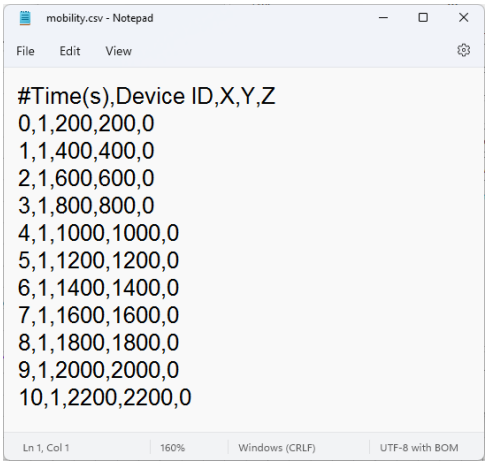


Figure 3-12: *Mobility.csv file with comma separated inputs added via text editor (notepad)*

To write comments, use # tag

Specify the new location of the specific device at the specific time

Simulations with older input format in mobility.txt file (as shown below) will continue to be supported, in the absence of mobility.csv file.

`$time <Time in seconds> ‘‘$node_(<NodeID - 1>) <X Coordinate> <Y Coordinate> <Z Coordinate>’’`

Example: `$time 1 ‘‘$node_(0) 400 400 0’’`

A sample file-based mobility example is present at `<NetSim Installed Directory>\Docs\ Sample Configuration\ MANET`.

It can have the coordinates of one node over time, followed by the coordinates of the next node over time and so on. However, for a given device, the entries in the file need to be in increasing order of time.

NetSim does not interpolate device mobility between two entries in the mobility file. This means that during the simulation the device “jumps” to the next point at that instant in time. Mobility is not “continuous” from one point to another over time. Continuous or fine-grained mobility can, however, be approximated by reducing the time interval between successive entries. Since Excel-based mobility files are supported, generating such fine-grained traces is straightforward.

3.10 Omni and sector (directional) antennas

NetSim MANET supports Omni and sector antennas. These antenna patterns are defined by gain values that vary as a function of direction in three dimensions. Thus, a transmitter’s antenna provides a gain value with respect to every other point in space. The same is true of a receiver’s antenna. For a given transmission, the gains of both the transmitter and receiver, with respect to each other, are considered.

Omni directional antennas have an isotropic pattern i.e., the gain in all directions is 0 dB. This makes them well-suited for scenarios where uniform coverage is required in all directions.

Sector antennas are modeled per the standard 2D parabolic antenna. The key characteristic of a sector antenna is its horizontal radiation pattern, which is designed to focus the signal in a specific direction. This directional pattern allows for targeted coverage, making sector antennas ideal for applications that require directed communication over a particular sector of space. The horizontal radiation pattern of a sector antenna is given by

$$A_{E,H}\left(\varphi\right) = -\min\left(12\left(\frac{\varphi}{\varphi_{3\mathrm{dB}}}\right)^2,\ A_m\right) \tag{1}$$

where,

- A_m is the front to back ratio (i.e., the ratio of power gained between the front and rear of a directional antenna). This is a user input with a default setting of 30 dB.
- φ is the angle formed between the direction of interest (i.e., the line connecting the transmitter and the receiver) and the antenna's boresight direction (azimuth angle).
- $\varphi_{3\text{dB}}$ is the 3 dB, or half power, beam width of the antenna.

The sector antenna gain is given by the expression.

$$A_E(\varphi, \theta) = G_{E,\max} - \min\{-[A_{E,H}(\varphi) + A_{E,V}(\theta)],\ A_m\} \tag{2}$$

where,

- $A_{E,V}(\theta)$ is currently 0 dB, and is a parameter reserved for future use to include the vertical radiation pattern of the antenna, and
- $G_{E,\max}$ is the maximum directional gain of the radiation element (in dB) i.e., the gain along the antenna boresight; this parameter is a user input, and the default value is 8 dBi.

The boresight angle denotes the azimuthal direction of maximum gain, or the highest radiated power, and is a user input. This can vary from 0 to 360 degrees.

Angle in NetSim is defined to start at 0 from the positive X-axis. In the network scenario, if positive Y points downward, the angle increases on clockwise rotation from the positive X-axis. If positive Y points upward, the angle increases in an anti-clockwise direction from the positive X-axis. The unit for angle is degrees.

The gain computations for a transmitter-receiver pair are given below.

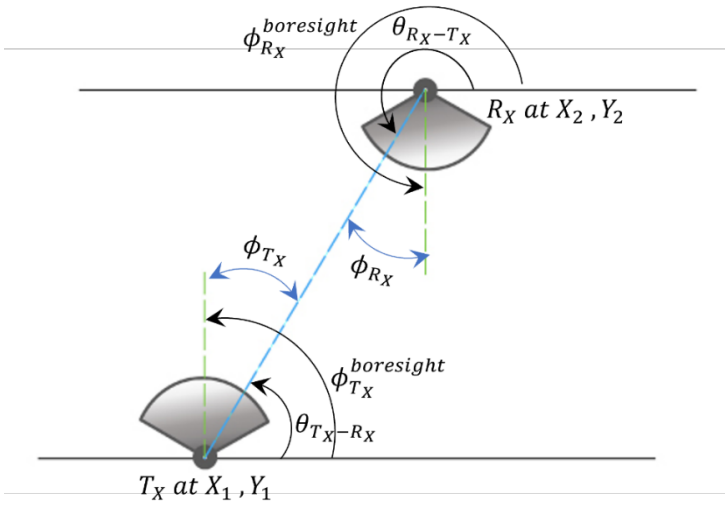


Figure 3-13: *The illustration shows a transmitter at X_1, Y_1 and a receiver at X_2, Y_2. The green lines represent the boresight azimuth direction of each antenna. The blue line is the line joining the transmitter and the receiver. ϕ_{tx} (and ϕ_{Rx}) are the angles formed between the line connecting the transmitter and the receiver and the transmit antennas (and receive antennas respectively) boresight direction.*

Let the transmitter be at X_1, Y_1 and the receiver at X_2, Y_2. Let $\phi_{T_X}^{\text{boresight}}$ and $\phi_{R_X}^{\text{boresight}}$ be the boresight azimuth angles of the transmitter and the receiver configured by the user through the GUI.

We know that,

$$\theta_{T_X-R_X} = \tan^{-1}\left(\frac{y_2 - y_1}{x_2 - x_1}\right) \tag{3}$$

and clearly,

$$\theta_{R_X-T_X} = \pi + \theta_{T_X-R_X} \tag{4}$$

$$\phi_{T_X} = \phi_{T_X}^{\text{boresight}} - \theta_{T_X - R_X} \tag{5}$$

$$\phi_{R_X} = \phi_{R_X}^{\text{boresight}} - \theta_{R_X - T_X} \tag{6}$$

The sign of ϕ_{T_X} or ϕ_{R_X} doesn't matter since the square of the term is used in the gain calculations.

$$A_{T_X}(\phi) = -\min\left[12\left(\frac{\phi_{T_X}}{\varphi_{3\text{dB}}}\right)^2,\ A_m\right] \tag{7}$$

$$A_{R_X}(\phi) = -\min\left[12\left(\frac{\phi_{R_X}}{\varphi_{3\text{dB}}}\right)^2,\ A_m\right] \tag{8}$$

$$T_X \text{ Gain} = G_{E,\max} - \min[-A_{T_X}(\phi),\ A_m] \tag{9}$$

$$R_X \text{ Gain} = G_{E,\max} - \min[-A_{R_X}(\phi),\ A_m] \tag{10}$$

Enabling the radio measurements log allows for the visualization of transmit and receive antenna gains for each transmission.

3.10.1 Limitations

- The vertical radiation pattern is not currently supported. Hence the horizontal radiation pattern is assumed to be valid for all values of the coordinate z.

3.11 How are packet collisions modelled in NetSim?

Packet Collision is a theoretical concept. In the real world, a packet is not a "physical" entity that can collide. Therefore, a "collision" is the interference of two radio waves (of the packets being sent).

In NetSim the collision model is as follows:

- At the transmitter, when a packet is being transmitted, if the sum of received power from all other nodes is greater than the receiver sensitivity then that packet is marked as collided.
- At the receiver, when a packet is being received, if the sum of the received power from all other nodes, except the node from where this packet is being transmitted, is greater than the receiver sensitivity then that packet is marked as collided.
- There is also a special case where a single packet can be marked as collided as explained in Figure 3-3.

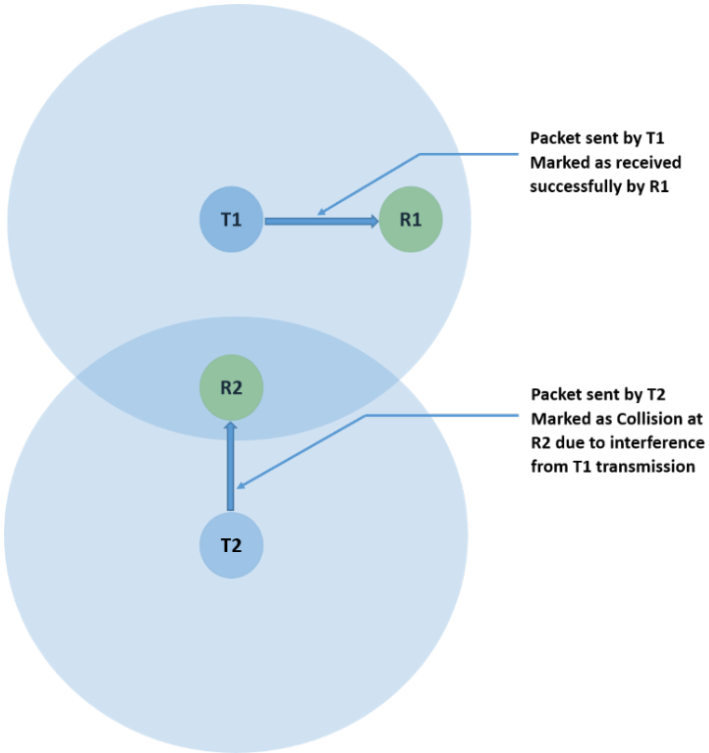


Figure 3-14: *Packet collisions modelled.*

Where,

- T1 and T2 are transmitters.
- R1 and R2 are receivers.

3.12 How is packet error decided in NetSim

The way the packet error is decided is given as follows:

- Compute Rx power at the receiver based on Tx Power, fewer losses due to Propagation. Propagation models cover HATA Urban, Sub Urban, COST 231 HATA, Indoor Home / Office, Factory, Lognormal, Free space. Apart from path loss, fading and shadowing can also be modeled.
- Rx Power / (Background Noise + Interference noise) = SINR. Here Interference noise is due to other transmissions within the network.
- Based on the SINR, BER is got from the respective modulation curve/table.
- Based on the BER, the packet error rate is decided.
- NetSim does not perform bitwise simulation since it would be too computationally intensive.

3.13 Broadcast in MANETs

In MANET, only neighbor (nodes within transmission range) broadcast is supported; multi-hop broadcast is not supported. This means that nodes that are 2-hops or more away will not receive the broadcast packets. The reason for not implementing a multi-hop broadcast is explained below.

The simplest form of broadcast in an ad hoc network is where a node transmits a broadcast packet, which is received by all neighboring nodes that are within transmission range. Upon receiving a broadcast packet, each node determines if it has transmitted the packet before. If not, then the packet is retransmitted.

This process allows for a broadcast packet to be disseminated throughout the ad hoc network. The process terminates when all nodes have received and transmitted the packet being broadcast at least once.

This method of broadcasting is required given the dynamic nature of ad hoc networks, i.e., nodes move around, and neighbors may change as the broadcast is happening. Now, since all nodes participate in the broadcast, it causes flooding and hence suffers from the Broadcast Storm Problem. Flooding is extremely costly and leaves the network completely congested.

Special algorithms are required to alleviate the broadcast storm problem, and custom development is necessary to implement them.

3.14 Radio measurement log file and plots

NetSim IEEE 802.11 Radio Measurements log file records pathloss, shadowing loss, fading loss, received power, transmitted power, SNR, and BER. This log can be enabled by clicking on configure reports in top ribbon ▷ Plots ▷ Select IEEE 802.11 Radio Measurements log under Network Logs as shown below

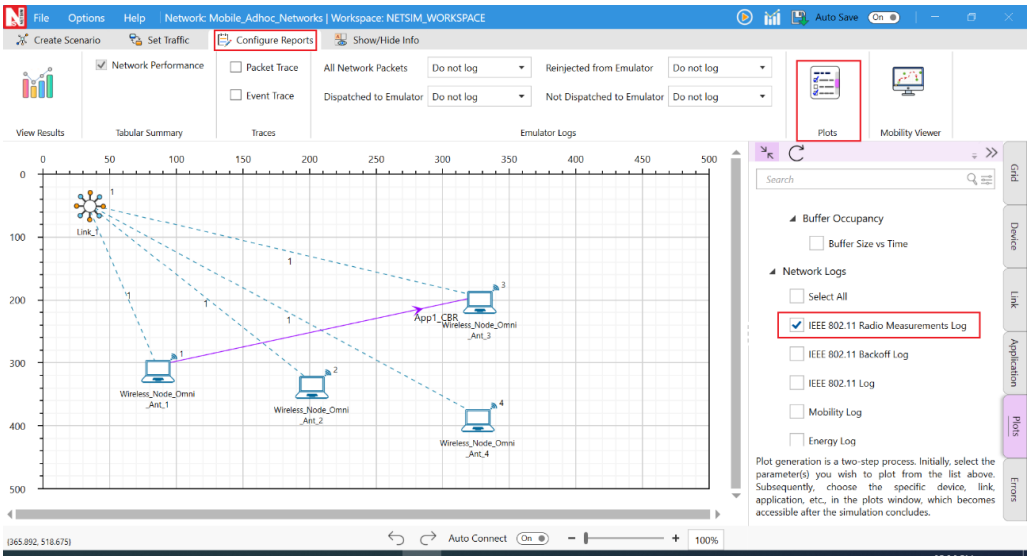


Figure 3-15: *Enabling the IEEE 802.11 Radio Measurements log*

The IEEE802.11 Radio Measurements log.csv file will contain the following information:

- Time in Milliseconds
- Transmitter Name
- Receiver Name
- Distance between the Transmitter and the Receiver in meters
- Packet ID
- Packet Type
- Control Packet Type
- Transmitter Power in dBm
- Pathloss in dB
- Shadowing Loss in dB
- Fading Loss in dB
- Total Loss in dB
- Transmitter Antenna Gain in dB

- Receiver Antenna Gain in dB
- Received Power in dBm
- Interference in dBm
- SNR in dB
- SINR in dB
- BER
- NSS (Number of Spatial Streams)
- MCS Index
- The log file can be accessed from the Simulations Results Window under the log as shown below.

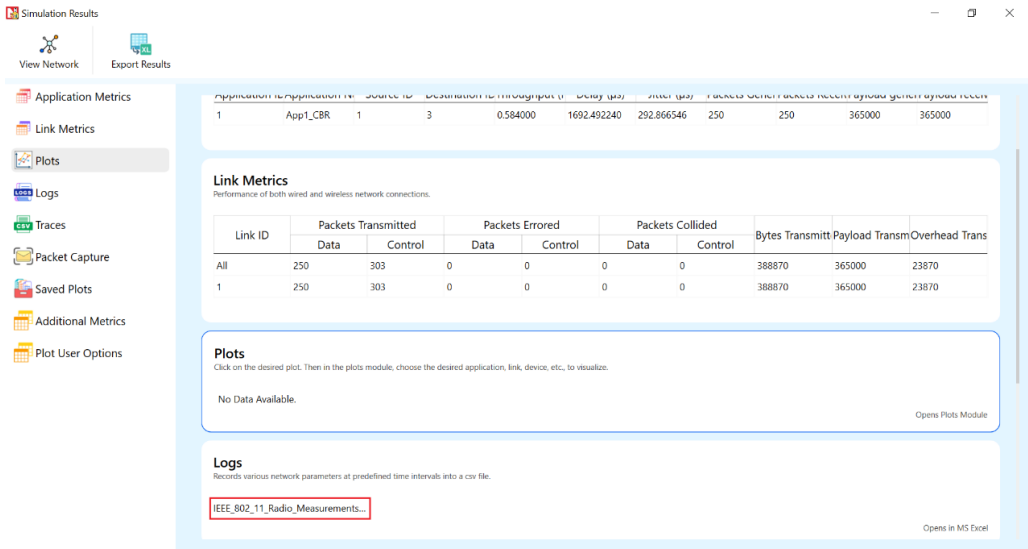
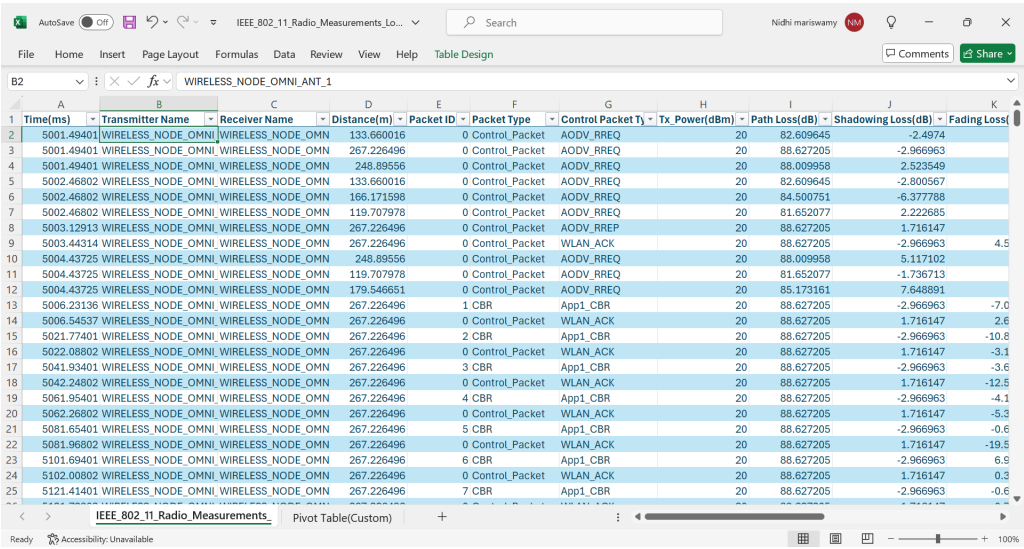


Figure 3-16: *Opening IEEE 802.11 Radio Measurements log from simulation results window*



Time(ms)	Transmitter Name	Receiver Name	Distance(m)	Packet ID	Packet Type	Control Packet Ty	Tx_Power(dBm)	Path Loss(dB)	Shadowing Loss(dB)	Fading Loss(
5001.49401	WIRELESS_NODE_OMNI	WIRELESS_NODE_OMN	133.660016	0	Control_Packet	AODV_RREQ	20	82.609645	-2.4974	
5001.49401	WIRELESS_NODE_OMNI	WIRELESS_NODE_OMN	267.226496	0	Control_Packet	AODV_RREQ	20	88.627205	-2.966963	
5001.49401	WIRELESS_NODE_OMNI	WIRELESS_NODE_OMN	248.89556	0	Control_Packet	AODV_RREQ	20	88.009958	2.523549	
5002.46802	WIRELESS_NODE_OMNI	WIRELESS_NODE_OMN	133.660016	0	Control_Packet	AODV_RREQ	20	82.609645	-2.800567	
5002.46802	WIRELESS_NODE_OMNI	WIRELESS_NODE_OMN	166.171598	0	Control_Packet	AODV_RREQ	20	84.500751	-6.377788	
5002.46802	WIRELESS_NODE_OMNI	WIRELESS_NODE_OMN	119.707978	0	Control_Packet	AODV_RREQ	20	81.652077	2.222685	
5003.12913	WIRELESS_NODE_OMNI	WIRELESS_NODE_OMN	267.226496	0	Control_Packet	AODV_RREP	20	88.627205	1.716147	
5003.44314	WIRELESS_NODE_OMNI	WIRELESS_NODE_OMN	267.226496	0	Control_Packet	WLAN_ACK	20	88.627205	-2.966963	4.5
5004.43725	WIRELESS_NODE_OMNI	WIRELESS_NODE_OMN	248.89556	0	Control_Packet	AODV_RREQ	20	88.009958	5.117102	
5004.43725	WIRELESS_NODE_OMNI	WIRELESS_NODE_OMN	119.707978	0	Control_Packet	AODV_RREQ	20	81.652077	-1.736713	
5004.43725	WIRELESS_NODE_OMNI	WIRELESS_NODE_OMN	179.546651	0	Control_Packet	AODV_RREQ	20	85.173161	7.648891	
5006.23136	WIRELESS_NODE_OMNI	WIRELESS_NODE_OMN	267.226496	1	CBR	App1_CBR	20	88.627205	-2.966963	-7.0
5006.54537	WIRELESS_NODE_OMNI	WIRELESS_NODE_OMN	267.226496	0	Control_Packet	WLAN_ACK	20	88.627205	1.716147	2.6
5021.77401	WIRELESS_NODE_OMNI	WIRELESS_NODE_OMN	267.226496	2	CBR	App1_CBR	20	88.627205	-2.966963	-10.8
5022.08802	WIRELESS_NODE_OMNI	WIRELESS_NODE_OMN	267.226496	0	Control_Packet	WLAN_ACK	20	88.627205	1.716147	-3.1
5041.93401	WIRELESS_NODE_OMNI	WIRELESS_NODE_OMN	267.226496	3	CBR	App1_CBR	20	88.627205	-2.966963	-3.6
5042.24802	WIRELESS_NODE_OMNI	WIRELESS_NODE_OMN	267.226496	0	Control_Packet	WLAN_ACK	20	88.627205	1.716147	-12.5
5061.95401	WIRELESS_NODE_OMNI	WIRELESS_NODE_OMN	267.226496	4	CBR	App1_CBR	20	88.627205	-2.966963	-4.1
5062.26802	WIRELESS_NODE_OMNI	WIRELESS_NODE_OMN	267.226496	0	Control_Packet	WLAN_ACK	20	88.627205	1.716147	-5.3
5081.65401	WIRELESS_NODE_OMNI	WIRELESS_NODE_OMN	267.226496	5	CBR	App1_CBR	20	88.627205	-2.966963	-0.6
5081.96802	WIRELESS_NODE_OMNI	WIRELESS_NODE_OMN	267.226496	0	Control_Packet	WLAN_ACK	20	88.627205	1.716147	-19.5
5101.69401	WIRELESS_NODE_OMNI	WIRELESS_NODE_OMN	267.226496	6	CBR	App1_CBR	20	88.627205	-2.966963	6.9
5102.00802	WIRELESS_NODE_OMNI	WIRELESS_NODE_OMN	267.226496	0	Control_Packet	WLAN_ACK	20	88.627205	1.716147	0.3
5121.41401	WIRELESS_NODE_OMNI	WIRELESS_NODE_OMN	267.226496	7	CBR	App1_CBR	20	88.627205	-2.966963	-0.6

Figure 3-17: *IEEE 802.11 Radio Measurements log file*

NetSim plots

NetSim also provides energy and Wi-Fi radio measurement plots for MANET. Enabling the plots is explained in the Section 2.6. Here is the plot showing the transmit energy vs. time for an AODV scenario with five nodes.

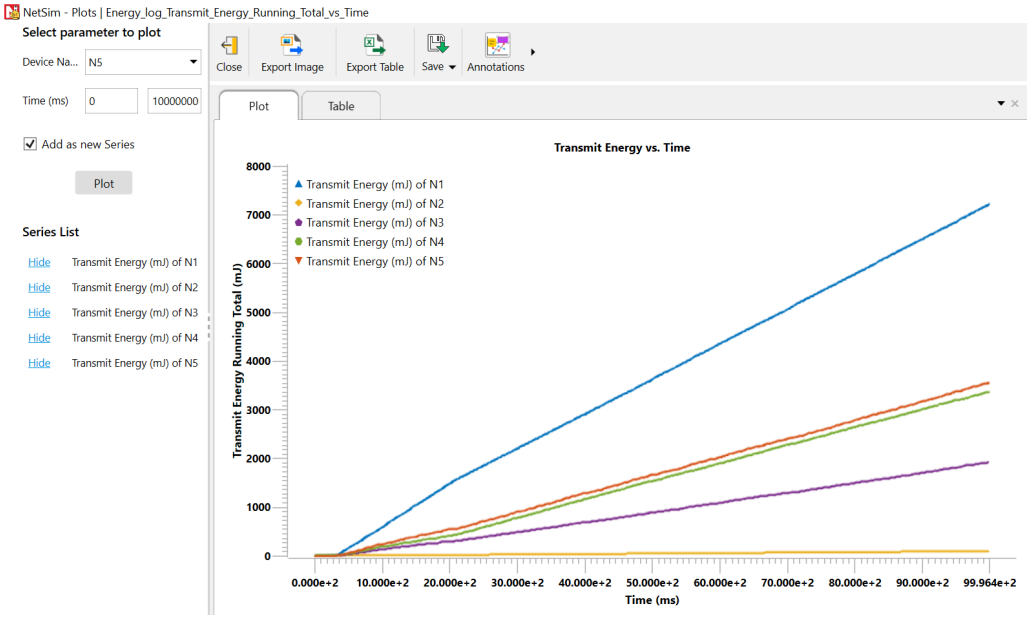


Figure 3-18: *Plot of Transmit Energy vs Time for MANET*

The Wi-Fi Radio measurement plots are shown in featured example in Section 4.9.

4 Featured Examples

Sample configuration files for all networks are available in Examples Menu in NetSim Home Screen. These files provide examples on how NetSim can be used – the parameters that can be changed and the typical effect it has on performance.

4.1 AODV Routing

The Ad hoc On Demand Distance Vector Routing (AODV) protocol defines 3 message types:

- Route Requests (RREQs): RREQ messages are used to initiate the route-finding process.
- Route Replies (RREPs): RREP messages are used to finalize the routes.
- Route Errors (RERRs): RERR messages are used to notify the network of a link breakage in an active route.

The route discovery process involves ROUTE REQUEST (RREQ) and ROUTE REPLY (RREP) packets. The source node initiates the route discovery process using RREQ packets. The generated route request is forwarded to the neighbors of the source node and this process is repeated till it reaches the destination. On receiving an RREQ packet, an intermediate node with route to destination or the destination node generates an RREP containing the number of hops required to reach the destination. All intermediate nodes that participate in relaying this reply to the source node create a forward route to destination.

RERR message processing is initiated when Node detects a link break for the next hop of an active route or receives a data packet destined for a node for which it has no (active) route.

Open NetSim and Select Examples→Mobile Ad hoc Networks→MANET AODV then click on the tile in the middle panel to load the example as shown in below screenshot Figure 4-1.

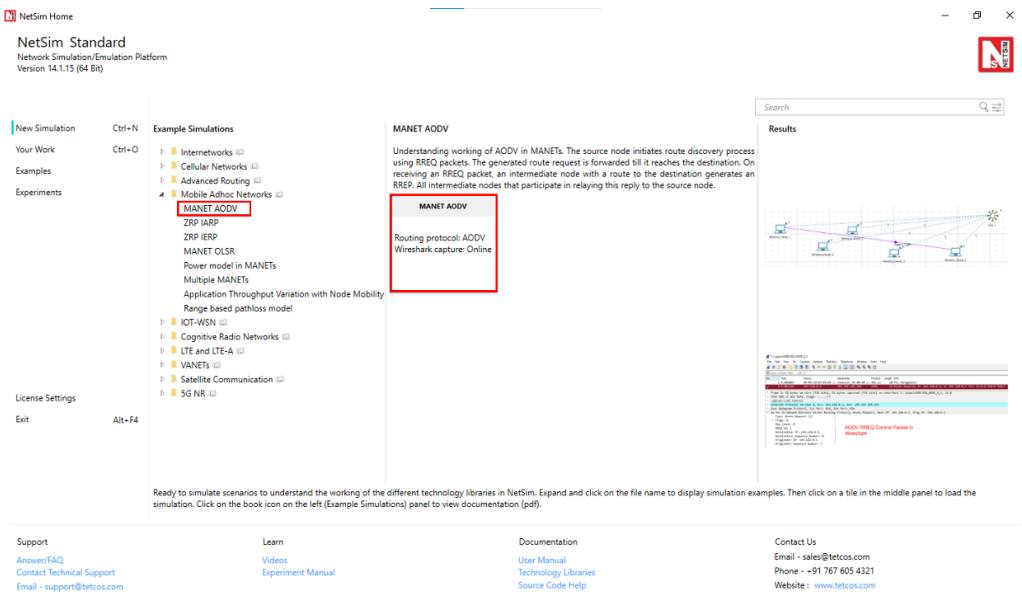


Figure 4-1: *List of scenarios for the example of MANET AODV*

The following network diagram illustrates, what the NetSim UI displays when you open the example configuration file see Figure 4-2.

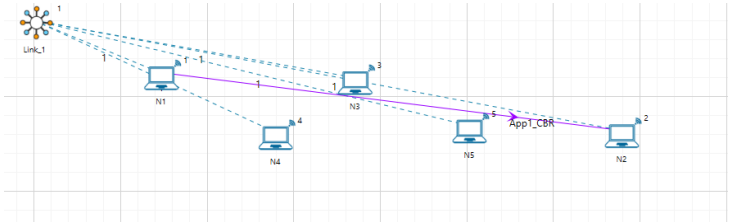
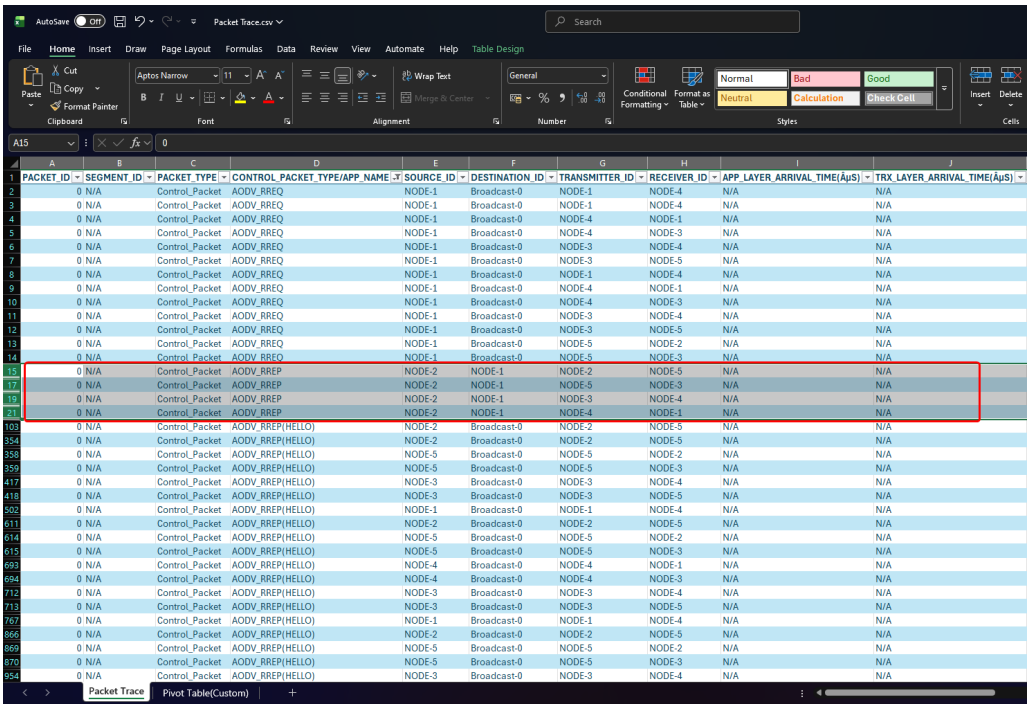


Figure 4-2: *Network set up for studying the MANET AODV*

4.1.1 Network Settings

- A network scenario is designed in NetSim GUI, consisting of 5 wireless nodes omni ants, named N1, N2, N3, N4, and N5, and one link in the "Standard MANET" of the "Mobile Ad hoc Network" library.
- In the general properties of N3, Wireshark Capture is set to Online. To configure any properties in the nodes, click on the node, expand the property panel on the right side, and change the properties as mentioned in the steps.
- In the position properties, set the mobility model as No-Mobility for all devices present in the GUI.
- The Medium Access Protocol is set to DCF in Interface 1(Wireless)→ Datalink layer of all the devices.
- Click on link and expand property panel on the right, and set the Channel Characteristics: Path loss only; Path loss model: Log Distance; Path loss exponent: 3.1.
- The routing protocol in the network layer is set to AODV for all nodes. As this is a global property, changing it on one node will affect all nodes.
- Packet trace is enabled by clicking on the configure report tab, allowing us to track the route that the packets have taken to reach the destination based on AODV.
- Configure the CBR application between two nodes by clicking on the set traffic tab from the ribbon at the top, selecting N1 as the source and N2 as the destination. Click on the created application and set the transport protocol to UDP, keeping the other properties as default.

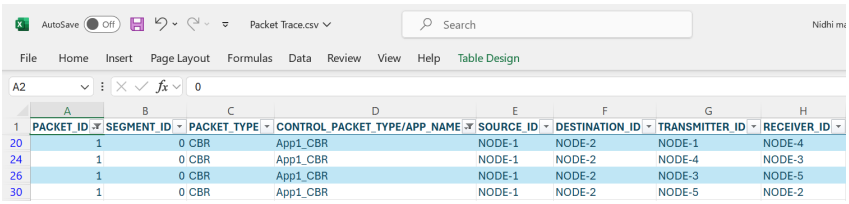
- Run simulation for 10 seconds.
- Open packet trace from simulation results window.
- The packet flow can be observed by filtering Control Packet Type/App Name to AODV RREQ, AODV RREP and AODV RREP (Hello) as shown below.



PACKET_ID	SEGMENT_ID	PACKET_TYPE	CONTROL_PACKET_TYPE/APP_NAME	SOURCE_ID	DESTINATION_ID	TRANSMITTER_ID	RECEIVER_ID	APP_LAYER_ARRIVAL_TIME(ÃµS)	TRX_LAYER_ARRIVAL_TIME(ÃµS)
2	0 N/A	Control_Packet	AODV_RREQ	NODE-1	Broadcast-0	NODE-1	NODE-4	N/A	N/A
3	0 N/A	Control_Packet	AODV_RREQ	NODE-1	Broadcast-0	NODE-1	NODE-4	N/A	N/A
4	0 N/A	Control_Packet	AODV_RREQ	NODE-1	Broadcast-0	NODE-4	NODE-1	N/A	N/A
5	0 N/A	Control_Packet	AODV_RREQ	NODE-1	Broadcast-0	NODE-4	NODE-3	N/A	N/A
6	0 N/A	Control_Packet	AODV_RREQ	NODE-1	Broadcast-0	NODE-3	NODE-4	N/A	N/A
7	0 N/A	Control_Packet	AODV_RREQ	NODE-1	Broadcast-0	NODE-3	NODE-5	N/A	N/A
8	0 N/A	Control_Packet	AODV_RREQ	NODE-1	Broadcast-0	NODE-1	NODE-4	N/A	N/A
9	0 N/A	Control_Packet	AODV_RREQ	NODE-1	Broadcast-0	NODE-4	NODE-1	N/A	N/A
10	0 N/A	Control_Packet	AODV_RREQ	NODE-1	Broadcast-0	NODE-4	NODE-3	N/A	N/A
11	0 N/A	Control_Packet	AODV_RREQ	NODE-1	Broadcast-0	NODE-3	NODE-4	N/A	N/A
12	0 N/A	Control_Packet	AODV_RREQ	NODE-1	Broadcast-0	NODE-3	NODE-5	N/A	N/A
13	0 N/A	Control_Packet	AODV_RREQ	NODE-1	Broadcast-0	NODE-5	NODE-2	N/A	N/A
14	0 N/A	Control_Packet	AODV_RREQ	NODE-1	Broadcast-0	NODE-5	NODE-3	N/A	N/A
15	0 N/A	Control_Packet	AODV_RREP	NODE-2	NODE-1	NODE-2	NODE-5	N/A	N/A
17	0 N/A	Control_Packet	AODV_RREP	NODE-2	NODE-1	NODE-5	NODE-3	N/A	N/A
19	0 N/A	Control_Packet	AODV_RREP	NODE-2	NODE-1	NODE-3	NODE-4	N/A	N/A
21	0 N/A	Control_Packet	AODV_RREP	NODE-2	NODE-1	NODE-4	NODE-1	N/A	N/A
102	0 N/A	Control_Packet	AODV_RREP(HELLO)	NODE-2	Broadcast-0	NODE-2	NODE-5	N/A	N/A
354	0 N/A	Control_Packet	AODV_RREP(HELLO)	NODE-2	Broadcast-0	NODE-2	NODE-5	N/A	N/A
358	0 N/A	Control_Packet	AODV_RREP(HELLO)	NODE-5	Broadcast-0	NODE-5	NODE-2	N/A	N/A
359	0 N/A	Control_Packet	AODV_RREP(HELLO)	NODE-5	Broadcast-0	NODE-5	NODE-3	N/A	N/A
417	0 N/A	Control_Packet	AODV_RREP(HELLO)	NODE-3	Broadcast-0	NODE-3	NODE-4	N/A	N/A
418	0 N/A	Control_Packet	AODV_RREP(HELLO)	NODE-3	Broadcast-0	NODE-3	NODE-5	N/A	N/A
502	0 N/A	Control_Packet	AODV_RREP(HELLO)	NODE-1	Broadcast-0	NODE-1	NODE-4	N/A	N/A
611	0 N/A	Control_Packet	AODV_RREP(HELLO)	NODE-2	Broadcast-0	NODE-2	NODE-5	N/A	N/A
614	0 N/A	Control_Packet	AODV_RREP(HELLO)	NODE-5	Broadcast-0	NODE-5	NODE-2	N/A	N/A
615	0 N/A	Control_Packet	AODV_RREP(HELLO)	NODE-5	Broadcast-0	NODE-5	NODE-3	N/A	N/A
693	0 N/A	Control_Packet	AODV_RREP(HELLO)	NODE-4	Broadcast-0	NODE-4	NODE-1	N/A	N/A
694	0 N/A	Control_Packet	AODV_RREP(HELLO)	NODE-4	Broadcast-0	NODE-4	NODE-3	N/A	N/A
712	0 N/A	Control_Packet	AODV_RREP(HELLO)	NODE-3	Broadcast-0	NODE-3	NODE-4	N/A	N/A
713	0 N/A	Control_Packet	AODV_RREP(HELLO)	NODE-3	Broadcast-0	NODE-3	NODE-5	N/A	N/A
767	0 N/A	Control_Packet	AODV_RREP(HELLO)	NODE-1	Broadcast-0	NODE-1	NODE-4	N/A	N/A
866	0 N/A	Control_Packet	AODV_RREP(HELLO)	NODE-2	Broadcast-0	NODE-2	NODE-5	N/A	N/A
869	0 N/A	Control_Packet	AODV_RREP(HELLO)	NODE-5	Broadcast-0	NODE-5	NODE-2	N/A	N/A
870	0 N/A	Control_Packet	AODV_RREP(HELLO)	NODE-5	Broadcast-0	NODE-5	NODE-3	N/A	N/A
954	0 N/A	Control_Packet	AODV_RREP(HELLO)	NODE-3	Broadcast-0	NODE-3	NODE-4	N/A	N/A

Figure 4-3: *AODV different control packet in Packet trace*

- Source Node 1 initiates route discovery process using RREQ packets. The generated route request is broadcast to the neighbors of the source node i.e. Node 4.
- Node 4 broadcasts the RREQ packet to its neighboring nodes i.e., Node 3, Node 1 since Node 4 does not have a route to destination Node 2.
- Similarly, Node3 broadcasts the RREQ packet to Node 5 and Node 4.
- Node 5 broadcasts to Node 2 and Node 3.
- On receiving a RREQ packet, the Destination Node 2 generates a RREP packet and transmits to Node 5, Node 2 → Node 5, Node 5 → Node 3, Node 3 → Node 4, and Node 4 → Node 1.
- After receiving the RREP, the Source Node 1 starts sending Data packets to Node 2 (Node 1 → Node 4 → Node 3 → Node 5 → Node 2). To observe the same, filter the CONTROL PACKET TYPE/APP NAME to APP1_CBR and PACKET ID to 1 in packet trace.



PACKET_ID	SEGMENT_ID	PACKET_TYPE	CONTROL_PACKET_TYPE/APP_NAME	SOURCE_ID	DESTINATION_ID	TRANSMITTER_ID	RECEIVER_ID
1	0	CBR	App1_CBR	NODE-1	NODE-2	NODE-1	NODE-4
1	0	CBR	App1_CBR	NODE-1	NODE-2	NODE-4	NODE-3
1	0	CBR	App1_CBR	NODE-1	NODE-2	NODE-3	NODE-5
1	0	CBR	App1_CBR	NODE-1	NODE-2	NODE-5	NODE-2

Figure 4-4: *Data packet flow from source to destination*

In Wireshark, filter AODV packets and select the RREQ packets and expand Ad hoc on-demand distance vector protocol to view the above-mentioned fields of RREQ as shown in Figure 4-4.

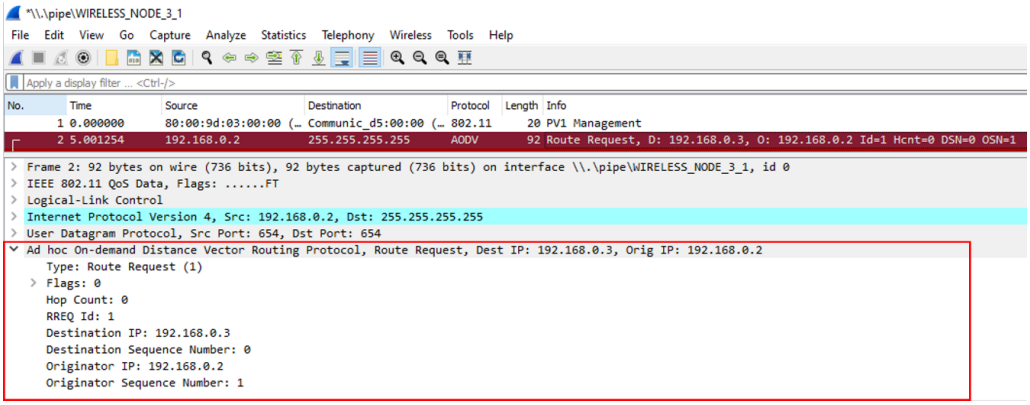


Figure 4-5: *AODV RREQ control packet in Wireshark*

Open Wireshark, filter AODV packets and select the RREP packets and expand Ad hoc on-demand distance vector protocol to view the above-mentioned fields of RREP as shown in Figure 4-5.

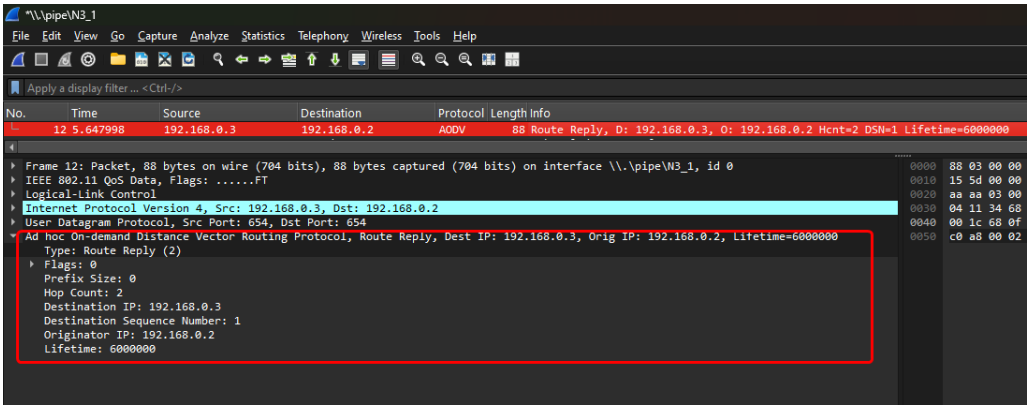


Figure 4-6: *AODV RREP control packet in Wireshark*

4.2 AODV Route Discovery and Route Maintenance

The following AODV examples extend the route-discovery and route-maintenance workflow shown in Section 4.1. They cover Expanding Ring Search, the G and D route-discovery flags, and RERR generation after a route break.

4.2.1 AODV Expanding Ring Search

Objective

This example shows how AODV uses Expanding Ring Search (ERS) to find a route to N6 with minimum overhead. N1 (source) sends a Route Request (RREQ) with a small TTL, limiting propagation radius. If N6 does not reply, N1 increases the TTL and retries. Three rings are needed: TTL 1 (radius 1), TTL 3 (radius 3), and TTL 5 (radius 5). N6 sits at five hops and replies on the third ring.

The same topology is used to validate the ERS sequence, packet-trace timing, and RREP return path.

Network Setup

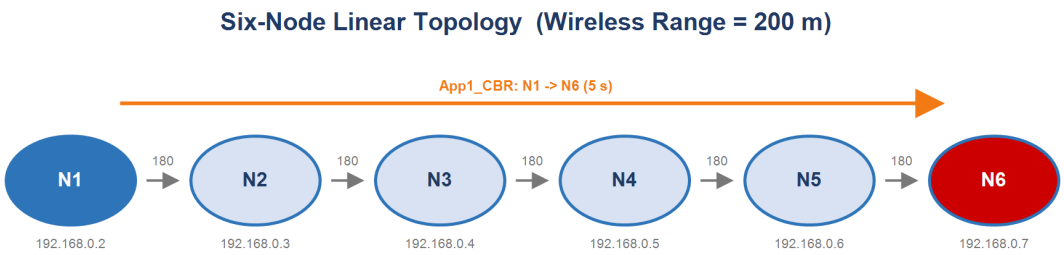


Figure 4-7: *Six-node linear topology. N1 = source, N6 = destination. Adjacent nodes are 180 m apart; wireless range is 200 m.*

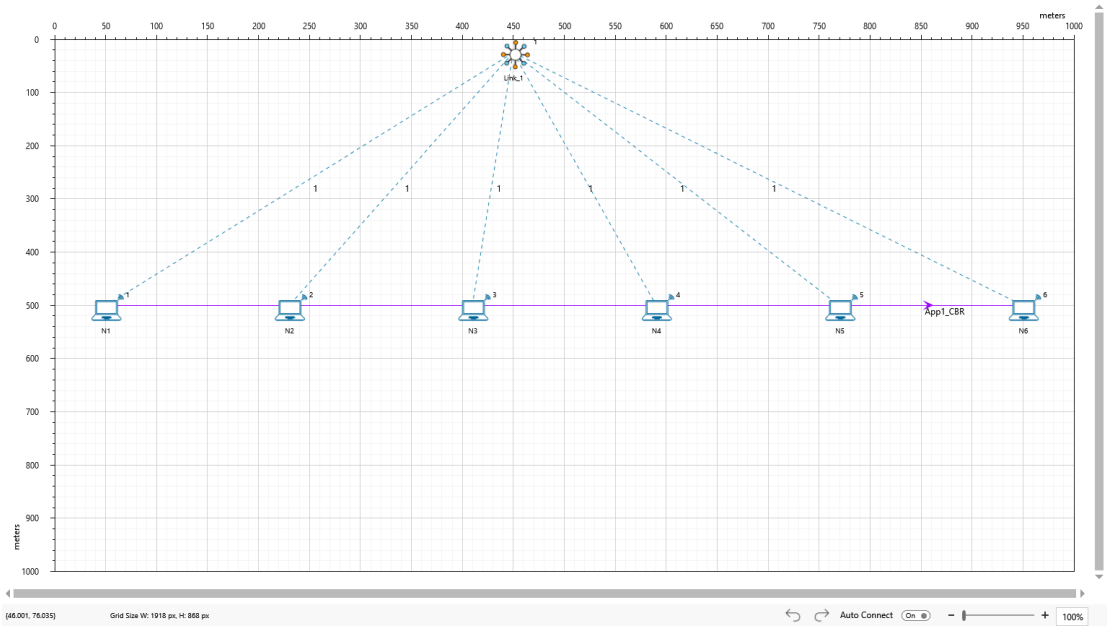


Figure 4-8: *NetSim topology view for the AODV Expanding Ring Search scenario.*

Node Configuration

Table 4-1: *Node Configuration*

Node	IP Address	Position (X, Y)	Role
N1	192.168.0.2	(50, 500)	CBR Source
N2	192.168.0.3	(230, 500)	Relay
N3	192.168.0.4	(410, 500)	Relay
N4	192.168.0.5	(590, 500)	Relay
N5	192.168.0.6	(770, 500)	Relay
N6	192.168.0.7	(950, 500)	CBR Destination

Network Settings

Table 4-2: *Network settings*

Parameter	Setting
Routing protocol	AODV
HELLO messages	Enabled
G flag	Disabled
D flag	Disabled
Propagation model	Range based
Wireless range	200 m
Adjacent-node distance	180 m
Simulation time	50 seconds
Packet trace	Enabled

Application

Table 4-3: *Application*

Parameter	Setting
Application name	App1_CBR
Type	CBR (UDP)
Source	N1
Destination	N6
Start time	5 seconds
Generation rate	584 kbps
Packet size	1460 bytes

At 5 seconds N1 has no route to N6. The first application packet triggers AODV route discovery.

RFC 3561 Expanding Ring Search Parameters

RFC 3561 Sections 6.4 and 10 define:

```
TTL_START = 1 TTL_INCREMENT = 2 TTL_THRESHOLD = 7
```

The expected search sequence for an unknown destination is:

TTL 1 → TTL 3 → TTL 5 → TTL 7 → NET_DIAMETER

N6 is five hops from N1. The TTL 1 and TTL 3 rings expire before reaching N6. The TTL 5 ring reaches N6 at exactly five hops.

Ring-traversal time (RING_TRAVERSAL_TIME) governs how long N1 waits between retries. The RREQ start times at N1 are 5.000000, 5.240000, and 5.640000 s, giving inter-ring gaps of 0.240 and 0.400 s.

Expanding Ring Search - Step by Step

The hop and RREP tables use `PHY_LAYER_END_TIME` from the packet trace, which records when each receiver finishes receiving the frame. The RREQ ring start times use `NW_LAYER_ARRIVAL_TIME` at N1.

The diagram below shows all three search rings on the six-node topology. Each semicircle represents the maximum propagation radius for that TTL value. The TTL 5 ring (red) reaches N6 at hop 5, triggering the RREP.

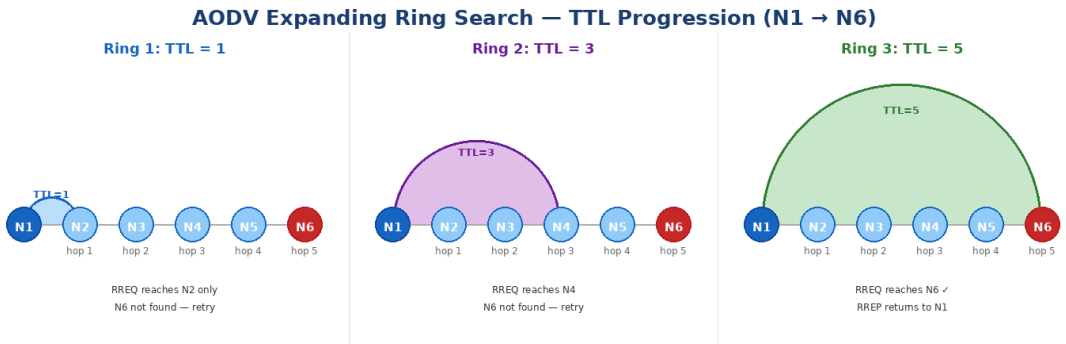


Figure 4-9: *AODV Expanding Ring Search TTL progression from N1 to N6. Ring 1 reaches N2, Ring 2 reaches N4, and Ring 3 reaches N6.*

Ring 1 - TTL = 1 (Radius: 1 hop)

N1 sends the first RREQ with TTL = 1. After the IP-layer decrement at N2, TTL = 0 and the RREQ cannot be forwarded. Only N2 receives it. N6 is not found; AODV waits RING_TRAVERSAL_TIME and retries.

Table 4-4: *Ring 1 - TTL = 1 (Radius: 1 hop)*

Receiver PHY-end time (s)	Transmitter	Receiver	Observation
5.001254	N1	N2	N2 receives the RREQ; TTL exhausted, cannot forward

Ring 2 - TTL = 3 (Radius: 3 hops)

N1 increments TTL by `TTL_INCREMENT = 2`, giving TTL = 3. The RREQ propagates through N2, N3, and reaches N4. N6 is still not reached, so AODV retries again.

Table 4-5: *Ring 2 - TTL = 3 (Radius: 3 hops)*

Receiver PHY-end time (s)	Transmitter	Receiver	Observation
5.241354	N1	N2	N2 receives the RREQ with TTL=3
5.242328	N2	N3	N3 receives the forwarded RREQ
5.243662	N3	N4	TTL exhausted at N4 - 3 hops reached, N6 not found

Ring 3 - TTL = 5 (Radius: 5 hops)

N1 increments TTL to 5. The RREQ propagates hop-by-hop to N6. N6 is the destination and generates a Route Reply (RREP).

Table 4-6: *Ring 3 - TTL = 5 (Radius: 5 hops)*

Receiver PHY-end time (s)	Transmitter	Receiver	Observation
5.641054	N1	N2	N2 receives the RREQ with TTL=5
5.642248	N2	N3	N3 receives the forwarded RREQ
5.643282	N3	N4	N4 receives the forwarded RREQ
5.644336	N4	N5	N5 receives the forwarded RREQ
5.645730	N5	N6	N6 receives - destination found, generates RREP

RREP Return Path

N6 sends an RREP as a unicast along the reverse route. Each node updates its routing table before forwarding toward N1.

Table 4-7: *RREP Return Path*

Receiver PHY-end time (s)	Transmitter	Receiver	Observation
5.646251	N6	N5	N5 receives the RREP
5.647266	N5	N4	N4 receives the forwarded RREP
5.648441	N4	N3	N3 receives the forwarded RREP
5.649516	N3	N2	N2 receives the forwarded RREP
5.650432	N2	N1	Route established at N1

Note: Expanding Ring Search applies to RREQ only. RREP travels as a unicast over the reverse route and does not use TTL rings.

Results

RREQ Propagation

The route discovery uses three RREQ rings. The first ring reaches N2, the second ring reaches N4, and the third ring reaches N6. N6 then sends the RREP back to N1 over the reverse route.

Ring Expansion Timeline

The packet trace shows N1 starting Ring 1 at 5.000000 s, Ring 2 at 5.240000 s, and Ring 3 at 5.640000 s. The inter-ring gaps are 0.240 s and 0.400 s. N6 receives the third-ring RREQ at 5.645730 s and the RREP reaches N1 at 5.650432 s.

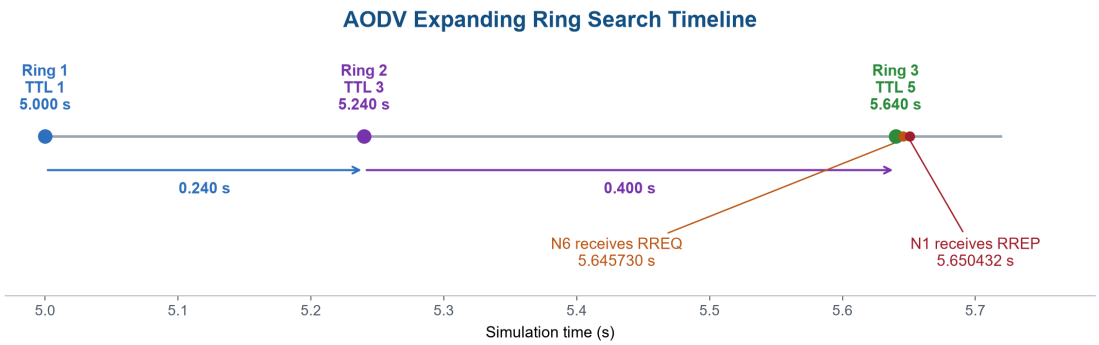


Figure 4-10: *Ring expansion timeline from the packet trace, with 0.240 s and 0.400 s gaps and receiver arrival times for the RREQ and RREP.*

Summary Table

Table 4-8: *Summary Table*

Measurement	Value
Search radii	1, 3, 5
RREQ transmissions	9
RREQ MAC payload bytes	396 B
RREQ PHY-layer bytes	756 B
RREQ reception rows in initial discovery	15
RREQ arrival at N6	5.645730 s
RREP arrival at N1	5.650432 s

Expanding Ring Search Overhead in This Example

In this six-node line topology, the destination is five hops away, so AODV reaches N6 only in the third ring. The packet trace therefore shows $1 + 3 + 5 = 9$ RREQ transmissions. Each RREQ uses 44 B of MAC payload in the NetSim packet trace, so the total RREQ MAC payload is 396 B.

This example is used to show the RFC-style ERS search sequence and packet-trace timing, not to claim a byte saving in this small topology. With `TTL_START = 1` and `TTL_INCREMENT = 2`, the corrected RREQ search radius expands as 1, 3, 5, and then 7 before switching to the network diameter if required. In this example, N6 is five hops away, so the third ring reaches the destination and triggers the RREP. This confirms that the search expands in controlled RFC-style rings instead of starting with a larger-than-required radius.

Table 4-9: *RREQ transmissions in the three ERS rings*

Ring	TTL	Radius	RREQ Tx	Outcome
1	1	1 hop	1	N6 not found
2	3	3 hops	3	N6 not found
3	5	5 hops	5	N6 found - RREP generated
Total	-	-	9	Route established

NetSim Behavior Notes

The packet trace should be read with the following NetSim timing details in mind.

Table 4-10: *NetSim behavior notes used when reading the packet trace*

Behavior	Trace effect
TTL handling at reception	After N2 receives Ring 1 with TTL = 1, the packet is not forwarded further. This is why Ring 1 reaches only N2 in the trace.
Ring traversal timer	The wait time between retries grows with the ring size. N1 starts the three RREQ rings 0.240 s and 0.400 s apart.
RREP forwarding	RREP is sent as a unicast over the reverse route. It is not part of the RREQ ring expansion.

RFC 3561 Compliance

Table 4-11: *RFC 3561 Compliance*

RFC 3561 Constant	Value	Effect in this example
TTL_START	1	First ring reaches 1 hop
TTL_INCREMENT	2	Each retry extends radius by 2 hops
TTL_THRESHOLD	7	Beyond this, NET_DIAMETER is used

Running the Example in NetSim

Run the saved AODV Expanding Ring Search scenario and open the packet trace from the simulation results window.

1. Open the six-node MANET scenario with AODV enabled.
2. Confirm that App1_CBR runs from N1 to N6 and starts at 5 seconds.
3. Run the simulation with packet trace enabled.
4. Filter the packet trace for AODV RREQ and AODV RREP control packets.
5. Check the RREQ rows for the three search rings: TTL 1, TTL 3, and TTL 5.
6. Check the RREP rows to verify that N6 replies and the route is established back at N1.

Use `Configuration.netsim`, `Packet Trace.csv`, and `Metrics.xml` to check the run output.

Conclusion

The six-node MANET example validates AODV Expanding Ring Search using the RFC 3561 TTL sequence 1, 3, and 5. The first two RREQ rings do not reach N6. The third ring reaches N6 at hop 5, N6 generates an RREP, and the RREP returns to N1 over the reverse route. The packet trace confirms the expected ERS behavior.

Reference: RFC 3561, Ad hoc On-Demand Distance Vector (AODV) Routing.
https://www.rfc-editor.org/rfc/rfc3561.html

4.2.2 AODV G/D Flag Scenario

Scenario Overview

This scenario demonstrates the effect of the AODV G (Gratuitous RREP) and D (Destination-Only RREP) flags on the route discovery process. Four wireless nodes start in a linear chain. Both saved cases use Random Waypoint mobility, with 10 m/s velocity, a 1 s calculation interval, and a 1 s pause. Two CBR applications generate one-hop and multi-hop traffic so that intermediate nodes build routing state before the flags are exercised.

It shows that the D flag restricts Route Reply (RREP) generation to the destination node, while the G flag causes an intermediate node that generates an RREP to also send a gratuitous RREP toward the destination. The packet trace is used to verify this behaviour by comparing the RREP source, destination, transmitter, receiver, and arrival-time fields.

Network Topology

The following network diagram illustrates the topology used to study the AODV G and D flags.

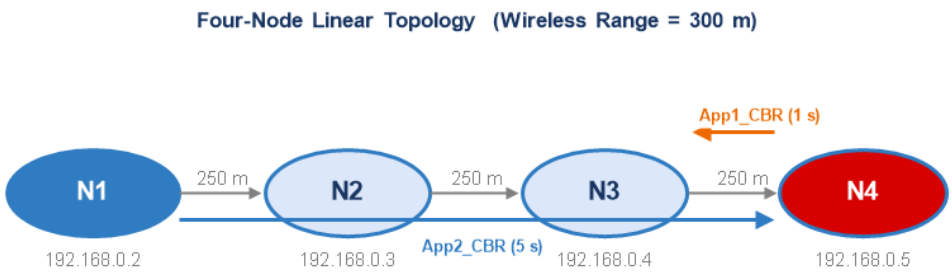


Figure 4-11: *Four-node linear topology.*

App1 (orange) runs N4 → N3 from 1 s. App2 (blue) runs N1 → N4 from 5 s.

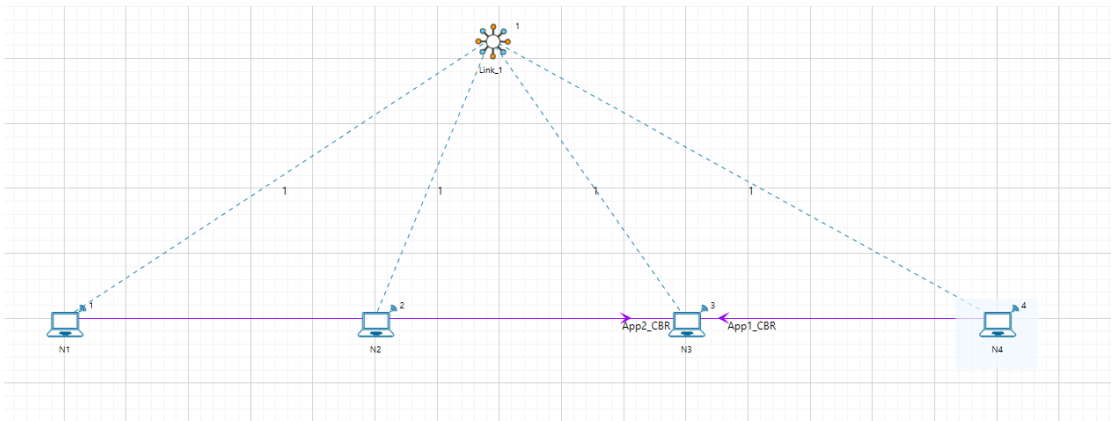


Figure 4-12: *NetSim topology view for the AODV G/D flag scenario.*

Simulation Configuration

1. Create a MANET scenario with four wireless nodes placed in a linear topology.
2. Use a wireless range of 300 m so that adjacent nodes can communicate, and non-adjacent nodes require multi-hop forwarding.
3. Set the mobility model to Random Waypoint for all nodes, with 10 m/s velocity, a 1 s calculation interval, and a 1 s pause.
4. Set the routing protocol in the network layer as AODV for all nodes.

5. Configure App1_CBR from Node 4 to Node 3 with start time 1 s.
6. Configure App2_CBR from Node 1 to Node 4 with start time 5 s.
7. Enable packet trace and filter the control packets for `AODV_RREP` and `AODV_RREP(GRATUITOUS)`.
8. Run the two cases separately by changing the G and D flag settings as shown below.

Simulation Parameters

Table 4-12: *Simulation parameters.*

Parameter	Setting
Number of nodes	4
Network type	MANET
Environment Size	1000 m x 1000 m
Node mobility	Random Waypoint (10 m/s; 1 s pause)
Routing protocol	AODV
HELLO messages	Enabled
Simulation time	50 seconds
Wireless range	300 m
Link property	Range based

Node Positions

Table 4-13: *Node positions and roles.*

Node	IP Address	Initial position (X, Y)	Role
Node 1	192.168.0.2	(100, 500)	App2 source
Node 2	192.168.0.3	(350, 500)	Relay
Node 3	192.168.0.4	(600, 500)	App1 dest / App2 relay
Node 4	192.168.0.5	(850, 500)	App1 source / App2 dest

Traffic Flows

Table 4-14: *CBR traffic flow configuration.*

App	Source	Destination	Start	Packet size	Rate
App1_CBR	Node 4	Node 3	1 s	1460 bytes	584 kbps
App2_CBR	Node 1	Node 4	5 s	1460 bytes	584 kbps

AODV Flag Behaviour

G Flag (Gratuitous RREP)

When the G flag is enabled, a node that sends an RREP also sends a separate gratuitous RREP toward the destination. This allows the destination to learn a route back to the source without initiating its own route discovery.

The G flag does not allow the intermediate node to reply; intermediate replies are controlled by the D flag. When D is disabled, an intermediate node with a valid route may reply, independent of the G flag.

In this scenario, Node 3 has a valid route to Node 4 and replies to App2's RREQ at 5 s.

Without the G flag, Node 3 sends a single RREP toward Node 1.

With the G flag enabled, Node 3 sends the normal RREP toward Node 1. It also sends a gratuitous RREP toward Node 4.

As a result, Node 4 learns a route to Node 1 from the gratuitous RREP generated by Node 3. In the packet trace, the entry AODV_RREP(GRATUITOUS) identifies the additional RREP sent toward the destination (Node 4).

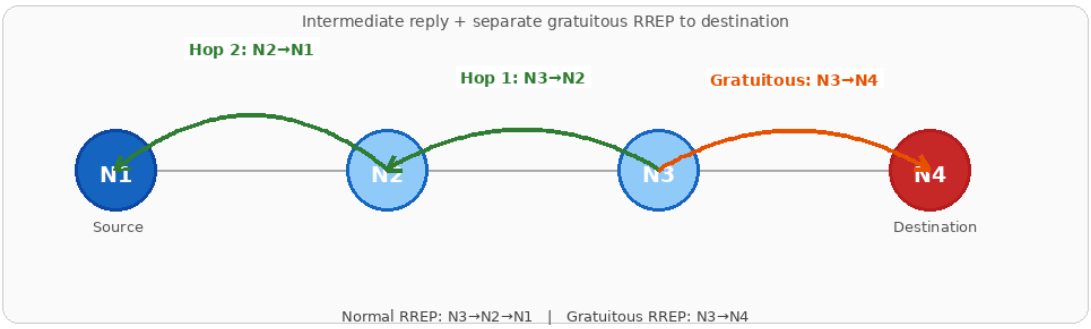


Figure 4-13: *AODV G/D flag figure.*

D Flag (Destination-Only RREP)

When the D flag is set, only the destination node may reply to an RREQ. Intermediate nodes with a valid cached route must forward the RREQ without replying. The D flag takes priority over the G flag; if both are set, only the destination replies and no gratuitous RREP is generated.

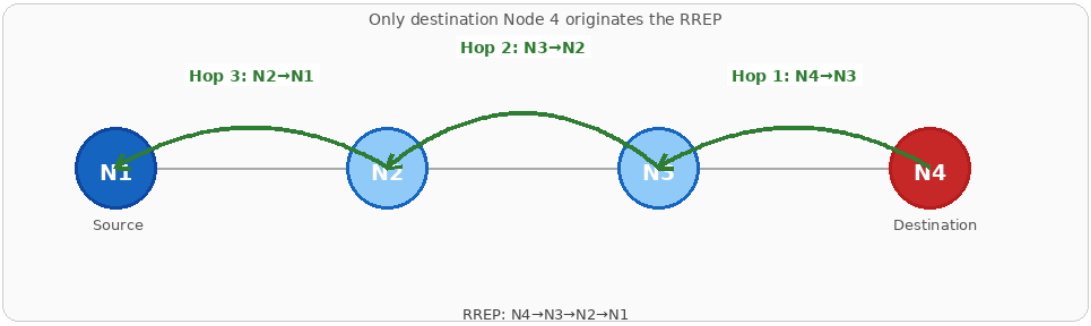


Figure 4-14: *AODV G/D flag figure.*

Simulation Results

Open the packet trace from the simulation results window. The output is interpreted by comparing the RREP source, destination, transmitter, receiver, and timing columns. The RREP tables use `PHY_LAYER_END_TIME` from the packet trace, which records when each receiver finishes receiving the frame.

Case 1: D Flag Enabled

Table 4-15: *Flag settings for Case 1 (D flag enabled).*

Flag	Setting
G Flag	Disabled
D Flag	Enabled

Only Node 4 (the destination) generates the RREP for App2. The reply is forwarded hop-by-hop back to Node 1, and the SOURCE_ID remains Node 4 in every row.

Table 4-16: *Packet trace results for Case 1 (D flag enabled).*

Receiver PHY-end time (s)	Packet Type	Control Type	Source → Dest	Tx → Rx	Source IP → Dest IP
5.245911	Control_Packet	`AODV_RREP`	NODE-4 → NODE-1	NODE-4 → NODE-3	192.168.0.5 → 192.168.0.2
5.246606	Control_Packet	`AODV_RREP`	NODE-4 → NODE-1	NODE-3 → NODE-2	192.168.0.5 → 192.168.0.2
5.247441	Control_Packet	`AODV_RREP`	NODE-4 → NODE-1	NODE-2 → NODE-1	192.168.0.5 → 192.168.0.2

Case 2: G Flag Enabled

Table 4-17: *Flag settings for Case 2 (G flag enabled).*

Flag	Setting
G Flag	Enabled
D Flag	Disabled

App1 starts at 1 s (Node 4 → Node 3, one-hop). Node 3 learns the route to Node 4 at approximately 1 s. When App2 starts at 5 s (Node 1 → Node 4), Node 1 broadcasts an RREQ. Node 3 holds a valid route to Node 4 and, because D is disabled, replies directly.

Because G is enabled, Node 3 also sends a separate RREP toward Node 4 - the gratuitous RREP - so Node 4 learns the route to Node 1.

Rows near 1.001775 s belong to App1 route discovery. Rows near 5.24 s include the App2 RREP back to Node 1 and the separate RREP(GRATUITOUS) toward Node 4.

Table 4-18: *Packet trace results for Case 2 (G flag enabled).*

Receiver PHY-end time (s)	Packet Type	Control Type	Source → Dest	Tx → Rx	Source IP → Dest IP
1.001775	Control_Packet	`AODV_RREP`	NODE-3 → NODE-4	NODE-3 → NODE-4	192.168.0.4 → 192.168.0.5
5.242889	Control_Packet	`AODV_RREP`	NODE-3 → NODE-1	NODE-3 → NODE-2	192.168.0.4 → 192.168.0.2
5.244025	Control_Packet	`AODV_RREP`	NODE-3 → NODE-1	NODE-2 → NODE-1	192.168.0.4 → 192.168.0.2
5.243584	Control_Packet	`AODV_RREP(GRATUITOUS)`	NODE-3 → NODE-4	NODE-3 → NODE-4	192.168.0.4 → 192.168.0.5
5.253911	Control_Packet	`AODV_RREP(GRATUITOUS)`	NODE-3 → NODE-4	NODE-3 → NODE-4	192.168.0.4 → 192.168.0.5

Note: The first row (AODV_RREP, SOURCE_ID=NODE-3, DEST_ID=NODE-4, arrival 1.001775 s) belongs to App1's one-hop route discovery at 1 s, where Node 3 learns the direct route to Node 4. It is a plain AODV_RREP, not a gratuitous RREP. The gratuitous mechanism only fires during App2's discovery at 5 s, which produces the two AODV_RREP(GRATUITOUS) rows near 5.24 s.

Message Sequence Diagrams

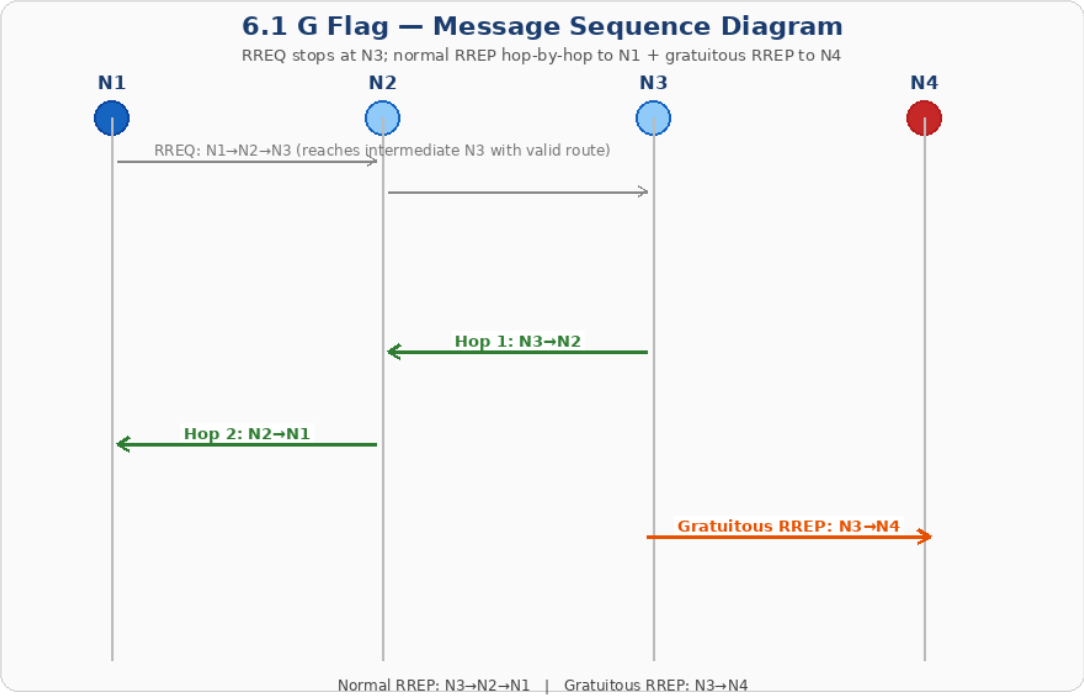


Figure 4-15: *AODV G/D flag figure.*

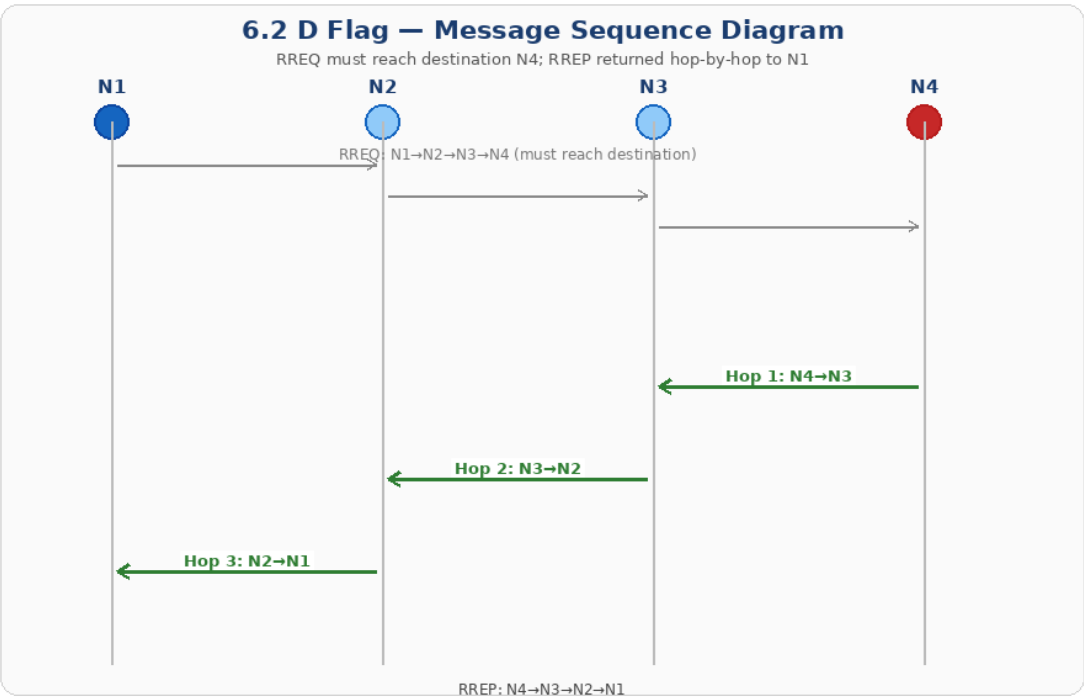


Figure 4-16: *AODV G/D flag figure.*

Running the Example in NetSim

Run the simulation for both flag configurations and open the packet trace from the simulation results window.

For Case 1, filter the control packets for AODV_RREP and observe that the RREP originates from Node 4, the destination.

For Case 2, filter AODV_RREP and AODV_RREP(GRATUITOUS). The normal RREP travels back toward Node 1, while the gratuitous RREP is sent separately toward Node 4.

Use the receiver PHY-end time column to separate the App1 discovery near 1 s from the App2 discovery near 5.24 s.

Additional NetSim Settings

Table 4-19: *Additional NetSim settings for the G and D flags.*

Setting	Value	Effect
G Flag	Enabled / Disabled	Enabled: the node sending an RREP also sends a separate RREP(GRATUITOUS) toward the destination, so the destination learns a route back to the originator
D Flag	Enabled / Disabled	Enabled: only destination node may reply; overrides G flag
HELLO Messages	Enabled	Nodes advertise presence to neighbors periodically

4.2.3 AODV RERR Route Maintenance

Objective

This example shows AODV route maintenance after an established route breaks mid-session. N1 discovers a route to N6, sends CBR traffic over it, then N4 moves out of range at 15 seconds. AODV detects the next-hop failure through the MAC layer and generates AODV_RERR.

This scenario is separate from Expanding Ring Search. Expanding Ring Search checks RREQ TTL behavior during initial route discovery. This scenario checks what AODV does after a working route fails.

Network Setup

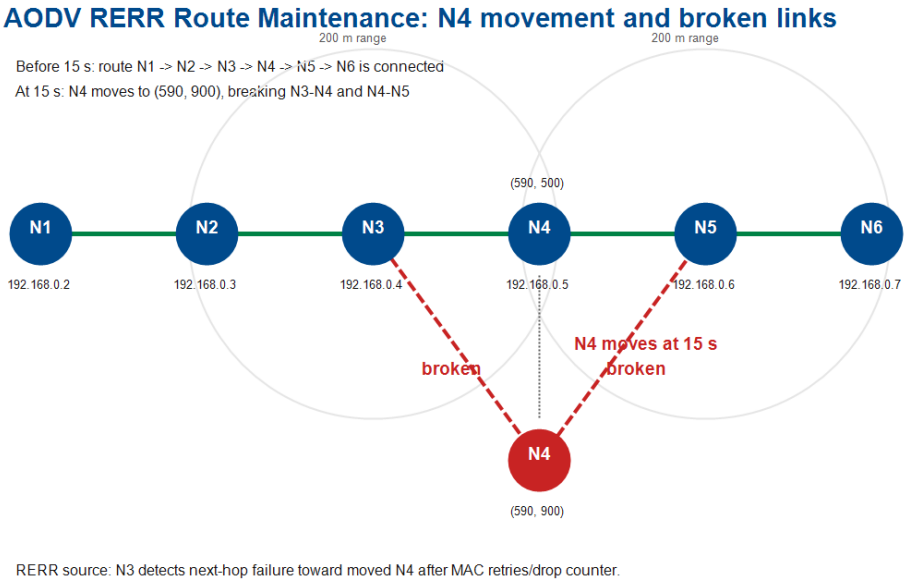


Figure 4-17: *N4 position before and after the 15 s mobility event. The moved N4 position breaks the N3–N4 and N4–N5 links.*

Node Configuration

Table 4-20: *Node Configuration*

Node	IP Address	Initial Position (X, Y)	Role
N1	192.168.0.2	(50, 500)	CBR Source
N2	192.168.0.3	(230, 500)	Relay
N3	192.168.0.4	(410, 500)	Relay - sends RERR
N4	192.168.0.5	(590, 500) → (590, 900)	Relay - moves at t=15 s
N5	192.168.0.6	(770, 500)	Relay
N6	192.168.0.7	(950, 500)	CBR Destination

Simulation Parameters

Table 4-21: *Simulation Parameters*

Parameter	Setting
Number of nodes	6
Routing protocol	AODV
HELLO messages	Enabled
G flag	Disabled
D flag	Disabled
Propagation model	Range based
Wireless range	200 m
Adjacent-node distance	180 m
Simulation time	35 seconds
Packet trace	Enabled

Application

Table 4-22: *Application*

Parameter	Setting
Application name	App1_CBR
Source	N1
Destination	N6
Transport	UDP
Start time	5 seconds
Rate	584 kbps
Packet size	1460 bytes
Inter-arrival time	20 000 microseconds

Controlled Route Break

File-based mobility is used to produce one controlled link break. The mobility file is stored in the scenario `ConfigSupport` folder as `mobility.csv` and contains one movement row:

```
#Time(s),Device ID,X,Y,Z
15,4,590,900,0
```

Initial positions are stored in Configuration.netsim. At 15 seconds, only N4 moves from (590, 500) to (590, 900). This places N4 beyond the 200 m range of both N3 and N5, breaking the established path at both ends of N4.

How RERR Is Triggered

After N4 moves, packets sent toward the former next hop fail at the 802.11 MAC layer. The AODV packet-drop callback counts drops and generates RERR when its counter exceeds ten. In this run it is the only RERR trigger, because an expired neighbour route no longer generates a RERR. The trace contains two RERR packets, both after the movement: N1 generates one at 15.957010 s and N3 generates one at 16.432025 s. No RERR packet appears before 15 s.

N1's RERR (transmitted to N2) is lost: its packet status is `Collided` at 15.996962 s, because N1's own frames to N2 collide repeatedly from about 15.02 s onwards and its drop counter exceeds ten. N3's RERR has a network-layer timestamp of 16.432025 s, about 1.43 s after the break, but is received by N2 at 20.379204 s. The 3.9 s difference falls between the MAC-layer arrival (16.432025 s) and the PHY-layer arrival (20.378436 s) of the frame; the trace does not show why it takes this long, so the cause is not identified here. The 16.432 s timestamp is not the arrival time at N2.

Packet Trace Evidence

The RREP, CBR, and RERR receiver-time tables use `PHY_LAYER_END_TIME` from the packet trace, which records when each receiver finishes receiving the frame; the route-restart table instead lists `NW_LAYER_ARRIVAL_TIME`.

Route Discovery (about 5 s)

At 5 seconds N1 has no route to N6 and triggers AODV route discovery. The RREP returns from N6 through all intermediate nodes, confirming the route:

Table 4-23: *Route Discovery (about 5 s)*

Receiver PHY-end time (s)	Packet type	Transmitter	Receiver
5.646251	AODV_RREP	N6	N5
5.647266	AODV_RREP	N5	N4
5.648441	AODV_RREP	N4	N3
5.649516	AODV_RREP	N3	N2
5.650432	AODV_RREP	N2	N1

Data Uses the Route (about 14.9 s)

A CBR packet at 14.9 s follows the path N1 → N2 → N3 → N4 → N5 → N6, confirming the route carried application traffic before the break.

Table 4-24: *Data Uses the Route (about 14.9 s)*

Receiver PHY-end time (s)	Packet ID	Transmitter	Receiver	Status
14.901914	496	N1	N2	OK
14.903642	496	N2	N3	OK
14.905650	496	N3	N4	OK
14.907338	496	N4	N5	OK
14.909106	496	N5	N6	OK

RERR After the Break

After N4 moves at 15 s, the former route cannot carry traffic to N6. The trace shows RERR generation at N1 (15.957010 s) and N3 (16.432025 s). Their packet-trace rows have different receiver times and packet status:

Table 4-25: *RERR rows after the 15 s route break*

Receiver PHY-end time (s)	Packet type	Transmitter	Receiver	Status
15.996962	AODV_RERR	NODE-1	NODE-2	Collided
20.379204	AODV_RERR	NODE-3	NODE-2	Successful

Packet-trace fields for the N3 → N2 row show the gap between its network-layer and PHY-layer timestamps:

Table 4-26: *Packet-trace fields for the upstream RERR row*

Field	Value
PACKET_TYPE	Control_Packet
CONTROL_PACKET_TYPE	AODV_RERR
SOURCE_ID	NODE-3
DESTINATION_ID	Broadcast-0
TRANSMITTER_ID	NODE-3
RECEIVER_ID	NODE-2
NW_LAYER_ARRIVAL_TIME	16 432 025.000 us
PHY_LAYER_ARRIVAL_TIME	20 378 436.110 us
PHY_LAYER_END_TIME	20 379 204.120 us
PACKET_STATUS	Successful
SOURCE_IP	192.168.0.4
DESTINATION_IP	255.255.255.255
NEXT_HOP_IP	255.255.255.255

Route Discovery Restarts

The CBR application continues after the break, and new RREQ attempts appear in the trace. The RREQs sent at 15.960000 s and 16.200000 s collide at N2. The RREQ sent at 16.600000 s reaches N2, and N2 answers with an RREP (received at N1 at 16.601735 s). The RREQ sent at 20.680000 s, after N3's RERR reached N2, is forwarded by N2 to N3 and goes no further. Most of these RREQs begin before N3's RERR reaches N2, so the trace does not support attributing them to receipt of that RERR. Because N4 stays out of range, the original path through N4 cannot be rebuilt.

Table 4-27: *Route Discovery Restarts*

Network-layer time (s)	Packet type	Transmitter	Receiver
15.960000	AODV_RREQ	N1	N2 (collided)
16.200000	AODV_RREQ	N1	N2 (collided)
16.600000	AODV_RREQ	N1	N2
20.680000	AODV_RREQ	N1	N2
20.682948	AODV_RREQ	N3	N2

Throughput at N6

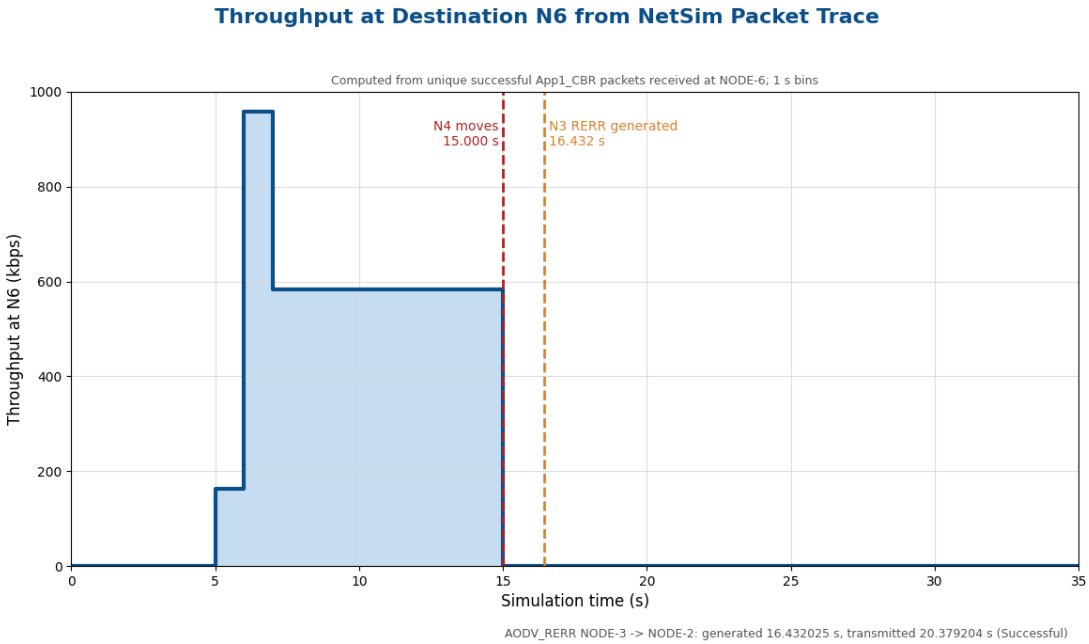


Figure 4-18: *Throughput at N6 from the packet trace, with N4's 15 s movement and the 16.432 s RERR generated at N3. N3's RERR is received by N2 at 20.379 s.*

At N6, throughput is present after route discovery and remains near the configured offered load until N4 moves at 15 s. After the link break, no more data packets reach N6 through the original path. The plot was generated from unique successful `App1_CBR` packets received at `NODE-6` in the NetSim packet trace.

What This Scenario Proves

Table 4-28: *What This Scenario Proves*

#	Claim	Evidence
1	Route exists before the break	RREP arrival rows 5.646-5.650 s show the five-hop reply chain N6 → N1
2	Data uses that route	CBR packet 496 at 14.9 s traverses N1 → N2 → N3 → N4 → N5 → N6
3	N4 movement breaks the route	N4 moves to (590, 900) at 15 s, placing it beyond 200 m from N3 and N5
4	AODV generates RERR	N1 generates RERR at 15.957010 s (Collided) and N3 at 16.432025 s; N3's RERR reaches N2 at 20.379204 s
5	Trigger is MAC drop count	AODV's packet-drop callback generates RERR when its counter exceeds ten
6	Route cannot be rebuilt	New RREQ attempts reach only N2 and N3 while N4 is out of range

Conclusion

The scenario shows AODV route maintenance around a controlled mobility break. The route is created at 5 s and carries traffic until N4 moves at 15 s. The trace then shows RERR generation at N1 and N3, delivery of N3's RERR to N2 at 20.379 s, and new route-discovery attempts while the application continues to send.

4.3 ZRP-IARP

Open NetSim and select Examples→Mobile Ad hoc Networks→ZRP IARP then click on the tile in the middle panel to load the example as shown in below screenshot Figure 4-6.

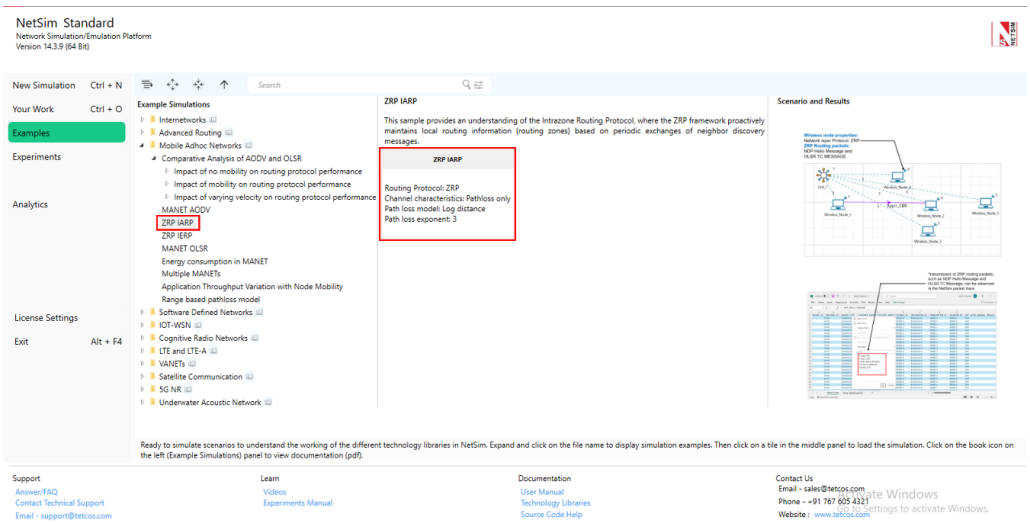


Figure 4-19: *List of scenarios for the example of ZRP-IARP*

The following network diagram illustrates what the NetSim UI displays when you open the example configuration file see Figure 4-19.

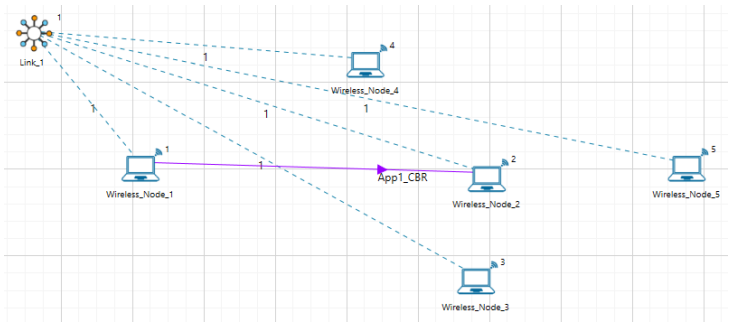


Figure 4-20: *Network set up for studying the ZRP-IARP*

4.3.1 Network Settings

1. Set grid length as 600m×300m from grid setting property panel on the right. This needs to be done before any device is placed on the grid.
2. A network scenario is designed in NetSim GUI consisting of 5 wireless nodes and one ad-hoc link in the "Standard MANET" module of MANET Network library.
3. To configure any properties in the nodes, click on the node, expand the property panel on the right side, and change the properties as mentioned in the below steps.

4. In the position properties set mobility model as no mobility for all devices present in GUI.
5. The Medium access protocol was set to DCF in Interface 1(wireless)→ Datalink layer of all Wireless nodes.
6. The routing protocol in the network layer is set to ZRP for all nodes. As this is a global property, changing it on one node will affect all nodes.
7. Click on link and expand property panel on the right and set the channel characteristics: path loss only, path loss model: log distance, path loss exponent: 3 under medium property.
8. Configure CBR application from set traffic tab in ribbon on the top between wireless node 1 to wireless node 2, click on the created application, expand the property panel on the right and set transport protocol to UDP.
9. Enable packet trace from configure reports tab in ribbon on the top.
10. Run simulation for 10 seconds.

4.3.2 Output

Open Packet trace from simulation results window and observe Control Packet Type/App Name.

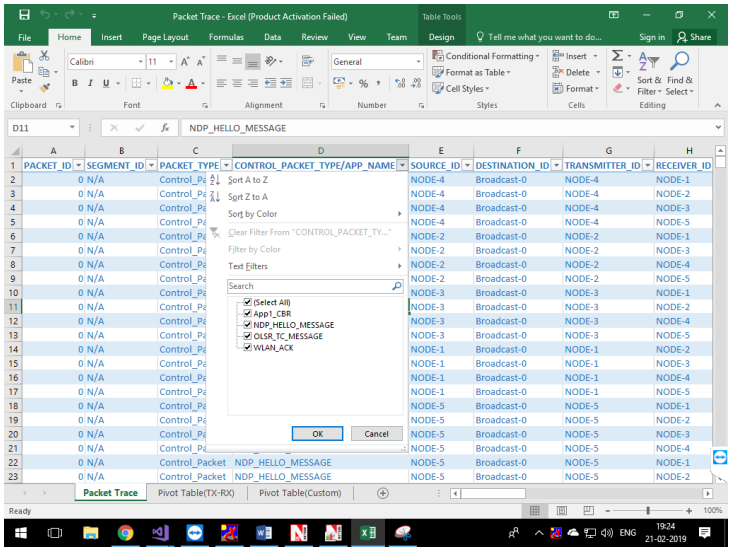


Figure 4-21: *Packet Trace*

Users can see only NDP Hello Message and OLSR TC MESSAGE since the destination node is within the zone. The ZRP framework proactively maintains local routing information (routing zones) based on periodic exchanges of neighbor discovery messages.

4.4 ZRP-IERP

Open NetSim and Select Examples → Mobile Ad hoc Networks → ZRP IERP then click on the tile in the middle panel to load the example as shown in below screenshot Figure 4-21.

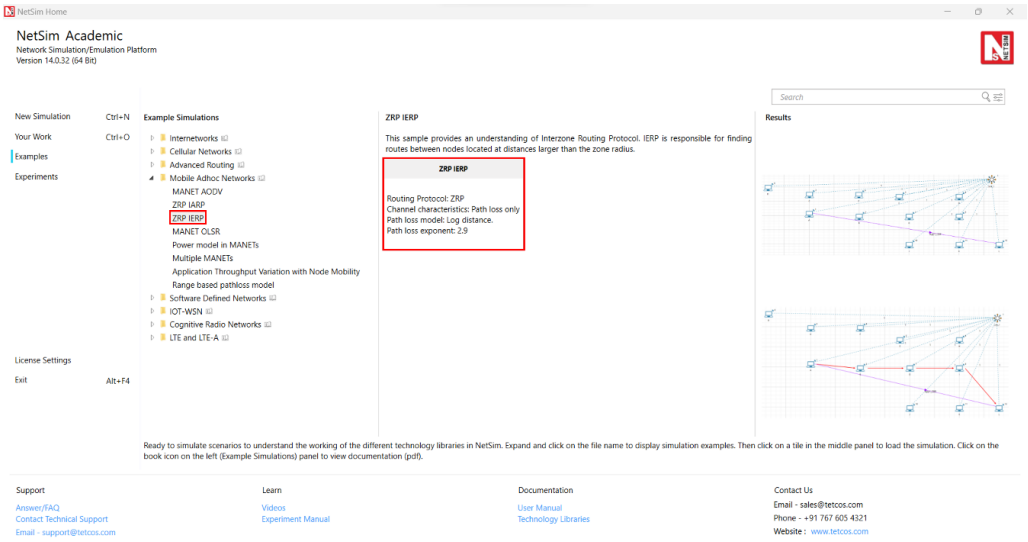


Figure 4-22: *List of scenarios for the example of ZRP-IERP*

The following network diagram illustrates what the NetSim UI displays when you open the example configuration file see Figure 4-22.

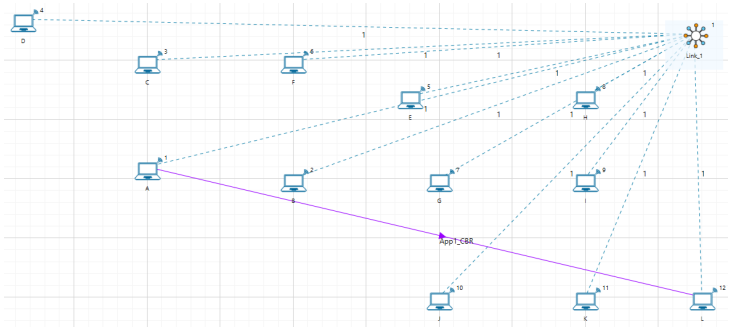


Figure 4-23: *Network set up for studying the ZRP-IERP*

4.4.1 Network Settings

1. Set grid length as 600m×300m from grid setting property panel on the right. This needs to be done before any device is placed on the grid.
2. A network scenario is designed in NetSim GUI consisting of 12 wireless nodes and one ad-hoc link in the "Standard MANET" of MANET Network library.
3. To configure any properties in the nodes, click on the node, expand the property panel on the right side, and change the properties as mentioned in the below steps.
4. In the position properties set mobility model as No mobility for all devices present in GUI.
5. The medium access protocol was set to DCF in Interface 1(wireless)→ Datalink layer of all wireless nodes.
6. The routing protocol in the network layer is set to ZRP for all nodes. As this is a global property, changing it on one node will affect all nodes.
7. Click on link and open property panel on the right and set the Channel characteristics: Log distance, Path loss exponent: 2.9.
8. Configure CBR application from wireless nodes 1 to wireless nodes 12 by clicking on set traffic tab in ribbon on the top. Click on the created application and set the transport protocol to UDP, keeping the other properties as default.

9. Enable Packet Trace from configure reports tab in the ribbon on the top.
10. Run simulation for 10 seconds.

4.4.2 Output

Open packet trace and filter Control Packet Type/App Name to IERP route request. In the below packet trace, node-1 is sending IERP route request packets to its peripheral nodes 2 and 3 since the destination is not present in node1's zone. Similarly, Node 2 transmits IERP route request packets to node 5 and 7 and node 3 transmits to node 4. To observe the IERP route request flow from the respective nodes filter the Transmitter Id to Node 1, Node 2 and Node 3 alone.

A	B	C	D	E	F	G	H
PACKET_ID	SEGMENT_ID	PACKET_TYPE	CONTROL_PACKET_TYPE/APP_NAME	SOURCE_ID	DESTINATION_ID	TRANSMITTER_ID	RECEIVER_ID
0	N/A	Control_Packet	IERP_ROUTE_REQUEST	NODE-1	NODE-5	NODE-1	NODE-2
0	N/A	Control_Packet	IERP_ROUTE_REQUEST	NODE-1	NODE-7	NODE-1	NODE-2
0	N/A	Control_Packet	IERP_ROUTE_REQUEST	NODE-1	NODE-4	NODE-1	NODE-3
0	N/A	Control_Packet	IERP_ROUTE_REQUEST	NODE-1	NODE-4	NODE-3	NODE-4
0	N/A	Control_Packet	IERP_ROUTE_REQUEST	NODE-4	NODE-6	NODE-4	NODE-3
0	N/A	Control_Packet	IERP_ROUTE_REQUEST	NODE-1	NODE-5	NODE-2	NODE-5
0	N/A	Control_Packet	IERP_ROUTE_REQUEST	NODE-4	NODE-6	NODE-4	NODE-3
0	N/A	Control_Packet	IERP_ROUTE_REQUEST	NODE-1	NODE-7	NODE-2	NODE-7
0	N/A	Control_Packet	IERP_ROUTE_REQUEST	NODE-5	NODE-3	NODE-5	NODE-6
0	N/A	Control_Packet	IERP_ROUTE_REQUEST	NODE-5	NODE-3	NODE-5	NODE-6
0	N/A	Control_Packet	IERP_ROUTE_REQUEST	NODE-4	NODE-6	NODE-4	NODE-3
0	N/A	Control_Packet	IERP_ROUTE_REQUEST	NODE-4	NODE-6	NODE-4	NODE-3
0	N/A	Control_Packet	IERP_ROUTE_REQUEST	NODE-5	NODE-3	NODE-5	NODE-6
0	N/A	Control_Packet	IERP_ROUTE_REQUEST	NODE-5	NODE-3	NODE-5	NODE-6
0	N/A	Control_Packet	IERP_ROUTE_REQUEST	NODE-4	NODE-6	NODE-4	NODE-3
0	N/A	Control_Packet	IERP_ROUTE_REQUEST	NODE-5	NODE-3	NODE-5	NODE-6
0	N/A	Control_Packet	IERP_ROUTE_REQUEST	NODE-4	NODE-6	NODE-4	NODE-3
0	N/A	Control_Packet	IERP_ROUTE_REQUEST	NODE-5	NODE-3	NODE-5	NODE-6
0	N/A	Control_Packet	IERP_ROUTE_REQUEST	NODE-4	NODE-6	NODE-4	NODE-3
0	N/A	Control_Packet	IERP_ROUTE_REQUEST	NODE-5	NODE-3	NODE-5	NODE-6
0	N/A	Control_Packet	IERP_ROUTE_REQUEST	NODE-5	NODE-9	NODE-5	NODE-7
0	N/A	Control_Packet	IERP_ROUTE_REQUEST	NODE-5	NODE-9	NODE-5	NODE-7
0	N/A	Control_Packet	IERP_ROUTE_REQUEST	NODE-5	NODE-9	NODE-5	NODE-7
0	N/A	Control_Packet	IERP_ROUTE_REQUEST	NODE-5	NODE-9	NODE-5	NODE-7
0	N/A	Control_Packet	IERP_ROUTE_REQUEST	NODE-5	NODE-9	NODE-5	NODE-7
0	N/A	Control_Packet	IERP_ROUTE_REQUEST	NODE-5	NODE-9	NODE-5	NODE-7

Figure 4-24: *Packet Trace*

Node 7 generates an IERP Route Reply packet in response to the IERP Route Request since the destination node, Node 12, is present in the Node 7 zone. IERP Route Reply packet routes through Node 7 → Node 2 and Node 2 → Node 1. And the data packet routes through Node 1 → Node 2, Node 2 → Node 7, Node 7 → Node 9 and Node 9 → Node 12.

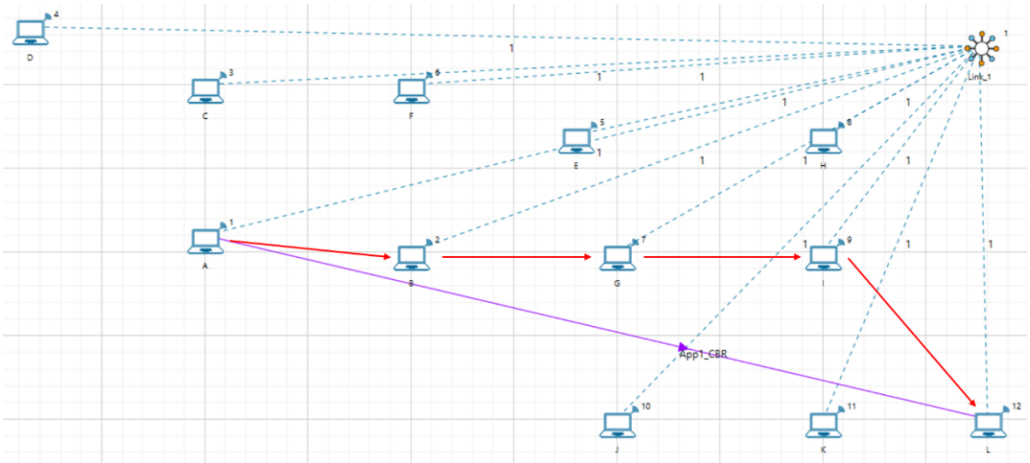


Figure 4-25: *Packets take the route 1 → 2 → 7 → 9 → 12 when running ZRP-IERP*

4.5 MANET-OLSR

4.5.1 Part-1: MPR Table Formation in NetSim

Introduction

In OLSR, only a subset of preselected nodes called MPR (Multi-Point Relays) are used to perform topological advertisements. Control messages (containing e.g. routing information) are broadcast and forwarded only by MPRs. This leads to a reduction in the number of transmitter nodes, in the overhead and in useless receptions of messages on nodes. The well-known storm problem due to broadcasting is avoided.

Creation of MPRs in OLSR

Neighbor Discovery

- Each node in the OLSR network periodically broadcasts HELLO messages to its immediate neighbors. These messages contain information about the node itself and its direct neighbors (1-hop neighbors).
- From these HELLO messages, a node learns about its 1-hop and 2-hop neighbors (the neighbors of its neighbors).

MPR Selection

Goal: The primary goal of selecting MPRs is to ensure that all 2-hop neighbors of a node can be reached through at least one of its MPRs.

A node selects a subset of its 1-hop neighbors as MPRs in such a way that these MPRs can cover all of its 2-hop neighbors. This means that every 2-hop neighbor must be reachable via at least one selected MPR.

The selection of MPRs is done based on degree of connectivity – nodes with a larger number of 2-hop neighbors might be preferred to cover more ground with fewer MPRs.

Route Calculation Using MPRs

Only the nodes selected as MPRs are responsible for forwarding control messages like Topology Control (TC) messages. This significantly reduces the number of transmissions, leading to less network overhead.

When a node needs to broadcast a TC message, it only forwards it to its MPRs, which in turn continue the broadcast, ensuring that the entire network receives the update.

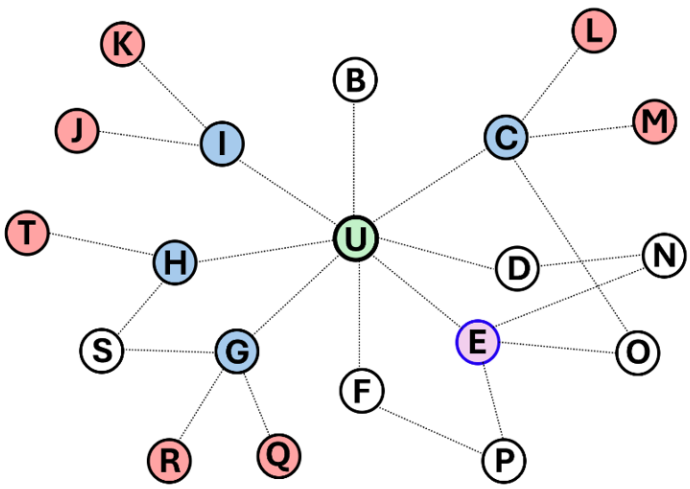


Figure 4-26: *Illustration of MPR Selection Algorithm. U is the node for which the MPR set is being calculated. Red nodes are the set of isolated nodes. Blue nodes are the set of MPR nodes obtained in the first step. Node E, in purple, is the MPR node selected in the second step. At the end of step 2, the remaining white nodes can be reached either directly or through one of the MPRs.*

In the first step, a node, node U (the green node in the diagram), selects relays from its one-hop neighbors (N1). These relays, referred to as MPR1 nodes (highlighted in blue), ensure that all isolated nodes in its

two-hop neighborhood (N2), shown in red, are reachable. For instance, node T is considered an isolated node that can only connect to node U through node H. Therefore, node H is elected as an MPR in this step, ensuring node T can reach U in two hops.

In the second step, node U focuses on covering any remaining nodes in its two-hop neighborhood that were not connected by the initial MPR selection. It considers nodes in N2 that remain uncovered, as well as one-hop neighbors not chosen in the first step, like nodes B, F, E, and D. Node U then selects additional MPRs based on their connectivity to maximize coverage. For example, node E is chosen because it can cover multiple nodes in N2 that would otherwise remain isolated. The algorithm concludes when all nodes within two hops of node U are connected, forming an efficient routing set, represented as MPR(U) = {C, E, I, H, G} in the diagram. This process ensures optimal coverage while minimizing redundant transmissions.

3D Network Topology

In the network topology illustrated in Figure 4-25, connecting lines indicate that two nodes are within communication range of each other, while the absence of a line signifies that the nodes are out of range. The specific distance requirements between nodes create a set of spatial constraints that cannot be satisfied when nodes are arranged only in the X-Y plane. To resolve this geometric limitation, the nodes are positioned in three-dimensional space, utilizing the Z-axis to achieve the required connectivity pattern. The coordinates of the nodes are given in the table below.

Table 4-29: *XYZ coordinates for the 20 nodes in this scenario*

Node ID	X	Y	Z
1 (U)	500.00	250.00	150
2	500.00	150.00	230
3	571.00	179.00	70
4	600.00	250.00	230
5	571.00	321.00	70
6	500.00	350.00	230
7	429.00	321.00	70
8	399.83	249.15	230
9	429.00	179.00	70
10	394.06	105.73	0
11	382.76	113.09	130
12	571.78	97.90	130
13	614.60	118.89	0
14	624.00	289.00	160
15	564.00	263.00	0
16	545.27	396.84	170
17	455.00	437.00	65
18	393.35	300.00	0
19	352.03	339.62	150
20	330.36	252.03	300

A 3D view of the topology is shown in Figure 4-26 below.

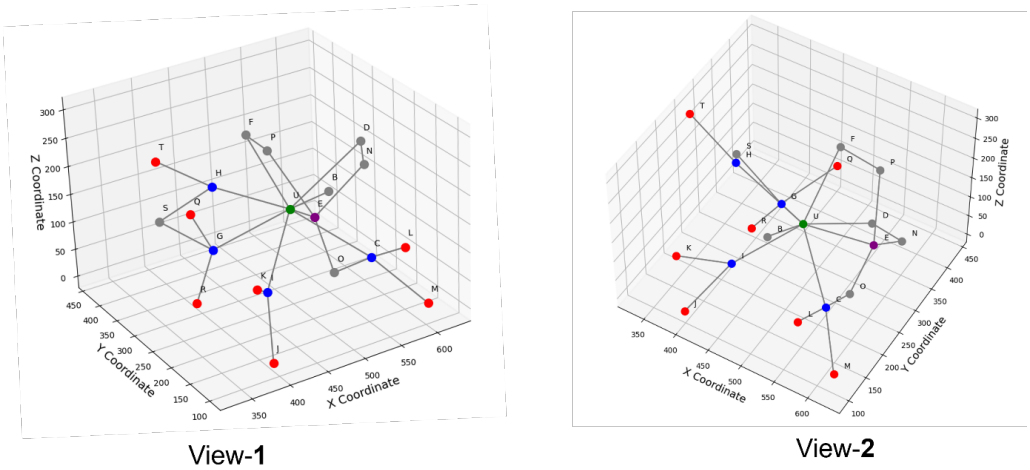


Figure 4-27: *3-D views of the topology. The range of the nodes is 130m. View-1 shows the network topology from a front right perspective, while View-2 provides a top down perspective.*

Network Setup

Open NetSim and Select Examples ▷ Mobile Ad hoc Networks ▷ MANET-OLSR then click on the tile in the middle panel to load the example as shown in below.

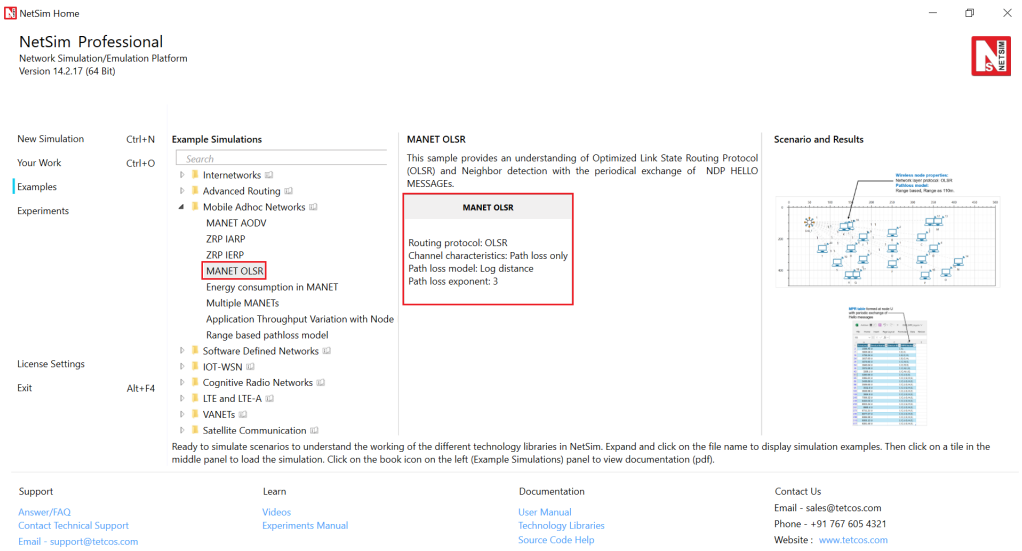


Figure 4-28: *List of scenarios for the example of MANET-OLSR*

The following steps were carried out to generate this scenario:

1. Set grid length as 1000 × 500 from grid setting property panel on the right. This needs to be done before any device is placed on the grid.
2. Create a scenario with 20 Nodes as shown in Figure 4-25.
3. Click on wireless node, expand the property panel on right and go to Position layer ▷ set Mobility model to No mobility in all nodes.
4. Set the Routing protocol to OLSR in Network layer of wireless nodes. As this is a global property, changing it on one node will affect all nodes.
5. Click on the link, expand link property on right and set the channel characteristics to pathloss and pathloss model as Range based, with range as 130m.

6. Enable OLSR Neighbor and MPR log by clicking on configure reports tab and plots as shown below.

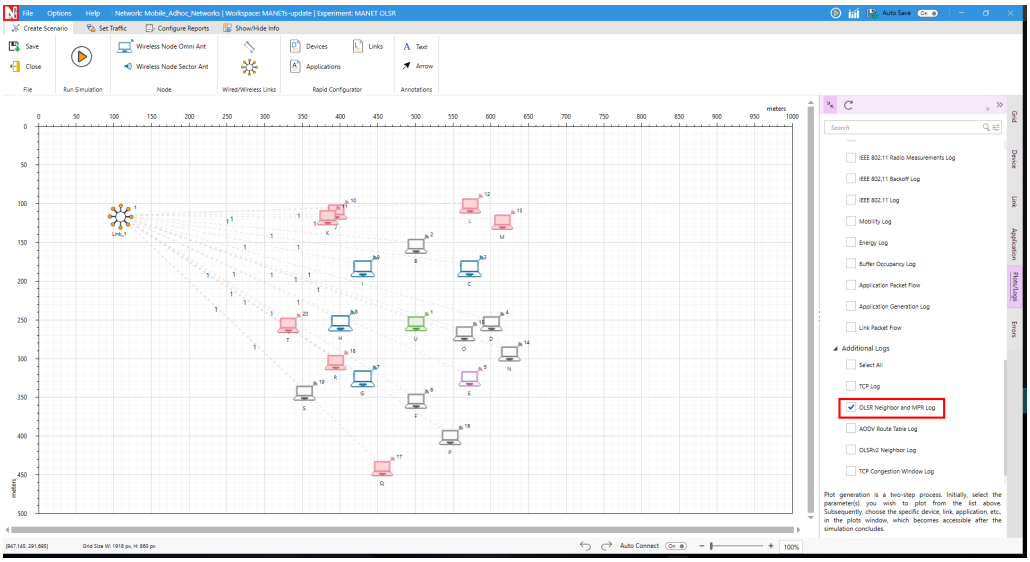
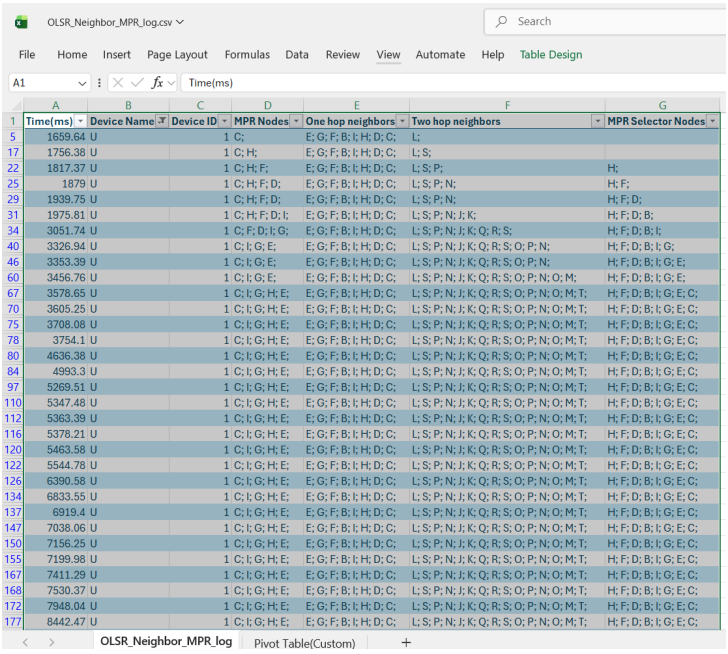


Figure 4-29: *Enabling OLSR Neighbor and MPR log*

7. Run the simulation for 50 seconds.

Results and Analysis

1. In the Results dashboard, click on logs tab and open OLSR_MPR_Log.csv file.
2. Filter the device name column to U.



Time(ms)	Device Name	Device ID	MPR Nodes	One hop neighbors	Two hop neighbors	MPR Selector Nodes
1659.64	U	1	C;	E; G; F; B; I; H; D; C;	L;	
1756.38	U	1	C; H;	E; G; F; B; I; H; D; C;	L; S;	
1817.37	U	1	C; H; F;	E; G; F; B; I; H; D; C;	L; S; P;	H;
1879	U	1	C; H; F; D;	E; G; F; B; I; H; D; C;	L; S; P; N;	H; F;
1939.75	U	1	C; H; F; D;	E; G; F; B; I; H; D; C;	L; S; P; N;	H; F; D;
1975.81	U	1	C; H; F; D; I;	E; G; F; B; I; H; D; C;	L; S; P; N; J; K;	H; F; D; B;
3051.74	U	1	C; F; D; I; G;	E; G; F; B; I; H; D; C;	L; S; P; N; J; K; Q; R; S;	H; F; D; B; I;
3326.94	U	1	C; I; G; E;	E; G; F; B; I; H; D; C;	L; S; P; N; J; K; Q; R; S; O; P; N;	H; F; D; B; I; G;
3353.39	U	1	C; I; G; E;	E; G; F; B; I; H; D; C;	L; S; P; N; J; K; Q; R; S; O; P; N;	H; F; D; B; I; G; E;
3456.76	U	1	C; I; G; E;	E; G; F; B; I; H; D; C;	L; S; P; N; J; K; Q; R; S; O; P; N; O; M;	H; F; D; B; I; G; E;
3578.65	U	1	C; I; G; H; E;	E; G; F; B; I; H; D; C;	L; S; P; N; J; K; Q; R; S; O; P; N; O; M; T;	H; F; D; B; I; G; E; C;
3605.25	U	1	C; I; G; H; E;	E; G; F; B; I; H; D; C;	L; S; P; N; J; K; Q; R; S; O; P; N; O; M; T;	H; F; D; B; I; G; E; C;
3708.08	U	1	C; I; G; H; E;	E; G; F; B; I; H; D; C;	L; S; P; N; J; K; Q; R; S; O; P; N; O; M; T;	H; F; D; B; I; G; E; C;
3754.1	U	1	C; I; G; H; E;	E; G; F; B; I; H; D; C;	L; S; P; N; J; K; Q; R; S; O; P; N; O; M; T;	H; F; D; B; I; G; E; C;
4636.38	U	1	C; I; G; H; E;	E; G; F; B; I; H; D; C;	L; S; P; N; J; K; Q; R; S; O; P; N; O; M; T;	H; F; D; B; I; G; E; C;
4993.3	U	1	C; I; G; H; E;	E; G; F; B; I; H; D; C;	L; S; P; N; J; K; Q; R; S; O; P; N; O; M; T;	H; F; D; B; I; G; E; C;
5269.51	U	1	C; I; G; H; E;	E; G; F; B; I; H; D; C;	L; S; P; N; J; K; Q; R; S; O; P; N; O; M; T;	H; F; D; B; I; G; E; C;
5347.48	U	1	C; I; G; H; E;	E; G; F; B; I; H; D; C;	L; S; P; N; J; K; Q; R; S; O; P; N; O; M; T;	H; F; D; B; I; G; E; C;
5363.39	U	1	C; I; G; H; E;	E; G; F; B; I; H; D; C;	L; S; P; N; J; K; Q; R; S; O; P; N; O; M; T;	H; F; D; B; I; G; E; C;
5378.21	U	1	C; I; G; H; E;	E; G; F; B; I; H; D; C;	L; S; P; N; J; K; Q; R; S; O; P; N; O; M; T;	H; F; D; B; I; G; E; C;
5463.58	U	1	C; I; G; H; E;	E; G; F; B; I; H; D; C;	L; S; P; N; J; K; Q; R; S; O; P; N; O; M; T;	H; F; D; B; I; G; E; C;
5544.78	U	1	C; I; G; H; E;	E; G; F; B; I; H; D; C;	L; S; P; N; J; K; Q; R; S; O; P; N; O; M; T;	H; F; D; B; I; G; E; C;
6390.58	U	1	C; I; G; H; E;	E; G; F; B; I; H; D; C;	L; S; P; N; J; K; Q; R; S; O; P; N; O; M; T;	H; F; D; B; I; G; E; C;
6833.55	U	1	C; I; G; H; E;	E; G; F; B; I; H; D; C;	L; S; P; N; J; K; Q; R; S; O; P; N; O; M; T;	H; F; D; B; I; G; E; C;
6919.4	U	1	C; I; G; H; E;	E; G; F; B; I; H; D; C;	L; S; P; N; J; K; Q; R; S; O; P; N; O; M; T;	H; F; D; B; I; G; E; C;
7038.06	U	1	C; I; G; H; E;	E; G; F; B; I; H; D; C;	L; S; P; N; J; K; Q; R; S; O; P; N; O; M; T;	H; F; D; B; I; G; E; C;
7156.25	U	1	C; I; G; H; E;	E; G; F; B; I; H; D; C;	L; S; P; N; J; K; Q; R; S; O; P; N; O; M; T;	H; F; D; B; I; G; E; C;
7199.98	U	1	C; I; G; H; E;	E; G; F; B; I; H; D; C;	L; S; P; N; J; K; Q; R; S; O; P; N; O; M; T;	H; F; D; B; I; G; E; C;
7411.29	U	1	C; I; G; H; E;	E; G; F; B; I; H; D; C;	L; S; P; N; J; K; Q; R; S; O; P; N; O; M; T;	H; F; D; B; I; G; E; C;
7530.37	U	1	C; I; G; H; E;	E; G; F; B; I; H; D; C;	L; S; P; N; J; K; Q; R; S; O; P; N; O; M; T;	H; F; D; B; I; G; E; C;
7948.04	U	1	C; I; G; H; E;	E; G; F; B; I; H; D; C;	L; S; P; N; J; K; Q; R; S; O; P; N; O; M; T;	H; F; D; B; I; G; E; C;
8442.47	U	1	C; I; G; H; E;	E; G; F; B; I; H; D; C;	L; S; P; N; J; K; Q; R; S; O; P; N; O; M; T;	H; F; D; B; I; G; E; C;

Figure 4-30: *MPR table formed at node U*

- Initially, U starts with a single MPR node, "C," at 1659.64 ms. Over time, more nodes are added to the MPR list (e.g., "H", "F", "D", "I", "G", "E"), with some nodes ("F", "D") being dropped and "H" being dropped and added again.
- We finally get the set of MPR nodes as "C; I; G; H; E".

4.5.2 Part-2: NDP and TC Messages in OLSR

Introduction

OLSR Hello Messages (NDP)

The OLSR Hello messages, part of the Neighbor Discovery Protocol (NDP), are exchanged between directly connected nodes. They enable each node to detect its one-hop and two-hop neighbors. These messages are used to establish and maintain neighbor relations, which are necessary for building an updated and accurate network topology. Hello messages include information about link status and the willingness of a node to participate in forwarding traffic for its neighbors. This process helps in selecting Multipoint Relays (MPRs), which minimize the number of transmissions required for routing updates.

OLSR TC Messages

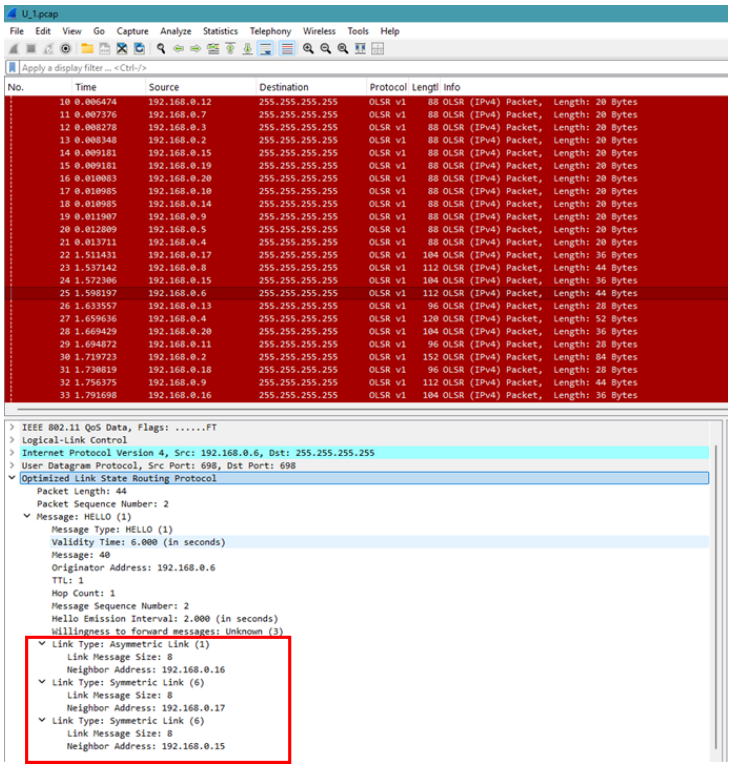


Figure 4-31: *OLSR Hello Message Broadcasting all Its Neighbors information with associated Link Type*

Network Setup

1. Consider the same scenario as Part -1.
2. Click on wireless node U, expand property panel on right and go to General Layer Set Wireshark Capture to Online.
3. Go to Configure Reports tab → Enable Packet trace.
4. Run the simulation for 50 seconds.

Results and Analysis

Wireshark is generated during the simulation of network.

The periodic exchange of Hello messages in OLSR dynamically updates the MPR (Multi-Point Relay) list of a device. You can observe this in the OLSR_MPR_Log.csv by filtering the Device Name column to "U."

Neighbor Discovery Process (NDP) or Hello Messages

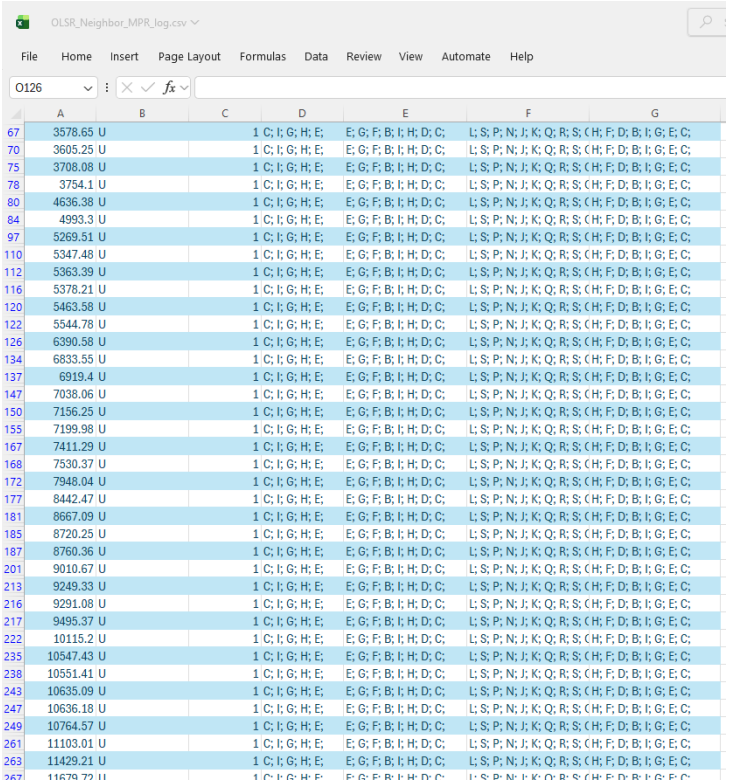
See Figure 4-32.

Initial MPR Selection: At 1659.64 ms, the device starts with a single MPR node, “C.” This means “C” was initially the best choice for relaying messages.

MPR List Evolution: Over time, more nodes like “H,” “F,” “D,” “I,” “G,” and “E” are added to the list, and “F” and “D” are dropped again. The final set is reached at 3578.65 ms. This indicates the device is adapting to network changes, selecting new relays as network conditions evolve.

Dynamic Changes: Nodes are added or removed from the MPR list based on their current efficiency in relaying messages, reflecting ongoing adjustments to the network topology.

- Multiple MPR Nodes: We finally get the set of MPR nodes as “C; I; G; H; E”.



	A	B	C	D	E	F	G
67	3578.65	U		1	C; I; G; H; E;	E; G; F; B; I; H; D; C;	L; S; P; N; J; K; Q; R; S; C H; F; D; B; I; G; E; C;
70	3605.25	U		1	C; I; G; H; E;	E; G; F; B; I; H; D; C;	L; S; P; N; J; K; Q; R; S; C H; F; D; B; I; G; E; C;
75	3708.08	U		1	C; I; G; H; E;	E; G; F; B; I; H; D; C;	L; S; P; N; J; K; Q; R; S; C H; F; D; B; I; G; E; C;
78	3754.1	U		1	C; I; G; H; E;	E; G; F; B; I; H; D; C;	L; S; P; N; J; K; Q; R; S; C H; F; D; B; I; G; E; C;
80	4636.38	U		1	C; I; G; H; E;	E; G; F; B; I; H; D; C;	L; S; P; N; J; K; Q; R; S; C H; F; D; B; I; G; E; C;
84	4993.3	U		1	C; I; G; H; E;	E; G; F; B; I; H; D; C;	L; S; P; N; J; K; Q; R; S; C H; F; D; B; I; G; E; C;
97	5269.51	U		1	C; I; G; H; E;	E; G; F; B; I; H; D; C;	L; S; P; N; J; K; Q; R; S; C H; F; D; B; I; G; E; C;
110	5347.48	U		1	C; I; G; H; E;	E; G; F; B; I; H; D; C;	L; S; P; N; J; K; Q; R; S; C H; F; D; B; I; G; E; C;
112	5363.39	U		1	C; I; G; H; E;	E; G; F; B; I; H; D; C;	L; S; P; N; J; K; Q; R; S; C H; F; D; B; I; G; E; C;
116	5378.21	U		1	C; I; G; H; E;	E; G; F; B; I; H; D; C;	L; S; P; N; J; K; Q; R; S; C H; F; D; B; I; G; E; C;
120	5463.58	U		1	C; I; G; H; E;	E; G; F; B; I; H; D; C;	L; S; P; N; J; K; Q; R; S; C H; F; D; B; I; G; E; C;
122	5544.78	U		1	C; I; G; H; E;	E; G; F; B; I; H; D; C;	L; S; P; N; J; K; Q; R; S; C H; F; D; B; I; G; E; C;
126	6390.58	U		1	C; I; G; H; E;	E; G; F; B; I; H; D; C;	L; S; P; N; J; K; Q; R; S; C H; F; D; B; I; G; E; C;
134	6833.55	U		1	C; I; G; H; E;	E; G; F; B; I; H; D; C;	L; S; P; N; J; K; Q; R; S; C H; F; D; B; I; G; E; C;
137	6919.4	U		1	C; I; G; H; E;	E; G; F; B; I; H; D; C;	L; S; P; N; J; K; Q; R; S; C H; F; D; B; I; G; E; C;
147	7038.06	U		1	C; I; G; H; E;	E; G; F; B; I; H; D; C;	L; S; P; N; J; K; Q; R; S; C H; F; D; B; I; G; E; C;
150	7156.25	U		1	C; I; G; H; E;	E; G; F; B; I; H; D; C;	L; S; P; N; J; K; Q; R; S; C H; F; D; B; I; G; E; C;
155	7199.98	U		1	C; I; G; H; E;	E; G; F; B; I; H; D; C;	L; S; P; N; J; K; Q; R; S; C H; F; D; B; I; G; E; C;
167	7411.29	U		1	C; I; G; H; E;	E; G; F; B; I; H; D; C;	L; S; P; N; J; K; Q; R; S; C H; F; D; B; I; G; E; C;
168	7530.37	U		1	C; I; G; H; E;	E; G; F; B; I; H; D; C;	L; S; P; N; J; K; Q; R; S; C H; F; D; B; I; G; E; C;
172	7948.04	U		1	C; I; G; H; E;	E; G; F; B; I; H; D; C;	L; S; P; N; J; K; Q; R; S; C H; F; D; B; I; G; E; C;
177	8442.47	U		1	C; I; G; H; E;	E; G; F; B; I; H; D; C;	L; S; P; N; J; K; Q; R; S; C H; F; D; B; I; G; E; C;
181	8667.09	U		1	C; I; G; H; E;	E; G; F; B; I; H; D; C;	L; S; P; N; J; K; Q; R; S; C H; F; D; B; I; G; E; C;
185	8720.25	U		1	C; I; G; H; E;	E; G; F; B; I; H; D; C;	L; S; P; N; J; K; Q; R; S; C H; F; D; B; I; G; E; C;
187	8760.36	U		1	C; I; G; H; E;	E; G; F; B; I; H; D; C;	L; S; P; N; J; K; Q; R; S; C H; F; D; B; I; G; E; C;
201	9010.67	U		1	C; I; G; H; E;	E; G; F; B; I; H; D; C;	L; S; P; N; J; K; Q; R; S; C H; F; D; B; I; G; E; C;
213	9249.33	U		1	C; I; G; H; E;	E; G; F; B; I; H; D; C;	L; S; P; N; J; K; Q; R; S; C H; F; D; B; I; G; E; C;
216	9291.08	U		1	C; I; G; H; E;	E; G; F; B; I; H; D; C;	L; S; P; N; J; K; Q; R; S; C H; F; D; B; I; G; E; C;
217	9495.37	U		1	C; I; G; H; E;	E; G; F; B; I; H; D; C;	L; S; P; N; J; K; Q; R; S; C H; F; D; B; I; G; E; C;
222	10115.2	U		1	C; I; G; H; E;	E; G; F; B; I; H; D; C;	L; S; P; N; J; K; Q; R; S; C H; F; D; B; I; G; E; C;
235	10547.43	U		1	C; I; G; H; E;	E; G; F; B; I; H; D; C;	L; S; P; N; J; K; Q; R; S; C H; F; D; B; I; G; E; C;
238	10551.41	U		1	C; I; G; H; E;	E; G; F; B; I; H; D; C;	L; S; P; N; J; K; Q; R; S; C H; F; D; B; I; G; E; C;
243	10635.09	U		1	C; I; G; H; E;	E; G; F; B; I; H; D; C;	L; S; P; N; J; K; Q; R; S; C H; F; D; B; I; G; E; C;
247	10636.18	U		1	C; I; G; H; E;	E; G; F; B; I; H; D; C;	L; S; P; N; J; K; Q; R; S; C H; F; D; B; I; G; E; C;
249	10764.57	U		1	C; I; G; H; E;	E; G; F; B; I; H; D; C;	L; S; P; N; J; K; Q; R; S; C H; F; D; B; I; G; E; C;
261	11103.01	U		1	C; I; G; H; E;	E; G; F; B; I; H; D; C;	L; S; P; N; J; K; Q; R; S; C H; F; D; B; I; G; E; C;
263	11429.21	U		1	C; I; G; H; E;	E; G; F; B; I; H; D; C;	L; S; P; N; J; K; Q; R; S; C H; F; D; B; I; G; E; C;

Figure 4-32: *MPR table formed at node U after periodic exchange of Hello messages.*

Topology Control (TC) Messages

TC (Topology Control) messages are periodically exchanged to inform nodes about the network’s topology. Each node updates its routing table based on the information in these messages, which include details about MPRs (Multi-Point Relays) and their connections.

When a node receives a TC message, it uses the link information to add or adjust routes in its table, ensuring it always has the most efficient paths to other nodes. This dynamic updating keeps the network adaptive to changes, such as new routes or broken links.

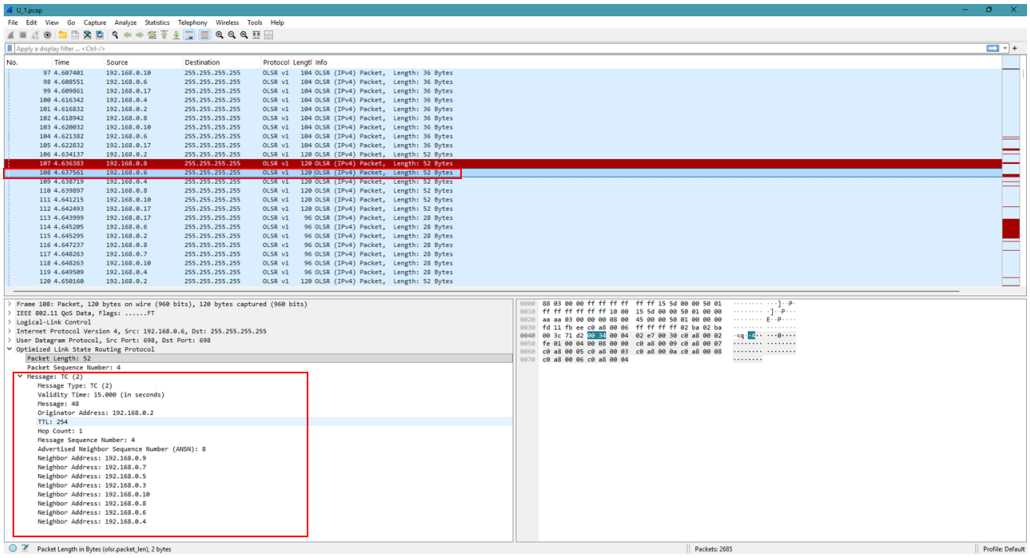
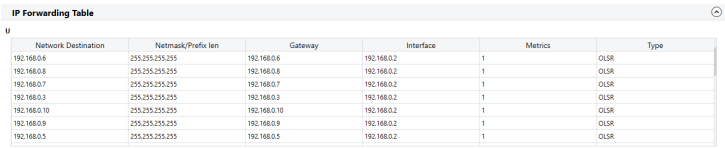


Figure 4-33: *TC Message Broadcasted from node to all its Neighbors, so that other nodes in the network can use this information to update their own routing tables.*



IP Forwarding Table

U

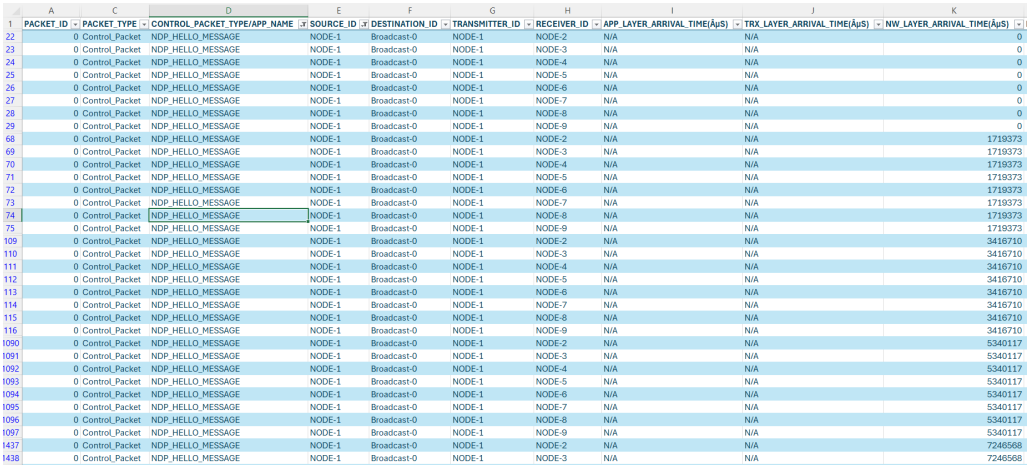
Network Destination	Netmask/Prefix len	Gateway	Interface	Metrics	Type
192.168.0.6	255.255.255.255	192.168.0.6	192.168.0.2	1	OLSR
192.168.0.8	255.255.255.255	192.168.0.8	192.168.0.2	1	OLSR
192.168.0.7	255.255.255.255	192.168.0.7	192.168.0.2	1	OLSR
192.168.0.3	255.255.255.255	192.168.0.3	192.168.0.2	1	OLSR
192.168.0.10	255.255.255.255	192.168.0.10	192.168.0.2	1	OLSR
192.168.0.9	255.255.255.255	192.168.0.9	192.168.0.2	1	OLSR
192.168.0.5	255.255.255.255	192.168.0.5	192.168.0.2	1	OLSR

Figure 4-34: *Routing Table of Node U*

The routing table shown in Figure 4-34 reflects these updates, displaying the optimal routes formed from the latest TC message exchanges.

How often are NDP Hello Messages and TC Messages exchanged in the Network?

Open Packet trace from simulation results window and filter Control Packet Type/App Name to NDP HELLO MESSAGE, Source ID to Node-1. All the nodes in the network exchange the NDP HELLO MESSAGEs periodically (for every 2 seconds—check NETWORK LAYER ARRIVAL TIME column) to detect the neighbors. In the below screenshot, Node-1 is broadcasting NDP HELLO MESSAGES to its neighbors 2, 3, 4, 5, 6, 7, 8 and 9 for every 2 seconds as shown below.



	A	C	D	E	F	G	H	I	J	K
1	PACKET_ID	PACKET_TYPE	CONTROL_PACKET_TYPE/APP_NAME	SOURCE_ID	DESTINATION_ID	TRANSMITTER_ID	RECEIVER_ID	APP_LAYER_ARRIVAL_TIME(ÂµS)	TRX_LAYER_ARRIVAL_TIME(ÂµS)	NW_LAYER_ARRIVAL_TIME(ÂµS)
22	0	Control_Packet	NDP_HELLO_MESSAGE	NODE-1	Broadcast-0	NODE-1	NODE-2	N/A	N/A	0
23	0	Control_Packet	NDP_HELLO_MESSAGE	NODE-1	Broadcast-0	NODE-1	NODE-3	N/A	N/A	0
24	0	Control_Packet	NDP_HELLO_MESSAGE	NODE-1	Broadcast-0	NODE-1	NODE-4	N/A	N/A	0
25	0	Control_Packet	NDP_HELLO_MESSAGE	NODE-1	Broadcast-0	NODE-1	NODE-5	N/A	N/A	0
26	0	Control_Packet	NDP_HELLO_MESSAGE	NODE-1	Broadcast-0	NODE-1	NODE-6	N/A	N/A	0
27	0	Control_Packet	NDP_HELLO_MESSAGE	NODE-1	Broadcast-0	NODE-1	NODE-7	N/A	N/A	0
28	0	Control_Packet	NDP_HELLO_MESSAGE	NODE-1	Broadcast-0	NODE-1	NODE-8	N/A	N/A	0
29	0	Control_Packet	NDP_HELLO_MESSAGE	NODE-1	Broadcast-0	NODE-1	NODE-9	N/A	N/A	0
68	0	Control_Packet	NDP_HELLO_MESSAGE	NODE-1	Broadcast-0	NODE-1	NODE-2	N/A	N/A	1719373
69	0	Control_Packet	NDP_HELLO_MESSAGE	NODE-1	Broadcast-0	NODE-1	NODE-3	N/A	N/A	1719373
70	0	Control_Packet	NDP_HELLO_MESSAGE	NODE-1	Broadcast-0	NODE-1	NODE-4	N/A	N/A	1719373
71	0	Control_Packet	NDP_HELLO_MESSAGE	NODE-1	Broadcast-0	NODE-1	NODE-5	N/A	N/A	1719373
72	0	Control_Packet	NDP_HELLO_MESSAGE	NODE-1	Broadcast-0	NODE-1	NODE-6	N/A	N/A	1719373
73	0	Control_Packet	NDP_HELLO_MESSAGE	NODE-1	Broadcast-0	NODE-1	NODE-7	N/A	N/A	1719373
74	0	Control_Packet	NDP_HELLO_MESSAGE	NODE-1	Broadcast-0	NODE-1	NODE-8	N/A	N/A	1719373
75	0	Control_Packet	NDP_HELLO_MESSAGE	NODE-1	Broadcast-0	NODE-1	NODE-9	N/A	N/A	1719373
109	0	Control_Packet	NDP_HELLO_MESSAGE	NODE-1	Broadcast-0	NODE-1	NODE-2	N/A	N/A	3416710
110	0	Control_Packet	NDP_HELLO_MESSAGE	NODE-1	Broadcast-0	NODE-1	NODE-3	N/A	N/A	3416710
111	0	Control_Packet	NDP_HELLO_MESSAGE	NODE-1	Broadcast-0	NODE-1	NODE-4	N/A	N/A	3416710
112	0	Control_Packet	NDP_HELLO_MESSAGE	NODE-1	Broadcast-0	NODE-1	NODE-5	N/A	N/A	3416710
113	0	Control_Packet	NDP_HELLO_MESSAGE	NODE-1	Broadcast-0	NODE-1	NODE-6	N/A	N/A	3416710
114	0	Control_Packet	NDP_HELLO_MESSAGE	NODE-1	Broadcast-0	NODE-1	NODE-7	N/A	N/A	3416710
115	0	Control_Packet	NDP_HELLO_MESSAGE	NODE-1	Broadcast-0	NODE-1	NODE-8	N/A	N/A	3416710
116	0	Control_Packet	NDP_HELLO_MESSAGE	NODE-1	Broadcast-0	NODE-1	NODE-9	N/A	N/A	3416710
1090	0	Control_Packet	NDP_HELLO_MESSAGE	NODE-1	Broadcast-0	NODE-1	NODE-2	N/A	N/A	5340117
1091	0	Control_Packet	NDP_HELLO_MESSAGE	NODE-1	Broadcast-0	NODE-1	NODE-3	N/A	N/A	5340117
1092	0	Control_Packet	NDP_HELLO_MESSAGE	NODE-1	Broadcast-0	NODE-1	NODE-4	N/A	N/A	5340117
1093	0	Control_Packet	NDP_HELLO_MESSAGE	NODE-1	Broadcast-0	NODE-1	NODE-5	N/A	N/A	5340117
1094	0	Control_Packet	NDP_HELLO_MESSAGE	NODE-1	Broadcast-0	NODE-1	NODE-6	N/A	N/A	5340117
1095	0	Control_Packet	NDP_HELLO_MESSAGE	NODE-1	Broadcast-0	NODE-1	NODE-7	N/A	N/A	5340117
1096	0	Control_Packet	NDP_HELLO_MESSAGE	NODE-1	Broadcast-0	NODE-1	NODE-8	N/A	N/A	5340117
1097	0	Control_Packet	NDP_HELLO_MESSAGE	NODE-1	Broadcast-0	NODE-1	NODE-9	N/A	N/A	5340117
1437	0	Control_Packet	NDP_HELLO_MESSAGE	NODE-1	Broadcast-0	NODE-1	NODE-2	N/A	N/A	7246568
1438	0	Control_Packet	NDP_HELLO_MESSAGE	NODE-1	Broadcast-0	NODE-1	NODE-3	N/A	N/A	7246568

Figure 4-35: *NDP HELLO MESSAGES to its neighbors 2, 3, 4, 5, 6, 7, 8 and 9 for every 2 seconds in Packet Trace.*

You can observe in the packet trace that hello packets are exchanged every 1.7 seconds. This is obtained from

Actual Message Interval = Message Interval – Jitter

This is applicable for TC messages as well.

Now filter Control Packet Type/App Name to OLSR TC Message, Source Id and Transmitter Id to Node-1. All nodes in the network, broadcasts Topology control (TC) messages (for every 5 seconds—check Network layer arrival time column) in order to build the topology information base. In the below screenshot, node-1 is broadcasting OLSR TC Message to its neighbors 2, 3, 4, 5, 6, 7, 8 and 9 for every 5 seconds as shown below.

	A	C	D	E	F	G	H	K
1	PACKET_ID	PACKET_TYPE	CONTROL_PACKET_TYPE/APP_NAME	SOURCE_ID	DESTINATION_ID	TRANSMITTER_ID	RECEIVER_ID	NW_LAYER_ARRIVAL_TIME(ÂµS)
10	0	Control_Packet	OLSR_TC_MESSAGE	NODE-1	Broadcast-0	NODE-1	NODE-2	4633567
11	0	Control_Packet	OLSR_TC_MESSAGE	NODE-1	Broadcast-0	NODE-1	NODE-3	4633567
12	0	Control_Packet	OLSR_TC_MESSAGE	NODE-1	Broadcast-0	NODE-1	NODE-4	4633567
13	0	Control_Packet	OLSR_TC_MESSAGE	NODE-1	Broadcast-0	NODE-1	NODE-5	4633567
14	0	Control_Packet	OLSR_TC_MESSAGE	NODE-1	Broadcast-0	NODE-1	NODE-6	4633567
15	0	Control_Packet	OLSR_TC_MESSAGE	NODE-1	Broadcast-0	NODE-1	NODE-7	4633567
16	0	Control_Packet	OLSR_TC_MESSAGE	NODE-1	Broadcast-0	NODE-1	NODE-8	4633567
17	0	Control_Packet	OLSR_TC_MESSAGE	NODE-1	Broadcast-0	NODE-1	NODE-9	4633567
66	0	Control_Packet	OLSR_TC_MESSAGE	NODE-1	Broadcast-0	NODE-1	NODE-2	4649414.522
67	0	Control_Packet	OLSR_TC_MESSAGE	NODE-1	Broadcast-0	NODE-1	NODE-3	4649414.522
68	0	Control_Packet	OLSR_TC_MESSAGE	NODE-1	Broadcast-0	NODE-1	NODE-4	4649414.522
69	0	Control_Packet	OLSR_TC_MESSAGE	NODE-1	Broadcast-0	NODE-1	NODE-5	4649414.522
70	0	Control_Packet	OLSR_TC_MESSAGE	NODE-1	Broadcast-0	NODE-1	NODE-6	4649414.522
71	0	Control_Packet	OLSR_TC_MESSAGE	NODE-1	Broadcast-0	NODE-1	NODE-7	4649414.522
72	0	Control_Packet	OLSR_TC_MESSAGE	NODE-1	Broadcast-0	NODE-1	NODE-8	4649414.522
73	0	Control_Packet	OLSR_TC_MESSAGE	NODE-1	Broadcast-0	NODE-1	NODE-9	4649414.522
20	0	Control_Packet	OLSR_TC_MESSAGE	NODE-1	Broadcast-0	NODE-1	NODE-2	4874325
21	0	Control_Packet	OLSR_TC_MESSAGE	NODE-1	Broadcast-0	NODE-1	NODE-3	4874325
22	0	Control_Packet	OLSR_TC_MESSAGE	NODE-1	Broadcast-0	NODE-1	NODE-4	4874325
23	0	Control_Packet	OLSR_TC_MESSAGE	NODE-1	Broadcast-0	NODE-1	NODE-5	4874325
24	0	Control_Packet	OLSR_TC_MESSAGE	NODE-1	Broadcast-0	NODE-1	NODE-6	4874325
25	0	Control_Packet	OLSR_TC_MESSAGE	NODE-1	Broadcast-0	NODE-1	NODE-7	4874325
26	0	Control_Packet	OLSR_TC_MESSAGE	NODE-1	Broadcast-0	NODE-1	NODE-8	4874325
27	0	Control_Packet	OLSR_TC_MESSAGE	NODE-1	Broadcast-0	NODE-1	NODE-9	4874325
06	0	Control_Packet	OLSR_TC_MESSAGE	NODE-1	Broadcast-0	NODE-1	NODE-2	4982255.694
07	0	Control_Packet	OLSR_TC_MESSAGE	NODE-1	Broadcast-0	NODE-1	NODE-3	4982255.694
08	0	Control_Packet	OLSR_TC_MESSAGE	NODE-1	Broadcast-0	NODE-1	NODE-4	4982255.694

Packet Trace | Pivot Table(Custom) | +

Figure 4-36: *OLSR TC Message sent to neighbors 2, 3, 4, 5, 6, 7, 8 and 9, every 5 seconds as can be seen in packet trace*

4.6 Comparative analysis of AODV and OLSR in static and mobility scenarios

4.6.1 Introduction

Mobile Ad hoc Networks (MANETs) are self-organizing wireless networks that operate without a fixed infrastructure. The dynamic and distributed nature of MANETs introduces challenges for routing protocols, particularly in handling topology changes and maintaining efficiency under varying traffic loads. Routing protocols for MANETs are broadly categorized into reactive and proactive approaches, each with distinct mechanisms and trade-offs.

- Ad-hoc On-Demand Distance Vector (AODV): AODV is a reactive routing protocol that establishes routes only when needed. As described by Clausen et al., AODV initiates route discovery by broadcasting route request packets, followed by unicast replies for discovered routes. This approach reduces control overhead in static or low-traffic networks. However, frequent route rediscoveries caused by mobility or high traffic densities can increase overhead.
- Optimized Link State Routing (OLSR): OLSR is a proactive protocol that maintains routing tables for all nodes in the network at all times. Clausen et al. highlight OLSR's use of Multipoint Relays (MPRs) to optimize control traffic by reducing redundant retransmissions during periodic updates. While OLSR ensures immediate route availability, it generates constant control traffic regardless of traffic flow or mobility.

This example evaluates and compares the performance of AODV and OLSR under three scenarios:

- No mobility: Nodes remain stationary throughout the simulation.

- With mobility: Nodes follow a random walk mobility model, but with a fixed speed.
- Varying velocity: Nodes move at speeds ranging from 0 m/s to 100 m/s.

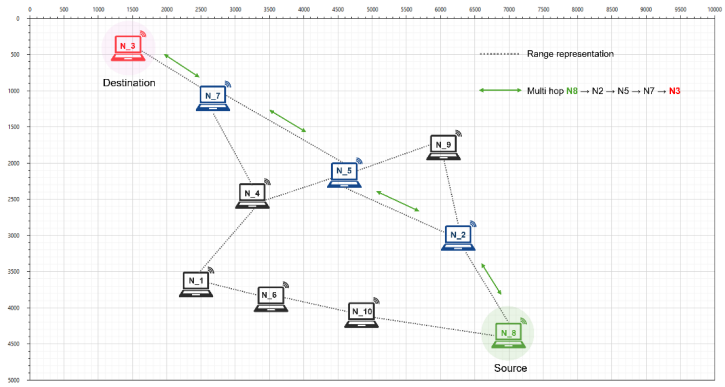


Figure 4-37: *10-Node Network Topology in MANET Simulation. The area of operation is set to* $10km \times 5km$*. Nodes communicate wirelessly using ad hoc routing protocols. The range of each node is set to* 1.7 *km and transmissions could be single hop or multi hop.*

We analyze which protocol is more resource-efficient under different network conditions, focusing on control traffic overhead, which refers to the additional network traffic generated by routing protocols during activities such as route discovery, updates, and maintenance. Since excessive control traffic can degrade performance in MANETs by consuming the already limited bandwidth, minimizing this overhead is crucial for efficient network operation. By comparing AODV and OLSR, this study provides insights into their suitability for specific scenarios, helping users make appropriate decisions based on trade-offs between scalability, bandwidth utilization, and control overhead.

4.6.2 Simulation Setup

4.6.3 Network Configuration

These settings have been pre-configured in the simulation environment.

Table 4-30: *Simulation parameters for MANET scenario*

Simulation Parameters	Value
Environment size	10000 × 5000
Mobility model	No mobility/Random walk
Velocity	10 m/s (only for Random walk)
Calculation Interval	2 second (only for Random walk)
Routing protocol	AODV/OLSR
MAC protocol	802.11 b
Transmission range	1700 m

1. Setting transmission range in NetSim: Click on the link connecting the nodes and open the link properties by navigating to the medium property in the link properties panel (RHS).
2. Set channel characteristics to pathloss, choose Range based for the pathloss model, and enter 1700 m in the Range (m) field.

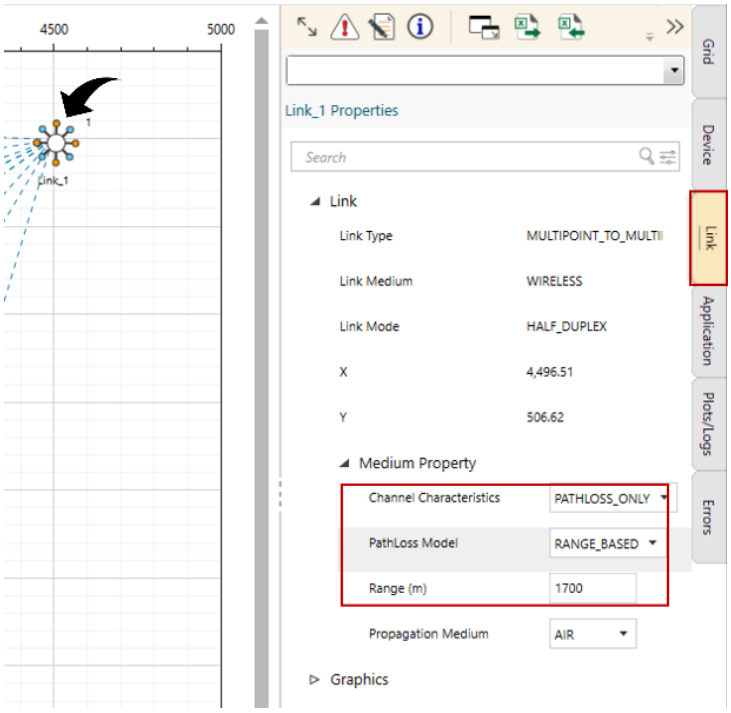


Figure 4-38: *Configuring transmission range in NetSim link properties.*

3. Enable packet trace from the configure reports tab in ribbon on top.
4. Run the simulation for 50 seconds.

4.6.4 Scenario Setup

For both AODV and OLSR, we simulate 30 cases with varying numbers of applications. Each case increases the number of applications by one, starting from a single application in Case 1 and going up to 30 applications in Case 30.

Repetition of Simulations

To reduce the impact of randomness, each scenario (AODV and OLSR) is repeated with three sets of simulations:

- Set 1: Initial set of source-destination pairs.
- Set 2: Second set of source-destination pairs.
- Set 3: Third set of source-destination pairs.

The final results are the averaged control traffic overhead across these three sets.

In the NetSim GUI, we provide configuration files for three cases: mobility, No-mobility, and velocity. The dataset includes:

- Mobility Cases:
  - Set 1: 30 cases of AODV and OLSR.
- No-Mobility Cases:
  - Set 1: 30 cases of AODV and OLSR.

Velocity Cases:

- Set 1: 10 cases of AODV and OLSR across varying velocities.
- The complete set of configuration files is available for download: GitHub Zip folder download link: https://github.com/NetSim-TETCOS/Comparision-Analysis-AODV-vs-OLSR-v15.1/archive/refs/heads/main.zip
- Click on the link above to download the comparative analysis examples.
- Extract the zip folder. The extracted project folder consists of case files for mobility, No-mobility, and velocity cases.
- Instructions for importing the workspace are provided in section 4.9.2 of the NetSim user manual.

4.6.5 Steps for Control Traffic Overhead Calculation

Running the Simulation

- Open NetSim and Select Examples→ Mobile Ad hoc Networks → Comparative Study of AODV and OLSR

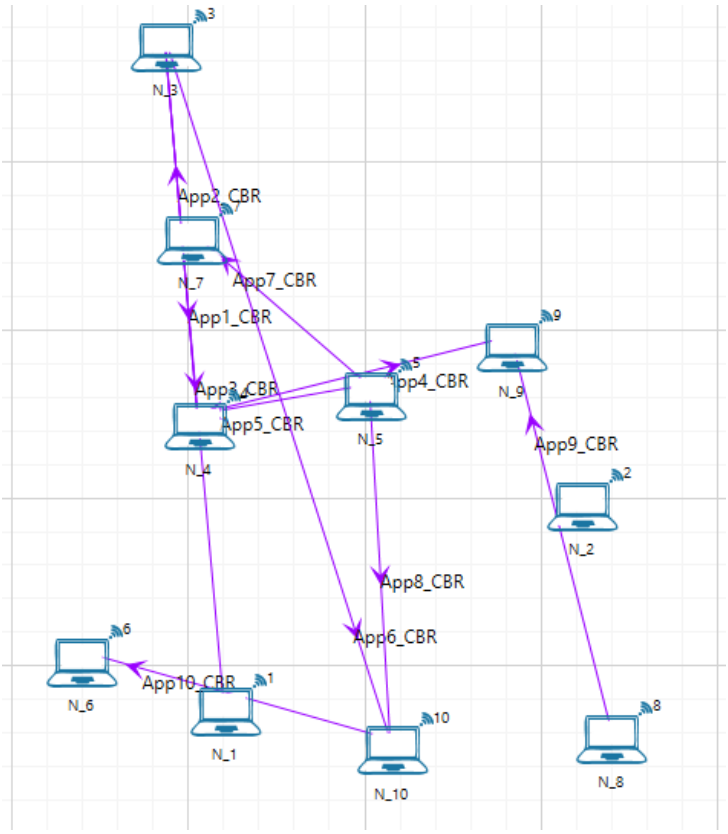


Figure 4-39: *Network Scenario. With 10 wireless nodes configured with 10 traffic flows with each other*

- For this example, Open the No-mobility case-10 scenario (e.g., No-mobility case ▷ Set-1 ▷ AODV ▷ case-10), which is configured with 10 applications.
- Run the simulation for 50 seconds.

Accessing the Packet Trace

- After each simulation is completed, open the packet trace file generated from the result dashboard.

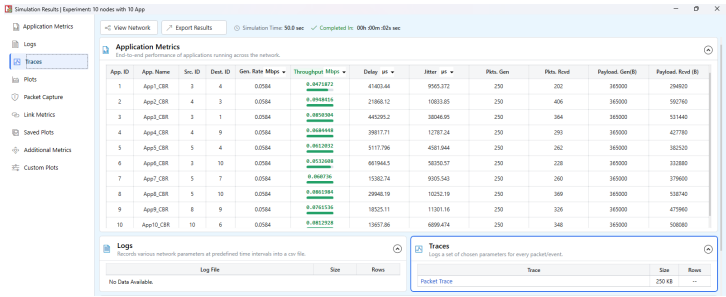


Figure 4-40: *Opening packet trace from simulation results window*

This file contains detailed logs for each packet transmitted in the network.

Filtering Control Packets for AODV

- In this step, we filter AODV control packets specifically to calculate the control traffic overhead accurately:

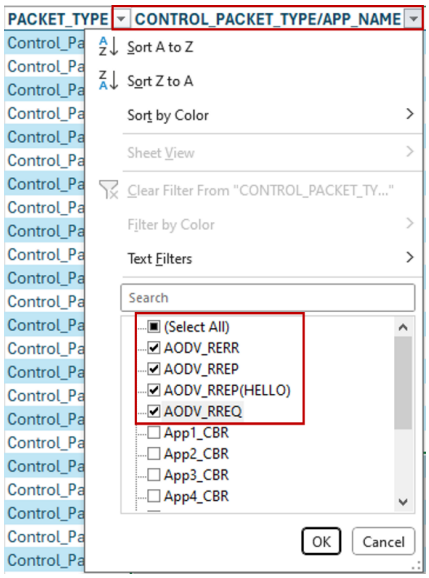


Figure 4-41: *Filtering AODV Control Packets in Packet Trace*

- Filtering by Control Packet Type: We select only rows where Control Packet Type/App Name specifies AODV. This isolates the AODV-specific control traffic data.

- Select the filtered (visible) rows only, copy them, and paste them as values into a new sheet. Working from this copy, rather than editing the trace in place, avoids misaligning columns partway through the file.

Removing Duplicate Entries

Broadcast control packets, used by AODV/OLSR during route discovery, are transmitted to all neighbouring nodes within range. In all these cases the same packet is received and logged by multiple nodes. Since each neighbor logs the packet, the same packet(s) appears multiple times in the data, however the packet(s) was transmitted only once by the sender.

CONTROL_PACKET_TYPE/APP_NAME	SOURCE_ID	DESTINATION_ID	TRANSMITTER_ID	RECEIVER_ID	PHY_LAYER_START_TIME(ÂµS)
AODV_RREQ	NODE-4	Broadcast-0	NODE-4	NODE-5	1054
AODV_RREQ	NODE-4	Broadcast-0	NODE-4	NODE-6	1054
AODV_RREQ	NODE-4	Broadcast-0	NODE-4	NODE-7	1054
AODV_RREQ	NODE-8	Broadcast-0	NODE-8	NODE-2	2128.11
AODV_RREQ	NODE-8	Broadcast-0	NODE-8	NODE-10	2128.11
AODV_RREQ	NODE-3	Broadcast-0	NODE-3	NODE-7	3162.22
AODV_RREQ	NODE-5	Broadcast-0	NODE-5	NODE-2	4136.33
AODV_RREQ	NODE-5	Broadcast-0	NODE-5	NODE-4	4136.33

Figure 4-42: *Observing the phy layer start time from packet trace*

- For example, a packet sent by Node-9 may be received by multiple nodes (Node-2, Node-5, etc.) at the same Phy layer start time(μS).
- By keeping only unique entries based on Packet Id and Phy layer start time(μS), we avoid counting the same packet multiple times, which would otherwise inflate the overhead.

- Steps to remove Duplicate packet entries.
- Go to Data ▷ Remove Duplicates in your spreadsheet software.
- In the Remove Duplicates dialog:

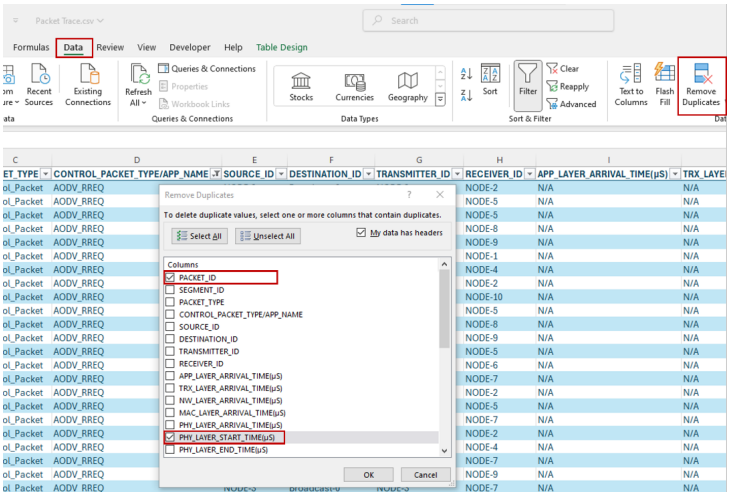
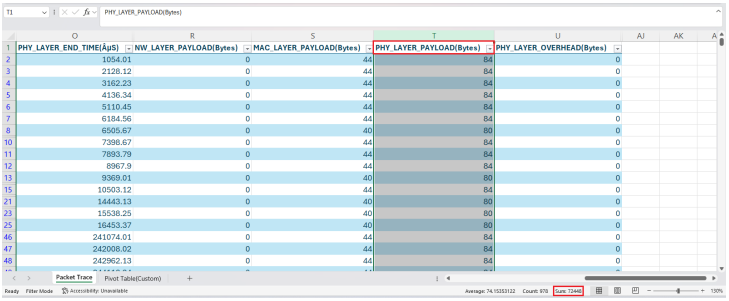


Figure 4-43: *Removing Duplicate AODV Control Packets Based on Packet ID and Start Time*

- Unselect all columns.
- Select only the columns Packet Id and Phy layer start time(μS).
- This will remove duplicate AODV control packets based on packet ID and start time.

Calculating the Control Traffic Overhead

In the filtered data, locate the Phy layer pay load(Bytes) column.



PHY_LAYER_END_TIME(ÂµS)	NW_LAYER_PAYLOAD(Bytes)	MAC_LAYER_PAYLOAD(Bytes)	PHY_LAYER_PAYLOAD(Bytes)	PHY_LAYER_OVERHEAD(Bytes)
1054.01	0	44	84	0
2128.12	0	44	84	0
3162.23	0	44	84	0
4136.34	0	44	84	0
5110.45	0	44	84	0
6184.56	0	44	84	0
6505.67	0	40	80	0
7398.67	0	44	84	0
7893.79	0	44	84	0
8967.9	0	44	84	0
9369.01	0	40	80	0
10503.12	0	44	84	0
14443.13	0	40	80	0
15538.25	0	40	80	0
16453.37	0	40	80	0
241074.01	0	44	84	0
242068.02	0	44	84	0
242962.13	0	44	84	0

Figure 4-44: *Calculating the total control traffic overhead*

- Finally, we calculate the sum of Phy layer payload (Bytes) for the unique entries. This gives the actual control traffic overhead for AODV, ensuring an accurate representation of the protocol's overhead without duplicate broadcasts.

- Record this sum as the Control traffic overhead for the current case of AODV.

Repeating for OLSR

- Configure the network layer protocol to OLSR instead of AODV.
- Follow steps from 4.5.5.3 to 4.5.5.5 to calculate the control traffic overhead for each case of OLSR.
- In step 4.5.5.3, filter for OLSR packets in the Control Packet Type/App Name column.

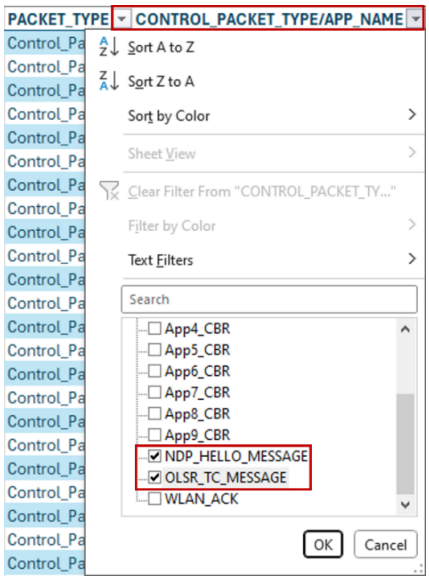


Figure 4-45: *Filtering OLSR Control Packets in Packet Trace*

- Here just filter NDP Hello Message and OLSR TC Message
- Repeat steps 4.5.5.1 to 4.5.5.5 for each case (Case 1 to Case 30) of AODV and OLSR.
- Record the overhead values.

Averaging Across Sets

- Calculate the control traffic overhead for Set 1.
- Repeat the calculations for Set 2 and Set 3.
- Average the control traffic overhead values from sets 1, 2, and 3 to obtain the final control traffic overhead for each case.
- We have used the same method to get the results for Mobility case also.

4.6.6 Results and Discussion

No-Mobility cases

The results for the No-mobility case of these three sets are as follows:

Table 4-31: *Control Traffic Overhead (Bytes) for OLSR and AODV across Three Simulation Sets (No-Mobility)*

	Set 1 (Ctrl Traffic OH)		Set 2 (Ctrl Traffic OH)		Set 3 (Ctrl Traffic OH)		Average	
No. of Traffic flows	AODV(B)	OLSR(B)	AODV(B)	OLSR(B)	AODV(B)	OLSR(B)	AODV(B)	OLSR(B)
1	25276	337744	26608	332604	25424	338508	25769	336285
2	26512	332272	27756	337432	26096	332536	26788	334080
3	29956	332100	28452	337924	28412	330964	28940	333663
4	28448	332352	27852	338800	28780	344332	28360	338495
5	31184	338072	29364	337444	29300	329504	29949	335007
6	34156	338808	31372	344048	30764	327204	32097	336687
7	36824	329824	33220	337948	31912	345092	33985	337621
8	31712	336500	37656	324372	33748	368900	34372	343257
9	35448	341080	42924	335100	36000	341600	38124	339260
10	41508	340796	46604	322212	36952	331584	41688	331531
11	51156	332928	52712	345028	38416	334984	47428	337647
12	52612	333864	56704	353120	42368	330508	50561	339164
13	48824	357500	52220	333248	51112	347320	50719	346023
14	61568	333596	71032	341244	44476	339456	59025	338099
15	53196	345812	3338444	343236	43796	347692	1145145	345580
16	3212772	330828	78944	336368	48760	347800	1113492	338332
17	69260	354124	4591508	320300	51304	328912	1570691	334445
18	70688	334332	1479268	332840	54728	306428	534895	324533
19	88004	344052	3743236	335128	59256	322376	1296832	333852
20	3494964	331564	2714064	339696	58964	331652	2089331	334304
21	81276	439124	3080404	354020	79352	365480	1080344	386208
22	96332	321508	1318024	327620	80684	337356	498347	328828
23	4329204	377064	4055504	331024	102640	332668	2829116	346919
24	3472184	305856	4234296	348820	2663884	363144	3456788	339273
25	4710804	350792	231788	308632	4568968	340712	3170520	333379
26	4662456	356240	3238128	308168	2520076	360532	3473553	341647
27	3975340	342372	1968176	358940	632632	400308	2192049	367207
28	145548	348804	4729492	409680	3807072	331168	2894037	363217
29	4795528	330848	3935808	341360	633404	373280	3121580	348496
30	4712640	322060	2959920	314228	2402416	340792	3358325	325693

This table presents the control overhead data for 30 traffic flows, averaged across three simulation sets (Set 1, Set 2, and Set 3) with unique source-destination pairs. The averaging process reduces randomness and ensures a more accurate comparison for both the no mobility and mobility scenarios.

Plot:

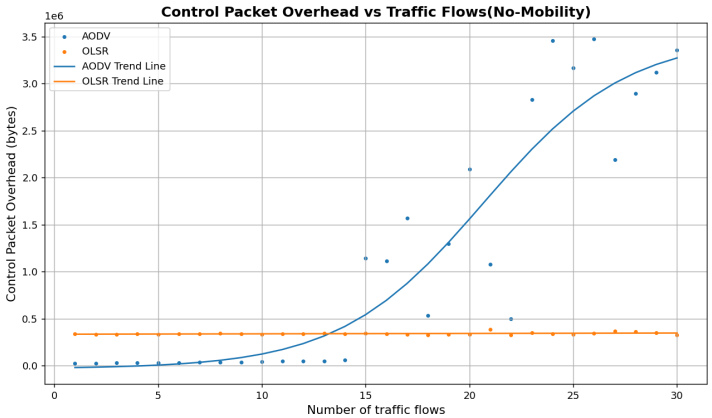


Figure 4-46: *Control Packet Overhead vs. Number of traffic flows in No-mobility case*

Mobility cases

The results for the Mobility case of these three sets are as follows:

Table 4-32: *Control Traffic Overhead (Bytes) for OLSR and AODV across Three Simulation Sets (Mobility)*

	Set 1 (Ctrl Traffic OH)		Set 2 (Ctrl Traffic OH)		Set 3 (Ctrl Traffic OH)		Average	
No. of flows	AODV(B)	OLSR(B)	AODV(B)	OLSR(B)	AODV(B)	OLSR(B)	AODV(B)	OLSR(B)
1	24796	359112	24796	359112	29540	489468	26377	402564
2	26752	341560	31180	396596	32804	420388	30245	386181
3	30036	386032	33800	344312	30808	475188	31548	401844
4	38792	339960	40928	324156	38684	434516	39468	366211
5	42876	365212	45656	438112	57996	441216	48843	414847
6	44568	358176	80892	442136	35228	422236	53563	407516
7	45124	516748	75656	517180	47976	543112	56252	525680
8	51080	488784	75308	398752	39596	405056	55328	430864
9	39368	451632	77372	334464	45416	394788	54052	393628
10	44188	545672	87204	322836	68532	390424	66641	419644
11	50924	488112	73108	448916	88312	366016	70781	434348
12	50580	437756	99428	491132	61676	451448	70561	460112
13	50020	572316	1324984	501440	3312560	469620	1562521	514459
14	52340	495428	133388	390100	70760	497212	85496	460913
15	4281644	356536	103052	356624	110044	362708	1498247	358623
16	71988	441464	120644	434644	85012	447036	92548	441048
17	84860	378196	120368	480872	4432492	401340	1545907	420136
18	89728	342416	1472148	439104	4173224	459288	1911700	413603
19	111268	383164	250280	453628	2421404	395980	927651	410924
20	235296	439260	649196	532424	3631988	343972	1505493	438552
21	3445588	391884	2988600	369680	4092480	448916	3508889	403493
22	1472268	552408	2217724	336496	4743016	611116	2811003	500007
23	363120	223388	1217496	433792	4498484	428920	2026367	362033
24	4164480	472156	1954392	441880	3203064	402104	3107312	438713
25	2368940	359488	4073800	411420	3628824	474124	3357188	415011
26	2039024	506504	2480164	346696	3988144	463688	2835777	438963
27	4125188	429776	3594640	388092	2402824	420096	3374217	412655
28	1286684	204464	4983008	328812	3500676	451236	3256789	328171
29	3220304	331768	4586016	414560	3703324	249056	3836548	331795

Table 4-32: *(continued)*

No. of flows	Set 1 (Ctrl Traffic OH)		Set 2 (Ctrl Traffic OH)		Set 3 (Ctrl Traffic OH)		Average	
	AODV(B)	OLSR(B)	AODV(B)	OLSR(B)	AODV(B)	OLSR(B)	AODV(B)	OLSR(B)
30	4522392	342076	3271620	348836	3638772	386184	3810928	359032

Plot:

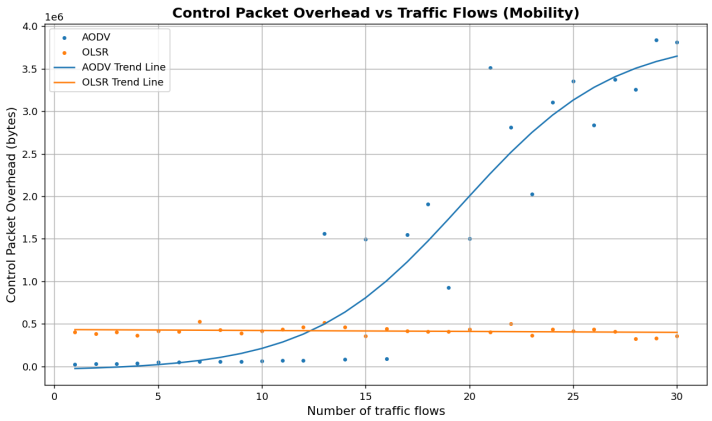


Figure 4-47: *Control packet overhead vs. Number of traffic flows in mobility case*

Discussion

AODV Control Overhead

- In both no-mobility and mobility scenarios, AODV has lower control traffic overhead than OLSR in every case with up to 12 traffic flows (about 26,000–34,000 bytes for 1–7 flows in the no-mobility case and 26,000–56,000 bytes in the mobility case, against 0.33–0.34 million bytes and 0.37–0.53 million bytes for OLSR). This is due to its reactive nature, where route discovery is triggered only when required. None of the cases with 1–7 flows has a high AODV overhead.
- With more traffic flows, the control overhead for AODV rises and varies widely between cases. In the no-mobility case, AODV overhead jumps from about 59,000 bytes at 14 flows to 1.15 million bytes at 15 flows, exceeds OLSR in every case from 15 flows onwards, and reaches about 3.5 million bytes at 26 flows. In the mobility case, AODV overhead exceeds OLSR at 13 and 15 flows and in every case from 17 flows onwards, and reaches about 3.8 million bytes at 29 flows. In the high-overhead runs the AODV control traffic consists mainly of RERR packets (for example, no-mobility Set 1 with 20 flows: 35,111 unique RERR packets, 3.37 MB of the 3.49 MB total, at about 1,000 RERR packets per second from about 15 s), so route error messages, not route requests, dominate the overhead there.

OLSR Control Overhead

- OLSR exhibits nearly constant control overhead across all traffic flows in both scenarios (about 0.32–0.39 million bytes in the no-mobility case and 0.33–0.53 million bytes in the mobility case). Its proactive nature, characterized by periodic routing updates, makes the overhead largely independent of traffic flow density.
- OLSR's overhead is higher than AODV's in every case with up to 14 flows (no-mobility) and in 14 of the first 16 cases (mobility). It is lower than AODV's in every case with 15 or more flows (no-mobility) and 17 or more flows (mobility).

Insights

The results highlight the following key observations:

- Low traffic loads: AODV has an advantage in sparse traffic scenarios (up to about 12 traffic flows), where its reactive mechanism minimizes unnecessary control overhead.
- High traffic loads: OLSR exhibits better scalability and predictability for networks with moderate to high traffic densities.
- Impact of Mobility: The overall trend is similar in the mobility and no-mobility scenarios, with traffic flow density being the dominant factor. The point at which AODV overhead rises above OLSR is about 15 flows without mobility and 13–17 flows with mobility. The AODV overhead is not monotonic in the number of flows in either scenario, so the 3-set average still varies from case to case.

4.6.7 Impact of Varying Velocity on Routing Protocol Performance

In MANETs, the velocity of nodes significantly influences the dynamics of the network topology. As nodes move faster, link breakages and re-establishments occur more frequently, posing challenges for routing protocols.

Scenario Setup

- These settings have been pre-configured in the simulation environment.

Table 4-33: *Scenario settings*

Simulation Parameters	Value
Environment Size	10000 × 5000
Mobility Model	Random Walk
Velocity	Varied from 0 m/s to 100 m/s
Calculation Interval	2 second
Routing Protocol	AODV/OLSR
MAC Protocol	802.11 b
Transmission range	1700 m

- We have considered a 10-node scenario with 10 traffic flows, where simulations were conducted in three separate sets for both AODV and OLSR.
- The control traffic overhead was calculated using the same methodology applied for the no-mobility and mobility scenarios at 4.5.5.
- The results of the simulations are presented in Table 4-34, which shows the averaged control traffic overhead for both protocols at varying velocities.

4.6.8 Velocity cases

The results for the Velocity case of these three sets are as follows:

Table 4-34: *Average control traffic overhead for AODV and OLSR under varying velocity*

	Set 1 (Ctrl Traffic OH)		Set 2 (Ctrl Traffic OH)		Set 3 (Ctrl Traffic OH)		Average	
Velocity	**AODV(B)**	**OLSR(B)**	**AODV(B)**	**OLSR(B)**	**AODV(B)**	**OLSR(B)**	**AODV(B)**	**OLSR(B)**
0	44860	433184	40432	595824	42952	447684	42748	492231
10	43836	428408	40256	508660	42852	553928	42315	496999
20	45072	458284	38856	587204	361100	513396	148343	519628
30	45892	541620	40712	320816	2769592	523700	952065	462045
40	130428	481696	416220	538768	424932	622292	323860	547585
50	76340	517228	1024256	377012	3137964	652376	1412853	515539
60	100644	412504	307452	305776	114876	335144	174324	351141
70	97752	358152	191216	349404	3853240	291908	1380736	333155
80	154396	448348	2663856	405992	2543512	357820	1787255	404053
90	128636	457136	3253468	330776	439268	344464	1273791	377459
100	1258308	421580	148656	357368	3755648	332572	1720871	370507

The averaged results from the three sets provide a comparison of the control traffic overhead for the two protocols under varying velocity conditions.

Plot:

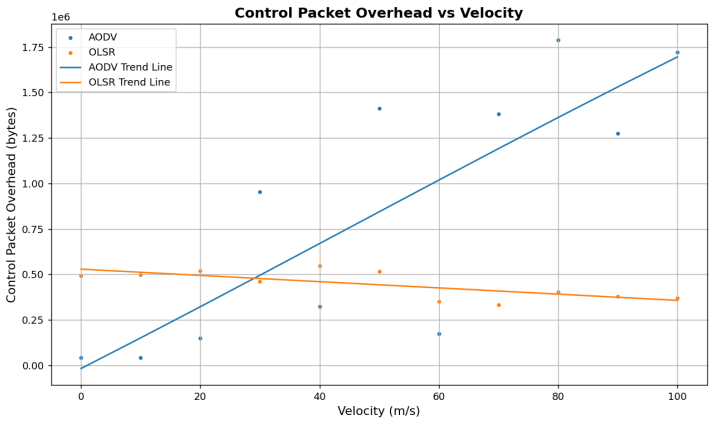


Figure 4-48: *Control traffic overhead for AODV and OLSR under varying velocity*

Discussion

AODV Control Overhead

- Low Velocity (0–30 m/s): The 3-set average AODV overhead is between about 42,000 and 148,000 bytes at 0, 10 and 20 m/s, but 30 m/s averages about 952,000 bytes because one of the three sets has a much higher overhead (2.77 million bytes). At 0 m/s all three sets stay between about 40,000 and 45,000 bytes.
- High Velocity (40–100 m/s): The average AODV overhead ranges from about 174,000 bytes (60 m/s) to 1.8 million bytes (80 m/s). The overhead varies strongly between sets at the same velocity (for example, 70 m/s: 98,000 to 3.85 million bytes), so the 3-set average does not increase smoothly with velocity.

OLSR Control Overhead

- Stable Performance Across Velocities: OLSR control overhead stays between about 0.33 and 0.55 million bytes across all velocity levels. This is due to its proactive routing approach, where periodic updates maintain routing tables regardless of node velocity.

- Minimal Impact of Velocity: At the highest velocity (100 m/s), OLSR's average overhead (about 0.37 million bytes) is within the range seen at lower velocities.

Insights

The results highlight the following key observations:

- Performance at low velocity: AODV's average overhead is lower than OLSR's at 0, 10 and 20 m/s, but not at 30 m/s.
- Performance at high velocity: OLSR's proactive updates keep the overhead within a narrow range, while AODV's average overhead is higher than OLSR's at 50, 70, 80, 90 and 100 m/s and lower at 40 and 60 m/s.

4.7 Multiple MANETS

Open NetSim and Select Examples→Mobile Ad hoc Networks→Multiple MANETs then click on the tile in the middle panel to load the example as shown in below screenshot Figure 4-29.

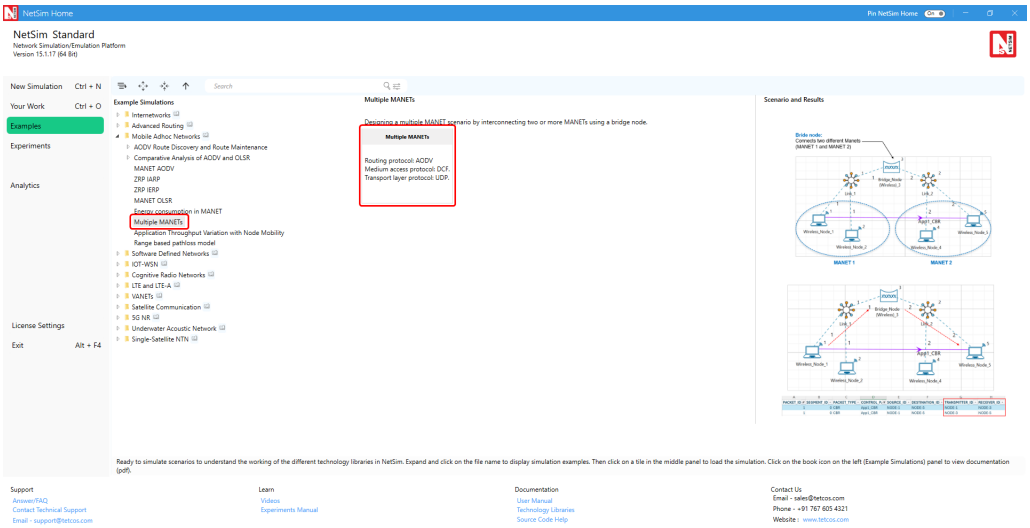


Figure 4-49: *List of scenarios for the example of Multiple-MANETs*

Multiple MANETs allows users to interconnect two or more MANETs using a bridge node. Click and drop wireless nodes, Ad hoc links and bridge node onto the grid environment and connect wireless nodes to ad hoc links to form two different MANETs using ad hoc links. Further, connect the two MANETs using a bridge node as shown below Figure 4-30.

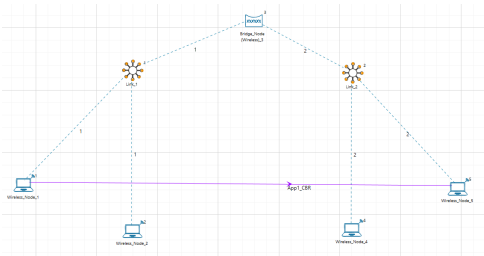


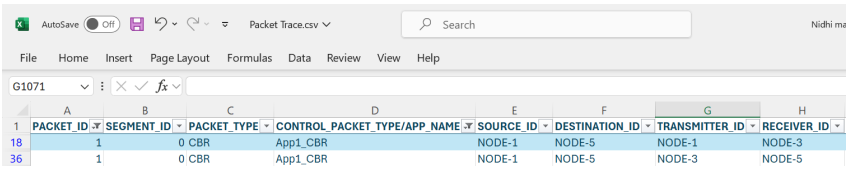
Figure 4-50: *Network set up for studying the Multiple-MANETs*

4.7.1 Network Settings

1. Grid length: 500m×250m from grid setting property panel on the right. This needs to be done before any device is placed on the grid.
2. A network scenario is designed in the NetSim GUI, comprising four wireless nodes and one bridge node, with two ad hoc links from the Interconnected MANETs module.
3. To configure any properties in the nodes, click on the node, expand the property panel on the right side, and change the properties as mentioned below.
4. The Medium access protocol was set to DCF in Interface (Wireless)→ Datalink layer of all wireless nodes and bridge node (Interface 1 and Interface 2).
5. The routing protocol in the network layer is set to AODV for all nodes. As this is a global property, changing it on one node will affect all nodes.
6. Create a CBR application from wireless node 1 to wireless node 5 from the set traffic tab in the ribbon on top. Click on the created application, and in the right-side property panel, set the transport protocol to UDP, keeping the other application properties as default.
7. Enable packet trace from the configure reports tab in ribbon on top.
8. Run simulation for 10 seconds.

4.7.2 Output

After simulation, open packet trace from simulation results window and filter Control Packet Type/App Name to App1 CBR and Packet Id to 1. In the below screenshot, users can observe that the packets from source node-1 are going to destination node-5 via bridge node-3.



PACKET_ID	SEGMENT_ID	PACKET_TYPE	CONTROL_PACKET_TYPE/APP_NAME	SOURCE_ID	DESTINATION_ID	TRANSMITTER_ID	RECEIVER_ID
1	0	CBR	App1_CBR	NODE-1	NODE-5	NODE-1	NODE-3
1	0	CBR	App1_CBR	NODE-1	NODE-5	NODE-3	NODE-5

Figure 4-51: *Packet transmitting from Source Node-1 to Destination Node-5 via Bridge Node-3 in Packet Trace.*

4.8 Energy consumption in MANETs

In this experiment, we aim to analyze the per-packet energy consumption for various types of packets, including data packets of different sizes, WLAN ACKs, and MANET control packets, across both DSR and AODV protocols.

Open NetSim and Select Examples→ Mobile Ad hoc Networks →Energy consumption in MANET then click on the tile in the middle panel to load the example as shown in below screenshot.

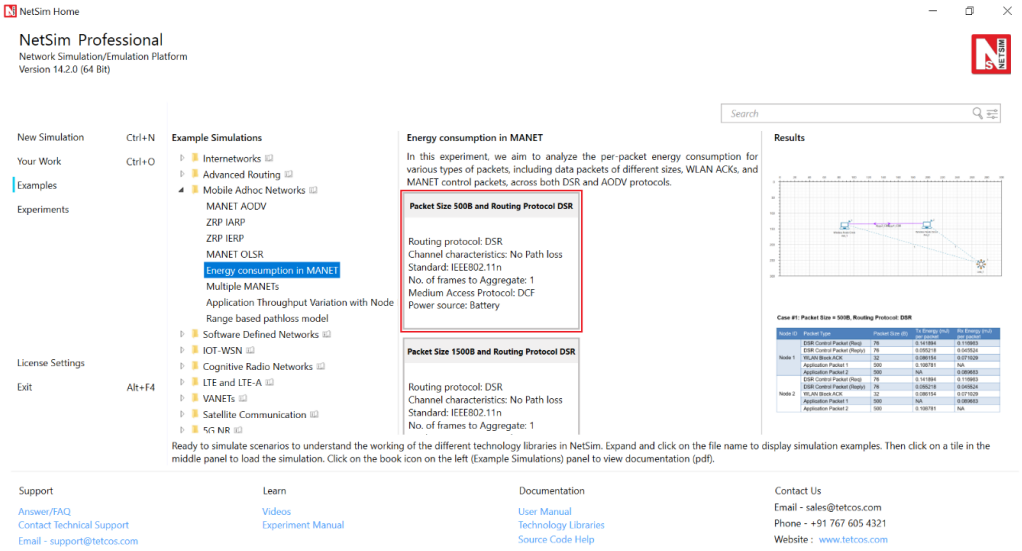


Figure 4-52: *List of scenarios for the example of Energy consumption in MANET.*

4.8.1 Network Setup

The following network diagram illustrates what the NetSim UI displays when you open the example configuration file.

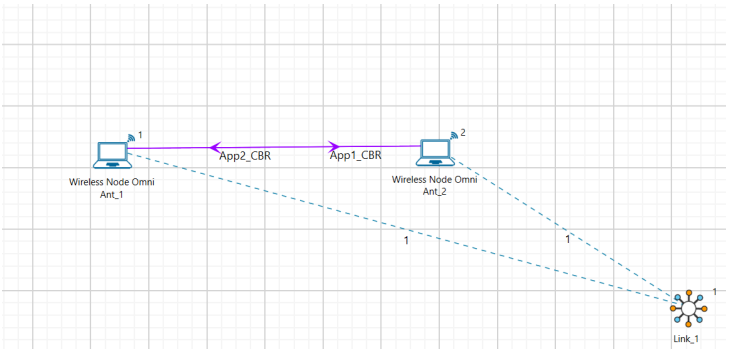


Figure 4-53: *Network setup for analyzing the energy consumption in MANETs.*

- The scenario comprises of two wireless node omni ants, wireless node omni ant1 and wireless node omni ant2.
- Wireless node omni ant1 sends data to wireless node omni ant2, and wireless node omni ant2 sends data to wireless node omni ant1.
- Set grid length to 400m×200m from grid setting property panel on the right. This needs to be done before any device is placed on the grid.
- Click on wireless node, expand the property panel on the right and set the following properties under different layers.

Table 4-35: *Wireless node Omni Ant properties configured in NetSim.*

Device Properties	Value
Position layer properties	
Mobility Model	No Mobility
Network layer properties	
Routing Protocol	DSR/AODV
Interface(Wireless) properties	
Datalink layer properties	
Medium Access Protocol	DCF
Physical layer properties	
Standard	IEEE802.11n
Number of Frames to Aggregate	1
Tx & Rx antenna counts	1 × 1
Power Source	Battery

Table 4-36: *Battery model properties configured in NetSim.*

Battery Properties	Value
Energy harvesting	Off
Initial energy (mAh)	300
Transmitting current	316.7
Idle mode current	227
Voltage (V)	3.6
Receiving current	261.1

Table 4-37: *Link properties configuration.*

Link Properties	Value
Channel characteristics	No pathloss

Table 4-38: *Application configured from wireless node omni Ant1 to wireless node omni ant2 and wireless node omni ant2 to wireless node omni ant1 for 2 cases, with packet size as 420B and 1420B. These are the application layer packet sizes and on adding overheads the packet sizes are 500B and 1500B respectively at the PHY layer.*

Application Properties	Value
Source Id	1 / 2
Destination Id	2 / 1
Packet size (B)	420 / 1420 / 428
Inter arrival time	20000
Mean generation rate (kbps)	168 / 568 / 171

- Click on the set traffic tab in the top ribbon/toolbar and create a CBR application as per the below settings.
- Enable the Packet trace under config reports tab on the top.

- Enable the IEEE 802.11 Radio measurement and Energy log under Network log as shown below.

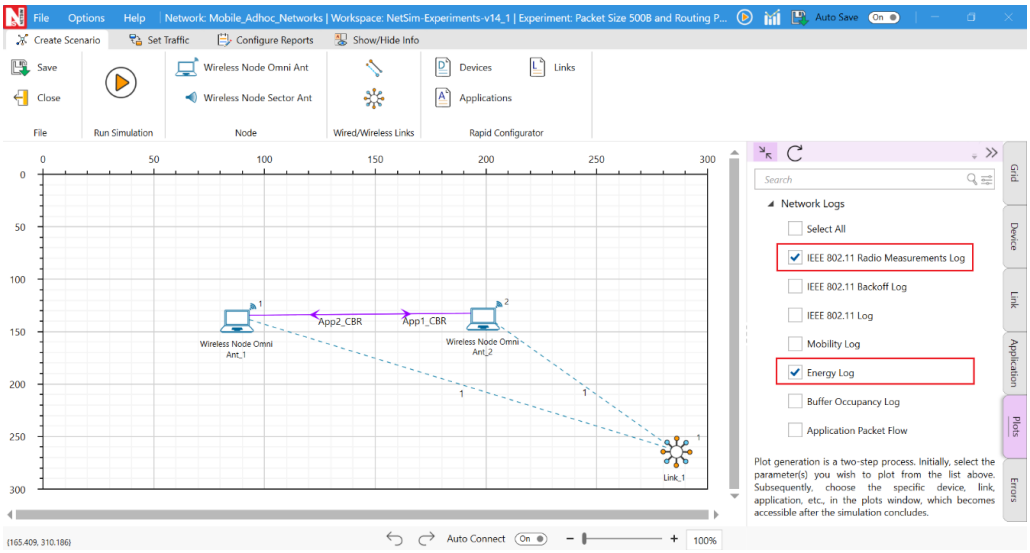


Figure 4-54: *Enabling IEEE 802.11 Radio measurement and Energy log.*

- Run the simulation for 100 seconds.

4.8.2 Energy Consumption

The energy consumption is given by

$$\text{ConsumedEnergy } (J) = V \text{ [Volts]} \times I \text{ [Amps]} \times t \text{ [sec]}$$

where V is the battery voltage, I is the current drawn and t is the time for which the node is in a particular mode.

This can be computed for each packet and for each mode using NetSim's energy log. The formula to be used is

$$\text{ConsumedEnergy (mJ)} = \frac{\text{Voltage } [V] \times \text{ModeCurrent [mA]} \times (\text{CurrentTime } (\mu s) - \text{ModeChangeTime } (\mu s))}{10^6}$$

where current time minus mode-change-time is the time for which the node is in a particular mode.

4.8.3 Energy consumption calculations

In the results window, open packet trace and Energy log file as shown below.

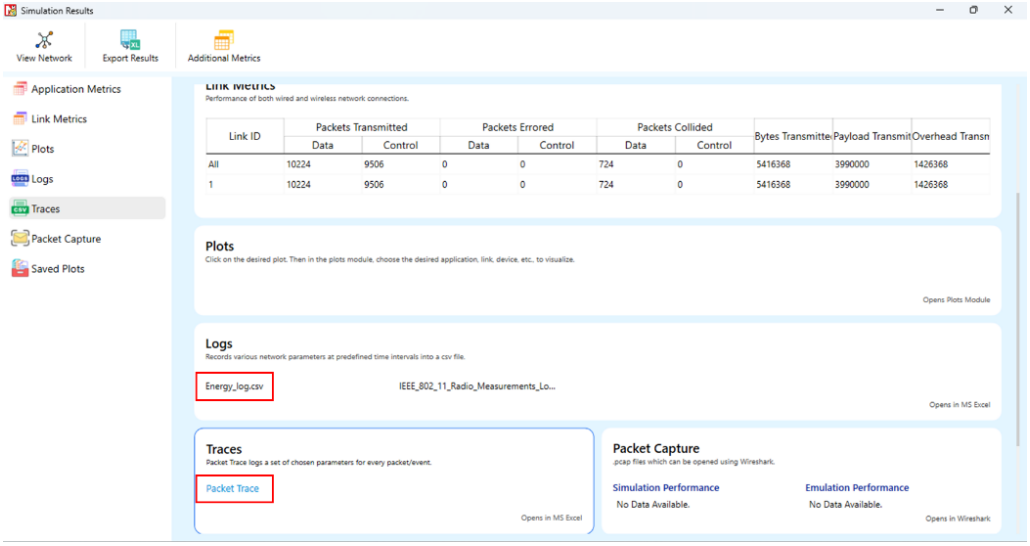


Figure 4-55: *Open Energy log file and packet trace from Results tab.*

Open Energy log file and Filter out previous mode to TRX on busy and RX on busy in Energy log as shown below.

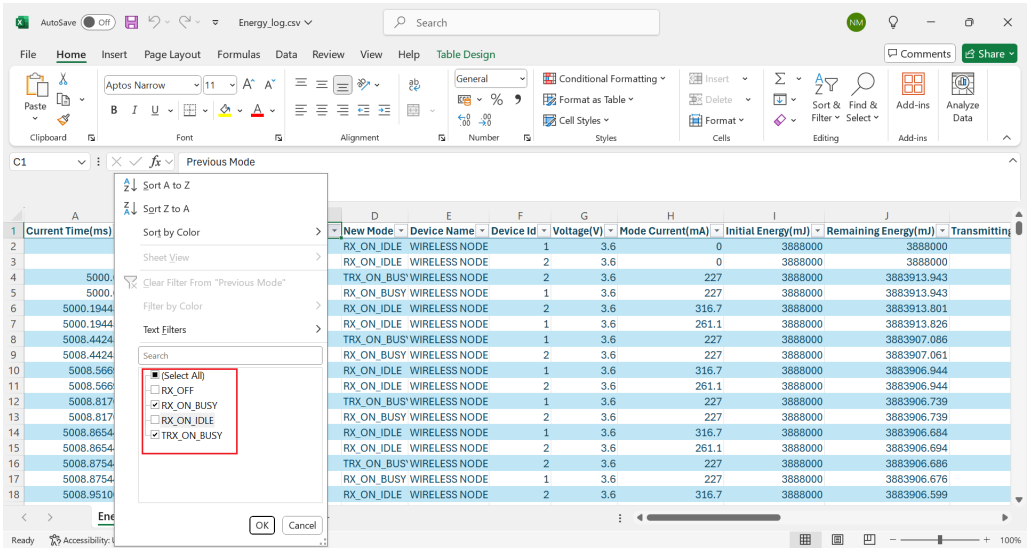


Figure 4-56: *Filter out RX on busy and TRX on busy under previous mode in Energy log file.*

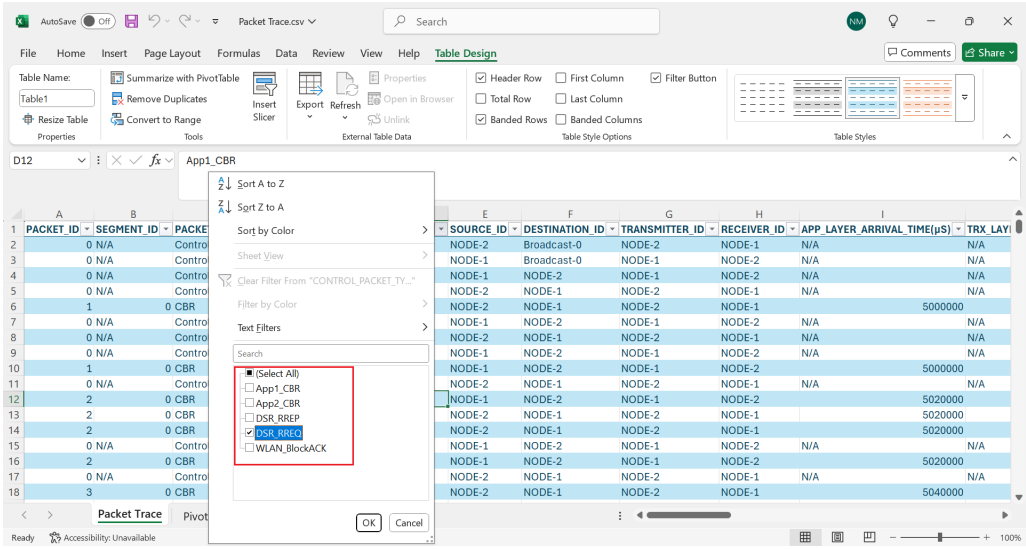


Figure 4-57: *Filter out the DSR RREQ under Control Packet Type/App Name in packet trace.*

To calculate the energy consumption for different packet types, first filter out the packet type in packet trace and note the PHY LAYER ARRIVAL TIME (μs) column value and cross-reference with Mode Switch Time(ms) column in the energy log file for TRX ON BUSY/RX ON BUSY case as shown below.

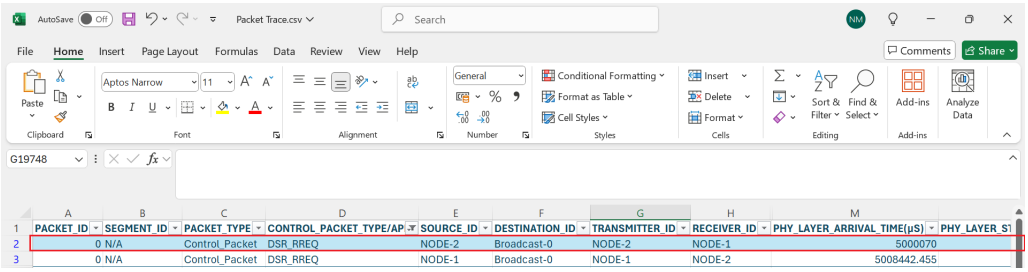


Figure 4-58: *Filter out packet type in packet trace.*

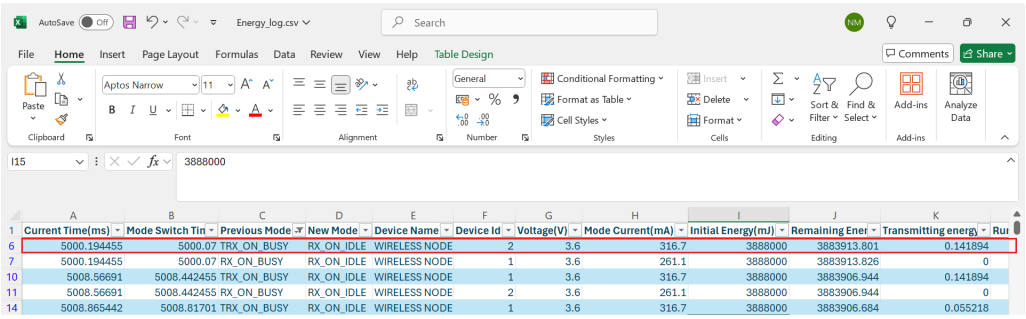


Figure 4-59: *Filter out mode to TRX ON BUSY and RX ON BUSY in Energy log. Record the Phy Layer Arrival Time from packet trace and cross-reference with Mode Switch Time in Energy log.*

In the calculations below, the voltage and mode current are user inputs while current time and mode switch time are recorded in the energy log file. The packet trace is only used to determine the type of packet (Route request, Data, WLAN ACK etc.).

- DSR Route Request
- Transmit Energy

$$\text{Energy (mJ)} = \frac{\text{Voltage}(V) \times \text{ModeCurrent(mA)} \times (\text{CurrentTime} - \text{ModeChangeTime})(\text{ms})}{10^3} \tag{11}$$

$= 3.6V \times 316.7\text{mA} \times (5008.5669 - 5008.4424) \times 10^{-3}$
$= 0.141894$

- Receive Energy

$$\text{Energy (mJ)} = \frac{\text{Voltage}(V) \times \text{ModeCurrent(mA)} \times (\text{CurrentTime} - \text{ModeChangeTime})(\text{ms})}{10^3} \tag{12}$$

$= 3.6V \times 261.1 \text{ mA} \times (5000.19446 - 5000.07) \times 10^{-3}$
$= 0.116983$

- DSR Route Reply
- Transmit Energy

$$\text{Energy (mJ)} = \frac{\text{Voltage}(V) \times \text{ModeCurrent(mA)} \times (\text{CurrentTime} - \text{ModeChangeTime})(\text{ms})}{10^3} \tag{13}$$

$= 3.6V \times 316.7\text{mA} \times (5008.865442 - 5008.81701) \times 10^{-3}$
$= 0.055218$

- Receive Energy

$$\text{Energy (mJ)} = \frac{\text{Voltage}(V) \times \text{ModeCurrent(mA)} \times (\text{CurrentTime} - \text{ModeChangeTime})(\text{ms})}{10^3} \tag{14}$$

$= 3.6V \times 261.1 \text{ mA} \times (5010.473342 - 5010.42491) \times 10^{-3}$
$= 0.045524$

- Application (Data) Packets, 500B
- Transmit Energy

$$\text{Energy (mJ)} = \frac{\text{Voltage}(V) \times \text{ModeCurrent(mA)} \times (\text{CurrentTime} - \text{ModeChangeTime})(\text{ms})}{10^3} \tag{15}$$

$= 3.6V \times 316.7\text{mA} \times (5009.21652 - 5009.121108) \times 10^{-3}$
$= 0.1087811$

- Receive Energy

$$\text{Energy (mJ)} = \frac{\text{Voltage}(V) \times \text{ModeCurrent(mA)} \times (\text{CurrentTime} - \text{ModeChangeTime})(\text{ms})}{10^3} \quad (16)$$

$= 3.6V \times 261.1 \text{ mA} \times (5010.76442 - 5010.669008) \times 10^{-3}$

$= 0.089683$

- WLAN Block ACK Packets
- Transmit Energy

$$\text{Energy (mJ)} = \frac{\text{Voltage}(V) \times \text{ModeCurrent(mA)} \times (\text{CurrentTime} - \text{ModeChangeTime})(\text{ms})}{10^3} \quad (17)$$

$= 3.6V \times 316.7\text{mA} \times (5009.302086 - 5009.22652) \times 10^{-3}$

$= 0.086154$

- Receive Energy

$$\text{Energy (mJ)} = \frac{\text{Voltage}(V) \times \text{ModeCurrent(mA)} \times (\text{CurrentTime} - \text{ModeChangeTime})(\text{ms})}{10^3} \quad (18)$$

$= 3.6V \times 261.1 \text{ mA} \times (5008.951008 - 5008.875442) \times 10^{-3}$

$= 0.071029$

4.8.4 Case#1: Results and discussion

Packet Size = 500B, Routing Protocol: DSR

Table 4-39: *Tx and Rx energy consumption per packet for 500-byte packet size and DSR routing protocol.*

Node ID	Packet Type	Size (B)	Tx Energy (mJ)	Rx Energy (mJ)
Node 1	DSR Ctrl Pkt (Req)	76	0.1418936	0.1169827
	DSR Ctrl Pkt (Reply)	76	0.0552183	0.0455241
	WLAN Block ACK	32	0.0861543	0.0710290
	App Packet 1	500	0.1087811	NA
	App Packet 2	500	NA	0.0896835
Node 2	DSR Ctrl Pkt (Req)	76	0.1418936	0.1169827
	DSR Ctrl Pkt (Reply)	76	0.0552183	0.0455241
	WLAN Block ACK	32	0.0861543	0.0710290
	App Packet 1	500	NA	0.0896835
	App Packet 2	500	0.1087811	NA

4.8.5 Case #2: Results and discussion

Packet Size = 1500B, Routing Protocol: DSR

Table 4-40: *Tx and Rx energy consumption per packet for 1500-byte packet size and DSR routing protocol.*

Node ID	Packet Type	Size (B)	Tx Energy (mJ)	Rx Energy (mJ)
Node 1	DSR Ctrl Pkt (Req)	76	0.1418936	0.1169827
Node 1	DSR Ctrl Pkt (Reply)	76	0.0552183	0.0455241
Node 1	WLAN Block ACK	32	0.0861543	0.0710290
Node 1	App Packet 1	1500	0.2351098	NA
Node 1	App Packet 2	1500	NA	0.1938339
Node 2	DSR Ctrl Pkt (Req)	76	0.1418936	0.1169827
Node 2	DSR Ctrl Pkt (Reply)	76	0.0552183	0.0455241
Node 2	WLAN Block ACK	32	0.0861543	0.0710290
Node 2	App Packet 1	1500	NA	0.1938339
Node 2	App Packet 2	1500	0.2351098	NA

- DSR Route Request is a broadcast packet.
- Broadcast packet is sent at "broadcast" rate which uses the lowest rate (MCS).
- Since this packet is sent at the lowest rate, the Transmission time is higher leading to higher energy consumption.
- DSR Route reply is a unicast packet.
- This is sent at "unicast" rate. This rate (MCS) is chosen per the Wi-Fi MCS-Rx Power tables. In this case it is MCS 7.
- Transmission time is lower than DSR Route Request leading to lower energy consumption, even though the RREQ and RREP packets are of the same size.
- We see that the application and WLAN ACK packets have the same transmit and received energies for both nodes.
- Compared to case #1, in case #2, the packet size is three times larger (1500B vs. 500B), but the energy consumption isn't proportionally higher (i.e., it is not three times more). This is because the difference in the total transmission times is not a factor of three.
- Recall that in Wi-Fi

$$\text{TotalTransmissionTime } (s) = \text{Pkt. TransmissionTime} + \text{PreambleTime}(s) \tag{19}$$

We know that

$$\text{Pkt. TransmissionTime } (\mu s) = \text{ceil}\left(\frac{\text{Pkt.Size}(B)}{\text{DataRate(Mbps)}}\right) \tag{20}$$

- While the packet transmission time is three times longer than before, the preamble time remains constant at 40 μs in both cases.
- For a packet size of $500B$ given a data rate of 72.2 Mbps

$$\text{Pkt. TransmissionTime } (\mu s) = \text{ceil}\left(\frac{500 \times 8}{72.2}\right) = 55.402 \tag{21}$$

$$\text{TotalTransmissionTime } (\mu s) = 55.402 + 40 = 95.402 \tag{22}$$

- For packet size = 1500B

$$\text{Pkt. TransmissionTime } (\mu s) = \text{ceil}\left(\frac{1500 \times 8}{72.2}\right) = 166.205 \tag{23}$$

$$\text{TotalTransmissionTime } (\mu s) = 166.205 + 40 = 206.205 \tag{24}$$

- We can see that the total transmission time is only about 2.16 times higher. Therefore, the total energy consumption is also 2.16 times higher than the previous scenario, and not three times higher.

4.8.6 Case #3: Results and discussion

Packet Size = 500B, Routing Protocol: AODV

We perform similar energy calculations for AODV and obtain the table below.

Table 4-41: *Tx and Rx energy consumption per packet for 500-byte packet size and AODV routing protocol.*

Node ID	Packet Type	Size (B)	Tx Energy (mJ)	Rx Energy (mJ)
Node 1	AODV Ctrl Pkt (Req)	88	0.1570949	0.1295152
Node 1	AODV Ctrl Pkt (Reply)	84	0.0562284	0.0463569
Node 1	WLAN Block ACK	32	0.0861543	0.0710290
Node 1	App Packet 1	500	0.1087811	NA
Node 1	App Packet 2	500	NA	0.0896835
Node 2	AODV Ctrl Pkt (Req)	88	0.1570949	0.1295152
Node 2	AODV Ctrl Pkt (Reply)	84	0.0562284	0.0463569
Node 2	WLAN Block ACK	32	0.0861543	0.0710290
Node 2	App Packet 1	500	NA	0.0896835
Node 2	App Packet 2	500	0.1087811	NA

- AODV Route Request is a broadcast packet.
- Broadcast packet is sent at "broadcast" rate which is the lowest MCS.
- Transmission time is higher leading to higher energy consumption.
- AODV Route reply is a unicast packet.
- This is sent at Unicast rate which is dependent on the received power. In this case it is sent at MCS 7 (QAM 64).
- We see that the application and WLAN ACK packets have the same transmit and receive energies as in DSR.
- Similar calculations as done for DSR can be carried out to verify the results.

Energy consumption plot:

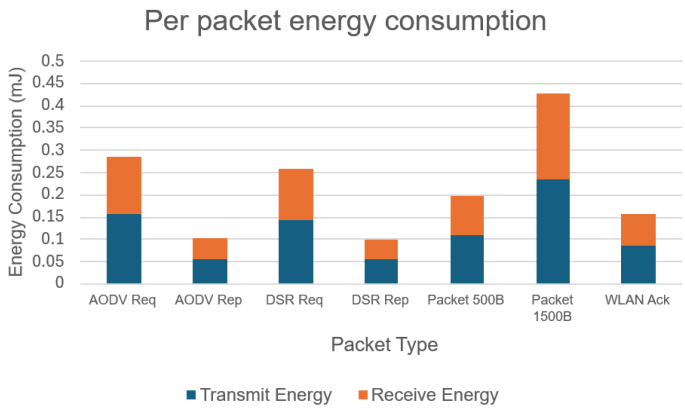


Figure 4-60: *Energy consumption plot for different packet types.*

4.9 Application throughput variation with node mobility

Objective

To simulate and analyze how application throughput varies as the node moves away. This analysis is done for 802.11n standard.

Open NetSim and Select Examples→Mobile Ad hoc Networks→Application Throughput Variation with Node Mobility then click on the tile in the middle panel to load the example as shown in below screenshot Figure 4-36.

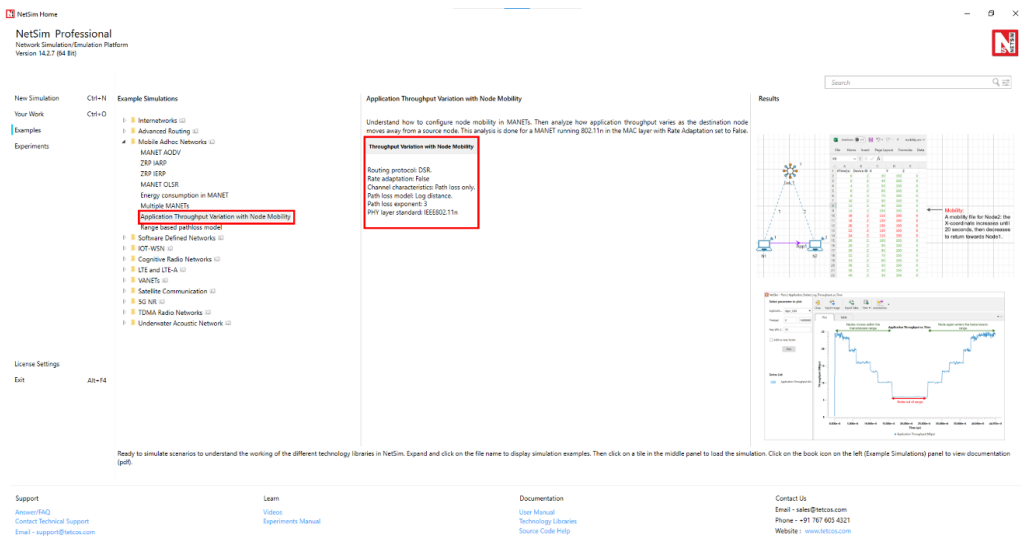


Figure 4-61: *List of scenarios for the example of Application Throughput Variation with Node Mobility*

The following network diagram illustrates what the NetSim UI displays when you open the example configuration file see Figure 4-37.

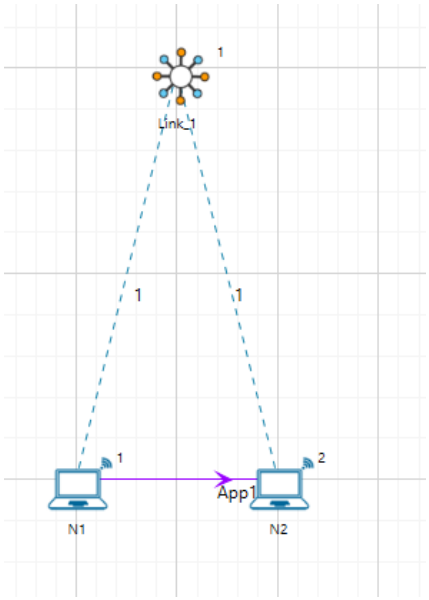


Figure 4-62: *Network set up for studying the Application Throughput variation with Node mobility*

4.9.1 Network Settings

1. The device coordinates were set as follows. To configure the below, click on device, expand the property panel on right and change the below in position layer.

Table 4-42: *Devices Position*

Device	X-Coordinate	Y-Coordinate
N1	10	100
N2	30	100

2. Set grid length to 300m×150m from grid setting property panel on the right. This needs to be done before any device is placed on the grid.
3. To configure any properties in the nodes, click on the node, expand the property panel on the right side, and change the properties as mentioned below.
4. IEEE standard was set to 802.11n in Interface (Wireless) ▷ Physical layer of both the wireless nodes.
5. The Physical layer properties were set as shown in Table 4-30.

Table 4-43: *Detailed Network Parameters*

Interface Parameters	Value
Standard	IEEE802.11n
No. of Frames to Aggregate	1
Transmitter Power	100mW
Bandwidth	20MHz
Standard Channel	1 (2412MHz)
Reference distance (d0)	1m
Guard Interval	400ns
Antenna	
Antenna height	1m
Antenna gain	0
Transmitting antennas	1
Receiving antennas	1

6. The Medium Access Protocol was set to DCF in Interface (Wireless)→ Datalink layer of both wireless nodes.
7. Click on N1 and expand right-hand side property panel and go to Network layer, Enable Static IP Route and Configure Static Route via GUI as shown below screenshot Figure 4-38.

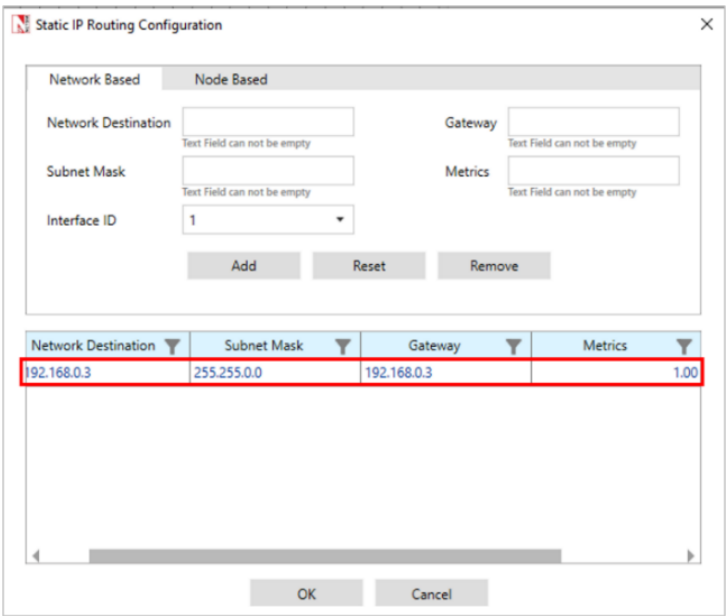


Figure 4-63: *Static IP Route configuration window*

8. Rate Adaptation was set as False in Interface (Wireless) → Datalink layer of both wireless nodes.
9. Click on link and expand right-hand side property panel and set the Channel characteristics as: Path loss only; Path loss model: Log Distance; Path loss exponent: 3 under medium property.
10. Click on the N1 and N2 devices and expand right-hand side property panel and set the mobility model to No Mobility in N1 and File Based Mobility in N2.
11. Click on set traffic tab and configure Unicast CBR application from N1 to N2. Click on the created application, and in the right-side property panel, set the transport protocol to UDP and the start time to 0s. Set the packet size to 1460 bytes and Inter Arrival Time to 476.2 microseconds, which gives a Generation Rate of 25Mbps.

12. Set application start time as 0 sec.
13. The Transport protocol was set to UDP.
14. Enable the Application throughput vs time plot and SNR vs time plot under Plots in Configure reports.
15. Run the simulation for 45 seconds.
16. Open the Application throughput vs. Time plot and uncheck the Accelerate plotting and observe the below plot.

File based mobility file

Table 4-44: *File based mobility file that models the movement of N2. The X co-ordinate is incrementally increased till the 20th second ensuring N2 moves away from node1. Subsequently, the X-coordinate is decreased to facilitate N2's return towards N1.*

Time(s)	Node ID	X (m)	Y (m)	Z (m)
0	2	30	100	0
2	2	40	100	0
4	2	50	100	0
6	2	60	100	0
8	2	70	100	0
10	2	80	100	0
12	2	90	100	0
14	2	100	100	0
16	2	110	100	0
18	2	120	100	0
20	2	130	100	0
22	2	120	100	0
24	2	110	100	0
26	2	100	100	0
28	2	90	100	0
30	2	80	100	0
32	2	70	100	0
34	2	60	100	0
36	2	50	100	0
38	2	40	100	0
40	2	30	100	0

4.9.2 Output

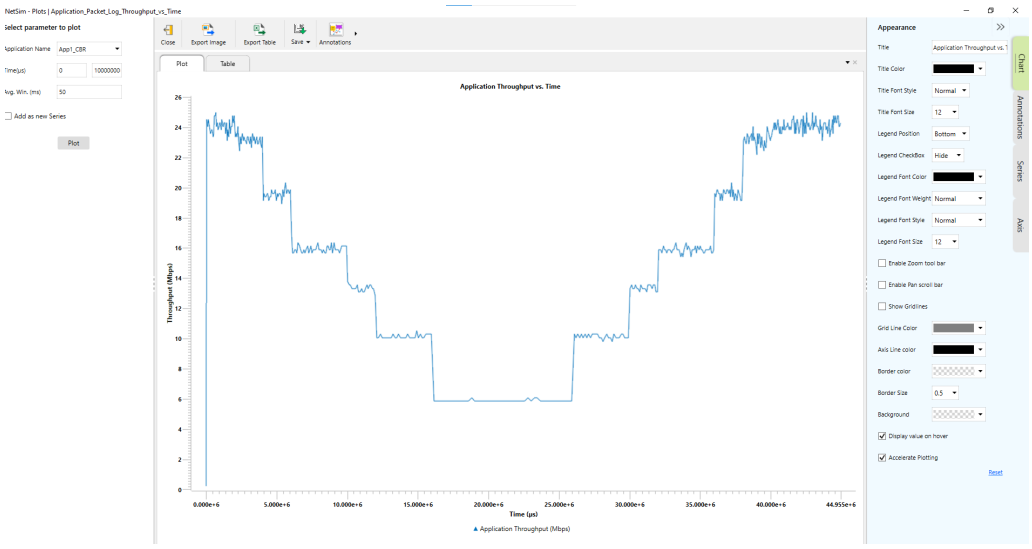


Figure 4-64: *Plot of Application throughput vs. time.*

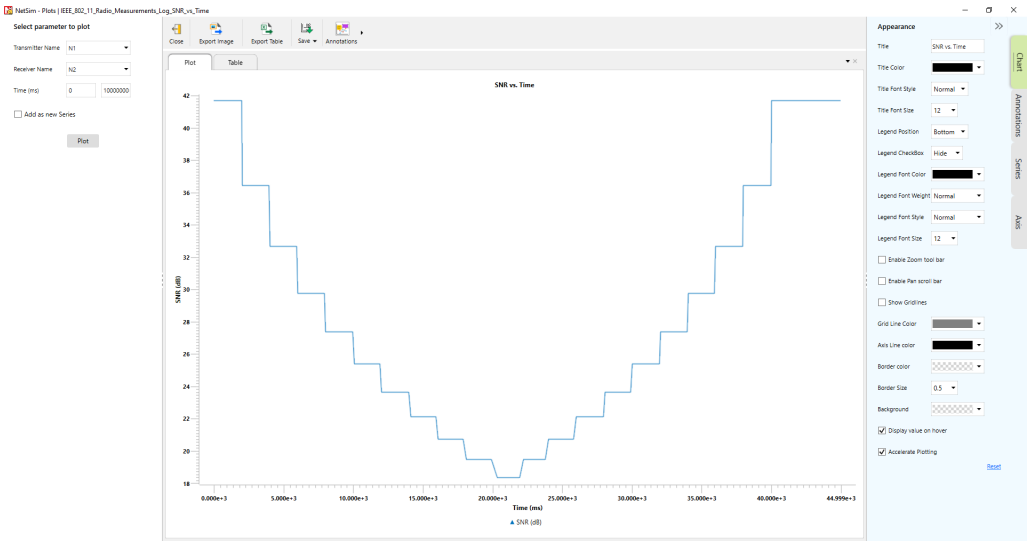


Figure 4-65: *Plot of SNR vs. Time.*

4.9.3 Inference

We can also observe the SNR vs. Time plot obtained after the simulation. The SNR is high initially and gradually drops as N2 moves away from N1. After N2 moves back towards the transmitter around the 20th second, we observe an increase in the SNR. The pathloss exponent in the log distance pathloss model was set to 3.0.

Recall that throughput is proportional to SNR. We observe that the Application throughput drops from 25 Mbps to 20 Mbps at around 4 seconds (the distance at this time is 50 m), and then to 15.8 Mbps at 6.5 seconds (the distance at this time is 70 m) and to 10.2 Mbps at 12 seconds (the distance at this time is 90 m) and then to 5.8 Mbps at 20 seconds (the distance at this time is 130 m). After the 20th second N2 starts moving back towards N1 and we can see the increase in throughput.

4.10 Performance of range based pathloss model in MANETs

Open NetSim and Select Examples→Mobile Ad hoc Networks→Range Based pathloss model then click on the tile in the middle panel to load the example as shown in below screenshot Figure 4-40.

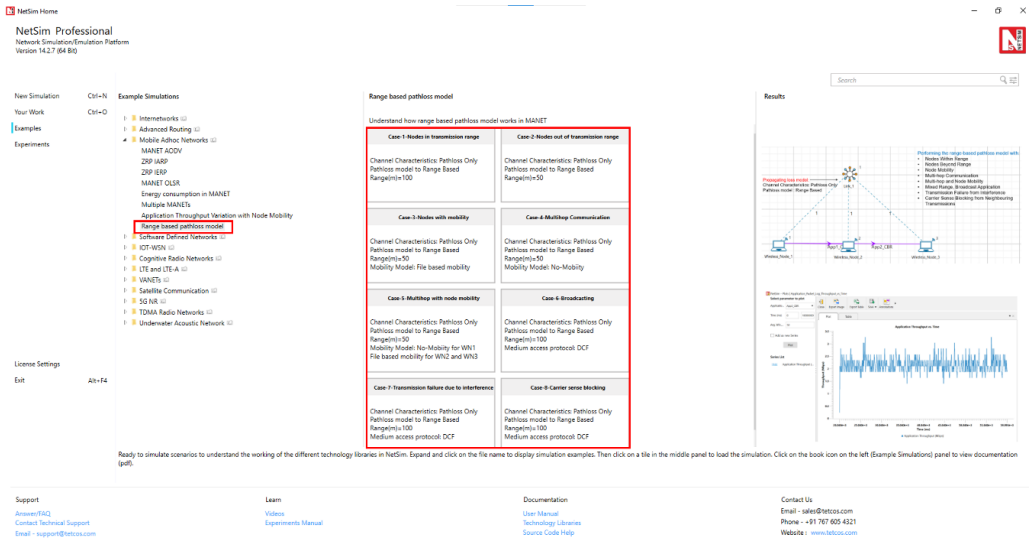


Figure 4-66: *List of scenarios for the example of Range based pathloss model.*

4.10.1 Introduction

The propagation loss depends only on the distance (range) between transmitter and receiver. There is a single Range attribute that determines the path loss.

This is not a real-world loss model but a theoretical model useful for experimentation. Receivers at or within Range meters see a 0 dB pathloss. Hence received power equals transmit power. Receivers beyond Range see a 1000 dB pathloss. Hence received power will be close to −1000 dBm i.e., zero in linear units.

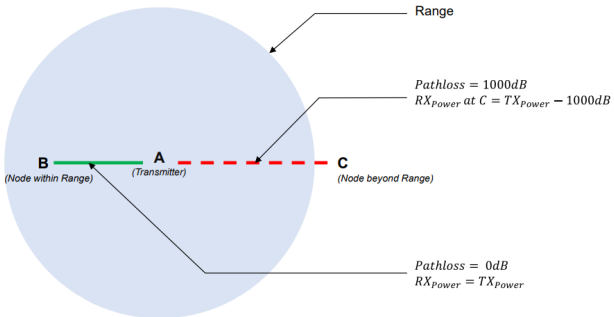


Figure 4-67: *This figure explains a typical range based pathloss model. In this example, Node A and B are in transmission range with each other which is denoted by a blue circle where the pathloss is 0dB i.e, successful transmission. And Node C is beyond the range i.e, pathloss is 1000dB all packets are errored and dropped. In NetSim, users must specify the range of the node in meters in the links.*

The Range-Based Path Loss Model can be used in networks such as Internetworks, MANET, VANET, TDMA, Pure ALOHA, Slotted ALOHA, Cognitive Radio, IoT, WSN, and UWAN.

In this example, we will study the different cases in Range based Pathloss using Mobile Ad hoc networks.

4.10.2 Case 1: Range Based Pathloss. Nodes Within Transmission Range.

The following network diagram illustrates what the NetSim UI displays when you open the example configuration file.

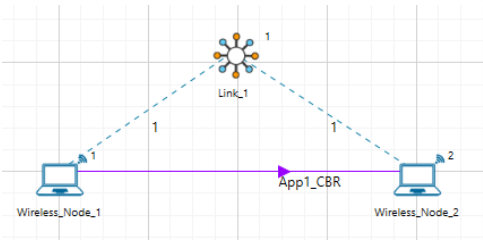


Figure 4-68: *For studying the range based pathloss model with range*

Network Settings

1. Set grid length as 400 × 200 from grid setting property panel on the right. This needs to be done before any device is placed on the grid.
2. Drop 2 wireless nodes and 1 Ad hoc link in Grid environment.
3. Device placements of wireless nodes are:

Table 4-45: *Device Placements*

Device	X-Coordinate	Y-Coordinate
Wireless Node-1	100	150
Wireless Node-2	180	150

4. Click on link and expand right-hand side property panel and set Channel Characteristics: Pathloss, Pathloss model to Range based, Range(m)=100 under medium property.
5. Clicking on Wireless node and expand right-hand side property panel and set, Mobility Model: No-Mobility, for both nodes under position property.
6. Click on the set traffic tab present in the top ribbon. Create a CBR Application with 11Mbps generation rate (Set Packet size: 1460, Inter Arrival Time: 1061.81 μs) from wireless node-1 to wireless node-2. Set Transport protocol to UDP and start time to zero.
7. Enable throughput vs time plot under Application and link performance by clicking on plots/logs tab in right panel.
8. Run the simulation for 100 seconds.

4.10.3 Output and Plot

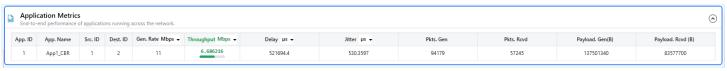


Figure 4-69: *Application Metrics window.*

In Simulation result window, click on Throughput vs Time plot under Application performance. After the plot window is displayed, click on the plot option.

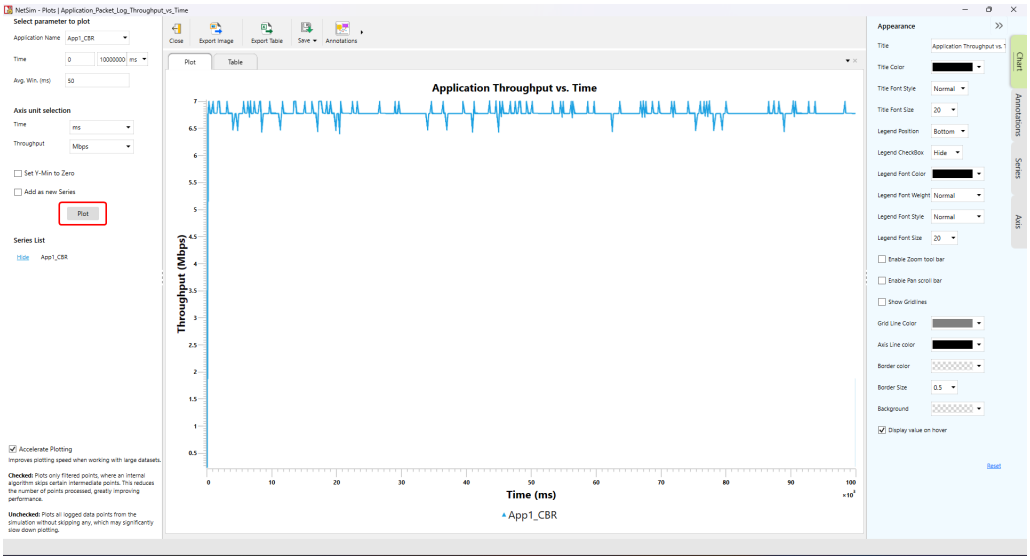


Figure 4-70: *Application throughput plot.*

Discussion

The above results show continuous data transmission and constant throughput throughout the simulation. This occurs because Wireless Node-2 is within the transmission range of Wireless Node-1. Specifically, the range-based parameter is set to 100 meters, and Wireless Node-2 is positioned 80 meters away from Wireless Node-1.

4.10.4 Case 2: Range Based Pathloss. Nodes Beyond Transmission Range.

The following network diagram illustrates what the NetSim UI displays when you open the example configuration file.

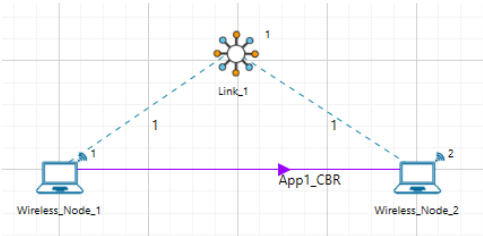


Figure 4-71: *For studying the range based pathloss model without range*

Network Settings

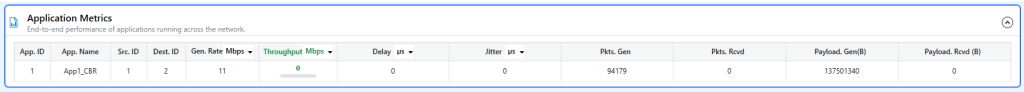
1. Set Grid length as 400 × 200 from grid setting property panel on the right. This needs to be done before any device is placed on the grid.
2. Drop 2 wireless nodes and 1 Ad hoc link in Grid environment.
3. Device placements of wireless nodes are:

Table 4-46: *Device placements*

Device	X-Coordinate	Y-Coordinate
Wireless Node-1	100	150
Wireless Node-2	180	150

4. Click on link and expand right-hand side property panel and set channel characteristics: Pathloss, Pathloss model to Range based, Range(m)=50 meters under medium property.
5. Click on wireless node and expand right-hand side property panel and set Mobility model: No-mobility, for both nodes under position property.
6. Click on the set traffic tab present in the top ribbon, create a CBR Application with 11Mbps generation rate (Set Packet size: 1460, Inter Arrival Time: 1061.81 μs) from wireless node-1 to Wireless node-2. Set Transport protocol to UDP and start time to zero.
7. In NetSim GUI Application and link throughput vs time plot should be enabled.
8. Run the Simulation for 100 seconds.

4.10.5 Output and Plot



App. ID	App. Name	Src. ID	Dest. ID	Gen. Rate Mbps	Throughput Mbps	Delay µs	Jitter µs	Pkts. Gen	Pkts. Rcvd	Payload. Gen(B)	Payload. Rcvd (B)
1	App1_CBR	1	2	11	0	0	0	94179	0	137501340	0

Figure 4-72: *Application Metrics window*

Discussion

From the results, it is evident that there is no data transmission since wireless node-2 is outside the range of wireless node-1. The range-based parameter is set to 50 meters, whereas wireless node-2 is placed 80 meters away from wireless node-1.

4.10.6 Case 3: Range Based Pathloss with Node Mobility.

The following network diagram illustrates what the NetSim UI displays when you open the example configuration file.

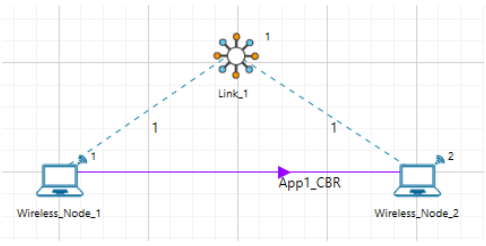


Figure 4-73: *For studying the range based pathloss with ping pong in nature*

Network Settings

1. Set grid length as 400 × 200 from grid setting property panel on the right. This needs to be done before any device is placed on the grid.
2. Drop 2 wireless nodes and 1 Ad hoc link in Grid environment.
3. Device placements of wireless nodes are:

Table 4-47: *Device Placements*

Device	X-Coordinate	Y-Coordinate
Wireless Node-1	100	150
Wireless Node-2	180	150

4. Click on link and expand right-hand side property panel and set Channel Characteristics: Pathloss, Pathloss model to Range based, Range(m)=100 under medium property.
5. Click on Wireless node and expand right-hand side property panel and set Mobility model: No-mobility to Wireless node 1 and Mobility model: File based mobility to wireless node 2 under position property.
6. The file-based mobility pattern specifies that at the 0th second, wireless node 2 will be within range, and at the 5th second, it will move out of range. This cycle of moving in and out of range every 5 seconds repeats continuously for 50-seconds.

#Time(s)	Device ID	X	Y	Z
0	2	190	150	0
5	2	210	150	0
10	2	190	150	0
15	2	210	150	0
20	2	190	150	0
25	2	210	150	0
30	2	190	150	0
35	2	210	150	0
40	2	190	150	0
45	2	210	150	0
50	2	190	150	0

Figure 4-74: *Mobility.csv file*

7. Click on the set traffic tab present in the top ribbon, create a CBR Application with 11Mbps generation rate (Set packet size: 1460, Inter Arrival Time: 1061.81 μs) from wireless node-1 to wireless node-2. Set Transport protocol to UDP and set start time to zero.
8. Clicking on wireless node 1 will open a right-hand side property panel and go to Network layer, enable Static IP Route and configure static route via GUI from Wireless node 1 to Wireless node 2 as shown in below table.

Table 4-48: *Static route configuration for wireless nodes.*

Devices	Network Dest.	Gateway	Subnet Mask	Metrics	Intf. ID
Wireless node 1	192.169.0.3	192.169.0.3	255.255.255.255	1	1

9. In NetSim GUI Application and link throughput vs time plot should be enabled.
10. Run the Simulation for 55 seconds.

4.10.7 Output and Plot

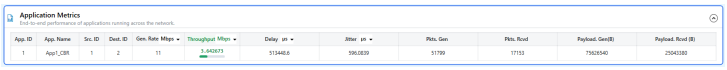


Figure 4-75: *Application Metrics window.*

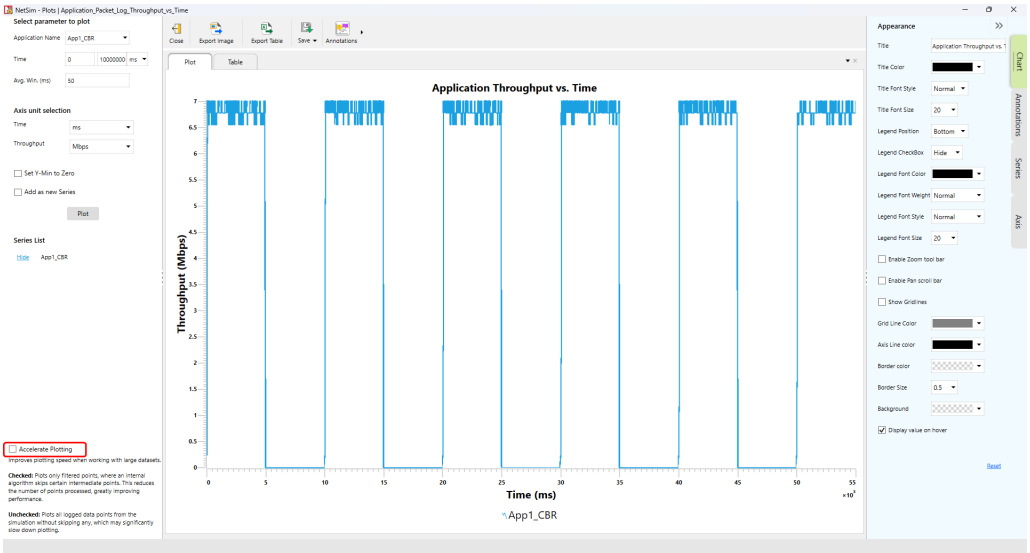


Figure 4-76: *Application throughput vs. Time.*

Discussion

The observed results from the plot indicate that data transmission commences at the 0th second and continues until the 5th second, followed by a pause in transmission until the 10th second. Transmission then resumes from the 15th second and ends at the 20th second. This pattern repeats until the 50th second. This behavior aligns with the mobility pattern defined for wireless node-2, in which it is within a 100-meter range of wireless node-1 for the first 5 seconds, and then moves out of range for the next 5 seconds, and so forth.

4.10.8 Case 4: Range Based Pathloss with Multi-hop Communication.

The following network diagram illustrates what the NetSim UI displays when you open the example configuration file.

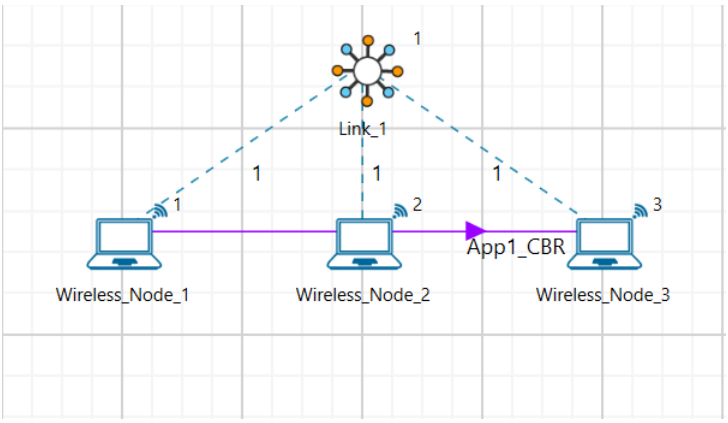


Figure 4-77: *Scenario for studying the range based pathloss model with multihop communication*

Network Settings

1. Set grid length as 400 × 200 from grid setting property panel on the right. This needs to be done before any device is placed on the grid.
2. Drop 3 wireless nodes and 1 Ad hoc link in grid environment.
3. Device placements of wireless nodes are:

Table 4-49: *Device Placements*

Device	X-Coordinate	Y-Coordinate
Wireless Node-1	100	150
Wireless Node-2	140	150
Wireless Node-3	180	150

4. Click on link and expand right-hand side property panel and set Channel characteristics: pathloss, pathloss model to Range based, Range(m)=50 under medium property.
5. Click on wireless node and expand right-hand side property panel and set Mobility model: No-mobility to 3 devices under position property.
6. Configure an application between any two nodes by selecting a CBR application with 11Mbps generation rate (Set packet size: 1460, Inter arrival time: 1061.81 μs) from wireless node-1 to wireless node-3 from the set traffic tab. Set transport protocol to UDP and set start time to zero.
7. Clicking on wireless node 1 and wireless node 2 will open a right-hand side property panel and go to Network layer, enable Static IP Route and configure static route via GUI from wireless node 1 to wireless node 2 and from wireless node 2 to wireless node 3 as shown in below table.

Table 4-50: *Static route configuration for Wireless Node 1 and 2*

Devices	Network Dest.	Gateway	Subnet Mask	Metrics	Intf. ID
Wireless node 1	192.169.0.4	192.169.0.3	255.255.255.255	1	1
Wireless node 2	192.169.0.4	192.169.0.4	255.255.255.255	1	1

8. Enable throughput vs time plot under Application and link performance by clicking on plots/logs tab in right panel.
9. Run the simulation for 100 seconds.

4.10.9 Output and Plot

Application Metrics
End-to-end performance of applications running across the network.

App. ID	App. Name	Src. ID	Dest. ID	Gen. Rate Mbps	Throughput Mbps	Delay µs	Jitter µs	Pkts. Gen	Pkts. Rcvd	Payload. Gen(B)	Payload. Rcvd (B)
1	App1_CBR	1	3	11	3.392222	593950.2	3691.473	94179	29043	137501340	42402780

Figure 4-78: *Application Metrics window.*

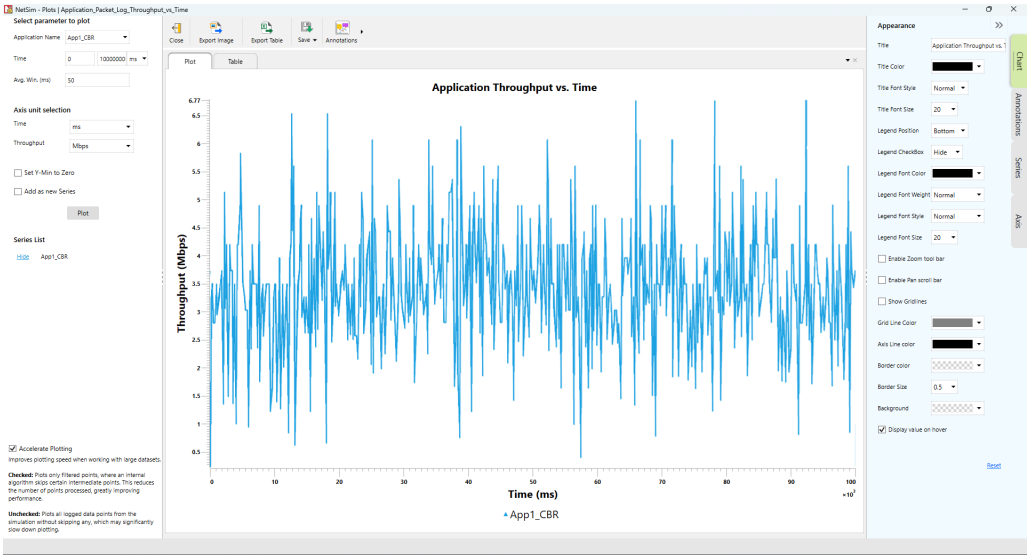


Figure 4-79: *NetSim Plot window.*

In the packet trace, filter packets of type Control Packet Type/APP Name to APP1 CBR. Now you can observe multi-hop communication between node 1 and wireless node 3 via node 2. The configuration involves an application from wireless node 1 to wireless node 3, with a range-based parameter set at 50 meters. The direct distance between wireless node 1 and wireless Node 3 is 80 meters, which exceeds the specified range. However, wireless node 2 is within range of both wireless node 1 and wireless node 3, acting as an intermediate node. Therefore, wireless node 1 utilizes wireless node 2 for data transmission, enabling effective multi-hop communication despite the distance constraints.

A	B	C	D	E	F	G	H	I	J
PACKET_ID	SEGMENT_ID	PACKET_TYPE	CONTROL_PACKET_TYPE/APP_NAME	SOURCE_ID	DESTINATION_ID	TRANSMITTER_ID	RECEIVER_ID	APP_LAYER_ARRIVAL_TIME(µS)	TRX_LAYER_ARRIVAL_TIME(µS)
1	0	CBR	App1_CBR	NODE-1	NODE-3	NODE-1	NODE-2	0	0
2	0	CBR	App1_CBR	NODE-1	NODE-3	NODE-1	NODE-2	1061.81	1061.81
3	0	CBR	App1_CBR	NODE-1	NODE-3	NODE-1	NODE-2	2123.62	2123.62
1	0	CBR	App1_CBR	NODE-1	NODE-3	NODE-2	NODE-3	0	0
2	0	CBR	App1_CBR	NODE-1	NODE-3	NODE-2	NODE-3	1061.81	1061.81
3	0	CBR	App1_CBR	NODE-1	NODE-3	NODE-2	NODE-3	2123.62	2123.62
4	0	CBR	App1_CBR	NODE-1	NODE-3	NODE-1	NODE-2	3185.43	3185.43
5	0	CBR	App1_CBR	NODE-1	NODE-3	NODE-1	NODE-2	4247.24	4247.24
6	0	CBR	App1_CBR	NODE-1	NODE-3	NODE-1	NODE-2	5309.05	5309.05
7	0	CBR	App1_CBR	NODE-1	NODE-3	NODE-1	NODE-2	6370.86	6370.86
8	0	CBR	App1_CBR	NODE-1	NODE-3	NODE-1	NODE-2	7432.67	7432.67
9	0	CBR	App1_CBR	NODE-1	NODE-3	NODE-1	NODE-2	8494.48	8494.48
4	0	CBR	App1_CBR	NODE-1	NODE-3	NODE-2	NODE-3	3185.43	3185.43
5	0	CBR	App1_CBR	NODE-1	NODE-3	NODE-2	NODE-3	4247.24	4247.24
6	0	CBR	App1_CBR	NODE-1	NODE-3	NODE-2	NODE-3	5309.05	5309.05
10	0	CBR	App1_CBR	NODE-1	NODE-3	NODE-1	NODE-2	9556.29	9556.29
11	0	CBR	App1_CBR	NODE-1	NODE-3	NODE-1	NODE-2	10618.1	10618.1
12	0	CBR	App1_CBR	NODE-1	NODE-3	NODE-1	NODE-2	11679.91	11679.91
7	0	CBR	App1_CBR	NODE-1	NODE-3	NODE-2	NODE-3	6370.86	6370.86
8	0	CBR	App1_CBR	NODE-1	NODE-3	NODE-2	NODE-3	7432.67	7432.67
13	0	CBR	App1_CBR	NODE-1	NODE-3	NODE-1	NODE-2	12741.72	12741.72
9	0	CBR	App1_CBR	NODE-1	NODE-3	NODE-2	NODE-3	8494.48	8494.48
10	0	CBR	App1_CBR	NODE-1	NODE-3	NODE-2	NODE-3	9556.29	9556.29

Figure 4-80: *Data transmission from WN1 to WN2*

A	B	C	D	E	F	G	H
PACKET_ID	SEGMENT_ID	PACKET_TYPE	CONTROL_PACKET_TYPE/APP_NAME	SOURCE_ID	DESTINATION_ID	TRANSMITTER_ID	RECEIVER_ID
1	0	CBR	App1_CBR	NODE-1	NODE-3	NODE-1	NODE-2
1	0	CBR	App1_CBR	NODE-1	NODE-3	NODE-2	NODE-3

Figure 4-81: *Filter the Packet Id to 1*

4.10.10 Case 5: Range Based Pathloss with Multi-hop Communication and Node Mobility

The following network diagram illustrates what the NetSim UI displays when you open the example configuration file.

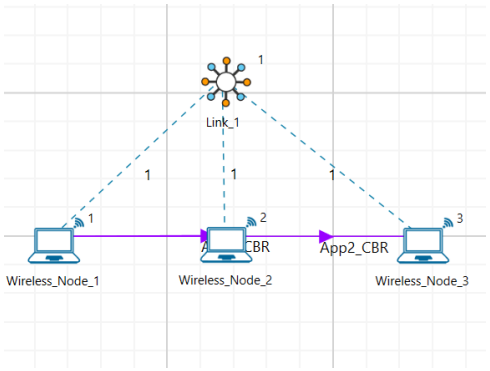


Figure 4-82: *Scenario for studying the range based pathloss model with Multihop and node mobility*

Network Settings

1. Set grid length as 500 × 250 from grid setting property panel on the right. This needs to be done before any device is placed on the grid.
2. Drop 3 wireless nodes and 1 Ad hoc link in Grid environment.
3. Device placements of wireless nodes are:

Table 4-51: *Device Placements*

Device	X-Coordinate	Y-Coordinate
Wireless Node-1	0	200
Wireless Node-2	50	200
Wireless Node-3	100	200

4. Click on link and expand right-hand side property panel and set Channel characteristics: Pathloss, Pathloss model to Range based, Range(m)=50 under medium property.
5. Click on wireless node and expand right-hand side property panel and set mobility model: No-mobility for wireless node 1 and mobility model: File based mobility for wireless node 2 and wireless node 3 under position property.
6. The file-based mobility file specifies the following sequence: At the 0th second, both wireless node 2 (WN2) and wireless Node 3 (WN3) will be out of range. By the 10th second, wireless node 2 will move within range of wireless node 1, while wireless node 3 remains out of range. Starting from the 20th second, wireless node 2 will stay within range, and wireless node 3 will also come into range of wireless node 2. This setup dictates the mobility and connectivity pattern for the nodes throughout the simulation.

#Time(s)	Device ID	X	Y	Z
0	2	51	200	0
0	3	100	200	0
10	2	49	200	0
10	3	100	200	0
20	3	98	200	0

Figure 4-83: *Mobility.csv file*

7. Click on the set traffic tab present in the top ribbon, create CBR application with 11Mbps generation rate (set packet size: 1460, Inter arrival time: 1061.81 μs) from wireless node-1 to wireless node-2 and wireless node-1 to wireless node-3. Set Transport protocol to UDP and set start time to zero.

8. Clicking on wireless node 1 and wireless node 2 will open a right-hand side property panel and go to Network layer, enable static IP route and configure static route via GUI from wireless node 1 to wireless node 2 and from wireless node 2 to wireless node 3 as shown in below table.

Table 4-52: *Static route configuration for wireless nodes 1 and 2*

Devices	Network Dest.	Gateway	Subnet Mask	Metrics	Intf. ID
Wireless node 1	192.169.0.4	192.169.0.3	255.255.255.255	1	1
Wireless node 1	192.169.0.3	192.169.0.3	255.255.255.255	1	1
Wireless node 2	192.169.0.4	192.169.0.4	255.255.255.255	1	1

9. In NetSim GUI Application throughput vs time and Link throughput vs time plots should be enabled.
10. Run the simulation for 60 seconds.

4.10.11 Output and Plot

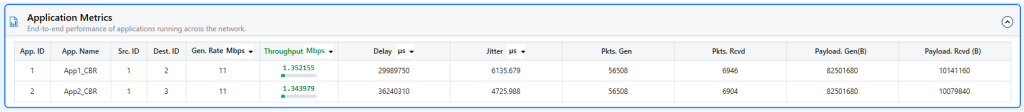


Figure 4-84: *Application Metrics window.*

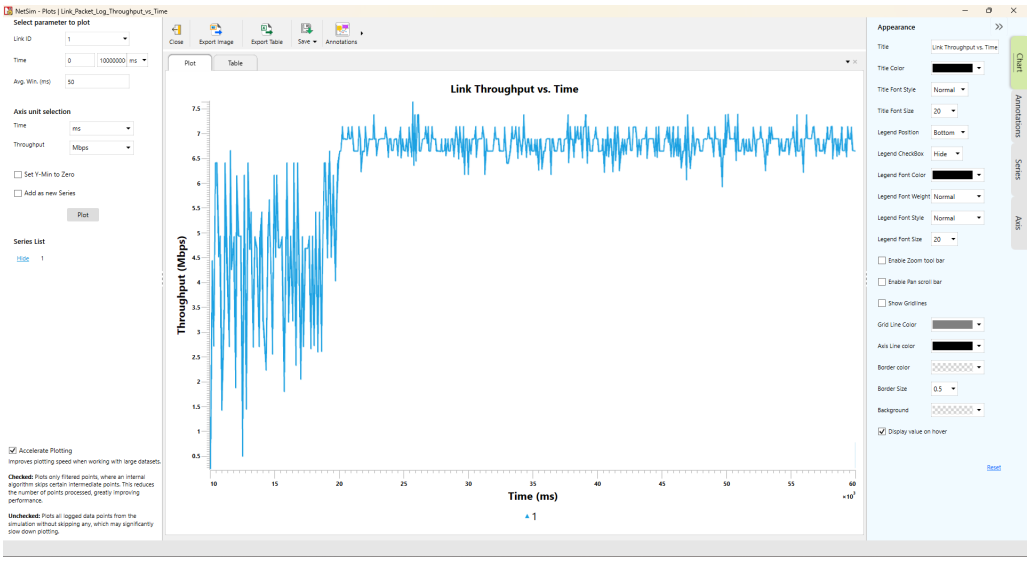


Figure 4-85: *NetSim Link Throughput plot window.*

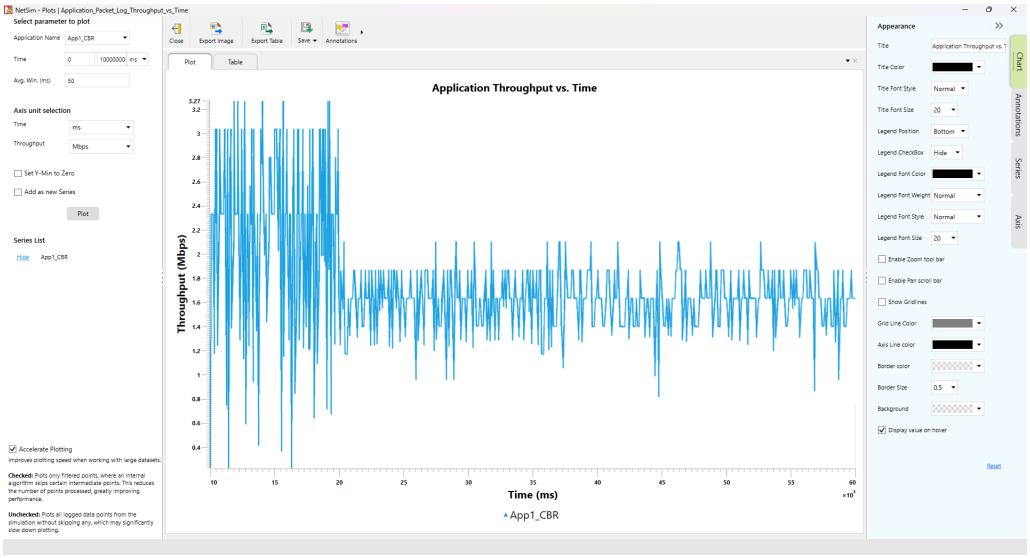


Figure 4-86: *Application 1 Throughput Plot window.*

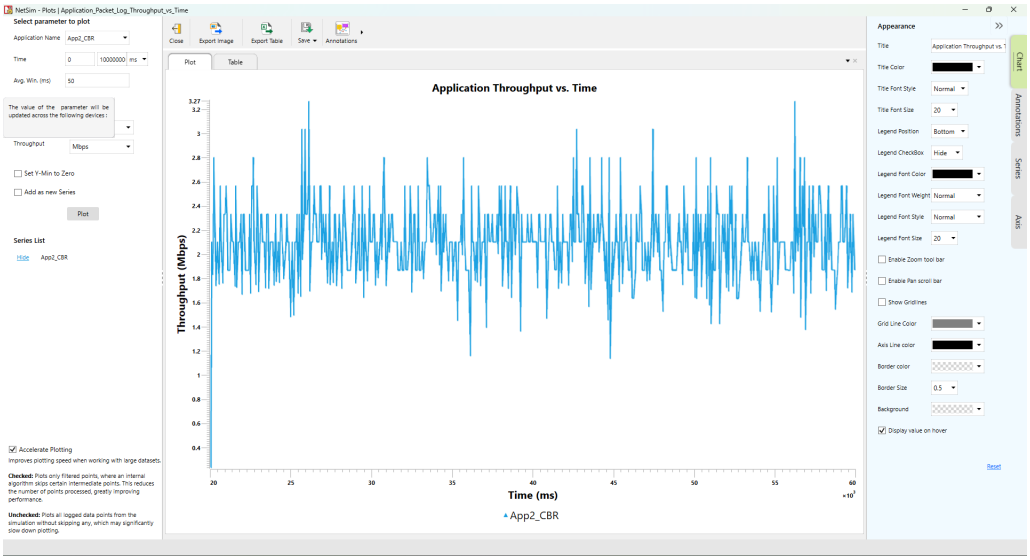


Figure 4-87: *Application 2 Throughput Plot window.*

4.10.12 Case 6: Range Based Pathloss. Nodes within range and beyond range. Broadcast Application

The following network diagram illustrates what the NetSim UI displays when you open the example configuration file.

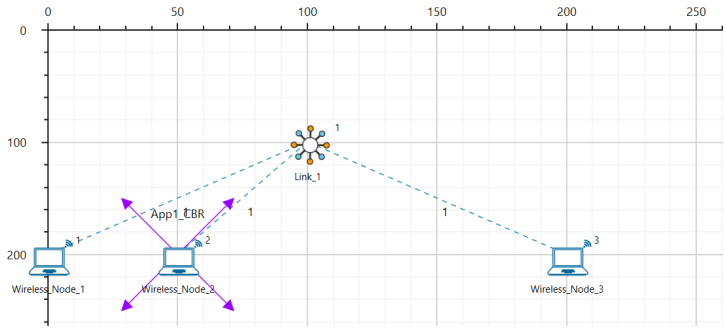


Figure 4-88: *Scenario for studying the range based pathloss within range out-of-range with broadcast application*

Network Settings

1. Set grid length as 600 × 300 from grid setting property panel on the right. This needs to be done before any device is placed on the grid.
2. Drop 3 wireless nodes and 1 ad hoc link in grid environment.
3. Device placements of wireless nodes are:

Table 4-53: *Device Placements*

Device	X-Coordinate	Y-Coordinate
Wireless Node-1	0	200
Wireless Node-2	50	200
Wireless Node-3	200	200

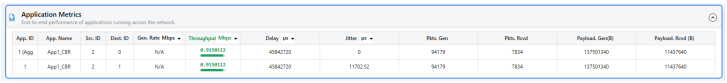
4. Click on link and expand right-hand side property panel and set Channel characteristics: Pathloss, Pathloss model to Range based, range(m)=100 under medium property.
5. Click on wireless node and expand right-hand side property panel and set Mobility model: No-mobility for all the nodes under position property and set Interface-1(wireless), Data link layer, Medium access protocol to DCF.
6. Click on the set traffic tab present in the top ribbon, create broadcast CBR application with 11Mbps generation rate (Set packet size: 1460, Inter arrival time: 1061.81 μs) from wireless node-2. Set Transport protocol to UDP and set start time to zero.
7. Run the simulation for 100 seconds.

Output

PACKET_ID	SEGMENT_ID	PACKET_TYPE	CONTROL_PACKET_TYPE/APP_NAME	SOURCE_ID	DESTINATION_ID	TRANSMITTER_ID	RECEIVER_ID
1	0	CBR	App1_CBR	NODE-2	Broadcast-0	NODE-2	NODE-1
2	0	CBR	App1_CBR	NODE-2	Broadcast-0	NODE-2	NODE-1
3	0	CBR	App1_CBR	NODE-2	Broadcast-0	NODE-2	NODE-1
4	0	CBR	App1_CBR	NODE-2	Broadcast-0	NODE-2	NODE-1
5	0	CBR	App1_CBR	NODE-2	Broadcast-0	NODE-2	NODE-1
6	0	CBR	App1_CBR	NODE-2	Broadcast-0	NODE-2	NODE-1
7	0	CBR	App1_CBR	NODE-2	Broadcast-0	NODE-2	NODE-1
8	0	CBR	App1_CBR	NODE-2	Broadcast-0	NODE-2	NODE-1
9	0	CBR	App1_CBR	NODE-2	Broadcast-0	NODE-2	NODE-1
10	0	CBR	App1_CBR	NODE-2	Broadcast-0	NODE-2	NODE-1
11	0	CBR	App1_CBR	NODE-2	Broadcast-0	NODE-2	NODE-1
12	0	CBR	App1_CBR	NODE-2	Broadcast-0	NODE-2	NODE-1
13	0	CBR	App1_CBR	NODE-2	Broadcast-0	NODE-2	NODE-1
14	0	CBR	App1_CBR	NODE-2	Broadcast-0	NODE-2	NODE-1
15	0	CBR	App1_CBR	NODE-2	Broadcast-0	NODE-2	NODE-1
16	0	CBR	App1_CBR	NODE-2	Broadcast-0	NODE-2	NODE-1
17	0	CBR	App1_CBR	NODE-2	Broadcast-0	NODE-2	NODE-1
18	0	CBR	App1_CBR	NODE-2	Broadcast-0	NODE-2	NODE-1
19	0	CBR	App1_CBR	NODE-2	Broadcast-0	NODE-2	NODE-1
20	0	CBR	App1_CBR	NODE-2	Broadcast-0	NODE-2	NODE-1
21	0	CBR	App1_CBR	NODE-2	Broadcast-0	NODE-2	NODE-1
22	0	CBR	App1_CBR	NODE-2	Broadcast-0	NODE-2	NODE-1

Figure 4-89: *Data transmission from WN1 to WN2*

From the packet trace analysis, it is evident that data transmission occurs between wireless node 1 (WN1) and wireless node 2 (WN2), but not between WN2 and WN3. This outcome is due to the broadcast application configured from WN2, with a range-based path loss setting of 100 meters. Since WN1 is within this range of WN2, data transmission is successfully occurring. However, WN3 is positioned outside the 100-meter range, resulting in no data transmission between WN2 and WN3.



App. ID	App. Name	Sec. ID	Dest. ID	Gen. Rate Mbps	Throughput Mbps	Delay µs	Jitter µs	Pkts. Gen	Pkts. Rcvd	Payload. Gen(B)	Payload. Rcvd (B)
1 (Agg	App1_CBR	2	0	N/A	0.959812	45842720	0	94179	7834	137501340	11437640
1	App1_CBR	2	1	N/A	0.959812	45842720	11702.52	94179	7834	137501340	11437640

Figure 4-90: *Application Metrics window.*

4.10.13 Case 7: Range Based Pathloss. Transmission Failure due to Interference.

The following network diagram illustrates what the NetSim UI displays when you open the example configuration file.

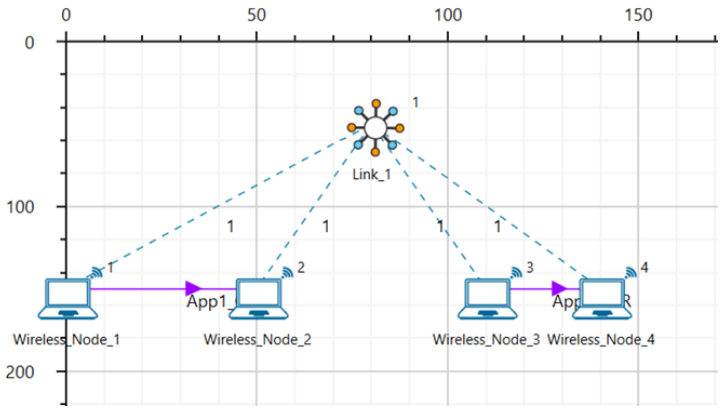


Figure 4-91: *Scenario for studying the range based pathloss Transmission failure due to interference from neighboring transmissions*

Network Settings

1. Set grid length as 400 × 200 from grid setting property panel on the right. This needs to be done before any device is placed on the grid.
2. Drop 4 wireless nodes and 1 Ad hoc link in grid environment.
3. Device placements of wireless nodes are:

Table 4-54: *Device Placements*

Device	X-Coordinate	Y-Coordinate
Wireless Node-1	0	150
Wireless Node-2	50	150
Wireless Node-3	110	150
Wireless Node-4	140	150

4. Click on link and expand right-hand side property panel and set Channel Characteristics: Pathloss, Pathloss model to Range Based, Range(m)=100.
5. Click on Wireless node and expand right-hand side property panel and set Mobility Model: No-Mobility for all the nodes under position property and set Interface-(Wireless), Data Link Layer, Medium access protocol to DCF.

6. Click on the set traffic tab present in the top ribbon, create CBR application with 11Mbps generation rate (Set Packet size: 1460, Inter Arrival Time: 1061.81 μs) from Wireless Node-1 to Wireless Node-2 and Wireless Node-3 to Wireless Node-4. Set Transport Protocol to UDP and Set start time to zero.
7. Run the Simulation for 100 seconds.

Output

In this scenario we have traffic from WN1 to WN2 and from WN3 to WN4. Note however that WN2 is within the range of WN3. In this scenario there is no successful data transmission between WN1 to WN 2. This is because WN2 is within the range of WN 3 which is transmitting to WN4. Packets sent from WN1 cannot be successfully received (decoded) at WN2 due to the interference of the WN3 to WN4 transmission.

Application Metrics
End-to-end performance of applications running across the network.

App. ID	App. Name	Src. ID	Dest. ID	Gen. Rate Mbps	Throughput Mbps	Delay µs	Jitter µs	Pkts. Gen	Pkts. Rcvd	Payload. Gen(B)	Payload. Rcvd (B)
1	App1_CBR	1	2	11	0	0	0	94179	0	137501340	0
2	App2_CBR	3	4	11	5.918723	23101140	911.499	94179	50674	137501340	73984040

Figure 4-92: *Application Metrics. We see that the first application has NIL throughput while the second application sees throughput.*

4.10.14 Case 8: Range Based Pathloss. Carrier sense (CS) blocking due to neighboring transmissions.

The following network diagram illustrates what the NetSim UI displays when you open the example configuration file.

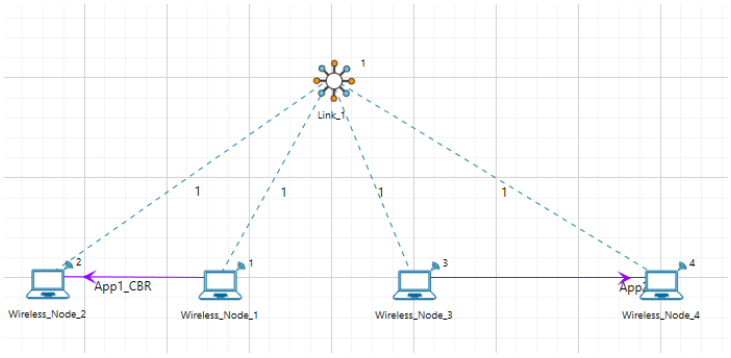


Figure 4-93: *Scenario for studying the range based pathloss Carrier sense (CS) Blocking due to neighboring transmissions.*

Network Settings

1. Set grid length as 400 × 200 from grid setting property panel on the right. This needs to be done before any device is placed on the grid.
2. Drop 4 wireless nodes and 1 Ad hoc link in Grid environment.
3. Device placements of wireless nodes are:

Table 4-55: *Device Placements*

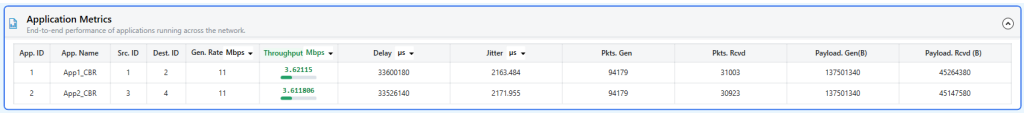
Device	X-Coordinate	Y-Coordinate
Wireless Node-1	50	150
Wireless Node-2	0	150
Wireless Node-3	100	150
Wireless Node-4	160	150

4. Click on link and expand right-hand side property panel and set Channel Characteristics: Pathloss, Pathloss model to Range Based, Range(m)=100.
5. Click on Wireless node and expand right-hand side property panel and set Mobility Model: No-Mobility for all the nodes under position property and set Interface-(Wireless), Data Link Layer, Medium access protocol to DCF.
6. Click on the set traffic tab present in the top ribbon, create CBR Application with 11Mbps generation rate (Set Packet size: 1460, Inter Arrival Time: 1061.81 μs) from Wireless Node-1 to Wireless Node-2 and Wireless Node-3 to Wireless Node-4. Set Transport Protocol to UDP and Set start time to zero.
7. Run the Simulation for 100 seconds.

Output

We observe that Wireless Node 1 (WN1) is within the range of Wireless Node 2 (WN2) and Wireless Node 3 (WN3), and WN3 is also within range of Wireless Node 4 (WN4). We have configured one application from WN1 to WN2 and another from WN3 to WN4.

In this scenario, WN1 is within the carrier sense range of both WN3 and WN4. Consequently, when WN3 transmits data to WN4, WN1 will delay its transmission due to carrier sense blocking, as it detects the active transmission nearby. Similarly, when WN1 is transmitting data to WN2, WN3 will wait before transmitting to WN4, as it is within the carrier sense range of WN1.


Application Metrics
End-to-end performance of applications running across the network.

App. ID	App. Name	Src. ID	Dest. ID	Gen. Rate Mbps	Throughput Mbps	Delay µs	Jitter µs	Pkts. Gen	Pkts. Rcvd	Payload. Gen(B)	Payload. Rcvd (B)
1	App1_CBR	1	2	11	3.62115	33600180	2163.484	94179	31003	137501340	45264380
2	App2_CBR	3	4	11	3.611806	33526140	2171.955	94179	30923	137501340	45147580

Figure 4-94: *Application Metrics window*

On the other hand, with only WN1-WN2 transmission the obtained throughput is 5.92 Mbps and with only WN3-WN4 transmission the throughput is 5.91 Mbps. We see that when both transmissions are occurring simultaneously the individual throughputs drop to 3.62 Mbps and 3.61 Mbps due to carrier sense blocking.

4.11 OLSRv2 Multi-cluster MANET

4.11.1 Scenario Overview

The OLSRv2 Multi-cluster MANET example is a 15-node pure wireless MANET simulation that demonstrates the key features of the Optimized Link State Routing Protocol Version 2 (OLSRv2), as implemented in NetSim Pro v15.1.

The network is arranged as three clusters connected by dedicated relay nodes. There is no wired infrastructure; every node is a wireless ad-hoc peer. The scenario uses a deterministic three-phase design: the topology is stable during each phase, with a single, precisely controlled node movement triggering each phase transition. This eliminates randomness and makes every protocol event predictable and verifiable.

Propagation Model – Range-Based Pathloss

The scenario uses range-based pathloss with a fixed range of 250 m. This is a deterministic, binary propagation model:

- Two nodes within 250 m of each other can communicate (link exists, perfect quality).
- Two nodes farther than 250 m apart cannot communicate (no link).

Configured via the wireless link's <`MEDIUM PROPERTY`> element:

```
PATHLOSS MODEL="RANGE BASED" RANGE="250"
```

The range applies globally to all nodes on the same wireless link. Combined with file-based mobility, the topology at every instant is fully determined by the node coordinates.

Mobility Model – File-Based

Instead of random mobility, this scenario uses file-based mobility (`FILE BASED MOBILITY`). A CSV file specifies the exact position of each moving node at exact times. Positions change instantaneously at the specified time with no interpolation and no randomness. Only Node 5 moves. All other 14 nodes are static. The mobility file contains three entries that define the three phases.

What This Example Demonstrates

- Flooding MPR and Routing MPR selection under varying willingness values.
- The catastrophic effect of `WILL_NEVER` when the willing bridge fails.
- The effect of TC hop limit truncation.
- Triggered TC messages and route recomputation after a deterministic topology change.
- Energy consumption differences across node roles.
- The full NHDP neighbour discovery lifecycle.
- Clean before, during, and after metrics for each phase.

Simulation Configuration

Use the following simulation configuration for the OLSRv2 Multi-cluster MANET example:

1. 15 wireless nodes in a MANET scenario with OLSRv2 as the routing protocol.
2. Wireless link properties:
   - Standard: 802.11g
   - Pathloss model: Range Based
   - Range: 250 m
   - Channel characteristics: Pathloss only
   - Propagation medium: Air
3. All 15 nodes placed at the Phase 1 coordinates shown below.
4. Willingness per node:
   - Nodes 7 and 13: `WILL_FLOODING = WILL_ROUTING = 15`
   - Node 6: `WILL_FLOODING = WILL_ROUTING = 0`
   - All others: default value 7
5. `TC HOP LIMIT = 4` on Node 5 only.

6. Mobility file: `ConfigSupport\mobility.csv`.
7. Node 5 mobility: `FILE BASED MOBILITY`. All other nodes: `NO MOBILITY`.
8. Four CBR/UDP flows with start time $t = 20$ s.
9. Link quality disabled.
10. Simulation time: 300 s.

4.11.2 Network Topology – Phase 1 (Baseline)

Parameter Configuration

Table 4-56: *Configuration Parameters for the Scenario*

Parameter	Value	Rationale
Hello Interval	2 s	Default
Hello Min Interval	0.5 s	Default (HELLO INTERVAL / 4)
HP Maxjitter	0.5 s	Default (HELLO INTERVAL / 4)
TC Interval	5 s	Default
TC Min Interval	1.25 s	Default (TC INTERVAL / 4)
TP Maxjitter	0.5 s	Same as HP Maxjitter
TC Hop Limit	255 (Node 5: 4)	Demonstrates hop-limit truncation
T hold time	15 s	Default (3 × TC INTERVAL)
A hold time	15 s	Default
Link Quality	Disabled	Default; not useful with range-based pathloss
Pathloss Model	Range-Based	Deterministic binary model
Range	250 m	Produces the designed topology
Wireless Standard	802.11g	Typical MANET radio
Mobility (Node 5)	File-Based	See the mobility listing below
Mobility (all others)	No Mobility	Static nodes
Simulation Time	300 s	Three phases with ample measurement windows

Topology Diagram

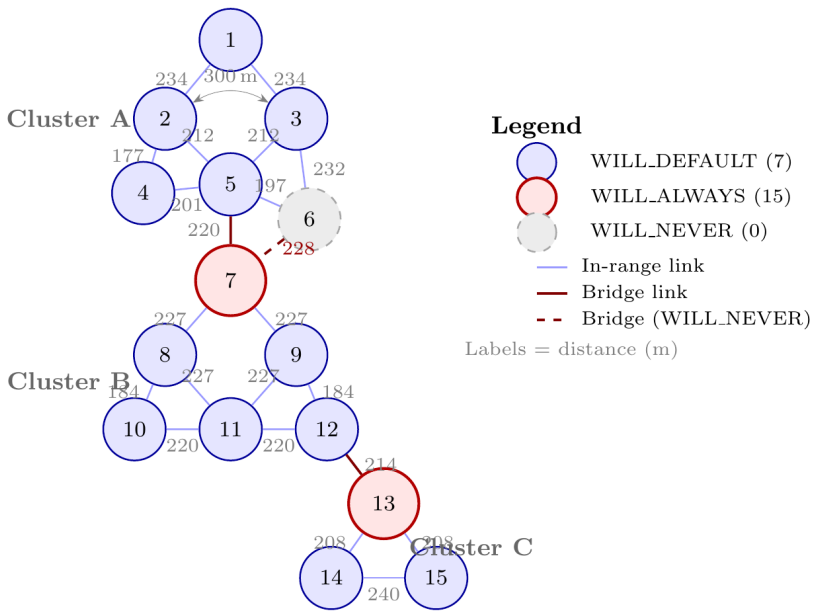


Figure 4-95: *Phase 1 topology (baseline). 22 links. All links are within the 250 m range. The dashed red link 6–7 exists physically but Node 6's* `WILL_NEVER` *prevents relaying.*

Node Placement

Table 4-57: *Node Positions (Phase 1), Roles, and Configuration*

Node	X	Y	Role	Willingness
1	300	1280	Cluster A hub	DEFAULT (7)
2	150	1100	Cluster A core	DEFAULT (7)
3	450	1100	Cluster A core	DEFAULT (7)
4	100	930	Cluster A edge	DEFAULT (7)
5	300	950	Cluster A / bridge to B	DEFAULT (7)
6	480	870	Cluster A / alt. bridge	NEVER (0)
7	300	730	Relay A–B	ALWAYS (15)
8	150	560	Cluster B core	DEFAULT (7)
9	450	560	Cluster B core	DEFAULT (7)
10	80	390	Cluster B edge	DEFAULT (7)
11	300	390	Cluster B edge	DEFAULT (7)
12	520	390	Cluster B / bridge to C	DEFAULT (7)
13	650	220	Relay B–C	ALWAYS (15)
14	530	50	Cluster C	DEFAULT (7)
15	770	50	Cluster C	DEFAULT (7)

Table 4-58: *Phase 1 Links (22 Total). All Distances Computed from the Node Coordinates*

Region	Link	Distance (m)	Note
Cluster A	1–2	234	
Cluster A	1–3	234	
Cluster A	2–4	177	
Cluster A	2–5	212	
Cluster A	3–5	212	
Cluster A	3–6	232	
Cluster A	4–5	201	
Cluster A	5–6	197	
Bridge A–B	5–7	220	Primary bridge
Bridge A–B	6–7	228	`WILL_NEVER`; cannot relay
Cluster B	7–8	227	
Cluster B	7–9	227	
Cluster B	8–10	184	
Cluster B	8–11	227	
Cluster B	9–11	227	
Cluster B	9–12	184	
Cluster B	10–11	220	
Cluster B	11–12	220	
Bridge B–C	12–13	214	Sole bridge
Cluster C	13–14	208	
Cluster C	13–15	208	
Cluster C	14–15	240	

Adjacency List

Table 4-59: *Key Non-Links Greater Than 250 m*

Pair	Distance (m)	Why It Matters
2–3	300	Forces Node 1 to need both 2 and 3 as MPRs
1–5	330	Node 1 cannot reach the bridge directly
4–7	283	No shortcut from Cluster A edge to relay
6–8	453	`WILL_NEVER` node isolated from Cluster B
8–9	300	Cluster B cores are not directly connected
7–11	340	Relay cannot skip the Cluster B core
9–13	394	No shortcut from Cluster B to relay C

4.11.3 Simulation Phases

The simulation is divided into three phases, each with a stable topology. Transitions between phases are triggered by a single, instantaneous position change of Node 5.

Table 4-60: *Simulation Phases*

Phase	Time Window	Node 5 Position	Topology State
1 – Baseline	$t = 0$ to $t = 50$ s	$(300, 950)$	Full connectivity (22 links)
2 – Bridge Break	$t = 50$ to $t = 150$ s	$(300, 1000)$	A–B disconnected (21 links)
3 – Recovery	$t = 150$ to $t = 300$ s	$(300, 950)$	Full connectivity restored (22 links)

Mobility File

The entire mobility specification is a three-line CSV file. Only Node 5 has entries; all other nodes use `NO MOBILITY`.

```
#Time(s),Device ID,X,Y,Z
0,5,300,950,0
50,5,300,1000,0
150,5,300,950,0
```

Node 5 is configured with:

```
<MOBILITY MODEL="FILE BASED MOBILITY" OPEN MOBILITY FILE="ConfigSupport\mobility.csv" />
```

Phase 1 – Baseline ($t = 0$ to $50\,s$)

Topology: the full 22-link network. Node 5 at $(300, 950)$ is within 220 m of Node 7, so the primary A–B bridge is active.

Protocol activity:

1. $t = 0$ to 16 s: NHDP convergence. HELLO messages propagate hop by hop and links transition `PENDING` → `HEARD` → `SYMMETRIC`. With `HELLO INTERVAL = 2 s` and a 7-hop diameter, full symmetric neighbour discovery completes by approximately 16 s.
2. $t = 5$ to 20 s: TC propagation. The first TC messages are generated at approximately 5 s. They flood via MPRs across the 7-hop chain, and routing tables populate progressively.
3. $t = 20$ s: all application flows start. The network is in steady state.
4. $t = 20$ to 50 s: baseline measurement window (30 s).

Steady-state routing paths:

Table 4-61: *Steady-State Routing Paths in Phase 1*

Flow	Path	Hops	Status
F1 (10→15)	10→11→12→13→15	4	Active
F2 (4→11)	4→5→7→8→11 (or via 9)	4	Active
F3 (14→1)	14→13→12→9→7→5→2→1	7	Active
F4 (3→8)	3→5→7→8	3	Active

Phase 2 – Bridge Break ($t = 50$ to $150\,s$)

At $t = 50$ s, Node 5 moves instantaneously from $(300, 950)$ to $(300, 1000)$, a shift of 50 m north.

Table 4-62: *Link Impact During the Bridge Break*

Link	Before (m)	After (m)	Status	Comment
5–7	220	270	BROKEN	Exceeds 250 m
5–2	212	180	OK	Closer
5–3	212	180	OK	Closer
5–4	201	212	OK	Slightly farther
5–6	197	222	OK	Slightly farther
1–5	330	280	Still out	Still greater than 250 m

The `WILL_NEVER` Trap

After the bridge break, the only physical link between Cluster A and Node 7 is 6↔7 (228 m, still within range). But Node 6 has `WILL_NEVER = 0`:

- Node 6 is not an MPR, so it does not relay TC messages.
- Node 6 does not forward data packets.
- The link 6–7 is physically present but useless for transit.

As a result, Clusters A and B+C become disconnected for all transit traffic. This is the most dramatic demonstration of `WILL_NEVER`: a single willing bridge fails, and the physically available alternative cannot compensate.

Protocol Events at $t = 50$ s

1. Node 5 and Node 7 detect link loss via NHDP (HELLO timeout, approximately 6 s = `H HOLD TIME`).
2. Link status transitions `SYMMETRIC` → `HEARD` → `LOST`.
3. Node 7 increments its ANSN (Advertised Neighbour Sequence Number) and generates a triggered TC message.
4. Node 5 also generates a triggered TC because it lost Node 7 as a neighbour.
5. The triggered TCs propagate via MPRs: Node 7's TC reaches Clusters B and C, while Node 5's TC reaches Cluster A. Neither TC crosses the broken bridge.
6. Every node that receives the TC recomputes its routing table using Dijkstra.
7. Routes between Clusters A and B/C are removed because no valid next hop exists.

Table 4-63: *Detection Timeline for the Bridge Break*

Time	Event
$t = 50$ s	Node 5 moves; link 5–7 physically breaks
$t \approx 52$ s	Next HELLO from Node 5 is not heard by Node 7, and vice versa
$t \approx 56$ s	L_SYM time expires on both ends, moving to `HEARD`
$t \approx 56$ s	`H HOLD TIME` (6 s) expires, moving to `LOST`
$t \approx 56$ s	Triggered TC messages from Nodes 5 and 7
$t \approx 58$ to 65 s	TC propagation and route settling complete across the remaining topology
$t \approx 58$ to 65 s	Routing tables update and flows F2, F3, and F4 begin dropping packets after reconvergence

Table 4-64: *Flow Status During Phase 2*

Flow	Status	Reason
F1 (10→15)	Active	Path 10→11→12→13→15 remains within B+C
F2 (4→11)	BROKEN	Requires the A→B bridge; no willing path exists
F3 (14→1)	BROKEN	Requires the B→A bridge; no willing path exists
F4 (3→8)	BROKEN	Requires A→B bridging; Node 6 cannot relay

Phase 3 – Recovery ($t = 150$ to $300\,s$)

At $t = 150$ s, Node 5 returns instantaneously to $(300, 950)$ and link 5–7 reforms.

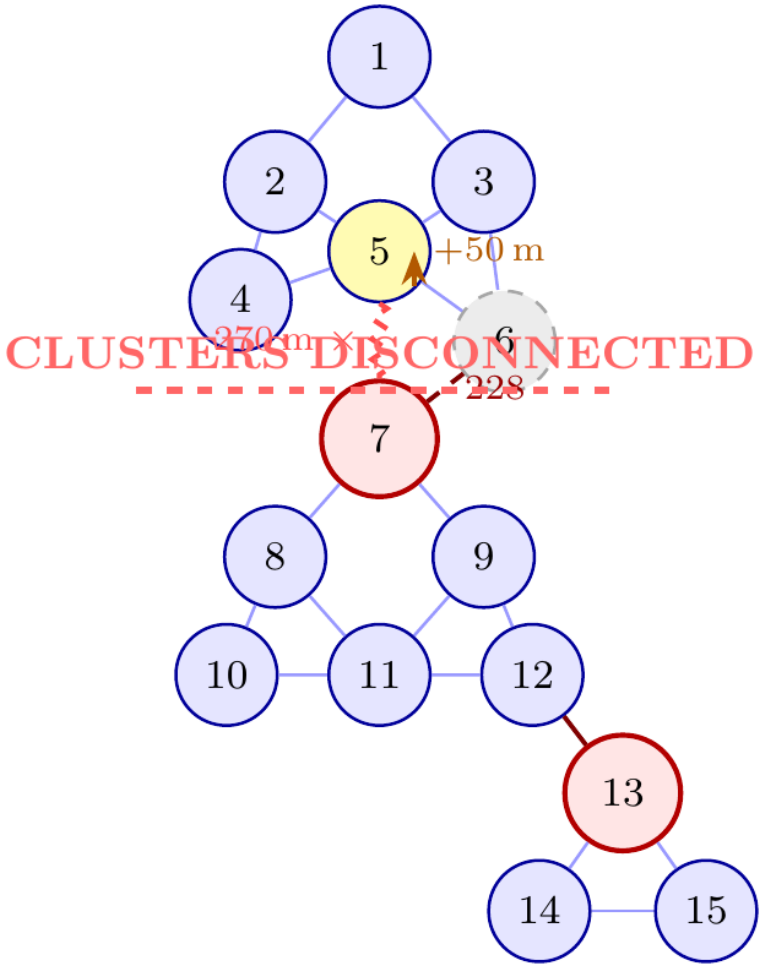


Figure 4-96: *Phase 2 topology. Node 5 moves 50 m north (highlighted yellow) and link 5–7 breaks because 270 m exceeds the 250 m range. The only remaining A–B connection is 6–7, but Node 6's* `WILL_NEVER` *blocks all relaying.*

Protocol Events at $t = 150$ s

1. Node 5's HELLO reaches Node 7 again. The link transitions `LOST` → `HEARD` → `SYMMETRIC`, requiring two HELLO exchanges and approximately 4 s.
2. Both nodes generate triggered TC messages announcing the restored neighbour.
3. Triggered TCs propagate via MPRs across both sides of the network.
4. Routing tables are recomputed with the restored bridge, and routes between Clusters A and B/C reappear after the protocol settles.
5. All four flows recover.

Table 4-65: *Recovery Timeline*

Time	Event
$t = 150$ s	Node 5 moves back; link 5–7 becomes physically available
$t \approx 152$ s	Node 5's HELLO is heard by Node 7, moving to `HEARD`
$t \approx 154$ s	Symmetric confirmation through the second HELLO, moving to `SYMMETRIC`
$t \approx 154$ s	Triggered TC messages from Nodes 5 and 7
$t \approx 156$ to 165 s	TC propagation and route settling continue
$t \approx 156$ to 165 s	Routing tables restore and flows F2, F3, and F4 resume delivery

Measurement window: $t = 165$ to 300 s (135 s of restored steady state).

Expected outcome: Phase 3 metrics should closely match Phase 1 metrics for all four flows, confirming that the protocol fully recovers from the disruption.

4.11.4 Features Exercised

MPR Selection – Flooding and Routing

Node 1's MPR Selection (all phases). Node 1's one-hop neighbours are {2, 3}. The two-hop set is:

- via Node 2: {4, 5}
- via Node 3: {5, 6}

The unique two-hop set is {4, 5, 6}. Node 2 covers {4, 5}, and Node 3 covers {5, 6}. Neither alone covers all three, so both are selected as MPRs.

Node 5's MPR Selection (Phases 1 and 3). The one-hop neighbours are {2, 3, 4, 6, 7}. The two-hop set is {1, 8, 9}. Node 2 or Node 3 covers {1}, while Node 7 covers {8, 9} and has `WILL_ALWAYS`, guaranteeing selection. Node 6 provides no unique two-hop coverage and has `WILL_NEVER`, so it is never selected.

Node 5's MPR Selection (Phase 2). Node 5 loses Node 7 as a neighbour. The new one-hop set is {2, 3, 4, 6}. The new two-hop set is {1, 7}, with Node 7 learned as a two-hop neighbour via Node 6. The MPR set shrinks relative to Phase 1.

Flooding vs. Routing MPRs. OLSRv2 maintains separate flooding and routing MPR sets (RFC 7181 Sections 18.1 and 18.4). Both are independently configurable via `WILL_FLOODING` and `WILL_ROUTING`.

Willingness Variation and the WILL_NEVER Trap

WILL_ALWAYS (15) is configured on Nodes 7 and 13, guaranteeing MPR selection. WILL_NEVER (0) is configured on Node 6, so it never relays. In Phase 1 this is merely observable; in Phase 2 it becomes critical because the only physical A–B link is 6–7, but WILL_NEVER renders it useless for transit, disconnecting the clusters. All other nodes use WILL_DEFAULT (7).

TC Hop Limit Truncation

Observe the effect of TC HOP LIMIT for Node 5 by comparing these two cases:

- TC HOP LIMIT = 255: Node 5's TC propagates normally through the connected topology.
- TC HOP LIMIT = 4: Node 5's TC is forwarded only up to the fourth hop. The node at hop 4 may receive the TC, but it does not retransmit it further.

Triggered TC Messages

Periodic TCs are generated every 5 s. However, topology changes trigger immediate TC messages, subject to TP MAXJITTER = 0.5 s. This scenario produces two triggered-TC bursts:

1. Around $t \approx 56$ to 65 s, after the link 5–7 loss.
2. Around $t \approx 154$ to 165 s, after the link 5–7 restoration.

NHDP Neighbour Discovery Lifecycle

Neighbour discovery is observable in three contexts:

1. Initial convergence ($t = 0$ to 16 s): all links transition PENDING → HEARD → SYMMETRIC.
2. Link loss (around $t \approx 52$ to 65 s): link 5–7 transitions SYMMETRIC → HEARD → LOST.
3. Link recovery (around $t \approx 150$ to 165 s): link 5–7 transitions LOST → HEARD → SYMMETRIC.

Energy Consumption by Role

All nodes use the default NetSim energy model (TX: 316.7 mA, RX: 261.1 mA, Idle: 227 mA, Voltage: 3.6 V).

Phase-dependent expected results:

- Node 7 (WILL_ALWAYS relay) has the highest TX energy in Phases 1 and 3 because it relays all cross-cluster traffic.
- In Phase 2, Node 7's TX energy drops because it no longer relays A→B traffic; only B→C traffic remains.
- Node 6 (WILL_NEVER) has the lowest TX energy in all phases.
- Node 13 remains consistently active as the B–C bridge, unaffected by Node 5's movement.

4.11.5 Traffic Flows

Four concurrent CBR/UDP flows start at $t = 20$ s and run for the remainder of the simulation.

Table 4-66: *Application Traffic Flows*

Flow	Src	Dst	Type	Packet Size	Rate	Purpose
F1	10	15	CBR/UDP	512 B	10 pkt/s	Entirely within B+C (4 hops); unaffected by the bridge break and serves as a control flow
F2	4	11	CBR/UDP	512 B	10 pkt/s	Crosses A→B via Node 7; breaks in Phase 2
F3	14	1	CBR/UDP	512 B	5 pkt/s	Longest path (7 hops); breaks in Phase 2
F4	3	8	CBR/UDP	256 B	20 pkt/s	`WILL_NEVER` bypass demonstration (3 hops); breaks in Phase 2

Per-Phase Flow Behaviour

F1 remains active across all three phases because its path 10→11→12→13→15 never crosses the A–B bridge. F2, F3, and F4 break during Phase 2 and recover in Phase 3 after reconvergence. The windows $t = 50$ to 65 s and $t = 150$ to 165 s are transient intervals rather than steady-state measurement periods.

4.11.6 Simulation Results – Phase by Phase

Phase 1 Observations (Baseline, $t = 20$ to $50\,s$)

Observation	Where to Look	Expected Result
MPR tables	OLSR link metrics log	Node 1 MPRs = {2, 3}. Node 5 MPRs include Node 7 (`WILL_ALWAYS`). Node 6 never appears as an MPR for any node.
WILL_NEVER effect	Routing tables	Node 6 is reachable as a destination but never used as a next hop. Flow F4 uses 3→5→7→8, not 3→6→7→8.
TC hop limit	Packet trace	Node 5's TCs are limited to 4 hops. The node at hop 4 may receive the TC but does not forward it further, so Nodes 14 and 15 are not reached directly. Other nodes' TCs propagate Node 5's information farther.
Full convergence	Routing table timestamps	All 15 nodes have complete routing tables by approximately $t = 16$ to 20 s.
All flows active	Application metrics	F1 remains active on its 4-hop path. F2–F4 are also active in the baseline phase on their respective paths.
Jitter	Packet trace	HELLO and TC transmission times vary by up to 0.5 s and 0.5 s, respectively, from the periodic schedule.

Phase 2 Observations (Bridge Break, $t = 50$ to $150\,s$)

Observation	Where to Look	Expected Result
Link 5–7 loss	NHDP logs	`SYMMETRIC` → `HEARD` → `LOST` on both Node 5 and Node 7 at approximately $t = 56$ s.

Triggered TC burst	Packet trace	Non-periodic TCs from Nodes 5 and 7 at approximately $t = 56$ s, clearly distinguishable from the 5 s periodic cadence.
ANSN increment	TC message headers	Nodes 5 and 7 increment ANSN in the triggered TCs.
MPR table change	OLSR link metrics log	Node 5's MPR set shrinks because it loses Node 7. Node 7's MPR set shrinks because it loses Node 5.
Route removal	Routing tables	All Cluster A nodes lose routes to Cluster B/C nodes and vice versa. Intra-cluster routes remain intact on both sides.
F1 unaffected	Application metrics	F1 remains active throughout because its path is entirely within B+C.
F2/F3/F4 broken	Application metrics	Delivery drops after approximately $t = 58$ to 65 s once reconvergence completes.
Node 7 energy drop	Energy logs	Node 7's TX energy rate decreases in Phase 2 because it no longer relays A→B data or TCs from Cluster A.
WILL_NEVER trap	Routing tables (Node 6)	Node 6 still has a `SYMMETRIC` link to Node 7, but its routing table shows no routes to Cluster B/C because Node 7's TCs are not relayed back via Node 6. Even if routes were present, Node 6 would still not forward.

Phase 3 Observations (Recovery, $t = 150$ to $300\,s$)

Observation	Where to Look	Expected Result
Link 5–7 recovery	NHDP logs	`LOST` → `HEARD` → `SYMMETRIC` on both ends at approximately $t = 154$ s.
Triggered TC burst	Packet trace	Second burst of non-periodic TCs from Nodes 5 and 7.
Route restoration	Routing tables	Cross-cluster routes reappear network-wide after recovery, typically around $t = 156$ to 165 s depending on settling time. Paths then match Phase 1 again.
F2/F3/F4 resume	Application metrics	Traffic resumes only after reconvergence. The interval around $t = 150$ to 165 s shows partial restoration while NHDP re-establishes the link and routing tables settle.
Metrics match Phase 1	All logs	Phase 3 steady-state metrics should closely match Phase 1, confirming full protocol recovery. Persistent differences indicate a convergence defect.

4.11.7 Running the Example in NetSim

Open the OLSRv2 Multi-cluster MANET example in NetSim Pro v15.1. The example is pre-configured with all node positions, willingness values, mobility, traffic flows, and protocol settings described in this section.

Observing Simulation Results and TC Hop Limit Effects

After running the simulation, the following outputs can be observed:

- OLSR link metrics log: displays MPR tables for each phase.
- Packet trace: observe the difference between the default hop limit (255) and Node 5 configured with hop limit 4, along with triggered TC bursts at $t \approx 56$ s and $t \approx 154$ s.
- Routing table snapshots: tables at different time instances, such as $t = 40$ s, $t = 100$ s, and $t = 200$ s, illustrate routing changes across phases.
- Application metrics: show delivery ratio and delay for each flow across phases.
- Energy logs: highlight variations in transmission energy across nodes, such as Node 7 versus Node 6.

TC Hop Limit

The TC Hop Limit controls how far a TC (Topology Control) message propagates in the network.

Steps to Verify the Effect

1. Run the simulation and open the Packet Trace window.
2. Apply the following filters:
   - Packet Type = `TC_MESSAGE`
   - Source Node = 5
3. Identify the TC messages generated by Node 5.
4. Track a selected TC packet as it propagates through neighbouring nodes across the network.

Expected Observations

- **For Node 5 (TC Hop Limit = 4):**
  - The TC message is forwarded only up to four hops from the source node.
  - When the hop count reaches 4, the packet is received by the node but is not forwarded further.
  - Nodes located beyond four hops do not receive this TC message directly.
- **For Other Nodes (Default TC Hop Limit = 255):**
  - TC messages propagate throughout the entire network without early termination.

Verification Method

Compare TC packets generated by Node 5 with TC packets generated by other nodes in the network. Only the TC messages originating from Node 5 will demonstrate early truncation due to the reduced hop limit, confirming the effect of the configured TC Hop Limit.

4.11.8 Conclusion

This example provides a fully deterministic evaluation of OLSRv2. Range-based pathloss and file-based mobility ensure that every topology change is precisely controlled and every protocol event is reproducible.

The three-phase design produces a clean before, during, and after comparison:

- Phase 1 establishes the baseline: MPR selection, TC propagation, multi-hop routing, and energy consumption under full connectivity.

- Phase 2 shows the protocol's response to a controlled disruption: triggered TC generation, route removal, ANSN updates, and the catastrophic impact of `WILL_NEVER` when the sole willing bridge fails.
- Phase 3 confirms complete recovery: link re-establishment, route restoration, and metric convergence back to baseline.

The control flow F1, which is unaffected by the bridge break, confirms that the disruption is localised to cross-cluster paths.

5 References

[1] "Mobile Ad hoc Networking (MANET): Routing Protocol Performance Issues and Evaluation Considerations," 1999. [Online]. Available: https://www.ietf.org/rfc/rfc2501.txt.

[2] D. Halperin, B. Greenstein, A. Sheth and D. Wetherall, "Demystifying 802.11n Power Consumption," 2010.

[3] "Ad hoc On-Demand Distance Vector (AODV) Routing," 2003. [Online]. Available: https://www.ietf.org/rfc/rfc3561.txt.

[4] N. M. a. E. F. A. Busson, "Analysis of the Multi-Point Relays selection in OLSR and Implications," Page No. 5, 2009.

[5] e. a. T. Clausen, "Optimized Link State Routing Protocol (OLSR)," RFC 3626, Section 3.5, October 2003.

[6] P.J.A.L.V Thomas Heide Clausen, "Comparative Study of Routing Protocols for Mobile Ad-hoc NETworks," 8 April 2010.

[7] "Optimized Link State Routing Protocol (OLSR)," 2003. [Online]. Available: https://www.ietf.org/rfc/rfc3626.txt.

[8] "IEEE 802.11-2012 – Standard for Wireless LAN," IEEE (Institute of Electrical and Electronics Engineers), 2012.

6 Latest FAQs

Up-to-date FAQs on NetSim's MANET library are available at

https://tetcos.freshdesk.com/support/solutions/folders/14000110331